



وزارت علوم، تحقیقات و فناوری
پژوهشگاه علوم و فناوری اطلاعات ایران
(ایرانداک)

مرحله آغازین طرح

«ایجاد و توسعه روبوداک: ربات پژوهشیار»

مجری:

آزاده محبی

همکاران اصلی:

عمار جلالی منش، رامین گلشایی، سهیل گنج‌فر

بهمن ۱۳۹۵



حقوق: پژوهشگاه علوم و فناوری اطلاعات ایران

طرح پژوهشی «مرحله آغازین طرح ایجاد و توسعه روبوداک: ربات پژوهشیار» پیرو قرارداد شماره ۳۳/د/۷۴۹ مورخ ۱۳۹۴/۱/۲۶ میان پژوهشگاه علوم و فناوری اطلاعات ایران (کارفرما) و خانم آزاده محبی (مجری) اجرا شده است. گزارش حاضر یک جلد از مستندات این طرح است.

این گزارش و تمامی حقوق مادی آن بر اساس قانون حمایت حقوق مولفان و مصنفان و هنرمندان، مصوب ۱۳۴۸ و اصلاحیه‌های بعدی آن و همچنین آیین‌نامه‌های اجرایی این قانون متعلق به پژوهشگاه علوم و فناوری اطلاعات ایران و هرگونه استفاده از تمامی یا پاره‌ای از آن، شامل: نقل قول، تکثیر، انتشار، کاربرد، نتایج، تکمیل و مانند آنها به صورت چاپی، الکترونیکی یا وسایل دیگر فقط با اجازه کتبی پژوهشگاه امکان‌پذیر است. نقل قول در حد هزار واژه در انتشارات علمی مانند کتاب و مقاله با درج اطلاعات کامل کتابشناختی، نیازی به مجوز پژوهشگاه ندارد.

صحت مندرجات گزارش بر عهده مجری طرح پژوهشی است.

شماره: ۲۳/۱۵۵۳۸
تاریخ: ۱۳۹۵/۱۲/۰۴
نماد: ندارد
پوست:

برنام‌نما

وزارت علوم، تحقیقات و فناوری
پژوهشگاه علوم و فناوری اطلاعات ایران
ایرانداک



پژوهشی، مدیریت دانش، آموزش

گواهی

انجام طرح پژوهشی:

- ◇ عنوان: مرحله آغازین طرح «ایجاد و توسعه روبوداک: رویات پژوهشیار»
- ◇ مجری: آزاده محبی
- ◇ همکاران: مهندس عمار جلالی‌منش، دکتر رامین گلشانی، دکتر سهیل گنج‌فر
- ◇ گروه پژوهشی: پرنیان افشار، امیر بادامچی، محمود بخشایش، فرنوش بیات، علی عالی‌حسینی، محمدرضا رمضانی، زهرا گمار
- ◇ شماره قرارداد: ۳۳/۴۴۹۴:۵
- ◇ تاریخ قرارداد: ۱۳۹۴/۴/۶:۵
- ◇ تاریخ شروع: ۱۳۹۴/۴/۸
- ◇ تاریخ پایان: ۱۳۹۵/۱۱/۱۳
- ◇ ناظر: دکتر محمود بابایی

در پژوهشگاه علوم و فناوری اطلاعات ایران گواهی می‌شود. گزارش پایانی این طرح، بر پایه داوری در نشست ۲۳۵ (تاریخ ۱۳۹۵/۱۲/۳) شورای پژوهش پژوهشگاه به تصویب رسیده است. لازم می‌دانم مراتب قدردانی خود را به مجری، همکاران و ناظر این طرح تقدیم دارم.

با سپاس
رویا پورتنقی
سرپرست معاونت پژوهش

مرحله آغازین طرح ایجاد و توسعه رובوداک: ربات پژوهشیار

مدیر طرح پژوهشی: آزاده محبی،

همکاران اصلی طرح پژوهشی: عمار جلالی منش، رامین گلشایی، سهیل گنجه‌فر

گروه پژوهشی: پرینان افشار، امیر بادامچی، محمود بخشایش، فرنوش بیات، علی عالی حسینی، محمدرضا رضانی،

زهرا گمار

پژوهشگاه علوم و فناوری اطلاعات ایران.

نشانی: تهران، خیابان انقلاب، چهارراه فلسطین، شماره ۱۰۹۰، طبقه ۴، اتاق ۴۰۴

سندوق پستی: ۱۳۱۵۷۷۳۳۱۴ تلفن: ۰۲۱-۶۶۴۹۴۹۸۰ (داخلی ۴۷۷)

وبگاه: <https://www.irandoc.ac.ir>

رایانامه: mohebi@irandoc.ac.ir

چکیده

یک پژوهشگر طی فرایند پژوهش، فعالیت‌های متنوعی را نظیر شناسایی و تحلیل مسئله پژوهش، خلق ایده‌های نوآورانه برای حل یک مسئله پژوهشی، طراحی پژوهش و انتخاب متدولوژی و ابزار مناسب، جمع‌آوری و سازماندهی اطلاعات و تحلیل آنها، و در نهایت تهیه گزارش‌های پژوهشی و انتشار آنها، انجام می‌دهد. وجود حجم وسیع اطلاعات در دسترس برای پژوهشگر و ابزارهای متنوع موجود برای به سرانجام رساندن هر یک از فعالیت‌های مزبور در فرایند پژوهش، باعث شده که فرایند پژوهش در بسیاری از اوقات با چالش‌های جدیدی مواجه شود. چالش‌هایی که عمدتاً ناشی از عدم امکان دسترسی سازمان‌یافته به اطلاعات پژوهشی، وجود ابزارهای جزیره‌ای و غیرمتمرکز برای تحلیل اطلاعات پژوهشی، عدم دسترسی پیوسته و متمرکز به اطلاعات و اخبار جدید در زمینه تازه‌های پژوهشی است. استفاده از یک دستیار ماشینی هوشمند، می‌تواند بخش مهمی از این چالش‌ها را مرتفع سازد. دستیار شخصی‌سازی شده هوشمند می‌تواند با توجه به علایق پژوهشی یک پژوهشگر و سوابق و اطلاعات پژوهشی وی، او را در بخش‌های مختلف فرایند پژوهش، یاری رساند. یکی از نمودهای دستیاران شخصی هوشمند، ربات‌های اجتماعی هستند که می‌توانند در قالب رفتارهای اجتماعی با یک فرد تعامل داشته باشند.

ربات اجتماعی، ربات خودکنترلی است که می‌تواند با انسان‌ها یا عامل^۱های خودمختار دیگر تعامل و ارتباط برقرار کند. این ربات‌ها در تعاملاتشان از قواعد و رفتارهای اجتماعی پیروی می‌کنند که برای انجام نقش خاصی در آنها تعبیه شده است. عمده کاربردهای ربات‌های اجتماعی در حال حاضر به حوزه‌های پزشکی (مانند کمک به افراد سالمند و کودکان معلول)، فروش و ارتباط با مشتری (مانند کمک به مشتری در فروشگاه‌ها برای ارائه خدمات بهتر و انتخاب محصولات) و آموزشی (مانند یادگیری و آموزش مفاهیم به کودکان) معطوف می‌شود. آنچه که در این طرح به طور خاص مورد توجه قرار می‌گیرد استفاده از امکانات اینگونه ربات‌ها برای هدف پژوهشیاری و مدیریت اطلاعات و یافته‌های علمی است.

روبوداک، یک ربات اجتماعی است که می‌تواند نقش یک منشی ماشینی را برای پژوهشگر ایفا نماید. در واقع، چشم‌انداز طرح پژوهشی ایجاد و توسعه روبوداک، توسعه یک ربات اجتماعی است که بتواند در مراحل مختلف فرایند پژوهش، در قالب یک دستیار ماشینی خدمات یکپارچه‌ای را برای پژوهشگر ارائه کند. طرح ایجاد و توسعه روبوداک، یک طرح میان‌رشته‌ای است که حوزه‌های پژوهشی نظیر زبانشناسی رایانشی، پردازش صوت و گفتار، مدیریت و تحلیل اطلاعات پژوهش، رباتیک و کنترل، و ارتباط انسان و ماشین را درگیر می‌نماید.

هدف از انجام طرح پژوهشی حاضر که مرحله آغازین از طرح ایجاد و توسعه روبوداک است، استفاده از یک پلتفرم رباتیک و سفارشی‌سازی آن برای توسعه بخشی از خدمات پژوهشی روبوداک است. خدماتی که روبوداک می‌تواند در این مرحله ارائه دهد، عبارتند از: جستجو در پایگاه داده گنج و گوگل اسکولار به صورت صوتی، با زبان فارسی و

¹Agent

انگلیسی، خواندن از روی متون فارسی و انگلیسی، و نمایش نتایج جستجو و دسته‌بندی موضوعی آنها در قالب نمودارهای مصورسازی اطلاعات. علاوه بر این موارد، روبوداک می‌تواند با کاربر ارتباط صوتی و تصویری داشته باشد و با استفاده از سیستم بینایی، کاربر را شناسایی کند و چند حرکت اولیه از جانب کاربر را تشخیص دهد. کلیه این خدمات در قالب یک بستر یکپارچه هوش مصنوعی با تلفیق و یکپارچه نمودن چند زیر سیستم در بستر اینترنت، از طریق یک ربات فیزیکی ارائه می‌شود. پلتفرم رباتیکی که برای توسعه روبوداک در این مرحله استفاده شده، پلتفرم ربات انسان‌نما Nao است. البته براساس نیازهای این مرحله از توسعه روبوداک، تغییراتی نیز در فیزیک ظاهری ربات، با استفاده از فناوری چاپ سه بعدی ایجاد شده است.

علاوه بر توسعه خدمات مزبور برای روبوداک در این مرحله، در انتهای گزارش این طرح، سبدهی از پروژه‌های پژوهشی تعریف شده که می‌تواند مسیر را برای توسعه سایر خدمات روبوداک در آینده نیز مشخص نماید.

کلیدواژه‌ها: ربات اجتماعی، تشخیص چهره، پردازش و تشخیص گفتار، جستجو و بازیابی اطلاعات، مصورسازی اطلاعات

فهرست مطالب

۱. کلیات پژوهش	۲
۱-۱. ربات فیزیکی؛ ضرورت یا امکان؟	۳
۱-۱-۱. آزمایشگاه‌های دورآزما	۴
۱-۱-۲. جسم اساس هوشمندی و تعامل موثر	۴
۲-۱. پیشینه کاربرد ربات‌های اجتماعی	۶
۳-۱. معرفی روبوداک	۸
۴-۱. ساختار گزارش طرح	۹
۲. خدمات روبوداک و مدل مفهومی	۱۲
۱-۲. طراحی خدمات روبوداک	۱۲
۱-۱-۲. خلق ایده	۱۲
۲-۱-۲. طراحی پژوهش	۱۳
۳-۱-۲. جمع‌آوری اطلاعات	۱۴
۴-۱-۲. سازمان‌دهی و تحلیل اطلاعات	۱۷
۵-۱-۲. تولید گزارش	۱۹
۶-۱-۲. انتشار	۲۱
۲-۲. قابلیت‌های اصلی ربات اجتماعی	۲۵
۳-۲. نیازمندی‌های اصلی برای سیستم روبوداک در مرحله آغازین	۲۷
۳. پلتفرم فیزیکی روبوداک	۳۱
۱-۳. بررسی انواع ربات‌های اجتماعی و قابلیت‌های آن‌ها	۳۱
۱-۱-۳. ربات Nao	۳۱
۲-۱-۳. ربات DARwIn-OP	۳۵
۳-۱-۳. ربات Pepper	۳۷

۳۹.....	۴-۱-۳. ربات Romeo
۴۱.....	۵-۱-۳. ربات Hubo II
۴۳.....	۶-۱-۳. ربات‌های سری HOVIS
۴۵.....	۷-۱-۳. ربات HOAP
۴۶.....	۸-۱-۳. ربات Maggie
۴۹.....	۹-۱-۳. ربات Sacarino
۵۱.....	۱۰-۱-۳. ربات Asimo
۵۲.....	۱۱-۱-۳. ربات Rabo
۵۴.....	۱۲-۱-۳. جمع‌بندی نتایج بررسی انواع پلتفرم‌ها
۵۵.....	۲-۳. اولویت‌بندی خدمات روبوداک
۵۷.....	۱-۲-۳. روش اولویت‌بندی خدمات
۵۹.....	۳-۳. انتخاب پلتفرم مناسب
۶۲.....	۴. تشریح قابلیت‌های پلتفرم Nao و سفارشی‌سازی آن
۶۲.....	۱-۴. مشخصات اصلی پلت فرم روبوداک
۷۳.....	۲-۴. سفارشی‌سازی ربات براساس قابلیت‌های مورد انتظار
۷۳.....	۱-۲-۴. الزامات برای دریافت ورودی‌ها
۷۳.....	۲-۲-۴. الزامات برای نمایش خروجی‌ها
۷۴.....	۳-۲-۴. الزامات برای پردازش‌ها
۷۴.....	۴-۲-۴. سیستم بینایی
۷۹.....	۵-۲-۴. میکروفون‌های روبوداک
۸۲.....	۶-۲-۴. پلتفرم روبوداک برای پردازش‌ها
۸۸.....	۷-۲-۴. پلتفرم روبوداک برای نمایش خروجی‌ها

۹۳	۸-۲-۴. بدنه فیزیکی ربات
۹۷	۳-۴. ارزیابی فیزیکی پلتفرم روبوداک با استفاده از شبیه سازی
۹۸	۴-۳-۱. اتصال یک قطعه خارجی به Nao در محیط ویباتر
۱۰۴	۴-۳-۲. بررسی پایداری روبوداک با استفاده از ویباتر
۱۱۱	۵. تشریح سیستم روبوداک و اجزای آن
۱۱۱	۵-۱. معماری سیستم روبوداک
۱۱۳	۵-۱-۱. سناریوهای تعاملی روبوداک و انسان
۱۱۸	۵-۱-۲. نمودار بلوکی ارتباطات فیزیکی اجزای سیستم روبوداک
۱۱۸	۵-۲. زیرسیستم بینایی
۱۱۹	۵-۲-۱. تشخیص چهره
۱۲۲	۵-۲-۲. بازشناسی چهره
۱۲۵	۵-۲-۳. تشخیص حرکت
۱۲۷	۵-۳. زیر سیستم پردازش گفتار
۱۲۸	۵-۳-۱. پردازش گفتار فارسی
۱۲۸	۵-۳-۲. پردازش گفتار انگلیسی
۱۲۸	۵-۴. زیرسیستم تبدیل متن به گفتار
۱۲۹	۵-۵. زیرسیستم جستجو و پردازش متن
۱۲۹	۵-۵-۱. جستجو و بازیابی اطلاعات در گنج
۱۲۹	۵-۵-۲. پیش پردازش عبارت پرس و جو
۱۳۲	۵-۵-۳. جستجوی عبارت
۱۳۲	۵-۵-۴. امتیازدهی
۱۳۴	۵-۶. تشریح کدهای برنامه اصلی سیستم روبوداک

۱۳۵	۵-۶-۱. کنترلر اصلی
۱۳۸	۵-۶-۲. برنامه زیرسیستم بینایی
۱۳۹	۵-۶-۳. برنامه کنترل پروژکتور
۱۴۰	۵-۶-۴. برنامه اتصال به زیرسیستم تبدیل متن به گفتار
۱۴۱	۵-۶-۵. برنامه اتصال به زیرسیستم تشخیص گفتار
۱۴۳	۵-۶-۶. برنامه اتصال به سیستم جستجو و بازیابی اطلاعات
۱۴۵	۵-۶-۷. برنامه سیستم جستجو و بازیابی اطلاعات
۱۴۷	۶. ارزیابی عملکرد سیستم روبوداک و اجزای آن
۱۴۷	۶-۱-۱. ارزیابی زیرسیستم بینایی
۱۴۸	۶-۱-۱. داده‌های آموزش و تست
۱۵۰	۶-۱-۲. ارزیابی الگوریتمهای تشخیص صورت
۱۵۳	۶-۱-۳. ارزیابی الگوریتمهای بازشناسی چهره
۱۵۸	۶-۲. ارزیابی زیرسیستم تشخیص گفتار
۱۶۱	۶-۳. ارزیابی زیرسیستم تبدیل متن به گفتار
۱۶۴	۶-۴. ارزیابی سیستم بازیابی اطلاعات
۱۶۴	۶-۴-۱. آماده‌سازی داده‌ها
۱۷۰	۶-۴-۲. نتایج زیرسیستم بازیابی اطلاعات
۱۷۴	۶-۴-۳. دسته‌بندی و نمایش مصور نتایج جستجو
۱۸۰	۷. توسعه قابلیت‌های روبوداک در آینده
۱۸۰	۷-۱. معرفی پروژه‌های پیشنهادی برای توسعه روبوداک
۱۸۲	۷-۲. شرح پروژه‌ها
۱۹۰	۷-۳. اولویت‌بندی پروژه‌های پیشنهادی برای توسعه خدمات روبوداک

۸. جمع‌بندی و نتیجه‌گیری	۱۹۶
۹. منابع	۱۹۸
پیوست ۱: جداول امتیازدهی خدمات روبوداک	۲۰۳
پیوست ۲: کدهای برنامه	۲۰۶

فهرست شکل‌ها

شکل ۱-۲- خدمات روبوداک براساس گام‌های اصلی پژوهش	۲۴
شکل ۲-۲- ورودی، پردازش و خروجی‌های روبوداک براساس خدمات مورد نیاز در مرحله آغازین توسعه	۲۹
شکل ۱-۳- نمای ربات NAO	۳۲
شکل ۲-۳- نمای ربات DARwIn-OP	۳۵
شکل ۳-۳- نمای ربات Pepper	۳۸
شکل ۴-۳- نمای ربات Romeo	۴۰
شکل ۵-۳- نمای ربات Hubo2	۴۲
شکل ۶-۳- نمای ربات‌های سری HOVIS	۴۳
شکل ۷-۳- نمای ربات HOAP-1	۴۵
شکل ۸-۳- نمای ربات HOAP-3	۴۶
شکل ۹-۳- نمای ربات Maggie	۴۷
شکل ۱۰-۳- نمای ربات Sacarino	۵۰
شکل ۱۱-۳- نمای ربات Asimo	۵۱
شکل ۱۲-۳- نمای سنسورها و زاویه دید Asimo	۵۲
شکل ۱۳-۳- نمای ربات Rabo	۵۳
شکل ۱-۴- شمای ربات Nao به عنوان پلتفرم روبوداک	۶۲
شکل ۲-۴- ابعاد ربات از نماهای مختلف	۶۳
شکل ۳-۴- میزان چرخش سر	۶۴
شکل ۴-۴- محدوده‌ی درجه آزادی بازو	۶۴
شکل ۵-۴- میزان چرخش لگن و ران پا	۶۵
شکل ۶-۴- محدوده‌ی درجه آزادی پا	۶۵
شکل ۷-۴- قسمت‌های موتوردار ربات	۶۶
شکل ۸-۴- محل قرارگیری سنسورها روی ربات	۶۷
شکل ۹-۴- محل قرارگیری سنسور دکمه‌ی فشاری	۶۸

- شکل ۴-۱۰- محل قرارگیری سنسور بامپر..... ۶۸
- شکل ۴-۱۱- محل قرارگیری بلندگوها..... ۶۸
- شکل ۴-۱۲- محل قرارگیری میکروفن‌ها..... ۶۹
- شکل ۴-۱۳- محل قرارگیری سنسور مادون قرمز در سر به عنوان چشم..... ۷۰
- شکل ۴-۱۴- محل قرارگیری سنسور وزن و نیرو..... ۷۰
- شکل ۴-۱۵- محل قرارگیری ژيروسکوپ و شتاب‌سنج..... ۷۱
- شکل ۴-۱۶- داده‌های فیلتر کالمن..... ۷۱
- شکل ۴-۱۷- محل قرارگیری سنسورهای سونار..... ۷۲
- شکل ۴-۱۸- محدوده پوششی دوربین‌ها..... ۷۲
- شکل ۴-۱۹- دوربین لایو کم..... ۷۵
- شکل ۴-۲۰- جایگاه دوربینها در هنگام آزمایش کیفیت پردازشی دو دوربین..... ۷۶
- شکل ۴-۲۱- هیستوگرام تصاویر گرفته شده از دو دوربین برای بررسی کنتراست..... ۷۷
- شکل ۴-۲۲- تصاویر گرفته شده توسط دو دوربین..... ۷۸
- شکل ۴-۲۳- روی صفحه‌ی بُرد رسپری..... ۸۴
- شکل ۴-۲۴- زیر صفحه‌ی بُرد رسپری..... ۸۴
- شکل ۴-۲۵- اجزای کلی برد..... ۸۵
- شکل ۴-۲۶- محل پردازنده‌ی آرم روی برد..... ۸۵
- شکل ۴-۲۷- محل تغذیه‌ی رسپری..... ۸۶
- شکل ۴-۲۸- محل قرارگیری HDMI..... ۸۶
- شکل ۴-۲۹- محل قرارگیری USB..... ۸۷
- شکل ۴-۳۰- محل قرارگیری پایه‌های GPIO..... ۸۸
- شکل ۴-۳۱- نمای کلی ویدئو پروژکتور..... ۸۹
- شکل ۴-۳۲- پروژکتور پس از جداسازی نگهدارنده موبایل..... ۹۰
- شکل ۴-۳۳- پایه‌های GPIO در رسپری و نقشه آن..... ۹۱
- شکل ۴-۳۴- مدار کنترلی استفاده شده برای کنترل ویدئو پروژکتور..... ۹۲
- شکل ۴-۳۵- شکل موج پالسی که با مدار قبل به ویدئو پروژکتور اعمال می‌شود..... ۹۳

- شکل ۴-۳۶- طراحی بخش اول محفظه کلاه-مانند برای جاسازی دوربین و ویدئو پروژکتور ۹۴
- شکل ۴-۳۷- نمایی از روبوداک به همراه دوربین و ویدئو پروژکتور جاسازی شده ۹۵
- شکل ۴-۳۸- محل جاسازی بُرد رسیپی، روی بدن ربات ۹۶
- شکل ۴-۳۹- بارگذاری فایل‌های proto حاوی کدهای ربات در ویباتر ۹۹
- شکل ۴-۴۰- نمایی از ربات Nao همراه اتصال دو شیء خارجی روی سر و کمر آن ۱۰۳
- شکل ۴-۴۱- نمودار شتاب‌سنج در حالت تعادل و عدم تعادل هنگام نشستن ۱۰۶
- شکل ۴-۴۲- نمودار شتاب‌سنج در حالت تعادل و عدم تعادل هنگام راه رفتن ۱۰۷
- شکل ۴-۴۳- نمودار ژيروسکوپ در حالت تعادل و عدم تعادل هنگام نشستن ۱۰۸
- شکل ۴-۴۴- نمودار ژيروسکوپ در حالت تعادل و عدم تعادل هنگام راه رفتن ۱۰۹
- شکل ۵-۱- نمودار اجزای سیستم ۱۱۲
- شکل ۵-۲- تعامل شناخت اولیه، «Greeting» ۱۱۶
- شکل ۵-۳- تعامل جستجو، «Search» ۱۱۷
- شکل ۵-۴- نمودار بلوکی سیستم روبوداک ۱۱۸
- شکل ۵-۵- نمونه‌هایی از پنجره‌های جستجو که برای استخراج ویژگی‌های Haar مورد استفاده قرار می‌گیرند ۱۲۰
- شکل ۵-۶- نمونه‌ای از نحوه انتقال پنجره‌های جستجو در تمام تصویر ۱۲۰
- شکل ۵-۷- نمونه‌ای از محاسبه ویژگی LBP برای یک همسایگی 3×3 ۱۲۱
- شکل ۵-۸- نمونه قطعه‌بندی تصویر و اعمال الگوریتم استخراج ویژگی LBP بر روی تصویر ۱۲۲
- شکل ۵-۹- لوگوی رنگی روبوداک برای تشخیص حرکت ۱۲۶
- شکل ۵-۱۰- مراحل اصلی جستجو و بازیابی اطلاعات از گنج ۱۲۹
- شکل ۶-۱- نمونه‌ای از تصاویر پایگاه داده AT&T Facedatabase ۱۴۸
- شکل ۶-۲- نمونه‌ای از تصاویر پایگاه داده Extended Yale Facedatabase B با تابش‌های نوری متفاوت ۱۴۹
- شکل ۶-۳- نمونه‌ای از تصاویر پایگاه داده Extended Yale Facedatabase B با جهت‌های مختلف ۱۴۹
- شکل ۶-۴- نمونه‌هایی از تصاویر استفاده شده در این پژوهش ۱۵۰
- شکل ۶-۵- نمونه‌هایی از تصاویر موقعیت ۱ ۱۵۰

- شکل ۶-۶- نمونه‌هایی از تصاویر موقعیت ۲ ۱۵۱
- شکل ۶-۷- نمونه‌هایی از تصاویر موقعیت ۳ ۱۵۱
- شکل ۶-۸- نمونه‌ای از تصاویر پیش‌پردازش شده برای آموزش و تست ۱۵۳
- شکل ۶-۹- نمایش نموداری نتایج بازشناسی چهره براساس الگوریتم Fisherfaces ، با استفاده از الگوریتم تشخیص صورت LBP، روی تصاویر موقعیت ۱ برای حدآستانه ۰/۶ تا ۰/۹ ۱۵۴
- شکل ۶-۱۰- نمایش نموداری نتایج بازشناسی چهره بر روی تصاویر موقعیت ۱ و ۲ با الگوریتم بازشناسی چهره Eigenface با حدآستانه ۰/۵ ۱۵۶
- شکل ۶-۱۱- نمونه‌ای از حذف نویز موجود در تصویر و عدم تاثیر اجسام ثابت هم‌رنگ لوگو در الگوریتم تشخیص حرکت ۱۵۹
- شکل ۶-۱۲- توزیع موضوعات سطح دوم در موضوعات سطح اول ۱۶۷
- شکل ۶-۱۳- نمودار گنج در یک نگاه ۱۷۶
- شکل ۶-۱۴- نمایش صفحه رابط کاربری اصلی ۱۷۷
- شکل ۶-۱۵- نمایش نتایج جستجو همراه با نمودار موضوعی هر رکورد ۱۷۸
- شکل ۶-۱۶- نمایش نتایج جستجو در یک نگاه ۱۷۸

فهرست جداول

جدول ۱-۳-اولویت بندی خدمات روبوداک.....	۵۸
جدول ۱-۴- مشخصات باتری ربات.....	۶۳
جدول ۲-۴- ابعاد ربات.....	۶۳
جدول ۳-۴- میزان چرخش سر.....	۶۴
جدول ۴-۴- بازه ی حرکتی بازو.....	۶۴
جدول ۵-۴- بازه ی حرکتی لگن و ران پا.....	۶۵
جدول ۶-۴- بازه ی حرکتی پا.....	۶۵
جدول ۷-۴- مشخصات موتور.....	۶۶
جدول ۸-۴- نوع موتور به کاررفته در سر.....	۶۶
جدول ۹-۴- نوع موتور به کاررفته در بازو.....	۶۶
جدول ۱۰-۴- نوع موتور به کاررفته در پای ربات.....	۶۷
جدول ۱۱-۴- نام سنسورها و محل آن ها.....	۶۷
جدول ۱۲-۴- مختصات بلندگوها در ربات.....	۶۸
جدول ۱۳-۴- مختصات میکروفن ها در ربات.....	۶۹
جدول ۱۴-۴- مشخصات دوربین سر.....	۷۲
جدول ۱۵-۴- مقایسه مشخصات اصلی دوربین ربات و لایو کم.....	۷۵
جدول ۱۶-۴- نتیجه مطالعات درباره نوع میکروفون.....	۸۱
جدول ۱۷-۴- مشخصات میکروفون لاجیتک اچ ۸۰۰.....	۸۱
جدول ۱۸-۴- مشخصات پردازش لپ تاپ.....	۸۲
جدول ۱۹-۴- مشخصات برد رسپری.....	۸۳
جدول ۲۰-۴- مشخصات کلی ویدئو پروژکتور.....	۹۰
جدول ۲۱-۴- ابعاد و وزن ویدئو پروژکتور.....	۹۰
جدول ۲۲-۴- آستانه تحمل وزن ربات برای اضافه نمودن یک شیء، برای حالات فیزیکی مخلف	
(وزن به کیلو گرم).....	۱۰۵

جدول ۴-۲۳- آستانه تحمل وزن ربات هنگام اضافه نمودن دو شیء به صورت همزمان، برای حالات فیزیکی مختلف (وزن به کیلوگرم).....	۱۰۵
جدول ۵-۱- نمونه‌ای از کلمات مفرد و جمع مکسر آنها	۱۳۱
جدول ۶-۱- نتایج تشخیص صورت و تشخیص چشم با تصاویر گرفته شده در موقعیت ۱	۱۵۲
جدول ۶-۲- نتایج تشخیص صورت و تشخیص چشم با تصاویر گرفته شده در موقعیت ۲	۱۵۲
جدول ۶-۳- نتایج تشخیص صورت و تشخیص چشم با تصاویر گرفته شده در موقعیت ۳	۱۵۲
جدول ۶-۴- نتایج بازشناسی چهره براساس الگوریتم Fisherfaces، با استفاده از الگوریتم تشخیص صورت LBP، روی تصاویر موقعیت ۱ برای حد آستانه‌های مختلف.....	۱۵۴
جدول ۶-۵- نتایج بازشناسی چهره براساس الگوریتم Fisherfaces، با استفاده از الگوریتم تشخیص صورت LBP، روی تصاویر موقعیت ۲ برای حد آستانه‌های مختلف.....	۱۵۴
جدول ۶-۶- نتایج بازشناسی چهره بر روی تصاویر موقعیت ۱ و موقعیت ۲ با الگوریتم Eigenface و حد آستانه ۰/۵	۱۵۵
جدول ۶-۷- نتایج ارزیابی زیرسیستم تشخیص گفتار روبوداک	۱۶۰
جدول ۶-۸- نتایج عملکرد زیر سیستم تبدیل متن به گفتار در تلفظ واژگان	۱۶۳
جدول ۶-۹- دسته‌بندی موضوعی، سطح اول و سطح دوم.....	۱۶۵
جدول ۶-۱۰- نتایج حاصل از جستجوی عبارت «تغییر حسابر س و دوره تصدی حسابر سی».....	۱۷۱
جدول ۶-۱۱- نتایج حاصل از جستجوی «کاهش نشانه‌های اختلال اضطراب اجتماعی»	۱۷۲
جدول ۶-۱۲- نتایج حاصل از جستجوی «حل عددی معادلات انتگرال».....	۱۷۳
جدول ۷-۱- پروژه‌های توسعه خدمات روبوداک.....	۱۸۱
جدول ۷-۲- پروژه‌های توسعه قابلیت‌های تعاملی، شناختی و حرکتی، و سایر کاربردها	۱۸۱
جدول ۷-۳- امتیاز پروژه‌های توسعه خدمات براساس معیارهای مختلف.....	۱۹۲
جدول ۷-۴- پروژه‌های توسعه قابلیت‌های تعاملی، شناختی و حرکتی، و سایر کاربردها براساس معیارهای مختلف	۱۹۳
جدول ۷-۵- اولویت اجرای پروژه‌ها.....	۱۹۴

فصل اول:

کلیات پژوهش

۱. کلیات پژوهش

پژوهش، مانند بسیاری از فعالیت‌های انسانی مدرن، در حال تبدیل شدن به یک تلاش پیچیده و چالش برانگیز است. دنبال کردن تعداد زیاد مقالات علمی و یافته‌های پژوهشی، سازمان‌دهی داده‌های علمی به صورت کارآمد برای دسترسی و استفاده آسان، ردیابی تازه‌های پژوهشی، ایده‌پردازی برای ارائه نوآوری در پژوهش، تحلیل حجم وسیع داده‌های پژوهشی، و مستندسازی مجموعه داده‌های پژوهشی در قالب یک پایگاه داده شخصی، از جمله چالش‌هایی هستند که عموماً یک پژوهشگر حین به سرانجام رساندن یک پژوهش با آن مواجه است. پژوهشگر معمولاً در بخشی از فرایند پژوهش، با مدیریت داده‌های علمی و سازماندهی آنها سروکار دارد. مدیریت داده‌های علمی اساساً به چندین مرحله تقسیم می‌شود: مدیریت جریان کاری، مدیریت فراداده، یکپارچه سازی داده‌ها، بایگانی داده‌ها و پردازش داده‌ها. به طور خاص برای انجام هریک از این مراحل ابزارهای متنوعی وجود دارد، لیکن یادگیری این روش‌ها و یکپارچه‌سازی آنها نیازمند تلاش زیادی است که حتی ممکن است استفاده یکپارچه آنها با هم سازگار نباشد.

یک دستیار ماشینی هوشمند می‌تواند به یک پژوهشگر در برطرف نمودن چالش‌های فوق یاری برساند. دستیار هوشمندی که بتواند ابزارهای موجودی که در فرایندها و مراحل یک پژوهش استفاده می‌شود، را به صورت یکپارچه براساس نیاز پژوهشگر و حوزه تخصصی وی ارائه دهد. چنین دستیار لازمی است الگوی مناسبی از رفتار پژوهشی پژوهشگر، علاقه‌مندی‌ها و حوزه‌های تخصصی وی داشته باشد. بنابراین در طول زمان با تعامل دستیار شخصی با پژوهشگر این الگو تقویت می‌شود. تا در نهایت برای هر پژوهشگر، خدمات سفارشی و مختص وی ارائه گردد.

دستیاران شخصی هوشمند، می‌توانند در قالب ربات‌های تعاملی و اجتماعی نمود پیدا کنند، ربات‌های فیزیکی که حتی از تعداد محدودی درجه آزادی حرکتی برخوردار هستند. ربات‌هایی که می‌توانند با فرد، تعاملی سازنده داشته باشند، و به صورت اجتماعی با محیط خود ارتباط داشته باشند، به عنوان ربات‌های اجتماعی خوانده می‌شوند. این ربات‌ها در بسیاری از موارد به عنوان دستیاران شخصی،

می‌توانند افراد را در انجام فعالیت‌های خود توانمندتر سازند. نمونه بارز بکارگیری این ربات‌ها در کمک به سالمندان و افرادی که از توانایی‌های جسمی رنج می‌برند. در این طرح پژوهشی، ربات اجتماعی پژوهشیار با عنوان «روبوداک» معرفی می‌شود که می‌تواند در آینده پاسخگوی نیازهای تخصصی یک پژوهشگر باشد. روبوداک، یک ربات اجتماعی است که می‌تواند پژوهشگر را در برطرف نموده چالش‌های پژوهشی حین فرایند پژوهش یاری رساند. در برخی از موارد یک پژوهشگر لازم است مجموعه‌ای از آزمایش‌های فیزیکی را انجام دهد که امکان حضور وی در فضای آزمایشگاهی میسر نباشد، در اینگونه موارد دستیار هوشمند وی، روبوداک، می‌تواند این آزمایش‌ها را کنترل کند و داده‌های مورد نیاز را جمع‌آوری کند و برای پژوهشگر ارسال نماید. انجام اعمال و آزمایش از راه دور و کنترل فضای آزمایشگاهی با عنوان دورآزمایش^۲ شناخته شده که در حال حاضر نیز کاربردهای متنوعی در علوم پزشکی دارد. روبوداک به پژوهشگر کمک می‌کند تا به کتابخانه‌ها و تجهیزاتی که به راحتی و به صورت فیزیکی در دسترس نیستند، دسترسی داشته باشد. به علاوه باید توانایی جستجوی علمی، یادداشت برداری، یادآوری وقایع علمی، ضبط صدا، و سازمان‌دهی منابع و مستندات را نیز داشته باشد. طرح حاضر، مرحله مقدماتی از توسعه ربات اجتماعی پژوهشیار است که طی آن با استفاده از یکی از پلتفرم‌های رباتیک موجود، چند قابلیت مورد انتظار روبوداک توسعه می‌یابد. از ویژگی‌های بارز این طرح ماهیت میان‌رشته‌ای آن است که حوزه‌های تحقیقاتی نظیر یادگیری ماشین، زبان‌شناسی رایانشی، پردازش متن و گفتار، پردازش و تحلیل اطلاعات و تعامل انسان و ماشین را درگیر می‌نماید.

۱-۱. ربات فیزیکی؛ ضرورت یا امکان؟

یکی از مسائلی که امروزه در حوزه رباتیک و سیستم‌های هوشمند، پژوهشگران و متخصصان امر را درگیر خود کرده، تصمیم‌گیری درباب نمود فیزیکی یا مجازی ربات‌ها یا عامل^۳‌های هوشمند است. ساخت ربات‌های انسان‌نما که دارای بدن فیزیکی و مهارت‌های اجتماعی باشند مستلزم صرف هزینه‌های هنگفتی است و این امر موجب می‌شود بسیاری از پژوهشگران اینگونه ربات‌ها را در قالب یک عامل مجازی هوشمند پیاده‌سازی کنند که فاقد جسم فیزیکی هستند. از سوی دیگر، از آنجا که

^۲ Tele-LAB

^۳ agent

حجم سنگینی از محاسبات و پردازش‌های ورودی و خروجی ربات‌ها توسط رایانه انجام می‌شود، از این‌رو در ربات‌هایی که وظایف پردازشی آنها بر نقش‌های حرکتی‌شان می‌چربد، عملاً احساس نیازی به فیزیک ربات دیده نمی‌شود. برای مثال، روبوداک یک ربات اجتماعی پژوهشیار است که قرار است به‌عنوان دستیار پژوهشی به محققان کمک کند. مهم‌ترین خدمات این روبات به پژوهشگران به‌طور کلی شامل جمع‌آوری اطلاعات، سازماندهی و تحلیل اطلاعات و تولید گزارش است. از آنجا که جمع‌آوری اطلاعات عمدتاً از طریق ورودی‌های گفتاری و متنی انجام می‌گیرد و مراحل سازماندهی و تحلیل اطلاعات و نیز ارائه‌ی گزارش با استفاده از سامانه‌های ایرانداک و الگوریتم‌های رایانه‌ای انجام می‌گیرند، لذا ممکن است این پرسش به‌وجود آید که کاربرد فیزیک ربات در این میان کجاست؟ آیا نمی‌توان همه این خدمات را با استفاده از یک عامل مجازی رایانه‌ای ارائه کرد؟ صرف‌نظر از اینکه یکی از اهداف پروژه روبوداک استفاده از یک ربات فیزیکی برای انجام آزمایش‌های راه‌دور در آزمایشگاه‌های دورآزماس، تحقیقات اخیر در حوزه‌های علوم شناختی و ربات‌های اجتماعی نشان می‌دهند که جسمیت‌یافتگی^۴ ربات (بخصوص در ربات‌های اجتماعی) نه یک امر تشریفاتی و لوکس بلکه ضرورتی با کارکردهای مهم ارتباطی و شناختی است.

۱-۱-۱. آزمایشگاه‌های دورآزما

اگرچه فاز اول پروژه روبوداک به طراحی مدل مفهومی خدمات پژوهشی و پیاده‌سازی بخشی از آنها اختصاص دارد، با این حال انجام آزمایش‌های راه‌دور به کمک روبوداک در چشم‌انداز این پروژه محوریت ویژه‌ای دارد. در این چشم‌انداز، روبوداک رباتی با قابلیت‌های حرکتی در نظر گرفته شده که در یک آزمایشگاه مستقر است و پژوهشگران به‌صورت راه‌دور به سامانه‌ی ربات متصل می‌شوند و دستور انجام برخی آزمایش‌ها را می‌دهند. در سناریوهای مشابه، پیش‌بینی شده که روبوداک در مکان‌هایی نظیر کتابخانه‌های مرجع مستقر شود و امکان دسترسی پژوهشگران به منابعی که غیرقابل اشتراک‌گذاری برای استفاده عموم هستند را فراهم آورد. بنابراین داشتن جسم فیزیکی با قابلیت‌های حرکتی برای روبوداک امری ضروری است.

۱-۱-۲. جسم اساسی هوشمندی و تعامل موثر

علوم شناختی به دو نسل تقسیم می‌شود. نسل اول شناختی الهام‌گرفته از ایده‌ای موسوم به ماشین تورینگ است که در آن هوشمندی هر سیستمی در گرو داشتن تعداد محدودی نماد انتزاعی و

^۴ embodiment

الگوریتم‌هایی برای ترکیب این نمادهاست. طبق این دیدگاه، کار با نمادها شرط لازم و کافی برای هوشمندی محسوب می‌شود (Newell & Simon, 1976). این ایده شالوده و منطقی رایانه‌های دیجیتال کنونی را تشکیل می‌دهد. در نسل دوم علوم شناختی که از اواخر دهه‌ی هفتاد میلادی به‌عنوان پارادایم جایگزین مطرح شد، محیط و جسم نیز جزو لاینفک کنشگری هوشمند محسوب می‌شوند. طبق این دیدگاه، هوشمندی صرفاً قدرت پردازشی مغز انسان (یا تراشه‌های سیلیکونی رایانه‌ها) در پردازش نمادهای انتزاعی نیست، بلکه با واسطه‌گری و محوریت جسم و در تعامل با محیط پیرامون پدیدار می‌شود (Pfeifer & Bongard, 2006). این نگاه متفاوت به پدیده‌ی هوشمندی و پردازش اطلاعات، حوزه‌هایی چون علوم اعصاب، هوش مصنوعی و زبان‌شناسی را متحول کرد. برای نمونه در حوزه هوش مصنوعی و رباتیک، تحقیقات عمدتاً به سمت وسویی رفته‌اند که در آنها اثر جسم و مشخصه‌های فیزیکی بر پردازش و کنش هوشمند ربات‌ها مطالعه می‌شود. مهم‌ترین فرضیه‌ای که در چارچوب این دیدگاه مطرح می‌شود این است: انسان‌ها دارای نوع بخصوصی رفتار و تفکر هوشمند هستند چون اساساً ویژگی‌های جسمانی بخصوصی دارند. بنابراین، اگر بخواهیم رفتار هوشمند انسانی را در ماشین‌ها و ربات‌ها شبیه‌سازی کنیم، باید آنها را صاحب بدن و جسم مشابهی سازیم.

روبوداک یک ربات اجتماعی است که قرار است به‌عنوان دستیار پژوهش با پژوهشگران تعامل کند. اجتماعی بودن این ربات بدان معناست که قرار است همانند یک عامل انسانی هوشمند با کاربر تعامل کند و به نیازها، خواسته‌ها، احساسات و کنش‌های کاربر واکنش نشان دهد. به‌طور متقابل، کاربرانی که با روبوداک تعامل می‌کنند نیز باید بتوانند به اطلاعات ارائه‌شده توسط ربات اعتماد کنند و ارتباط تعاملی آنها با ربات تجربه‌ای لذت‌بخش و طبیعی برایشان باشد. جسمانی بودن این ربات مشخصه لاینفک اجتماعی بودن آن است و زمینه‌ی لازم برای تعامل طبیعی و موثر با کاربران را فراهم می‌سازد.

برای نمونه، یک ربات اجتماعی پژوهشیار نظیر روبوداک باید بتواند بدون اینکه کاربر مجبور باشد هر بار برای دریافت خدمات پژوهشیاری، خود را به ربات بشناساند و تنظیمات خدمات سفارشی‌شده خود را بر روی سیستم فراخوانی کند، کاربران مختلف را شناسایی کرده و هوشمندی حافظه‌مدار از خود نشان دهد. این شناسایی و تمیز هوشمند کاربران از یکدیگر نیازمند جسم فیزیکی و قابلیت‌های حرکتی است که ربات را قادر می‌سازد محیط پیرامون خود را همانند یک انسان بکاود، کاربران را از یکدیگر متمایز کند و به تعامل اجتماعی با آنها پردازد.

از سوی دیگر، تحقیقات اخیر نشان می‌دهد که تعامل کاربران انسانی با ربات‌های فیزیکی (در مقایسه با ربات‌ها یا عامل‌های مجازی) تجربه‌ی ناب‌تری از تعامل، لذت‌بخشی، حس اعتماد و پذیرش را در کاربران برمی‌انگیزد (Pfeifer & Bongard, 2006; Wainer, Feil-Seifer, Shell, & Mataric, 2002).

برای نمونه، پژوهش‌های اخیر در حوزه‌ی تعامل انسان و ربات نشان می‌دهد که افراد تعامل خود با ربات‌های فیزیکی را مثبت‌تر و طبیعی‌تر از ربات‌های مجازی ارزیابی می‌کنند و اینکه ربات‌های فیزیکی بیشتر از ربات‌های مجازی قادرند حس اعتماد کاربر را جلب کنند (Bainbridge, Hart, Kim, & Scassellati, 2011). نکته دیگری که در خدمات ارائه‌شده توسط روبوداک محوریت دارد و داشتن فیزیک انسان‌نما برای ربات را ضروری می‌سازد، ارائه‌ی خدمات آموزشی درباب شیوه پژوهش به کاربران است. پژوهش‌های تجربی نشان می‌دهند که ربات‌های انسان‌نمای دارای جسم فیزیکی بیش از ربات‌های مجازی جدی گرفته می‌شوند و اطلاعات‌شان آموزنده و معتبر تلقی می‌شود (Kidd & Breazeal, 2004).

۱-۲. پیشینه کاربرد ربات‌های اجتماعی

ربات‌های اجتماعی اخیراً در حوزه‌های مختلفی مورد استفاده قرار گرفته‌اند. کاربرد ربات‌های اجتماعی را می‌توان در سه دسته تقسیم‌بندی کرد:

- کاربردهای تخصصی که در آن‌ها ربات مجموعه خاصی از وظایف را با کمترین تعامل با کاربر انجام می‌دهد.
- وظایف عمومی که در آن‌ها تعداد زیادی کاربر با ربات در تعامل هستند و ربات دارای یک مدل تعاملی کلی و واحد برای ارتباط کوتاه مدت با کاربران بدون نیاز به ایجاد سازگاری با نیازهای تخصصی هر کاربر است.
- کاربردهای شخصی که در آن‌ها ربات برای تعاملات طولانی مدت^۵ با کاربران خاص طراحی شده است و قابلیت‌های پیچیده و سازگاری با نیازهای فرد را دارد.

در این طرح، هدف کاربردهای شخصی با تعامل طولانی مدت است. ادبیات ربات‌های اجتماعی با کاربرد تعاملات طولانی مدت اخیراً در (Leite, Martinho, & Paiva, 2010) مورد مطالعه قرار

⁵ long-term interaction

گرفته است که در آن ویژگی‌های اصلی که ربات‌های اجتماعی باید به منظور تعامل طولانی مدت با کاربر داشته باشند، معرفی شده است. براساس این مطالعه، حوزه‌های کاربردی اصلی که در آنها تعامل طولانی ربات با انسان در نظر گرفته شده، عبارتند از سلامت و درمان، آموزش، محیط‌های کاری و فضاهای عمومی و خانه. یکی از رایج‌ترین حوزه‌های کاربردی ربات‌های اجتماعی، کمک کردن به افراد ناتوان است. ربات‌های اجتماعی برای کمک به افراد ناتوان برای همراهی، خدمات خانگی و سلامت طراحی شده‌اند (Yumakulov, Yergens, & Wolbring, 2012). به خصوص، برخی از تحقیقات اخیر در کاربرد ربات‌های اجتماعی در سلامت به درمان کودکان اوتیستی^۶ اختصاص یافته است (Aleami, Meghdari, Basiri, & Taheri, 2015). ربات‌های استفاده شده در محیط‌های خانگی معمولاً برای سرگرمی، وظایف روزمره و کمک به افراد مسن و معلول مورد استفاده قرار می‌گیرند (Yan, Jr, & Poo, 2014). به عنوان نمونه، پارو^۷ یک ربات اجتماعی تعاملی است که برای کمک به افراد مسن طراحی شده است (Kidd, Taggart, & Turkle, 2006). استفاده از ربات دستیار برای کنترل از راه دور در منازل برای برطرف کردن نیازهای اولیه آن‌ها مانند مسیریابی بین موانع توسط میچاود (Michaud, 2007) مورد بررسی قرار گرفته است. برایان^۸ یک ربات انسان‌نماست که از کمر به بالا عملکردی شبیه انسان دارد و برای تحقیقات در زمینه عکس العمل سیستم رباتیک به حالات انسانی مورد استفاده قرار گرفته است (Nejat & Ficocelli, 2008). روبوی^۹ (Mitsunaga, 2002) یک ربات انسان‌نماست که برای تعامل با مردم در زندگی روزمره طراحی شده است. در حوزه‌های تجاری نیز، از ربات اجتماعی برای تشخیص مشتریان بالقوه از بقیه مردم در یک فروشگاه استفاده شده است (Kanda, 2008). در حوزه استفاده از ربات‌های اجتماعی در محیط‌های اداری، یک ربات راهنمای اداره برای کمک به بازدید کنندگان در محیط اداری طراحی شده است (Pacchierotti, Christensen, & Jensfelt, 2006). مگی^{۱۰} (Gonzalez-Pacheco, 2011) توسط آزمایشگاه رباتیک دانشگاه کارلوس مادرید^{۱۱} توسعه داده شد که به خصوص برای پلتفرم بازی سفارشی سازی شده است (Ailamaki, Kantere, & Dash, 2010). ربات‌های سری هویس^{۱۲} و ربات

⁶ Autistic

⁷ Paro

⁸ Brian

⁹ Robovie

¹⁰ Maggie

¹¹ University Carlos III of Madrid

¹² HOVIS

ساکارینو^{۱۳} عمدتاً برای خدمات خانگی مورد استفاده قرار می‌گیرند (Feil-Seifer & Mataric, 2002).

پروژه‌ها و مطالعات متعددی برای توسعه پلتفرم برای ربات‌های اجتماعی تا کنون انجام شده‌اند که ربات‌های انسان‌نما مانند نائو^{۱۴}، پپر^{۱۵} و رومئو^{۱۶}، که توسط شرکت سافت بانک^{۱۷} / الدباران^{۱۸} ارائه شده‌اند، از این دسته هستند.

با توجه به مطالعات صورت گرفته، استفاده از ربات‌های اجتماعی در زمینه سلامت برای بهبود زندگی افراد مسن، افراد با اختلالات ذهنی و معلولین متمرکز شده و تا کنون پژوهشی در زمینه استفاده از ربات‌های اجتماعی به عنوان دستیار پژوهشگر ارائه نشده است.

۳-۱. معرفی روبوداک

روبوداک یک ربات اجتماعی پژوهشیار است که در فرایند پژوهش به پژوهشگر یاری می‌رساند. بر همین اساس خدماتی که می‌تواند این ربات اجتماعی به پژوهشگر ارائه دهد در قالب گام‌های اصلی پژوهش قابل تعریف است. روبوداک در حقیقت نقش یک منشی ماشینی را برای یک پژوهشگر جهت انجام امور پژوهشی ایفا می‌کند که به خصوص در امور زیر به پژوهشگر کمک می‌کند:

- جستجو و بازیابی اطلاعات و داده‌های نیمه‌ساخت یافته و غیرساخت یافته مرتبط با یک حوزه پژوهشی از منابع اطلاعاتی مختلف با پلت فرم‌های متنوع
- ذخیره‌سازی، نمایه‌سازی، برچسب‌زنی خودکار اطلاعات و داده‌های نیمه‌ساخت یافته و غیرساخت یافته (مشمول بر اطلاعات چندرسانه‌ای) از طریق دستورات گفتاری و نوشتاری مطابق با درخواست‌ها و نیازهای پژوهشگر
- قابلیت جستجو در پایگاه‌های اطلاعات علمی برای بازیابی اطلاعات مورد نیاز کاربر از طریق دستورات گفتاری و نوشتاری
- ارائه اطلاعات بازیابی شده به صورت سازماندهی شده به پژوهشگر

¹³ Sacarino

¹⁴ Nao

¹⁵ Pepper

¹⁶ Romeo

¹⁷ Softbank

¹⁸ Aldebaran

- بازنمایی و اطلاع‌رسانی در مورد آخرین پیشرفت‌ها و دستاوردهای پژوهشی تعامل با سایر ربات‌ها یا عامل‌ها برای دریافت نظرات و دسترسی به روزآمدترین پژوهش‌ها
 - یاری در انجام آزمایش‌های پژوهشی (به صورت فیزیکی) از راه دور (با داشتن قابلیت‌های دورآزمایش^{۱۹}) و همچنین امکان راه رفتن و انجام حرکات فیزیکی مانند جابجا نمودن اشیاء کوچک نظیر مستندات پژوهشی و مدارک علمی
 - قابلیت اسکن کردن و تصویربرداری از متون و بکارگیر قابلیت‌های نویسه‌خوان نوری^{۲۰} برای تبدیل مستندات تصویری به مستندات دیجیتالی و خواندن از روی متون فارسی و لاتین
 - یادگیری منش و سبک پژوهشی یک پژوهشگر و برقراری تعامل با وی مطابق با آن
- هر یک از قابلیت‌ها فوق می‌تواند در قالب خدمات روبوداک در گام‌های اصلی پژوهش بازتعریف شود. گام‌های اصلی پژوهش که ربات می‌تواند در انجام آن به پژوهشگر کمک کند، عبارتند از:

- خلق ایده
 - طراحی پژوهش
 - جمع‌آوری اطلاعات
 - سازماندهی و تحلیل اطلاعات
 - تولید گزارش
 - انتشار
- روبوداک می‌تواند در هر یک از مراحل فوق خدماتی را به عنوان یک دستیار شخصی ماشینی به پژوهشگر ارائه دهد. در این مرحله از توسعه روبوداک، که مرحله آغازین و اولیه آن محسوب می‌شود، روی ارائه خدمات ائلیه نظیر جستجو و خواندن از روی متون تاکید می‌شود. همچنین برای مدل تعاملی روبوداک با پژوهشگر نیز روی روش‌های تعامل صوتی و تصویری تمرکز می‌شود.

۴-۱. ساختار گزارش طرح

این گزارش علاوه بر این فصل، از هفت فصل دیگر تشکیل شده است. در فصل بعدی، مدل مفهومی روبوداک، انواع خدماتی که می‌توان از روبوداک انتظار داشت، و نیازمندی‌های اصلی برای

¹⁹ Tele-Lab

²⁰ Optical Character Recognition-OCR

روبوداک در این مرحله از توسعه معرفی می‌گردد. در فصل سوم، پلتفرم فیزیکی رובوداک مشخص می‌شود. در این فصل ابتدا انواع پلتفرم‌های ربات‌های اجتماعی معرفی می‌شوند سپس پلتفرم مناسب با توجه به الزامات و محدودیت‌های پیش رو انتخاب می‌شود.

در فصل چهارم، قابلیت‌های پلتفرم انتخابی تشریح می‌شود و راه‌حل‌های بکارگرفته شده برای سفارشی‌سازی آن معرفی می‌شوند.

در فصل پنجم، سیستم روبوداک، معماری و اجزای آن تشریح می‌گردد. در این فصل معماری مفهومی و فیزیکی روبوداک به همکار بخشی از کدهای اصلی برنامه روبوداک مشخص تشریح می‌گردد.

در فصل ششم، نتایجی از پیاده‌سازی بخش‌های مختلف سیستم روبوداک ارائه می‌گردد و عملکرد زیرسیستم‌های آن مورد بررسی قرار می‌گیرد.

در فصل هفتم، برنامه توسعه روبوداک در آینده تشریح می‌شود و مجموعه‌ای از پروژه‌ها برای توسعه آن معرفی و اولویت‌بندی می‌شوند. در فصل هشتم نیز خلاصه‌ای از دستاوردهای این طرح ارائه می‌گردد.

فصل دوم:

خدمات روبوداک و مدل مفهومی

۲. خدمات روباتک و مدل مفهومی

در این فصل، خدمات اصلی که روباتک می‌تواند به یک پژوهشگر ارائه دهد، مشخص می‌شود. همچنین مدل مفهومی روباتک برای مرحله آغازین ترسیم می‌گردد.

۲-۱. طراحی خدمات روباتک

در این بخش، نوع خدماتی را که روباتک می‌تواند در هر یک از مراحل اصلی پژوهش برای پژوهشگر ارائه دهد، با توجه به قابلیت‌های متصور برای آن، مشخص شده است. بدیهی است که این خدمات بسته به توانمندی‌های روباتک و توسعه آن در فازهای بعدی می‌تواند ارتقاء یابد. در شکل ۲-۱ خدمات روباتک برای هر یک از گام‌های پژوهش مشخص شده است. در ادامه شرح مختصری برای هر یک از خدمات به تفکیک گام‌های فرایند پژوهش ارائه می‌شود.

۲-۱-۱. خلق ایده

منظور از خلق ایده فرایندی که طی آن مساله پژوهشی شناسایی و تعریف می‌شود و یک مسیر جدید پژوهشی برای حل آن ترسیم می‌گردد. خدماتی که روباتک می‌تواند برای این گام ارائه دهد عبارتند از:

□ دریافت و سازماندهی ایده‌های پژوهشگر

برای تولید و توسعه ایده، روش‌های متنوعی وجود دارد، نظیر تفکر غایب (فکر کردن درباره آنچه که غایب است)، طوفان فکری، تجزیه مساله به مسائل کوچکتر، نگاشت ذهن، روش اسکمپر^{۲۱}، به چالش کشیدن مفروضات و استوری‌برد. برای عملیاتی شدن هر یک از این روش‌ها ابزارهای متنوعی با پشتیبانی کامپیوتر و فناوری وجود دارد. به عنوان مثال ابزارهای نقشه‌های مفهومی برای خلق و سازماندهی ایده‌ها بسیار موثر هستند. بنابراین در قدم اول، پژوهشگر می‌تواند ایده‌های اولیه خود را در اختیار روباتک قرار دهد تا ربات بتواند با برقراری ارتباط مفهومی بین آن و سایر زمینه‌های پژوهشی، ایده را سازماندهی کند. این سازماندهی مسلماً باید براساس روش‌های پردازش زبان طبیعی و

²¹ SCAMPER

پردازش متن صورت بگیرد. حتی می‌توان امکان استفاده همزمان از چندین روش خلق ایده را نیز با استفاده از روباتیک فراهم کرد. در گام‌های بعدی نیز روباتیک می‌تواند با استفاده از ابزارهایی که برای پشتیبانی روش‌های خلق ایده وجود دارد، به پژوهشگر در توسعه ایده کمک کند.

□ کمک به توسعه پژوهش و بهبود ایده

استفاده از ابزارهایی برای مدیریت و توسعه ایده (مانند SPIGIT و IdeasMine) می‌تواند در توسعه و بهبود ایده موثر باشد. در حال حاضر نیز ابزارهایی که مبتنی بر روش‌های جمع‌سپاری^{۲۲} هستند مانند ابزارهای طوفان فکری آنلاین، برای توسعه ایده می‌توانند استفاده شوند. علاوه بر اینکه روباتیک می‌تواند با دریافت ایده‌های پژوهشگر در سازماندهی و توسعه آنها موثر باشد، می‌تواند با رصد حوزه‌های تحقیقاتی مورد علاقه پژوهشگر، وی را از نوآوری‌ها و تازه‌های آن حوزه‌ها باخبر سازد تا زمینه‌ساز خلق ایده‌های جدید برای پژوهشگر شود.

۲-۱-۲. طراحی پژوهش

در این گام از پژوهش، پژوهشگر درصدد طراحی مسیر برای حل مساله پژوهشی است. برای این منظور لازم است که پژوهشگر ابزارها، متدولوژی‌ها و روش‌های مورد نیاز برای جمع‌آوری اطلاعات پژوهش و توسعه آن را طراحی یا تعیین نماید. روباتیک نیز می‌تواند در قالب خدمات زیر به پژوهشگری یاری برساند:

□ کمک در انتخاب متدولوژی و ابزارهای پژوهش

متدولوژی یا روش انجام پژوهش، یکی از مهم‌ترین بخش‌های هر پژوهش علمی است. یکی از خدمات ربات پژوهشیار، ارائه مشاوره در انتخاب روش‌شناسی مناسب بر اساس نوع پژوهش است. مثلاً بر اساس اینکه یک پژوهش ماهیت کیفی یا کمی و یا تلفیقی دارد، ممکن است روش‌شناسی متناسب با آن از قبیل آزمایشگاهی یا توصیفی پیشنهاد شود. همچنین به همراه انتخاب متدولوژی، ابزارهای مناسب برای انجام پژوهش (نظیر پرسشنامه، نرم‌افزارهای در دسترس و غیره) پیشنهاد می‌شود. برای این ارائه این خدمت و پشتیبانی پژوهشگر در انتخاب متدولوژی مناسب، روباتیک می‌تواند از روش‌های تصمیم‌گیری نظیر درخت تصمیم استفاده کند.

²² Crowdsourcing

□ کمک در انتخاب و طراحی شیوه جمع‌آوری اطلاعات

مرحله دیگر در انجام پژوهش‌های علمی، جمع‌آوری داده‌ها و تحلیل آنهاست. در این مرحله نیز، ربات پژوهشیار بر اساس ماهیت کمی یا کیفی پژوهش ممکن است روش‌های مختلف جمع‌آوری داده‌ها نظیر مشاهده، پرسشنامه، مصاحبه و غیره را پیشنهاد دهد. برای هر یک از این روش‌ها نیز شیوه تحلیل آنها و ابزارهای تحلیل می‌تواند توسط روباتیک پیشنهاد شود.

۲-۱-۳. جمع‌آوری اطلاعات

یکی از مهمترین گام‌های پژوهش، جمع‌آوری و سازماندهی اطلاعات مورد نیاز است. پژوهشگر لازم است براساس روش و متدولوژی که برای انجام پژوهش در نظر گرفته، اطلاعات مورد نیاز را اعم از مقالات و مستندات تحقیقاتی پیشین و داده‌های مرتبط با حوزه پژوهش، جمع‌آوری نماید تا با تحلیل این اطلاعات در گام‌های بعدی، اهداف پژوهش حاصل شود. نقش ربات پژوهشیار در این گام بسیار پررنگ است. امکان جستجوی (هوشمند) اطلاعات در پایگاه‌های مختلف، دسته‌بندی و مصورسازی نتایج جستجو، و ذخیره‌سازی آنها از جمله خدمات کلیدی است که روباتیک می‌تواند ارائه دهد. خدمات تفصیلی روباتیک برای این گام در قالب موضوعات زیر قرار می‌گیرند:

□ جستجو در پایگاه وب و گنج

جستجوی اطلاعات جزو مهمترین و اساسی‌ترین توانمندی روباتیک محسوب می‌شود. علاوه بر جستجو در وب، لازم است که امکان جستجو در پایگاه‌های تخصصی علمی، به طور خاص پایگاه گنج، نیز توسط روباتیک فراهم شود.

□ جستجوی هم‌زمان در موتورهای علمی و پایگاه دانش روباتیک

با توجه به افزایش حجم اطلاعات ذخیره شده در وب، یک موتور جستجو واحد، هرچقدر هم قوی باشد، نمی‌تواند جوابگوی نمایه‌سازی تمامی این اطلاعات باشد. بنابراین راه‌حلی که معمولاً برای توسعه نتایج جستجو و پوشش بیشتر اطلاعات پیشنهاد می‌شود، جستجوی هم‌زمان در چندین موتور جستجو و تجمیع و یکپارچه‌سازی نتایج بازیابی است، که به آن «جستجوی هم‌بسته»^{۲۳} هم می‌گویند. در این نوع جستجو، یک عبارت پرس‌وجو به چندین پایگاه داده یا منابع اطلاعاتی با قالب‌های همخوان با آن پایگاه‌ها ارسال می‌شود، نتایج جستجو با هم ادغام می‌شود و در نهایت در یک فرمت

²³ Federated search

مناسب و واحد، با کمترین افزونگی، برای کاربر نمایش داده می‌شود. در حال حاضر موتورهای جستجو همزمان متنوعی به صورت منبع باز و آزاد وجود دارد. برخی از آنها براساس روش‌های فراجستجو^{۲۴} عمل می‌کنند. در همین راستا، ربات پژوهشیار نیز می‌تواند با داشتن قابلیت جستجوی همزمان در چندین پایگاه اطلاعاتی، نتایج جستجو را توسعه و بهبود دهد.

علاوه بر آن، خود ربات نیز دارای یک پایگاه دانش است که اطلاعات مربوط به نیازهای پژوهشگر را در آن ذخیره کرده و تاریخچه‌ای از علاقه‌مندی‌ها و اطلاعات مقالات و مستندات علمی پژوهشگر را دربر دارد. ربات پژوهشیار می‌تواند جستجو را به صورت همزمان در پایگاه اطلاعاتی خود در بین اطلاعات نمایه‌سازی شده نیز انجام دهد و نتایج جستجو در پایگاه‌های اطلاعاتی بیرونی را با این نتایج تلفیق نماید. پایگاه دانش روباتک می‌تواند تاریخچه جستجوهای پژوهشگر را نیز دربرداشته باشد. در نهایت این ربات می‌تواند با توجه به این تاریخچه، و درک مناسبی که از علاقه‌مندی‌های پژوهشگر (با فرایند یادگیری و تطبیق با پژوهشگر حاصل شده) پیدا کرده، جستجو در پایگاه‌های اطلاعاتی را غنی‌سازی کند و نتایجی را بازایی نماید که با نیازهای پژوهشگر بیشتر منطبق هستند.

□ توسعه کلیدواژه‌ها براساس اصطلاح‌نامه یا آنتولوژی

روباتک می‌تواند با اتکا به ارتباط معنایی واژگان که براساس اصطلاح‌نامه یا آنتولوژی تعریف شده، عبارات جستجو را غنی‌تر کند تا نتایج بهتری بازایی شوند. حتی روباتک با توسعه کلیدواژه‌ها می‌تواند دامنه واژگان تخصصی پژوهشگر را نیز توسعه دهد.

□ قابلیت دسته‌بندی و سازماندهی متون بازایی شده

با توجه به حجم بالای اطلاعات در دنیای امروز، سازماندهی و دسته‌بندی اطلاعات به منظور بازایی اثربخش آن بسیار مورد توجه است. پژوهشگران نیز از این قاعده جدا نیستند و برای اینکه بتوانند در جریان یک پژوهش به صورت موثر از اطلاعاتی که جمع‌آوری کرده‌اند استفاده کنند باید بتوانند آنها را به خوبی دسته‌بندی کنند. البته این دسته‌بندی همیشه کار ساده‌ای نبوده و در مواردی که حجم اطلاعات بالا است تقریباً غیرممکن می‌نماید. در چنین مواردی استفاده از روش‌های دسته‌بندی اطلاعات که مبتنی بر تکنیک‌های داده‌کاوی و هوش مصنوعی است می‌تواند موثر واقع شود. روباتک با مجهز بودن به ابزارهای داده‌کاوی و هوش مصنوعی، اطلاعاتی را که کاربر در اختیار او

²⁴ Meta search

قرار داده و همچنین از گذشته در خاطر (حافظه) دارد، به شکل هوشمند سازماندهی و دسته‌بندی کرده و بدین وسیله در پرسش‌های کاربر پیشنهاداتی را نیز ارائه خواهد کرد. علاوه بر آن روباتک می‌تواند نتایج حاصل از جستجوی پژوهشگر را نیز به صورت معنایی دسته‌بندی کند تا تحلیل آنها ساده‌تر گردد.

□ ذخیره‌سازی رکوردهای بازیابی شده و ترجیحات پژوهشگر به صورت همزمان استفاده از ترجیحات پژوهشگر برای ذخیره‌سازی اطلاعات بازیابی شده، به معنی اینست که دسته‌بندی و نمایه‌گذاری اطلاعات براساس حوزه‌های پژوهشی، ترجیحات و بازخوردهای کاربر صورت بگیرد. روباتک می‌تواند با تعاملی که با پژوهشگر دارد، رفتار پژوهشی و علاقه‌مندی‌های وی را یاد بگیرد و در نتیجه در فرایند ذخیره‌سازی رکوردها، ترجیحات پژوهشگر را لحاظ کند. بنابراین عبارت همزمان به مفهوم سفارشی کردن و شخصی‌سازی کردن فرایند ذخیره‌سازی و دسته‌بندی اطلاعات براساس نیاز پژوهشگر است. بنابراین روباتک می‌تواند فرایندهای خود را با رفتار پژوهشی یک پژوهشگر تطبیق دهد.

□ ذخیره‌سازی معنایی به صورت شبکه‌ای براساس پایگاه داده گرافی پایگاه‌های داده گرافی برای ذخیره‌سازی داده‌هایی بکار می‌روند که ارتباطات پیچیده و بسیاری بین آنها دیده می‌شود که با استفاده از پایگاه‌های داده رابطه‌ای نمی‌توان تمامی این ارتباطات را مدل کرد. نمونه بارز این داده‌ها مربوط به شبکه‌های اجتماعی هستند. داده‌های مربوط به یک پژوهش نیز می‌تواند به صورت شبکه‌ای ذخیره شود. به طور خاص اطلاعات مربوط به مقالات و مستندات پژوهشی پیشین و مرتبط می‌توانند براساس ارتباط معنایی که براساس فراداده‌های آنها تعریف شده، به صورت شبکه‌ای ذخیره‌سازی شوند و نمایش پیدا کنند. برای این منظور ابزارهای متنوعی وجود دارد که روباتک می‌تواند از آنها هم استفاده کند، که مهمترین آنها Neo4j است که ابزار قوی برای پایگاه داده گرافی به شمار می‌رود.

□ تحلیل ساختاری و مصورسازی نتایج جستجو مستندات علمی معمولاً دارای ساختار نوشتاری خاصی هستند. به عنوان مثال محتوی یک مقاله علمی معمولاً دارای بخش‌هایی نظیر چکیده، مقدمه، روش، نتایج،... است. تحلیل ساختاری مستندات علمی بازیابی شده می‌تواند در تحلیل، دسته‌بندی و نمایش آنها موثر باشد. علاوه بر آن نمایش نتایج جستجو به صورت تصویری و عینی با استفاده از روش‌های مصورسازی اطلاعات و ابزارهای مربوط به آن

می‌تواند پژوهشگر را در استخراج مطالب مرتبط یاری برساند. در حال حاضر ابزارهای متنوعی برای مصورسازی نتایج جستجو وجود دارد که تمرکز آنها روی دسته‌بندی و خوشه‌بندی نتایج است. برای نمایش تصویری نتایج جستجو لازم است که ارتباط مفهومی بین رکوردهای بازیابی شده برقرار شود. این ارتباط یا می‌تواند در سطح فراداده رکوردها صورت بگیرد یا در سطح محتوی آنها. بکارگیری مفاهیم وب معنایی برای برقراری ارتباط معنایی بین رکوردهای بازیابی شده ضروری است. روباتک در این زمینه می‌تواند علاوه بر بهره‌گیری از ابزارهای مصورسازی موجود، تحلیل عمیق‌تری را نیز روی نتایج جستجو مطابق با نیازهای پژوهشگر انجام دهد و نتایج را برای اساس علاقه‌مندی‌های پژوهشگر و حوزه‌های مطالعاتی وی به صورت تصویری نمایش دهد.

□ ارزیابی و اعتبارسنجی رکوردهای علمی

در حال حاضر موتورهای جستجو قابلیت بازیابی حجم وسیعی از مستندات علمی را برای یک عبارت جستجو دارند. در بین این رکوردها ممکن است نتایجی هم ارائه شود که کاملاً به عبارت جستجو مرتبط هستند لیکن از نظر علمی و استناد از اعتبار کافی و قوی برخوردار نیستند. بنابراین علاوه بر نمایش نتایج جستجو براساس میزان ارتباط آنها با عبارت جستجو، لازم است که پژوهشگر بداند که یک رکورد علمی بازیابی شده از نظر علمی و شاخص‌های آکادمیک تا چه میزان معتبر و قابل استناد است. برای این منظور می‌توان شاخص‌های متعددی را در نظر گرفت، به عنوان مثال بررسی اعتبار منبعی که آن رکورد علمی از آنجا بازیابی شده، تعداد استنادها به آن، نویسندگان و وابستگی علمی آنها، و میزان به روز بودن رکورد بازیابی شده. بنابراین روباتک نیز می‌تواند براساس این شاخص‌ها رکوردهای علمی بازیابی شده را دسته‌بندی کند و مواردی را که دارای اعتبار بیشتری هستند برای پژوهشگر مشخص نماید.

۲-۱-۴. سازمان‌دهی و تحلیل اطلاعات

در این گام از پژوهش، پژوهشگر با جمع‌آوری اطلاعات مورد نیاز خود، لازم است که براساس شیوه و روش پژوهش، این اطلاعات را با توجه به هدف پژوهش سازماندهی و تحلیل نماید. در حقیقت در این گام، پژوهشگر نوآوری اصلی در پژوهش را با توجه به اطلاعات جمع‌آوری شده، توسعه می‌دهد. برای این منظور، در دسترس بودن ابزارهای متناسب با پژوهش برای تحلیل اطلاعات، ترسیم نمودارهای مفهومی و حتی آماری، مسلماً می‌تواند در برآورده ساختن اهداف پژوهش موثر باشد. علاوه بر آن، تحلیل مستندات علمی مرتبط با پژوهش موردنظر و تحلیل محتوی مستندات و ارتباط

آنها با هم (با کمک روش‌های علم‌سنجی) می‌تواند به درک بهتر مساله پژوهشی و پژوهش‌های مرتبط با آن کمک کند. در این گام روبوداک با ارائه خدمات پشتیبانی زیر می‌تواند به پژوهشگر یاری برساند:

□ مدیریت ارجاعات پژوهشگر

امروزه با توجه به حجم بالای انتشارات علمی در دنیا، پژوهشگران در جریان پژوهش‌های خود، مدارک علمی زیادی را مورد استفاده قرار داده و در نوشته‌جات علمی خود از آنها استفاده می‌کنند. مدیریت این مستندات و همچنین ارجاعات پژوهشگر درون یک مقاله یا کتاب از اهمیت و ارزش بالایی برخوردار است. در حال حاضر نرم‌افزارهای مختلفی همچون اندنوت^{۲۵} محصول شرکت تامسون رويترز یا زوترو^{۲۶} - که یک پروژه کد باز است - تولید شده‌اند که به پژوهشگر در مدیریت ارجاعات و تولید کتابخانه‌ها کمک می‌کنند.

روبوداک می‌خواهد به صورت فعال و در تعامل دوسویه با کاربر خود ارجاعات و کتابخانه‌های مستندات علمی را مدیریت کرده و همچنین بتواند بر اساس تحلیل‌های اولیه متن و جستجوی کلیدواژه‌های در کتابخانه‌ها، پیشنهاداتی را نیز برای ارجاع درون متون و مقالات علمی ارائه نماید.

□ ارائه جعبه ابزار تحلیل اطلاعات

تحلیل اطلاعات در بسیاری از پژوهش‌ها از اهمیت برخوردار است به گونه‌ای که پژوهشگر لازم است اطلاعات را از منظر کمی و کیفی مورد تحلیل و مقایسه قرار دهد. در حال حاضر ابزارهای متعددی برای تحلیل اطلاعات وجود دارد که از آن جمله می‌توان به نرم‌افزار اکسل^{۲۷} تولید شرکت مایکروسافت و یا نرم‌افزار اس.پی.اس.اس^{۲۸} که برای تحلیل آماری اطلاعات استفاده می‌شود، اشاره کرد.

در این راستا روبوداک می‌تواند به عنوان یک دستیار در تحلیل اطلاعات، بیشتر از منظر کمی، مورد استفاده قرار گرفته و کاربر را در استفاده از نرم‌افزارهای پیش‌گفته یاری دهد. به عنوان مثال از روبوداک خواسته می‌شود که مجموعه‌ای از داده‌های استخراج شده را در قالب خاصی کنار هم چیده

²⁵ Endnote

²⁶ Zotero

²⁷ Excel

²⁸ SPSS

و آنها را نرمال سازی کند. نمونه ای دیگر بررسی همبستگی بین دو ستون از داده ها با ضریب اطمینان مشخص است.

□ تحلیل شبکه مولفان و واژگانی پایگاه دانش (ابزارهای علم سنجی)

ابزارهای علم سنجی می توانند در بسیاری از پژوهش ها به خصوص در مرحله مرور ادبیات و نگاشت نقشه دانش آن حوزه، پژوهشگران را یاری دهند. این ابزارها کمک می کنند که پژوهشگر انتشارات کلیدی در حوزه مورد نظر، دانشمندان محوری، سیر تطور حوزه و همچنین مفاهیم اصلی را شناسایی کنند.

روباتک با استفاده از تکنیک های علم سنجی خواهد توانست علاوه بر دسته بندی اطلاعات (مقالات) به شکل هوشمند، نویسندگان کلیدی، مفاهیم کلیدی و مقالات پراستناد و مهم را شناسایی کرده و این اطلاعات را به فراخور نیاز در اختیار کاربر خود قرار دهد. روباتک این کار را با استفاده از ابزارهای شناسایی هم تالیفی، هم واژگانی و تحلیل استنادی انجام خواهد داد. روباتک به یک پایگاه دانش مجهز خواهد شد که اطلاعات را به شکل شبکه ای و هستان شناسانه و در قالب جداولی با ارتباطات مفهومی ذخیره سازی می نماید.

□ مصورسازی اطلاعات و ترسیم نمودار

گفته می شود که یک تصویر در بعضی از موارد به اندازه هزار کلمه مفهوم را به مخاطب انتقال می دهد. در همین راستا یکی از روش هایی که به خصوص در ارائه نتایج پژوهش از اهمیت زیادی برخوردار است مصورسازی اطلاعات و ترسیم نمودار است.

روباتک بر اساس اطلاعات و همچنین ابزارهای تحلیلی که در اختیار دارد، به پژوهشگر در مصورسازی و ترسیم نمودار کمک خواهد کرد. قالب این مصورسازی ها و نمودارها از پیش برای روباتک تعریف شده و او با تزریق اطلاعات به درون قالب ها تصاویر و نمودارها را در تولید و در اختیار کاربر قرار خواهد داد. به عنوان مثال روباتک می تواند دو ستون از اطلاعات را که توسط کاربر در اختیار او قرار داده شده به شکل نمودار میله ای یا دایره ای ترسیم کرده و در قالب نرم افزاری مورد نظر به او بازگرداند.

۲-۱-۵. تولید گزارش

هر پژوهش علمی دستاوردها و نتایجی دارد که بدون ارائه و انتشار آن در مجلات علمی فرآیند پژوهش پایان یافته محسوب نمی شود. تهیه گزارش پژوهش برای انتشار آن فرایندی که طی آن

پژوهشگر یافته‌های علمی خود را معمولاً در قالب یک مقاله علمی تهیه می‌کند. پژوهشگر لازم است که علاوه بر تنظیم ساختار گزارش براساس نیازمندی‌های محل انتشار، متن گزارش را نیز بر همان ساختار تهیه کند. در این راستا، روبوداک می‌تواند خدمات زیر را ارائه دهد:

□ کمک در ویرایش ادبی و علمی متن

پس از اینکه متنی اولیه از نتایج پژوهش آماده شد، این متن خصوصاً اگر به زبانی غیر از فارسی (مثلاً انگلیسی) تولید شود نیازمند ویرایش‌های مختلف زبانی خواهد بود. این ویرایش‌ها می‌تواند شامل ویرایش و خطایابی دستوری و نیز تحلیل‌های عمیق متنی برای ارزیابی انسجام متن باشد. روبوداک در این زمینه می‌تواند با بهره‌گیری از ابزارهای ویرایش خودکار متون فارسی و انگلیسی که بر پایه پردازش زبان طبیعی طراحی شده‌اند، به پژوهشگر در ویرایش ادبی و علمی متن یاری برساند.

□ تولید گزارش اولیه به همراه نمودار

پس از ویرایش اولیه متن، گزارشی حاصل می‌شود که بسیاری از خطاهای احتمالی آن پالایش شده است. همچنین نیاز است که متعلقات دیگری از جمله جداول و اشکال در مکان‌های صحیح خود در گزارش قرار گیرند. بسته به اینکه مجله موردنظر کدام قالب را برای قرارگیری اشکال و جداول توصیه کرده، روبوداک می‌تواند پژوهشگر را در چینش صحیح اشکال و جداول در متن گزارش یاری کند.

□ کنترل اصالت گزارش

قبل از اینکه مقاله به مجله‌ای جهت انتشار فرستاده شود، ربات پژوهشیار محتوای آن را به‌لحاظ همسانی با دیگر پژوهش‌های انجام‌شده کنترل می‌کند و موارد احتمالی مشکوک به سرقت علمی را برای پژوهشگر مشخص می‌کند. در پردازش‌های سرقت علمی می‌توان از پایگاه و الگوریتم‌های سامانه همانندجو استفاده کرد.

□ مدیریت ارجاعات گزارش

در این مرحله، ربات تمامی ارجاعات قرارگرفته در متن را کنترل و تایید می‌کند. این کنترل‌ها می‌تواند به شیوه‌های گوناگونی انجام شود. برای نمونه، کنترل ارجاعات می‌تواند در سطح درون‌متنی و بین شماره اشکال و جداول و ارجاع به آنها در داخل متن گزارش باشد. در سطحی دیگر، ربات پژوهشیار، ارجاعات درون متن را با منابع انتهای گزارش تطبیق می‌دهد. همچنین بر اساس قالب مجله، ربات قادر خواهد بود شیوه‌های ارجاع‌دهی مختلف را انتخاب و اعمال کند.

□ تایپ گزارش

در نگارش مقالات علمی، گاهی پژوهشگر نیاز به این دارد که حاصل یادداشت‌برداری‌های خود را که به صورت دستی نوشته شده، در قالب متن تایپ شده درآورد. برای این منظور، می‌توان با استفاده از قابلیت تبدیل گفتار به متن^{۲۹} در ربات پژوهشیار، دست‌نوشته‌ها را برای ربات خواند تا به صورت متن تایپی ذخیره شوند.

□ ذخیره گزارش به صورت ساخت یافته

یکی از قابلیت‌ها ربات، ساختاردهی به یک گزارش علمی خواهد بود. به این صورت که اجزای مختلف گزارش از قبیل مقدمه، نتایج، نمودارها و تصاویر، منابع و غیره هر کدام به صورت مستقل قابل ذخیره‌سازی و بازیابی خواهد بود. این امر می‌تواند در تحلیل‌های آماری بعدی و استخراج اطلاعات ساختارمند از گزارش‌ها سودمند باشد.

۲-۱-۶. انتشار

برای انتشار یافته‌های پژوهش، لازم است که منبع مورد نظر (مجله) که متناسب با حوزه پژوهشی است شناسایی گردد و نتیجه پژوهش (عموما در قالب مقاله) براساس استاندارد نگارشی آن منبع تهیه شود. پس از ارسال مقاله نیز، پژوهشگر معمولا درگیر پیگیری فرایند داوری و اصلاحات احتمالی می‌شود. بنابراین روباتک در این گام می‌تواند در موضوعات زیر به پژوهشگر کمک کند:

□ کمک در انتخاب مجله و منبع مناسب برای انتشار یافته‌های پژوهش

به دلیل افزایش چشم‌گیر تعداد مقالات تولید شده، انتخاب مناسب مجله در مقایسه با سال‌های قبل اهمیت ویژه‌ای را پیدا کرده است. نکاتی که در انتخاب یک منبع برای انتشار لازم است در نظر گرفته شود عبارتست از حوزه کاری و تحقیقاتی، ضریب تاثیر مجله، زمان انتشار، هزینه انتشار، قالب مقاله در ارسال اولیه مقاله، نمایه آن در پایگاه‌های معتبر، تعداد شمارگان و میزان سادگی در پذیرش. روباتک با بهره‌گیری از یک سیستم خودکار می‌تواند اطلاعات مربوط به پایگاه داده خود را که بر اساس حوزه کاری پژوهشگر شکل گرفته به روزرسانی نموده و براساس خواسته‌های پژوهشگر فهرست مجلات پیشنهادی را بر اساس اولویت در اختیار پژوهشگر قرار دهد. علاوه بر آن روباتک می‌تواند با اتصال به سیستم‌های جستجوی مجله در الزویر و اشپرینگر، بر غنای پایگاه داده خود بیفزاید.

²⁹ Text to speech

□ تبدیل قالب مقاله

یکی از امور زمان بر در ارسال مقاله به مجلات مختلف، تهیه مقاله در قالب مجله مورد نظر است. حذف تعدادی مجله به دلیل قالب خاص پذیرش مقاله، شانس پذیرش مقاله را کاهش می‌دهد. لذا روباتک می‌تواند در تبدیل خودکار قالب نوشتاری مقاله به قالب‌های مدنظر یک مجله کمک کند. البته ارائه این خدمت به صورت کاملاً خودکار در حال حاضر چالش برانگیز به نظر می‌رسد، لیکن با وجود زبان‌های نوشتاری مانند لاتکس^{۳۰} این امر قابل دسترس در آینده خواهد بود. به عنوان مثال روباتک می‌تواند با بهره‌گیری از یک رویه هوشمند، پایگاهی از قالب‌های مختلف مجلات را جمع‌آوری نموده تا در زمان نیاز در کوتاه‌ترین زمان یک مقاله را در قالب‌های مختلف تولید نماید.

□ ردیابی مقالات پژوهشگر و اطلاع رسانی

پیگیری وضعیت مقالات ارسالی به مجلات و کنفرانس‌ها یکی از مهمترین دغدغه‌های پژوهشگر پس از ارسال مقاله پژوهشی است. هنگامی که تعداد مقالات ارسالی از حدی بیشتر می‌شود، پیگیری امور مربوط به آنها نیازمند مدیریت دقیق است. مقالات ارسالی به مجلات پس از اتمام داوری عموماً نیاز به اصلاحات دارند که معمولاً زمان انجام این اصلاحات هم محدود است. روباتک می‌تواند به عنوان یک منشی پژوهشی در مدیریت و اطلاع رسانی در این زمینه کمک کند.

یکی از شاخص‌هایی که پژوهشگران در دنیا براساس آن سنجیده می‌شوند، میزان ارجاعات به مقالات منتشر شده ایشان است. پایگاه‌هایی مانند گوگل اسکولار و وب آو ساینس^{۳۱} میزان ارجاعات را بصورت روزانه گزارش می‌نمایند. روباتک می‌تواند با اتصال و دریافت اطلاعات این پایگاه‌ها، گزارشی از وضعیت موجود پژوهشگر در اختیار وی قرار دهد.

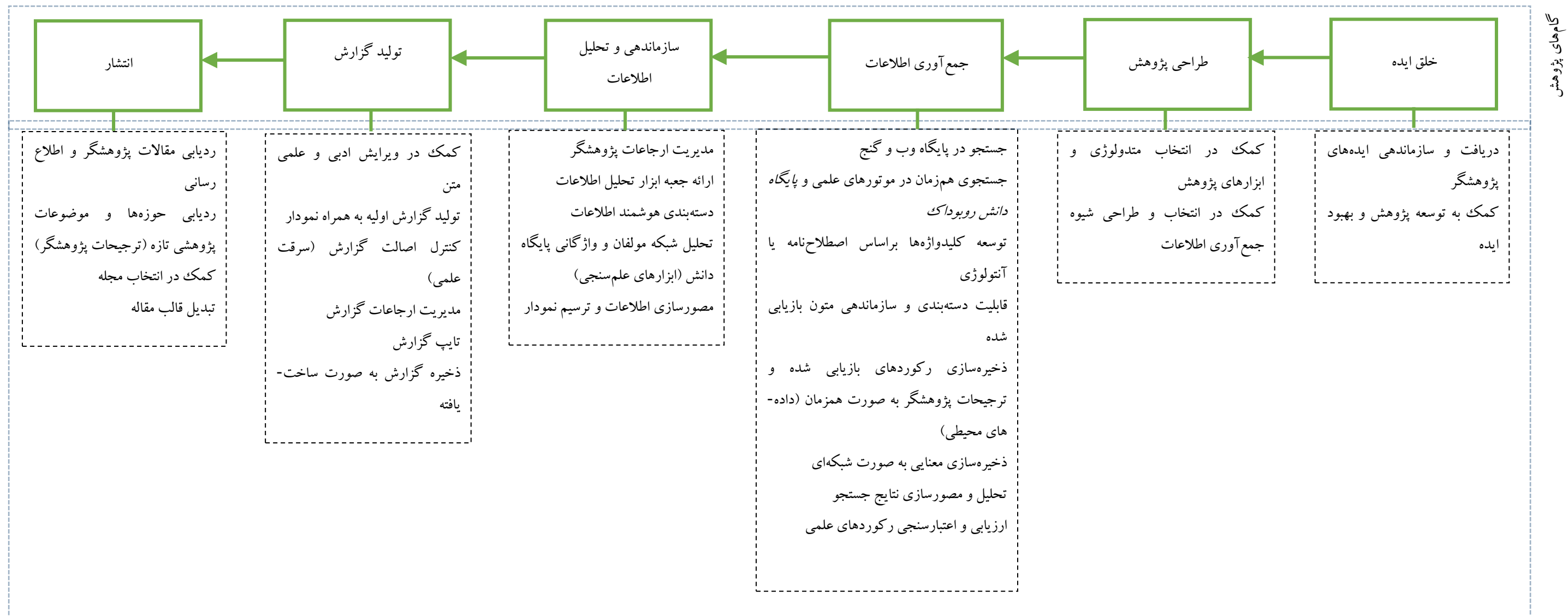
□ ردیابی حوزه‌ها و موضوعات پژوهشی تازه (ترجیحات پژوهشگر)

روزانه در پایگاه‌های معتبر شاهد انتشار مقالات جدید و تحقیقات پژوهشی جدید هستیم. از طرفی به دلیل رشد فراوان مقالات، حجم انبوهی از تحقیقات در حال شکل‌گیری است. به دلیل حجم بالای تحقیقات، ردیابی حوزه‌ها و موضوعات پژوهشی بصورت دستی و سنتی و یا استفاده از ابزار جستجوی ساده در پایگاه‌های داده، کاری زمان بر و مشکل است. لذا برای رفع این مشکل، روباتک با اطلاع از زمینه‌های تحقیقاتی پژوهشگر خود و ایجاد یک مدل مفهومی بطور خودکار و روزانه عمل جستجو در

³⁰ Latex³¹ Web of Science

یافته‌های جدید را با سرعت مناسب انجام داده و حوزه‌های پژوهشی جدید را در قالب‌های مختلف در اختیار پژوهشگر می‌گذارد.

لازم به ذکر است خدماتی که در این بخش برای روباتك ارائه شد، می‌تواند در فازهای بعدی توسعه روباتك، ارتقاء و گسترش یابد.



شکل ۱-۲- خدمات روبوداک براساس گام‌های اصلی پژوهش

۲-۲. قابلیت‌های اصلی ربات اجتماعی

پیش از بررسی انواع ربات‌های اجتماعی، لازم است که ویژگی‌های اصلی که یک ربات را در دسته ربات‌های اجتماعی قرار می‌دهد، مشخص شود، این ویژگی‌ها عبارتند (Salichs et al., 2006):

- چندحسی بودن^{۳۲}: قابلیت چندحسی بودن به انسان‌ها این امکان را می‌دهد که به صورت بدون وقفه^{۳۳} بین حس‌های مختلف تعاملی، از حس بینایی به شنوایی، به حس لامسه، براساس تغییرات در شرایط و ترجیحات کاربر، حرکت کنند. یک ربات اجتماعی هم باید رابط‌های چندحسی داشته باشد، که این رابط‌ها بتوانند صحبت، متن نوشته‌شده، زبان بدن، ایما و اشارات، حرکت لب‌ها یا چشم و سایر راه‌های ارتباطی را با هم یکپارچه کنند تا درک بهتری از انسان داشته باشند و با او موثرتر ارتباط برقرار کنند.
- شخصیت^{۳۴}: ربات‌های اجتماعی معمولاً دارای شخصیت‌های متمایزی هستند. شخصیت نوعی مدل مفهومی است که رفتار، باورها و نیت را در قالب رفتاری منسجم و باثبات هدایت می‌کند. شخصیت‌دار کردن ربات موجب می‌شود که کاربران درک خوبی از رفتار ربات داشته باشند و انتظاراتشان در حد توانایی‌های ربات باشد. برای نمونه، ربات کیسمت^{۳۵} در دانشگاه ام.آی.تی شخصیت یک کودک را دارد که موجب می‌شود خطاها و سوءبرداشت‌های ربات طبیعی به نظر بیاید.
- سازگاری^{۳۶}: یکی از تفاوت‌های مهم میان جانداران و غیرجانداران، قابلیت سازگاری است. یک ربات اجتماعی باید تعاملی سازگارانه بسته به سطح دانش کاربر داشته باشد. این تعامل می‌تواند از سطح بسیار تخصصی مهندسان هوش مصنوعی تا کاربران غیرحرفه‌ای را شامل شود.
- خودمختاری^{۳۷}: ربات‌های اجتماعی باید سطح بالایی از خودمختاری را داشته باشند تا بتوانند در موقعیت‌های مختلف فعالیت کنند. در برخی موقعیت‌ها یک ربات اجتماعی باید قادر باشد به صورت خودمختار عمل کند تا برای مثال سوء برداشت‌های کاربر را تصحیح کند یا از

³² Multimodality

³³ Seamlessly

³⁴ Personality

³⁵ Kismet

³⁶ Adaptivity

³⁷ Autonomy

عمل به دستورات کاربر امتنا کند تا مانع بروز خطا و تناقض در سیستم شود. ربات‌های سنتی بر مبنای الگوی «ارباب و برده» عمل می‌کنند و بیشتر نقش یک ابزار را دارند. اما برای ربات‌های شخصی و اجتماعی این الگو نمی‌تواند مناسب باشد. مثلاً در مورد ربات‌هایی که نقش حیوانات خانگی را ایفا می‌کنند، ربات باید حس جاندار بودن را القا کند نه صرفاً یک ماشین محسوب شود. ویژگی ضروری موجودات زنده که باید در برخی ربات‌ها وجود داشته باشد، خودمختاری مبتنی بر خودانگیزش است. این ربات‌ها باید قادر باشند به جای تبعیت کورکورانه از فرامین اپراتور، به‌طور خودمختار خودشان اعمال و اهدافشان را انتخاب کنند.

□ قابلیت یادگیری: یادگیری یکی از ویژگی‌های مهم ربات‌های اجتماعی است. ربات باید بتواند از تجربه‌های تعاملات قبلی‌اش یاد بگیرد. یادگیری به ربات اجتماعی کمک می‌کند اعمال جدید را یاد بگیرد یا موارد تکراری را به‌صورت روتین یا خودکار انجام دهد. در تعامل میان انسان و ربات، برخی محققان یادگیری یک ربات را به دو دسته «یادگیری برای برقراری ارتباط» (یعنی یادگیری اینکه چگونه با دیگر کاربران انسانی ارتباط برقرار کند) و «یادگیری از ارتباط» (که مثلاً شامل یادگیری از طریق مشاهده می‌شود) تقسیم کرده‌اند (Klingspor, Demiris, & Kaiser, 1997). بریزیل و همکاران در سال ۲۰۰۴ تعدادی از چالش‌های اساسی را برای طراحی ربات‌هایی که از راهنمایی انسان‌ها یاد می‌گیرند، فهرست کرده‌اند (Breazeal et al., 2004). این چالش عبارتند از: دانستن آنچه مهم است، دانستن اینکه کدام کنش را انجام دهد، دانستن اینکه چگونه موفقیت را تشخیص دهد و اشتباهات را تصحیح کند و اینکه بداند چگونه کاوش کند.

□ همکاری^{۳۸}: یک ربات اجتماعی باید بتواند با دیگر ربات‌ها برای دستیابی به اهداف فردی و جمعی همکاری کند. این اهداف بسته به اینکه ربات‌های همکاری‌کننده هدف‌محور هستند یا نه، ممکن است به‌صورت آشکار برای ربات مشخص باشند یا نباشند.

□ واکنش‌گری^{۳۹}: واکنش‌گری یکی دیگر از ویژگی‌های مهم ربات‌های اجتماعی است. یکی ربات واکنش‌گر می‌تواند محیط خود را ادراک کرده و به‌موقع به تغییرات آن واکنش نشان

³⁸ cooperativeness

³⁹ reactivity

دهد. برای مثال، یک ربات اجتماعی باید بتواند مواردی را تشخیص دهد که در آن دانش، خواسته‌ها و نیت انسانی با دانش، قابلیت‌ها و کارکردهای ربات در تضاد است.

□ قابلیت کنشگری^{۴۰}: یک ربات اجتماعی همچنین باید کنشگر باشد. به این معنی که یک ربات نه تنها باید به رویدادهای خارجی واکنش نشان دهد، بلکه باید بتواند رفتاری هدف‌گرا^{۴۱} از خود نشان دهد و در صورت لزوم ابتکار عمل را در دست بگیرد (مثلاً اعمال برخلاف انتظار از خود نشان دهد) تا اطلاعات انسان را تصحیح کند یا راه‌حل‌های جایگزین پیشنهاد دهد. در این شرایط، رفتار ربات به‌واسطه برنامه‌ها و اهداف منطقی هدایت می‌شود.

۳-۲. نیازمندی‌های اصلی برای سیستم روباتیک در مرحله آغازین

سیستم روباتیک، مانند هر سیستم دیگری دارای سه جزء اصلی است: ورودی، پردازنده، و خروجی. در این بخش، الزامات روباتیک برای هر یک از این سه جزء، با توجه به قابلیت‌های مورد انتظار در مرحل آغازین از توسعه روباتیک، مورد بررسی قرار می‌گیرد.

درکی که یک ربات اجتماعی از محیط خود دارد، در قالب ورودی‌ها، دریافت می‌شود و پس از پردازش آنها، با توجه به قابلیت‌های روباتیک، خروجی مناسب تولید و ارائه می‌گردد. داده‌های ورودی می‌توانند از نوع صوت، تصویر، دما، یا درک تغییرات حرکتی از طریق سنسورهای شتاب‌سنج یاژیروسکوپ باشند.

در مرحله آغازین از توسعه روباتیک، خدمات زیر مورد انتظار است:

۱. درک گفتار و متن فارسی و لاتین (در قالب یکسری جملات مشخص و از پیش تعیین شده)
۲. جستجو در متون بویژه پایگاه اطلاعاتی گنج و بازیابی اطلاعات به همراه دسته‌بندی اولیه متون همراه با بکارگیری اصطلاحنامه‌های تخصصی
۳. خواندن متون فارسی و انگلیسی
۴. تشخیص تعداد محدودی از علایم و اشارات کاربر (در قالب یکسری علایم از پیش تعریف شده)

⁴⁰ Proactiveness

⁴¹ Goal-oriented

برای این منظور، لازم است که امکانات رباتی که به عنوان پلتفرم روبوداک در نظر گرفته شده، بررسی شود و محدودیت‌های احتمالی آن برای حصول به خدمات فوق، شناسایی گردد و راه حل مناسب برای برطرف کردن آن محدودیت‌ها و سفارشی نمودن پلتفرم ارائه شود.

خدمت ۱ نیازمند وجود یک سیستم پردازش صوت و گفتار است که بتواند گفتار پیوسته را به متن تبدیل نماید.

خدمت ۲ نیازمند یک زیرسیستم پردازش متن است که بتواند عبارت مورد نظر کاربر را در پایگاه داده گنج جستجو نماید و نتایج را با استفاده از ابزارهای مصورسازی اطلاعات و با استفاده از رویه‌های دسته‌بندی به کاربر نمایش دهد.

خدمت ۳ نیز نیازمند وجود یک سیستم تبدیل متن به گفتار است که از این طریق روبوداک بتواند با فرد به صورت صوتی ارتباط برقرار کند و صحبت نماید.

خدمت ۴ نیازمند وجود سیستم پردازش تصویر و ویدئو است که بتواند علائم محدود و ساده‌ای را به صورت تصویری از کاربر دریافت نماید و آن را پردازش کند.

بنابراین برای بخش ورودی سیستم روبوداک، داده‌های زیر دریافت می‌شود:

- داده‌های تصویر و ویدئویی
- داده‌های صوتی
- فرامینی که مستقیم از طریق کاربر با دسترسی به رابط کاربری سیستم روبوداک ارسال می‌شود. برای این منظور لازم است که یک رابط کاربری گرافیکی به همراه وسیله‌ای برای نمایش آن به کاربر نیز در نظر گرفته شود تا کاربر بتواند در صورت نیاز از طریق آن رابط کاربری هم دستورات خود را به روبوداک ارسال نماید.

برای حصول به خدمات ذکر شده و با توجه به داده‌های فوق، لازم است که روبوداک قابلیت‌های زیر را داشته باشد:

- پردازش صوت و گفتار و تبدیل آن به متن
- پردازش تصویر و تشخیص یک علامت یا حالت مشخص
- پردازش عبارت جستجو و بکارگیری الگوریتم بازیابی اطلاعات مناسب در گنج با توجه به اطلاعات اصطلاحنامه‌های تخصصی در ایرانداک و نمایش نتایج جستجو با استفاده از رویه‌های مصورسازی اطلاعات

▪ پردازش متن و تبدیل آن به صوت

در نهایت روبوداک لازم است که خروجی را به کاربر در قالب‌های زیر نمایان سازد:

- صوت
- حرکات بدنه فیزیکی ربات که به نوعی حالت رفتاری ربات را نشان می‌دهد
- تصویر نتایج جستجو به کاربر از طریق یک وسیله فیزیکی

در شکل ۲-۲ سیستم روبوداک براساس ورودی‌ها، پردازش‌ها و خروجی‌های مورد نیاز در مرحله آغازین توسعه، به صورت شماتیک نمایش داده شده است.

برای محقق شدن هر بخش از امکانات مشخص شده در شکل ۲-۲، لازم است که پلتفرم روبوداک قابلیت‌هایی را داشته باشد. در فصل بعدی براساس نیازها و قابلیت‌های مورد انتظار برای روبوداک در این مرحله، حداقل قابلیت‌های فیزیکی مورد نیاز برای روبوداک تشریح می‌گردد.



پردازش‌ها

- ✓ پردازش صوت و گفتار و تبدیل آن به متن
- ✓ پردازش تصویر و تشخیص یک علامت یا حالت مشخص
- ✓ پردازش عبارت جستجو و بکارگیری الگوریتم بازیابی اطلاعات مناسب در گنج با توجه به اطلاعات اصطلاحنامه‌های تخصصی در ایرانداک و نمایش مصوراطلاعات
- ✓ پردازش متن و تبدیل آن به صوت

ماهیت ورودی‌ها

- داده‌های تصویر و ویدئویی
- داده‌های صوتی
- فرامینی که مستقیم از طریق رابط کاربری سیستم روبوداک ارسال می‌شود.

ماهیت خروجی‌ها

- صوت
- حرکات بدنه فیزیکی
- صفحه نمایش نتایج جستجو

شکل ۲-۲- ورودی، پردازش و خروجی‌های روبوداک براساس خدمات مورد نیاز در مرحله آغازین توسعه

فصل سوم:

پلتفرم فیزیکی روبوداک

۳. پلتفرم فیزیکی روبات‌اک

در این فصل ابتدا انواع پلتفرم‌های موجود برای استفاده در زمینه ربات‌های اجتماعی معرفی می‌شوند. سپس با توجه به مجموعه‌ای از نیازها و محدودیت‌ها پلتفرم مناسب برای روبات‌اک از بین موارد بررسی شده، مشخص می‌گردد.

۳-۱. بررسی انواع ربات‌های اجتماعی و قابلیت‌های آن‌ها

از آنجاییکه در این طرح اولویت بر استفاده از پلتفرم‌های آماده ربات‌های اجتماعی انسان‌نماست، بنابراین شناسایی این پلتفرم‌ها و موارد استفاده آنها ضروری است. با شناسایی پلتفرم‌های موجود می‌توان با توجه به خدمات مورد نیاز روبات‌اک و محدودیت‌های طرح، نمونه مناسب را انتخاب نمود. بنابراین در این قسمت تعدادی از ربات‌های اجتماعی و پلتفرم‌های مطرح در پژوهش‌ها در این زمینه و قابلیت‌های آن‌ها معرفی شده‌اند. برای شناسایی این ربات‌ها مقالات حوزه ربات‌های اجتماعی و آزمایشگاه‌های رباتیک (Fridin, 2014; Gorostiza & Salichs, 2011; Malik, Yussof, & Hanapiiah, 2014; Meltzoff, Brooks, Shon, & Rao, 2010; Rodriguez-Lizundia, Marcos, Zalama, Gómez-García-Bermejo, & Gordaliza, 2015; Tzafestas, 2014) مطالعه شده و از بین آنها، مواردی که قابلیت‌های آنها بیش از بقیه به روبات‌اک نزدیک است استخراج شده است. برای هر یک از ربات‌ها مشخصات و قابلیت‌های فیزیکی و پردازشی اعم از ساختار سخت‌افزاری، حرکت، بینایی و قابلیت‌های تشخیص صوت، قابلیت‌های پردازشی داخلی، امکان اتصال و ارتباط از راه دور و کنترل، و انواع سنسورها، براساس مستندات قابل دسترس، معرفی شده است.

۳-۱-۱. ربات Nao

مجموعه ربات‌های Nao، ربات انسان‌نما به طول ۵۵cm هستند که قابلیت‌هایی را نظیر حرکت، تشخیص، شنیدن و حتی صحبت با انسان دارند (Aldebaran, 2015). این ربات‌ها ابتدا در سال ۲۰۰۶ توسط شرکت فرانسوی آلدباران^{۴۲} تولید شدند و به بازار عرضه شدند. نمای ظاهری این ربات در شکل

⁴² Aldebaran

۱-۳ قابل مشاهده است. از این ربات‌ها نسخه‌های متعددی ارائه شده که معمول‌ترین آنها نسخه ربات‌کاپ و نسخه آکادمیک است. نسخه ربات‌کاپ با قابلیت‌های حرکتی پیشرفته معمولاً به عنوان پلتفرمی برای بخشی از مسابقات ربات‌کاپ استفاده می‌شود. نسخه آکادمیک آن، علاوه بر قابلیت‌های حرکتی نسخه ربات‌کاپ دارای قابلیت‌های پردازشی پیشرفته‌تری است. اکثر نسخه‌های این ربات را می‌توان به راحتی از طریق شرکت‌های داخلی و نمایندگان مجاز تامین‌کننده اصلی در کشور تامین کرد.

ساختار سخت‌افزاری:

- ۲۵ درجه آزادی با محرک‌های موتور الکتریکی
- شبکه سنسوری، ۲ دوربین، ۴ میکروفون، محدوده یاب صوتی، ۲ فرستنده و گیرنده مادون قرمز، یک برد اینرسی، ۹ سنسور لمسی و هشت سنسور فشار.
- وسایل ارتباطی مختلف شامل سنتی ساینر صوتی، لامپ‌های led و ۲ بلندگو
- دارای دو CPU، یکی ATOM 1.6 ghz با سیستم عامل لینوکس واقع در سر ربات که براساس برنامه نرم‌افزاری NAOqi^{۴۳} کار می‌کند و دیگری واقع در نیم تنه
- باتری با مشخصات 6.84-watt-hour. این باتری ۱/۵ ساعت کار خودکار را فراهم می‌سازد.



شکل ۱-۳ - نمای ربات NAO

^{۴۳} برنامه نرم‌افزاری که مختص ربات‌های Nao برای کنترل و برنامه‌ریزی آنها توسط شرکت Aldebaran توسعه داده شده است.

قابلیت‌های حرکتی:

- راه رفتن در جهات مختلف
- راه رفتن Nao با استفاده از یک مدل دینامیکی ساده انجام می‌گیرد که با استفاده از سنسور اتصالات پایداری ربات تامین می‌گردد. Nao می‌تواند بر روی انواع سطوح کف، مانند فرش، کاشی و چوبی راه برود.
- این ربات دارای یک ماژول حرکتی بر اساس سینماتیک معکوس و یک ماژول مدیریت افتادن است که مانع آسیب دیدن آن در هنگام افتادن می‌گردد.

بینایی:

- دارای دو دوربین با قابلیت ردیابی، یادگیری، تشخیص تصاویر و صورت است.
- Nao با استفاده از دو دوربین 1220 p با سرعت ۳۰ فریم بر ثانیه عمل دیدن را انجام می‌دهد. دوربین اول که در پیشانی قرار گرفته اسکن افقی انجام داده و دوربین دوم که در سطح دهان ربات قرار گرفته محیط اطراف را با سرعت بالا اسکن می‌کند. نرم افزار جاسازی شده در ربات امکان پردازش تصاویر را فراهم می‌کند. در ربات Nao الگوریتم‌هایی برای آشکارسازی و تشخیص صورت و شکل‌ها تعبیه شده است. همچنین این ربات قادر به تشخیص فردی است که با آن صحبت می‌کند. کیت توسعه نرم‌افزار^{۴۴} ربات Nao این اجازه را به کاربر می‌دهد که با رابط OpenCV^{۴۵} (یک نرم‌افزار منبع-باز برای پردازش تصویر) ارتباط برقرار کند.

قابلیت‌های شنیداری:

- ربات Nao از چهار میکروفن برای دریافت صدا استفاده می‌کند.
- ربات دارای قابلیت تشخیص صدا و قابلیت تبدیل متن به صدا به ۱۹ زبان است.
- با استفاده از یک ماژول NAOqi در ربات Nao کار پردازش صوت برای تشخیص جهت صدا در هر بار شنیدن انجام می‌شود. با استفاده از این ماژول قابلیت‌های زیر در ربات Nao ایجاد می‌شود:

○ تشخیص انسان و ردیابی

⁴⁴ Software Development Kit (SDK)

⁴⁵ Open Source Computer Vision library originally developed by Intel

- تشخیص صدا در محیط نویزی
 - تشخیص گفتار در یک جهت مشخص
 - تشخیص سخنگو در یک جهت مشخص.
 - قابلیت مونیتورینگ از راه دور
- به دلیل محدودیت پردازش CPU ربات بهتر است پردازش صوت در یک کامپیوتر و یا سرور صورت پذیرد. ماژول NAOqi از پروتکل SOAP^{۴۶} برای ارسال و دریافت صدا بر روی اینترنت استفاده می کند. این ماژول محیط آزمایشگاهی مناسبی را برای پردازش صوت فراهم می سازد.
- سنسورهای لامسه و محدوده یاب صوتی:
- ربات Nao علاوه بر دوربین ها و میکرفن ها از سنسورهای دیگری نیز استفاده می نماید.
 - بر روی بخش انتهایی هر دست ربات ۳ سنسور لامسه و فشار قرار دارد.
 - این ربات از LED های نورانی برای تعیین نوع تماس، و ۲ کانال صوتی، ۲ کانال برای ارسال و ۲ کانال برای دریافت استفاده می نماید. این سنسورها برای تشخیص فاصله از موانع و محیط بکار گرفته می شوند و محدوده تشخیص از 1cm تا 3m است.
- اتصالات:
- ربات NAO از Wi-Fi و Ethenrt پشتیبانی می کند. علاوه بر آن در این ربات از ۲ سنسور مادون قرمز برای ارتباط با اشیا که در محل چشم های ربات قرار دارند استفاده شده است.
 - این ربات از استاندارد بی سیم IEEE 802.11b/g/n استفاده می کند و می تواند در شبکه- های WPA و WEP نیز استفاده شود.
 - با استفاده از تکنولوژی XMPP می توان از راه دور با NAO ارتباط برقرار نموده و تصاویری از دوربین های آن را دریافت کرد.
- بطور نمونه کارهای که به کمک ربات Nao انجام شده به شرح زیر است:
- بر اساس آدرس IP می توان موقعیت ربات را تشخیص داده و اطلاعات هواشناسی آن نقطه را دریافت نمود.
 - می توان از ربات Nao برای اتصال آن به ویکیپدیا و دریافت اطلاعات استفاده نمود.

⁴⁶ Simple Object Access Protocol

□ می‌توان ربات Nao را به یک ایستگاه رادیویی اینترنتی متصل نموده و اخبار آنرا دریافت نمود.

۲-۱-۳. ربات DARwIn-OP

ربات انسان‌نمای فوق پیشرفته DARwIn-OP محصول شرکت روباتیس^{۴۷} با سرعت پردازش بالا، سنسورهای متنوع، سروموتورهای قدرتی و قابلیت حرکات دینامیک، رباتی بسیار مناسب برای انجام فعالیت‌های آموزشی و تحقیقاتی است (Robotis 2015). از این ربات در لیگ ربات‌های انسان‌نما استفاده می‌شود. همچنین پلتفرم نرم‌افزاری و سخت‌افزاری باز این ربات^{۴۸} و قابلیت تعمیر و نگهداری آن توسط خود کاربر از برتری‌های آن نسبت به محدود ربات‌های انسان‌نمای پیشرفته موجود در بازار جهان است. نمای این ربات در شکل ۲-۳ آمده است. در ادامه برخی از قابلیت‌های این ربات تشریح می‌شوند.



شکل ۲-۳- نمای ربات DARwIn-OP

ساختار سخت‌افزاری:

- وزن ۲/۹ کیلوگرم و قد ۴۵/۵ سانتیمتر
- دارای کامپیوتر داخلی به عنوان کنترل‌کننده اصلی با مشخصات پردازشی ۱/۶ GHz Intel Atom Z530 دارای ۴ GB flash SSD
- کنترل‌کننده ثانوی (CM-037) با مشخصات STMicroelectronics 32F103RE ARM Cortex 32-bit CPU (clocked @ 72MHz)

⁴⁷ ROBOTIS

⁴⁸ Open Platform

- دارای ۲۰ درجه آزادی (۶ درجه در هر پا، ۳ درجه در هر دست، ۲ درجه در گردن)
- یک عدد سروموتور اضافه برای تعمیر و نگهداری یا افزایش درجات آزادی
- چرخ‌دنده‌های فلزی در سروموتور با طول عمر بالا (Dynamixel MX-28T)
- کیت با قابلیت تعمیر و نگهداری توسط خود کاربر به همراه دستورالعمل‌های ساده
- قرار گرفتن در حالت Standby برای پایین آوردن مصرف
- بهره‌گیری از باس داینامیکسل سرعت بالا (۴/۵ Mbps) برای کنترل مفصل
- دارای باتری (برای ۳۰ دقیقه فعالیت)، شارژر و آداپتور برای تغذیه بیرونی
- قابلیت اتصال به وسایل جانبی مختلف
- دارای ژيروسکوپ سه محوره، شتاب‌سنج سه محوره، سه عدد کلید، میکروفن ردیابی (انتخابی)
- باتری و شارژ:

- باتری لیتیم پولیمر LB-21 که توسط شرکت روباتیس برای این ربات ارائه شده و دارای ظرفیت شارژ بیشتری از باتری اصلی آن است.
- قابل شارژ به کمک شارژر LBC-01 که همان شارژر DARwIn-OP و ^{۴۹}Premium BIOLOID است.
- دارای ماژول محافظت داخلی PCM جهت حفاظت باتری در برابر شارژ بیش از اندازه، خالی شدن شارژ آن و جریان زیاد است.

بینایی:

- دارای دوربین USB زاویه باز با رزولوشن MP۲

قابلیت‌های حرکتی:

- دارای مفصل‌های ماژولار و قابل تغییر
- دارای ۲۰ درجه آزادی (۶ درجه در هر پا، ۳ درجه در هر دست، ۲ درجه در گردن)
- سرعت راه رفتن ۲۴ سانتیمتر در ثانیه و زمان هر گام به میزان ۰/۲۵ ثانیه (قابل تغییر)

^{۴۹} ربات Bioloid یک کیت آموزشی و سرگرمی است که نسخه‌های متعددی دارد و توسط شرکت روباتیس ارائه شده است.

□ زمان بلند شدن از حالت خوابیده به پشت ۳/۹ ثانیه و خوابیده به رو ۲/۸ ثانیه (قابل تغییر)

قابلیت‌های شنیداری:

□ ۲ دارای عدد میکروفن

□ حاوی سنسورهای لامسه و محدوده یاب صوتی

اتصالات:

□ شبکه Wi-Fi 802.11 b/g/n

□ بلندگو ohm 1W۸

□ پورت تصویر HDMI

□ پورت ورودی و خروجی صدا

□ دو عدد پورت USB

□ پورت شبکه Ethernet

□ سوکت miniSD

نرم‌افزار و سخت‌افزار این ربات کاملاً به صورت منبع باز است به طوریکه مجموعه کیت اطلاعاتی DW-EK01 اطلاعات تمامی بوردها و قطعات الکترونیکی استفاده شده در DARwIn-OP از جمله کامپیوتر اصلی داروین با سیستم عامل Ubuntu Linux نصب شده بر روی آن را دارد. این ربات را می‌توان به راحتی از طریق شرکت‌های داخلی و نمایندگان مجاز تامین‌کننده اصلی در کشور تامین کرد.

۳-۱-۳. ربات Pepper

ربات انسان نمای Pepper در گروه ربات‌های شخصی قرار می‌گیرد و می‌تواند به عنوان یکی از اعضای خانواده مورد استفاده قرار گیرد. این ربات توانایی کارهای خانه را ندارد، ولی می‌تواند افراد را شناسایی کند، با آن‌ها صحبت کند، احساسات خاص خود را ابراز کند و اطلاعات دریافتی از اینترنت مانند پیام‌ها و یا پیش‌بینی آب و هوا را به اطلاع دیگران برساند (Aldebaran and SoftBank)

5102). شرکت سافت‌بانک^{۵۰} در توصیف Pepper گفته که این ربات اولین ربات شخصی با توانایی ابراز احساسات خاص خود است. Pepper دارای سنسورها و سیستم هوش مصنوعی مبتنی بر فناوری ابری است که آن را از دیگر ربات‌ها متمایز کرده است. علاوه بر این در مقایسه با دیگر ربات‌های مشابه دارای قیمت پایین‌تری است. هزینه خرید یکی از آن‌ها ۱۶۰۰ دلار است. در شکل ۳-۳ نمایی از این ربات نشان داده شده است. در ادامه برخی قابلیت‌های این ربات توضیح داده می‌شوند.

ساختار سخت‌افزاری:

- وزن ۲۸ کیلوگرم و قد ۱۲۰ سانتیمتر
- باتری لیتیوم با ظرفیت 30.0Ah/795Wh، کارکرد باتری تقریباً ۱۲ ساعت

حرکت:

- سرعت حرکت در حدود ۲ کیلومتر در ساعت و می‌تواند تا ارتفاع ۱,۵ سانتیمتر را بالا رود.
- درجه آزادی: سر با ۲ درجه، شانه چپ و راست ۲ درجه، آرنج با ۲ درجه چرخش، مچ با ۱ درجه، دست با ۵ درجه، انگشتان ۱ درجه، کمر با ۲ درجه، زانو با ۱ درجه و پایه با ۳ درجه.



شکل ۳-۳- نمای ربات Pepper

⁵⁰ SoftBank

بینایی:

□ سنسورهای سر: Mic x 4, RGB camera x 2, 3D sensor x 1, Touch sensor x 3

قابلیت‌های شنیداری:

□ ۴ میکروفون

□ سنسورهای لامسه و محدوده یاب صوتی:

○ سنسور قفسه سینه: Gyro sensor x1

○ سنسور سر: Mic x4, RGB camera x2, 3D sensor x1, Touch sensor x3

○ سنسور دست: Touch sensor x2

○ سنسور پاها: Sonar sensor x2, Laser sensor x6, Bumper sensor x3, Gyro sensor x1

اتصالات:

□ اتصال بی سیم IEEE 802.11 a/b/g/n (2.4GHz/5GHz)

□ اتصال Ethernet x1 (10/100/1000 base T)

منبع باز:

□ پلتفرم این ربات مشابه ربات NAO با سیستم عامل NAOqi است

۳-۱-۴. ربات Romeo

این ربات با همکاری چندین شرکت و آزمایشگاه ایجاد شده که در بین آنها شرکت آلدباران به عنوان تامین کننده اصلی هم دیده می شود. در واقع ربات Romeo را به عنوان برادر بزرگتر Nao می شناسند (Aldebaran and SoftBank 2015). این ربات برای توسعه تحقیقات در زمینه ربات های دستیار به افراد سالخورده و کم توان تهیه شده است. قد این ربات ۱۴۰ سانتیمتر است که با توجه به آن می تواند در را باز کند، از پله ها بالا یا پایین بیاید و حتی اشیائی را از روی میز بردارد. در واقع شرکت آلدباران با توسعه این ربات حوزه تحقیقاتی «ربات های دستیار»^{۵۱} را به صورت ویژه دنبال می کند. قیمت این ربات حدود ۳۳۰،۰۰۰ دلار است. نمای این ربات در شکل ۳-۴ آمده است.

⁵¹Assistant robot



شکل ۳-۴- نمای ربات Romeo

ساختار سخت افزاری:

□ ۱۴۰ سانتی متر قد و وزن ۳۶/۶ کیلوگرم

□ دارای چهار کامپیوتر برای بینایی، شنیدن، حرکت و پردازش های هوش مصنوعی

حرکت:

□ راه رفتن در جهات مختلف، بالا رفتن و پایین آمدن از پله با ۳۷ تا ۴۱ درجه آزادی (در مدل -

های مختلف)

بینایی:

□ دارای ۴ دوربین دوبعدی

□ دارای دوربین عمق یاب پشت چشم ها با رزولوشن تا 320x240 و قدرت ۲۰ فریم در ثانیه

قابلیت های شنیداری و صوتی:

□ یک میکروفن x4 روی سر

□ یک بلندگو x2

سنسورهای لامسه و محدوده یاب صوتی:

□ یک سنسور tactile روی سر

اتصالات:

□ اتصال Ethernet با مشخصات 10/100/1000 base T - 1xRJ45

□ شبکه بی سیم با استاندارد IEEE 802.11 a/b/g/n

علاوه بر موارد فوق، ربات Romeo دارای قابلیت‌های اصلی زیر است، که آن را از سایر ربات‌های اجتماعی متمایز می‌کند:

- قابلیت‌های ادراک انسانی، تعدادی از آنها عبارتند از: تشخیص حضور انسان، آشکارسازی صورت^{۵۲}، توصیف چهره، تحلیل نگاه خیره، شناسایی صورت^{۵۳} ردیابی فرد و صورت فرد، شناسایی حالت انسان، و آشکارسازی حرکت، تشخیص احساسات، تشخیص گفتار، تحلیل گفتار، و تحلیل ریتم گفتار.
- قابلیت ادراک وضعیت خود ربات، شامل: تشخیص وضعیت باتری، دمای بدن، وضعیت پاها، وضعیت تعادل ربات، تشخیص افتادن، و تشخیص برخورد.
- درک اشیاء، شامل: قطعه بندی سه بعدی، خوانش بارکد، تشخیص نزدیکی شیء، تشخیص اشیاء، ردیابی اشیاء، درک معنایی
- درک محیط شامل: تشخیص نشانه‌های ساختگی و طبیعی (لندمارک)، تشخیص تاریکی، تشخیص مکان، ردیابی محل، ردیابی صدا، و درک معنایی محیط
- ادراک محرک، که تعدادی از آنها عبارتند از: تشخیص صدا، تشخیص حرکت، محلی سازی صدا، تشخیص برخورد خارجی، و ناظر و آگاه از تماس

۳-۱-۵. ربات Hubo II

این ربات از سری ربات‌های Hubo است که توسط «موسسه پیشرفته علوم و فناوری کره» (KAIST)^{۵۴} توسعه یافته است. این ربات‌ها از قابلیت‌های پردازشی و حرکتی بالایی برخوردار هستند (HUBO 5102). سری اخیر این ربات‌ها در مسابقه رباتیک DARPA در سال ۲۰۱۵، مقام نخست را کسب کردند. سری Hubo II هم یک ربات انسان‌نماست که قابلیت‌های حرکتی پیشرفته‌تری را نسبت به سری‌های قبلی خود دارد. این ربات رتبه سوم را در بین ربات‌های دوندۀ در دنیا دارد. در حال حاضر قیمت تعیین شده برای این ربات حدود ۴۰۰،۰۰۰ دلار است. نمای این ربات در شکل ۳-۵ نمایش داده شده است. در ادامه برخی قابلیت‌های این ربات توصیف می‌گردند (اطلاعات محدودی در وب درباره ویژگی‌های این ربات در دسترس است).

^{۵۲} face detection

^{۵۳} face recognition

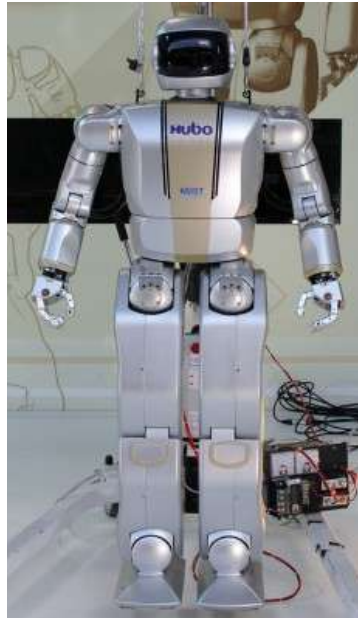
^{۵۴} Korean Advanced Institute for Science and Technology

قابلیت‌های سخت‌افزاری:

- دارای قد ۱۲۵ سانتی متر و وزنی حدود ۴۵ کیلوگرم
- دارای ۴۱ درجه آزادی
- قابلیت راه رفتن با سرعت ۱/۵ کیلومتر بر ساعت و قابلیت دویدن با سرعت ۳/۶ کیلومتر بر ساعت
- دارای باتری ۴۸ ولت با توان ۳۸۰ Wh

بینایی:

- دارای دوربین از نوع CCD در سر
- سنسورهای لامسه و محدوده یاب صوتی:
- کامپیوتر اصلی و کنترل کننده جاسازی شده در قفسه سینه
- استفاده از سنسور سه محوره در هر مچ دست ربات
- استفاده از سنسور اینرسی در بدن
- استفاده از سنسور سه محوره در مچ پا



شکل ۳-۵- نمای ربات Hubo2

۳-۱-۶. ربات‌های سری HOVIS

ربات‌های سری HOVIS^{۵۵} توسط شرکت کره‌ای DTS Robot ساخته شده‌اند و همانطور که از نام آن پیداست، هدف اصلی از طراحی این ربات‌ها تحقق هدف «خدمتگزار خانگی» بوده است (DST Robot 2015). نمای ظاهری این ربات در شکل ۳-۶ آمده است. قیمت این ربات‌های بین ۱۰۰۰ تا ۲۰۰۰ دلار بسته به قابلیت‌های آنها متغیر است. برخی از مهم‌ترین ویژگی‌های ربات‌های HOVIS عبارتند از:

- دارای پایانه تعاملی موبایلی (مبتنی بر اندروید)
- قابلیت کاربردی از مبتدی تا خبره
- راهکارهای اپلیکیشنی آنی و متنوع
- ارتقاپذیری
- طراحی تعاملی و کاربرپسند

ربات‌های HOVIS در سه نسخه Lite، Eco+ و Genie تولید شده‌اند. شکل ۳-۶ نمای این ربات‌ها را نشان می‌دهد.



شکل ۳-۶- نمای ربات‌های سری HOVIS

⁵⁵ Home service

ربات HOVIS Lite یک مجموعه کیت رباتیک پایه است که برای کاربران مبتدی طراحی شده است. این ربات دارای سنسور PSD، سنسور صدا و نور است. همچنین دارای سنسور ژيروسکوپ و شتاب‌سنج سه محوری است. این ربات را می‌توان به وسیله بسته ارتقای HOVIS به نسخه‌های Eco و Genie ارتقا داد.

ربات نسخه Eco+ نسخه ارتقایافته HOVIS Eco با ۲۰ درجه آزادی است و دارای پوشش بدن گونه است که نمایی انسان‌نما تر به آن می‌دهد. این نسخه نیز دارای سنسورهای ژيروسکوپ و شتاب‌سنج بوده و از طریق LED هایی که در سر آن جاسازی شده است، کاربر این امکان را می‌یابد تا طیف وسیعی از احساسات گوناگون در ربات برنامه‌ریزی کند. برخی از ویژگی‌های ربات HOVIS Eco+ عبارتند از:

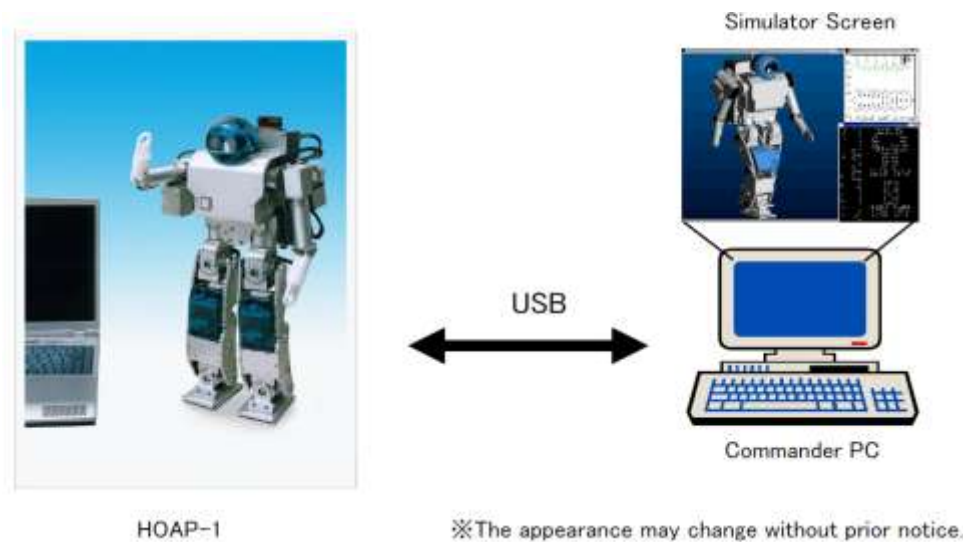
- دارای کیت رباتیک برای هر نوع سطح کاربری (از مبتدی تا خبره)
- ۲۰ درجه آزادی با بدنه انسان‌نما
- دارای LED های با رنگ‌های مختلف، سنسور لمس سر، سنسور ژيروسکوپ و شتاب‌سنج
- دارای برنامه DR-Sim (شبیه‌سازی سه بعدی حرکت)، DR-Visual Logic (ویراستار گرافیکی وظایف ربات).

ربات‌های نسخه Genie پیشرفته‌ترین نسخه در سری ربات‌های HOVIS هستند که کاملاً برای ارائه خدمات خانگی روزمره طراحی شده‌اند. این نسخه با استفاده از دستگاه اینترنت موبایل مبتنی بر اندروید، قادر است بسیاری از خدمات خانگی را به صورت راه دور ارائه دهد. این ربات‌ها قادرند به صورت خودمختار حرکت کنند، هنگام مواجهه با موانع توقف کنند و به طرق مختلف با انسان‌ها تعامل داشته باشند. ربات‌های Genie دارای قابلیت بازشناسی گفتار هستند و می‌توانند به بسیاری از دستورات صوتی و گفتار احساسی پاسخ دهند.

به عنوان مثال، ربات‌های Genie می‌توانند افراد را با زنگ از خواب بیدار کنند، اطلاعات مربوط به آب و هوا و دما را در اختیار قرار دهند و برنامه روزانه شخص را گزارش کنند. همچنین به عنوان یکی از کاربردهای از راه دور، کاربری telepresence است که می‌توان با نصب اپلیکیشن کنترل از راه، فضای خانه را از راه دور مشاهده کرد و دستوراتی را برای انجام به ربات ارسال کرد.

۳-۱-۷. ربات HOAP

ربات انسان‌نمای HOAP محصول شرکت ژاپنی فوجیتسو^{۵۶} است که در سال ۲۰۰۴ نهمین جایزه نوآوری‌های فناوریانه را از آن خود کرد (Robotics Today 2015). نسخه‌های مختلفی از این ربات تولید شده که از بین آنها می‌توان به HOAP-1 و HOAP-3 اشاره کرد. نسخه HOAP-1 یک ربات انسان‌نمای مینیاتوری است که به کامپیوتر متصل می‌شود به‌عنوان ابزاری آزمایشگاهی در اختیار محقق قرار می‌گیرد تا پژوهشگران بتوانند قابلیت‌های کنترل حرکتی و نحوه‌ی تبادل اطلاعات با انسان را مطالعه کنند. این ربات همچنین دارای رابط نرم‌افزاری به‌منظور سهولت ایجاد و توسعه برنامه‌ها به‌صورت امن و کارآمد است. نمای ربات‌های HOAP-1 و HOAP-3 در شکل ۳-۷ و شکل ۳-۸ نشان داده شده است.



شکل ۳-۷- نمای ربات HOAP-1

ربات انسان‌نمای HOAP-3 نسخه‌ی تکامل‌یافته‌تر از سری ربات‌های HOAP است که به برخی از ویژگی‌های آن در زیر اشاره شده است:

- دارای قد ۶۰ سانتی متر و وزن ۹ کیلوگرم
- دارای محورهای متحرک برای سر و دست‌ها
- دارای ۲۸ درجه آزادی
- دارای دوربین CCD با قابلیت ۳۰ فریم در ثانیه
- دارای میکروفن، بلندگو و چراغ‌های LED برای نشان‌دادن احساسات

⁵⁶ FUJITSO

- سنسور اندازه‌گیری مسافت و سنسور نیروی فشار دست
- دارای برنامه‌شیه‌ساز که نحوه کارکرد یک برنامه را قبل از اجرای آن شبیه‌سازی می‌کند
- دارای سیستم‌های بازشناسی گفتار، سنتز گفتار، بازشناسی تصویر، کنترل حرکت به‌صورت بی‌سیم
- سیستم عامل RT-Linux؛ قابلیت برنامه‌نویسی و توسعه متن‌باز به زبان‌های C/C++
- استفاده از رابط USB برای شبکه LAN تغییر یا افزودن اکچویتر و سنسورهای جدید را ممکن می‌سازد.
- کنترل باسیم و بی‌سیم



شکل ۳-۸- نمای ربات HOAP-3

۳-۱-۸. ربات Maggie

ربات Maggie یک ربات اجتماعی شخصی است که توسط آزمایشگاه رباتیک دانشگاه کرلوس III مادرید در اسپانیا ساخته شده و نمونه اولیه آن در سال ۲۰۰۵ معرفی شد (Salichs, et al. 2006). این ربات امکان جمع‌آوری اطلاعات از اینترنت و ارائه گزارش‌هایی مانند گزارش وضع هوا، اخبار،... و چک کردن ایمیل را دارد. نمای این ربات در شکل ۳-۹ آمده است. ربات Maggie براساس ویژگی‌های زیر طراحی شده است:

- جذابیت ظاهری: طراحی ظاهری آن براساس ساختار یک دختر عروسک مانند انجام شده

- واکنش نشان دادن ظاهری^{۵۷}: نمایش بازخوردهایی به صورت ایما و اشاره، مثلاً حرکت سر، چشم و ابرو برای بیان وضعیت تعامل مثلاً ناراحتی، درک کردن، ...
- چندماژولی: تعامل چند ماژولی در قالب علائم و اجزای صورت، حسگرهای لمسی، ارتباط گفتاری، ...
- متحرک بودن: ربات می‌تواند در مکان‌های مختلف حرکت کند.



شکل ۳-۹- نمای ربات Maggie

ساختار سخت‌افزاری:

- قد ۱۴۰ سانتیمتر
- دارای دو چرخ در پایه ربات و دو چرخ جانبی برای جابجایی آن
- دارای ۱۲ سپر (ضربه‌خور)، ۱۲ سنسورهای نوری مادون قرمز، و ۱۲ سنسور ultrasound
- دارای یک محدوده‌یاب لیزری در بالای پایه اصلی
- کنترل موقعیت با استفاده چهار سروو موتور
- استفاده از یک کنترلر میکرو در پایه ربات وصل برای کنترل موتورهای
- سه کامپیوتر داخلی متصل به هم در داخل ربات:
 - کامپیوتر پایه‌ای داخلی: یک PC با سیستم عامل لینوکس، برای حرکت چرخ‌ها، سنسورهای صوتی، سنسورهای مادون قرمز، سپرها (ضربه‌خور)، سر و پلک‌ها.

⁵⁷ expressiveness

○ کامپیوتر Audiovisual Expression: یک کامپیوتر تبلت با سیستم عامل ویندوز.

امکان بازخورد از طریق ارتباط کلامی و تصویری

○ کامپیوتر اصلی: یک لپ تاب با سیستم عامل لینوکس برای پردازش بینایی و کنترل

مهارت‌های ربات.

حرکت:

□ دو درجه آزادی در سر برای چرخش به سمت راست و چپ و چرخش به سمت بالا و پایین

□ حرکت پلک‌ها

□ دو بازو (بدون انگشت و انتهای دست) هر یک دارای یک درجه آزادی

بینایی:

□ روی بدنه یک دوربین رنگی وجود دارد که برای ردیابی افراد استفاده می‌شود. با استفاده از

دوربین عمقیاب کینکت سیستم بینایی ربات ویژگی‌های بیشتری را پیدا کرده است. از این

طریق می‌توان یک درک سه‌بعدی از محیط ربات بدست آورد. بر همین اساس می‌توان

حرکات کاربر و ایما و اشارات وی را به صورت لحظه‌ای تشخیص داد.

قابلیت‌های شنیداری و گفتاری:

□ دارای میکروفون بلوتوث و دو بلندگو متصل به لپ تاب اصلی.

□ یک موتور تشخیص گفتار قوی برای صحبت به زبان اسپانیایی.

□ در زمینه تشخیص گفتار و صوت دارای قابلیت زیر است:

○ تبدیل متن به گفتار حسی^{۵۸}: در اینجا از یک چارچوب نرم افزاری خارجی با نام

Loquendo استفاده می‌شود. امکان تحلیل زبان‌های مختلف را دارد و در حال

حاضر زبان اسپانیایی و زبان انگلیسی با لهجه بریتیش و آمریکایی را پشتیبانی می‌-

کند. همچنین امکان ایجاد ایما و اشاره حین صحبت کردن را هم دارد مانند

خندیدن، خمیازه، سوت زدن،....

○ تشخیص خودکار گفتار: سیستم تشخیص گفتار این ربات به صورت مستقل از

گوینده^{۵۹} طراحی شده بنابراین هر شخصی بدون نیاز به آموزش قبلی ربات، می‌-

⁵⁸ Emotional Text To Speech-ETTS

⁵⁹ Speaker-independent

تواند با آن گفتگو کند. بر همین اساس از نرم‌افزار پیشرفته Loquendo استفاده می‌شود. این ربات زبان‌های اسپانیایی، انگلیسی با لهجه آمریکایی و انگلیسی را درک می‌کند.

○ شناسایی گوینده: این ربات می‌تواند هویت شخصی را که با آن در حال گفتگو است با استفاده از یک پایگاه داده صداها که از قبل ذخیره شده تشخیص دهد که در این بخش نیز از قابلیت‌های نرم‌افزار Loquendo استفاده می‌کند.

○ مدیریت گفتگو: سیستم گفتگو یک سیستم خودکار است که قادر به تقلید رفتار یک انسان در یک مکالمه است. بنابراین این قابلیت جریان و روند مکالمه و گفتگو را کنترل می‌کند، به گونه‌ای که ورودی‌های گفتار را دریافت می‌کند، در قبال آنها پیغام تولید می‌کند و به کاربر ارائه می‌دهد. در حال حاضر برای گفتگو از DialogSkill براساس VoiceXML استفاده می‌شود.

سنسورهای لامسه و محدوده یاب صوتی:

□ سنسورهای لمسی: یکی روی هر شانه، یکی بالای سر، دو عدد روی سینه، دو عدد روی نیمه بالایی پشت، و سه عدد روی هر بازو. هر سنسور دارای active zone حدود 5 cm² است و با تماس انسان فعال می‌شود. علاوه بر آن، ربات دارای سنسورهای infrared، RFID، و لیزر برای تشخیص فاصله اشیاء است.

۳-۱-۹. ربات Sacarino

این ربات یک ربات خدمت‌دهنده است که هدف آن کارکردن در هتل‌ها و اطلاع‌رسانی به مهمانان درباره خدمات هتل و همراهی آن‌ها در فضای هتل است. این ربات دارای چهار چرخ و دو موتور برای چرخش و انتقال است. نمای این ربات در شکل ۳-۱۰ نشان داده شده است (Rodriguez, et al. 2015). در ادامه ویژگی‌های این ربات تشریح می‌شوند.



شکل ۳-۱۰- نمای ربات Sacarino

ساختار سخت افزاری:

- دو موتور بدون جاروبک (80W/24V, 6.2N/m). یک موتور برای چرخش و یک موتور برای انتقال
- باتری LiFePo4 26V/40Ah، تولید کننده ۳ تا ۵ ساعت نیرو
- لیزر LMS100 برای مکان یابی و ساخت نقشه
- قسمت های کشویی برای حمل روزنامه و کالاهای دیگر
- دو بازو با دو درجه آزادی
- شکل دهی به دهان از طریق LED برای نشان دادن حالات مختلف و حرکت دهان هنگام صحبت کردن

بینایی:

- دوربین Playstation- 3
- قابلیت تشخیص چهره و دنبال کردن
- قابلیت تنظیم روشنایی
- قابلیت های شنیداری:
- میکروفون برای فیلتر کردن اختلال و تشخیص صدا
- سنسورهای محدوده یاب صوتی: حلقه ای از سنسورهای ۱۶ ultrasonic برای کشف موانع

۳-۱-۱۰. ربات Asimo

پیشرفته ترین ربات انسان نما ربات Asimo می باشد که دارای ارزشی حدود ۲/۵ میلیون دلار است. آخرین نسخه آن در سال ۲۰۱۱ ساخته شده که از آخرین فناوری‌ها در زمینه خودمختاری، قابلیت حفظ تعادل و سیستم تشخیص برخورددار است و توسط شرکت هوندا ارائه شده است (Honda 5102).

Asimo قابلیت حرکت در محیط های بسیار شلوغ مانند، فروشگاه های بزرگ، موزه ها و ایستگاه ها را دارد، می تواند به خوبی انسان از پله ها بالا و پایین برود و با سرعت ۹ km/h بدود. نمای این ربات در شکل ۳-۱۱ آمده است. این ربات مانند سایر ربات های اجتماعی از طریق شبکه بی سیم ارتباط برقرار می کند.



شکل ۳-۱۱- نمای ربات Asimo

ساختار سخت افزاری:

- دارای قد ۱۳۰ cm و وزن ۴۸kg
- ۵۷ درجه آزادی با محرک های موتور الکتریکی
- باتری ۵۱/۸ ولتی با زمان عملکرد ۱ ساعت
- دارای سنسورهای شتاب و جیرو
- استفاده از سنسورهای ۶ محوره در پاها
- قابلیت تشخیص حرکات از پیش برنامه ریزی شده

حرکت:

- راه رفتن در جهات مختلف و روی پله ها

بینایی:

- دارای دوربین های HD با رزولوشن 720x1080

□ قابلیت تشخیص اشیاء متحرک

قابلیت‌های شنیداری:

□ قابلیت پردازش دستورات صوتی به صورت کامل

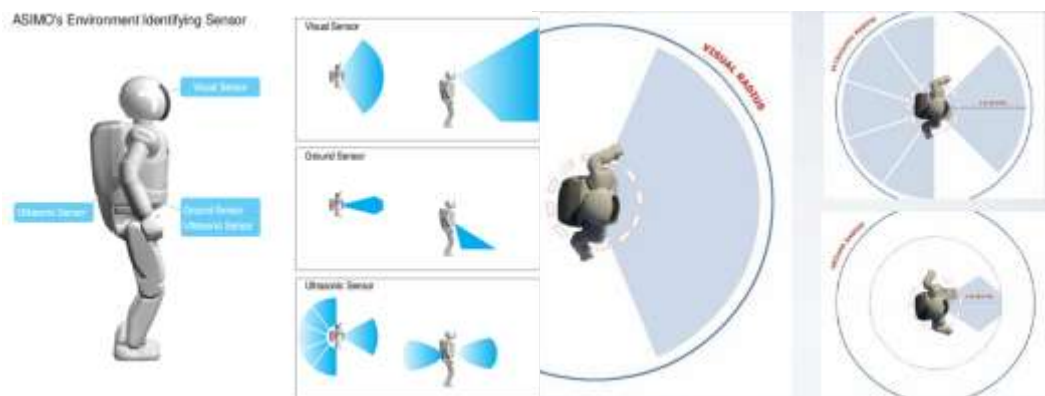
سنسورهای محدوده یاب صوتی:

□ دارای سنسورهای مادون قرمز و لیزری (سنسور لیزر وظیفه تشخیص زمین و هر مانعی به

فاصله ۲ متری را دارد)

□ دارای سنسورهای سونار برای تشخیص موانع جلوی ربات تا ۳ متر (سنسورهای این ربات در

□ شکل ۳-۱۲ نشان داده شده‌اند)



شکل ۳-۱۲- نمای سنسورها و زاویه دید Asimo

۳-۱-۱۱. ربات Rabo

ربات Rabo توسط تیم ربات انسان‌نما در مرکز تحقیقات مکاترونیک دانشگاه آزاد قزوین طراحی شده است. ساختار اصلی این ربات از DARwIn-OP الهام گرفته شده است. ربات Rabo با هدف شرکت در مسابقات لیگ ربات انسان‌نما (بخشی از سری مسابقات ربات‌های فوتبالیست) طراحی شده و قابلیت‌های حرکتی متنوعی دارد. نمای این ربات در شکل ۳-۱۳ نشان داده شده است. در ادامه مشخصات این ربات معرفی می‌شوند.

ساختار سخت‌افزاری:

□ قد ۵۸ سانتیمتر و وزن ۵/۲ کیلوگرم

□ پردازنده Intel Atom Z530

□ دارای شارژر و دو باتری با مشخصات lithium polymer 6800 mAh

□ بدنه آلومینیومی



شکل ۳-۱۳- نمای ربات Rabo

حرکت:

□ قابلیت حرکت در تمام جهات

□ تولید مسیر به صورت آنلاین

□ دارای ۲۰ درجه آزادی، ۶ درجه در هر پا، ۳ درجه در هر دست و ۲ درجه در سر

بینایی:

□ دوربین Logitech Webcam C920

□ قابلیت پردازش تصویر با امکان تشخیص اشیاء از پیش تعریف شده

سنسورها:

□ سنسورهای اینرسی 3x Gyro و 3x accelerometer

ارتباطات:

□ کنترل گر Realtek RTL8111F-CG GbE

□ اتصال بی سیم 802.11 b/g/n

از آنجاییکه این ربات دارای قابلیت‌های پردازش تصویر محدودی است و امکان پردازش صوت را ندارد، به طور کامل نمی‌تواند نیازهای روبوداک را در این مرحله برآورده سازد. براساس مذاکرات

انجام شده، تیم رباتیک این دانشگاه امکان توسعه قابلیت‌ها پردازشی تصویری و قابلیت‌ها پردازش صوتی (به صورت محدود) را دارد. لیکن به دلیل هزینه بالای پیشنهادات این تیم برای توسعه و سفارشی‌سازی Rabo، و زمان زیادی که برای طراحی و توسعه باید صرف شود، عملاً این ربات، گزینه مناسبی برای روبوداک در این مرحله نیست.

۳-۱-۱۲. جمع‌بندی نتایج بررسی انواع پلتفرم‌ها

در این بخش یازده نمونه پلتفرم ربات اجتماعی معرفی شد. تمامی این نمونه‌ها (به جز ربات Rabo)، با توجه به ویژگی‌های ربات‌های اجتماعی که در بخش دوم به آنها اشاره شد، حداقل قابلیت‌های یک ربات اجتماعی را دارند، بنابراین هر یک را می‌توان به عنوان گزینه مناسبی برای پلتفرم روبوداک از حیث اجتماعی بودن در نظر گرفت.

از لحاظ قابلیت‌های حرکتی نیز، کلیه نمونه‌های عنوان شده، حداقل قابلیت‌های حرکتی لازم برای روبوداک (حرکت ربات از یک مکان تا مکان دیگر در قالب راه رفتن یا حرکت با چرخ، و امکان حرکت دادن دست‌ها و سر) را دارند. البته برخی از نمونه پلتفرم‌های مطرح شده دارای قابلیت‌های حرکتی پیشرفته‌ای هستند که فراتر از نیازی است که برای روبوداک تصور شده است. به عنوان مثال ربات Asimo یا سری ربات‌های Hubo برای انجام کارهای فیزیکی سنگین (حتی دویدن) طراحی شده‌اند که فراتر از دامنه کاربردی روبوداک است. در واقع وجود چنین قابلیت‌های حرکتی پیشرفته‌ای برای روبوداک لازم نیست و منجر به افزایش هزینه‌های تامین و نگهداری برای روبوداک خواهد شد. از بین نمونه‌های ارائه شده در این بخش، برخی از آنها نظیر Pepper، Maggie، Sacarino، HOAP-3 و HOVIS ربات‌هایی هستند که از حیث قابلیت‌های اجتماعی بودن و برقراری تعامل با کاربر پیشرفته‌تر از بقیه هستند. به عنوان مثال ربات Pepper امکان ابراز احساسات را دارد، یا ربات Maggie دارای سیستم مدیریت گفتگو و شناسایی گوینده است. در صورتی که امکان تامین و پشتیبانی این ربات‌ها وجود داشته باشد، این نمونه‌ها گزینه‌های مناسبی برای روبوداک به شمار می‌آیند.

ربات‌های Nao و DARwIn دسته ربات‌های هستند که به صورتی تجاری در دسترس هستند و در مسابقات روبوکاپ از برخی از نسخه‌های آنها به صورت پلتفرم استاندارد استفاده می‌شود. به همین دلیل این دسته از ربات‌ها از بلوغ نسبی بیشتری نسبت به سایر ربات‌ها برخوردار هستند. طی سال‌های اخیر نسخه‌های متعددی از این دو ربات به بازار و پژوهشگران ارائه شده است. تیم‌های تحقیقاتی رباتیک در ایران نیز از این دو پلتفرم در قالب پلتفرم‌های آماده استفاده می‌کنند و با انواع چالش‌ها و

امکانات این ربات‌ها آشنایی کامل دارند. به دلیل بکارگیری این دو پلتفرم در مسابقات روباتک، هر دو از قابلیت‌ها حرکتی بالایی (به خصوص در پاها) برخوردار هستند.

آنچه که برای انتخاب پلتفرم مناسب برای روباتک ضروری است، وجود سنسورها و دوربین‌های متنوع برای درک محیط، قابلیت حرکتی قابل قبول، قابلیت‌ها برقراری ارتباط با کاربر از طریق صوت، ایما و اشاره و تکان دادن اجزای فیزیکی و البته قابلیت پردازش اولیه ورودی سنسورهاست.

البته هزینه‌های تامین ربات و پشتیبانی، و نیز امکان تامین ربات در زمان کوتاه نیز جزو معیارهای کلیدی برای انتخاب پلتفرم مناسب به شمار می‌آید، که نقش تعیین‌کننده‌ای را در انتخاب گزینه مناسب ایفا می‌کند. در بخش بعدی معیارهای اصلی برای انتخاب پلتفرم مناسب برای روباتک معرفی می‌شوند و با توجه به این معیارها و قابلیت‌ها ربات‌ها، پلتفرم مناسب برای روباتک شناسایی می‌شود.

۳-۲. اولویت‌بندی خدمات روباتک

در این بخش خدمات اولویت‌دار روباتک برای مرحله آغازین از بین خدمات تعریف، مشخص شوند. با توجه به دامنه طرح پژوهشی، و خروجی‌های مورد انتظار، لازم است که خدمات تعریف شده برای روباتک در بخش قبل، اولویت‌بندی شوند تا مشخص شود که کدامیک از آنها لازم است برای مرحله آغازین توسعه روباتک ایجاد شوند.

برای انتخاب معیارهای مناسب جهت اولویت‌بندی و در نهایت اولویت‌بندی خدمات براساس معیارها، از فرایند تصمیم‌گیری گروهی بر مبنای نظر افراد خبره استفاده شده است. برای این منظور از نظر متخصصین در حوزه‌های زیر بهره گرفته شده است:

- متخصص رباتیک و کنترل (دکتری مهندسی برق و کنترل) - ۱ نفر
- متخصص هوش مصنوعی و بینایی ماشین (دکتری مهندس سیستم گرایش هوش مصنوعی و کارشناسی ارشد سیستم‌های هوشمند) - ۲ نفر
- متخصص زبان‌شناسی رایانشی (دکتری زبان‌شناسی) - ۱ نفر
- متخصص پردازش اطلاعات و پایگاه داده (کارشناسی ارشد مهندسی سیستم) - ۱ نفر

لازم به ذکر است که هر یک از متخصصین فوق دارای حداقل ده سال سابقه پژوهش در زمینه‌های تخصصی مربوط هستند.

فرایند تصمیم‌گیری گروهی بر مبنای روش تصمیم‌گیری مبتنی بر توافق^{۶۰} انجام شده است. این روش در مواردی که نیاز به مشارکت کامل اعضای یک گروه در فرایند تصمیم‌گیری و توافق کامل آنها روی نتیجه وجود دارد، مفید است. در این روش لازم است که اعضای گروه روی حصول به یک هدف مشترک و واحد توافق داشته باشند و در نهایت یک منفعت جمعی را دنبال کنند (Hartnett, 1102).

در همین راستا به دلیل متفاوت بودن تخصص افراد شرکت کننده در گروه، لازم است که نظر همه آنها کاملاً در فرایند تصمیم‌گیری لحاظ شود و در نهایت پس از بحث و بررسی جامع، یک توافق نهایی حاصل شود. بنابراین روش تصمیم‌گیری مبتنی بر توافق می‌تواند در این زمینه راه‌گشا باشد. در همین راستا و با توجه به نتایج حاصل از مباحثه اعضای گروه متخصصین، معیارهای زیر برای اولویت‌بندی خدمات روبوداک در فاز آغازین در نظر گرفته شده‌اند:

□ تطابق با الزامات مرحله آغازین

□ محوریت

□ هزینه

□ پیچیدگی اجرا

منظور از تطابق با الزامات مرحله آغازین، میزان تطابق هر خدمت با قابلیت‌ها و الزاماتی است که باید در مرحله آغازین توسعه روبوداک ایجاد شوند.

محوریت یک خدمت بیانگر میزان کلیدی بودن آن در مجموع قابلیت‌های روبوداک است. برخی از خدمات تعریف شده، محوریت زیادی دارند یعنی وجود آنها لازمه بوجود آمدن سایر خدمات نیز هست. مثلاً خدمت جستجو، یک خدمت کلیدی با محوریت بالا است زیرا برای برآورده ساختن بسیاری از خدمات دیگر، باید ربات قابلیت جستجو را حتماً داشته باشد.

پیچیدگی اجرا به مفهوم اینست که تا چه میزان برآورده ساختن یک خدمت نیازمند انجام فعالیت‌های اجرایی و پژوهشی متنوعی است. مسلماً هرچه پیچیدگی اجرای یک خدمت بالاتر باشد، زمان زیادی نیز برای عملیاتی کردن آن باید صرف شود.

⁶⁰ Consensus decision making

با توجه به محدودیت منابع مالی، هزینه نیز به عنوان یکی از معیارهای کلیدی در فاز آغازین در نظر گرفته شده است.

۳-۲-۱. روش اولویت بندی خدمات

برای اولویت بندی خدمات براساس معیارهای ذکر شده، ابتدا میزان اهمیت هر معیار مشخص شده، سپس برای هر خدمت نسبت به هر معیار براساس نظر متخصصین (با توجه به روش تصمیم گیری مبتنی بر توافق) امتیازی در نظر گرفته شده که عددی بین ۰ تا ۳ است. امتیاز ۰ برای یک خدمت نسبت به یک معیار به منزله اینست که آن خدمت نمی تواند معیار را برآورده سازد و امتیاز ۳ به معنی اینست که خدمت می تواند کاملاً معیار را به بهترین شکل محقق کند. در نهایت امتیاز نهایی هر خدمت براساس میانگین وزنی امتیازهای آن خدمات در همه معیارها است. بنابراین امتیاز هر خدمت (S_i) به صورت زیر محاسبه می شود:

$$S_i = \sum_{j=1}^4 W_j \times R_{ij}$$

که W_j وزن معیار j ام، و R_{ij} امتیاز خدمت i نسبت به معیار j است.

با توجه به اهمیت محقق شدن قابلیت های روباتیک برای مرحله آغازین، وزن معیار «تطابق با الزامات مرحله آغازین» ۱/۵ برابر وزن سایر معیارها در نظر گرفته شده و سایر معیارها وزن مساوی دارند.

در پیوست جدول امتیازهای نهایی خدمات نسبت به معیارها آمده است. امتیازهای نهایی هر خدمات، پس از محاسبه میانگین وزنی امتیاز آن در همه معیارها در جدول ۳-۱ آمده است. بنابراین با توجه به این جدول خدماتی که در اولویت هستند، مشخص شده اند. از آنجاییکه پنج خدمت اول امتیازهای نزدیک تری به هم دارند (بین ۲ و ۲/۳۳) این خدمات می توانند به عنوان خدمات اصلی و اولویت دار برای روباتیک در مرحله آغازین در نظر گرفته شوند در حالیکه امتیاز خدمت ششم ۱/۵ است که با پنج خدمت اول فاصله بیشتری دارد.

جدول ۳-۱- اولویت‌بندی خدمات روبوداک

خدمات	امتیاز نهایی
جستجو در پایگاه وب و گنج	۲/۳۳
توسعه کلیدواژه‌ها براساس اصطلاح‌نامه یا آنتولوژی	۲/۲۲
تحلیل و مصورسازی نتایج جستجو	۲/۰۵
تایپ گزارش	۲
قابلیت دسته‌بندی و سازماندهی متون بازیابی شده	۲
ردیابی حوزه‌ها و موضوعات پژوهشی تازه (ترجیحات پژوهشگر)	۱/۵۵
جستجوی هم‌زمان در موتورهای علمی و پایگاه دانش روبوداک	۱/۵۵
کنترل اصالت گزارش (سرقت علمی)	۱/۴۴
کمک در انتخاب متدولوژی و ابزارهای پژوهش	۱/۴۴
کمک در انتخاب مجله مناسب برای انتشار یافته‌های پژوهش	۱/۴۴
ارزیابی و اعتبارسنجی رکوردهای علمی	۱/۳۳
ذخیره‌سازی رکوردهای بازیابی شده و ترجیحات پژوهشگر به صورت همزمان (داده‌های محیطی)	۱/۲۲
ردیابی مقالات پژوهشگر و اطلاع‌رسانی	۱/۱۱
مدیریت ارجاعات گزارش	۱/۱۱
مصورسازی اطلاعات و ترسیم نمودار	۱/۱۱
مدیریت ارجاعات پژوهشگر	۱
کمک در انتخاب و طراحی شیوه جمع‌آوری اطلاعات	۱
کمک در ویرایش ادبی و علمی متن	۰/۸۸
تحلیل شبکه مولفان و واژگانی پایگاه دانش (ابزارهای علم‌سنجی)	۰/۷۷
ذخیره گزارش به صورت ساخت یافته	۰/۷۲
دریافت و سازماندهی ایده‌های پژوهشگر	۰/۶۶
ارائه جعبه ابزار تحلیل اطلاعات	۰/۶۶
تولید گزارش اولیه به همراه نمودار	۰/۶۶
ذخیره‌سازی معنایی به صورت شبکه‌ای	۰/۴۴
تبدیل قالب مقاله	۰/۳۳
کمک به توسعه پژوهش و بهبود ایده	۰/۲۲

۳-۳. انتخاب پلتفرم مناسب

تقریباً همه مواردی که در بخش قبل به عنوان پلتفرم‌های ربات‌های اجتماعی عنوان شدند، می‌توانند خدمات مشخص شده برای روبات‌اک را در مرحله آغازین برآورده سازند. بنابراین، مهمترین عامل‌ها برای انتخاب پلتفرم مناسب برای روبات‌اک، هزینه و سهولت دسترسی است. علاوه بر آن منبع-باز بودن پلتفرم نیز بسیار اهمیت دارد. اینکه پلتفرم تا چه حد می‌تواند پاسخگوی سایر خدمات در مراحل بعد باشد، نیز معیاری است که می‌تواند در انتخاب گزینه مناسب موثر باشد. پس معیارهای اصلی برای انتخاب پلتفرم مناسب عبارتند از:

- پشتیبانی خدمات در مرحله آغازین
- سهولت دسترسی و پشتیبانی سخت‌افزاری
- هزینه
- قابلیت توسعه و پشتیبانی سایر خدمات
- منبع باز بودن

معیار سهولت دسترسی با توجه به محدودیت زمانی برای این طرح، اهمیت ویژه‌ای دارد. در بین پلتفرم‌های بررسی شده، ربات‌های Nao و DARwIn-OP است که در داخل کشور قابل دسترسی است و سایر ربات‌ها را یا باید از خارج از کشور خریداری نمود یا مانند ربات Rabo قابلیت‌های محدودی دارند. با توجه به بررسی‌های انجام شده، و با وجود موانع تحریم، امکان خریداری ربات‌ها از خارج کشور بسیار چالش‌برانگیز خواهد بود. حتی با فرض تهیه آنها، خدمات پس از فروش نیز معمولاً با توجه به مسائل تحریم، از سوی شرکت تامین‌کننده، به درستی فراهم نمی‌شود.

از طرفی با توجه به محدودیت‌های هزینه‌ای و زمانی، گزینه ربات Rabo (نسخه توسعه یافته آن) نیز قابل تامین نیست. بنابراین انتخاب پلتفرم مناسب از بین ربات‌های Nao و DARwIn-OP خواهد بود. از بین این دو ربات، Nao قابلیت‌های پردازش صوت و تصویر وسیع‌تری را نسبت به DARwIn-OP فراهم می‌کند، در عین حال، سخت‌افزار DARwIn-OP برخلاف ربات Nao کاملاً منبع-باز است.

ربات Nao دارای دو نسخه اصلی روبوکاپ (که برای مسابقات روبوکاپ با پلتفرم استاندارد استفاده می‌شوند) و آکادمیک است. تفاوت اصلی این دو نسخه در پردازشگرهایی که در ربات قرار داده شده (پردازشگرهای داخل ربات در نسخه آکادمیک قویتر است) و مسلماً قیمت (نسخه آکادمیک چهار برابر گرانتر است) آنهاست. با توجه به آنکه بسیاری از پردازش‌هایی که روبات‌اک انجام می‌دهد

می‌تواند در سروری خارج از فیزیک ربات صورت گیرد، بنابراین وجود پردازشگرهای قوی در داخل سخت‌افزار ربات در این فاز از پروژه ضروری نیست.

از طرفی مهمترین مزیتی که ربات DARwIn-OP نسبت به Nao دارد، منبع-باز بودن سخت افزار ربات است به گونه‌ای که نقشه تمام اجزای سخت‌افزاری آن به صورت آزاد در وب سایت تامین کننده موجود است. البته برای این مرحله از پروژه، نیازی به تغییرات سخت‌افزاری در پلتفرم نیست. بنابراین منبع-باز بودن سخت‌افزار الزامی را ایجاد نمی‌کند.

نسخه روبات‌کاپ ربات Nao از نظر هزینه‌ای بسیار به صرفه‌تر از DARwIn-OP است (نصف قیمت) و در عین حال قابلیت‌های پردازشی مورد نیاز روبات‌اک را هم دارد. بنابراین با توجه به آنکه نسخه روبات‌کاپ Nao می‌تواند نیازهای روبات‌اک را برآورده سازد و از نظر هزینه‌ای نیز با صرفه‌تر از موارد دیگر است، این ربات برای پلتفرم روبات‌اک مناسب است.

فصل چهارم:

تشریح قابلیت‌های پلتفرم Nao

و سفارشی‌سازی آن

۴. تشریح قابلیت‌های پلتفرم Nao و سفارشی‌سازی آن

در این بخش ابتدا قابلیت‌های ربات انسان نما تهیه شده به عنوان پلتفرم اولیه روبوداک ارائه می‌شود. رباتی که به عنوان پلتفرم اولیه برای روبوداک در نظر گرفته شده، Nao نام دارد که نسخه روبوکاپ آن در این مرحله تهیه شده است. سپس با توجه با نیازهای روبوداک، راه‌حل مناسب برای سفارشی‌سازی آن ارائه می‌شود.

۴-۱. مشخصات اصلی پلت فرم روبوداک

ربات Nao یک ربات انسان‌نماست که یکی از پرکاربردترین پلتفرم برای پژوهش‌ها در زمینه رباتیک به‌شمار می‌رود. این ربات را شرکت سافت بانک^{۶۱} به بازار ارائه می‌کند که اخیراً با گروه تحقیقاتی آلدباران^{۶۲}، که تولید کننده اسبق این ربات به شمار می‌رفت، ادغام شده است. شمای اصلی نسخه روبوکاپ این ربات در شکل ۴-۱ داده شده است.



شکل ۴-۱- شمای ربات Nao به عنوان پلتفرم روبوداک

در ادامه ویژگی‌های سخت‌افزاری این ربات مورد بررسی قرار می‌گیرد.

• پردازنده و مادربرد^{۶۳}: پردازنده‌ی اصلی این ربات دارای مشخصات زیر است:

- x86 AMD GEODE 500MHz CPU

^{۶۱} SoftBank

^{۶۲} Aldebaran

^{۶۳} Mother board

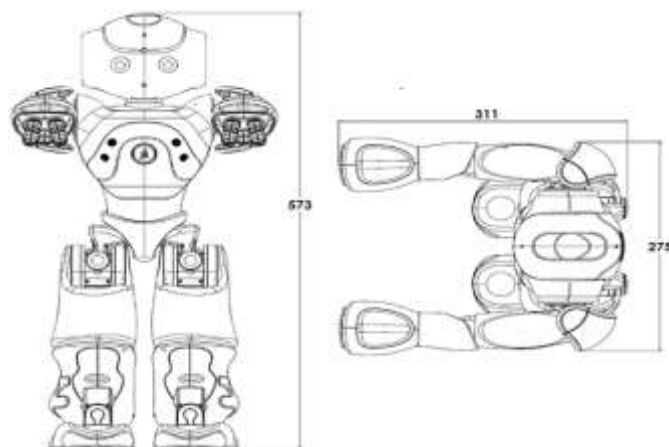
- 256 MB SDRAM / 2 GB flash memory

• مشخصات باتری: ظرفیت و توان باتری در جدول ۴-۱ مشخص شده است.

جدول ۴-۱ مشخصات باتری ربات

نوع باتری	Lithium ion
ولتاژ نامی/ظرفیت جریان	21.6 V / 2.15 Ah
ماکسیم ولتاژ شارژ	24.9 V
جریان شارژ توصیه شده	2 A
حداکثر شارژ / جریان دشارژ	3.0 A / 2.0 A
انرژی	27.6 Wh
مدت زمان شارژ	~ 5h00
مدت نگهداری شارژ	~ 60 min (active use) ~ 90 min (Normal use)

- ابعاد: ابعاد این ربات در بیشترین مقدار خود مطابق جدول ۴-۲ و شکل ۴-۲ آمده است.



شکل ۴-۲- ابعاد ربات از نماهای مختلف

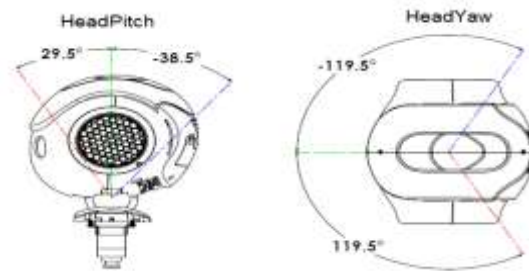
جدول ۴-۲- ابعاد ربات

عرض (mm)	عمق (mm)	طول (mm)
572	113	375

- مفاصل: مفاصل در قسمت‌های مختلف بدن ربات قرار گرفته‌اند. که عبارتند از:

○ مفاصل سر: شکل ۴-۳ نشان‌دهنده میزان چرخش سر و جدول ۴-۳ نشان‌دهنده

اتصالات مختلف سر و ویژگی‌های آن‌ها است.



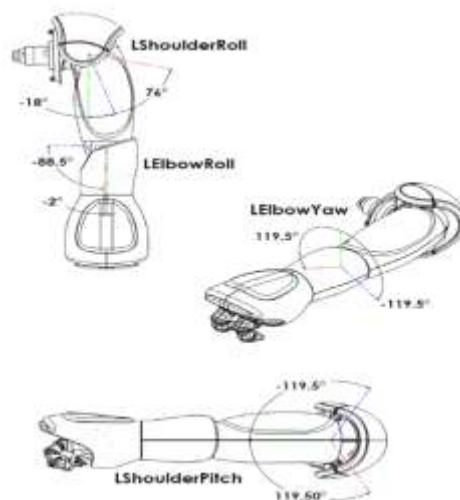
شکل ۴-۳- میزان چرخش سر

جدول ۴-۳- میزان چرخش سر

نام اتصال	نوع حرکت	بازه (بر حسب درجه)	بازه (بر حسب رادیان)
چرخش سر بالا به پایین	سر بالا و پایین می شود روی محور (Z)	-119.5 to 119.5	-2.0857 to 2.0857
چرخش سر حول گردن	سر عقب و جلو می گردد روی محور (Y)	-38.5 to 29.5	-0.6720 to 0.5149

○ مفصل بازو: شکل ۴-۴ درجه آزادی بازو را نشان می دهد. همچنین، نام مفصل و

ویژگی های آنها در جدول ۴-۴ ارائه شده است.

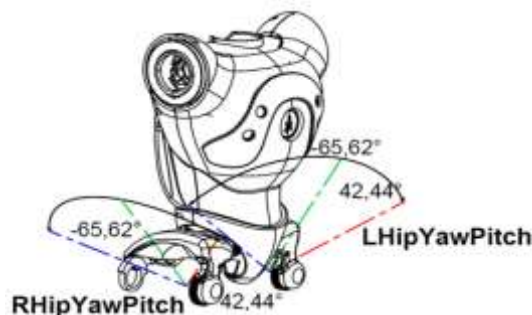


شکل ۴-۴- محدوده‌ی درجه آزادی بازو

جدول ۴-۴- بازه‌ی حرکتی بازو

نام مفصل	نوع حرکت	بازه (بر حسب درجه)	بازه (بر حسب رادیان)
ShoulderPitch	دوران شانه حول محور (Y)	-119.5 to 119.5	-2.0857 to 2.0857
ShoulderRoll	دوران شانه حول محور (Z)	-18 to 76	-0.3142 to 1.3265
ElbowYaw	دوران شانه حول محور (X)	-119.5 to 119.5	-2.0857 to 2.0857
ElbowRoll	دوران آرنج حول محور (Z)	-88.5 to -2	1.5446 to 0.0349

- مفصل لگن: ویژگی‌های مفصل لگن در شکل ۵-۴ و جدول ۵-۴ زیر قابل مشاهده است.

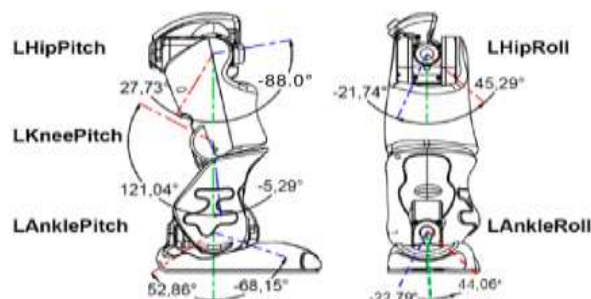


شکل ۵-۴- میزان چرخش لگن و ران پا

جدول ۵-۴- بازه‌ی حرکتی لگن و ران پا

نام مفصل	نوع حرکت	بازه (برحسب درجه)	بازه (برحسب رادیان)
LHipYawPitch	چرخش حول محورهای (Y-Z 45)	-65.62 to 42.44	-1.145303 to 0.740810
RHipYawPitch	چرخش حول محورهای (Y-Z 45)	-65.62 to 42.44	-1.145303 to 0.740810

- مفاصل پا: ویژگی‌های مفاصل پا در شکل ۶-۴ و جدول ۶-۴ قابل مشاهده است.

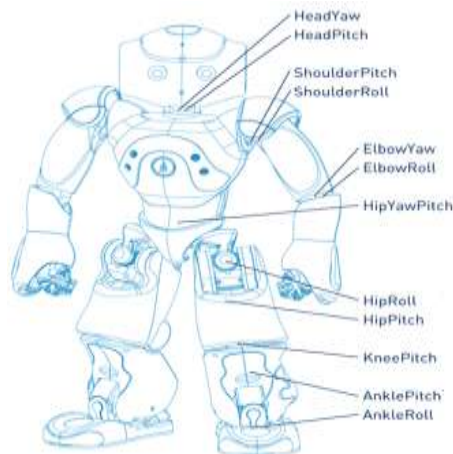


شکل ۶-۴- محدوده‌ی درجه آزادی پا

جدول ۶-۴- بازه‌ی حرکتی پا

نام مفصل	نوع حرکت	بازه (برحسب درجه)	بازه (برحسب رادیان)
HipRoll	چرخش مفصل ران پا به چپ و راست	-21.74 to 45.29	-0.379472 to 0.790477
HipPitch	چرخش مفصل ران پا به جلو و عقب	-88.00 to 27.73	-1.535889 to 0.484090
KneePitch	چرخش مفصل زانو به جلو و عقب	-2.59 to 121.04	-0.092346 to 2.112528
AnklePitch	چرخش مچ پا به جلو و عقب	-68.15 to 52.86	-1.189516 to 0.922747
AnkleRoll	چرخش مچ پا به چپ و راست	-22.79 to 44.06	-0.397880 to 0.769001

- **موتورها:** شکل ۷-۴ نشان دهنده این است که در چه قسمت‌هایی از بدن ربات، موتور به کار گرفته شده است.



شکل ۷-۴- قسمت‌های موتوردار ربات

سه نوع موتور در Nao به کار گرفته است. مشخصات این موتورها در جداول جدول ۷-۴

جدول ۸-۴، جدول ۹-۴، جدول ۱۰-۴ آمده است.

جدول ۷-۴- مشخصات موتور

موتور نوع ۱	موتور نوع ۲	موتور نوع ۳	مدل موتور
22NT82213P	17N88208E	16GT83210E	
8 300 rpm 1±%0	8 400 rpm %21±	10 700 rpm ±10%	سرعت بی‌باری
68 mNm ±8%	9,4 mNm ±8%	14,3 mNm ±8%	گشتاور از حرکت واداشتن
16.1 mNm	4.9 mNm	6.2 mNm	گشتاور نامی

جدول ۸-۴- نوع موتور به کار گرفته در سر

مفاصل	موتور
چرخش سر بالا به پایین	نوع ۳
چرخش سر حول گردن	نوع ۳

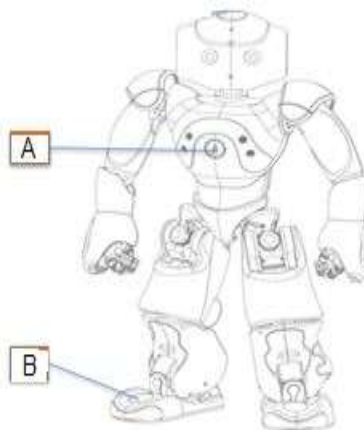
جدول ۹-۴- نوع موتور به کار گرفته در بازو

مفاصل	موتور
چرخش شانه به چپ و راست	نوع ۳
چرخش شانه به بالا و پایین	نوع ۳
چرخش آرنج به چپ و راست	نوع ۳
چرخش شانه به بالا و پایین	نوع ۳

جدول ۴-۱۰- نوع موتور به کاررفته در پای ربات

موتور	مفاصل
نوع ۱	مفصل ران به جلو و عقب
نوع ۱	مفصل ران به چپ و راست
نوع ۱	مفصل زانو به جلو و عقب
نوع ۱	مفصل لگن به جلو و عقب
نوع ۱	مفصل مچ پا به جلو و عقب
نوع ۱	مفصل مچ پا به چپ و راست

○ سنسورهای محیطی: شکل ۴-۸ و جدول ۴-۱۱ نشان‌دهنده محل قرارگیری سنسورهای ربات است.

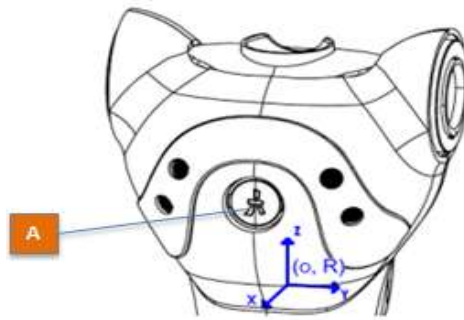


شکل ۴-۸ - محل قرارگیری سنسورها روی ربات

جدول ۴-۱۱- نام سنسورها و محل آن‌ها

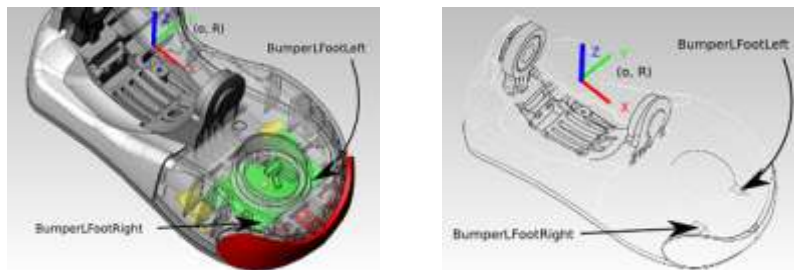
محل	قسمت
دکمه‌ی فشاری روی سینه	A
بامپ‌های روی پا	B

○ دکمه فشاری روی بدن: از این دکمه برای روشن و خاموش کردن ربات و برای ارسال برخی فرمان‌ها استفاده می‌شود و محل آن در شکل ۴-۹ قابل مشاهده است.



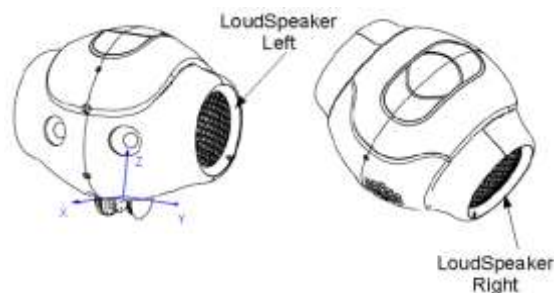
شکل ۹-۴- محل قرارگیری سنسور دکمه‌ی فشاری

- بامپ‌های روی پا: دو سنسور بامپر روی هر کدام از پاها تعبیه شده که در مدل ربوکاپ فقط یکی از سنسورها برای هر پا فعال است. شکل ۴-۱۰ این سنسورها را نشان می‌دهد.



شکل ۴-۱۰- محل قرارگیری سنسور بامپر

- **بلندگوها:** Nao دارای دو بلندگو استریو است که محل آن‌ها در شکل ۲-۱۱ نشان داده شده است. جدول ۴-۱۲ نیز بیان‌کننده محل این بلندگوها است.

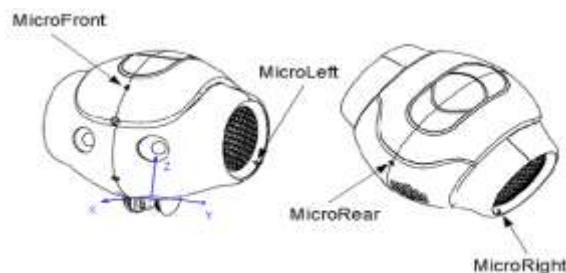


شکل ۴-۱۱- محل قرارگیری بلندگوها

جدول ۴-۱۲- مختصات بلندگوها در ربات

محل قرارگیری	Z(m)	Y(m)	X(m)
چپ	6250.0	3540.0	8300.0
راست	6250.0	-3540.0	8300.0

- **میکروفن‌ها:** در مدل روباتکپ تنها ۲ میکروفن در کنار بلندگو تعبیه شده است که در شکل ۲-۱۲ مشاهده می شود. پهنای باند میکروفن‌ها از ۳۰۰ هرتز تا ۸ کیلوهرتز است که نمی توان از آنها برای دریافت و پردازش صوت استفاده نمود زیرا معمولا محدوده فرکانسی پردازش صوت از ۲۰ هرتز تا ۱۸ کیلوهرتز است. مشخصات محل میکروفون‌ها در جدول ۴-۱۳ ارائه شده است.



شکل ۴-۱۲- محل قرارگیری میکروفن‌ها

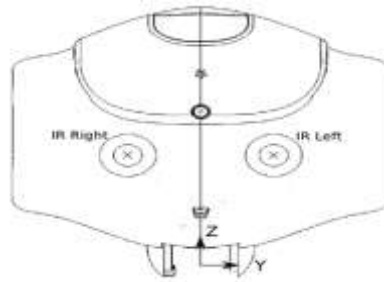
جدول ۴-۱۳- مختصات میکروفن‌ها در ربات

نام میکروفن	Z(m)	Y(m)	X(m)
میکروفن جلو	670.0	0.0	9840.0
میکروفن عقب	4180.0	0.0	-640.0
میکروفن چپ	1330.0	6060.0	-5910.0
میکروفن راست	1330.0	-6060.0	-5910.0

- **سنسور مادون قرمز:** این سنسور برای برقراری ارتباط بین ربات‌ها کاربرد دارد و دارای ویژگی‌های زیر است:

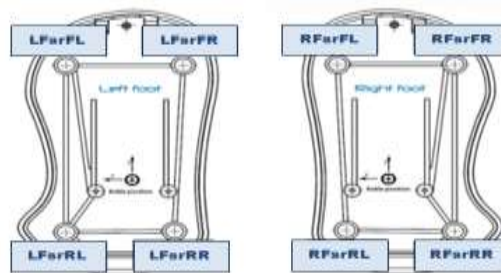
- ✓ طول موج ۹۴۰ نانومتر
- ✓ زاویه تابش مثبت ، منفی ۶۰ درجه
- ✓ توان نامی ۸ میلی وات بر دوره

شکل ۴-۱۳ و شکل ۴-۱۴ جایگاه این سنسور را نشان می دهد.



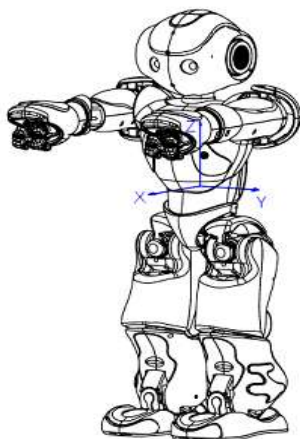
شکل ۴-۱۳- محل قرارگیری سنسور مادون قرمز در سر به عنوان چشم

- **سنسورهای وزن و نیرو:** این سنسورها از نوع مقاومتی هستند. بر اثر فشار ناشی از نیروی وزن میزان مقاومت تغییر کرده و نیرو اندازه گیری می گردد. برای هر پا چهار عدد سنسور تعبیه شده است که از صفر نیوتون تا ۲۵ نیوتون را می توانند اندازه گیری نمایند. شکل ۲-۱۴ محل قرارگیری این سنسورها را نشان می دهد.



شکل ۴-۱۴- محل قرارگیری سنسور وزن و نیرو

- **سنسور شتاب سنج وژیروسکوپ:** این دو سنسور باهم در داخل واحد اجزای داخلی ربات قرار دارند. سنسور ژيروسکوپ دومحوره با دقت ۵ درجه در هر محور و حداکثر سرعت ۵۰۰ درجه در ثانیه یک سنسور شتاب سنج با دقت ۲ گرانش زمین (g) با خطای یک دهم هست. شکل ۴-۱۵ جایگاه این سنسورها را نشان می دهد که محل تقاطع محوره های مختصات نمایش داده شده در شکل است.

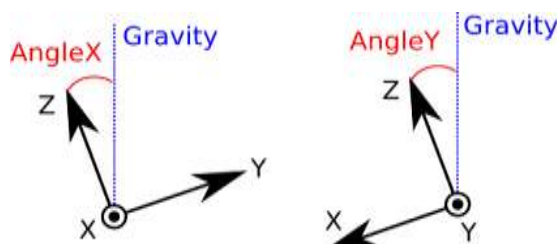


شکل ۴-۱۵- محل قرارگیری ژيروسکوپ و شتابسنج

- داده‌های خروجی سنسور شتاب سنج و ژيروسکوپ: داده‌های خروجی یک تخمینی از سرعت و جهت بدن در مختصات کروی است.

الگوریتم محاسبه زاویه بدن:

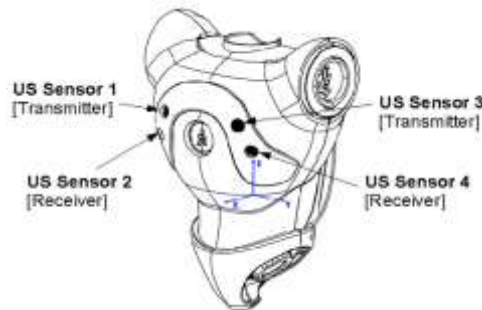
سنسور ژيروسکوپ تغییرات چرخش دور محورهای خود، و سنسور شتابسنج شتاب نسبی را نسبت به شتاب زمین گزارش می‌کند. در حالت استاتیک هنگامی که ربات ثابت ایستاده سنسور ژيروسکوپ ناشی از لرزش‌های ناچیز محیط، خطای بسیار زیادی را به‌عنوان داده گزارش می‌کند و در حالت دینامیکی هنگامی که سیستم در حال حرکت است سنسور شتاب سنج خطای بسیار زیادی را به علت سرعت بالای تغییرات گزارش می‌کند. در این ربات با کمک فیلتر کالمن از نفوذ اطلاعات احتمالاً ناصحیح برای هر سنسور در زمان‌های مختلف جلوگیری شده است. شکل ۴-۱۶ خروجی سنسورها را پس از عبور از فیلتر کالمن نمایش می‌دهد.



شکل ۴-۱۶- داده‌های فیلتر کالمن

- سنسورهای سونار: برای سنسور سونار دو عدد ساطع کننده و دو عدد دریافت کننده در نظر گرفته شده است. فرکانس کاری ۴۰ کیلوهرتز و حساسیت آن‌ها تا ۸۶- دسی بل است. رزولوشن اسمی سنسورها ۱ سانتی‌متر است. رنج اندازه‌گیری از ۲۵ سانتی‌متری شروع و

حداکثر تا ۲/۵۵ متری خاتمه می‌یابد. زاویه مؤثر تابش ۶۰ درجه است. جایگاه این سنسورها در شکل ۱۷-۴ نشان داده شده است.



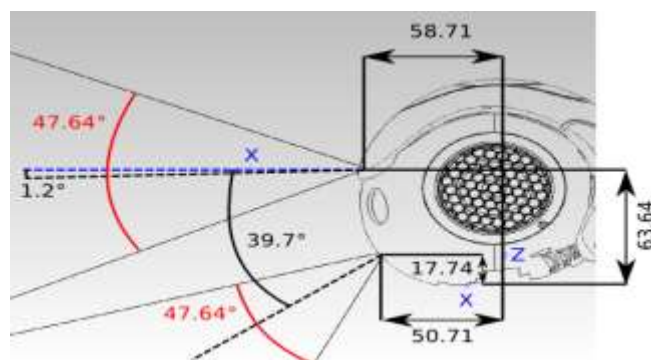
شکل ۱۷-۴- محل قرارگیری سنسورهای سونار

○ **دوربین‌ها:** دو دوربین بر روی سر قرار گرفته‌اند. ابعاد تصویر ۶۰۰ در ۴۰۰ پیکسل بوده و ۳۰ فریم بر ثانیه حداکثر نرخ فیلم‌برداری است. جدول ۱۴-۴ بیان‌کننده ویژگی‌های دوربین است.

جدول ۱۴-۴- مشخصات دوربین سر

مدل دوربین	MT9M114
خروجی دوربین	960p@30fps (YUV422 color space)
میدان دید	72.6°DFOV (60.9°HFOV, 47.6°VFOV)
محدوده‌ی زوم	30cm ~ infinity
نوع زوم	Fixed focus

سنسورهای نوری به‌کاررفته تصویر را با فرمت YUV دریافت می‌کنند سپس تصویر به‌صورت RGB تبدیل می‌گردد. شکل ۱۸-۴ محدوده پوششی دوربین‌ها را نشان می‌دهد.



شکل ۱۸-۴- محدوده پوششی دوربین‌ها

۴-۲. سفارشی سازی ربات براساس قابلیت های مورد انتظار

پلتفرم نانو نمی تواند کلیه الزامات روباتیک را در این مرحله برآورده سازد. بنابراین لازم است که این پلتفرم براساس نیازهای این مرحله سفارشی شود و در صورت لزوم سخت افزارهای دیگری به آن اضافه شود.

۴-۲-۱. الزامات برای دریافت ورودی ها

لازمه دریافت ورودی ها، وجود ابزارها و سخت افزارهای زیر است:

- دوربین با رزولوشن بالا با قابلیت تصویربرداری به صورت لحظه ای
- میکروفون برای دریافت صوت با قابلیت حذف نویز و صدای محیط
- ورودی از طریق کیبورد یا موس، یا استفاده از صفحه نمایش به صورت LCD یا ویدئو پروژکتور برای دریافت نتایج و فرامین از کاربر از طریق یک رابط کاربری گرافیکی.

برای دریافت دستورات غیر صوتی یا تصویری، موس و کیبورد وایرلس در نظر گرفته شده که کاربر می تواند نتیجه ورود اطلاعات را نیز در یک صفحه نمایش خروجی که از طریق ویدئو پروژکتور جاسازی شده در ربات، نمایش داده می شود، مشاهده نماید.

۴-۲-۲. الزامات برای نمایش خروجی ها

روباتیک پس از تعامل با فرد، و دریافت درخواست وی، لازم است که نتایجی نظیر نتایج جستجو و بازیابی اطلاعات را برای کاربر نمایش دهد. برای این منظور می توان از یک صفحه نمایش یا یک دیتا پروژکتور استفاده نمود. چنین امکانات نمایشی در پلتفرم نانو وجود ندارد. بنابراین سه راه حل اولیه برای این منظور وجود دارد:

۱. استفاده از یک نمایشگر جانبی مانند پروژکتور خارجی، صفحه نمایش کامپیوتر یا لپ تاپ

که خارج از بدنه ربات قرار دارد.

۲. استفاده از یک نمایشگر کوچک متصل به ربات

۳. استفاده از یک پروژکتور کوچک که در بدنه ربات جاسازی شده است

استفاده از راه حل های ۲ و ۳ چالش بیشتری را به دنبال خواهد داشت، لیکن یکپارچگی بدنه فیزیکی روباتیک با استفاده از این راه حل ها بیشتر حفظ می شود. از این بین، راه حل سوم یعنی جاسازی یک دیتا پروژکتور کوچک در بدنه ربات بهتر است، زیرا استفاده از یک نمایشگر، هر چند کوچک، به

دلیل ابعاد آن، می‌تواند تعادل ربات را تحت تاثیر زیادی قرار دهد. بنابراین راه‌حل سوم انتخاب می‌شود که در بخش‌های بعدی جزئیات آن تشریح خواهد شد.

۴-۲-۳. الزامات برای پردازش‌ها

سیستم پردازش نائو، نسخه روباتک، برای پردازش‌های محدود کاربرد دارد، لیکن از آنجاییکه ما نیاز به پردازشگرهایی داریم که سه کارکرد اصلی پردازش تصویر پردازش گفتار و پردازش متن را انجام دهند. از طرفی هر یک از ابزارهای موجود برای انجام این پردازش‌ها تحت سیستم‌عامل‌های خاصی اجرا می‌شوند. بنابراین لازم است که سیستم روباتک پردازنده‌های دیگری را داشته باشد و به صورت ارتباط کلاینت-سرور برای ارسال و دریافت درخواست‌ها با آنها ارتباط داشته باشد. در بخش‌های بعدی تجهیزات سخت‌افزاری مورد استفاده که علاوه بر نائو در نظر گرفته شده معرفی می‌شوند.

۴-۲-۴. سیستم بینایی

ربات Nao مجهز به دو دوربین مشابه در قسمت بالا و پایین صورت است. این دوربین‌ها کیفیتی برابر با ۱/۲۲ مگاپیکسل دارند و می‌توانند با سرعت ۳۰ فریم در ثانیه تصویربرداری کنند. دوربین پایین تنها برای مشاهده پایین پا در زمان بازی فوتبال طراحی شده و بنابراین برای انجام کارهایی مانند تشخیص صورت یا حالت مناسب نیست.

برای استفاده از دوربین بالای ربات، آزمایش‌های فراوانی انجام شد. به علت کیفیت پایین دوربین، کارهایی مانند تشخیص صورت یا حالت با خطای بسیار زیادی همراه هستند و این دوربین عملاً قابل استفاده نیست. به علاوه، سرعت تصویربرداری این دوربین در عمل بسیار پایین‌تر از ۳۰ فریم در ثانیه است. نکته مهم دیگر نیز این است که گستره دید دوربین ربات پایین بوده و تنها قادر به تصویربرداری از اجسام نزدیک و در زاویه روبه‌رو است. در نتیجه، تحقیقات وسیعی برای خرید یک دوربین و اتصال آن به ربات انجام شد. معیارهای زیر برای انتخاب دوربین در نظر گرفته شد:

هزینه: هزینه خرید دوربین باید با بودجه پروژه هم‌خوانی داشته باشد.

- کیفیت: کیفیت باید به گونه‌ای باشد که تشخیص صورت و حالت با دقت بالایی انجام شود.
- ابعاد: ابعاد دوربین باید کوچک باشد تا بتوان آن را در بدنه ربات جاسازی کرد.
- وزن: وزن دوربین باید به گونه‌ای باشد که تعادل ربات را بر هم نزنند.
- شکل: شکل دوربین باید به گونه‌ای باشد که قابل جاسازی در بدنه ربات باشد.

- نحوه ارسال تصاویر (با سیم یا بدون سیم): ارسال بی سیم دارای این مزیت است که نیازی نیست که دستگاه پردازنده در نزدیکی ربات باشد و وجود سیم مشکلی برای تعادل ربات ایجاد نمی کند.

پس از بررسی تعداد زیادی از دوربین های موجود در بازار دوربین لایو کم^{۶۴} از شرکت Creative انتخاب شد که در شکل ۴-۱۹ نمایش داده شده است. مشخصات اصلی آن در جدول ۴-۱۵ در مقابل مشخصات دوربین ربات قابل مشاهده است.



شکل ۴-۱۹- دوربین لایو کم

جدول ۴-۱۵- مقایسه مشخصات اصلی دوربین ربات و لایو کم

مشخصات	نوع دوربین	دوربین ربات	لایو کم
رزولوشن		۱/۲۲ مگاپیکسل	۱۲ مگاپیکسل
زومینگ خودکار		ندارد	دارد
لنز گسترده		ندارد	دارد

۴-۲-۴-۱. مقایسه قابلیت های پردازشی دوربین ربات با دوربین لایو کم

علاوه بر رزولوشیون تصویر که یکی از مهمترین معیارها برای مقایسه کیفیت دوربین هاست، معیارهای دیگری نیز برای مقایسه عملکرد و خروجی دو دوربین استفاده شده است. برای این منظور تصاویر خروجی متعددی از دو دوربین در شرایط یکسان از لحاظ نور و موقعیت، مورد بررسی قرار گرفته است، به گونه ای که برای انجام این آزمایش ها، دوربین لایو کم نزدیک دوربین ربات، روی سر آن، متصل شده (شکل ۴-۲۰) تا تصاویر (تقریباً) یکسان به صورت همزمان از هر دو دوربین ثبت گردد و با هم مقایسه شوند.

⁶⁴ Live Cam



شکل ۴-۲۰- جایگاه دوربین‌ها در هنگام آزمایش کیفیت پردازشی دو دوربین

یکی معیارهای معمول برای مقایسه کیفیت دو دوربین، علاوه بر معیار رزولوشیون، کنتراست^{۶۵} تصاویر است. تصاویری که کنتراست بالاتری دارند، برای کارهای پردازشی مناسب‌تر هستند. زیرا تمیز دادن اشیا از یکدیگر ساده‌تر صورت می‌گیرد. برای نشان دادن کنتراست تصویر می‌توان از هیستوگرام^{۶۶} آن استفاده کرد. تصویری که هیستوگرام آن در برگیرنده رنج وسیع‌تری از شدت^{۶۷} است، کنتراست بالاتری دارد.

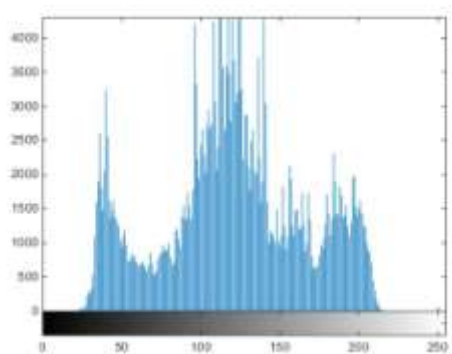
شکل ۴-۲۱ هیستوگرام تصویر گرفته شده از دو دوربین را در کنار یکدیگر نمایش می‌دهد. همانطور که در این تصویر قابل مشاهده است، تصویر دوربین لایو کم کنتراست بالاتری داشته و بنابراین برای کارهای پردازشی مناسب‌تر است.

⁶⁵ Contrast

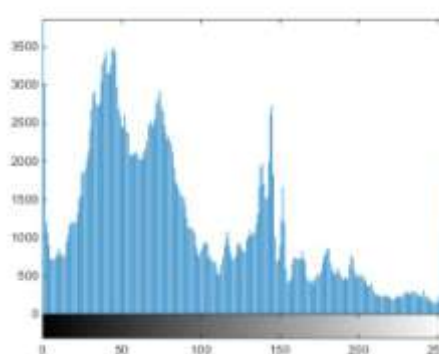
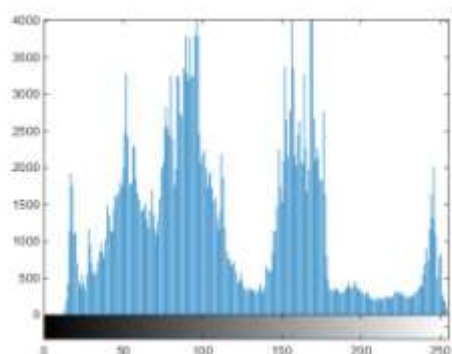
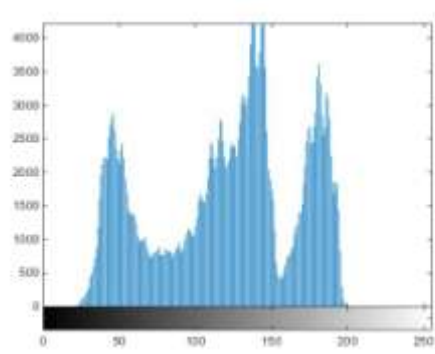
⁶⁶ Histogram

⁶⁷ Intensity

لایو کم



دوربین ربات



شکل ۴-۲۱- هیستوگرام تصاویر گرفته شده از دو دوربین برای بررسی کنتراست

شکل ۴-۲۲ تصاویری را که توسط دوربین ربات و لایو کم گرفته شده، در کنار هم نشان می‌دهد. این تصاویر در شرایط یکسانی گرفته شده‌اند و همانطور که مشاهده می‌شود، تشخیص چشم، که یکی از

ویژگی‌های اصلی در الگوریتم‌های تشخیص صورت است، تنها توسط تصویر دوربین لایو کم ممکن بوده است. به علاوه وضوح تصویر دوربین لایو کم بسیار بالاتر است.

لایو کم



دوربین ربات



شکل ۴-۲۲- تصاویر گرفته شده توسط دو دوربین

برای مقایسه عملکرد دوربین ربات و دوربین لایو کم در تشخیص چهره، مجموعه‌ای از تصاویر گرفته شده توسط دو دوربین در شرایط تقریباً یکسان بررسی شدند و آزمایش تشخیص چهره روی آنها اعمال شد. کلیه این تصاویر حاوی چهره یک فرد است. الگوریتم تشخیص چهره لازم است که بتواند

وجود چهره را در هر تصویر تشخیص دهد. بر اساس این آزمایش، دوربین ربات در ۷۱ درصد موارد می‌تواند وجود چهره را در تصویر تشخیص دهد در حالیکه این عدد برای دوربین لایو کم ۹۰ درصد است.

علاوه بر کیفیت تصاویر و دقت الگوریتم‌های تشخیص چهره، زمان پاسخگویی دوربین نیز اهمیت بسیار در سیستم روبات‌ک دارد، زیرا لازم است که روبات‌ک در زمان نزدیک به زمان واقعی و به صورت لحظه‌ای پردازش تصویر را انجام دهد. برای این منظور نیز زمان پاسخگویی الگوریتم پردازش تصویر روی هر دو دوربین بررسی شد. بر این اساس، زمان پاسخگویی دوربین لایو کم ۰/۰۱ تا ۰/۰۳ ثانیه است، در حالی که این زمان برای دوربین ربات ۰/۵ تا ۰/۷ ثانیه است که عملاً برای کارهایی مانند تشخیص حرکت و تشخیص حالات غیر قابل استفاده است. با اینکه در مشخصات اصلی ربات، ذکر شده که دوربین ربات می‌تواند با سرعت ۳۰ فریم در ثانیه تصویربرداری کند، اما در عمل، به علت طولانی بودن فرایند دریافت تصویر تنها می‌توان حدود یک فریم در ثانیه دریافت نمود. البته استفاده از دوربین ربات به علت اینکه از پیش در بدنه ربات تعبیه شده، ساده‌تر است. لیکن به علت مشکلات ذکر شده، خرید دوربین خارجی غیر قابل اجتناب است. برای جاسازی این دوربین در ربات، محفظه کلاه ماندی طراحی شده که در قسمت آخر این گزارش مورد بررسی قرار می‌گیرد.

۴-۲-۵. میکروفون‌های روبات‌ک

استفاده از میکروفون‌های ربات Nao برای دریافت و پردازش صوت، به دلیل وجود نویز بالا امکانپذیر نیست. زیرا میکروفون‌ها در سر ربات و در کنار فن‌های موجود در سر قرار دارند و عملاً صدای فن، نویز بالایی را در صوت دریافت شده ایجاد می‌کند. برای رفع این چالش سه راه‌حل بررسی شدند:

- تغییر جایگاه قرارگیری میکروفون‌های ربات: ابتدا تلاش شد با دسترسی به میکروفون‌های Nao و تغییر نوع، تغییر مکان قرارگیری میکروفون‌ها و افزایش کیفیت آنها این مشکل حل شود. با گشودن سر Nao و محفظه روکش ربات میکروفون‌های ربات در دسترس خواهند بود، و می‌توان آنها را کمی از عامل نویز (فن ربات) دور نمود. این آزمایش صورت گرفت و ابتدا میکروفون‌ها از محل نویز دور شدند ولی همچنان صوت دریافتی از کیفیت کافی برخوردار نبود. البته می‌توان نویز را تا حدی با تطبیق امپدانس^{۶۸} حذف نمود ولی به دلیل کوچکی دیافراگم خازن و نوع کیفی میکروفون‌ها سیگنال صوتی دریافت شده همچنان نویز

⁶⁸ Impedance matching

بالا و کیفیت پایینی را دارد که برای سیستم‌های پردازش صوت و تبدیل صوت به گفتار مناسب نیست.

- جایگزینی میکروفون‌های ربات با میکروفون‌های مناسب‌تر: در مرحله بعد تصمیم به تعویض میکروفون ربات گرفته شد تا کیفیت مطلوب‌تر گردد. برای این منظور از یک میکروفون خازنی استفاده گردید، اما همچنان نتیجه نامطلوب بود. دلیل آن ماهیت سیستم این نوع میکروفون‌ها و ابعاد دیافراگم آن است که سیگنال صوتی را به خوبی دریافت نمی‌کنند.
- استفاده از یک میکروفون خارجی بی‌سیم، خارج از بدن ربات: در مرحله آخر میکروفون‌های موجود در بازار مورد بررسی قرار گرفتند تا از یک میکروفون خارجی برای ارتباط با ربات استفاده شود.

جدول ۴-۱۶ خلاصه نتایج بررسی میکروفون ربات و نیز انواع موجود در بازار به عنوان میکروفون خارجی را نشان می‌دهد. مهمترین ویژگی‌ها در این جدول، کیفیت بالای صدای دریافتی توسط میکروفون و قابلیت‌های حذف نویز است. بنابراین با توجه به محدودیت‌های بودجه‌ای، میکروفون از نوع هدمیک و ماژول وایرلس آنالوگ انتخاب گردید که به کاربر متصل می‌شود و به صورت بیسیم سیگنال صوتی را برای سیستم ارسال می‌کند. از بین مدل‌های موجود در بازار، در نهایت میکروفون لاجیتک اچ ۸۰۰^{۶۹} انتخاب شد. ویژگی‌های این میکروفون در جدول ۴-۱۷ آمده است.

⁶⁹ Logitech h 800

جدول ۴-۱۶- نتیجه مطالعات درباره نوع میکروفون

نوع میکروفون	محل نصب	مزایا	معایب	چالش ها	قیمت
حرفه ای	روی ربات	کیفیت خوب	برد و زاویه محدود برای جمع آوری صدا	حجم بزرگ و چگونگی جاسازی	بالا
	روی میز و به کامپیوتر	کیفیت خوب	برد و زاویه محدود برای جمع آوری صدا	حذف صدای محیط	بالا
خازنی و کریستالی	روی ربات	ابعاد کوچک	برد و زاویه محدود برای جمع آوری صدا- نویز فراوان	کریستالی ناموجود است	بسیار کم
گان(فیلم برداری تلویزیونی)	روی میز و به کامپیوتر	برد فوق العاده زاویه جمع آوری صدا مناسب	حجم بسیار بزرگ	ممکن نبودن باز کردن میکروفون و دسترسی به قسمت اصلی	خیلی بالا
وایرلس یقه‌ای	روی ربات	استفاده در هر فاصله از ربات در عین کیفیت عالی	-	برقراری ارتباط با ربات و حذف صدای محیط	متوسط تا زیاد
	لپ تاپ	استفاده در هر فاصله از ربات	خارج از بدنه ربات	حذف صدای محیط	متوسط تا زیاد
هدمیک و ماژول وایرلس آنالوگ	لپ تاپ	دارای برد عالی در حالت بیسیم و در نوع دیافراگمی دارای زاویه مناسب، مقاوم در برابر نویز	اتصال به کاربر و خارج از بدنه ربات	-	متوسط تا زیاد

جدول ۴-۱۷- مشخصات میکروفون لاجیتک اچ ۸۰۰

حساسیت	۹۷ دسی بل
پاسخ فرکانس	۱۰۰ تا ۶۵۰۰ هرتز
نحوه اتصال	وایرلس- بلوتوث- با سیم
شعاع پاسخ گویی بیسیم	۱۲ متر

۴-۲-۶. پلتفرم روباتیک برای پردازش‌ها

سیستم‌های پردازش متون و گفتار نیازمند پردازنده قوی با قدرت پردازشی بالا هستند. بنابراین با توجه به قابلیت‌های پردازشی محدود ربات Nao که برای پلتفرم روباتیک در نظر گرفته شده، لازم است اینگونه پردازش‌ها در یک پردازنده مجزا که می‌تواند به صورت بیسیم با ربات ارتباط داشته باشد، صورت بگیرد. برای این منظور از یک لپ‌تاپ مدل ASUS K45VD-D با قابلیت پردازشی که در جدول ۴-۱۸ آمده، استفاده شده است.

جدول ۴-۱۸- مشخصات پردازش لپ‌تاپ

مشخصات	ابعاد	مشخصات
فیزیکی	وزن	۲,۴ کیلوگرم
پردازنده مرکزی	سازنده پردازنده	Intel
	سری پردازنده	Core i7
	مدل پردازنده	QM 2.3GHz ۳۶۱۰
	فرکانس پردازنده	۳,۳ GHz
	حافظه Cache	۶ مگابایت
حافظه RAM	ظرفیت حافظه	۶ گیگابایت
	نوع حافظه	DDR3
حافظه داخلی	ظرفیت حافظه داخلی	۷۵۰ گیگابایت
	مشخصات حافظه داخلی	RPM ۵۴۰۰
پردازنده گرافیکی	شرکت سازنده پردازنده گرافیکی	NVIDIA
	مدل پردازنده گرافیکی	GeForce 610M DDR3 VRAM
	حافظه اختصاصی پردازنده گرافیکی	GB ۲
	کل حافظه گرافیکی	MB ۳۰۷۲
صفحه نمایش	اندازه صفحه نمایش	۱۴ اینچ
	نوع صفحه نمایش	TFT LED-backlit LCD
	دقت صفحه نمایش	۱۳۶۶ x ۷۶۸
	توضیحات صفحه نمایش	HD Glare LED Backlight
امکانات	درایو نوری	DVD-RW
	توضیحات درایو نوری	Super-Multi DVD
	وبکم	دارد
	توضیحات وبکم	۰,۳ مگاپیکسل
	مشخصات اسپیکر	اسپیکرهای داخلی استریو از نوع Altec Lansing
	کارت خوان	دارد
	شبکه بی‌سیم	دارد
	توضیحات شبکه بی‌سیم	Integrated 802.11 b/g/n
	بلوتوث	بلوتوث داخلی
	پورت شبکه (Ethernet)	دارد

۱۰۰۰/۱۰۰/۱۰	توضیحات پورت شبکه	
دارد	پورت VGA	
دارد	پورت HDMI	
۲	تعداد پورت USB 3.0	
۱	تعداد پورت USB 2.0	

علاوه بر لپ تاپ، از یک پردازنده کوچک با عنوان رسپری پای^{۷۰} (نسخه ۳) برای پردازش صوت و تصویر استفاده شده است. این پردازنده مستقیماً به دوربین و نمایشگر خروجی (ویدئو پروژکتور که در به عنوان نمایشگر خروجی در نظر گرفته شده است) متصل است و روی بدنه ربات قرار می‌گیرد. در حقیقت کلیه پردازش‌های ویدئو و تصویر در این پردازنده انجام می‌شود. در ادامه مشخصات آن تشریح می‌گردد.

۴-۲-۱. پردازنده رسپری

پردازنده رسپری پای، یک بُرد کارآمد برای پردازش‌های جانبی است. این بُرد شامل یک پردازنده‌ی «آرم»^{۷۱} هست. از این بُرد می‌توان به‌عنوان کامپیوتر شخصی، تبلت یا لپ‌تاپ با قیمت ارزان، کاربردهای اینترنتی، مرکز رسانه، رباتیک، اتوماسیون صنعتی/خانگی، خدمات ابری، سرور پرینت، نظارت بر امنیت، دوربین‌های وب، بازی، نقطه‌ی دسترسی بی‌سیم، سنجش و مانیتورینگ محیط زیست (مثلاً ایستگاه آب و هوایی) استفاده کرد.

البته این بُرد به‌صورت «زمان حقیقی»^{۷۲} عمل نمی‌کند یعنی از آن برای پردازش‌های خیلی سریع مانند درایو موتورها نمی‌توان استفاده کرد زیرا پاسخ پردازش را در همان لحظه در اختیار نمی‌گذارد. امکانات این بُرد در جدول ۴-۱۹ ارائه شده است.

جدول ۴-۱۹- مشخصات بُرد رسپری

Broadcom BCM2837	مدل پردازنده (آرم)
Quadcore ARM Cortex-A53 64 Bit	واحد پردازش مرکزی
1.2 GHz	سرعت کلاک
1 GB	حافظه دسترسی تصادفی
400 MHz VideoCore IV®	واحد پردازش گرافیکی

^{۷۰} Raspberry pi 3

^{۷۱} ARM

^{۷۲} Real time

1 x 10 / 100 Ethernet (RJ45 Port)	اتصال به شبکه
802.11n wireless LAN (WiFi) and Bluetooth 4.1	اتصال wireless
4 x USB 2.0	خروجی‌های USB
2 x 20 Pin Header	GPIOها (پایه‌های قابل برنامه‌ریزی)
51-pin MIPI	اتصال دوربین
DSI 15 Pin / HDMI Out / Composite RCA	اتصال صفحه‌نمایش
2.5 A	تغذیه لازم (ظرفیت جریان)
Push/pull Micro SDIO	کارت حافظه
Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT	عملکرد سیستم
Audio Output 3.5mm jack, HDMI	خروجی صوتی

ساختار برد:

نمای برد رسپری در شکل ۲۳-۴ و شکل ۲۴-۴ قابل مشاهده است.

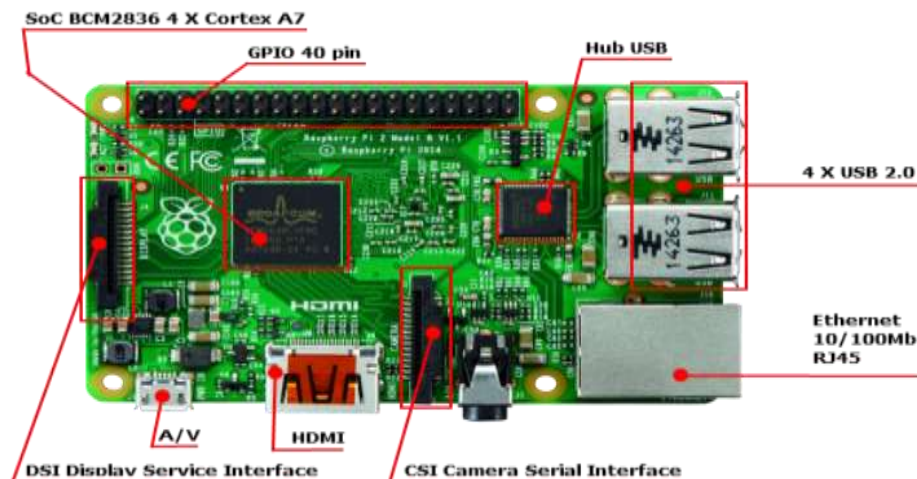


شکل ۲۳-۴- روی صفحه‌ی بُرد رسپری



شکل ۲۴-۴- زیر صفحه‌ی بُرد رسپری

اجزای کلی این برد در شکل ۲۵-۴ نمایش داده شده است.



شکل ۴-۲۵-۱ اجزای کلی برد

در ادامه از بین اجزای نمایش داده شده، فقط بخش‌هایی که در ربات استفاده می‌گردد، شرح داده می‌شود.

الف- سیستم تراشه (Broadcom BCM2837 SoC) در شکل ۴-۲۶:

- مشخصات تراشه:

64 bit

Quad Core

1.2 GHz

ARM Cortex-A53 Processor (ARM V8 ISA)

- واحد پردازش گرافیکی

400 MHz

Videocore IV Multimedia Co-Processor



شکل ۴-۲۶-۲ محل پردازنده‌ی آرم روی برد

ب- ورودی و خروجی‌های جانبی:

در این قسمت انواع ورودی و خروجی رسپری مورد بررسی قرار می‌گیرد.

- ورودی تغذیه‌ی راسپری:
ورودی تغذیه‌ی راسپری یک ورودی micro USB است که می‌توان با یک تطبیق‌دهنده^{۷۳} از برق شهر آن را تغذیه کرد که این تغذیه 5 V و 2.5 A است. در شکل ۴-۲۷ محل این ورودی مشخص شده است.



شکل ۴-۲۷- محل تغذیه‌ی راسپری

- اتصالات صفحه‌نمایش و صدا:
۱- HDMI (شکل ۴-۲۸):
- سیگنال دیجیتال
- سیگنال ویدئو و صدا
- DVI نمی‌تواند سیگنال صوتی عبور دهد.
- رزولوشن 1920 X 1200

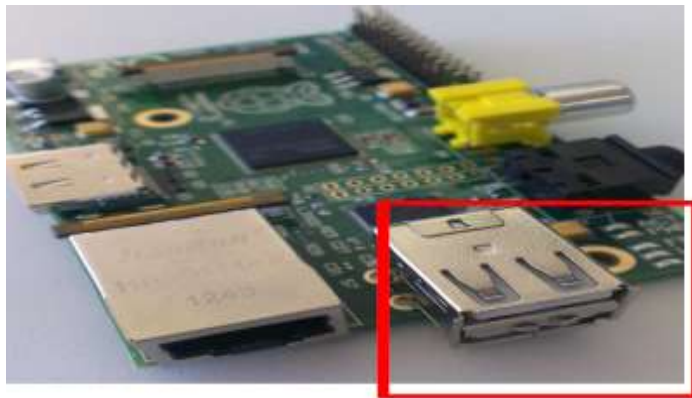


شکل ۴-۲۸- محل قرارگیری HDMI

⁷³ Adaptor

۲-USB:

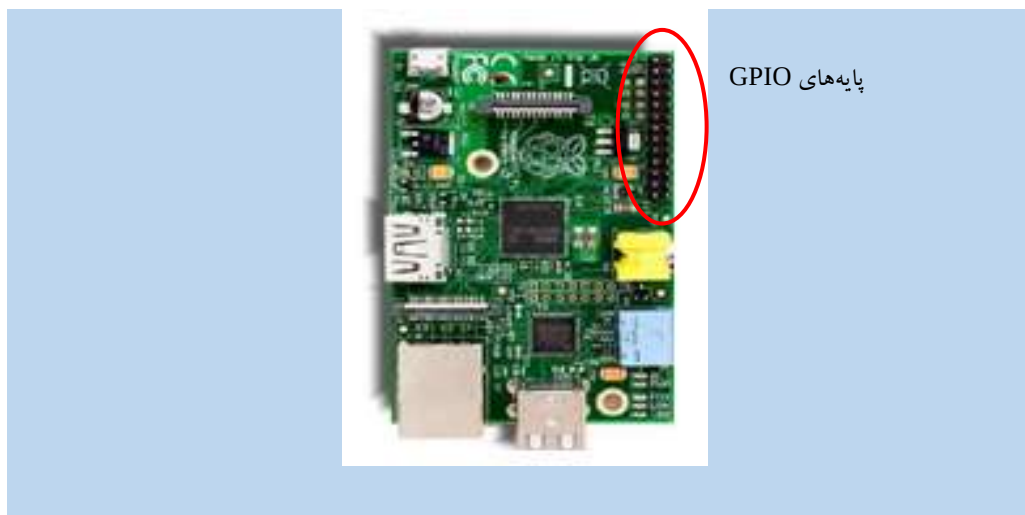
این برد شامل چهار USB 2 هست که علاوه بر آن می توان به وسیله ی USB hub تعداد بیشتری نیز داشت. شکل ۴-۲۹ محل قرارگیری این اتصال را نشان می دهد.



شکل ۴-۲۹-محل قرارگیری USB

- پایه های کنترلی GPIO:

این پایه ها پایه های قابل کنترل می باشند که می توان آن ها را به صورت ورودی یا خروجی پیکربندی کرد. سپس به وسیله ی این پایه ها تجهیزات مختلف را می توان کنترل کرد. مثلاً به وسیله ی آن می توان یک وسیله الکتریکی را کنترل کرد. همچنین این پایه ها برای ارسال اطلاعات به صورت سریال (پایه های USART) کاربرد دارد. علاوه بر آن این پایه ها می توانند از سنسورها اطلاعات بگیرند و پردازش کنند (مثلاً سنسور دما). کلیه پایه ها دیجیتال بوده و برای استفاده از سنسورها، باید از تراشه آنالوگ به دیجیتال استفاده کرد. همچنین از این پایه ها می توان ولتاژ ۵ V ، 3.3 V و زمین نیز گرفت. همچنین دارای پایه های SPI و I2C نیز هست. شکل ۴-۳۰ محل قرارگیری این پایه ها را نشان می دهد.



شکل ۴-۳۰- محل قرارگیری پایه‌های GPIO

برای استفاده از این پایه‌ها ابتدا باید کتابخانه‌های مربوطه را روی راسپری نصب کرد و سپس می‌توان با فراخوانی ماژول مناسب، از آن‌ها در برنامه استفاده کرد.

۴-۲-۷. پلتفرم روباتک برای نمایش خروجی‌ها

روباتک برای تولید خروجی‌های مورد نیاز، لازم است که ابزارها یا وسایل زیر را داشته باشد:

- بلندگو برای خروجی صوت
- درجه آزادی حرکت برای نمایش حرکات فیزیکی
- نمایشگر به صورت LCD یا ویدئو پروژکتور برای نمایش نتایج جستجو به کاربر

برای بلندگو، به راحتی می‌توان از بلندگوی موجود در بدنه ربات Nao استفاده کرد.

ربات Nao مورد استفاده دارای ۲۱ درجه آزادی حرکتی است که برای این میزان درجه آزادی برای طراحی حرکات ارتباطی ساده مانند راه رفتن، نشستن، بالابردن دست یا حرکت دادن سر، مناسب است.

ربات Nao هیچ نمایشگر خارجی ندارد، بنابراین لازم است که یک نمایشگر به صورت جاسازی شده در ربات، و مستقل از لپ تاپی که در قالب پردازشگر با ربات ارتباط دارد، در نظر گرفته شود.

۴-۲-۷-۱. نمایشگر برای روبوداک

برای نمایشگر خارجی روبوداک، می‌توان از صفحات نمایشگر LCD یا از ویدئو پروژکتور استفاده نمود. نمایشگرهای LCD بزرگ که بتوانند اطلاعات را به خوبی نمایش بدهند، وزن بالایی دارند و اتصال آنها به ربات می‌تواند تعادل ربات و حرکات آن را با مشکل مواجه کند. اما ویدئو پروژکتورها در ابعاد متنوع و با وزن‌های متنوع‌تری وجود دارند، و استفاده از آنها برای نمایشگر خارجی، امکانپذیر است. به همین منظور با بررسی‌های انجام شده، و با توجه به محدودیت‌های اندازه، وزن، سازگاری با سیستم‌عامل ربات و سایر سیستم عامل‌ها، از یک ویدئو پروژکتور جیبی با مدل زیر استفاده شده است.

Aiptek MobileCinema i55 iphone 5/5s projector

شکل ۴-۳۱ نمایی از این ویدئو پروژکتور را نشان می‌دهد. مشخصه بارز این ویدئو پروژکتور اینست که می‌تواند بالای سر Nao جاسازی شود و ابعاد آن به گونه‌ای است که وزن ویدئو پروژکتور را در سطح سر پخش می‌نماید.



شکل ۴-۳۱-نمای کلی ویدئو پروژکتور

امکانات ویدئوپروژکتور شامل موارد زیر است:

- ✓ محدوده‌ی تصویراندازی تا دو متر
- ✓ اندازه‌ی صفحه نمایش تا یک و نیم متر
- ✓ رزولوشن 046 X 480 Pixel
- ✓ وزن 160 g

این ویدئوپروژکتور دارای لامپ LED با قدرت ۵۰ لومن نور و دارای باتری با قابلیت شارژ تا 0004 mAh است. همچنین قابل اتصال از طریق HDMI و USB است و با سیستم عامل‌های ویندوز،

آندروید و iOS هم سازگاری دارد. مشخصات کلی این ویدئو پروژکتور در جدول ۴-۲۰ آمده و ابعاد آن نیز در جدول ۴-۲۱ ارائه شده است.

ویدئو پروژکتور به پردازنده رسپری متصل است و از این طریق کنترل می شود.

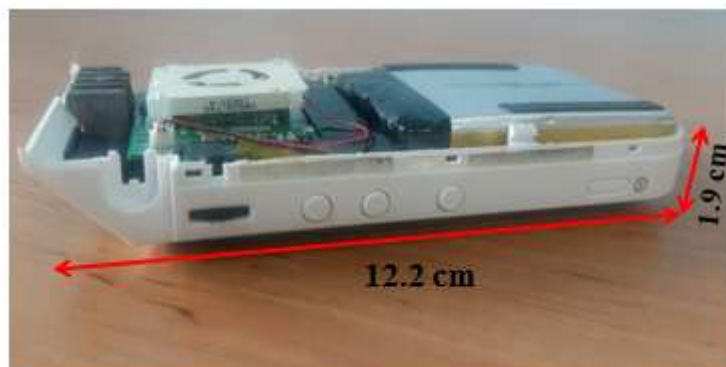
جدول ۴-۲۰- مشخصات کلی ویدئو پروژکتور

FWVGA	نوع سیگنال
DLP	فناوری استفاده شده
۲۰۰۰۰ ساعت	عمر لامپ
۱۲۰ دقیقه	زمان استفاده
ندارد	تعداد بلندگو
ندارد	خروجی بلندگو
USB, HDMI	پورت ها
WVGA	فرمت فایلی که پشتیبانی می کند

جدول ۴-۲۱- ابعاد و وزن ویدئو پروژکتور

عرض	ارتفاع	عمق	وزن
9.31 cm	7.6 cm	3 cm	160 g

به منظور کاهش ابعاد پروژکتور، قسمتی از بدنه که برای نگهداری گوشی موبایل استفاده می شود جداسازی شد. شکل ۴-۳۲ پروژکتور و ابعاد آن را پس از این جداسازی نمایش می دهد.



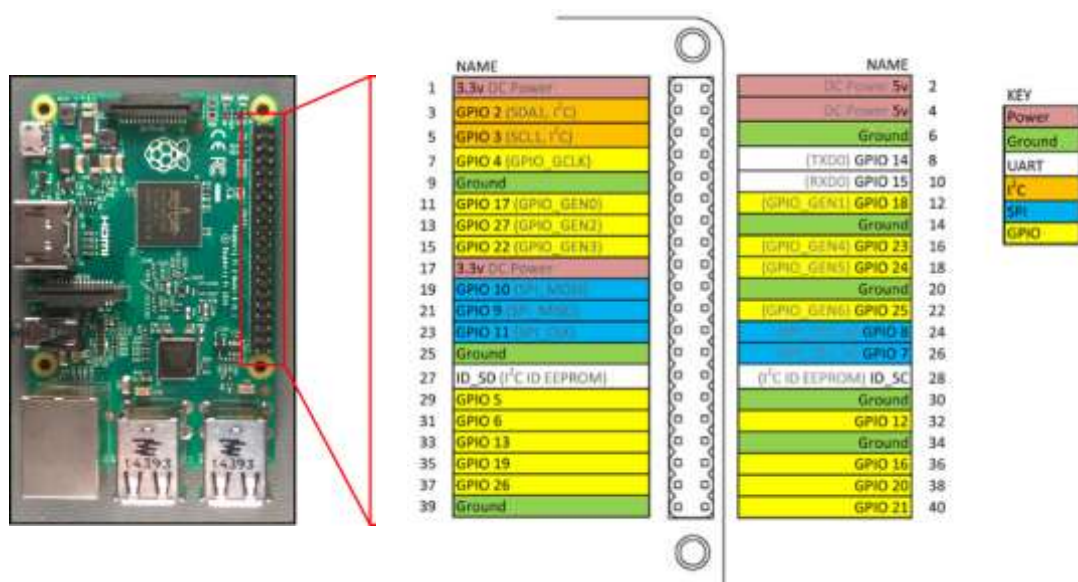
شکل ۴-۳۲ - پروژکتور پس از جداسازی نگهدارنده موبایل

۴-۲-۷-۲. طراحی مدار برای کنترل ویدئو پروژکتور

براساس ویژگی های ویدئو پروژکتور، روشن و خاموش کردن ویدئو پروژکتور باید با استفاده از یک میکروسویچ به صورت دستی انجام شود. اما با توجه به اینکه ویدئو پروژکتور روی سر ربات قرار

دارد نمی‌توان در موقع نیاز و در حین عملکرد ربات با این میکروسویچ آن را روشن و خاموش کرد. بنابراین روشن و خاموش کردن ویدئو پروژکتور باید با استفاده از یک کد به صورت غیر دستی صورت بگیرد. در واقع سیستم روباتک باید فرمان بدهد که چه زمان ویدئو پروژکتور روشن باشد و چه زمانی خاموش شود.

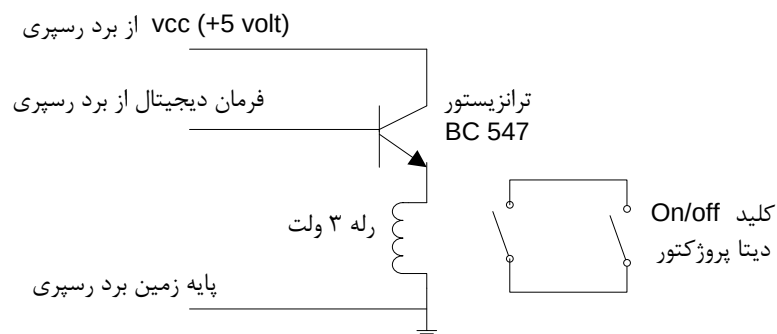
برای روشن و خاموش کردن ویدئو پروژکتور از پایه‌های کنترلی GPIO استفاده می‌شود. زیرا ریسری روی ربات قرار دارد و می‌توان بدون دخالت دست و با استفاده از دستورات کنترلی در قالب کد برنامه‌تویسی، آن را روشن و خاموش کرد. در ادامه کار با این پایه‌های کنترلی توضیح داده می‌شود. پردازنده ریسری دارای پایه‌های کنترلی جی.پی.آی. است که در شکل ۴-۳۳ این پایه‌ها و نام هریک از آن‌ها مشخص شده است.



شکل ۴-۳۳- پایه‌های GPIO در ریسری و نقشه آن

برای استفاده از این پایه‌های کنترلی ابتدا باید کتابخانه‌ی این پایه‌ها را نصب و بعد از آن برای پیکربندی این پایه‌ها در ابتدای برنامه ماژول آن را فراخوانی کرد. سپس تمام پایه‌ها را می‌توان با استفاده از برنامه کنترل کرد.

برای این کار یک کلید قابل کنترل یعنی یک سویچ با یک پایه‌ی کنترلی نیاز است. در اینجا از یک ترانزیستور نوع BC547 و یک رله DC استفاده می‌شود. مدار مربوط به این اتصالات در شکل ۴-۳۴ نمایش داده شده است.



شکل ۴-۳۴- مدار کنترلی استفاده شده برای کنترل ویدئو پروژکتور

همانطور که در این شکل مشاهده می‌شود خروجی دیجیتال برد راسپری به پایه ترانزیستور وارد گردیده و آن را روشن می‌نماید. با روشن شدن ترانزیستور جریان کافی از بوبین رله DC عبور نموده و کانتکت نرمالی باز، آنرا می‌بندد. کانتکت نرمالی باز رله بطور موازی با کلید روشن و خاموش ویدئو پروژکتور در مدار قرار گرفته است. با بسته شدن کانتکت رله ویدئو پروژکتور اگر خاموش باشد، روشن و اگر روشن باشد، خاموش می‌گردد. در نهایت در راسپری کافی است یک کد نرم‌افزاری قرار گیرد که یک پالس حدود ۳ یا ۴ ثانیه‌ای را به پایه‌ی کنترلی راسپری ارسال کند. زیرا برای روشن کردن میکروسویچ باید آن را به صورت دستی به مدت ۳ تا ۴ ثانیه نگه داشت. با دادن این پالس به پایه‌ی کنترلی ترانزیستور، این ترانزیستور روشن شده و به همان مدت دو سر کانتکت رله را به هم متصل می‌کند که باعث روشن شدن ویدئو پروژکتور می‌شود.

با کد نرم‌افزاری زیر می‌توان یک پالس با زمان مورد نظر به پایه‌های کنترلی جی.پی.آی. فرستاد:

```
import RPi.GPIO as GPIO
import time
def main():
    GPIO.cleanup()
    GPIO.setmode(GPIO.BOARD) # to use Raspberry Pi board pin numbers
    GPIO.setup(31, GPIO.OUT) # set up GPIO output channel
    while True:
        GPIO.output(31, GPIO.HIGH) # set RPi board pin 11 low. Turn off LED.
        time.sleep(3)
        GPIO.output(31, GPIO.LOW) # set RPi board pin 11 high. Turn on LED.
        time.sleep(1)
    main()
```

با قرار دادن این برنامه، راسپری به پایه‌ی جی.پی.آی. یک پالس ۳ ثانیه‌ای می‌دهد که باعث وصل ۳ ثانیه‌ای ترانزیستور و رله و روشن شدن ویدئو پروژکتور می‌شود و با اجرای دوباره این برنامه ویدئو

پروژکتور خاموش می‌شود. با اجرای برنامه شکل پالسی که به ویدئو پروژکتور داده می‌شود به صورت شکل ۳۵-۴ است.



شکل ۳۵-۴- شکل موج پالسی که با مدار قبل به ویدئو پروژکتور اعمال می‌شود

همان‌طور که مشاهده می‌شود یک پالس ۳ ثانیه‌ای توسط مدار شکل ۳۴-۴ با برنامه داده شده به رسانی به ویدئو پروژکتور اعمال می‌شود که اجرای برنامه هم باعث خاموش شدن و هم باعث روشن شدن ویدئو پروژکتور از راه دور و به وسیله‌ی کد نرم‌افزاری خواهد شد. نکته‌ی مهم این است که باید حتماً ماژول پایه GPIO را ابتدای برنامه اصلی قرار داد و سپس از این پایه‌های کنترلی استفاده کرد.

۴-۲-۸. بدنه فیزیکی ربات

با توجه به بخش‌های قبل، تجهیزات سخت‌افزاری که برای سفارشی سازی Nao به ربات اضافه شده، عبارتند از:

- دوربین
- ویدئو پروژکتور
- بُرد رسانی
- میکروفون

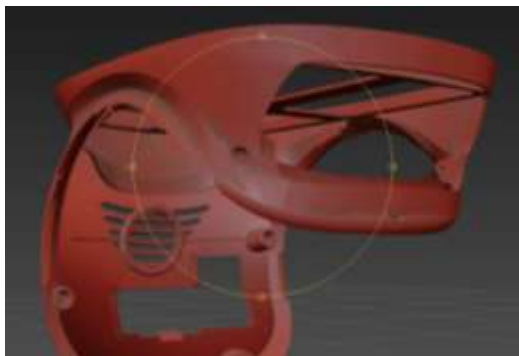
از بین تجهیزات فوق، میکروفون به کاربر متصل است و در بدنه فیزیکی ربات قرار نمی‌گیرد. بنابراین برای جاسازی باقی تجهیزات، لازم است بخش‌های از بدنه فیزیکی ربات مدنظر قرار گیرد که کمترین اثر را روی حرکت و حفظ تعادل آن دارد.

بر همین اساس، دوربین و ویدئوپروژکتور روی سر ربات، و بُرد رسانی روی کمر ربات به صورت عمودی قرار می‌گیرند.

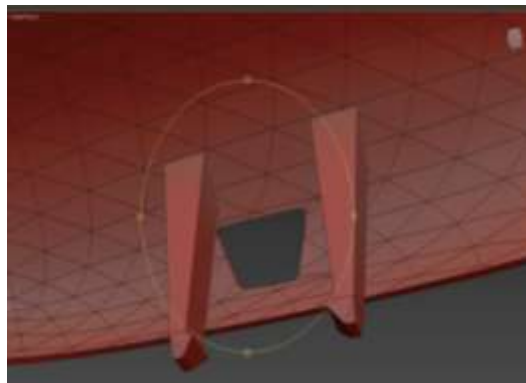
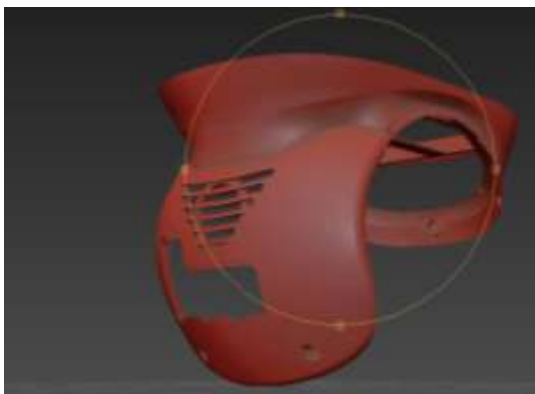
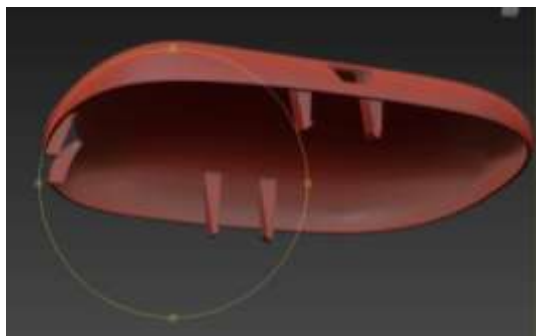
برای جاسازی دوربین و ویدئو پروژکتور، یک محفظه کلاه-مانند، با استفاده از نرم‌افزار Solidworks طراحی شده است. این محفظه دارای دو بخش است: بخش اول محفظه‌ای است که دوربین و ویدئو

پروژکتور را روی سر ربات ثابت می‌کند، و بخش دوم به عنوان یک درپوش روی بخش اول قرار می‌گیرد. شکل ۳۶-۴ نمایی از طراحی این دو بخش را از زوایای مختلف نمایش می‌دهد. با استفاده از فناوری چاپ سه بعدی، این دو بخش به صورت دو قطعه فیزیکی تهیه شده است و با جاسازی پروژکتور، روی سر ربات قرار گرفته است. شکل ۳۷-۳ نمایی از روبوداک را همراه با دوربین، محافظه نگهدارنده نشان می‌دهد.

محافظه نگهدارنده دوربین و پروژکتور



درپوش روی محافظه نگهدارنده



یکپارچه سازی قطعات روی سر ربات



شکل ۳۶-۴- طراحی بخش اول محافظه کلاه-مانند برای جاسازی دوربین و ویدئو پروژکتور



شکل ۴-۳۷-نمایی از روبوداک به همراه دوربین و ویدئو پروژکتور جاسازی شده

برای جاسازی بُرد رسی‌ری در پایین کمر ربات، یک محفظه کوچک در نظر گرفته شده که با یک کمر بند و بند کشی به ربات متصل شده است. در شکل ۴-۳۸ محل جاسازی کیف در بدنه ربات نمایش داده شده است.



شکل ۴-۳۸- محل جاسازی بُرد رسی‌ری، روی بدن ربات

۴-۳. ارزیابی فیزیکی پلتفرم روبات‌اک با استفاده از شبیه‌سازی

با توجه به این که پلتفرم ربات Nao همراه خود تجهیزاتی را مانند رسپری، ویدئو پروژکتور، دوربین و کابل‌های مربوطه، با خود حمل می‌کند، لازم است که تعادل ربات بررسی شود. برای شبیه‌سازی عملکرد روبات‌اک و آزمایش تعادل آن، پس از تغییرات صورت گرفته روی بدنه آن، از نرم‌افزار ویباتز^{۷۴} استفاده شده است. این نرم‌افزار اولین بار در سال ۱۹۹۶ ارائه شده و تا به امروز بیش از ۰۰۰۱ دانشگاه و مؤسسه تحقیقاتی در سطح بین‌المللی از قابلیت‌های آن بهره می‌برند. در ایران نیز دانشگاه تهران، دانشگاه شهید بهشتی و دانشگاه خواجه نصیر از جمله کاربران رسمی این نرم‌افزار هستند. از جمله قابلیت‌های ویباتز می‌توان به موارد زیر اشاره کرد:

- قابلیت شبیه‌سازی دینامیکی ربات (از نظر جرم، مرکز ثقل و ...)
- استفاده از چندین ربات به صورت همزمان در محیط شبیه‌سازی
- قابلیت برنامه‌نویسی با استفاده از زبان‌ها C/C++، Python، Java و برنامه‌های Matlab
- استفاده از جعبه ابزار نرم‌افزار Matlab برای پردازش تصاویر در ربات
- استفاده از خروجی نرم‌افزارهای Solidworks, AutoCad, Blender, 3D Max studio و ... جهت مدل‌سازی ربات
- استفاده از کتابخانه ربات‌های معروف مانند: iRobot, Khepera, NAO, Pioneer, Kuka, Bioloids,
- استفاده از سنسورهای آماده همچون: فاصله‌یاب، ژيروسکوپ، شتاب‌سنج، GPS، قطب‌نما و ...

محیط نرم‌افزار ویباتز به گونه‌ای است که می‌توان عملکرد ربات‌های پراستفاده مانند Nao را همراه با کلیه مشخصات فیزیکی آنها شبیه‌سازی نمود. در بدنه Nao دو قطعه اصلی اضافه شده است: (۱) کلاه روی سر ربات شامل: دیتا پروژکتور، دوربین و پوسته کلاه، و (۲) کیف پشت ربات، شامل: رسپری و باتری. علاوه بر آن، سیم‌های اتصالی نیز از پشت ربات از رسپری به تجهیزات داخل کلاه وصل شده است. بنابراین برای شبیه‌سازی عملکرد روبات‌اک، ابتدا لازم است که در بدنه ربات Nao در محیط ویباتز نیز اجزایی اضافه شود. در ادامه شیوه برنامه‌نویسی در محیط ویباتز برای اضافه نمودن یک جسم خارجی به بدنه Nao تشریح می‌گردد.

⁷⁴ Webots

۴-۳-۱. اتصال یک قطعه خارجی به Nao در محیط ویباتز

برای اتصال یک قطعه به ربات در ویباتز، ابتدا لازم است که قطعه به محیط ویباتز وارد^{۷۵} شود. برای این منظور باید فایل طراحی را به صورت VRML و با پسوند .wrl ذخیره نمود. این طراحی را می توان با استفاده از نرم افزارهای SolidWorks ، AutoCAD ، Blender ، 3D Studio Max ، SketchUp و غیره انجام داد.

برای وارد کردن قطعه به محیط ویباتز، ابتدا باید یکی از دنیاهای پیش فرض در محیط ویباتز اجرا شود، به عنوان مثال دنیای nao.wbt. پس از آن با استفاده از گزینه import VRML 2 در منوی ویباتز، فایل طراحی شده با فرمت .wrl در محیط دنیا وارد می شود. پس از این مرحله قطعه طراحی شده به محیط ویباتز انتقال یافته و قابل مشاهده است، لیکن هنوز این قطعه به بدنه ربات متصل نشده است.

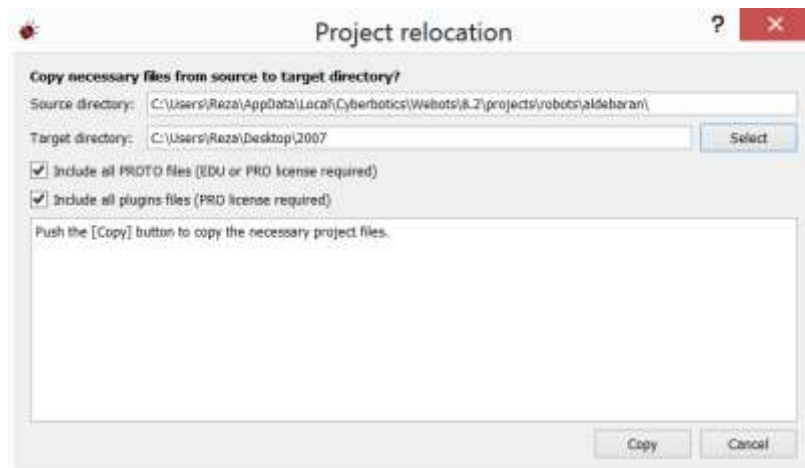
بنابراین لازم است که تغییراتی در کد برنامه Nao در محیط ویباتز ایجاد شود و این قطعات به عنوان بخشی از بدنه اصلی ربات تعریف شوند.

برای این منظور ابتدا لازم است که کدهای Nao در محیط ویباتز بارگذاری شوند که به صورت زیر عمل می شود:

- ابتدا کلیه فایل های proto که حاوی کدهای ربات هستند، را انتخاب می کنیم تا در یک پوشه مشخص بارگذاری شوند (با استفاده از گزینه save as در منوی file). با انتخاب بارگذاری فایل های proto، درون یک پوشه، چهار پوشه مجزا درست می شود که فایل های کدهای Nao در پوشه protos قرار دارند (شکل ۴-۳۹). پوشه ی worlds دنیای ربات است که با تغییر در فایل proto در این دنیا نیز تغییرات ایجاد می شود.

⁷⁵ Import

controllers	۲۰۱۶/۱۲/۰۹ ب.ظ ۱۲:۰۸	File folder
motions	۲۰۱۶/۱۲/۰۹ ب.ظ ۱۲:۰۸	File folder
protos	۲۰۱۶/۱۲/۰۹ ب.ظ ۱۲:۰۸	File folder
worlds	۲۰۱۶/۱۲/۰۹ ب.ظ ۱۲:۰۸	File folder



controllers	۲۰۱۶/۱۲/۰۹ ب.ظ ۱۲:۰۸	File folder
motions	۲۰۱۶/۱۲/۰۹ ب.ظ ۱۲:۰۸	File folder
protos	۲۰۱۶/۱۲/۰۹ ب.ظ ۱۲:۰۸	File folder
worlds	۲۰۱۶/۱۲/۰۹ ب.ظ ۱۲:۰۸	File folder

شکل ۴-۳۹- بارگذاری فایل‌های proto حاوی کدهای ربات در ویاتر

- پوشه protos را باز کرده و فایل nao.proto را باز می‌کنیم. در این فایل اجزای بدن Nao به صورت کد قرار دارد و با اضافه کردن کد قطعه طراحی شده، قطعه روی بدن Nao ثابت می‌شود و به عنوان جزئی از بدن Nao در نظر گرفته می‌شود.
- فایل VRML قطعه طراحی شده را می‌توانیم با یک ویرایشگر ساده متن مانند notpad++ ویرایش کنیم تا در فایل حاوی کد Nao قرار دهیم. کد قطعه طراحی شده به صورت زیر است:

```
DEF Sphere001-397 Transform {
  translation - 85.0 6.2-13.2
  rotation 0 1 4 14
  scale 1.2 1.2 1.2
  children [
    Shape {
      appearance Appearance {
        } material Material
        diffuseColor 0.03137 0.2392 0.5412
```

```
{
  geometry DEF Sphere001-793-FACES IndexedFaceSet {
```

```
coordIndex [...]  
coord Coordinate {  
point [...]
```

```
    {  
      {  
        {  
          [
```

این کد دارای قسمت translation برای تغییر موقعیت قطعه‌ی طراحی شده و rotation برای چرخش آن و scale برای تغییر اندازه‌ی آن است.

- فایل nao.poroto را با استفاده از یک ویرایشگر متن مانند notepad++ باز کرده و کد قطعه طراحی شده در آن قرار می‌دهیم. این طراحی را روی هر قسمت ربات مثلاً سر و کمر می‌توان قرار داد. ما دیتا پروژکتور را در قسمت سر قرار داده و رسپری را روی کمر قرار می‌دهیم. همچنین چگالی این قطعه قابل تنظیم است که با توجه به حجم آن می‌توان چگالی آن را تغییر داد.

- کدهای طراحی در nao.poroto مربوط به این فایل به صورت زیر ذخیره می‌شود:
فایل nao.poroto را که در پوشه‌ی porotos قرار دارد باز می‌کنیم. هر فایل طراحی که در این کد گذاشته شود عضوی از ربات در دنیای مورد نظر محسوب می‌شود و به عنوان یک قطعه خارجی، به ربات متصل خواهد شد. برای اینکه هر یک از قطعات در جای درست خود روی ربات قرار بگیرند باید کد طراحی مربوط به هر قطعه را در قسمت مشخص در nao.poroto قرار داد. برای این کار باید کد طراحی دیتا پروژکتور را در قسمت سر و کد طراحی رسپری را در قسمت کمر قرار داد که به صورت زیر است (کدهای مربوط به قطعات طراحی شده، با رنگ قرمز مشخص شده‌اند):

```
DEF BODY_SLOT Group {  
    children IS bodySlot  
}  
  
DEF JERSEY_SLOT Slot {  
    type "nao jersey"  
    endPoint IS jersey  
}  
  
DEF Sphere001-397 Transform {  
    translation - 70.0-10.0 10.0  
    rotation 5141.3 001 001 0  
    scale 0.4 2.4 1.9  
    children [  
    Shape {  
    appearance Appearance {
```

```

    } material Material
      diffuseColor 0.03137 0.2392 0.5412
    {
      geometry DEF Sphere001-793-FACES IndexedFaceSet {
        coordIndex [...]
        coord Coordinate {
          point [...]
            {
              {
                {
                  [
                    {
                      DEF CHEST_MAIN_SKIN Shape {
                        { appearance DEF MAIN_SKIN_APP Appearance
                        { material Material
                        .....
                      }
                    }
                  {
                    geometry IndexedFaceSet
                    coord DEF CHEST_COORD Coordinate {
                      [...] point
                    }
                    coordIndex [...]
                  {
                    {
                      DEF CHEST_SECONDARY_SKIN Shape {
                        appearance DEF SECONDARY_SKIN_APP Appearance {
                          { material DEF SECONDARY_SKIN_MAT Material
                          .....
                        }
                      }
                    }
                  .....
                coordIndex [...]
                }
                }
                ...
              }
            DEF Sphere001-397 Transform {
              translation - 10.0-21.0 0500.0
              rotation 5141.3 0 0 0
              scale 1.2 2 1.5
              children [

```

```

Shape {
appearance Appearance {
    } material Material
    diffuseColor 0.03137 0.2392 0.5412
}

{
geometry DEF Sphere001-793-FACES IndexedFaceSet {
coordIndex [...]
coord Coordinate {
point [...]
    {
    {
    {
    [
    {
    [
contactMaterial IS contactMaterial
    boundingObject Transform {
    translation 0 0 0.055
    children [
    Capsule {
    height 0.04
    radius 0.05
    {
    [
    {
physics Physics {
    density -1
}%.    if version == "3.3" then%.{
    mass 18.89995
    centerOfMass [
    52570100.0 344540.0
    [
    inertiaMatrix [
    75962200.0 6599100.0 92520600.0-1.19245e- 50
    2.79999e-05 7.82114e- 60
    [
    %.{
    elseif version == "4.0" then%.{
    mass 13.76975
    centerOfMass [
    98440.0 -10100.0
    [
    inertiaMatrix [
    0.00688097 0.0023346 0.00249658 4.09863e- 50
    -3.51465e- 05 8.73519e-60
    [
    %.{
    else%.{
    mass 14.78375
    centerOfMass ]

```



```

7404340.0 -396299000.0
[
  inertiaMatrix [
    000.0304616 200.098262 200.075694 4.09863e- 50
    -3.61413 e-05 8.73519e-60
  ]
  %{\
end%\{
  {
}
}
DEF RShoulderPitch Hinge2Joint {
{...

```

- در پوشه worlds دنیا را باز کرده و مشاهده می‌شود که فایل طراحی روی بدن ربات قرار دارد و به آن متصل شده است. حال با استفاده از نرم افزار Choregraphe 2.1.4 به نرم‌افزار webots متصل شده و می‌توان ربات را کنترل کرد و فرامینی برای انجام حرکات فیزیکی به ربات داد تا بتوانیم عملکرد فیزیکی و پایداری ربات را حین اتصال این قطعات بررسی کنیم.

در شکل ۴-۴۰ نمایی از ربات نائو که روی سر و پشت آن یک شیء خارجی متصل شده، نمایش داده شده است.



شکل ۴-۴۰- نمایی از ربات نائو همراه اتصال دو شیء خارجی روی سر و کمر آن

۴-۳-۲. بررسی پایداری روبوداک با استفاده از ویباتز

با گذاشتن قطعات مورد نظر روی ربات ممکن است پایداری ربات تحت تاثیر قرار بگیرد. در این قسمت پایداری ربات همراه این اجسام در سه حالت نشستن، راه رفتن و بلند شدن بررسی می‌شود. برای انجام این اعمال (نشستن، راه رفتن و بلند شدن) از نرم‌افزار Choregraphe 2.1.4 استفاده می‌شود که تحت شبکه به ویباتز متصل می‌گردد. حرکات فیزیکی اصلی روبوداک که می‌تواند پایداری آن را با چالش مواجه کند، عبارتند از: بلندشدن، راه رفتن و نشستن.

در نهایت لازم است که آستانه پایداری ربات را براساس نرم‌افزار شبیه‌ساز ویباتز، برای وزن‌های مختلف روی سر و کمر مشخص کنیم تا مطمئن شویم که اشیاء اضافه شده خللی در پایداری ربات ایجاد نمی‌کنند. برای این منظور چند آزمایش را طراحی کرده‌ایم:

۱. بررسی آستانه پایداری ربات برای هنگامیکه وزن سر تغییر می‌کند ولی وزن کمر

براساس وزن پیش فرض ربات تنظیم شده است.

۲. بررسی آستانه پایداری ربات برای هنگامیکه وزن کمر تغییر می‌کند ولی وزن سر براساس

وزن پیش فرض ربات تنظیم شده است.

۳. بررسی آستانه پایداری برای حالتی که وزن هر دو شیء روی سر و کمر تغییر می‌کند.

برای دستیابی آستانه برای حالت اول و دوم، ابتدا از یک وزن کوچک شروع کرده‌ایم و عملکرد ربات را از حیث تعادل یا عدم تعادل، در حالات اصلی حرکت در محیط ویباتز بررسی کرده‌ایم. سپس با افزایش تدریجی وزن (۱۰ گرم افزایش یا کاهش در هر آزمایش)، مقدار آستانه مشخص شده است.

برای دستیابی به آستانه تعادل برای حالت سوم، شرایط نزدیک به روبوداک را با وزن‌هایی که در دنیای واقعی روی ربات قرار داده شده، شروع می‌کنیم. سپس در هر مرحله از آزمایش، تغییرات وزن را در قالب افزایش یا کاهش ۲۰ گرمی وزن برای هر یک از اشیاء، اعمال می‌کنیم و برهمین اساس عملکرد ربات را بررسی می‌نماییم.

لازم به ذکر است که در این شبیه‌سازی از تغییرات ممان اینرسی به دلیل نزدیک بودن اشیاء به مرکز ثقل ربات صرفه نظر شده است. همچنین قیود اعمالی توسط موتورها و مقاومت مصالح نیز توسط این شبیه‌ساز در نظر گرفته نشده‌اند. نتایج این آزمایش‌ها در جدول ۴-۲۲ و جدول ۴-۲۳ آمده است. این جدول بیشینه وزن قابل تحمل ربات برای حفظ تعادل را در حالات‌های مختلف نمایش می‌دهد. برای

دستیابی به هر یک از این مقادیر حرکات ربات چند بار (۱۰ بار) تکرار شده است و نتایج حاصل شده براساس حداکثر وزنی است که در دفعات تکرار مختلف حاصل شده است. در اوزانی که در جدول آمده، وزن سر یا بدن نیز لحاظ شده است. وزن سر نائو در حالتی که شیء روی آن قرار نگیرد ۰٫۶۸ کیلوگرم است. اعداد این جداول به معنی آستانه تحمل ربات برای انجام حرکات مشخص شده در جدول است. به عنوان مثال در جدول جدول ۴-۲۲، در حالتیکه شیء مانند پروژکتور روی سر قرار دارد، بیشینه وزنی که ربات می‌تواند تحمل کند (همراه با وزن سر نائو)، به میزان ۱٫۵۲ کیلوگرم است. در واقع اگر ۱۰ گرم به این وزن اضافه شود، و وزن به ۱٫۵۳ برسد، و ۱۰ بار آزمایش نشستن تکرار شود، حداقل در یکی از دفعات آزمایش، ربات تعادل خود را از دست می‌دهد.

از آنجاییکه مدلسازی حرکت بلندشدن در ویباتر متفاوت از دستورات ارسالی توسط Choregraphe است، امکان شبیه‌سازی عملکرد ربات در این حرکت وجود ندارد. در واقع، ربات در هیچ کدام از آزمایش‌ها، حتی زمانی که ربات در حالت پیش‌فرض ربات نائو قرار دارد، قادر به بلند شدن نیست.

جدول ۴-۲۲- آستانه تحمل وزن ربات برای اضافه نمودن یک شیء، برای حالات فیزیکی مختلف (وزن به کیلوگرم)

وضعیت اشیاء روی بدن	حرکت ربات	نشستن	راه رفتن
همراه پروژکتور	وزن سر	۱/۵۲	۹/۶۰
	وزن بدن	۱/۰۴	۱/۰۴
همراه رسپری	وزن سر	۰/۶۸	۰/۶۸
	وزن بدن	۳/۰۳	۱۲/۷۴

جدول ۴-۲۳- آستانه تحمل وزن ربات هنگام اضافه نمودن دو شیء به صورت همزمان، برای حالات فیزیکی مختلف (وزن به کیلوگرم)

راه رفتن				نشستن				حرکت ربات / وزن اشیاء روی بدن
۰/۹۰	۰/۹۰	۰/۹۰	۰/۹۰	۰/۹۰	۰/۸۶	۰/۸۶	۰/۹۰	وزن سر
۱/۳۵	۱/۳۵	۱/۳۱	۱/۳۱	۱/۳۵	۱/۳۵	۱/۳۱	۱/۳۱	وزن بدن

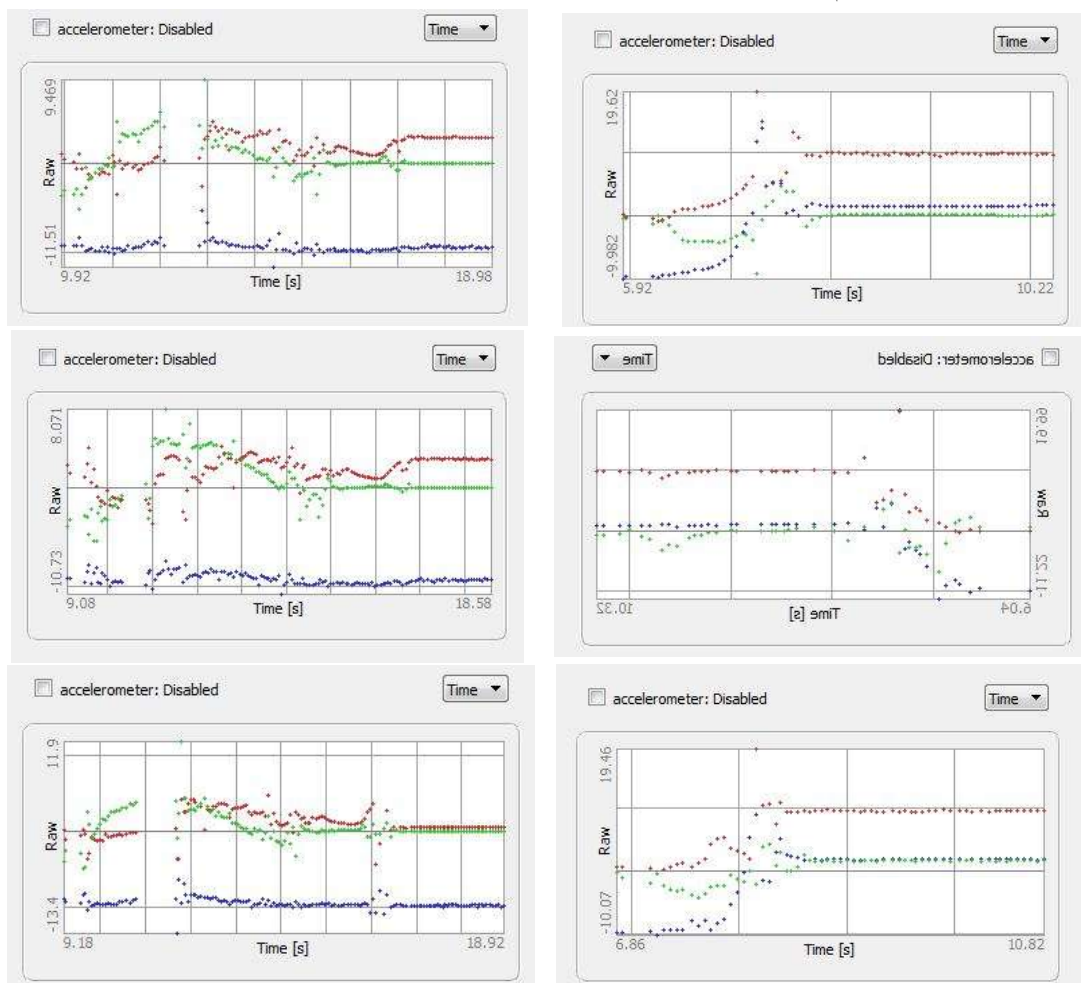
با توجه به آنکه وزن پروژکتور، دورین و محفظه کلاه ماندی که روی سر ربات قرار دارد، به میزان حدود ۲۰۰ گرم، و وزن رسپری و محفظه نگهدارنده آن حدود ۱۰۰، بنابراین براساس نتایج شبیه

سازی، می‌توان گفت که اضافه نمودن این قطعات روی ربات نمی‌تواند تاثیری در نشستن یا راه رفتن آن داشته باشد.

علاوه بر نتایج فوق، نمودارهای شتاب‌سنج وژیروسکوپ ربات هنگام حفظ تعادل و عدم تعادل نیز بررسی شده و چند نمونه در شکل‌های زیر نمایش داده شده‌اند. این نمودارها برای حالات نشستن و راه رفتن، برای سنسورهای شتاب‌سنج وژیروسکوپ ترسیم شده‌اند. در هر شکل نمودارهای مربوطه برای سه بار تکرار نمایش داده شده است. در نمودارهای مربوط بهژیروسکوپ، دوران سنسورژیروسکوپ نسبت به زمان در دو محور نمایش داده شده است. در نمودارهای شتاب‌سنج، میزان تغییرات شتاب اندازه‌گیری شده توسط شتاب‌سنج در سه محور X, Y, Z (با سه رنگ مختلف) نمایش داده شده است. با توجه این نمودارها مشاهده می‌شود، هنگامیکه ربات تعادل خود را از دست می‌دهد، یک ناپیوستگی در آنها دیده می‌شود و الگوی رفتار آنها، هنگام عدم تعادل تغییر می‌کند.

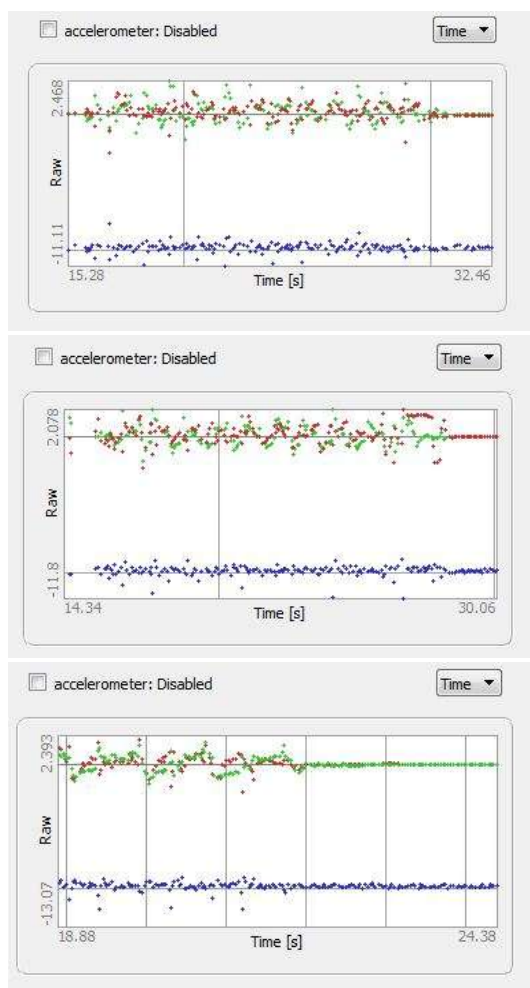
تعادل

عدم تعادل

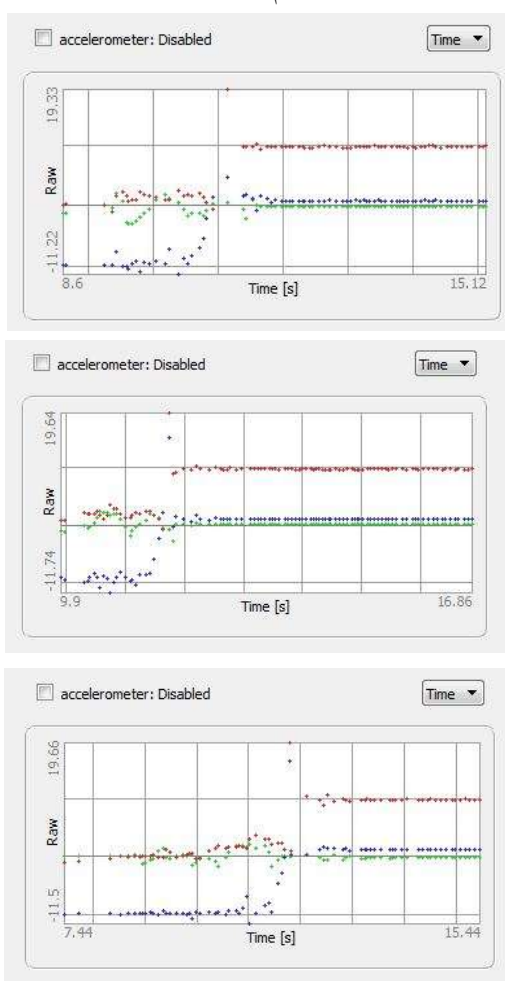


شکل ۴-۱- نمودار شتاب‌سنج در حالت تعادل و عدم تعادل هنگام نشستن

تعادل

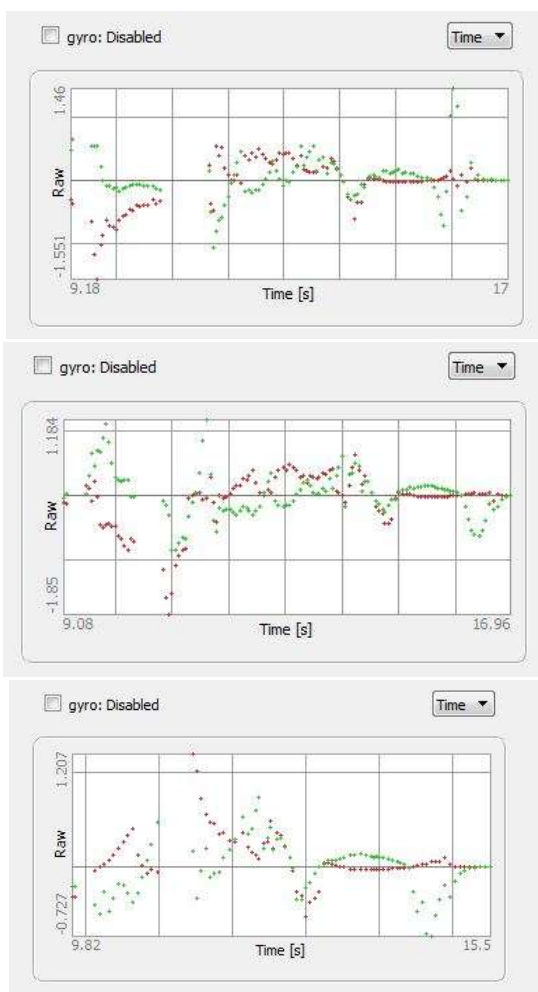


عدم تعادل

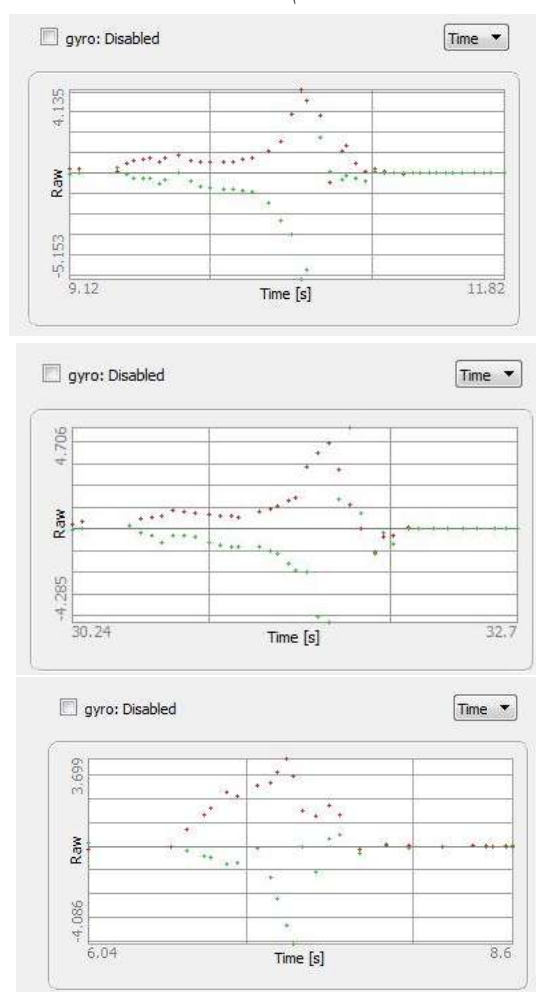


شکل ۴-۲- نمودار شتابسنج در حالت تعادل و عدم تعادل هنگام راه رفتن

تعادل

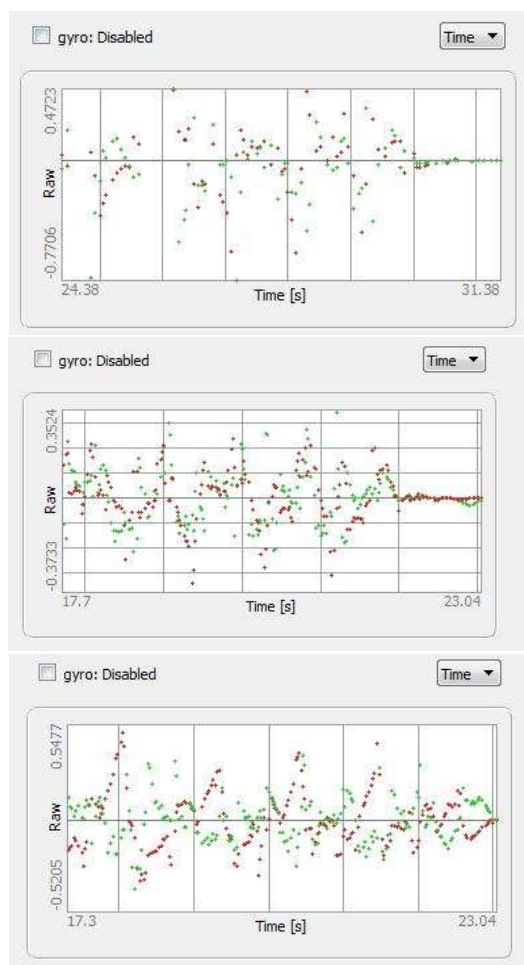


عدم تعادل

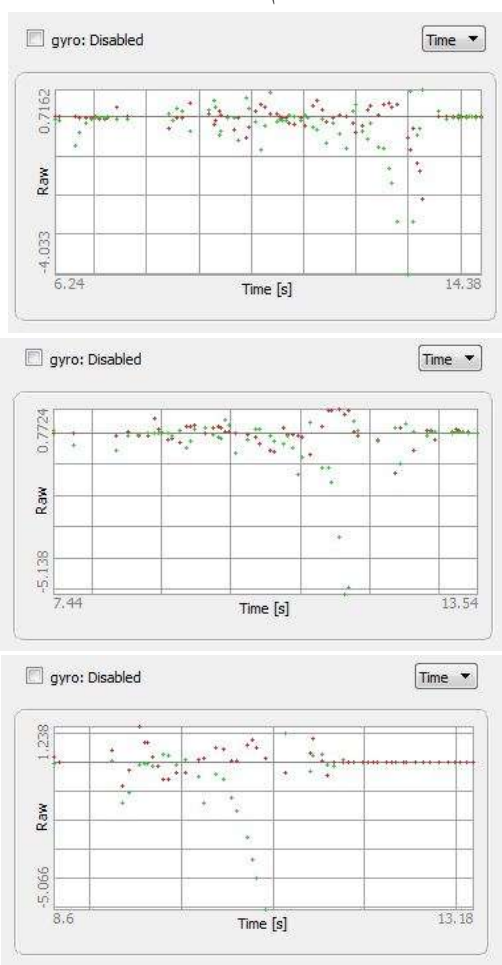


شکل ۴-۳- نمودار ژيروسکوپ در حالت تعادل و عدم تعادل هنگام نشستن

تعادل



عدم تعادل



شکل ۴-۴- نمودار ژيروسکوپ در حالت تعادل و عدم تعادل هنگام راه رفتن

فصل پنجم:

تشریح سیستم روبوداک و اجزای آن

۵. تشریح سیستم روبوداک و اجزای آن

در این فصل، ابتدا در بخش اول معماری سیستم روبوداک تشریح می‌گردد و اجزای اصلی آن و ارتباطات بین آنها مشخص می‌شوند. برای شرح معماری روبوداک ابتدا ساختار مفهومی آن، نمودار اجزای^{۷۶} و نمودار فعالیت^{۷۷} ترسیم می‌شوند. سپس نمودار بلوکی^{۷۸} برای مشخص نمودن ارتباطات فیزیکی اجزای سیستم ارائه می‌شود.

در بخش‌های بعد، هر یک از زیرسیستم‌های اصلی روبوداک تشریح می‌گردند و عملکرد آنها به صورت کامل مشخص می‌شود.

در بخش انتهایی نیز کدهای اصلی برنامه سیستم روبوداک براساس عملکرد هر یک از زیرسیستم‌ها ارائه می‌گردد. لازم به ذکر است که در این فصل تنها بخش‌های اصلی کدها و برنامه‌های اصلی آورده شده است. شرح کامل کدها به صورت جداگانه در پیوست آمده است.

۵-۱. معماری سیستم روبوداک

با توجه به مدل مفهومی روبوداک که در فصل دوم به آن اشاره شد، روبوداک در این مرحله لازم است که بتواند با کاربر به صورت گفتاری ارتباط برقرار کند، عبارت مورد نظر کاربر را در گنج جستجو کند و نتایج را در قالب الگوهای مصورسازی داده‌ها نمایش دهد و از روی یک متن فارسی بخواند. بنابراین براساس آنچه در فصل دوم نیز مشخص شد، عمده پردازش‌های روبوداک حول کارکردهای اصلی زیر قرار می‌گیرد:

- پردازش صوت و گفتار و تبدیل آن به متن
- پردازش تصویر و تشخیص یک علامت یا حالت مشخص
- پردازش عبارت جستجو و بکارگیری الگوریتم بازیابی اطلاعات مناسب در گنج با توجه به اطلاعات اصطلاحنامه‌های تخصصی در ایرانداک و نمایش مصوراطلاعات
- پردازش متن و تبدیل آن به صوت

⁷⁶ Component diagram

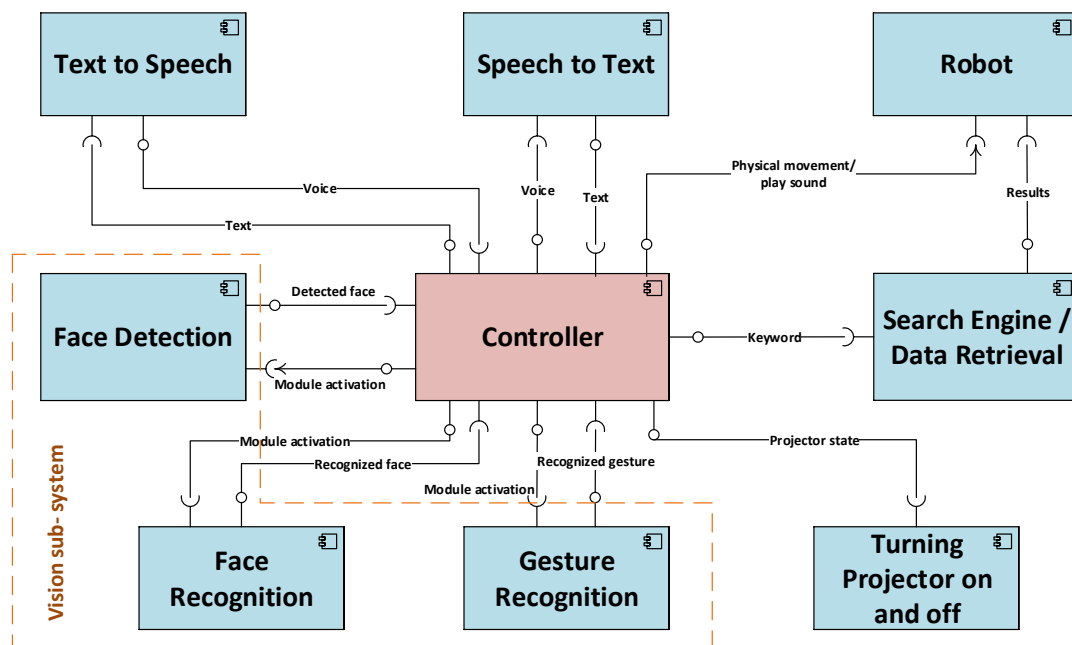
⁷⁷ Activity diagram

⁷⁸ Block diagram

برهمن اساس پنج زیر سیستم اصلی در معماری روباتک درنظر گرفته شده که هر یک بخشی از کارکردهای روباتک را برآورده می‌سازند. این زیر سیستم‌ها عبارتند از:

- زیر سیستم بینایی: برای تشخیص صورت، بازشناسی چهره و شناسایی برخی علائم و حرکات
- زیر سیستم تشخیص گفتار: برای تبدیل گفتار لاتین و فارسی به متن
- زیر سیستم جستجو و بازیابی اطلاعات: برای جستجو و بازیابی اطلاعات در پایگاه گنج به صورت فارسی، و در گوگل اسکولار به صورت انگلیسی
- زیر سیستم تبدیل متن به گفتار: برای خواندن از روی یک متن

هر یک از این زیرسیستم‌ها، خود از کارکردهای اصلی تشکیل شده‌اند. علاوه بر این زیر سیستم‌ها، دو ماژول اصلی هم در سیستم روباتک وجود دارد که عبارتند از: ماژول اتصال به پروژکتور برای عملیات روشن یا خاموش کردن آن، و ماژول ربات برای کنترل حرکات فیزیکی آن. با توجه به این زیرسیستم‌ها، نمودار اجزای سیستم روباتک به صورت شکل ۱-۵ است.



شکل ۱-۵- نمودار اجزای سیستم

در این نمودار اجزای face recognition، face detection و gesture recognition همگی زیرسیستم بینایی را تشکیل می‌دهند. این نمودار شامل اجزای زیر است:

۱- کنترلر (controller): وظیفه یکپارچه سازی و هدایت سایر اجزا را بر عهده دارد و آن‌ها را در زمان مناسب فراخوانی می‌کند.

۲- زیر سیستم تبدیل متن به گفتار (T2S): این زیرسیستم با دریافت یک فایل متنی، آن را به فایل صوتی تبدیل می‌کند.

۳- زیرسیستم تشخیص گفتار (S2T): با دریافت صوت از کاربر می‌تواند آن را پردازش کرده و به متن تبدیل کند.

۴- زیر سیستم بینایی شامل:

- Face detection: با فعال شدن این ماژول، دوربین روشن شده و تشخیص چهره آغاز می‌شود.

- Face recognition: در صورت تشخیص صورت، این ماژول برای شناسایی چهره فعال می‌شود.

- Gesture recognition: توسط این ماژول می‌توان حرکت دست کاربر را شناسایی کرد.

۵- ماژول اتصال به پروژکتور (Turning projector on and off): با ارسال فرمان به این ماژول، می‌توان پروژکتور را در مواقع لزوم روشن یا خاموش کرد.

۶- ماژول ربات (Robot): ماژول ربات مربوط به بدنه فیزیکی ربات بوده و فرمان‌های فیزیکی مانند نشستن یا بلند شدن را اجرا می‌کند.

۷- زیر سیستم جستجو و بازیابی اطلاعات (Search Engine/data retrieval): این زیرسیستم دربردارنده کلیه قسمت‌های مربوط به موتور جستجو و پایگاه داده است که از طریق یک سرور وب اتصال آن با کنترلر اصلی برقرار می‌شود.

براساس نمودار اجزای نمایش داده شده در شکل ۵-۱، کنترلر، درخواست‌ها را به زیر سیستم‌ها می‌فرستد و خروجی‌های لازم را از آنها دریافت می‌نماید.

۵-۱-۱. سناریوهای تعاملی روباتک و انسان

شیوه عملکرد کنترلر براساس کرکردهای اصلی است که برای روباتک در این مرحله از توسعه روباتک تعریف شده است. لیکن برای مشخص شدن دقیق نحوه عملکرد آن، و نحوه تعامل یک فرد یا روباتک، لازم است که نمودار فعالیت سیستم روباتک نیز ترسیم گردد تا سناریوهای تعاملی و

توالی انجام کار توسط کنترلر مشخص شود. بنابراین در این بخش نمودار فعالیت سیستم روبوداک برای تعامل با یک فرد ترسیم می‌گردد.

با توجه به الزامات روبوداک برای این مرحله از توسعه، نوع تعامل روبوداک و یک فرد را در قالب دو دسته اصلی در نظر می‌گیریم:

- تعامل شناخت اولیه یا «Greeting»: تعامل شناخت اولیه، شروع تعامل روبوداک با یک فرد است. این تعامل به منظور شناسایی فرد تعامل کننده و برقراری ارتباط صوتی و تصویری اولیه با وی صورت می‌گیرد. در قالب این تعامل، روبوداک، فرد را با استفاده از سیستم بینایی شناسایی می‌کند و پاسخ چند پیغام صوتی وی را می‌دهد. در نهایت در صورتیکه فرد تعامل کننده با ربات ناشناس تشخیص داده شود، آنگاه ربات به درخواست‌های بعدی او پاسخ نمی‌دهد.
- تعامل جستجو یا «Search»: پس از اینکه روبوداک یک ارتباط اولیه با فرد برقرار نمود، و فرد را به عنوان یک فرد آشنا شناسایی کرد، آنگاه درخواست فرد را پاسخ می‌دهد. برای پاسخگویی به درخواست فرد، لازم است که روبوداک از نظر فیزیکی در حالتی قرار بگیرد که بتواند نتایج درخواست‌ها از طریق پروژکتور نمایش دهد. برای این منظور روبوداک:

□ در صورتی که نشسته باشد، بلند می‌شود،

□ موقعیت فیزیکی بدن ربات در حالت نمایش قرار می‌گیرد، و

□ پروژکتور را روشن می‌کند.

براساس الزامات توسعه روبوداک در این مرحله، روبوداک لازم است که درخواست جستجو

در پایگاه‌های علمی و نمایش نتیجه را، به همراه خواندن از روی یک متن پاسخ دهد.

بنابراین درخواست‌های فرد در سه دسته اصلی قرار می‌گیرد:

- درخواست جستجو و نمایش اطلاعات روی پروژکتور:

○ جستجو یک عبارت فارسی در گنج: کاربر عبارت جستجو را به صورت

صوتی (یا از طریق صفحه کلید و رابط کاربری) به روبوداک اعلام

می‌کند. روبوداک، از طریق سیستم تشخیص گفتار فارسی (در صورت

نیاز)، گفتار کاربر را به متن تبدیل می‌کند و عبارت مورد نظر کاربر را از

متن استخراج می‌کند. در نهایت با استفاده از زیر سیستم جستجو و بازیابی اطلاعات، عبارت را در گنج جستجو می‌کند و نتایج را در یک صفحه نمایش از طریق پروژکتور نمایش می‌دهد.

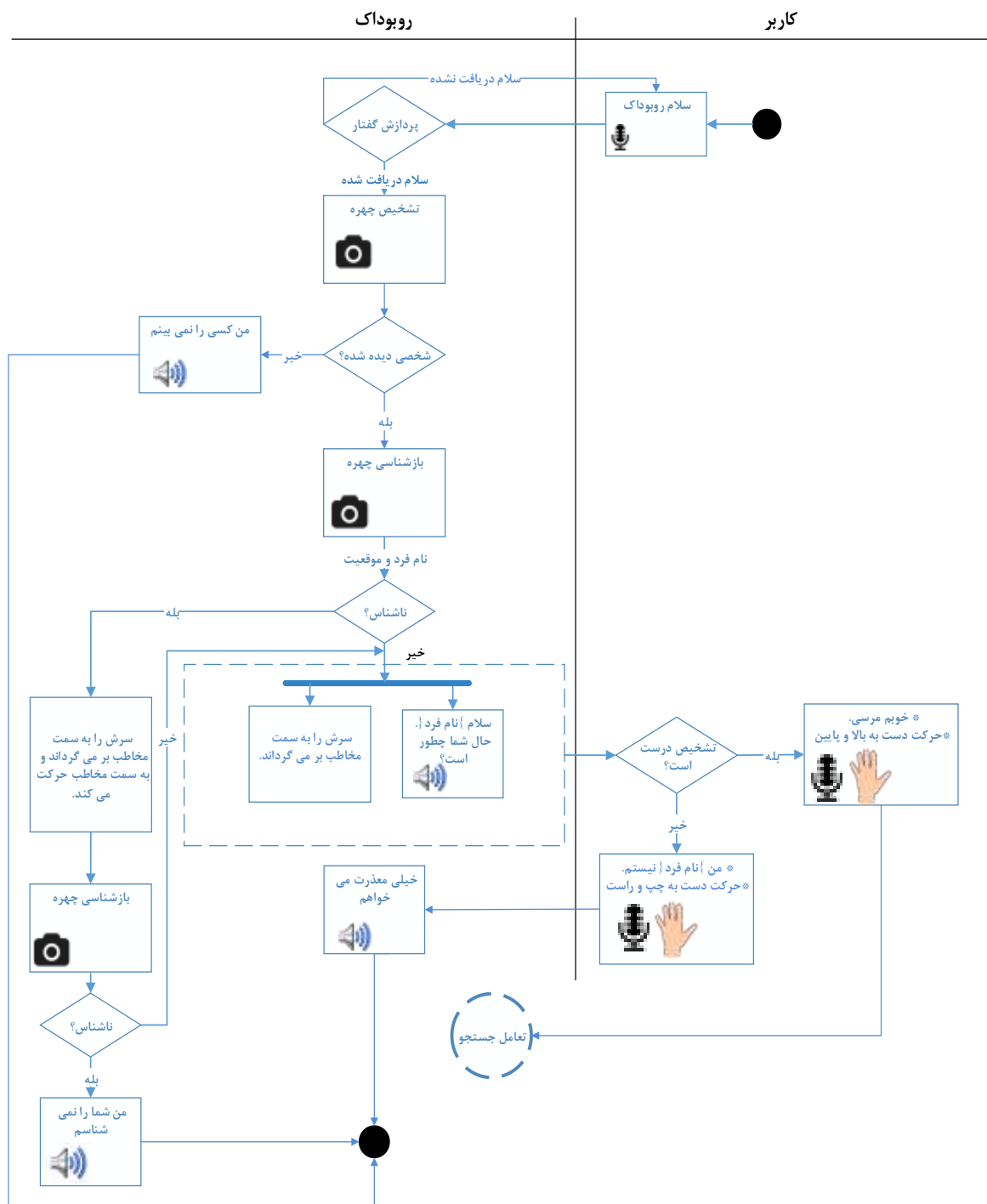
○ جستجوی یک عبارت انگلیسی در گوگل اسکولار: کاربر عبارت جستجو را به صورت صوتی به کاربر اعلام می‌کند. روبوداک با استفاده از سیستم تشخیص گفتار انگلیسی، عبارت صوتی را به متن تبدیل می‌کند. سپس آن عبارت را در گوگل اسکولار جستجو می‌نماید و نتایج را در یک صفحه نمایش از طریق پروژکتور نمایش می‌دهد.

- درخواست خواندن از روی متن: با اعلام درخواست خواند از روی یک متن، صفحه مورد نظر در رابط کاربری باز می‌شود و کاربر متن مورد نظر را از طریق صفحه کیبورد و موس وایرلس در رابط کاربری وارد می‌کند و روبوداک از طریق سیستم تبدیل متن به گفتار، متن را می‌خواند.
- درخواست خاتمه کار: فرد با اعلام اتمام کار به صورت صوتی به روبوداک فرمان اتمام کار با سیستم را اعلام می‌کند. روبوداک صوت دریافت شده را پردازش می‌کند و در صورتی که مربوط به اتمام کار باشد، به حالت اولیه برمی‌گردد.

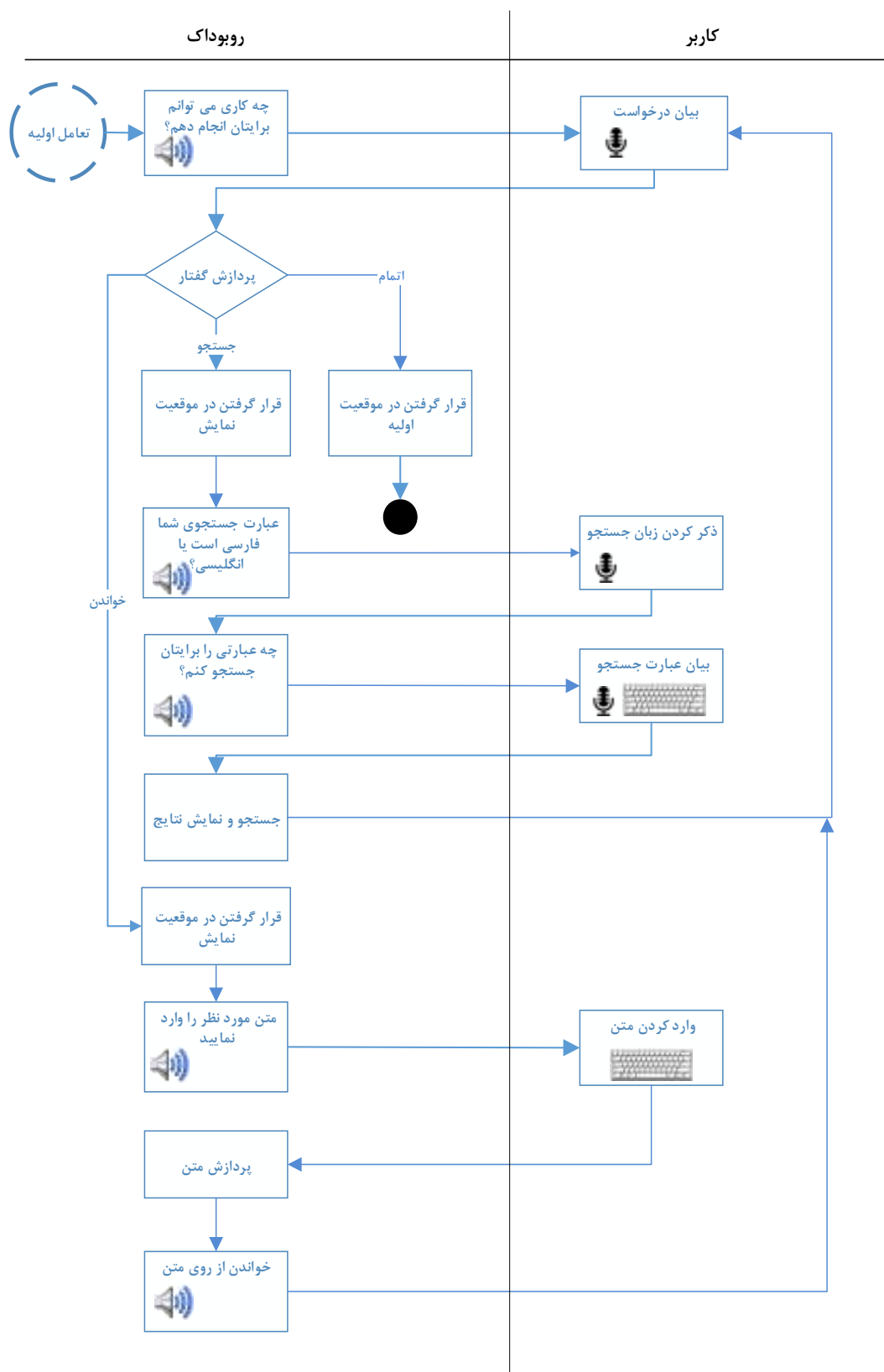
در کلیه تعاملات فرد و روبوداک، برقراری ارتباط صوتی در اولویت است، به این مفهوم که درخواست‌های فرد به صورت صوتی به روبوداک ارسال می‌شود و روبوداک پس از تشخیص گفتار، درخواست مورد نظر را پاسخ می‌دهد.

جزئیات تعامل فرد و روبوداک در قالب دو نوع تعامل «Greeting» و «Search» با استفاده از نمودار فعالیت در شکل ۵-۲ و

شکل ۵-۳ نمایش داده شده است.



شکل ۵-۲- تعامل شناخت اولیه، «Greeting»

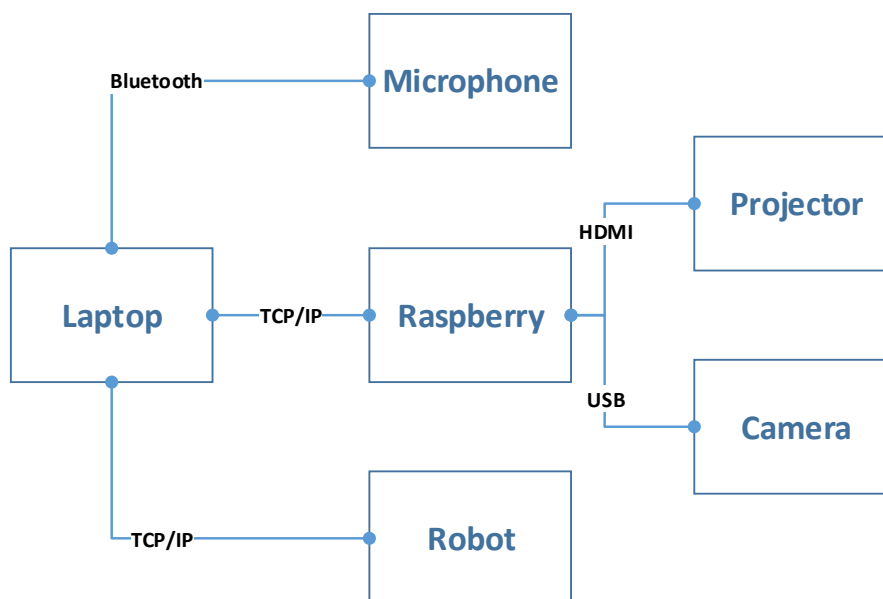


شکل ۳-۵- تعامل جستجو، «Search»

۲-۱-۵. نمودار بلوکی ارتباطات فیزیکی اجزای سیستم روبوداک

در این بخش نمودار فیزیکی ارتباطات بین اجزای مختلف سیستم روبوداک مشخص می‌شود. در این نمودار نوع ارتباط بین بخش‌های مختلف سیستم مشخص شده است. برای نمایش این ارتباطات، از نمودار بلوکی استفاده شده که در شکل ۴-۵ ترسیم شده است. همانگونه که در شکل مشخص شده، این ارتباطات در یکی از انواع زیر قرار می‌گیرند:

- ارتباطات کابلی مانند HDMI و USB: پروژکتور و دوربین از طریق ارتباطات کابلی، به ترتیب از طریق پورت HDMI و USB به رسپری متصل هستند.
- ارتباطات بیسیم مانند Bluetooth و TCP/IP: ارتباط بین رسپری و لپ تاپ، و ربات و لپ تاپ از طریق شبکه در قالب پروتکل TCP/IP است و ارتباط بین میکروفون و لپ تاپ از طریق Bluetooth است.



شکل ۴-۵- نمودار بلوکی سیستم روبوداک

۲-۵. زیرسیستم بینایی

بخش پردازش تصویر در این پژوهش شامل سه بخش تشخیص صورت^{۷۹}، بازشناسی چهره^{۸۰} و تشخیص حرکت^{۸۱} است. در ادامه توضیحاتی درباره‌ی الگوریتم‌های استفاده شده در هر بخش ارائه

^{۷۹} Face Detection

^{۸۰} Face Recognition

^{۸۱} Gesture Detection

می‌شود. برای پیاده‌سازی این الگوریتم‌ها و استفاده از آنها در سیستم بینایی روبوداک، از کتابخانه OpenCV استفاده شده است. توضیحات مربوط به نتایج پیاده‌سازی این الگوریتم‌ها با استفاده از OpenCV در فصل بعدی ارائه خواهد شد.

۵-۲-۱. تشخیص چهره

تشخیص صورت به معنای تشخیص بخشی از تصویر به عنوان صورت انسان است. برای تشخیص صورت دسته‌بندی‌های مختلفی وجود دارد. گاهی این الگوریتم‌ها بسیار ساده هستند و فقط کاربرد خاص دارند، مانند تشخیص صورت بر اساس رنگ. البته این نوع از الگوریتم‌ها در برابر تغییرات روشنایی مقاوم نیستند و در صورتیکه رنگ پوست افراد تغییر کند نتایج مناسبی را ارائه نمی‌دهند. علاوه بر روش‌های مبتنی بر رنگ، روش‌های دیگری نیز هستند که براساس الگوریتم‌های دسته‌بندی^{۸۲} عمل می‌کنند که در آنها، ویژگی‌های تشخیص معمولاً براساس الگوی اجزای صورت تعریف می‌شود.

دو روش پرکاربرد در این زمینه، الگوریتم‌های Haar-Cascades و LBP-Cascades هستند (Viola & Jones, 2001). در ادامه توضیح مختصری از عملکرد هر یک از این دو روش ارائه می‌شود.

۵-۲-۱-۱. دسته‌بندی‌کننده Haar Cascades

دسته‌بندی‌کننده Haar feature-based cascades یکی از اثربخش‌ترین الگوریتم‌های تشخیص اشیاء است که در سال ۲۰۰۱ به وسیله ویلا و جونز^{۸۳} پیشنهاد شد (Viola & Jones, 2001). این الگوریتم بر مبنای روش‌های یادگیری ماشین طراحی شده که برای تشخیص اشیاء (اعم از صورت یا هر شیء دیگری) از آن استفاده می‌شود. این الگوریتم از دو مرحله آموزش و تشخیص تشکیل شده است. در مرحله آموزش، مجموعه‌ای از تصاویر حاوی شیء مورد نظر (تصاویر مثبت^{۸۴}) و مجموعه‌ای از تصاویر بدون شیء مورد نظر (تصاویر منفی^{۸۵}) با اندازه‌ی تصویرهای یکسان، در نظر گرفته می‌شود و دسته‌بندی‌کننده Haar براساس این دو مجموعه آموزش داده می‌شود.

⁸² Classification

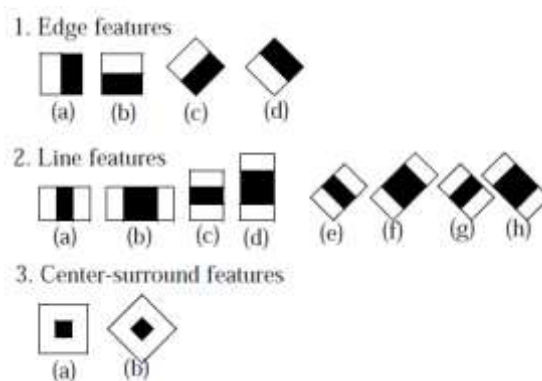
⁸³ Viola and Jones

⁸⁴ Positive

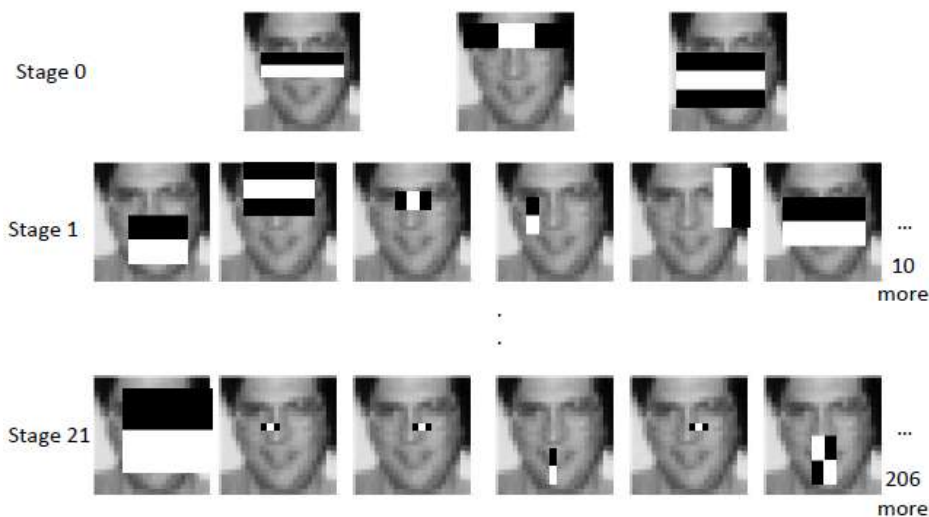
⁸⁵ Negative

برای تشخیص صورت، مسلماً لازم است که مجموعه‌ای از تصاویر حاوی صورت و مجموعه‌الی دیگر فاقد صورت برای آموزش در نظر گرفته شود. با آموزش دسته‌بندی کننده، ویژگی‌های شیء (یا صورت) استخراج می‌شود.

در این روش، برای جستجوی تصویر صورت در کل تصویر دریافت شده، پنجره‌های جستجو^{۸۶} را در کل تصویر حرکت داده و تمام موقعیت‌ها را با دسته‌بند مورد نظر، مورد بررسی قرار می‌دهد. نمونه‌هایی از پنجره‌های جستجویی که در این الگوریتم مورد استفاده قرار می‌گیرد در شکل ۵-۵ و شکل ۵-۶ نشان داده شده‌اند.



شکل ۵-۵- نمونه‌هایی از پنجره‌های جستجو که برای استخراج ویژگی‌های Haar مورد استفاده قرار می‌گیرند



شکل ۵-۶- نمونه‌ای از نحوه انتقال پنجره‌های جستجو در تمام تصویر

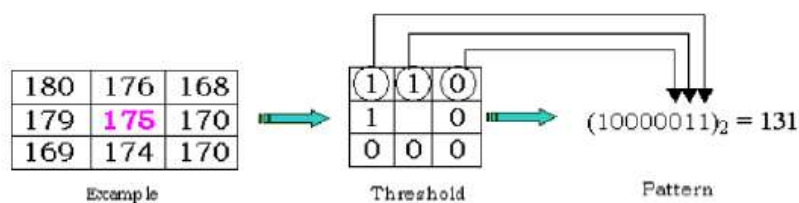
⁸⁶ Search Window

هر دسته از این پنجره‌های جستجو، برای پیدا کردن بخشی از صورت مورد استفاده قرار می‌گیرند. هر ویژگی یک مقدار واحد است که از تفاضل مجموع پیکسل‌های زیر مستطیل سفید از مجموع پیکسل‌های زیر مستطیل سیاه بدست می‌آید. محاسبه این مقدار برای همه‌ی مکان‌های تصویر و تمام اندازه‌های ممکن برای پنجره جستجو و همچنین برای تمام پنجره‌های جستجو، باعث بدست آمدن مقدار بسیار زیادی از ویژگی‌ها می‌شود. همین امر باعث می‌شود که زمان آموزش و تست با استفاده از این ویژگی افزایش پیدا کند (Lienhart, Kuranov, "Face Detection using Haar Cascades," n.d.; & Pisarevsky, 2003).

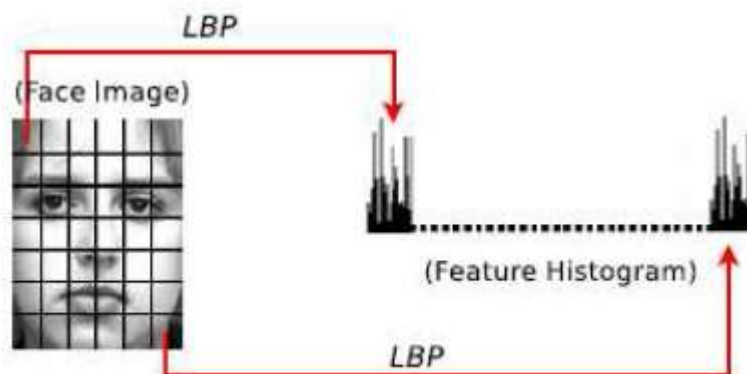
۲-۱-۲-۵. دسته‌بندی کننده LBP-Cascades

سرعت آموزش و تشخیص دسته‌بندی کننده LBP، در شرایط مجموعه‌های آموزش و تست یکسان، بسیار بیشتر از Haar است. از لحاظ مقایسه‌ی دقت، این امر وابستگی زیادی به داده‌های آموزشی مورد استفاده دارد. به عنوان مثال، در شرایطی که تغییرات روشنایی زیادی در تصاویر آموزش وجود داشته باشد، دسته‌بند LBP در برابر این تغییرات مقاوم‌تر عمل می‌کند (Zafeiriou, Zhang, & Zhang, 2015).

استخراج ویژگی LBP به این گونه است که تمام پیکسل‌های یک تصویر را با استفاده از یک آستانه-گیری به صورت باینری برچسب گذاری می‌کند. این آستانه‌گیری بر اساس پیکسل مرکزی از یک مربع همسایگی $n \times n$ از آن پیکسل انجام می‌شود. سپس این اعداد باینری را به صورت رشته‌ای و پشت سر هم قرار می‌دهند و یک کد ایجاد می‌شود. از این کد برای تشخیص بافت یک قطعه از تصویر استفاده می‌شود (Chang-yeon, 2008). نمونه‌ای از انجام این محاسبات بر روی یک تصویر، در شکل ۵-۷ و شکل ۵-۸ نشان داده شده است.



شکل ۵-۷- نمونه‌ای از محاسبه ویژگی LBP برای یک همسایگی ۳*۳



شکل ۵-۸- نمونه قطعه بندی تصویر و اعمال الگوریتم استخراج ویژگی LBP بر روی یک تصویر

از آنجایی که تعداد ویژگی‌های LBP مورد نیاز برای تشخیص صورت کم‌تر از ویژگی‌های Haar است و همچنین نیاز به حافظه و محاسبات کم‌تری برای استخراج این ویژگی‌ها وجود دارد، این ویژگی‌ها برای سیستم‌های جاسازی شده^{۸۷} و سیستم‌های با حافظه کم مانند موبایل، بسیار مناسب هستند (Chang-yeon, 2008). سیستم‌های جاسازی شده سیستم‌های کامپیوتری هستند که در سیستم‌های مکانیکی یا الکتریکی بزرگ‌تر برای انجام محاسبات بلادرنگ^{۸۸} مورد استفاده قرار می‌گیرند (Embedded system, n.d.).

ویژگی‌های LBP، حاوی اطلاعات بیشتری نسبت به ویژگی‌های Haar هستند. به همین دلیل برای انجام یک دسته‌بندی خوب، تعداد ویژگی‌های مورد نیاز برای دسته‌بندی LBP کم‌تر از تعداد ویژگی‌هایی است که برای دسته‌بندی Haar مورد نیاز است. بنابراین، در حالی که دسته‌بندی بر اساس ویژگی‌های LBP تقریباً با دقت دسته‌بندی بر اساس ویژگی‌های Haar انجام می‌شود، محاسبات آن اغلب در حدود ۱۰ برابر فشرده‌تر (با تعداد کمتری از ویژگی) هستند و معمولاً بسیار سریع‌تر در اکثر سیستم عامل‌ها (از جمله کامپیوترهای معمولی) انجام می‌شود (Bilaniuk et al., 2014).

۵-۲-۲. بازشناسی چهره

بازشناسی چهره طی ۱۰ سال گذشته یکی از محبوب‌ترین موضوعات پژوهشی در زمینه بینایی ماشین بوده است. مسئله بینایی ماشین را می‌توان به صورت «شناسایی یک یا چند نفر در تصاویر ویدیویی یا تصاویر ثابت از یک صحنه با استفاده از یک پایگاه داده‌ی ذخیره شده از تصاویر» بیان کرد (Face

⁸⁷ Embedded Systems

⁸⁸ Real-time Computing

استخراج ویژگی‌های مفید از تصویر و انجام نوعی دسته‌بندی از آن‌هاست. (recognition homepage,” n.d). به عبارت دیگر، بازشناسی چهره به صورت خودکار شامل

الگوریتم‌های بازشناسی چهره شامل دو دسته هستند: الگوریتم‌های مبتنی بر تصویر ثابت و الگوریتم‌های مبتنی بر تصاویر ویدیویی (“Face recognition homepage,” n.d). در طول چند سال گذشته تحقیقات بسیاری پیرامون بازشناسی چهره در تصاویر متوالی انجام شده است. بازشناسی چهره از تصاویر ویدیویی به دلیل کاهش کیفیت تصاویر و کوچک بودن اندازه صورت در تصویر، کار دشوارتری است.

بازشناسی چهره نیز یک روش دسته‌بندی محسوب می‌شود که نیاز به مرحله آموزش دارد. در برنامه‌های کاربردی، معمولاً تصویر ورودی صرفاً شامل صورت انسان نیست و ممکن است که پس زمینه‌های پیچیده و زیادی داشته باشند. به همین دلیل معمولاً فرایند بازشناسی چهره از دو بخش اصلی تشکیل می‌شود: تشخیص چهره (صورت) و شناسایی هویت چهره. بنابراین قبل از بازشناسی چهره، نیاز به تشخیص صورت انسان در تصویر مورد نظر است. بعد از تشخیص صورت، بخشی از تصویر که به عنوان صورت تشخیص داده شده، برای احراز هویت به الگوریتم بازشناسی چهره ارسال می‌شود. دو الگوریتم پرکاربرد در این زمینه، الگوریتم‌های Eigenfaces و Fisherfaces هستند (Jaiswal, 1102). در ادامه توضیحات مختصری درباره هر یک از این دو الگوریتم ارائه می‌شود.

۵-۲-۱. الگوریتم Eigenfaces

دسته‌بندی کننده Eigenfaces یک رویکرد جامع از بازشناسی چهره را بیان می‌کند. در این الگوریتم ویژگی‌های تصویر براساس یک فرایند کاهش بُعد، استخراج می‌شوند. در واقع، پس از استخراج صورت از تصویر اصلی، تصویر صورت در قالب یک بردار، به یک فضای برداری با بُعد کمتر انتقال می‌یابد. فضای جدید این قابلیت را دارد که ویژگی‌های اصلی تصویر را نمایان کند. بنابراین اگر یک تصویر با اندازه $p \times q$ را در قالب یک بردار pq بُعدی در نظر بگیریم، در فضای معمول این تصویر دارای pq ویژگی است. لیکن همه این ویژگی‌های برای بازشناسی چهره مناسب و مفید نیستند. نتیجه، برای سادگی در دسته‌بندی، باید تصویر را به فضایی با بُعد کمتر انتقال داد که در فضای جدید تنها ویژگی‌های مفید از تصویر باقی بماند (Fisher, 1936). در این الگوریتم، برای کاهش بعد از روش آنالیز مولفه‌های اصلی (PCA)^{۸۹} استفاده می‌شود. در این روش کاهش بعد، سعی می‌شود که

^{۸۹} Principal Component Analysis

جهت‌هایی برای کاهش بعد انتخاب شوند که واریانس داده‌ها در فضای جدید، بیشترین مقدار ممکن باشد. روش کلی الگوریتم Eigenface به صورت زیر است:

- انتقال همه‌ی تصاویر آموزش به فضای با بعد پایین با الگوریتم PCA
- انتقال تصویر مورد نظر برای بازشناسی چهره به بعد پایین با الگوریتم PCA
- بدست آوردن نزدیک‌ترین همسایه بین ویژگی‌های تصویر مورد نظر و تصاویر آموزش

ایرادى که در این روش وجود دارد این است که برای کاهش بعد و انجام انتقال، از هیچ برچسب کلاسی استفاده نمی‌کند. به عبارت دیگر، بدون در نظر گرفتن برچسب کلاس‌های موجود در تصویر (به عنوان مثال کلاس صورت و غیر صورت)، کاهش بعد را انجام می‌دهد و هیچ آموزشی مبتنی بر ایجاد مرز بین کلاس‌ها نمی‌بیند. همچنین اگر واریانس به دلیل یک عامل خارجی مانند نور زیاد باشد، آنگاه تبدیل انجام شده یک تبدیل بهینه نخواهد بود و اطلاعات موجود در محورهای با بیشترین واریانس، دارای اطلاعات متمایزی نیستند و خطای دسته‌بندی افزایش خواهد یافت. بنابراین برای رفع این مشکل از روش آنالیز تفکیک خطی^{۹۰} در بازشناسی چهره استفاده می‌شود (Belhumeur,

Hespanha, & Kriegman, 1997). الگوریتم Fisherfaces

تجزیه و تحلیل مؤلفه‌های اصلی که هسته روش Eigenfaces محسوب می‌شود، ترکیب خطی از ویژگی‌ها را بدست می‌آورد به طوری که واریانس کل داده‌ها بیشینه شود. با وجود اینکه روش PCA روش خوبی برای کاهش بعد داده‌هاست، اما برای کاهش بُعد برچسب کلاسی داده‌ها را در نظر نمی‌گیرد، بنابراین ممکن است که مقدار زیادی از اطلاعات در هنگام تبدیل از بین بروند. و داده‌های موجود در یک کلاس به اشتباه در چند کلاس جداگانه دسته‌بندی شوند.

روش تجزیه و تحلیل تفکیک خطی یک روش کاهش بُعد است که برای این کار برچسب کلاسی داده‌ها را نیز در نظر می‌گیرد (Fisher, 1936). در این روش، به جای بیشینه کردن واریانس همه‌ی داده‌ها، نسبت پراکندگی بین داده‌های کلاس‌ها را نسبت به پراکندگی درون هر کلاس افزایش می‌دهد. این روش در سال ۱۹۹۷ برای بازشناسی چهره مورد استفاده قرار گرفت (Belhumeur et al., 1997). ایده اصلی این روش، کمینه کردن واریانس درون کلاسی و بیشینه کردن واریانس بین کلاسی است.

^{۹۰} Linear Discriminant Analysis

در روش Fisherfaces، برای هر کلاس یک ماتریس تبدیل آموزش داده می‌شود. بنابراین میزان روشنایی در مقایسه با روش Eigenfaces، در نتیجه الگوریتم تأثیر نخواهد داشت. این الگوریتم به دلیل مقاومت در برابر تغییرات روشنایی، یکی از محبوب‌ترین و پرستفاده‌ترین الگوریتم‌های بازشناسی چهره است. نکته مهمی که می‌توان ذکر کرد این است که عملکرد این الگوریتم بسیار وابسته به داده‌های ورودی است. به عبارت دیگر، اگر Fisherfaces تنها با تصاویر با روشنایی خوب آموزش داده شده باشد و برای تست از تصاویر با روشنایی بد استفاده کنیم، الگوریتم اجزاء سازنده^{۹۱} اشتباهی را تولید خواهد کرد. زیرا آن ویژگی‌های آموزش دیده ممکن است که در تصاویر با روشنایی نامناسب دیده نشوند. و این نتیجه هم منطقی است زیرا مدل آموزش دیده هیچ فرصتی برای آموزش روشنایی نداشته است.

۵-۲-۳. تشخیص حرکت

در این پژوهش روبوداک قادر خواهد بود که حرکت دست را در دو جهت افقی (چپ به راست یا راست به چپ) و عمودی (بالا به پایین یا پایین به بالا) تشخیص دهد و براساس قوانینی که تعریف شده، براساس این تشخیص عملیاتی را انجام دهد.

برای تشخیص حرکت، از الگوریتم تشخیص رنگ، استفاده شده است.

رنگ دست برای تشخیص حرکت نمی‌تواند معیار خوبی باشد زیرا رنگ‌های متنوعی در محیط شبیه رنگ دست موجود است و همین امر عملیات تشخیص را با چالش مواجه می‌کند. بنابراین برای افزایش دقت تشخیص، از یک رنگ خاص که معمولاً در محیط دیده نمی‌شود استفاده می‌شود. لوگوی روبوداک براساس این رنگ بازطراحی شده و در دست کاربر قرار می‌گیرد (شکل ۵-۹). سیستم بینایی روبوداک با شناسایی این لوگو، رنگ آن و جهت حرکت آن، می‌تواند عملیات مورد نظر را انجام دهد.

برای شناسایی رنگ و حرکت آن اینگونه عمل می‌شود که در هر فریم از ویدئو، مکان میانگین رنگ مورد نظر (که در اینجا رنگ سبز روشن در نظر گرفته شده) که از یک اندازه‌ی مشخص بزرگ‌تر هستند، محاسبه می‌شود. سپس این مکان در فریم‌های مختلف با هم مقایسه می‌شود. با توجه به نسبت این مکان‌ها می‌توان تشخیص داد که لوگوی مورد نظر به صورت افقی حرکت داده شده است یا به صورت عمودی. اندازه‌ی مشخصی که به عنوان حد کمینه در نظر گرفته می‌شود، به منظور جلوگیری

⁹¹ Components

از تاثیر نويز در محاسبات برنامه است. همچنين برای حذف تأثير اجسامی با رنگ مورد نظر، که ممکن است در پس زمینه موجود باشند، از میانگین گیری استفاده شده که در نهایت حرکت جسم مورد نظر تعیین کننده نتیجه نهایی باشد.



شکل ۵-۹- لوگوی رنگی روبوداک برای تشخیص حرکت

مراحل اجرای الگوریتم تشخیص حرکت برای یک فریم f :

۱. تشخیص نواحی رنگی در تصویر بر اساس طیف RGB مورد نظر: $R_1^f, R_2^f, \dots, R_N^f$ (متغیر R_n^f نشان دهنده ناحیه رنگی n ام در فریم f است).

۲. باینری کردن تصویر بر اساس نواحی رنگی تشخیص داده شده

۳. حذف نواحی رنگی کوچکتر از حد آستانه از تصویر باینری با استفاده از الگوریتم

Morph. نواحی جدید باقی مانده: $R_1^{f*}, R_2^{f*}, \dots, R_k^{f*}$

۴. محاسبه میانگین مرکز نواحی رنگی در تصویر: $C^f = \frac{1}{k} \sum_{k=1}^K C_j^f$

($C^f(x, y)$ یک بردار مکانی است و متغیر C_j^f نشان دهنده مرکز ناحیه j در فریم f است).

بعد از اعمال مراحل بالا برای هر فریم، شبه کد زیر بر روی میانگین مرکزهای هر دو فریم متوالی C^f و C^{f+1} اعمال می شود. در این شبه کد، F تعداد کل فریم های دریافتی است و x_f و y_f به ترتیب مکان میانگین مرکز نواحی در فریم f در راستای x و y را نشان می دهند. خروجی حلقه اصلی این برنامه ۴ حالت ممکن است:

(۱) No_Gesture: هیچ حرکتی در برنامه تشخیص داده نشده است.

(۲) Move_badGesture: حرکت تشخیص داده شده جزء دو حرکت عمودی و یا افقی تعریف شده نبوده است.

(۳) Move_vertical: حرکت افقی تشخیص داده شده است.

۴) Move_horizontal: حرکت عمودی تشخیص داده شده است.

در نهایت برای مشخص شدن نوع حرکت، بین متغیرهای No_Gesture، Move_badGesture، Move_vertical و Move_horizontal ماکزیمم گیری می شود و متغیری که بیشترین مقدار را داشته باشد، به عنوان نوع حرکت نهایی به خروجی اعلام می شود.

شبه کد تشخیص حرکت برای F فریم متوالی

```

For i=1 : F
    If  $(x_f - x_{f+1} > \varepsilon_1)$  or  $(x_f - x_{f+1} > \varepsilon_1)$ 
        If  $(|x_f - x_{f+1}| - |y_f - y_{f+1}| > \varepsilon_2)$ 
            If  $(x_f - x_{f+1} > y_f - y_{f+1})$ 
                Move_horizontal= Move_horizontal+1
            Else if  $(x_f - x_{f+1} < y_f - y_{f+1})$ 
                Move_vertical= Move_vertical +1
            End
        Else
            Move_badGesture= Move_badGesture+1
        End
    Else
        No_Gesture= No_Gesture+1
    end
end
Gesture=Max(Move_horizontal, Move_vertical, Move_badGesture, No_Gesture)

```

۵-۳. زیر سیستم پردازش گفتار

زیر سیستم پردازش گفتار، گفتار کاربر را در قالب صوت از طریق میکروفون وایرلس دریافت می کند و آنرا به متن تبدیل می نماید. این زیر سیستم برای دو منظور استفاده می شود: دریافت فرمان های صوتی از کاربر و دریافت عبارات پرس و جوی کاربر در قالب گفتار پیوسته، برای جستجو در پایگاه داده گنج و گوگل اسکولار^{۹۲}.

پایگاه داده گنج شامل مجموعه‌ای از رکوردهای علمی سطح کشور است که عمدتاً از پایان‌نامه تشکیل شده است. برای پردازش گفتار فارسی و انگلیسی از دو برنامه جداگانه استفاده می‌شود که در ادامه هر یک معرفی می‌شوند.

۵-۳-۱. پردازش گفتار فارسی

در قسمت پردازش گفتار فارسی و به منظور تبدیل گفتار فارسی به متن، از ماژول «نویسا»^{۹۳} که توسط شرکت «عصر گویش پردازش» توسعه داده شده، استفاده شده است. صوت از طریق میکروفون به ماژول نویسا منتقل شده و مورد پردازش قرار می‌گیرد تا متن مورد نظر تولید شود. نویسا نسخه‌های متفاوتی دارد که تفاوت اصلی آن‌ها در بانک واژگان است که برای کاربران مختلف مانند پزشکان، مدیران، وکلا و کاربران عمومی اختصاصی شده‌اند. در سیستم روباتک نسخه عمومی نویسا مورد استفاده قرار گرفته است. از این ماژول می‌توان در کارهای مختلفی که نیاز به تایپ دارند مانند گزارش نویسی و نامه نگاری استفاده کرد. به علاوه، نویسا قابلیت دریافت فرمان‌های صوتی را نیز دارد. این ماژول دقت و سرعت بالایی دارد و می‌تواند با محیط‌ها و کاربران مختلف تطبیق پیدا کند. نویسا به صورت مستقل از گوینده طراحی شده است به این مفهوم که برای استفاده از آن نیازی به آموزش صدای گوینده نیست و تنها لازم است که صدای محیط طی یک فرایند آماده‌سازی اولیه توسط سیستم ثبت شود.

۵-۳-۲. پردازش گفتار انگلیسی

برای پردازش گفتار انگلیسی، از سیستم تبدیل گفتار به متن گوگل استفاده شده است که به صورت آنلاین عمل می‌کند. مزیت این سیستم استفاده از حجم عظیم داده‌های آموزشی است که باعث شده تشخیص گفتار انگلیسی با دقت بسیار بالایی انجام شود و افراد با لهجه‌های مختلف بتوانند از آن استفاده کنند.

۵-۴. زیرسیستم تبدیل متن به گفتار

به منظور تبدیل متن به گفتار از نرم افزار «گویا» استفاده شده که توسط شرکت دانش بنیان «پکتوس»^{۹۴} توسعه داده شده است. این نرم‌افزار می‌تواند فایل‌های متنی را که با فرمت .txt ذخیره شده‌اند دریافت کرده و به فایل‌های صوتی با فرمت .wav تبدیل کند. صوت تولید شده بسیار نزدیک به صدای انسان

^{۹۳} <http://www.nevisasoft.com/>

^{۹۴} <http://pactos.ir/>

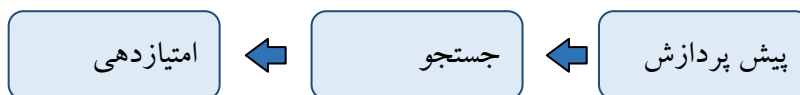
است و به علاوه می توان ویژگی های مختلف صدا مانند سرعت و زیر و بمی آن را به صورت دلخواه تغییر داد. این نرم افزار عمدتاً با هدف کمک به نابینایان توسعه داده شده، اما می توان در کاربردهای مختلف دیگری نیز از آن استفاده کرد.

۵-۵. زیرسیستم جستجو و پردازش متن

زیر سیستم جستجو و پردازش متن، عبارت فارسی یا انگلیسی را در پایگاه داده گنج یا گوگل اسکولار جستجو می کند و نتایج را بازیابی می نماید. برای بازیابی اطلاعات از گوگل اسکولار مستقیماً از جستجوی گوگل استفاده می شود و نتایج در قالب صفحات وب گوگل اسکولار نمایش داده می شود. برای بازیابی از گنج، یک الگوریتم جستجو و بازیابی اطلاعات طراحی شده و نتایج جستجو نیز براساس یک الگوی جدید به صورت مصور نمایش داده می شود. بنابراین در ادامه این الگوریتم و شیوه نمایش نتایج تشریح خواهد شد.

۵-۵-۱. جستجو و بازیابی اطلاعات در گنج

جستجو و بازیابی اطلاعات در گنج از سه مرحله اصلی تشکیل شده است (شکل ۵-۱۰): پیش پردازش عبارت پرس و جو، جستجو و بازیابی عبارت در گنج، و امتیازدهی رکوردهای گنج براساس شباهت با عبارت پرس و جو. در ادامه هریک از این مراحل تشریح می گردد.



شکل ۵-۱۰- مراحل اصلی جستجو و بازیابی اطلاعات از گنج

۵-۵-۲. پیش پردازش عبارت پرس و جو

در سیستم های بازیابی اطلاعات متنی، برای افزایش دقت بازیابی اطلاعات، معمولاً عملیات پیش پردازش روی عبارت پرس و جو انجام می شود. در این پژوهش عملیات زیر برای پیش پردازش متن بکار گرفته شده است:

- استانداردسازی عبارت پرس و جو براساس استاندارد پایگاه داده: منظور از استانداردسازی، ویرایش نگارشی عبارت پرس و جو است به گونه ای که نگارش آن متناسب با نگارش استاندارد بکار رفته در پایگاه داده باشد. در فصل بعدی عملیات پاکسازی و استانداردسازی

داده‌های پایگاه گنج تشریح خواهد شد. هر گونه عملیاتی که روی پایگاه داده برای پاکسازی و استانداردسازی داده‌ای انجام شده، لازم است که روی عبارت پرس و جو نیز انجام شود.

- ریشه‌یابی^{۹۵}: از جمله کارهای دیگری که در مرحله پیش پردازش انجام می‌شود ریشه‌یابی افعال و کلمات است. ریشه‌یابی یکی از مهم‌ترین مراحل کار در سیستم‌های پردازش متن است بخصوص اگر در این سیستم‌ها فرایند بازیابی اطلاعات انجام شود. به عنوان مثال هنگامی که در متنی از کلمه run، running و یا runs استفاده شود همگی به مفهوم دویدن اشاره می‌کنند و در نهایت در فرایند بازیابی اطلاعات می‌توان به عنوان یک مفهوم از آن‌ها استفاده کرد. در زبان فارسی نیز یک رویکرد برای ریشه‌یابی و تبدیل کلماتی مانند افعال به ریشه اصلی آنها، استفاده از بن ماضی افعال است. برای مثال افعال «رفتم»، «رفتی»، «می‌روم»، «می‌رویم»، «خواهم رفت»، «بروم» و «رفته بودم» به بن ماضی آن‌ها یعنی «رفت» تبدیل می‌شود. ریشه‌یابی نیز لازم است که علاوه بر متون پایگاه داده، روی عبارت پرس‌وجوی کاربر انجام اعمال شود. ریشه‌یابی افعال تا حد بسیار زیادی موجب افزایش دقت سیستم‌های بازیابی اطلاعات خواهد شد. بخصوص در زبان فارسی به دلیل وجود افعال بسیار زیاد از یک فعل و نگارش مختلف آن‌ها می‌تواند نرخ اصابت^{۹۶} را افزایش دهد. علاوه بر آن، نشانه‌های جمع در زبان فارسی یعنی «ات»، «ان»، و «ها» در فرایند ریشه‌یابی حذف می‌شود و کلمات جمع به مفرد تبدیل می‌شوند. به عنوان مثال کلماتی نظیر «داده‌ها» و «گزارشات» به ریشه مفرد آنها یعنی «داده» و «گزارش» تبدیل می‌شود.
- حذف کلمات اضافی (کلمات توقف^{۹۷}): از جمله کارهای دیگری که در پیش پردازش متن باید انجام شود حذف کلمات زائد است. کلمات زائد کلماتی هستند که از طرفی بسیار زیاد در متون استفاده می‌شوند و از طرفی دیگر بارمعنایی کمی را دارا هستند. برای مثال حروف اضافه مثل «را»، «از»، «اما» و «هر» کلماتی هستند که در اکثر متون یافت می‌شوند ولی ارزش معنایی ندارند. برای حذف این کلمات از فهرست‌هایی به نام Stop List استفاده می‌شود که در اکثر زبان‌ها تعداد بسیار زیادی از آن‌ها موجود است.

^{۹۵} Stemming

^{۹۶} Hit rate

^{۹۷} Stop words

- تبدیل کلمات جمع به کلمات مفرد: جمع مکسر عربی که بسیار در زبان فارسی استفاده می‌شود نیز می‌تواند به کلمه مفرد آن تبدیل گردد. توجه به این نکته ضروری است که اگرچه گاهی اوقات تبدیل جمع مکسر به مفرد کلمه می‌تواند معنای کلمه را بکلی تغییر دهد ولی از آن جا که این کار بر روی عبارت پرس‌وجو و متن پایگاه داده به صورت یکسان انجام می‌شود و هدف ما از پردازش متن، بازیابی اطلاعات است، این تغییر معنا صدمه‌ای به نتیجه بازیابی سیستم نخواهد زد. برای این منظور فهرستی از کلمات مفرد و معادل جمع مکسر آنها در برنامه قرار می‌گیرد که براساس این فهرست می‌توان این تبدیل را انجام داد. در جدول ۵-۱ تعدادی از کلمات مفرد به همراه جمع مکسرشان آورده شده است.

جدول ۵-۱- نمونه‌ای از کلمات مفرد و جمع مکسر آنها

مفرد	جمع مکسر
قصه	قصص
قلب	قلوب
لذت	لذات
مکتب	مکاتب
معدن	معادن
ملک	املاک
مرض	امراض
دوره	ادوار
رکن	ارکان

علاوه بر عملیات فوق، در این پژوهش روی عبارت پرس و جویی که توسط کاربر از طریق سیستم پردازش گفتار دریافت می‌شود، تغییر دیگری صورت می‌گیرد. کاربر برای جستجوی یک عبارت فارسی، آن عبارت را به همراه جمله «عبارت را جستجو کن» به ربات اعلام می‌کند. به عنوان مثال کاربر اعلام می‌کند که:

«عبارت روشهای پیشرفته یادگیری ماشین را جستجو کن»

این عبارت که از طریق سیستم تبدیل صوت به گفتار به متن تبدیل شده، پیش پردازش اولیه می شود تا بخش «را جستجو کن» از آن حذف شود. با اعمال سایر پیش پردازش ها، در نهایت عبارت فوق به عبارت پرس و جو زیر تبدیل می شود:

«روش پیشرفته یادگیری ماشین»

۵-۳. جستجوی عبارت

برای جستجوی عبارت پرس و جو، پس از پیش پردازش، باید این عبارت با همه فیلدهای اطلاعاتی موجود در یک رکورد علمی مقایسه شود. در این پژوهش از رکوردهای علمی پایان نامه های فارسی موجود در پایگاه گنج استفاده شده است که شرح آن در فصل بعدی ارائه خواهد شد. هر یک از رکوردهای علمی دارای سه فیلد اطلاعاتی اصلی هستند: عنوان، چکیده، و کلمات کلیدی. یکی از روش های مقایسه عبارت پرس و جو با رکوردهای یک پایگاه داده اینست که آن عبارت به کلمات تشکیل دهنده آن شکسته شود (به این عملیات Tokenize می گویند). سپس براساس مدل های زبانی تک-گرمی^{۹۸} یا دو-گرمی^{۹۹}، کلمات آن عبارت با رکوردها مقایسه شود. در اینجا از مدل زبانی تک-گرمی استفاده شده است. به این مفهوم که عبارت پرس و جو به کلمات تشکیل دهنده آن تقسیم می شود، سپس عملیات پیش پردازش روی کلمات آن صورت می گیرد.

عملیات شکستن یک متن به کلمات تشکیل دهنده آن لازم است که روی کلیه رکوردهای پایگاه داده نیز انجام شود. در نهایت کلمات هر عبارت پرس و جو با کلمات تشکیل دهنده هر رکورد اطلاعاتی مقایسه می شود و میزان شباهت یک رکورد اطلاعاتی با عبارت پرس و جو در قالب امتیاز هر رکورد اطلاعاتی برای یک عبارت پرس و جو محاسبه می گردد.

۵-۴. امتیازدهی

برای امتیازدهی و بازیابی رکوردهای مرتبط با عبارت پرس و جو از روش BM25F استفاده شده است. این روش یکی از روش های مرسوم در بازیابی اطلاعات به حساب می آید که در بسیار از موتورهای جستجوی مطرح در دنیا از جمله Lucene نیز اخیراً از این روش استفاده شده است.

روش وزن دهی BM25 (Robertson, 2010) یک الگوی وزن دهی بر اساس رویکرد احتمالی است که یک متن بر مبنای احتمال رخداد واژه های عبارت پرس و جو، رتبه بندی می شود. این روش بیشتر در

⁹⁸ Uni-gram

⁹⁹ Bi-gram

موتورهای جستجو برای رتبه دهی ارتباط عبارت پرس وجو با اسناد به کار می رود. در این روش، برای هر سند، یک امتیاز تعلق می گیرد که یک نمایش ساده برای تعریف این امتیاز براساس ارتباط بین عبارت پرس وجو Q با n کلمه (w_i) و هر سند d ، به صورت زیر است:

$$score_{BM25}(Q, d) = \sum_{i=1}^n IDF(w_i) \frac{tf(w_i, d)}{k_1 \left((1-b) + b \frac{dl_d}{avdl} \right) + tf(w_i, d)} \quad (1)$$

که $tf(w_i, d)$ فراوانی واژه w_i را در سند d نشان می دهد، و $IDF(w_i)$ معکوس فراوانی سندی^{۱۰۰} است که به صورت زیر تعریف می شود:

$$IDF(w_i) = \log \frac{N - n_i + 0.5}{n_i + 0.5} \quad (2)$$

منظور از n_i تعداد اسنادی است که واژه w_i را دارند و منظور از N تعداد کل اسناد موجود در پایگاه داده است.

عبارت dl_d در فرمول BM25 بیانگر طول یک سند d برحسب واژگان آن است و $avdl$ میانگین طول اسناد در کل پایگاه داده است. دو پارامتر k_1 و b بسته به شرایط مسئله تنظیم می شوند، البته معمولاً مقادیر $k_1 \in [1.2, 2]$ و $b \in [0.5, 0.8]$ برای آنها لحاظ می شود.

تغییرات متعددی برای بهبود روش BM25 تاکنون ارائه شده است. از جمله این تغییرات، روش BM25F است که در آن محل رخداد واژگان یک عبارت پرس وجو در یک سند نیز در محاسبه امتیاز هر سند در نظر گرفته می شود (Robertson, Zaragoza, & Taylor, 2004). به عنوان مثال رخداد یک کلمه در عنوان یک متن یا در بخش های مختلف آن می تواند نتایج متفاوتی را در بازیابی ارائه کند. در روش BM25، تمامی اسناد به صورت ساختار نیافته و بدون هیچ تمایزی در نظر گرفته می شود. اما بعضی اسناد به عنوان مثال مقالات علمی شامل بخش هایی نظیر عنوان، خلاصه، بدنه اصلی و کلمات کلیدی است. اگر یک واژه در عنوان یک سند علمی، می تواند بیانگر ارتباط معنایی بیشتر آن سند با آن واژه باشد. بنابراین در روش BM25F، با در نظر گرفتن درجه اهمیت، یا همان وزن، برای بخش های مختلف یک سند، سعی شده تا بازیابی اطلاعات بهبود پیدا کند.

¹⁰⁰ Inverse document frequency

فرض کنید که یک سند d ، دارای S بخش (stream) مختلف و هر بخش دارای طول l_s باشد. بنابراین فراوانی واژه w_i را در بخش S از سند به صورت زیر نشان می‌دهیم:

$$tf^*(w_i, d) = \sum_{s=1}^S v_s \cdot tf(w_i, d_s) \quad (۳)$$

که v_s بیانگر وزن بخش S در سند است. همچنین داریم:

$$dl_d^* = \sum_{s=1}^S v_s \cdot l_s \quad (۴)$$

بنابراین امتیاز یک سند d براساس روش BM25F به صورت زیر محاسبه می‌شود:

$$score_{BM25F}(Q, d) = \sum_{i=1}^n IDF(w_i) \cdot \frac{tf^*(w_i, d)}{k_1 \left((1-b) + b \frac{dl_d^*}{avdl} \right) + tf^*(w_i, d)} \quad (۵)$$

۵-۶. تشریح کدهای برنامه اصلی سیستم روبوداک

برنامه اصلی یا کنترلر روی لپ تاپ قرار دارد که به زبان پایتون نوشته شده است. این برنامه از چندین تابع فرعی و یک تابع اصلی تشکیل شده که در ابتدای برنامه کتابخانه‌های مورد نیاز پایتون فراخوانی می‌شود. برنامه ارتباط با سیستم پردازش صوت و تبدیل گفتار فارسی به متن نیز روی لپ تاپ قرار دارد. همچنین برنامه اتصال به سیستم پکتوس نیز داخل برنامه کنترلر اصلی قرار داده شده است. سایر کدها شامل کدهای سیستم بینایی، کد نمایش صفحه جستجو و اتصال به سرور جستجوی گنج، و کد کنترل پروژکتور روی رسپری قرار داده شده است. تابع اصلی برنامه روی رسپری، یک تابع main است که پس از تنظیم port و HOST که آدرس IP خود رسپری است، همواره در انتظار دریافت درخواست از درخواست‌کننده است.


```
if __name__ == "__main__":
    HOST, PORT = "192.168.1.55", 9998

    # Create the server, binding to localhost on port 9999
    try:
        server = SocketServer.TCPServer((HOST, PORT), MyTCPHandler)
        # Activate the server; this will keep running until you
        # interrupt the program with Ctrl-C
        server.serve_forever()
    except Exception, e:
        print type(e)
        print e
```

در ادامه توضیحاتی درباره کد برنامه بخش‌های اصلی و توابع اصلی سیستم روبوداک ارائه می‌شود. شرح تفصیلی کدها به صورت کامل در پیوست ارائه شده است.

۵-۶-۱. کنترلر اصلی

در این برنامه مجموعه‌ای از توابع فرعی طراحی شده که هر یک مربوط به فرخوانی بخشی از سیستم روبوداک هستند. علاوه بر آن از مجموعه‌ای از کتابخانه‌های پایتون نیز استفاده شده است. کلیه توابع و کتابخانه‌ها در یک برنامه اصلی یا همان کنترلر، براساس سناریوهای تعامل روبوداک با انسان فراخوانی می‌شوند. در ادامه توضیحات مربوط به هر یک ارائه می‌شود.

۵-۶-۱-۱. توابع فرعی

توابع فرعی برای وظایف مختلف مانند خواندن و یا جستجو کردن طراحی شده‌اند و بر حسب نیاز در برنامه اصلی فراخوانی می‌شوند. این توابع عبارتند از:

- record: این تابع به منظور ضبط صدای کاربر طراحی شده است. این تابع قادر است با مقایسه امواج صوتی با یک آستانه سکوت را تشخیص داده و صوت را فقط در زمان صحبت کاربر ضبط کند. پس از ضبط صدا، فایل در آدرس از پیش تعیین شده ذخیره می‌شود.
- English: از این تابع زمانی استفاده می‌شود که کاربر دستور جستجو به زبان انگلیسی را بدهد. در این تابع فایل ضبط شده در تابع record برای تبدیل شدن به متن انگلیسی به کتابخانه تبدیل گفتار به متن گوگل ارسال می‌شود.
- search: در صورت دریافت فرمان جستجو، این تابع فراخوانی می‌شود. ابتدا زبان جستجو از کاربر پرسیده می‌شود. اگر زبان مورد نظر فارسی باشد، عبارت جستجو

توسط نویسا دریافت شده و وب سرویس برای جستجو در گنج فراخوانی می‌شود. در صورت انتخاب زبان انگلیسی تابع English فراخوانی می‌شود.

- read: این تابع زمانی که کاربر دستور خواندن از روی متن را صادر کند، استفاده می‌شود. ابتدا متن مورد نظر از وب سرویس دریافت شده و سپس برای تبدیل به گفتار به برنامه پکتوس ارسال می‌شود. پس از تولید فایل، صوت توسط بلندگوی ربات پخش می‌شود.

- ConnectToGooyesh: این تابع واسطه بین برنامه اصلی و برنامه نویسا است. در این تابع یک ارتباط سوکتی بین دو برنامه ایجاد می‌شود تا از این پس نویسا بتواند دستورات را از برنامه اصلی دریافت کند.

- SendToGooyesh: پس از برقراری ارتباط بین نویسا و برنامه اصلی، از این تابع برای ارسال دستورات به نویسا استفاده می‌شود. این دستورات شامل موارد زیر است:

- Start: برای شروع تشخیص گفتار و تبدیل به متن

- Stop: برای اتمام تشخیص صوت

- Search: در هنگام دریافت این دستور توسط نویسا، چندین پیش پردازش

مانند حذف کلمات اضافی رو گفتار دریافت شده صورت می‌گیرد تا متن

تولید شده مناسب عملیات جستجو باشد.

- Pactus: توسط این تابع متن‌های کوتاه برای تبدیل به گفتار به پکتوس ارسال می‌شوند.

- setStiffnesses: این تابع سختی موتورهای ربات را فعال و غیرفعال می‌کند.

- HeadYaw: از این تابع برای حرکت عمودی سر ربات استفاده می‌شود.

- HeadPitch: این تابع برای حرکت افقی سر ربات استفاده می‌شود.

- SendToServer: این تابع زمانی فراخوانی می‌شود که نیاز با ارتباط سوکتی میان برنامه اصلی

و برنامه‌های موجود در رسپری باشد. دستورات مختلفی برای رسپری ارسال می‌شوند:

- Detection: برای تشخیص صورت

- Recognition: برای شناسایی صورت

- Gesture: برای شناسایی حرکت دست

- On و off: برای روشن و خاموش کردن پروژکتور

- Search: برای باز کردن وب سرویس و جستجوی عبارت فارسی یا انگلیسی
- Read: برای باز کردن وب سایت مربوط به وارد کردن متن برای خواندن
- knownrec: مهمترین تابع فرعی برنامه است و مربوط به زمانی می باشد که ربات کاربر را شناسایی کرده و کاربر می تواند تعاملات خود را آغاز کند. در ابتدا از کاربر درخواست می شود تا دستور خود را بیان کند. این دستورات شامل موارد زیر می باشد:
 - جستجو: زمانی که کاربر قصد جستجوی یک عبارت را دارد باید کلمه جستجو را بیان کند.
 - خواندن: زمانی که کاربر قصد خواندن از روی یک متن را دارد باید کلمه بخوان را بیان کند.
 - اتمام: زمانی که کاربر قصد باید تعامل را دارد باید عبارت تمام شد را بیان کند.
- پس از بیان یک دستور ربات منتظر دستور بعدی کاربر می ماند.
- unknownrec: اگر ربات صورتی را تشخیص داده ولی آن را شناسایی نکند، این تابع فراخوانی می شود. در این صورت ربات به سمت کاربر آمده و دوباره تلاش می کند تا صورت او را شناسایی کند.

۵-۶-۱-۲. کتابخانه های پایتون سیستم روبوداک

در برنامه اصلی سیستم روبوداک از مجموعه ای از کتابخانه های از پیش تعریف شده در زبان پایتون استفاده شده است که این کتابخانه ها عبارتند از:

- Sys: مربوط به عملیاتی است که توسط سیستم عامل قابل انجام است.
- Alproxy: دربرگیرنده کلیه توابع ربات نائو به زبان پایتون است.
- Time: مربوط به زمان سیستم عامل و زمان بین المللی است.
- Almath: دربرگیرنده کلیه توابع ریاضیاتی است.
- Os: برای دسترسی به آدرس ها و فایل ها استفاده می شود.
- Socket: برای سوکت زدن بین برنامه های مختلف استفاده می شود.
- Subprocess: برای اجرای یک برنامه در پس زمینه به صورت موازی با برنامه اصلی استفاده می شود.
- Shutil: مربوط به انجام عملیات پیشرفته بر روی فایل های سیستم است.

- Ftplib: مربوط به دسترسی به فایل های FTP است.
- Codecs: برای کد گذاری متغیرها استفاده می شود.
- Speech- recognition: دربرگیرنده توابع تشخیص گفتار است.
- Array: برای ساخت ماتریس و آرایه استفاده می شود.
- Struct: برای تبدیلات بین متغیرهای بایتی استفاده می شود.
- Pyaudio: کتابخانه صوتی پایتون است.
- Wave: برای خواندن و نوشتن فایل های با فرمت wav. استفاده می شود.
- Urllib: شامل پروتکل های دسترسی به اینترنت است.

۳-۱-۶-۵. تابع اصلی کنترلر

تابع اصلی وظیفه برقراری ارتباط بین سایر توابع و فراخوانی به موقع آن ها را برعهده دارد. در این تابع ابتدا برنامه با فراخوانی برنامه نویسا منتظر می ماند تا کاربر کلمه سلام را بیان کند. پس از دریافت این کلمه، دستور تشخیص صورت برای رسپری ارسال شده و رسپری نتیجه را باز می گرداند. اگر پاسخ مثبت باشد، مجددا دستور شناسایی ارسال شده و بر حسب پاسخ دریافت شده، توابع knownrec و unknownrec فراخوانی می شود و برنامه در این توابع ادامه پیدا می کند.

۲-۶-۵. برنامه زیرسیستم بینایی

برنامه زیر سیستم بینایی با فراخوانی یکی از توابع فرعی "Recognition"، "Detection"، یا "Gesture" فعال می شود. این برنامه روی رسپری از طریق فراخوانی تابع main روی رسپری اجرا می شود. در واقع کنترلر اصلی درخواست اتصال به زیر سیستم بینایی را به تابع main موجود در رسپری ارسال می کند و رسپری بر همین اساس توابع مربوطه را فراخوانی می نماید.

هنگامیکه دستور فراخوانی "Recognition" باشد، با دستور os.system فایل exe مربوط به بازشناسی چهره اجرا می شود. پس از پایان برنامه بازشناسی چهره، خروجی آن در فایل rec.txt ذخیره می شود که در این برنامه آن فایل را خوانده و محتوای آن را در متغیر result برای ارسال به درخواست کننده، ذخیره می کند.

```
#-----Run Predict Face Recognition-----
if RequestString=="Recognition":
    os.system(ProgramPath+"WebcamFaceRec")
    f = open(ProgramPath+'rec.txt', 'r')
    result=f.read()
```

محتوای موجود در فایل rec.txt به صورت زیر است که در سطر اول کلمه کلیدی "person:" می آید و در ادامه ی آن اسم فرد شناسایی شده نوشته می شود. در سطر دوم کلمه کلیدی "position:" می آید و در ادامه ی آن آرایه ای دو بعدی که شامل شماره پیکسل سطر و ستون مرکز صورت تشخیص داده شده است، نوشته می شود. علت ارسال مختصات مرکز صورت اینست که ربات بتواند براسا موقعیت تشخیص داده شده، سر خود را به سمت کاربر بچرخاند.

```
person : Afshar
position : [460, 122]
```

هنگامیکه دستور فراخوانی "Detect" باشد، با دستور os.system فایل exe مربوط به تشخیص صورت اجرا می شود. پس از پایان آن برنامه، خروجی آن در فایل detect.txt ذخیره می شود که در این برنامه آن فایل را خوانده و محتوای آن را در متغیر result برای ارسال به درخواست کننده، ذخیره می کند. مقدار موجود در فایل detect.txt یا true است و یا false.

```
#----- Run Face Detection -----
elif RequestString=="Detect":
    os.system(ProgramPath+"detect_face")
    detectfile = open(ProgramPath+'detect.txt', 'r')
    detectresult0=detectfile.read()
    detectresult2=detectresult0.split()
    result=detectresult2[0]
```

اگر دستور دریافت شده کلمه ی "Gesture" باشد، با دستور os.system فایل exe مربوط به تشخیص حرکت اجرا می شود. پس از پایان آن برنامه، خروجی آن در فایل GestureResult.txt ذخیره می شود که در این برنامه آن فایل را خوانده و محتوای آن را در متغیر result برای ارسال به درخواست خواه، ذخیره می کند. مقدار موجود در فایل GestureResult.txt یا کلمه No است و یا Yes.

```
#----- Run Gesture Detection -----
elif RequestString=="gesture":
    os.system(ProgramPath+"GestureRecognition")
    detectfile = open(ProgramPath+'GestureResult.txt', 'r')
    detectresult0=detectfile.read()
    detectresult2=detectresult0.split()
    result=detectresult2[0]
```

۳-۶-۵. برنامه کنترل پروژکتور

برنامه کنترل روشن یا خاموش کردن پروژکتور روی رسپری با فراخوانی تابع On و Off در کنترلر اجرا می شود. اگر دستور دریافت شده کلمه ی "off" باشد، رسپری با استفاده از کدهای نوشته شده در

زیر، برای خاموش کردن پروژکتور، به مدت ۵ ثانیه جریان را به سیم متصل به پروژکتور برقرار می‌کند.

```
#-----Turn off projector-----
elif RequestString=="off":
    GPIO.cleanup()
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(11,GPIO.OUT)

    GPIO.output(11,GPIO.HIGH)
    time.sleep(5)
    GPIO.output(11,GPIO.LOW)
    time.sleep(1)
```

اگر دستور دریافت شده کلمه‌ی “on” باشد، ابتدا با استفاده از دستور `webbrowser.open`، آدرس وب مورد نظر را باز می‌کند، سپس با استفاده از کدهای مربوطه برای روشن کردن پروژکتور، به مدت ۵ ثانیه جریان را به سیم متصل به پروژکتور برقرار می‌کند.

```
#-----Turn on projector and open Ganj URL in Browser-----
elif RequestString=="on":

    #open Ganj URL in Browser
    webbrowser.open('http://10.7.6.172/')

    #Turn on projector
    GPIO.cleanup()
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(11,GPIO.OUT)

    GPIO.output(11,GPIO.HIGH)
    time.sleep(5)
    GPIO.output(11,GPIO.LOW)
    time.sleep(1)
```

در صورتی که درخواست گرفته شده از نوع اول باشد و شامل هیچ یک از دستورات ذکر شده نباشد، خطای نامعتبر بودن درخواست در خروجی چاپ می‌شود.

```
else:
    print RequestString
    print "This input is not valid!.."
```

۵-۶-۴. برنامه اتصال به زیرسیستم تبدیل متن به گفتار

برای خواندن از روی یک متن، ابتدا دستور خواندن به روبوداک داده می‌شود سپس صفحه وب دریافت متن مورد نظر باز می‌شود و کاربر با وارد کردن متن مورد نظر، دستور خواند را ارسال می‌کند. کد این برنامه به صورت زیر است:

if "read" in text2:

```
completeURL="http://10.7.6/۱۷۲.TextToSpeech.aspx"

print completeURL
webbrowser.open(completeURL)
```

۵-۶-۵. برنامه اتصال به زیر سیستم تشخیص گفتار

برای اتصال به سیستم نویسا جهت بازشناسی گفتار، برنامه‌ای در زبان C# نوشته شده است که شرح آن در ادامه ارائه می‌شود. این برنامه جهت تبدیل گفتار به متن با استفاده از واسط برنامه‌نویسی نرم‌افزار^{۱۰۱} نویسا است که از طریق پروتکل TCP/IP با برنامه کنترلر ارتباط دارد. این برنامه پس از برقراری ارتباط با برنامه کنترلر، در حالتی منتظر ارسال دستور قرار می‌گیرد. برحسب اینکه چه دستوری از برنامه کنترلر ارسال شود، نرم‌افزار واسط در حالت‌های مختلف قرار می‌گیرد. برنامه کنترلر دستورات شروع بازشناسی، توقف بازشناسی را در مراحل مختلف به برنامه واسط ارسال می‌کند تا با فراخوانی توابع مربوط در واسط برنامه‌نویسی نرم‌افزار نویسا، فعالیت موردنظر انجام گرفته و نتیجه به برنامه کنترلر ارسال شود. علاوه بر این، این برنامه وظیفه حذف کلمات زائد از نتیجه بازشناسی را برحسب درخواست برنامه کنترلر انجام می‌دهد. در ادامه توابع اصلی این برنامه معرفی می‌شوند.

• RobotoInteraction: این تابع برای برقراری ارتباط با برنامه کنترلر است.

```
private void RobotoInteraction()
{
    IPAddress ipAddress = IPAddress.Parse(txtInterfaceIP.Text);
    int portNumber = int.Parse(txtInterfacePort.Text);
    IPEndPoint localEndPoint = new IPEndPoint(ipAddress, portNumber);
    Listener = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
        ProtocolType.Tcp);
    Listener.Bind(localEndPoint);
    Listener.Listen(MaxConnections);
    Client = Listener.Accept();
    ReceivedBuffer = new byte[MessageLength];
    AsyncCallback callback = new AsyncCallback(OnReceivedData);
    Client.BeginReceive(ReceivedBuffer, 0, MessageLength, SocketFlags.None,
        callback, null);
}
```

¹⁰¹ Software Development Kit (SDK)

- ```
IsClientConnect = true;
{
```
- **OnReceivedData:** وظیفه دریافت داده‌ها از برنامه کنترلر و انتخاب فعالیت موردنظر بسته به دستور ارسالی از برنامه کنترلر به عهده این تابع است.
  - **StartRecognition:** این تابع از توابع مربوط به واسط برنامه‌نویسی نرم‌افزار نویسا است که با هدف شروع بازشناسی گفتار فراخوانی می‌شود.
  - **StopRecognition:** این تابع جهت توقف بازشناسی گفتار و از توابع مربوط به واسط برنامه‌نویسی نرم‌افزار نویسا است.
- کدهای برنامه مربوط به سه تابع فوق در ادامه آمده است.

```
private void OnReceivedData(IAsyncResult ar)
{
 int TextLength = 0;
 for (int i = 0; i < ReceivedBuffer.Length; i++)
 {
 if (ReceivedBuffer[i] != 0)
 TextLength++;
 }
 byte[] Text = new byte[TextLength];
 Array.Copy(ReceivedBuffer, 0, Text, 0, TextLength);
 string Message = Encoding.ASCII.GetString(Text);
 Message = Message.Trim();
 if (Message == "startt")
 {
 ProcessRecognizedText = false;
 SimpleNevisaInterface.StartRecognition();
 NevisaInterface.NewSpeechHypothesis+=
 NevisaInterface_NewSpeechHypothesis;

 NevisaInterface.DoTransferHypothesis+=NevisaInterface_DoTransferHypothesis;
 {
 else if (Message == "stoppp")
 {
 if (ProcessRecognizedText)
 ProcessRecognizedText = false;
 SimpleNevisaInterface.StopRecognition();
 }
 else if (Message == "search")
 {
 SimpleNevisaInterface.StartRecognition();
 }
 }
 }
}
```



```

 ProcessRecognizedText = true;
 {
 rtxtReceivedData.Invoke((MethodInvoker)delegate
 {
 rtxtReceivedData.AppendText(Message + "\r\n");
 rtxtReceivedData.ScrollToCaret();
 });
 AsyncCallback callback = new AsyncCallback(OnReceivedData);
 Client.BeginReceive(ReceivedBuffer, 0, MessageLength, SocketFlags.None,
 callback, null);
 }

```

- InitializeStemmingTools: این تابع جهت راه‌اندازی اولیه بخش حذف کلمات زائد است.

```

private void InitializeStemmingTools()
{
 string RootWordsPath = @"D:\Nevisa01\Nevisa01\Res\root words.txt"; //root words
 st = new StemmingTools(RootWordsPath);
 stoptexts = System.IO.File.ReadAllText(@"D:\Nevisa01\Nevisa01\StopWord.txt");
 beforeWords = stoptexts.ToLower().Split(splitChars).ToList<string>();
 stop = beforeWords.ToArray();
}

```

#### ۵-۶-۶. برنامه اتصال به سیستم جستجو و بازیابی اطلاعات

کاربر می‌تواند یک عبارت فارسی را در گنج جستجو کند یا یک عبارت انگلیسی را در گوگل اسکولار. کنترلر اصلی با دریافت صوت از کاربر، آن را به متن تبدیل می‌کند و دستور را برای رسپری ارسال می‌کند.

در تابع جستجو در رسپری، در صورتی که کلمه “searchfarsi” فراخوانی شود، ابتدا متن دریافت شده با نشانگر کاما جداسازی می‌شود. بخش اول دستور همان کلمه “searchfarsi” است و بخش دوم شامل رشته‌ای به زبان فارسی است که باید در گنج جستجو شود.

بخش دوم درخواست بعد از اینکه با استفاده از تابع url\_fix فضای خالی آن کدگذاری شد، در ادامه آدرس وب جستجو در گنج اضافه می‌شود و سپس با استفاده از دستور webbrowser.open، صفحه وب مربوطه باز می‌شود. با این دستور در حقیقت یک درخواست به سرور جستجوی گنج از طریق وب ارسال می‌شود.

```

else:
 textsplited=RequestString.split(",")
 searchString=textsplited[1]
 completeURL="http://10.7.6.172/GanjSearch.aspx?q=%s/" %url_fix(searchString)
 print completeURL
 webbrowser.open(completeURL)

```

در صورتی که کلمه “searchengilish” در رشته درخواست گرفته شده موجود باشد، ابتدا درخواست گرفته شده با نشانگر کاما جداسازی می‌شود. بخش اول دستور همان کلمه “searchengilish” است و بخش دوم شامل رشته‌ای به زبان انگلیسی است که باید در گوگل اسکولار جستجو شود. بخش دوم درخواست بعد از اینکه با استفاده از تابع url\_fix فضای خالی آن کدگذاری شد، در ادامه آدرس وب مربوط به گوگل اسکولار اضافه می‌شود و در نهایت با استفاده از دستور webbrowser.open، صفحه وب مربوطه باز می‌شود. در انتها هم متغیر result به عنوان پاسخ به درخواست کننده ارسال می‌شود.

if "searchengilish" in text2:

```

 print "searchengilish"-----
 textsplited=text2.split(",")
 searchString=textsplited[1]
 completeURL="https://scholar.google.com/scholar?q=%s "

```

%url\_fix(searchString (

```

 print completeURL
 webbrowser.open(completeURL)

```

self.request.sendall(result)

در صورت ایجاد مشکل در ارتباط با درخواست کننده استثنا<sup>۱۰۲</sup> ایجاد شده در خروجی نمایش داده می‌شود.

```

except Exception, e:
 print type(e)
 print e

```

---

<sup>102</sup> Exception

#### ۵-۶-۷. برنامه سیستم جستجو و بازیابی اطلاعات

این برنامه با هدف جستجو در بین مستندات فارسی در پایگاه داده گنج توسعه داده شده است. کارکردهای اصلی این برنامه عبارتند از:

- پیش‌پردازش عبارت پرس‌وجو: عبارت پرس‌وجو براساس قوانین پاک‌سازی داده‌ها پیش‌پردازش می‌شود.

- جستجو و بازیابی اطلاعات: جستجو و امتیازدهی رکوردها براساس الگوریتم BM25F.

در ادامه توابع اصلی که برای پیاده‌سازی الگوریتم BM25 طراحی شده، معرفی می‌شوند.

- SearchByBM25: این تابع، تابع اصلی برنامه جستجو است که در آن توابع پیش‌پردازش و محاسبه امتیاز هر رکورد براساس الگوریتم BM25F قرار دارد. در این تابع با فراخوانی توابع فرعی، در ۱۰۰ رکورد با بیشترین امتیاز براساس امتیاز مرتب‌سازی و نمایش داده می‌شوند.
- CalculateArticleLength: براساس فرمول محاسبه امتیاز هر رکورد در BM25F، طول هر کدام از رکوردهای گنج در این تابع محاسبه می‌شود. مجموع طول عنوان، چکیده و کلمات کلیدی، طول هر رکورد را نشان می‌دهد.
- CalculateTF: مقدار فراوانی هر کلمه از عبارت پرس‌وجو در الگوریتم BM25 را در هر رکورد محاسبه می‌کند که در آن تابع OccurrencesCount تعداد تکرار کلمه مورد نظر را در یک رشته تعیین می‌کند. این تابع برای مشخص کردن وجود یک کلمه کلیدی و تعداد تکرار آن در هر یک از عنوان، چکیده و یا کلمات کلیدی مقاله می‌باشد.

کلیه کدهای مربوطه در پیوست آمده است.

## **فصل ششم:**

# **ارزیابی عملکرد سیستم روبوداک و**

## **اجزای آن**

## ۶. ارزیابی عملکرد سیستم روباتیک و اجزای آن

در این فصل، عملکرد سیستم روباتیک مورد ارزیابی قرار می‌گیرد. سیستم روباتیک براساس تعاملات طراحی در فصل پنجم که شرح آنها در نمودارهای فعالیت آمده است، به درستی با کاربر ارتباط برقرار می‌کند. منطق و توالی عملیات تعامل روباتیک با یک فرد کاملاً براساس همان نمودارهای فعالیت برنامه‌ریزی و اجرا شده است. بنابراین عملکرد کل سیستم براساس همان نمودارها و تعاملاتی که تعریف شده، به درستی صورت می‌گیرد.

بنابراین آنچه که از حیث ارزیابی عملکرد روباتیک نیاز به بررسی دارد، عملکرد هر یک از اجزای سیستم روباتیک به تنهایی است. زیرا تعاملات روباتیک به عنوعی طراحی شده که در صورتیکه هر یک از زیرسیستم‌های بینایی، تشخیص صوت، جستجو و بازیابی اطلاعات به درستی عمل نکنند، روباتیک بتواند تعامل خود را با کاربر ادامه دهد.

به عنوان مثال، برای عملیات تشخیص چهره، امکان اشتباه روباتیک در تشخیص چهره در طراحی تعاملات دیده شده است. در واقع اگر روباتیک بازخورد تشخیص نادرست فرد را دریافت کند، معذرت خواهی می‌کند و دوباره سعی می‌کند که وی را شناسایی کند.

به همین منظور در این بخش عملکرد هر یک از زیرسیستم‌های روباتیک مورد بررسی قرار می‌گیرد و نتایج پیاده‌سازی و ارزیابی آنها ارائه می‌گردد.

### ۶-۱. ارزیابی زیرسیستم بینایی

برای آزمایش و ارزیابی زیر سیستم بینایی از کتابخانه OpenCV استفاده شده است. کتابخانه Opencv، یک کتابخانه‌ی بینایی ماشین است که از سال ۱۹۹۹ شروع به کار کرده است. تمرکز این کتابخانه بر روی پردازش برخط<sup>۱۰۳</sup> تصاویر و همچنین پیاده‌سازی الگوریتم‌های بینایی ماشین است. این کتابخانه شامل رابط‌های برنامه‌نویسی به زبان‌های Python، C++ و C است و به طور گسترده در پروژه‌های دانشگاهی و محصولات تجاری مورد استفاده قرار می‌گیرد.

برای آموزش دسته‌بندی کننده تشخیص صورت، نیاز به هزاران تصویر صورت از افراد مختلف، در شرایط مختلف است که جمع‌آوری این تعداد داده کار آسانی نیست. با توجه به پژوهش‌های زیادی که در این زمینه توسط محققین انجام شده، مجموعه‌های استاندارد از این تصاویر تهیه شده و در دسترس همگان قرار گرفته است. در کتابخانه OpenCV، دسته‌بندهای آموزش دیده شده با استفاده از ویژگی‌های استخراج شده از این تصاویر برای دو الگوریتم Haar و LBP در دو فایل haarcascade\_frontalface\_default.xml و lbpcascade\_frontalface.xml موجود هستند. در نسخه‌های ۲,۴ به بعد، کلاس FaceRecognizer برای بازشناسی چهره اضافه شده است.

### ۶-۱-۱. داده‌های آموزش و تست

برای آموزش مدل بازشناسی چهره، لازم است مجموعه تصاویری از صورت تمام اشخاصی که باید بازشناسی شوند، وجود داشته باشد. در حالت کلی، برای آموزش و تست یک الگوریتم و مقایسه آن با دیگر روش‌ها از مجموعه داده‌های استاندارد و آماده‌ای مانند مجموعه‌ی موجود در وبسایت Face Recognition<sup>۱۰۴</sup> استفاده می‌شود. در این وبسایت مجموعه‌های مختلفی وجود دارد که می‌توان به AT&T Facedatabase، Yale Facedatabase A و Extended Yale Facedatabase B اشاره کرد. از ویژگی‌های تصاویر موجود در این پایگاه داده‌ها می‌توان به شرایط متفاوتی مانند تغییر روشنایی محیط، تغییرات نوری متفاوت (تابش نور از راست، چپ و مستقیم)، حالات متفاوت صورت (خوشحال، نرمال، غمگین، خواب آلوده، شگفت‌زده، چشم باز یا بسته، چشمک زدن و ...) و جزئیات صورت (عینک داشتن و یا نداشتن) در گرفتن تصاویر اعمال شده اشاره کرد. شکل ۶-۱ تا شکل ۶-۳ نمونه‌هایی از تصاویر موجود در این پایگاه داده‌های استاندارد را نشان می‌دهند.

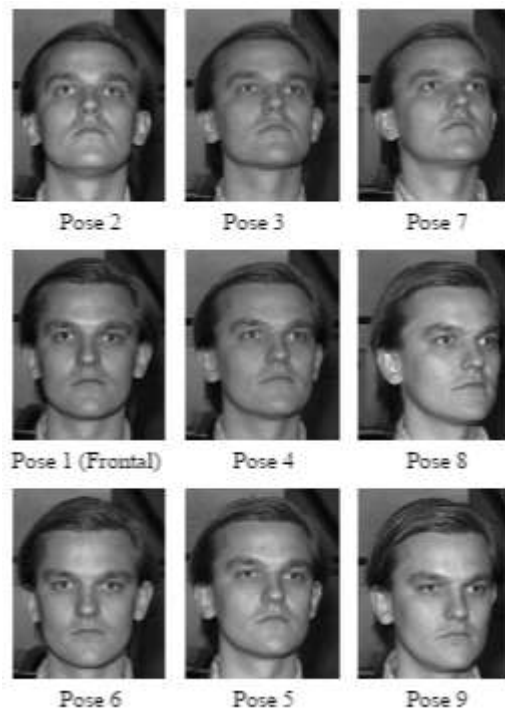


شکل ۶-۱- نمونه‌ای از تصاویر پایگاه داده AT&T Facedatabase

<sup>104</sup> <http://face-rec.org/>



۲-۶- نمونه ای از تصاویر پایگاه داده Extended Yale Facedatabase B با تابش های نوری متفاوت



۳-۶- نمونه ای از تصاویر پایگاه داده Extended Yale Facedatabase B با جهت های مختلف

در این پژوهش سعی شده است که شرایط استاندارد داده های آموزشی رعایت شود. این تصاویر برای ۳ نفر (۱ مرد و ۲ زن)، همراه با تصاویری از پس زمینه، با جهات و تابش نور متفاوت تولید شده اند. برای هر نفر ۸۰ تصویر با اندازه  $۴۸۰ \times ۶۴۰$  گرفته شده، که نمونه هایی از این تصاویر برای آموزش، در شکل ۴-۶ نشان داده شده است.



شکل ۶-۴- نمونه هایی از تصاویر استفاده شده در این پژوهش

#### ۶-۱-۲. ارزیابی الگوریتم های تشخیص صورت

برای تعیین بهترین الگوریتم تشخیص صورت در شرایط آزمایشگاهی که این پژوهش در آن انجام شده، آزمایش های مختلفی در سه موقعیت از اتاق با شرایط نوری متفاوت انجام شده است که نتایج آن ها در جدول های ۶-۱ تا ۶-۴ آمده است. شرح این سه موقعیت در ادامه بیان شده است:

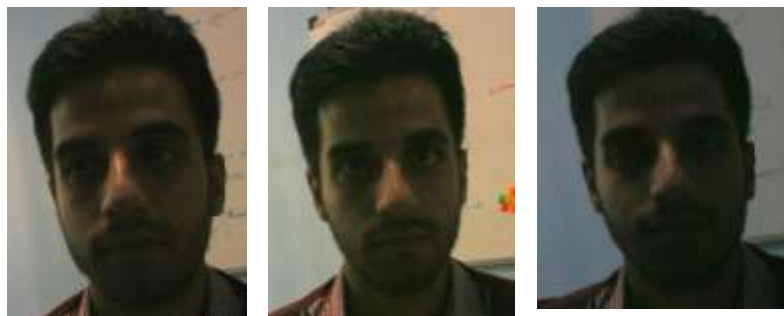
- موقعیت ۱: با تابش نور از سمت بالا و سمت چپ به صورت و پس زمینه بدون نور.
- موقعیت ۲: با تابش نور پس زمینه که باعث تاریکی در تصویر صورت می شود.
- موقعیت ۳: با تابش نور از سمت بالا و راست تصویر.

نمونه هایی از تصاویر گرفته شده از این سه موقعیت در شکل ۶-۵ تا شکل ۶-۷ نشان داده شده اند.

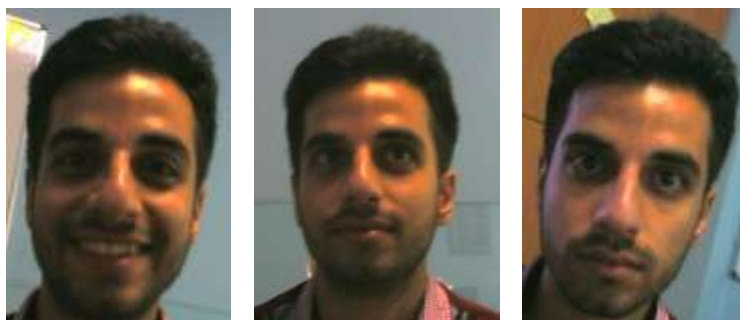


شکل ۶-۵- نمونه هایی از تصاویر موقعیت ۱





شکل ۶-۶- نمونه هایی از تصاویر موقعیت ۲



شکل ۶-۷- نمونه هایی از تصاویر موقعیت ۳

صورت انسان ویژگی‌های مختلفی دارد که چشم از جمله مهم‌ترین آن‌هاست. به همین دلیل در کاربردهایی که در رابطه با تشخیص صورت و بازشناسی چهره هستند، برای افزایش دقت برنامه از دسته‌بندی کننده‌های تشخیص چشم نیز کمک گرفته می‌شود. در این پژوهش سه نوع دسته‌بندی کننده برای تشخیص چشم مورد آزمایش قرار گرفته است که فایل‌های موجود در کتابخانه OpenCV برای دسته‌بندی کننده‌ها عبارتند از:

- دسته‌بندی کننده اول (split):

haarcascade\_righteye\_2splits.xml و haarcascade\_lefteye\_2splits.xml

- دسته‌بندی کننده دوم (mcs):

haarcascade\_mcs\_righteye.xml و haarcascade\_mcs\_lefteye.xml

- دسته‌بندی کننده سوم (tree):

haarcascade\_eye\_tree\_eyeglasses.xml و haarcascade\_eye.xml

عملکرد تشخیص صورت و چشم در تصاویر گرفته شده در موقعیت ۱ با ۸۴۱ تصویر در جدول ۶-۱ نشان داده شده‌اند.

جدول ۱-۶- نتایج تشخیص صورت و تشخیص چشم با تصاویر گرفته شده در موقعیت ۱

| تعداد تشخیص هر دو چشم به طور همزمان | تعداد تشخیص چشم راست | تعداد تشخیص چشم چپ | تعداد تشخیص صورت | الگوریتم دسته بندی کننده تشخیص چشم | الگوریتم دسته بندی کننده تشخیص صورت |
|-------------------------------------|----------------------|--------------------|------------------|------------------------------------|-------------------------------------|
| ۱۱۳                                 | ۱۳۱                  | ۳۹۷                | ۷۹۵              | splite                             | LBP                                 |
| ۴۷۰                                 | ۴۸۳                  | ۴۸۹                | ۷۹۵              | mcs                                |                                     |
| ۶۰۸                                 | ۶۵۱                  | ۶۷۵                | ۷۹۵              | tree                               |                                     |
| ۱۲۰                                 | ۱۴۰                  | ۳۳۳                | ۷۷۲              | splite                             | Haar                                |
| ۴۶۲                                 | ۴۷۱                  | ۴۹۱                | ۷۷۲              | mcs                                |                                     |
| ۵۳۹                                 | ۵۶۶                  | ۶۴۵                | ۷۷۲              | tree                               |                                     |

عملکرد تشخیص صورت و چشم در تصاویر گرفته شده در موقعیت ۲ با ۱۰۰۸ تصویر در جدول ۲-۶ نشان داده شده‌اند.

جدول ۲-۶- نتایج تشخیص صورت و تشخیص چشم با تصاویر گرفته شده در موقعیت ۲

| تعداد تشخیص هر دو چشم به طور همزمان | تعداد تشخیص چشم راست | تعداد تشخیص چشم چپ | تعداد تشخیص صورت | الگوریتم دسته بندی کننده تشخیص چشم | الگوریتم دسته بندی کننده تشخیص صورت |
|-------------------------------------|----------------------|--------------------|------------------|------------------------------------|-------------------------------------|
| ۶۵                                  | ۱۳۱                  | ۲۳۰                | ۹۴۹              | splite                             | LBP                                 |
| ۴۹۲                                 | ۵۳۴                  | ۵۲۸                | ۹۴۹              | mcs                                |                                     |
| ۷۰۷                                 | ۸۱۵                  | ۷۷۱                | ۹۴۹              | tree                               |                                     |
| ۳۵                                  | ۵۳                   | ۱۹۵                | ۸۱۴              | splite                             | Haar                                |
| ۳۷۳                                 | ۴۲۹                  | ۴۱۱                | ۸۱۴              | mcs                                |                                     |
| ۵۰۲                                 | ۶۸۸                  | ۵۵۷                | ۸۱۴              | tree                               |                                     |

عملکرد تشخیص صورت و چشم در تصاویر گرفته شده در موقعیت ۳ با ۳۹۶ تصویر در جدول ۳-۶ نشان داده شده‌اند.

جدول ۳-۶- نتایج تشخیص صورت و تشخیص چشم با تصاویر گرفته شده در موقعیت ۳

| تعداد تشخیص هر دو چشم به طور همزمان | تعداد تشخیص چشم راست | تعداد تشخیص چشم چپ | تعداد تشخیص صورت | الگوریتم دسته بندی کننده تشخیص چشم | الگوریتم دسته بندی کننده تشخیص صورت |
|-------------------------------------|----------------------|--------------------|------------------|------------------------------------|-------------------------------------|
| ۱۱۶                                 | ۱۹۴                  | ۱۷۵                | ۳۴۶              | splite                             | LBP                                 |
| ۲۴۸                                 | ۲۷۳                  | ۲۵۵                | ۳۴۶              | mcs                                |                                     |
| ۲۷۲                                 | ۳۰۹                  | ۲۹۳                | ۳۴۶              | tree                               |                                     |
| ۸۹                                  | ۱۹۲                  | ۱۳۲                | ۲۹۶              | splite                             | Haar                                |
| ۲۲۸                                 | ۲۳۸                  | ۲۳۶                | ۲۹۶              | mcs                                |                                     |
| ۲۵۵                                 | ۲۷۲                  | ۲۷۱                | ۲۹۶              | tree                               |                                     |

با توجه به نتایج بدست آمده بر روی تصاویر هر سه موقعیت ذکر شده، مشاهده می‌شود که عملکرد دسته‌بندی کننده LBP برای تشخیص صورت در شرایط آزمایشگاهی این پژوهش، از دسته‌بندی کننده Haar بهتر است. به همین دلیل آزمایش‌های بازشناسی چهره را با استفاده از دسته‌بندی کننده LBP ادامه می‌دهیم. همچنین با مقایسه‌ی عملکرد دسته‌بندی کننده‌های تشخیص چشم مشاهده می‌شود که عملکرد دسته‌بند tree بهتر از دسته‌بندی کننده‌های splite و mcs است.

پس از جداسازی بخش صورت از تصاویر مربوطه، تصویر چهره به الگوریتم‌های پیش‌پردازنده ارسال می‌شود تا اندازه و شدت نور و روشنایی تصاویر استاندارد شوند. تصاویر نشان داده شده در شکل ۶-۸ نمونه‌ای از تصاویر پیش‌پردازش شده هستند که برای آموزش و تست سیستم بازشناسی چهره مورد استفاده قرار می‌گیرند.



شکل ۶-۸- نمونه‌ای از تصاویر پیش‌پردازش شده برای آموزش و تست

### ۶-۱-۳. ارزیابی الگوریتم‌های بازشناسی چهره

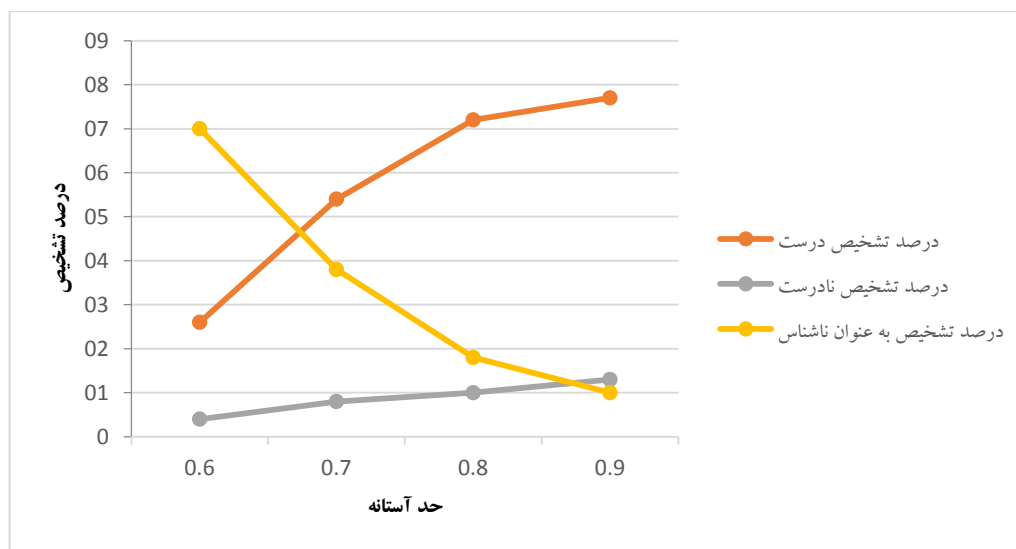
نتیجه‌ی اجرای یک برنامه‌ی بازشناسی چهره سه حالت «ناشناس»، «درست» و «نادرست» می‌تواند باشد. اگر ضریب تشابه تصویر صورت دریافت شده به تصاویر آموزش از حد آستانه کم‌تر باشد، تشخیص به ناشناس بودن فرد می‌دهد در غیر این صورت نام یکی از افرادی که در مجموعه آموزش قرار دارد و تصویر گرفته شده به تصویر آن مشابه‌تر است را به عنوان خروجی اعلام می‌کند. اگر نام فرد گفته شده درست باشد، تشخیص را با برچسب «درست» علامت می‌زنیم و در غیر این صورت، با برچسب «نادرست» علامت زده می‌شود.

نتایج بدست آمده برای بازشناسی چهره بر روی تصاویر موقعیت ۱ با استفاده از دسته‌بندی کننده LBP برای تشخیص صورت و دسته‌بندی کننده Fisherfaces برای بازشناسی چهره با حد آستانه ۰/۶ تا ۰/۹ در جدول ۶-۴ و شکل ۶-۹ نشان داده شده است. همچنین برای تصاویر موقعیت ۲ در جدول ۶-۵ و شکل ۶-۱۰ آمده است. با توجه به نتایج بدست آمده، مشاهده می‌شود که هر چه حد آستانه بیشتر شود، درصد تشخیص درست بیشتر می‌شود. اما درصد تشخیص نادرست نیز بیشتر می‌شود. در واقع با

توجه به نوع استفاده، باید یک تعادلی بین میزان اهمیت تشخیص نادرست و میزان اهمیت تشخیص ناشناس قرار داده شود. هرگاه تشخیص ناشناس در اولویت بیشتری نسبت به تشخیص نادرست باشد، حد آستانه را کاهش می‌دهیم.

جدول ۶-۴- نتایج بازشناسی چهره براساس الگوریتم Fisherfaces، با استفاده از الگوریتم تشخیص صورت LBP، روی تصاویر موقعیت ۱ برای حد آستانه‌های مختلف

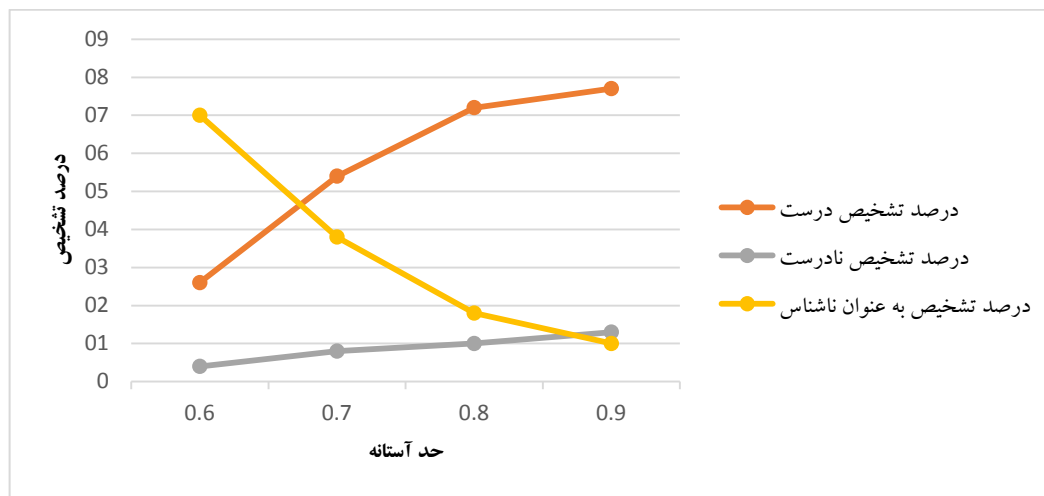
| حد آستانه       | ۰/۶ | ۰/۷ | ۰/۸ | ۰/۹ | درصد بازشناسی |
|-----------------|-----|-----|-----|-----|---------------|
| درست            | ۱۳  | ۵۵  | ۶۳  | ۶۷  |               |
| نادرست          | ۴   | ۱۹  | ۲۳  | ۲۳  |               |
| به عنوان ناشناس | ۸۳  | ۲۶  | ۱۴  | ۱۰  |               |



شکل ۶-۹- نمایش نموداری نتایج بازشناسی چهره براساس الگوریتم Fisherfaces، با استفاده از الگوریتم تشخیص صورت LBP، روی تصاویر موقعیت ۱ برای حد آستانه ۰/۶ تا ۰/۹

جدول ۶-۵- نتایج بازشناسی چهره براساس الگوریتم Fisherfaces، با استفاده از الگوریتم تشخیص صورت LBP، روی تصاویر موقعیت ۲ برای حد آستانه‌های مختلف

| حد آستانه       | ۰/۶ | ۰/۷ | ۰/۸ | ۰/۹ | درصد بازشناسی |
|-----------------|-----|-----|-----|-----|---------------|
| درست            | ۲۶  | ۵۴  | ۷۲  | ۷۷  |               |
| نادرست          | ۴   | ۸   | ۱۰  | ۱۳  |               |
| به عنوان ناشناس | ۷۰  | ۳۸  | ۱۸  | ۱۰  |               |

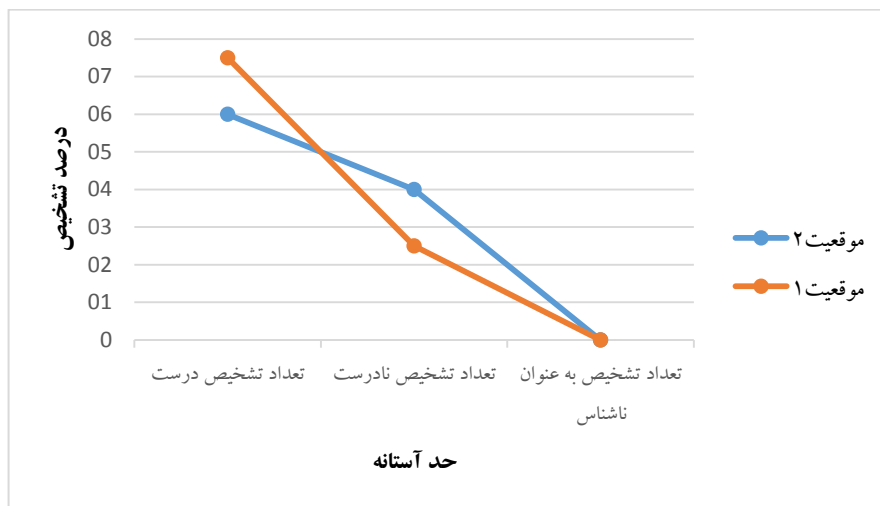


شکل ۶-۱۰ - نمایش نموداری نتایج بازشناسی چهره براساس الگوریتم Fisherfaces، با استفاده از الگوریتم تشخیص صورت LBP، روی تصاویر موقعیت ۲ برای حد آستانه ۰/۶ تا ۰/۹

نتایج بازشناسی چهره با الگوریتم Eigenface برای موقعیت ۱ و ۲ در جدول ۶-۶ و شکل ۶-۱۰ نشان داده شده است. نتایج بدست آمده نشان می‌دهد که استفاده از الگوریتم FisherFaces برای این پژوهش، مخصوصاً برای تصاویر با شرایط موقعیت ۲ بهتر است.

جدول ۶-۶ - نتایج بازشناسی چهره بر روی تصاویر موقعیت ۱ و موقعیت ۲ با الگوریتم Eigenface و حد آستانه ۰/۵

| موقعیت ۲ | موقعیت ۱ | موقعیت تصویر    |
|----------|----------|-----------------|
|          |          | درصد بازشناسی   |
| ۶۰       | ۷۵       | درست            |
| ۴۰       | ۲۵       | نادرست          |
| ۰        | ۰        | به عنوان ناشناس |



شکل ۶-۱۰- نمایش نموداری نتایج بازشناسی چهره بر روی تصاویر موقعیت ۱ و ۲ با الگوریتم بازشناسی چهره Eigenface با حد آستانه ۰/۵

#### ۴-۱-۶- ارزیابی الگوریتم تشخیص حرکت

در الگوریتم تشخیص حرکت استفاده شده، ۴ پارامتر وجود دارد:

- پارامتر مقدار کمینه برای اندازه جسم رنگی در محیط: این پارامتر متناسب با محیط و وجود و عدم وجود جسم همرنگ لوگو و اندازه و فاصله‌ی لوگو تا دوربین تنظیم می‌شود. در واقع پارامتری برای تعریف اندازه نویز در محیط است.
- پارامتر سرعت حرکت لوگو: لوگو را باید آهسته‌تر از سرعت دریافت فریم‌ها توسط دوربین، حرکت داد.
- پارامتر  $\varepsilon_1$ : یک حد کمینه برای میزان تغییرات پیکسلی حرکت لوگو، برای قبول کردن انجام یک حرکت است.
- پارامتر  $\varepsilon_2$ : یک حد کمینه برای مقایسه میزان اختلاف حرکت در جهت افقی و عمودی است. اگر این اختلاف کم باشد، به طور قطع نمی‌توان گفت که حرکت افقی و یا عمودی رخ داده است و به نوعی حرکت مورب تعبیر می‌شود.

شرایط مختلف محیطی می‌تواند در عملکرد الگوریتم تاثیرگذار باشد. به طور دقیق‌تر، وجود یا عدم وجود جسم همرنگ در محیط و جهت حرکت آن می‌تواند عملکرد الگوریتم را تحت تاثیر قرار دهد. بنابراین برای ارزیابی عملکرد الگوریتم، شرایط زیر را بررسی می‌کنیم:

(۱) فرض: حرکت لوگو در محیط و عدم وجود جسم همرنگ لوگو در محیط

(۱-۱) تشخیص صحیح حرکت: فاصله و اندازه لوگو به گونه‌ای باشد که اندازه تصویر لوگو از حد آستانه بزرگ‌تر باشد.

(۲-۱) تشخیص عدم حرکت: فاصله و اندازه لوگو به گونه‌ای باشد که اندازه تصویر لوگو از حد آستانه کوچک‌تر باشد.

(۲) فرض: حرکت لوگو در محیط و وجود جسم ثابت هم‌رنگ لوگو در محیط

(۱-۲) تشخیص صحیح حرکت: اندازه جسم در تصویر از حد آستانه کوچک‌تر باشد.

براساس الگوریتم، جسم از تصویر محیط حذف می‌شود

(۲-۲) تشخیص صحیح حرکت: اندازه جسم در تصویر از حد آستانه بزرگ‌تر باشد. چون

جسم ثابت است، تاثیری در مسیر حرکت میانگین مرکزهای نواحی رنگی نخواهد

داشت. در نتیجه مسیر حرکت میانگین مرکزهای نواحی رنگی در راستای حرکت

لوگو خواهد بود.

(۳) فرض: حرکت لوگو در محیط وجود جسم متحرک هم‌رنگ لوگو

(۱-۳) تشخیص حرکت صحیح: اندازه جسم در تصویر از حد آستانه کوچک‌تر باشد.

براساس الگوریتم، جسم از تصویر محیط حذف می‌شود.

(۲-۳) تشخیص حرکت صحیح: اندازه جسم رنگی در تصویر از حد آستانه بزرگ‌تر باشد و

در راستای حرکت لوگو حرکت کند.

(۳-۳) تشخیص حرکت صحیح: اندازه جسم رنگی در تصویر از حد آستانه بزرگ‌تر باشد

ولی دامنه حرکت آن از حرکت لوگو کمتر باشد.

(۴-۳) تشخیص حرکت نامشخص: اندازه جسم رنگی در تصویر از حد آستانه بزرگ‌تر

باشد، دامنه حرکت آن از حرکت لوگو بیشتر باشد ولی در راستای حرکت لوگو

حرکت نکند.

در شکل ۱۱-۶ نمونه‌هایی از مراحل انجام الگوریتم تشخیص حرکت بر اساس رنگ لوگو ی

روباتک، نشان داده شده است. در این تصاویر نحوه غلبه الگوریتم تشخیص حرکت استفاده شده، بر

وجود رنگ‌های مشابه ثابت در محیط، نشان داده شده است. نقطه قرمز رنگ نشان دهنده ی میانگین

محاسبه شده برای مرکز اجسام رنگی تشخیص داده شده است. همانگونه که دیده می‌شود، با حرکت

لوگو به سمت پایین، میانگین گرفته شده از مرکز اجسام رنگی تشخیص داده شده نیز به سمت پایین حرکت کرده و در نتیجه وجود جسم ثابت هم‌رنگ لوگو تأثیری در نتیجه الگوریتم ندارد.

#### ۶-۲. ارزیابی زیرسیستم تشخیص گفتار

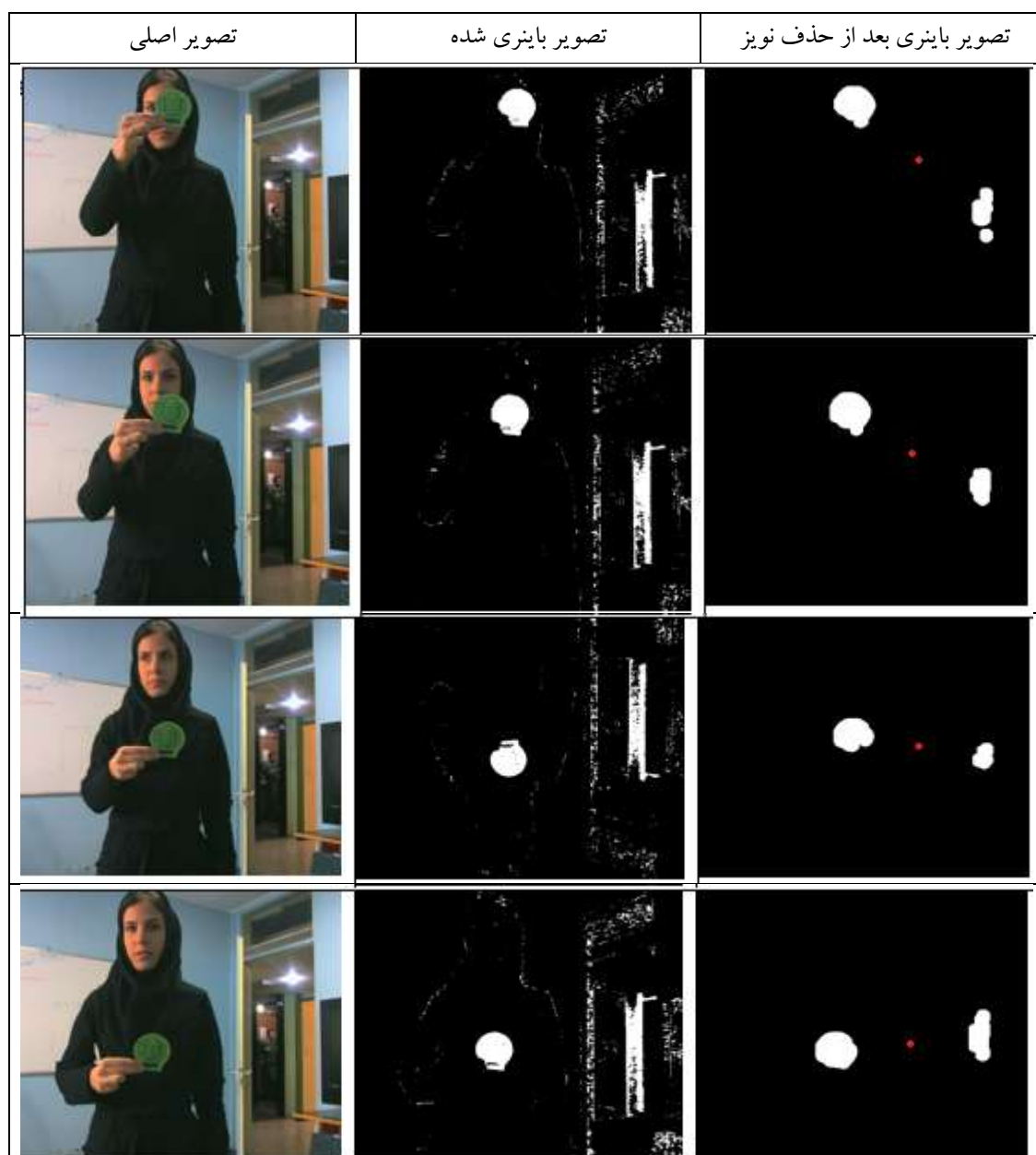
برای ارزیابی عملکرد زیرسیستم تشخیص گفتار در روبوداک، چندین عبارت به سیستم از طریق میکروفون وایرلس ارسال شده است. از آنجائیکه تعاملات صوتی با روبوداک به صورت دستوری و گفتاری است، عملکرد زیرسیستم تشخیص گفتار نیز با عبارات دستوری و گفتاری مورد بررسی قرار گرفته است.

برای این منظور عباراتی در دسته‌های زیر به این زیرسیستم ارسال شده که نتایج آن در جدول ۶-۶ آمده است:

- عبارات تک واژه‌ای
- عبارات تا ۵ واژه
- عبارات بیش از ۵ واژه

برای هر یک از این عبارات عملکرد زیرسیستم براساس درصد واژگان مشترک بین عبارت اصلی و متنی که زیرسیستم تشخیص داده است در جدول زیر آمده است. همانطور که مشاهده می‌شود، بدلیل اینکه برای افزایش دقت سیستم تشخیص گفتار، از مدل‌های آماری استفاده شده، این سیستم در تشخیص عبارات طولانی بهتر عمل می‌کند.





شکل ۶-۱۱- نمونه ای از حذف نویز موجود در تصویر و عدم تاثیر اجسام ثابت هم رنگ لوگو در الگوریتم تشخیص حرکت

جدول ۶-۷- نتایج ارزیابی زیرسیستم تشخیص گفتار روبوداک

| عبارت اصلی                | خروجی سیستم تشخیص گفتار   | درصد واژگان مشترک |
|---------------------------|---------------------------|-------------------|
| سلام                      | سلام                      | ۱۰۰               |
| ممنون                     | من و                      | ۰                 |
| جست و جو                  | جست و جو و                | ۰                 |
| فارسی                     | فارسی                     | ۱۰۰               |
| بله                       | بله                       | ۱۰۰               |
| عبارت                     | بیارد                     | ۰                 |
| نیستم                     | نیستم                     | ۱۰۰               |
| سیستم                     | سیستم                     | ۱۰۰               |
| فناوری                    | فناوری                    | ۱۰۰               |
| الگوریتم                  | الگوریتم                  | ۱۰۰               |
| روش                       | ورزش                      | ۰                 |
| انگلیسی                   | انگلیسی                   | ۱۰۰               |
| تحلیل                     | تحلیل                     | ۱۰۰               |
| خواص                      | حواس                      | ۰                 |
| مشتري                     | مشتري                     | ۱۰۰               |
| مهارت                     | مهارت                     | ۱۰۰               |
| هزینه                     | هزینه                     | ۱۰۰               |
| دقت                       | بن بست                    | ۰                 |
| بهینه                     | بهینه                     | ۱۰۰               |
| مشترک                     | مشترک                     | ۱۰۰               |
| گیاه شناسی و اصلاح خاک    | گیاه شناسی و اصلاح خاک    | ۱۰۰               |
| هنر و اندیشه در معماری    | هنر و اندیشه در معماری    | ۱۰۰               |
| جداسازی اتم هیدروژن       | جداسازی اتم هیدروژن       | ۱۰۰               |
| خواص مکانیکی نانولامپ     | خواص مکانیکی نانولامپ     | ۱۰۰               |
| تحلیل پایداری تونل        | تحلیل پایداری تونل        | ۱۰۰               |
| رضایت مشتری و بازاریابی   | رضایت مشتری و بازاریابی   | ۱۰۰               |
| مربی ورزش و سبک رهبری     | مربی ورزش و سبک رهبری     | ۱۰۰               |
| استفاده از الگوریتم ژنتیک | استفاده از الگوریتم ژنتیک | ۱۰۰               |
| اثر اقتصاد مقاومتی        | اثر اقتصاد مقاومتی        | ۱۰۰               |
| تاریخچه شعر فارسی         | تاریخچه شعر فارسی         | ۱۰۰               |

تک واژه ای

کمتر از ۵ واژه

| عبارت اصلی                                       | خروجی سیستم تشخیص گفتار                          | درصد واژگان مشترک |
|--------------------------------------------------|--------------------------------------------------|-------------------|
| استفاده از الگوریتم ژنتیک در خودکار سازی         | استفاده از الگوریتم ژنتیک در خودکار سازی         | ۱۰۰               |
| نقش فناوری اطلاعات در برنامه توسعه               | بخش فناوری اطلاعات در برنامه توسعه               | ۸۰                |
| اثر هیجان در رضایت از زندگی و امید به زندگی      | اثر هیجان و رضایت از زندگی و امید به زندگی       | ۱۰۰               |
| کاهش مصرف انرژی با استفاده از تابع هزینه نفت خام | کاهش مصرف انرژی با استفاده از تابع هزینه نفت خام | ۱۰۰               |
| اصلاح اجتماعی و جلوگیری از جرم                   | اصلاح اجتماعی و جلوگیری از جرم                   | ۱۰۰               |
| نقش دولت در برنامه ریزی اقتصادی و سیاسی          | نقش دولت در برنامه ریزی اقتصادی و سیاسی          | ۱۰۰               |
| بهره برداری از زمین و مدیریت جنگل                | بهره برداری از زمین و مدیریت جنگل                | ۱۰۰               |
| نقش عدالت سازمانی در رضایت شغلی                  | بخش عدالت سازمانی در رضایت شغلی                  | ۸۰                |
| ارائه روش بهینه سازی هزینه حمل و نقل             | ارائه روش بهینه سازی هزینه حمل و نقل             | ۱۰۰               |

### ۳-۶. ارزیابی زیرسیستم تبدیل متن به گفتار

برای ارزیابی عملکرد زیرسیستم تبدیل متن به گفتار، چندین جمله به سیستم ارسال شده و نتیجه گفتاری و تلفظی سیستم بررسی شده است. این جملات به صورت تصادفی از پایگاه داده گنج استخراج شده اند. در

جدول ۸-۶ نتایج عملکرد زیرسیستم تبدیل متن به گفتار از لحاظ درستی تلفظ واژگان در یک جمله آمده است. همانطور که مشاهده می‌شود، برخی از کلمات تخصصی علمی توسط این سیستم به درستی تلفظ نمی‌شود. البته امکان اضافه کردن تلفظ درست این کلمات به دامنه واژگانی سیستم وجود دارد که لازم است توسط تامین کننده اصلی سیستم صورت بگیرد.

در خواندن جملات، علایم سجاوندی نقطه و ویرگول رعایت می‌شود، به گونه‌ای که انتهای جمله با لحن متفاوتی خوانده می‌شود و برای ویرگول مکث کوتاهی در خواند رعایت می‌شود. لیکن در برخی از موارد ترکیبات مضاف و مضاف الیه و کسره به درستی رعایت نمی‌گردد.

جدول ۸-۶- نتایج عملکرد زیر سیستم تبدیل متن به گفتار در تلفظ واژگان

| ردیف | جملات                                                                                                                                                                                                                                                                                                                                            | واژگان با خطا در تلفظ               |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
| ۱    | فلاوانوئیدها دسته‌ای از فنولیک‌های گیاهی ثانویه با خواص آنتی اکسیدانی هستند.                                                                                                                                                                                                                                                                     | فلاوانوئیدها،<br>فنولیک‌های         |
| ۲    | برعکس کنوانسیون‌های بین‌المللی، جایگاه کرامت انسانی از منظر دین اسلام ثابت است.                                                                                                                                                                                                                                                                  | بین<br>المللی، کنوانسیون‌ها<br>ای   |
| ۳    | ادبیات شفاهی هر ملت بیانگر فرهنگ، روحیات و باورهای آن ملت است.                                                                                                                                                                                                                                                                                   | هر                                  |
| ۴    | تشخیص لهجه از روی شکل موج گفتار یکی از شاخه‌های نسبتاً جدید در علم پردازش گفتار می‌باشد.                                                                                                                                                                                                                                                         | -                                   |
| ۵    | موتور سوئیچ رلوکتانسی یکی از اولین موتورهای ساخته شده توسط بشر می‌باشد.                                                                                                                                                                                                                                                                          | رلوکتانسی                           |
| ۶    | تلفات خاک و رواناب تابعی از عوامل مختلف از قبیل ویژگیهای خاک، توپوگرافی و مدیریت اراضی است.                                                                                                                                                                                                                                                      | رواناب                              |
| ۷    | هدایت آبی اشباع و پارامتر طول‌درشت موینگی از ویژگی‌های هیدرولیکی خاک می‌باشند.                                                                                                                                                                                                                                                                   | -                                   |
| ۸    | تیو استال‌ها یکی دیگر از گروه‌های محافظت کننده مهم کربونیل هستند.                                                                                                                                                                                                                                                                                | تیو استال‌ها                        |
| ۹    | کنترل ولتاژ و توان راکتیو یکی از اساسی ترین مسائل پیش روی شبکه‌های توزیع به حساب می‌آید.                                                                                                                                                                                                                                                         | -                                   |
| ۱۰   | مبدل‌های تک‌فاز به سه‌فاز برای درایو موتورهای القایی و تغذیه‌ی بارهایی که به منبع تک‌فاز دسترسی دارند به کار می‌رود.                                                                                                                                                                                                                             | تک فاز با کسره<br>تلفظ شده است.     |
| ۱۱   | محافظت از ترکیبات کربونیل دار نقش مهمی را در در سنتزهای چند مرحله‌ای در شیمی آلی، شیمی دارویی، شیمی کربوهیدرات‌ها و طراحی دارو ایفا می‌کند. استال‌ها یکی از مهمترین گروه‌های محافظت کننده آلدهیدها و کتون‌ها در برابر حمله هسته دوست‌ها، اکسید کننده‌ها، معرف‌های آلی فلزی و معرف‌های بنیادی برای مثال در سنتز کربوهیدرات‌ها و استروئیدها هستند. | آلدهیدها،<br>کتون‌ها،<br>استروئیدها |
| ۱۲   | آیا نظام جدید آموزشی باعث بهبود عملکرد دانش آموزان می‌شود؟                                                                                                                                                                                                                                                                                       | -                                   |
| ۱۳   | نقش فرهنگ سیاسی نخبگان سیاسی در توسعه نیافتگی سیاسی به چه اندازه بوده است؟                                                                                                                                                                                                                                                                       | نیافتگی                             |
| ۱۴   | چگونه می‌توان سیستم‌های خبره را به کمک الگوریتم‌های شبکه عصبی بهینه کرد؟                                                                                                                                                                                                                                                                         | -                                   |

#### ۶-۴. ارزیابی سیستم بازیابی اطلاعات

در این بخش ابتدا داده‌های پایگاه داده گنج معرفی می‌شوند و عملیات آماده‌سازی آنها تشریح می‌شود. سپس نتایجی از سیستم بازیابی اطلاعات و نمایش مصور نتایج جستجو ارائه می‌شود.

##### ۶-۴-۱. آماده‌سازی داده‌ها

سیستم بازیابی اطلاعات روی مجموعه‌ای منتخب از داده‌های پایگاه داده گنج ایرانداک پیاده‌سازی شده است. این مجموعه شامل ۳۹۹۹۷ رکورد است. هر رکورد یک پایان نامه فارسی است مشخصات عنوان، چکیده، کلمات کلیدی و موضوع برای آن در پایگاه داده وجود دارد. مدل داده‌ای این مجموعه داده همان مدل داده‌ای پایگاه گنج است که در محیط SQL قابل دسترسی است.

هر رکورد اطلاعاتی (پایان نامه) می‌تواند تعدادی نامحدودی کلمه کلیدی داشته باشد و این تعداد برای پایان نامه‌های مختلف، متفاوت است. برای افزایش کارایی و سرعت پاسخگویی برنامه، همه کلمات کلیدی هر رکورد به صورت یک رشته متنی به هم متصل گردیده است. این تغییر به صورت آفلاین و تنها یکبار انجام می‌شود و همه کلمات کلیدی را از سطرهای مختلف به یک رشته متنی تبدیل می‌کند. این تغییر باعث افزایش سرعت برنامه و کاهش پیچیدگی زمانی آن از  $O(n^3)$  به  $O(n^2)$  می‌گردد که در عمل کارایی سامانه جستجو را تا حد بسیار زیادی افزایش خواهد داد.

برای آماده‌سازی داده‌ها دو عملیات «مرتب‌سازی موضوعات» و «پاکسازی و استانداردسازی داده‌ها» انجام شده است که توضیحات مربوط به هر یک در ادامه آمده است.

##### ۶-۴-۱-۱. مرتب‌سازی موضوعات داده‌ها

برای دسته‌بندی داده‌ها و نمایش مصور آنها، لازم است که یک دسته‌بندی موضوعی از داده‌ها وجود داشته باشد که البته اکثر داده‌های گنج دارای این دسته‌بندی موضوعی هستند. برای دسته‌بندی موضوعی دو سطح اصلی در نظر گرفته شده است:

- سطح اول یا موضوعات اصلی شامل: فنی و مهندسی، علوم پایه، علوم اجتماعی، و هنر و ادبیات
- سطح اول یا موضوعات فرعی: هر موضوع اصلی شامل مجموعه‌ای از موضوعات فرعی است که در جدول ۶-۹ نمایش داده شده است.

جدول ۶-۹- دسته‌بندی موضوعی، سطح اول و سطح دوم

| سطح اول                 | سطح دوم                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| فنی و مهندسی (۲۷ موضوع) | <ul style="list-style-type: none"> <li>کشاورزی، عمومی و میان رشته‌ای</li> <li>مهندسی برق و الکترونیک</li> <li>مهندسی مکانیک</li> <li>علوم کامپیوتر، سخت افزار و معماری</li> <li>زمین شناسی مهندسی</li> <li>مهندسی عمومی و میان رشته‌ای</li> <li>مهندسی بافت و سلول</li> <li>علوم کامپیوتر، مهندسی نرم افزار</li> <li>رباتیک و کنترل</li> </ul>                                                                                                        | <ul style="list-style-type: none"> <li>مهندسی شیمی</li> <li>مهندسی صنایع</li> <li>مهندسی متالورژی</li> <li>مکانیک</li> <li>مهندسی دریایی</li> <li>مهندسی نفت</li> <li>فناوری اطلاعات</li> <li>مهندسی عمران</li> <li>مهندسی مصنوعات</li> <li>تحقیق در عملیات</li> </ul>                                                                                                                                                                                                                                                           |
| علوم پایه (۴۷ موضوع)    | <ul style="list-style-type: none"> <li>ریاضیات</li> <li>شیمی آلی</li> <li>زیست شناسی</li> <li>زمین شناسی</li> <li>شیمی تجزیه</li> <li>ریاضیات کاربردی</li> <li>شیمی فیزیک</li> <li>آمار و احتمالات</li> <li>فیزیک مواد چگال</li> <li>فیزیک ذرات و میدانها</li> <li>فیزیک هسته ای</li> <li>شیمی کاربردی</li> <li>شیمی عمومی و میان رشته ای</li> <li>فیزیک کاربردی</li> <li>زیست شناسی سلولی</li> <li>علوم و فناوری نانو</li> <li>علوم اعصاب</li> </ul> | <ul style="list-style-type: none"> <li>فیزیک عمومی و میان رشته ای</li> <li>شیمی معدنی و هسته ای</li> <li>فیزیک اتمی، مولکولی و شیمیایی</li> <li>بیوشیمی و زیست شناسی</li> <li>ملکولی</li> <li>علوم کامپیوتر، کاربردهای میان رشته ای</li> <li>زیست شناسی دریایی و آب شیرین</li> <li>علوم مواد. مواد زیستی</li> <li>زیست شناسی ریاضی و کامپیوتری</li> <li>علوم مواد. مواد ترکیبی</li> <li>علوم مواد. میان رشته ای</li> <li>علوم اجتماعی (زیست پزشکی)</li> <li>علوم مواد. منسوجات</li> <li>علوم مواد، نمونه سازی و قطعات</li> </ul> |

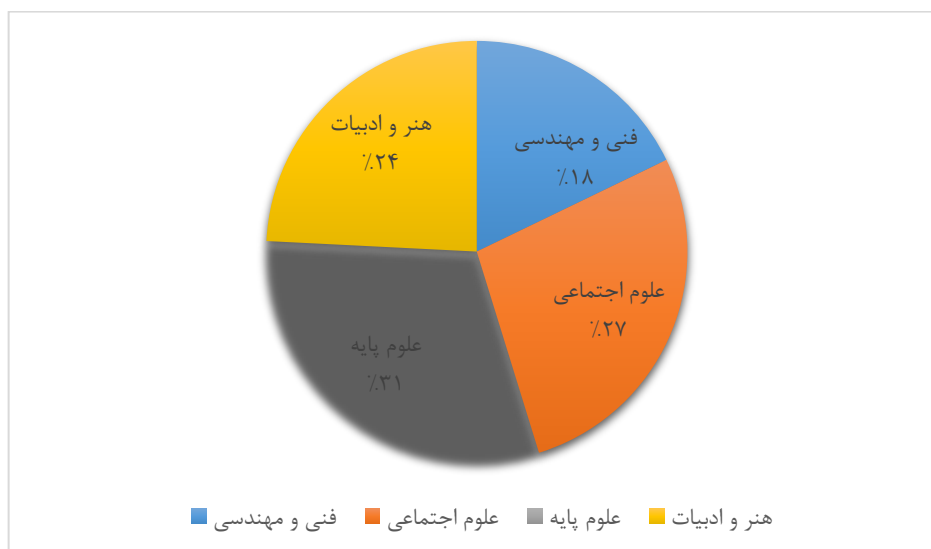
|                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>علوم اجتماعی</b><br/>(۴۲ موضوع)</p> | <ul style="list-style-type: none"> <li>• روانشناسی</li> <li>• اقتصاد کشاورزی، سیاستگذاری کشاورزی</li> <li>• علوم رفتاری</li> <li>• مدیریت</li> <li>• اقتصاد</li> <li>• حقوق</li> <li>• جغرافیا</li> <li>• آموزش و پرورش و تحقیقات تربیتی</li> <li>• روانشناسی تربیتی</li> <li>• روانشناسی عمومی و میان رشته‌ای</li> <li>• علوم سیاسی</li> <li>• تاریخ</li> <li>• جامعه‌شناسی</li> <li>• ارگونومیک</li> <li>• سوء مصرف مواد (خدمات اجتماعی)</li> </ul> | <ul style="list-style-type: none"> <li>• روانشناسی بالینی</li> <li>• علوم اجتماعی (میان رشته‌ای)</li> <li>• علوم اطلاعات ، کتابداری و دانش‌شناسی</li> <li>• ارتباط‌شناسی (علوم اجتماعی)، ارتباطات</li> <li>• اوقات فراغت، سرگرمی، ورزش، گردشگری و هتلداری</li> <li>• جمعیت‌شناسی</li> <li>• روانشناسی کاربردی</li> <li>• روانشناسی ، روانکاوی</li> <li>• مطالعات خانواده</li> <li>• خدمات اجتماعی</li> </ul> | <ul style="list-style-type: none"> <li>• علوم اجتماعی (روشهای ریاضی)</li> <li>• حمل و نقل</li> <li>• مطالعات ناحیه‌ای</li> <li>• قوم‌شناسی</li> <li>• سالمندی (خدمات اجتماعی)</li> <li>• روانشناسی بیولوژیک</li> <li>• روانشناسی بالینی</li> <li>• جرم‌شناسی، بزهکاری</li> <li>• مردم‌شناسی</li> <li>• کسب و کار، بازرگانی</li> <li>• روابط بین‌الملل</li> <li>• روانشناسی اجتماعی</li> <li>• مطالعات زنان</li> <li>• برنامه‌ریزی و توسعه</li> <li>• مطالعات فرهنگی</li> <li>• تاریخ علوم اجتماعی</li> <li>• مسایل اجتماعی</li> </ul> |
| <p><b>هنر و ادبیات</b><br/>(۳۶ موضوع)</p> | <ul style="list-style-type: none"> <li>• ادبیات فارسی</li> <li>• ادبیات انگلیسی</li> <li>• معماری</li> <li>• ادبیات عربی</li> <li>• شهرسازی</li> <li>• هنرهای نمایشی</li> <li>• نقاشی</li> <li>• باستان‌شناسی</li> <li>• ادبیات فرانسه</li> <li>• فلسفه هنر</li> <li>• ادبیات آلمانی</li> <li>• طراحی</li> <li>• گرافیک و چاپ</li> </ul>                                                                                                              | <ul style="list-style-type: none"> <li>• طراحی پارچه</li> <li>• ادبیات کودکان و نوجوانان</li> <li>• سینما (فیلمنامه نویسی)</li> <li>• ادبیات اسپانیایی</li> <li>• خوشنویسی</li> <li>• ادبیات تطبیقی، ادبیات جهان</li> <li>• سینما (تدوین)</li> <li>• ادبیات ترکی</li> <li>• هنرهای نمایشی (طراحی صحنه)</li> <li>• مرمت و احیای بافت‌های تاریخی</li> <li>• پژوهش در هنر</li> </ul>                            | <ul style="list-style-type: none"> <li>• ادبیات نمایشی</li> <li>• نقاشی ایرانی ، نگارگری</li> <li>• موسیقی ایرانی</li> <li>• عکاسی و فیلمبرداری</li> <li>• هنرهای تزئینی</li> <li>• موسیقی‌شناسی، موسیقی‌های قومی</li> <li>• سینما (کارگردانی)</li> <li>• ادبیات روسی</li> <li>• موسیقی کلاسیک</li> <li>• فرش و صنایع دستیافت</li> <li>• هنرهای نمایشی (بازیگری)</li> <li>• صنایع دستی و هنرهای سنتی</li> </ul>                                                                                                                       |



در نهایت براساس این دسته‌بندی موضوعی، برای هر رکورد اطلاعاتی، فیلدهای اطلاعاتی زیر وجود دارد:

- شناسه
- عنوان: ارائه مدل طراحی همزمان شبکه حمل و نقل همگانی چند مده (ریلی و اتوبوس) با استفاده از الگوریتم ژنتیک
- موضوع سطح اول، مثلاً فنی و مهندسی
- موضوع سطح دوم، مثلاً مهندسی عمران
- کلمات کلیدی
- چکیده

در شکل ۶-۱۲ توزیع تعداد موضوعات سطح دوم در موضوعات سطح اول نمایش داده شد است.



شکل ۶-۱۲- توزیع موضوعات سطح دوم در موضوعات سطح اول

#### ۶-۴-۱-۲. پاکسازی و استانداردسازی داده‌ها

مجموعه داده‌های استفاده شده از نظر نگارشی اشکالات فراوانی دارند که برخی از آنها عبارتند از:

- استفاده از کاراکترهای غیر معتبر در بخش‌های مختلف داده‌ها
- فیلدهای خالی موجود در جداول پایگاه داده
- غلط‌های املایی در عنوان و کلمات کلیدی
- استفاده از کیبوردهای غیر استاندارد برای وارد کردن اطلاعات توسط کاربران
- استفاده از عبارت‌ها به صورت چند زبانه در یک خانه از جدول پایگاه داده

- وجود یک کلمه به شکل‌های مختلف یا با کاراکترهای متفاوت در قسمت‌های مختلف
- استفاده از کدهای کاراکتری مختلف برای کاراکترهایی مانند «ی» و «ک»

برای برطرف نمودن این مشکلات یک فرایند پاکسازی اولیه روی مجموعه داده انجام شده است. بخشی از این فرایند به صورت دستی و بخشی از براساس مجموع‌ای از قوانین به صورت خودکار و بخش دیگر به صورت خودکار لحاظ شده است.

- فرایند دستی پاکسازی داده‌های کلمات کلیدی: فرایند دستی پاکسازی داده‌ها تنها بر روی فیلد کلمات کلیدی اعمال شده است. زیرا این فیلد در بخش نمایش مصور اطلاعات (که در بخش‌های بعدی توضیح داده خواهد شد استفاده خواهد شد) برای انجام این فرایند، سامانه‌ای طراحی شده که به طور همزمان چندین کاربر به صورت دستی می‌توانند صحت داده‌ها را بررسی نمایند. براساس این فرایند، هر کاربر، برای هر کلمه کلیدی یکی از سه برچسب صحیح، نادرست و مبهم را انتخاب می‌کند. برچسب صحیح به این معنی است که نگارش کلمه صحیح است، نادرست به این معنی است که نگارش کلمه با استفاده از مجموعه‌ای از قوانین پاکسازی به صورت خودکار می‌تواند اصلاح شود و برچسب مبهم به این مفهوم است به راحتی نمی‌توان نگارش کلمه را با استفاده از قوانین خودکار اصلاح نمود. در نهایت کلمات نادرست با استفاده از قوانین خودکار اصلاح شده‌اند. استفاده از قوانین خودکار برای اصلاح کلمات کلیدی مبهم خود نیازمند پردازش‌های جداگانه و استفاده از مدل‌های صرفی و نحو زبانی است که از دامنه پژوهشی این طرح خارج است. بنابراین برای کاهش تعداد رکوردهایی که کلمات کلیدی مبهم دارند، دو راهکار در نظر گرفته شد:

- یک بررسی روی توزیع کلمات کلیدی مبهم روی رکوردها انجام شد. تعداد بسیار زیادی از کلمات کلیدی مبهم تنها یکبار یا فقط در یک رکورد آمده بودند، با حذف آنها و رکوردهای متناظر با آنها، این کلمات از پایگاه داده حذف شدند.
- از طرف دیگر موضوعات سطح دوم نیز بررسی شد، برخی از موضوعات تنها تعداد محدودی رکورد اطلاعاتی را داشتند در حالیکه کلمات کلیدی مبهم بسیار را دربرداشتند. این موضوعات نیز به طور کامل از پایگاه داده حذف شدند.

در نهایت تعداد موضوعات سطح دوم به ۱۵۲ موضوع کاهش یافت در حالیکه تعداد رکوردها از ۳۹۹۹۷ رکورد به ۳۱۸۴۰ و تعداد کلمات کلیدی از ۱۴۳۲۹۲ به تعداد ۴۶۶۰۳، یعنی بیش از یک سوم حالت اولیه کاهش یافت.

- فرایند خودکار پاکسازی داده‌ها: در فرایند خودکار پاکسازی داده‌ها، با استفاده از مجموعه‌ای از قوانین، تا حد امکان، داده‌ها به یک فرمت نگارشی استاندارد تبدیل می‌شوند. این فرایند برای کلیه فیلدهای اطلاعاتی یک رکورد انجام می‌شود. مجموعه قوانینی که در این فرایند برای استانداردسازی داده‌ها بکار گرفته شده عبارتند از:

- تبدیل انواع «ی» به یک نویسه با کدینگ یکسان
- تبدیل انواع «ک» به یک نویسه با کدینگ یکسان
- حذف نمودن همزه آخر از کلمات مانند حذف همزه از املاء. همانطور که گفته شد چون این حذف هم بر منابع و هم از پرس‌وجوی کاربر اجرا می‌شود ایجاد مشکل نخواهد نمود.

- حذف تشدید و تنوین از همه کلمات
- تبدیل چند ( دو یا بیشتر ) فاصله به یک فاصله
- تبدیل فاصله انگلیسی به فاصله فارسی
- تبدیل ا به آ
- تبدیل ئ به ی
- تبدیل و به و
- حذف تمامی اعراب گذاری‌ها
- تبدیل نیم فاصله‌ها به فرم یکسان (Shift + Ctrl + 2)
- حذف تمامی «» (Shift+J) از عبارات به دلیل بی معنا بودن آن. برای مثال «تــــ» به «تا» تبدیل می‌گردد.

- وارد کردن نیم فاصله در موارد خاص مثل «ها» در آخر کلمات و «می» در ابتدای افعال
- حذف «ۀ» از آخر کلمات
- تبدیل همه اعداد انگلیسی به اعداد فارسی

همانطور که ذکر شد این مراحل بر روی همه فیلدهای منابع شامل چکیده، کلمات کلیدی و عنوان انجام شده است و بر روی پرس و جوی کاربر نیز انجام می‌شود. نکته دیگری که باید به آن توجه کرد این است که با توجه به این که پیش پردازش متن می‌تواند متن اولیه را دست خوش تغییر بسیار کرده و حتی در مواردی باعث شود که معنای متن تغییر کند لازم است که برای هر فیلد اطلاعاتی یکی فیلد اطلاعاتی دیگر اضافه شود و متن پیش پردازش شده در آن ذخیره گردد. لذا الگوریتم‌های جستجو بر روی فیلدهای پیش پردازش شده اعمال خواهند شد و الگوریتم‌های نمایش و بازیابی بر روی فیلدهای اصلی عمل خواهند کرد. این کار در اکثر موتورهای جستجو انجام می‌شود.

#### ۶-۴-۲. نتایج زیر سیستم بازیابی اطلاعات

همانگونه که در فصل قبل عنوان شد، برای بازیابی اطلاعات از روش BM25F شده است. پیش از جستجو هر عبارت در پایگاه گنج، عملیات پیش پردازشی روی داده‌ها انجام می‌شود که این عملیات مجموعه‌ای از همان قوانینی هستند که در بخش ۶-۴-۱ برای پاکسازی داده‌ها اعمال شده است. برای نمایش عملکرد الگوریتم BM25F چند عبارت پرس و جو به زیر سیستم جستجو و بازیابی داده شده است. رکودها براساس امتیازشان نسبت به هر عبارت پرس و جو به ترتیب نمایش داده می‌شوند. در جداول ۶-۹ تا ۶-۱۱، چند نمونه نتیجه زیر سیستم بازیابی اطلاعات برای عبارات پرس و جو زیر آمده است:

- تغییر حسابرس و دوره تصدی حسابرسی
- کاهش نشانه‌های اختلال اضطراب اجتماعی
- حل عددی معادلات انتگرال

جدول ۶-۱۰- نتایج حاصل از جستجوی عبارت «تغییر حسابرس و دوره تصدی حسابرسی»

| ردیف | مشخصات رکوردهای بازیابی شده برای عبارت<br>«تغییر حسابرس و دوره تصدی حسابرسی» |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | امتیاز |
|------|------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| ۱    | عنوان                                                                        | بررسی رابطه مدت تصدی حسابرس با کیفیت حسابرسی                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | ۱۴۳/۶۸ |
|      | چکیده                                                                        | کیفیت حسابرسی برای تداوم فعالیت موسسات حسابرسی بسیار حیاتی است و آینده حرفه حسابرسی تا حد قابل ملاحظه‌ای به آن بستگی دارد از آنجایی که کیفیت حسابرسی یک ساختار پنهان چند بعدی دارد اندازه گیری آن مشکل است بنابراین تعریف جامعی از کیفیت حسابرسی که در برگیرنده همه انواع حسابرسی باشد وجود ندارد از این رو این تحقیق به بررسی دو شاخص مهم در تعیین کیفیت حسابرسی یعنی دوره تصدی حسابرس و اندازه موسسه حسابرسی پرداخته است مساله تحقیق این است که آیا دوره تصدی حسابرس با کیفیت حسابرسی ارتباط معناداری دارد... |        |
| ۲    | عنوان                                                                        | اتکا حسابرسان مستقل بر کار حسابرسی داخلی از دیدگاه اندازه موسسات حسابرسی                                                                                                                                                                                                                                                                                                                                                                                                                                        | ۱۳۹/۱۷ |
|      | چکیده                                                                        | حسابرسان مستقل همواره در صدد انجام حسابرسی با کارایی بیشتر بدون کاهش در کیفیت و اثربخشی بهتر آن هستند استفاده‌ی مطلوب از کار حسابرسی داخلی می تواند باعث بهبود کارایی اثربخشی حسابرسی‌های مستقل شود و ارزش حسابرس داخلی را برای سازمان صاحبکار افزایش دهد این پژوهش به افزایش نیاز برای اتکا بر کار حسابرس داخلی تاکید می کند و حسابرس داخلی را به عنوان منبعی برای حسابرس مستقل معرفی می کند....                                                                                                               |        |
| ۳    | عنوان                                                                        | تاثیر کیفیت عملکرد حسابرسی داخلی بر زمان اجرای حسابرسی مستقل                                                                                                                                                                                                                                                                                                                                                                                                                                                    | ۱۲۹/۶۲ |
|      | چکیده                                                                        | تحقیق حاضر به بررسی کیفیت عملکرد حسابرسی داخلی بر زمان اجرای حسابرسی مستقل می پردازد در این پژوهش تعداد روزهای بین پایان سال مالی و تاریخ ارایه گزارش حسابرس مستقل به عنوان زمان انجام کار حسابرسی تعریف شده است افزایش مدت زمان مزبور باعث ایجاد عدم تقارن اطلاعاتی برای استفاده کنندگان از گزارشات مالی شرکت می گردد که این موضوع باعث کاهش محتوای اطلاعات در گزارش‌های حسابداری به صورتی بالقوه می شود علاوه بر این تاخیر در ارایه گزارش حسابرسی به عنوان نماینده‌ای برای کارایی حسابرسی عمل می کند....      |        |
| ۴    | عنوان                                                                        | بررسی تاثیر جنسیت حسابرس بر کیفیت حسابرسی مستقل                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | ۱۲۸/۱۶ |
|      | چکیده                                                                        | این پژوهش به بررسی تاثیر تفاوت‌های جنسیتی بر کیفیت حسابرسی مستقل می پردازد جهت تعیین کیفیت حسابرسی مستقل از سه معیار توانایی حل مساله ریسک و استقلال حسابرس هاردیس و بریچ استفاده شده است اطلاعات مورد نیاز در این پژوهش از طریق پرسش نامه‌های توزیع شده بین حسابرسان شاغل عضو جامعه حسابداران رسمی ایران جمع آوری گردید نتایج آزمون آماری t دو نمونه‌ای نشان می دهد که اگرچه تفاوت‌های جنسیتی بر رابطه بین کیفیت حسابرسی و توانایی حسابرس در حل مساله تاثیر گذار است....                                       |        |
| ۵    | عنوان                                                                        | بررسی دیدگاه استفاده کنندگان از صورت‌های مالی در خصوص تاثیر چرخش حسابرس بر کیفیت حسابرسی                                                                                                                                                                                                                                                                                                                                                                                                                        | ۱۱۷/۵۹ |
|      | چکیده                                                                        | در این تحقیق دیدگاه استفاده کنندگان از صورت‌های مالی در خصوص تاثیر چرخش اجباری موسسات حسابرسی بر کیفیت حسابرسی مورد مطالعه قرار گرفته است در واقع این تحقیق پاسخی به این پرسش است که چرخش اجباری موسسات حسابرسی چه تاثیری بر کیفیت حسابرسی دارد منظور از چرخش موسسات حسابرسی جا به جایی و تعویض حسابرسان در دوره‌های زمانی مشخص و کیفیت حسابرسی نیز تابع ترکیبی از شایستگی توانایی کشف اشتباه‌ها و استقلال حسابرسان انگیزه گزارش موارد کشف شده می باشد....                                                      |        |

جدول ۶-۱۱- نتایج حاصل از جستجوی «کاهش نشانه‌های اختلال اضطراب اجتماعی»

| ردیف | مشخصات رکوردهای بازیابی شده برای عبارت<br>«کاهش نشانه‌های اختلال اضطراب اجتماعی»                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | امتیاز |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| ۱    | عنوان<br>بررسی و مقایسه اثربخشی گروه درمانی شناختی رفتاری هیجان مدار و گروه درمانی شناختی رفتاری بر اختلال اضطراب جدایی و اضطراب اجتماعی کودکان                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | ۱۰۲/۹۸ |
|      | چکیده<br>هدف از پژوهش حاضر بررسی و مقایسه‌ی اثربخشی گروه درمانی شناختی رفتاری CBT و گروه درمانی شناختی رفتاری هیجان مدار ECTB بر علایم اضطرابی تنظیم شناختی هیجانات و مدیریت هیجانات خشم و غم در کودکان مبتلا به اضطراب اجتماعی و اضطراب جدایی بود این پژوهش به شیوه آزمایشی و با طرح پیش آزمون پس آزمون و پیگیری سه ماه بعد با گروه کنترل اجرا شد جامعه آماری این پژوهش شامل کلیه‌ی دختران و پسران ساله‌ی مبتلا به یکی از اختلال‌های اضطراب جدایی یا اختلال اضطراب اجتماعی در سال ....                                                                                                                                                                                                 |        |
| ۲    | عنوان<br>تعیین شیوع اختلال بدشکلی بدن و میزان همبندی آن با اختلال افسردگی اساسی و سواس فکری عملی اضطراب اجتماعی و اختلالات خوردن در دانشجویان دانشگاه شیراز و علوم پزشکی شیراز                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | ۸۹/۲۱  |
|      | چکیده<br>اگر نگرانی در مورد وضعیت ظاهری بدن تشدید یابد و موجب ناراحتی چشمگیر اشتغال ذهنی و آسیب به کارکردهای اجتماعی شغلی و سایر جنبه‌های زندگی شود به عنوان اختلال بدشکلی بدن تشخیص داده می شود با توجه به شیوع بالای اختلال بدشکلی بدن در دانشجویان و همبندی با سایر اختلالات این پژوهش با هدف تعیین شیوع اختلال بدشکلی بدن و میزان همبندی آن با اختلال افسردگی اساسی و سواس فکری عملی اضطراب اجتماعی و اختلالات خوردن صورت گرفته است...                                                                                                                                                                                                                                              |        |
| ۳    | عنوان<br>تاثیر موسیقی درمانی در کودکان با اختلال ADD توام با اختلال اضطراب و کودکان با اختلال اضطراب                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | ۸۴/۰۲  |
|      | چکیده<br>اختلال کمبود توجه و تمرکز ADD و اختلال اضطراب شایعترین اختلال عصبی رفتاری دوران کودکی می باشند گرچه برنامه‌های دارو درمانی کار درمانی و بازی درمانی در کاهش علائم این اختلالات سهم عمده‌ای دارند اما برای بهبود روابط و تعاملات اجتماعی و عاطفی ایشان به مداخلات دیگری که نقش مکملی را در درمان ایفا می کنند نیاز است یکی از تاثیرگذارترین برنامه‌ها برگزاری جلسات موسیقی درمانی می باشد هدف این مطالعه بررسی تاثیر موسیقی در کودکان با اختلال ADD که همزمان اختلال اضطراب را نیز دارند و کودکان با اختلال اضطرابی می باشد در این مطالعه شبه تجربی کودک ADD تا ۱۱ ساله که اختلال اضطراب را نیز داشتند کودک تا ۱۱ ساله با اختلال اضطراب تحت پوشش موسسه مهرآفرین انتخاب شدند ... |        |
| ۴    | عنوان<br>مقایسه علائم اختلال خوردن باورهای اختلال خوردن شیوه فرزندپروری کنترل طرد والدین و نشخوار فکری در نوجوانان دارای نشانه‌های اختلال خوردن همراه با افسردگی و بدون افسردگی                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | ۸۳/۱۳  |
|      | چکیده<br>زمینه و اهداف هدف پژوهش حاضر مقایسه علایم اختلال خوردن باورهای اختلالات خوردن شیوه فرزند پروری کنترل طرد والدین و نشخوار فکری در افراد دارای نشانه‌های اختلال خوردن همراه با افسردگی و بدون افسردگی بود طرح پژوهشی به کار رفته در این تحقیق علی مقایسه‌ای میباشد بدین منظور نمونه‌ای به حجم که نمره صفر داشتند حذف شدند و نفر دارای نشانه‌های اختلال خوردن بدون افسردگی و نفر دارای نشانه‌های اختلال خوردن با افسردگی به روش نمونه گیری خوشه‌ای چند مرحله‌ای تصادفی از دانش آموزان دختر شهر شیراز انتخاب شدند و مورد آزمون قرار گرفتند...                                                                                                                                      |        |

| ردیف | مشخصات رکوردهای بازیابی شده برای عبارت<br>«کاهش نشانه‌های اختلال اضطراب اجتماعی» |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | امتیاز |
|------|----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| ۵    | عنوان                                                                            | اثربخشی برنامه درمانگری کندال بر کاهش نشانه‌های اضطرابی بهبود پیشرفت تحصیلی و نظم جویی شناختی هیجانی کودکان مبتلا به اختلال‌های اضطرابی                                                                                                                                                                                                                                                                                                                                                                                                  | ۷۳/۱۳  |
|      | چکیده                                                                            | اختلال‌های اضطرابی در میان اختلال‌های روان پزشکی کودکان گسترده‌ترین و فراوان‌ترین اختلال‌ها هستند به گونه‌ای که تحول شناختی اجتماعی و هیجانی کودک را مختل می کنند شیوع بالا و ایجاد پیامدهای منفی درازمدت بر تحول کودکان نیاز به درمان موثر را برجسته می کند پژوهش حاضر با هدف بررسی اثربخشی آموزش برنامه درمانی کندال بر کاهش نشانه‌های بالینی اضطراب بهبود عملکرد تحصیلی و بهبود نشانه‌های نظم جویی شناختی هیجان کودکان مبتلا به اختلال‌های اضطرابی انجام شد این پژوهش شبه آزمایشی با طرح پیش آزمون پس آزمون با گروه کنترل می باشد.... |        |

جدول ۶-۱۲- نتایج حاصل از جستجوی «حل عددی معادلات انتگرال»

| ردیف | مشخصات رکوردهای بازیابی شده برای عبارت<br>«حل عددی معادلات انتگرال» |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | امتیاز |
|------|---------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| ۱    | عنوان                                                               | حل عددی معادلات انتگرال و معادلات دیفرانسیل انتگرال با استفاده از روش آنالیز هموتویی                                                                                                                                                                                                                                                                                                                                                                                                 | ۸۵/۰۵  |
|      | چکیده                                                               | این پایان نامه در چهار فصل تنظیم شده است فصل اول را به بیان پیشینه‌های ریاضی این پایان نامه و مقدمه‌ای بر معادلات انتگرال و معادلات دیفرانسیل انتگرال و همچنین به کاربردهای آنها اختصاص می دهیم در فصل دوم تکنیکهای مختلفی را برای حل معادلات انتگرال بررسی می کنیم و با ارایه مثال‌هایی از روش‌های مختلف حل معادلات انتگرال جهت درک بهتر و ساده تر شدن آن‌ها متمرکز می شویم...                                                                                                      |        |
| ۲    | عنوان                                                               | حل عددی معادلات انتگرال دیفرانسیل غیرخطی توسط روش نیوتن تاو                                                                                                                                                                                                                                                                                                                                                                                                                          | ۷۶/۱۳  |
|      | چکیده                                                               | در این پایان نامه ابتدا معادلات انتگرال را معرفی خواهیم کرد سپس به بیان دسته بندی معادلات انتگرال تعاریف و قضایای مورد نیاز می پردازیم در فصل دوم مقدمه‌ای از آنالیز حقیقی و تابع لاپلاس را بیان خواهیم کرد فصل سوم را به بیان چند روش از روش‌های حل عددی و تحلیلی معادلات انتگرال و معادلات انتگرال دیفرانسیل اختصاص خواهیم داد ....                                                                                                                                                |        |
| ۳    | عنوان                                                               | حل عددی معادلات انتگرالی با استفاده از توابع پایه‌ای شعاعی                                                                                                                                                                                                                                                                                                                                                                                                                           | ۷۶/۰۱  |
|      | چکیده                                                               | در این پایان نامه به حل انواع معادلات انتگرالی به عنوان یکی از مهمترین ابزار علوم پایه و مهندسی می پردازیم روش‌های ارائه شده شامل تلفیقی از روش هم محلی توابع پایه‌ای شعاعی تفاضلات متناهی و برخی دیگر از روش‌های عددی می باشد با توجه به این که روش توابع پایه‌ای شعاعی برای ابعاد بالا انعطاف پذیری مناسبی دارد هدف اصلی پایان نامه توسعه روش توابع پایه‌ای شعاعی برای حل انواع معادلات انتگرال دوبعدی و لترا فردهلم و لترا فردهلم و نیز حل معادلات انتگرال دیفرانسیل با مشتقات... |        |
| ۴    | عنوان                                                               | روش تکرار توسیع عددی برای حل معادلات انتگرال فردهلم خطی                                                                                                                                                                                                                                                                                                                                                                                                                              | ۷۳/۵۲  |
|      | چکیده                                                               | بسیاری از مسائل علوم فیزیک و مهندسی به معادلات انتگرال خطی منجر می شوند در عمل تعداد بسیار کمی از آن‌ها را می توان به روش تحلیلی حل نمود و جواب دقیق آن‌ها را به دست آورد بنابراین از روش‌های عددی برای محاسبه جواب تقریبی آن‌ها استفاده می گردد در این پایان نامه به بررسی دو روش عددی حل معادلات انتگرال فردهلم خطی و دستگاه آن می پردازیم روش اول یک روش تکرار توسیع عددی بر اساس فرم‌های برداری توابع تکانه بلوکی و ماتریس عملیاتی انتگرال می باشد....                           |        |

| ردیف | مشخصات رکوردهای بازیابی شده برای عبارت<br>«حل عددی معادلات انتگرال» |                                                                                                                                                                                                                                                                                                                                                                                                                                                      | امتیاز |
|------|---------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| ۵    | عنوان                                                               | حل معادلات انتگرال دیفرانسیل کسری با روش تبدیل دیفرانسیل تعمیم یافته                                                                                                                                                                                                                                                                                                                                                                                 | ۷۲/۷۹  |
|      | چکیده                                                               | معادلات انتگرال یکی از ابزارهای مهم در ریاضیات کاربردی و محض است این نوع معادلات در مدل سازی بسیاری از پدیده‌های غیرخطی پدیده‌های فیزیکی و علوم مهندسی ظاهر می شوند اکثر پدیده‌های فیزیکی و مسائل مهندسی مانند دینامیک سیالات مکانیک کوانتومی انتقال حرارت رشد جمعیت و وراثت مطالعه رفتار راکتورهای هسته‌ای انتقال بیماری و را می توان از طریق مدل سازی ریاضی آن‌ها درک کرد در واقع بعد از بیان فیزیکی این مسائل می توان مدل ریاضی آن را بیان کرد... |        |

### ۶-۴-۳. دسته‌بندی و نمایش مصور نتایج جستجو

برای دسته‌بندی نتایج جستجو از نمودارهایی برای مصورسازی نمایش نتایج جستجو استفاده شده است. هر یک از رکوردهای گنج علاوه بر عنوان و چکیده، دارای مجموعه‌ای از کلمات کلیدی است که عمدتاً توسط نمایه‌سازان با استفاده از اصطلاحنامه‌های تخصصی، تخصیص داده شده است. علاوه بر آن، برای هر رکورد اطلاعاتی، یک موضوع سطح اول و یک موضوع سطح دوم نیز مشخص شده است.

برای دسته‌بندی نتایج جستجو از موضوعات سطح اول استفاده می‌کنیم. یک رویکرد ساده برای دسته‌بندی، اینست که برای هر رکورد تنها یک موضوع را به عنوان موضوع اصلی در نظر بگیریم، که چنین وضعیتی هم اکنون برای داده‌های گنج در نظر گرفته شده است.

لیکن با چنین رویکردی ماهیت رکوردهای میان رشته‌ای را نمی‌توان مشخص کرد. برای این منظور روشی را پیشنهاد می‌دهیم که براساس آن بتوان، برای هر رکورد یا مجموعه‌ای از رکوردها مشخص کرد که چه میزان به هر موضوع اصلی تعلق دارند. بنابراین به دنبال برداری موضوعی برای هر رکورد هستیم که نشان دهد هر رکورد به چه میزان به هر موضوع تعلق دارد.

فرض کنیم که مجموعه کلیه کلمات کلیدی را با  $K$ ، تعداد موضوعات سطح اول را با  $S$  و تعداد موضوعات سطح دوم را با  $L$  نشان دهیم. آنگاه می‌توانیم برای یک کلمه کلیدی  $k_i$ ، براساس تعداد رکوردهایی که آن کلمه کلیدی را دارند، موضوعات سطح اول و دوم را تخصیص دهیم.

ماتریسی را در نظر می‌گیریم که سطرهای آن کلمات کلیدی و ستون‌های آن موضوعات سطح دوم هستند. هر درایه  $t_{ij}$  از این ماتریس بیانگر وزن هر کلمه کلیدی در هر موضوع سطح دوم است:

$$t_{ij} = \frac{1}{M_i} * \#(\text{رکوردها با کلمه کلیدی } k_i \text{ و موضوع } l_j) \quad (1)$$



که  $M_i$  تعداد کل رکوردهایی است که کلمه کلیدی  $k_i$  را دارند. بنابراین برای کلمه کلیدی  $k_i$  یک بردار موضوعی به صورت زیر با طول تعداد کل موضوعات سطح دوم ( $L$ ) در پایگاه داده خواهیم داشت:

$$X_i = [t_{i1} \quad t_{i2} \dots t_{iL}] \quad (۲)$$

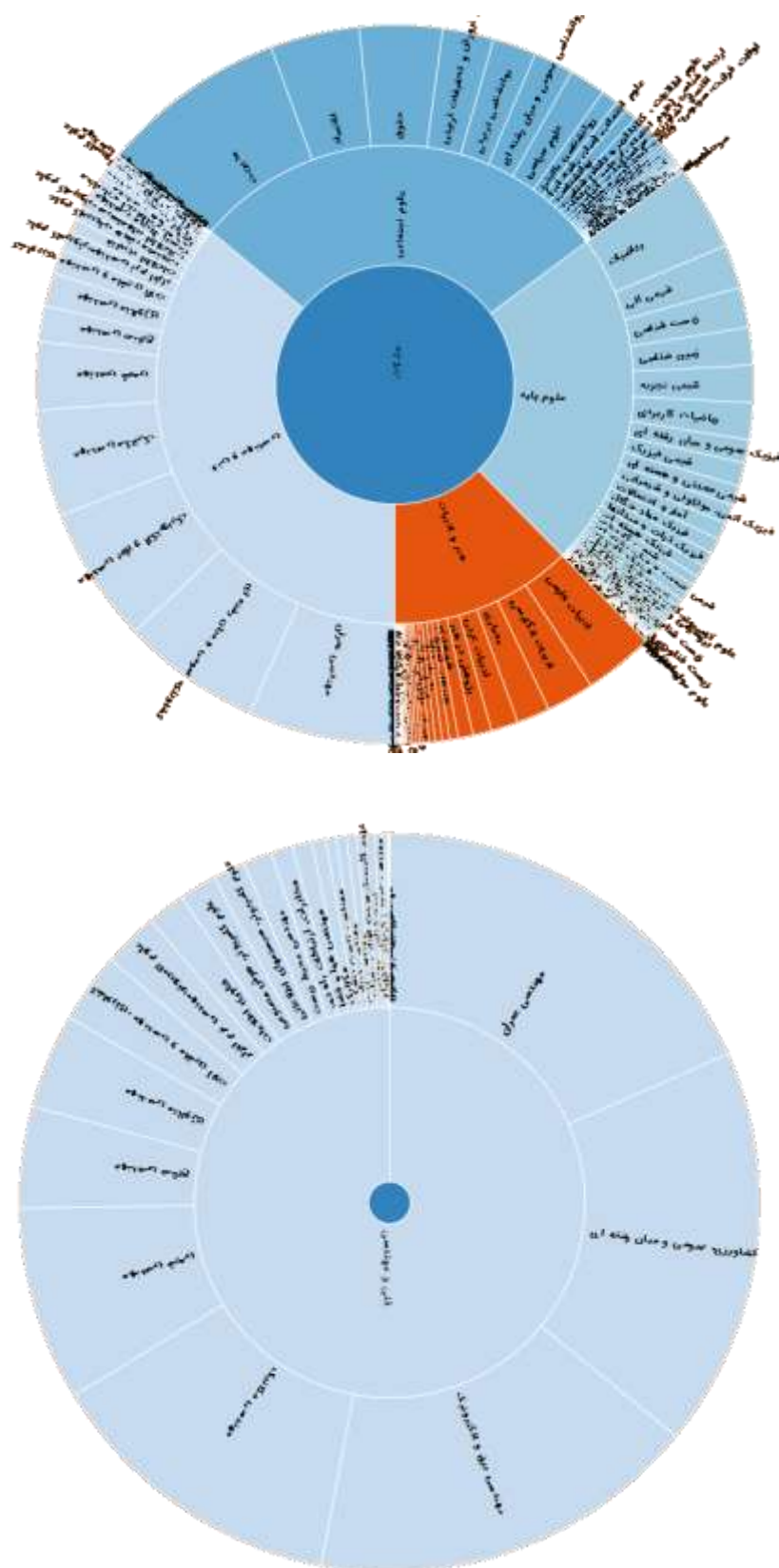
از طرفی برای هر موضوع سطح دوم نیز یک بردار بر حسب کلمات کلیدی خواهیم داشت. طول این بردار، به طول تعداد کل کلمات کلیدی پایگاه داده ( $K$ ) است:

$$Y_j = \begin{bmatrix} t_{1j} \\ t_{2j} \\ \vdots \\ t_{Kj} \end{bmatrix} \quad (۳)$$

بنابراین اگر یک رکورد  $d$ ، به عنوان مثال اگر  $A_d$  مجموعه تمام کلمات کلیدی این رکورد باشد، آنگاه می‌توان یک بردار موضوعی برای این رکورد براساس کلمات کلیدی آن محاسبه کرد. برای این منظور می‌توانیم مجموع بردارهای موضوعی کلمات کلیدی این رکورد را به عنوان بردار موضوعی این رکورد در نظر بگیریم. بنابراین اگر  $A_d$  این بردار موضوعی باشد، به صورت زیر محاسبه می‌شود:

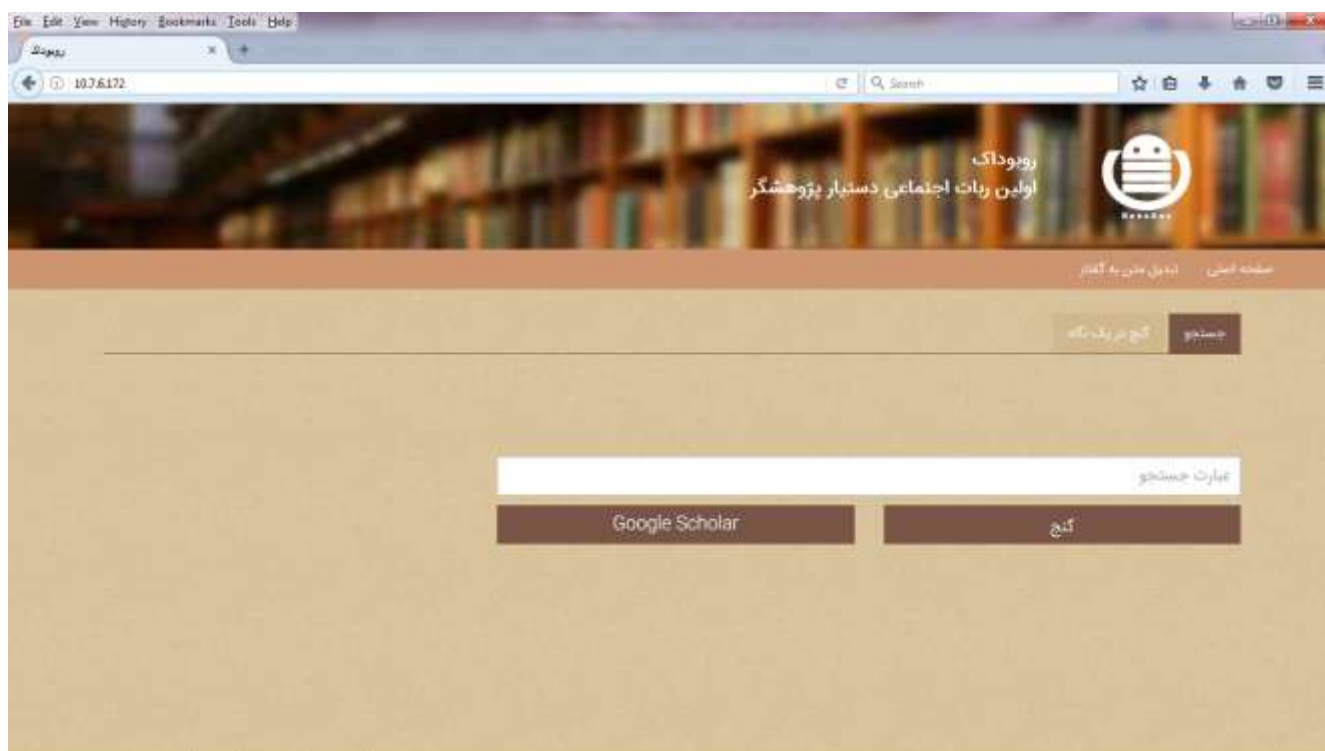
$$B_d = \frac{1}{|A_d|} \sum_{k_i \in A_d} X_i \quad (۴)$$

که  $|A_d|$  بیانگر تعداد اعضای مجموعه  $A_d$ ، یعنی تعداد کلمات کلیدی رکورد  $d$  است. این بردار برای موضوعات سطح دوم است و می‌توان برداری مشابه برای موضوعات سطح دوم نیز به تجميع این بردار حاصلی نمود. بنابراین با استفاده از این رویکرد می‌توان برای هر رکورد یک نمایش چند موضوعی (مانند شکل ۶-۱۳) را بدست آورد. در صفحه رابط کاربری علاوه بر نوار جستجو، بخشی به عنوان «گنج در یک نگاه» وجود دارد. این بخش نمای مصوری از اطلاعات گنج براساس موضوعات مختلف را نمایش می‌دهد که در شکل ۶-۱۳ آمده است. این نمودار یک نمودار تعاملی است که با کلیک کردن روی هر موضوع اصلی یک نمودار جدید با زیر موضوعات آن نمایش داده می‌شود. با کلیک روی موضوع فنی و مهندسی، یک نمای جدیدی از موضوع فنی و مهندسی ایجاد می‌شود که در نمودار زیرین نمایش داده شده است.



شکل ۶-۱۳- نمودار گنج در یک نگاه

در شکل ۶-۱۴ نمایشی از صفحه رابط کاربری زیر سیستم جستجو آمده است. نتایج جستجوی عبارت «فناوری اطلاعات» در شکل ۶-۱۵ آمده است. همانطور که مشاهده می‌شود، علاوه بر عبارت پرس‌جو، برای هر رکورد بازایی شده، گروه‌بندی موضوعات اصلی نیز در کنار آن نمایش داده شده است. همچنین می‌توان در یک بخش جداگانه، نتایج جستجو را نیز در قالب منوی «نتایج جستجو در یک نگاه»، در قالب نمودار مشاهده نمود که نمونه‌ای از آن در شکل ۶-۱۶ آمده است.



شکل ۶-۱۴- نمایش صفحه رابط کاربری اصلی



شکل ۶-۱۵- نمایش نتایج جستجو همراه با نمودار موضوعی هر رکورد



شکل ۶-۱۶- نمایش نتایج جستجو در یک نگاه

## **فصل هفتم:**

# **توسعه قابلیت‌های روبوداک در آینده**

## ۷. توسعه قابلیت‌های روباتک در آینده

این طرح به عنوان یک مرحله مقدماتی برای توسعه روباتک است. در این فصل ابتدا برنامه‌های و طرح‌های پیشنهادی برای توسعه روباتک در آینده ارائه می‌شود. سپس اولویت‌بندی آنها براساس مجموعه‌ای از معیارها مشخص خواهد شد.

### ۷-۱. معرفی پروژه‌های پیشنهادی برای توسعه روباتک

برای توسعه روباتک، مجموعه‌ای از پروژه‌ها برای ایجاد خدمات و قابلیت‌های جدید و ارتقاء قابلیت‌های کنونی، طراحی شده است. این پروژه‌ها براساس مجموعه خدمات طراحی شده برای روباتک که در فصل دوم به آنها اشاره شد، در جلسات کارشناسانه متشکل از متخصصین این حوزه تعریف شده است.

پروژه‌های پیشنهادی، در دو دسته اصلی قرار می‌گیرند:

۱. پروژه‌هایی که مستقیماً منجر به توسعه خدماتی برای روباتک می‌شوند
۲. پروژه‌هایی که در راستای توسعه قابلیت‌های تعاملی ربات با محیط و ارتقاء سیستم شناختی و حرکتی آن هستند.

نتیجه هر یک از پروژه‌های نوع اول، توسعه یک یا چند خدمت روباتک در مراحل مختلف پژوهش است. در پروژه‌های نوع دوم، تمرکز روی توسعه قابلیت‌های عمومی ربات و ارتقاء سیستم تعاملی آن است. علاوه بر آن می‌توان از پلتفرم طراحی شده، فارغ از کاربرد دستیار پژوهشگر، برای کاربردهای دیگری نیز استفاده نمود. در جدول ۷-۱ و ۷-۲ به ترتیب فهرست پیشنهادی پروژه‌های نوع اول و نوع دوم آمده است.

**جدول ۱-۲- پروژه‌های توسعه خدمات روبات‌ک**

| مرحله پژوهش               | پروژه توسعه روبات‌ک                                                                                         |
|---------------------------|-------------------------------------------------------------------------------------------------------------|
| خلق ایده                  | ایجاد قابلیت کمک به ایده پردازی: کمک به توسعه، تکامل و سازماندهی ایده‌های پژوهشگر                           |
| طراحی پژوهش               | ایجاد قابلیت کمک به انتخاب روش پژوهش: کمک در انتخاب متدولوژی و ابزارهای پژوهش و طراحی شیوه جمع آوری اطلاعات |
| جمع آوری اطلاعات          | ایجاد قابلیت جستجوی هم زمان در موتورهای علمی و پایگاه دانش روبات‌ک                                          |
|                           | توسعه کیفیت جستجوی کاربر از طریق ایجاد قابلیت پیشنهاد کلیدواژه براساس اصطلاح‌نامه یا آنتولوژی               |
|                           | ایجاد قابلیت بازنمایی نتایج جستجو از طریق دسته‌بندی، تحلیل ساختاری، مصورسازی و سازماندهی متون بازیابی شده   |
|                           | ایجاد قابلیت ذخیره و بازیابی داده‌های جستجو به صورت بافت-آگاه (Context-aware)                               |
|                           | ایجاد قابلیت ارزیابی و اعتبارسنجی رکوردهای علمی                                                             |
| سازماندهی و تحلیل اطلاعات | ایجاد قابلیت مدیریت منابع و ارجاعات پژوهش                                                                   |
|                           | ایجاد قابلیت سازماندهی هوشمند اطلاعات بر اساس ترجیحات پژوهشگر                                               |
|                           | ایجاد قابلیت مصورسازی پایگاه دانش پژوهشگر با استفاده از تحلیل شبکه مولفان، واژگانی و سایر ابزارهای علم‌سنجی |
|                           | ایجاد قابلیت ارزیابی عملکرد پژوهشی و تاثیر پژوهشگر                                                          |
| تولید گزارش               | ایجاد قابلیت ویرایش ادبی و علمی متن                                                                         |
|                           | ایجاد قابلیت کنترل اصالت گزارش (تشابه یابی متن)                                                             |
|                           | ایجاد قابلیت دسترسی پژوهشگر در تایپ، قالب بندی گزارش، مدیریت ارجاعات و صفحه‌بندی و انتخاب مجله              |

**جدول ۲-۲- پروژه‌های توسعه قابلیت‌های تعاملی، شناختی و حرکتی، و سایر کاربردها**

| نوع قابلیت‌ها   | پروژه توسعه روبات‌ک                                                                                                                    |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------|
| شناختی و تعاملی | ارتقا سیستم تشخیص هویت به شکل چند وجهی ( استفاده از صوت، تصویر و داده های بایومتریک به صورت هم زمان)                                   |
|                 | ارتقا سطح هوشیاری ربات نسبت به محیط                                                                                                    |
|                 | ارتقا زیرسیستم مدیریت گفتگو و تعامل صوتی روبات‌ک                                                                                       |
|                 | ایجاد قابلیت تعامل روبات‌ک با سایر ربات‌ها                                                                                             |
|                 | ایجاد قابلیت ردیابی مقالات پژوهشگر و اطلاع رسانی حوزه‌ها و موضوعات پژوهشی تازه                                                         |
| حرکتی           | ایجاد قابلیت دسترسی در فعالیت‌های آزمایشگاهی به صورت فیزیکی                                                                            |
|                 | ارتقا قابلیت حرکتی-تعاملی روبات‌ک در ارتباط با انسان (ایما و اشاره، تشخیص موانع، رفتار عاطفی)                                          |
| سایر کاربردها   | طراحی پلتفرم‌های تخصصی برای روبات‌ک در کاربردهای خاص، نمونه: بانکداری برای امور مشتریان، بانکداری مجازی، آموزش به کودکان، هتلداری،.... |

## ۲-۷. شرح پروژه‌ها

در این بخش شرح مختصری برای هر یک از پروژه‌های پیشنهادی ارائه می‌شود. در این شرح علاوه بر توصیف پروژه، با یک نگاه سیستمی، ورودی‌های و داده‌های اصلی که برای عملیاتی شدن یک خدمت یا قابلیت مورد نیاز است، پردازش‌های لازم برای تولید آن قابلیت یا خدمت و در نهایت نتایج مورد انتظار پس از بکارگیری آن خدمت یا قابلیت نیز معرفی شده است.

| عنوان پروژه: ایجاد قابلیت کمک به ایده پردازی: کمک به توسعه، تکامل و سازماندهی ایده های پژوهشگر                                                                                                                                                                                                                                                                                                                                                                |                                            |                                                            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|------------------------------------------------------------|
| <p><b>شرح:</b> هر پژوهشی از یک ایده آغاز می‌شود که ممکن است در ذهن پژوهشگر به صورت پراکنده و غیرمنسجم وجود داشته باشند. با استفاده از این قابلیت، ربات می‌تواند ایده‌های پژوهشگر را در قالب گفتاری، عکس‌برداری از ایده‌های مکتوب پژوهشگر و نیز دریافت از پایگاه ایده‌ها و پیشینه‌ی جستجوی پژوهشگر دریافت کند، پردازش‌های لازم از جمله پردازش صوت، تصویر و متن بر روی آنها انجام دهد و خروجی را به صورت ساخت‌یافته و گرافیکی بر روی صفحه‌ی نمایش نشان دهد.</p> |                                            |                                                            |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                 | پردازش‌های اصلی                            | خروجی‌ها و نتایج مورد انتظار                               |
| گفتار، تصویر ایده، داده‌های جستجو، پایگاه ایده‌ها                                                                                                                                                                                                                                                                                                                                                                                                             | پردازش تصویر و صوت، پردازش متن، تحلیل گراف | ارائه ایده‌های ثبت شده به صورت صوتی، نمایش گرافیکی ایده‌ها |

| عنوان پروژه: ایجاد قابلیت کمک به انتخاب روش پژوهش: کمک در انتخاب متدولوژی و ابزارهای پژوهش و طراحی شیوه جمع‌آوری اطلاعات                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                            |                                                                |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|----------------------------------------------------------------|
| <p><b>شرح:</b> با توجه به این موضوع که روباتک یک ربات پژوهشیار است، توانایی او در پشتیبانی از راهبری پژوهش بسیار حائز اهمیت است. اولین و مهمترین بخش یک پژوهش انتخاب روش پژوهش و ابزارهای پژوهشی مورد استفاده در روش مورد نظر است. برای تحقق این هدف روباتک به یک پایگاه دانش روش پژوهش و همچنین یک موتور استنتاج و کمک‌تصمیم‌مجهز خواهد شد. پژوهشگر در ارتباط گفتاری با ربات نوع پژوهش و نیازهای پژوهشی خود را بیان می‌کند و روباتک از طریق گفتار و نمایش مفاهیم و مطالب او را راهنمایی خواهد کرد. پس از انتخاب روش پژوهش روباتک ابزارهای پژوهشی روش مورد نظر را معرفی و در صورت نیاز به جمع‌آوری اطلاعات برای استفاده از این ابزارها، فرم‌ها و نمونه سوالات را در اختیار پژوهشگر قرار خواهد داد. خروجی این قابلیت روباتک می‌تواند به صورت یک سرویس مستقل توسط ایرانداک عرضه شود.</p> |                                                            |                                                                |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | پردازش‌های اصلی                                            | خروجی‌ها و نتایج مورد انتظار                                   |
| گفتار (به صورت تعاملی)، پایگاه روش‌های پژوهش، نمونه پژوهش‌های انجام شده، پایگاه پژوهش‌های پژوهشگر                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | درخت تصمیم، پردازش صوت، استدلال نمونه بنیان <sup>۱۰۵</sup> | ارائه روش پژوهش (به صورت صوتی)، نمایش گرافیکی و متنی روش پژوهش |



| عنوان پروژه: ایجاد قابلیت جستجوی هم زمان در موتورهای علمی و پایگاه دانش روبوداک                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                        |                                                                                                  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| <p><b>شرح:</b> از مشکلاتی که پژوهشگر در جریان پژوهش خود با آن دست به گریبان است، نیاز به جستجو در موتورهای جستجو علمی مختلف است. همچنین پژوهشگر نمی تواند نتایج جستجو فعلی را با اطلاعات قبلی، و یا نتایج پژوهش های پیشین خود تطبیق دهد. برای رفع این مشکل روبوداک به یک فرآیند جستجوگر مجهز خواهد شد که نتایج جستجو در پایگاه های اطلاعات علمی مختلف و همچنین سامانه گنج ایرانداک را به صورت آمیخته و دسته بندی شده ارائه خواهد کرد. همچنین پایگاه دانش روبوداک که حاوی ترجیحات و پیشینه جستجو و پژوهش های پژوهشگر است به عنوان یک مرجع مورد استفاده خواهد بود. این قابلیت روبوداک می تواند به عنوان یک سرویس ارزش افزوده بر روی سامانه گنج پژوهشگاه قرار گیرد.</p> |                                                                                                        |                                                                                                  |
| ورودی ها و داده های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | پردازش های اصلی                                                                                        | خروجی ها و نتایج مورد انتظار                                                                     |
| گفتار، پیشینه جستجوهای پژوهشگر                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | پردازش متن فارسی و انگلیسی، خلاصه سازی خودکار، پردازش صوت، روش های امتیازدهی و اولویت بندی نتایج جستجو | ارائه نتایج جستجو، نمایش مصور نتایج جستجو، عکس العمل حرکتی ربات نسبت به کیفیت و کمیت نتایج جستجو |

| عنوان پروژه: توسعه کیفیت جستجوی کاربر از طریق ایجاد قابلیت پیشنهاد کلیدواژه براساس اصطلاح نامه یا آنتولوژی                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                   |                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------|
| <p><b>شرح:</b> این قابلیت با پیاده سازی امکان پیشنهاد کلیدواژه ها بر اساس اصطلاح نامه یا هستان شناسی، به ارتقای تجربه ی جستجوی کاربر در سامانه ی روبوداک خواهد انجامید. با استفاده از این قابلیت، داده ها می توانند به صورت گفتاری یا درون سیستمی (از پایگاه هایی نظیر اصطلاح نامه ها و پیشینه ی جستجوهای پژوهشگر) وارد گردند. سپس پردازش زبان گفتاری و نوشتاری بر روی آنها صورت گرفته و محاسبات مربوط به پیش بینی واژه ها انجام خواهد شد. نتایج می توانند به صورت صوتی یا به صورت مصور بر روی صفحه نمایش نشان داده شوند.</p> |                                                                                   |                                           |
| ورودی ها و داده های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | پردازش های اصلی                                                                   | خروجی ها و نتایج مورد انتظار              |
| گفتار، اصطلاح نامه ها و روابط معنایی واژگان، پیشینه جستجوهای پژوهشگر                                                                                                                                                                                                                                                                                                                                                                                                                                                          | پردازش متن، پردازش صوت، پیشنهادگری خودکار <sup>۱۰۶</sup> و الگوریتم های پیشنهادگر | ارائه نتایج جستجو، نمایش مصور نتایج جستجو |

| عنوان پروژه: ایجاد قابلیت بازنمایی نتایج جستجو از طریق دسته‌بندی، تحلیل ساختاری، مصورسازی و سازماندهی متون بازایی شده                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                        |                                                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|------------------------------------------------------------------------------|
| <p><b>شرح:</b> با استفاده از این قابلیت، نتایج جستجو به پژوهشگر در قالب مدل‌های و شیوه‌های مصورسازی اطلاعات نمایش داده می‌شود. این قابلیت به پژوهشگر در مرحله جمع‌آوری اطلاعات کمک می‌کند تا بتواند تحلیل مناسبی از رکوردها و متون بازایی شده بدست آورد. نمایش مصور نتایج جستجو در قالب روش‌های دسته‌بندی متون، در برخی از موتورهای جستجو اکتشافی دیده می‌شود. برخی از این نمونه‌ها عبارتند از Oskope InfoCrystal و TileBars. موتور جستجوی اکتشافی بیشتر برای شرایطی کاربرد دارد که کاربر عبارت پرس‌وجو مشخص و دقیقی ندارد. در این موارد معمولاً با چند کلمه ساده جستجو آغاز می‌شود، سپس با نمایش مصور اطلاعات و نتایج جستجو در قالب نمودارهای تعاملی، کاربر می‌تواند نتایج را در قالب نمودارها، گراف‌ها و نقشه‌های مفهومی مشاهده نماید. سپس با استفاده از ابزارهایی نظیر فیلتر کردن نتایج جستجو براساس یک ویژگی، عبارت پرس‌وجوی خود را دقیق‌تر نماید.</p> |                        |                                                                              |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | پردازش‌های اصلی        | خروجی‌ها و نتایج مورد انتظار                                                 |
| اصطلاح‌نامه‌ها و روابط معنایی واژگان،<br>پیشینه جستجوهای پژوهشگر                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | پردازش متن، تحلیل گراف | ارائه نتایج جستجو، نمایش مصور نتایج جستجو، بازنمایی نتایج جستجو در قالب گراف |

| عنوان پروژه: ایجاد قابلیت ذخیره و بازایی داده‌های جستجو به صورت بافت-آگاه <sup>۱۰۷</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                 |                                                                              |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|------------------------------------------------------------------------------|
| <p><b>شرح:</b> موتورهای جستجوی بافت آگاه، در بازایی اطلاعات از اطلاعات دیگری به جز خود عبارت پرس‌وجو نیز استفاده می‌کنند که به آنها داده‌های محیطی نیز می‌گویند. براساس این قابلیت روبوداک می‌تواند اطلاعات مربوط به جستجوهای پیشین پژوهشگر را همراه با داده‌های محیطی نظیر، زمان و مکان جستجو، موقعیت سایر وسایل هوشمند ارتباطی که در محیط تعامل کاربر در حال فعالیت است، عملکرد و حالت ارتباطی فرد با سیستم، صوت یا حالات فیزیکی کاربر، ذخیره‌سازی نماید تا با استفاده از این اطلاعات در نهایت خدمات بهتری را به پژوهشگر ارائه دهد.</p> |                                                                 |                                                                              |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | پردازش‌های اصلی                                                 | خروجی‌ها و نتایج مورد انتظار                                                 |
| گفتار، تصویر پژوهشگر برای شناسایی وی، شرایط محیطی،<br>پیشینه جستجوهای پژوهشگر،<br>اطلاعات حسگرها، اطلاعات لاگ <sup>۱۰۸</sup> جستجو                                                                                                                                                                                                                                                                                                                                                                                                        | داده کاوی، پردازش تصویر، پردازش داده‌های چندوجهی <sup>۱۰۹</sup> | ارائه نتایج جستجو، نمایش مصور نتایج جستجو، بازنمایی نتایج جستجو در قالب گراف |

<sup>107</sup> Context-aware<sup>108</sup> Log<sup>109</sup> Multi-modal

| عنوان پروژه: ایجاد قابلیت ارزیابی و اعتبارسنجی رکوردهای علمی                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                             |                                                                                               |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| <p><b>شرح:</b> با ایجاد این قابلیت، امکان ارزیابی و اعتبارسنجی رکوردهای علمی در فضای وب یا پایگاه‌های اطلاعاتی علمی فراهم می‌شود. در حال حاضر حجم وسیعی از اطلاعات علم در پایگاه‌های داده علمی موجود است که برخی از آنها از اعتبار کمی برخوردار هستند یا ماهیت جعلی دارند. با استفاده از این قابلیت می‌توان رکوردهای علمی را براساس معیارهایی نظیر محل انتشار، نویسندگان، تعداد استنادها، تعداد دفعات بازیابی و ذخیره‌سازی، و موسسه نویسندگان ارزیابی نمود و امتیازی را برای آنها در نظر گرفت.</p> |                                                             |                                                                                               |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | پردازش‌های اصلی                                             | خروجی‌ها و نتایج مورد انتظار                                                                  |
| گفتار، فراداده کیفیت و اعتبار رکوردهای علمی                                                                                                                                                                                                                                                                                                                                                                                                                                                        | پردازش الگوهای تصمیم، پردازش متن، پردازش صوت، یادگیری ماشین | نمایش نتایج ارزیابی و امتیاز رکوردهای علمی، عکس‌العمل ربات نسبت به کیفیت رکوردهای بازیابی شده |

| عنوان پروژه: ایجاد قابلیت مدیریت منابع و ارجاعات پژوهش                                                                                                                                                                                                                |                                                      |                              |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|------------------------------|
| <p><b>شرح:</b> در حال حاضر سیستم‌ها و ابزارهای متعددی برای مدیریت ارجاعات و منابع پژوهشی در متون علمی وجود دارد، مانند Endnote، Zotero، Mendely، یا نمونه‌های فارسی نظیر پژوهیار وجود دارد. این قابلیت امکان یکپارچه‌سازی این ابزارها را در روبوداک فراهم می‌کند.</p> |                                                      |                              |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                         | پردازش‌های اصلی                                      | خروجی‌ها و نتایج مورد انتظار |
| گفتار، تصویر نسخه چاپی منابع علمی، کتابخانه های منابع علمی پژوهشگر، داده‌های مرجع                                                                                                                                                                                     | پردازش متن، پردازش صوت، پردازش تصویر با تمرکز بر OCR | نمایش کتابخانه پژوهشگر       |

| عنوان پروژه: ایجاد قابلیت سازماندهی هوشمند اطلاعات بر اساس ترجیحات پژوهشگر                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                        |                              |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|------------------------------|
| <p><b>شرح:</b> از مهمترین بخش‌های یک پژوهش و همچنین از مهمترین عوامل موفقیت یک پژوهشگر سازماندهی اطلاعات پژوهشی و استفاده موثر از آنهاست. با افزایش حجم اطلاعات و گذشت زمان، قدرت سازماندهی اطلاعات در پژوهشگر کاهش می‌یابد. برای کمک به بهبود این وضعیت روبوداک به یک زیرسیستم سازماندهی هوشمند اطلاعات مجهز خواهد شد. در واقع اطلاعات پژوهشی در پایگاه دانش روبوداک از منظرهای متفاوتی نمایه می‌شود و از انواع اصطلاح‌نامه‌های تخصصی و همچنین آنتولوژی‌های سفارشی برای نمایه‌سازی و بازیابی اطلاعات استفاده می‌گردد. برای بازیابی و بازنمایی اطلاعات پژوهشی از الگوریتم‌های هوشمند بازیابی اطلاعات استفاده خواهد شد. این قابلیت با توجه به یادگیرنده بودن ربات، به مرور زمان و با توجه به ترجیحات و بازخوردهایی که پژوهشگر به روبوداک می‌دهد بهبود می‌یابد. توسعه چنین امکانی در ربات، بستر مناسبی برای توسعه خدمات ارزش افزوده‌ای همچون، نمایه‌سازی ماشینی، جستجوی هوشمند، سازماندهی بر پایه آنتولوژی و تحلیل و بازنمایی اطلاعات در ایرانداک را فراهم خواهد کرد.</p> |                        |                              |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | پردازش‌های اصلی        | خروجی‌ها و نتایج مورد انتظار |
| گفتار، ترجیحات پژوهشگر، پیشینه جستجوهای پژوهشگر                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | پردازش متن، تحلیل گراف | نمایش اطلاعات سازماندهی شده  |

| عنوان پروژه: ایجاد قابلیت مصورسازی پایگاه دانش پژوهشگر با استفاده از تحلیل شبکه مولفان، واژگانی و سایر ابزارهای علم‌سنجی                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                    |                              |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|------------------------------|
| <p><b>شرح:</b> امروزه با توجه به حجم بالای اطلاعات علمی که روزانه در حال تولید است، ردیابی و کشف روندهای پژوهشی و مطالعه ادبیات یک موضوع پژوهشی پیچیده‌تر از گذشته شده است. ابزارهای علم‌سنجی علاوه بر اینکه می‌توانند برای مطالعه وضعیت حوزه‌های مختلف علمی مورد استفاده قرار گیرند، به پژوهشگران این امکان را می‌دهند که مسیر خود را در انبوه مقالات و متون علمی پیدا کنند. از مهمترین ابزارهای علم‌سنجی که برای این مهم مورد استفاده قرار می‌گیرند، تحلیل شبکه مولفان، تحلیل شبکه‌های واژگانی و تحلیل استنادات است. در این پروژه قصد داریم روبوداک را به این ابزارها مجهز کنیم، تا به پژوهشگر در تحلیل اطلاعات جمع‌آوری شده و همچنین شناخت دانشمندان موثر و پراستناد، مقالات و مجلات مهم در حوزه پژوهشی مورد نظر یاری رساند. خروجی چنین قابلیت به طور مستقل می‌تواند یک خدمت ارزش افزوده برای سیستم‌های اطلاعات علمی پژوهشگاه به حساب آید و به طور مشخص بر روی پایگاه پایان‌نامه‌ها و رساله‌ها (گنج) نصب گردد.</p> |                                    |                              |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | پردازش‌های اصلی                    | خروجی‌ها و نتایج مورد انتظار |
| گفتار، پایگاه دانش پژوهشگر، ترجیحات پژوهشگر، پایگاه ایده‌ها                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | پردازش متن، تحلیل گراف، پردازش صوت | نمایش مصور اطلاعات           |

| عنوان پروژه: ایجاد قابلیت ارزیابی عملکرد پژوهشی و تاثیر پژوهشگر                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                    |                                                      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------|
| <p><b>شرح:</b> امروزه در دنیای آکادمیک، کیفیت آثار پژوهشی مهم‌تر از کمیت آن تلقی می‌شود که می‌توان آن را با بررسی تاثیر آثار پژوهشگر یا عملکرد پژوهشی وی سنجید. یکی از مکاناتی که روبوداک می‌تواند به‌عنوان دستیار پژوهش در اختیار کاربران قرار دهد، امکان رصد عملکرد پژوهشی پژوهشگر است. داده‌های موردنیاز در اجرای این قابلیت، داده‌های گفتاری و نیز داده‌های پایگاه‌های استنادی خواهد بود. سپس داده‌ها پردازش زبانی و تحلیل گراف خواهند شد و نتایج حاصل از تحلیل وضعیت پژوهشی کاربر به‌صورت گرافیکی از جمله در قالب درخت دانش نمایش داده خواهد شد.</p> |                                    |                                                      |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | پردازش‌های اصلی                    | خروجی‌ها و نتایج مورد انتظار                         |
| گفتار، داده‌های پایگاه‌های استنادی                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | پردازش متن، تحلیل گراف، پردازش صوت | نمایش عملکرد پژوهشی پژوهشگر، نمایش درخت دانش پژوهشگر |

| عنوان پروژه: ایجاد قابلیت ویرایش ادبی و علمی متن                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                           |                                                                                                  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| <p><b>شرح:</b> این قابلیت جزو قابلیت‌های کلیدی روبوداک ذیل خدمت تولید گزارش است. برای اینکه بتوان گزارش تولیدشده را به‌لحاظ ادبی و علمی ویرایش کرد، ابتدا باید داده‌ها به‌صورت گفتاری، متن الکترونیکی یا متن تصویری به سامانه وارد شوند. سپس پردازش‌های مربوطه از جمله پردازش متن، صوت، بازشناسی حروف (OCR) و استدلال نمونه‌بنیان انجام خواهد گرفت. خروجی پردازش‌ها می‌تواند به‌صورت گفتاری (درباره‌ی صحت متن) یا تصویری (مشخص کردن موارد ویرایشی و نمایش متن تصحیح‌شده) باشد.</p> |                                                                           |                                                                                                  |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                      | پردازش‌های اصلی                                                           | خروجی‌ها و نتایج مورد انتظار                                                                     |
| گفتار، تصویر متن دست‌نویس، نمونه داده‌های متون علمی                                                                                                                                                                                                                                                                                                                                                                                                                                | پردازش متن، پردازش صوت، پردازش تصویر با تمرکز بر OCR، استدلال نمونه‌بنیان | گفتار درباره صحت متن، موارد ویرایشی، متن اصلاح‌شده، عکس‌العمل فیزیکی ربات نسبت به کیفیت ادبی متن |

| عنوان پروژه: ایجاد قابلیت کنترل اصالت گزارش (تشابه‌یابی متن)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                          |                                                                            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------|----------------------------------------------------------------------------|
| <p><b>شرح:</b> این قابلیت امکان وجود تشابه احتمالی بین گزارش پژوهشگر و مکتوبات پایگاه‌های علمی را بررسی می‌کند. ورودی این سامانه، داده‌های متنی پژوهشگر و نیز داده‌های درون‌سیستمی (از جمله داده‌های پایگاه‌های اطلاعات علمی و استنادی) است. داده‌ها پس از ورود به سیستم، تحت پردازش متن، تشخیص الگو و پردازش نحوی قرار می‌گیرند. در نهایت، نتایج به‌دست‌آمده به‌صورت تصویری و در قالب نمایش تشابهات و نمایش شاخص‌های میزان اصالت متن نشان داده خواهند شد. همچنین پیش‌بینی شده که ربات می‌تواند واکنش فیزیکی نیز به نتایج بررسی مشابهت متون داشته باشد.</p> |                                          |                                                                            |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | پردازش‌های اصلی                          | خروجی‌ها و نتایج مورد انتظار                                               |
| گفتار، داده‌های متنی، داده‌های پایگاه‌های اطلاعات علمی و استنادی، داده‌های میان‌زبانی                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | پردازش متن، تشخیص الگو، پردازش نحوی متون | نمایش تشابهات، نمایش شاخص‌های میزان اصالت متن، عکس‌العمل نسبت به اصالت متن |

| عنوان پروژه: ایجاد قابلیت دستیاری پژوهشگر در تایپ، قالب‌بندی گزارش، مدیریت ارجاعات، و صفحه‌بندی و انتخاب مجله                                                                                                                                                                                                                                                                                                                                                                               |                                    |                              |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|------------------------------|
| <p><b>شرح:</b> بر اساس این قابلیت، روبوداک قادر خواهد بود آماده‌سازی‌های نهایی گزارش از جمله تایپ، صفحه‌بندی، کنترل ارجاعات و انتخاب مجله‌ی مناسب برای انتشار گزارش را انجام دهد. داده‌های ورودی عمدتاً به‌صورت گفتاری، کتابخانه‌ی قالب‌های معین و نیز کتابخانه‌های منابع علمی وارد سامانه خواهد شد. سپس پردازش‌های مربوطه از جمله پردازش زبان، پردازش صوت و تشخیص الگو بر روی داده‌ها انجام خواهد گرفت. در انتها، نتایج پردازش‌ها به‌صورت تصویری بر روی صفحه نمایش نشان داده خواهد شد.</p> |                                    |                              |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                               | پردازش‌های اصلی                    | خروجی‌ها و نتایج مورد انتظار |
| گفتار، کتابخانه‌های منابع علمی پژوهشگر، کتابخانه قالب‌ها                                                                                                                                                                                                                                                                                                                                                                                                                                    | پردازش متن، تشخیص الگو، پردازش صوت | نمایش متن گزارش              |

| عنوان پروژه: ایجاد قابلیت ردیابی مقالات پژوهشگر و اطلاع رسانی حوزه‌ها و موضوعات پژوهشی تازه (ترجیحات پژوهشگر)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                        |                                                                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|-----------------------------------------------------------------------|
| <p><b>شرح:</b> امروزه خدمت اشاعه اطلاعات گزیده (SDI) به یکی از مهمترین خدمات بنگاه‌های اطلاع‌رسانی و موتورهای جستجوی اطلاعات علم و فناوری بدل گشته است. البته قالب این خدمات به جستجوی یک کلیدواژه به صورت مستمر، فرستادن اطلاعات یک رخداد یا مجله خاص یا ردیابی انتشارات یک فرد خاص برای کاربران محدود می‌گردد. با توجه به حجم بالای انتشارات و وقت گیر بودن فرایند جستجو در صورتی که این خدمت هوشمندانه و موثر ارائه گردد، کمک بزرگی به پژوهشگر خواهد بود. در این پژوهش کاربردی روبوداک به یک موتور اشاعه اطلاعات گزیده هوشمند مجهز خواهد شد. این موتور امکان ردیابی ترجیحات پژوهشگر در قالب جستجوی کلیدواژه، اطلاع‌رسانی رخدادهای علمی، ردیابی انتشار مجلات و همچنین ردیابی ارجاع به مقالات پژوهشگر را ایجاد خواهد کرد. نتایج این ردیابی‌ها می‌تواند در قالب‌های مختلف متنی و بصری به کاربر روبوداک ارائه گردد. نتیجه این پژوهش کاربردی می‌تواند در قالب یک ماژول اشاعه اطلاعات گزیده در پورتال ایرانداک و سامانه گنج قرار گیرد.</p> |                                        |                                                                       |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | پردازش‌های اصلی                        | خروجی‌ها و نتایج مورد انتظار                                          |
| گفتار، پایگاه‌های اطلاعات علمی، ترجیحات پژوهشگر                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | پردازش متن، الگوهای خزش <sup>۱۱۰</sup> | آگاه سازی صوتی، نمایش رکوردهای جدید علمی، نمایش عملکرد پژوهشی پژوهشگر |

| عنوان پروژه: ارتقا سیستم تشخیص هویت به شکل چند وجهی (استفاده از صوت، تصویر و داده‌های بایومتریک به صورت هم زمان)                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                        |                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|------------------------------|
| <p><b>شرح:</b> روبوداک می‌تواند از طرق مختلفی کاربر را شناسایی کند و با وی تعامل نماید. در حال حاضر سیستم تشخیص هویت فرد، براساس زیرسیستم بینایی و با استفاده از الگوریتم‌های تشخیص چهره انجام می‌شود. این سیستم می‌تواند توسعه یابد تا اطلاعات دیگری را نظیر صوت و داده‌های اثر انگشت نیز برای افزایش دقت تشخیص بکار بندد. بنابراین لازم است که کلیه این داده‌های که از سنسورها و وسایل مختلفی توسط سیستم روبوداک دریافت شده‌اند، پردازش شوند و با هم یکپارچه گردند تا در نهایت براساس کلیه این اطلاعات دقت سیستم تشخیص هویت افزایش یابد.</p> |                                                                        |                              |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | پردازش‌های اصلی                                                        | خروجی‌ها و نتایج مورد انتظار |
| گفتار، تصویر پژوهشگر از طریق دوربین، اثر انگشت از طریق سنسورهای مختلف و دوربین، سایر داده‌های بایومتریک یک فرد                                                                                                                                                                                                                                                                                                                                                                                                                                 | پردازش تصویر و صوت، تلفیق داده‌های چندوجهی، الگوریتم‌های کدینگ اطلاعات | تشخیص هویت فرد               |

| عنوان پروژه: ارتقا سطح هوشیاری ربات نسبت به محیط                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                |                                         |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|-----------------------------------------|
| <p><b>شرح:</b> در حال حاضر روبوداک قابلیت تشخیص صورت‌های مختلف را در تصاویری که با استفاده از دوربین خود دریافت می‌کند، دارد. بنابراین از یک سطح هوشیاری نسبت به محیط براساس حضور یا عدم حضور یک فرد در دامنه دید خود برخوردار است. همچنین اگر کاربر از طریق مسکروفون یا آن صحبت کند، آن را درک می‌کند و پاسخ مناسب را ارائه می‌دهد. برای ارتقاء سطح هوشیاری روبوداک می‌تواند علاوه بر پردازش چهره، نسبت به سایر اشیاء موجود در محیط نیز شناخت پیدا کند و سایر صوت‌های موجود در محیط را نیز پردازش و تحلیل نماید. همچنین می‌توان از سنسورهای شناختی دیگری نظیر سنسورهای دما، سنسورهای تشخیص ارتعاشات و نور نیز استفاده نمود تا روبوداک درک بهتری نسبت به محیط اطراف خود پیدا کند.</p> |                                                                                                |                                         |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | پردازش‌های اصلی                                                                                | خروجی‌ها و نتایج مورد انتظار            |
| اطلاعات سنسورهای مختلف، گفتار، تصویر                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | پردازش صوت و تصویر، پردازش اطلاعات سنسورها، الگوریتم‌های کدینگ اطلاعات، تلفیق داده‌های چندوجهی | عکس العمل (فیزیکی) روبوداک نسبت به محیط |

| عنوان پروژه: ارتقا زیرسیستم مدیریت گفتگو و تعامل صوتی روبوداک                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                               |                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|--------------------------------------------------------|
| <p><b>شرح:</b> روبوداک در حال حاضر قابلیت تشخیص گفتار فارسی را دارد و می‌تواند با کاربر در قالب مکالمه از پیش مشخص شده تعامل داشته باشد. قابلیت‌های تشخیص گفتار روبوداک در این پروژه افزایش می‌یابد و خطاهای احتمالی در تشخیص گفتار کاهش می‌یابد. با ارتقا زیرسیستم مدیریت گفتگو می‌توان یک مدل تعاملی برای روبوداک مشخص نمود که در قالب این مدل تعاملی، روبوداک بتواند به راحتی با کاربر تعامل فعالانه داشته باشد. در نهایت با استفاده از این مدل لازم نیست که برای هر نوع سناریوی تعاملی جملات از پیش مشخص شده وجود داشته باشد، بلکه روبوداک می‌تواند براساس این مدل و قابلیت‌های تصمیم‌گیری جملات ساده‌ای را برای کاربر بیان کند. مسلماً یکی از الزامات توسعه این قابلیت استفاده از مدل‌های زبانی و فناوری‌های تولید زبان طبیعی است.</p> |                                                                                               |                                                        |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | پردازش‌های اصلی                                                                               | خروجی‌ها و نتایج مورد انتظار                           |
| داده‌های تعاملی کاربر با روبوداک، گفتار                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | پردازش زبان طبیعی، پردازش صوت، تولید زبان طبیعی، تحلیل عواطف و تمایلات پژوهشگر <sup>۱۱۱</sup> | گفتار ربات، حرکات فیزیکی حین صحبت کردن برای بیان حالات |

<sup>۱۱۱</sup> Sentiment analysis

| عنوان پروژه: ارتقا قابلیت حرکتی-تعاملی روبات‌ک در ارتباط با انسان (ایما و اشاره، تشخیص موانع، رفتار عاطفی)                                                                                                                                                                                                                                                                                           |                    |                                                  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|--------------------------------------------------|
| <p><b>شرح:</b> یکی از الزامات بلوغ رفتار تعاملی یک ربات اجتماعی، امکان تشخیص ایما و اشاره و حالات و عواطف فرد است. در حال حاضر روبات‌ک می‌تواند یک حرکت دست را بیانگر تایید یا عدم تایید است تشخیص دهد. برای ارتقا قابلیت‌های ربات برای تشخیص ایما و اشاره یا عواطف، لازم است که زیرسیستم بینایی ارتقاء یابد و زیر سیستم تشخیص صوت نیز بتواند اطلاعاتی نظیر لحن صدا و شیوه بیان آن را تحلیل کند.</p> |                    |                                                  |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                                                                                                                                                        | پردازش‌های اصلی    | خروجی‌ها و نتایج مورد انتظار                     |
| گفتار، تصاویر فرد و حرکات وی                                                                                                                                                                                                                                                                                                                                                                         | پردازش صوت و تصویر | نتیجه شناسایی حرکات فرد و انجام عملیات مورد نیاز |

| عنوان پروژه: ایجاد قابلیت دستیاری در فعالیت‌های آزمایشگاهی                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>شرح:</b> یکی از اهداف اصلی توسعه روبات‌ک اینست که ربات بتواند در انجام فعالیت‌های آزمایشگاهی که پژوهشگر خود قادر به حضور و انجام دادن آن نیست، کمک کند. انجام آزمایش‌ها مختل از راه دور بدون حضور فیزیکی را با عنوان دورآزمای می‌شناسند. برای انجام این آزمایش‌ها معمولاً لازم است که از ربات‌های مشخصی با قابلیت‌های حرکتی خاص استفاده شود. بنابراین برای توسعه این قابلیت‌ها لازم است که از پلتفرم‌های رباتیک دیگری با قابلیت‌های حرکتی متفاوت استفاده شود.</p> |

| عنوان پروژه: ایجاد قابلیت تعامل روبات‌ک با سایر ربات‌ها                                                                                                                                                                                                                     |                                                  |                                   |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|-----------------------------------|
| <p><b>شرح:</b> این قابلیت امکان شناسایی سایر ربات‌ها و برقراری تعامل با آنها را در محیط برقرار می‌سازد. حتی روبات‌ک می‌تواند با وسایل و ابزارهایی که می‌توانند از محیط پیام دریافت کنند یا پیامی به محیط ارسال کنند، ارتباط داشته باشد و برای آنها پیام‌هایی ارسال کند.</p> |                                                  |                                   |
| ورودی‌ها و داده‌های مورد نیاز                                                                                                                                                                                                                                               | پردازش‌های اصلی                                  | خروجی‌ها و نتایج مورد انتظار      |
| اطلاعات سنسورهای محیطی، صوت                                                                                                                                                                                                                                                 | پردازش صوت، پردازش تصویر، تلفیق داده‌های چندوجهی | تعامل ربات با محیط و سایر ربات‌ها |

مواردی که در جداول فوق بیان شد، به قابلیت‌های روبات‌ک مربوط می‌شود. لیکن با نگاهی وسیع‌تر پلتفرم ربات اجتماعی که به زبان فارسی تعامل می‌کند را می‌توان برای زمینه‌های دیگری نظیر آموزش، تبلیغات و بازاریابی و امور مشتریان در بانک‌ها، نیز تعمیم داد.

### ۳-۲. اولویت‌بندی پروژه‌های پیشنهادی برای توسعه خدمات روبات‌ک

در این بخش، براساس مجموعه‌ای از معیارها اولویت اجرایی پروژه‌های پیشنهادی مشخص می‌گردد. سپس امتیاز هر پروژه پیشنهادی نسبت به هر معیار مشخص می‌شود. براساس این معیارها عبارتند از:



□ محوریّت: مفهوم این معیار اینست که تا چه میزان یک پروژه برای توسعه روباتک و خدمات آن کلیدی محسوب می‌شود. در ازای پیاده‌سازی هر پروژه خدماتی برای روباتک توسعه داده می‌شود و بخشی از آنها خدمات کلیدی و محوری به حساب می‌آیند و برای توسعه سایر خدمات روباتک ضروری هستند.

□ هزینه: این معیار به مفهوم بار هزینه‌ای است که توسعه یک پروژه می‌تواند داشته باشد. امتیاز این معیار به صورت معکوس در نظر گرفته می‌شود، به این معنی که امتیاز بالا برای این معیار یعنی هزینه کم و برعکس.

□ سادگی پیاده‌سازی: این معیار بیانگر میزان پیچیدگی پروژه از لحاظ نوع فعالیت‌های آن و روش و الگوریتم‌های مورد نیاز برای توسعه آن است. امتیاز بالا برای این معیار به معنی اینست که اجرای پروژه از پیچیدگی کمی برخوردار است.

□ قابلیت تجاری‌سازی: قابلیت تجاری‌سازی به این مفهوم است که تا چه میزان با پیاده‌سازی پروژه می‌توان به راحتی محصول آن را تجاری نمود و حتی فارغ از بدنه فیزیکی ربات، آن را برای ارائه به بازار آماده نمود.

□ کلیدی بودن برای ایرانداک در حال حاضر: این معیار مشخص می‌کند که یک پروژه، برای ایرانداک طبق برنامه راهبردی و توسعه، تا چه میزان کلیدی است.

□ بلوغ فعلی: این معیار مشخص می‌کند که بلوغ فعلی روباتک تا چه حد در راستای انجام پروژه است. در واقع خدماتی که تا کنون برای روباتک توسعه پیدا کرده تا چه میزان می‌تواند نیازهای پروژه را پاسخ دهد.

برای هر یک از پروژه‌ها نسبت به معیارهای فوق یک امتیاز از ۰ تا ۱۰ تعلق می‌گیرد و در نهایت میانگین امتیازها از همه معیارها برای هر پروژه، امتیاز نهایی آن را مشخص می‌کند. پروژه‌های با بیشترین امتیاز در اولویت اول برای توسعه قرار دارند. برای امتیازدهی پروژه‌ها از نظرات مجموعه از متخصصین این حوزه (مشابه متخصصین بخش ۳-۲) استفاده شده است. نتایج امتیازها در جدول ۷-۳ و ۷-۴ آمده است. پروژه پیشنهادی برای حوزه سایر کاربردها در این اولویت‌بندی لحاظ نشده زیرا با حوزه چشم‌انداز روباتک فاصله دارد.

جدول ۷-۳- امتیاز پروژه‌های توسعه خدمات بر اساس معیارهای مختلف

| عنوان پروژه                                                                                                 | معیارهای اولویت بندی | محوریت | هزینه (صرفه اقتصادی) | سادگی پیاده سازی | بلوغ فعلی | قابلیت تجاری سازی | کلیدی بودن برای ایرانداک | امتیاز نهایی |
|-------------------------------------------------------------------------------------------------------------|----------------------|--------|----------------------|------------------|-----------|-------------------|--------------------------|--------------|
| ایجاد قابلیت کمک به ایده پردازی: کمک به توسعه، تکامل و سازماندهی ایده‌های پژوهشگر                           | ۴                    | ۴      | ۴                    | ۳                | ۱         | ۶                 | ۵                        | ۳/۸          |
| ایجاد قابلیت کمک به انتخاب روش پژوهش: کمک در انتخاب متدولوژی و ابزارهای پژوهش و طراحی شیوه جمع‌آوری اطلاعات | ۵/۵                  | ۷      | ۷                    | ۷                | ۱         | ۷/۵               | ۸                        | ۶            |
| ایجاد قابلیت جستجوی هم زمان در موتورهای علمی و پایگاه دانش روبوداک                                          | ۸                    | ۵      | ۵/۵                  | ۴/۵              | ۸         | ۸                 | ۸                        | ۶/۵          |
| توسعه کیفیت جستجوی کاربر از طریق ایجاد قابلیت پیشنهاد کلیدواژه براساس اصطلاح‌نامه یا آنتولوژی               | ۶/۵                  | ۶      | ۶                    | ۶/۵              | ۳/۵       | ۶                 | ۹                        | ۶/۲          |
| ایجاد قابلیت بازنمایی نتایج جستجو از طریق دسته‌بندی، تحلیل ساختاری، مصورسازی و سازماندهی متون بازیابی شده   | ۶                    | ۷      | ۶/۵                  | ۷                | ۶         | ۶                 | ۶                        | ۶/۴          |
| ایجاد قابلیت ذخیره و بازیابی داده‌های جستجو به صورت بافت-آگاه (Context-aware)                               | ۴                    | ۶/۵    | ۵/۵                  | ۱                | ۳         | ۳/۵               | ۳/۹                      | ۴            |
| ایجاد قابلیت ارزیابی و اعتبارسنجی رکوردهای علمی                                                             | ۶                    | ۷      | ۵                    | ۱                | ۶/۵       | ۵/۵               | ۵/۲                      | ۶            |
| ایجاد قابلیت مدیریت منابع و ارجاعات پژوهش                                                                   | ۵                    | ۴      | ۵                    | ۱                | ۴/۵       | ۴                 | ۳/۹                      | ۵            |
| ایجاد قابلیت سازماندهی هوشمند اطلاعات بر اساس ترجیحات پژوهشگر                                               | ۸                    | ۷      | ۶                    | ۱                | ۴         | ۳/۵               | ۴/۹                      | ۸            |
| ایجاد قابلیت مصورسازی پایگاه دانش پژوهشگر با استفاده از تحلیل شبکه مولفان، واژگانی و سایر ابزارهای علم‌سنجی | ۶                    | ۷      | ۷/۵                  | ۱                | ۴         | ۳                 | ۴/۸                      | ۶            |
| ایجاد قابلیت ارزیابی عملکرد پژوهشی و تاثیر پژوهشگر                                                          | ۵/۵                  | ۷/۵    | ۸                    | ۱                | ۶         | ۴                 | ۵/۳                      | ۵/۵          |

| معیارهای اولویت بندی                                                                            |  | محوریت | هزینه (صرفه اقتصادی) | سادگی پیاده سازی | بلوغ فعلی | قابلیت تجاری سازی | کلیدی بودن برای ایرانداک | امتیاز نهایی |
|-------------------------------------------------------------------------------------------------|--|--------|----------------------|------------------|-----------|-------------------|--------------------------|--------------|
| عنوان پروژه                                                                                     |  |        |                      |                  |           |                   |                          |              |
| ایجاد قابلیت ویرایش ادبی و علمی متن                                                             |  | ۷      | ۴                    | ۴                | ۱         | ۸                 | ۷                        | ۵/۲          |
| ایجاد قابلیت کنترل اصالت گزارش (تشابه یابی متن)                                                 |  | ۷      | ۳                    | ۴                | ۱         | ۸                 | ۸/۵                      | ۵/۳          |
| ایجاد قابلیت دستیاری پژوهشگر در تایپ، قالب بندی گزارش، مدیریت ارجاعات و صفحه بندی و انتخاب مجله |  | ۴      | ۴                    | ۳                | ۱         | ۶                 | ۵                        | ۴/۳          |

جدول ۴-۷- پروژه های توسعه قابلیت های تعاملی، شناختی و حرکتی، و سایر کاربردها براساس معیارهای مختلف

| معیارهای اولویت بندی                                                                                 |  | محوریت | هزینه (صرفه اقتصادی ) | سادگی پیاده سازی | بلوغ فعلی | قابلیت تجاری سازی | کلیدی بودن برای ایرانداک | امتیاز نهایی |
|------------------------------------------------------------------------------------------------------|--|--------|-----------------------|------------------|-----------|-------------------|--------------------------|--------------|
| عنوان پروژه                                                                                          |  |        |                       |                  |           |                   |                          |              |
| ارتقا سیستم تشخیص هویت به شکل چند وجهی ( استفاده از صوت، تصویر و داده های بایومتریک به صورت هم زمان) |  | ۷      | ۶/۵                   | ۷                | ۱         | ۴                 | ۵/۵                      | ۵/۲          |
| ارتقا سطح هوشیاری ربات نسبت به محیط                                                                  |  | ۹      | ۸                     | ۷                | ۵         | ۴                 | ۲                        | ۵/۸          |
| ارتقا زیرسیستم مدیریت گفتگو و تعامل صوتی روبوداک                                                     |  | ۷      | ۵                     | ۶                | ۳         | ۳                 | ۱                        | ۴/۲          |
| ایجاد قابلیت تعامل روبوداک با سایر ربات ها                                                           |  | ۹      | ۶                     | ۵                | ۴         | ۶                 | ۳                        | ۵/۵          |
| ایجاد قابلیت ردیابی مقالات پژوهشگر و اطلاع رسانی حوزه ها و موضوعات پژوهشی تازه                       |  | ۵      | ۴                     | ۵                | ۱         | ۱                 | ۱                        | ۲/۸          |
| ایجاد قابلیت دستیاری در فعالیت های آزمایشگاهی به صورت فیزیکی                                         |  | ۸/۵    | ۶                     | ۴                | ۳         | ۲                 | ۱                        | ۴/۱          |
| ارتقا قابلیت حرکتی-تعاملی روبوداک در ارتباط با انسان (ایما و اشاره، تشخیص موانع، رفتار عاطفی)        |  | ۷      | ۳                     | ۴                | ۱         | ۷                 | ۲                        | ۴            |

اولویت های پیشنهادی برای اجرای پروژه ها در جدول ۷-۵ آمده است.

جدول ۷-۵- اولویت اجرای پروژه‌ها

| اولویت | عنوان پروژه                                                                                                 |
|--------|-------------------------------------------------------------------------------------------------------------|
| ۱      | ایجاد قابلیت جستجوی هم‌زمان در موتورهای علمی و پایگاه دانش روبات‌ک                                          |
| ۲      | ایجاد قابلیت بازنمایی نتایج جستجو از طریق دسته‌بندی، تحلیل ساختاری، مصورسازی و سازماندهی متون بازیابی شده   |
| ۳      | توسعه کیفیت جستجوی کاربر از طریق ایجاد قابلیت پیشنهاد کلیدواژه براساس اصطلاح‌نامه یا آنتولوژی               |
| ۴      | ایجاد قابلیت کمک به انتخاب روش پژوهش: کمک در انتخاب متدولوژی و ابزارهای پژوهش و طراحی شیوه جمع‌آوری اطلاعات |
| ۵      | ارتقا سیستم تشخیص هویت به شکل چند وجهی (استفاده از صوت، تصویر و داده‌های بیومتریک به صورت هم‌زمان)          |
| ۶      | ارتقا زیرسیستم مدیریت گفتگو و تعامل صوتی روبات‌ک                                                            |
| ۷      | ایجاد قابلیت ارزیابی عملکرد پژوهشی و تاثیر پژوهشگر                                                          |
| ۸      | ایجاد قابلیت کنترل اصالت گزارش (تشابه‌یابی متن)                                                             |
| ۹      | ایجاد قابلیت ارزیابی و اعتبارسنجی رکوردهای علمی                                                             |
| ۱۰     | ایجاد قابلیت ویرایش ادبی و علمی متن                                                                         |
| ۱۱     | ایجاد قابلیت ردیابی مقالات پژوهشگر و اطلاع‌رسانی حوزه‌ها و موضوعات پژوهشی تازه                              |
| ۱۲     | ایجاد قابلیت سازماندهی هوشمند اطلاعات بر اساس ترجیحات پژوهشگر                                               |
| ۱۳     | ایجاد قابلیت مصورسازی پایگاه دانش پژوهشگر با استفاده از تحلیل شبکه مولفان، واژگانی و سایر ابزارهای علم‌سنجی |
| ۱۴     | ایجاد قابلیت دستیاری پژوهشگر در تایپ، قالب‌بندی گزارش، مدیریت ارجاعات و صفحه‌بندی و انتخاب مجله             |
| ۱۵     | ارتقا سطح هوشیاری روبات نسبت به محیط                                                                        |
| ۱۶     | ارتقا قابلیت حرکتی-تعاملی روبات‌ک در ارتباط با انسان (ایما و اشاره، تشخیص موانع، رفتار عاطفی)               |
| ۱۷     | ایجاد قابلیت دستیاری در فعالیتهای آزمایشگاهی                                                                |
| ۱۸     | ایجاد قابلیت ذخیره و بازیابی داده‌های جستجو به صورت بافت-آگاه                                               |
| ۱۹     | ایجاد قابلیت مدیریت منابع و ارجاعات پژوهش                                                                   |
| ۲۰     | ایجاد قابلیت کمک به ایده‌پردازی: کمک به توسعه، تکامل و سازماندهی ایده‌های پژوهشگر                           |
| ۲۱     | ایجاد قابلیت تعامل روبات‌ک با سایر روبات‌ها                                                                 |

## **فصل هشتم:**

### **جمع بندی و نتیجه گیری**

## ۸. جمع‌بندی و نتیجه‌گیری

در این طرح مرحله آغازین از پروژه پژوهشی توسعه روبوداک، یک ربات پژوهشیار، به انجام رسیده است. توسعه روبوداک فرایند پژوهشی میان رشته‌ای و مدت‌دار است که در هر بازه‌ای از زمان، بخشی از خدمات روبوداک توسعه می‌یابد. در این طرح پژوهشی ابتدا یک مدل مفهومی از انواع خدماتی که روبوداک در قالب مراحل مختلف یک پژوهش می‌تواند به پژوهشگر ارائه دهد، مشخص شد. سپس با استناد به الزامات توسعه‌ای در این مرحله و بررسی انواع پلتفرم‌های ربات‌های اجتماعی، پلتفرم رباتیک Nao برای پیاده‌سازی اولیه انتخاب شد.

پلتفرم Nao، محدودیت‌هایی را برای ایجاد قابلیت‌های روبوداک در برداشت که با سفارشی‌سازی این پلتفرم، سعی شد تا این محدودیت‌ها مرتفع گردند. سفارشی‌سازی بدنه فیزیکی ربات Nao در قالب موضوعات زیر صورت گرفت:

- اضافه نمودن امکان نمایش توسط ربات، با استفاده از یک دیتا پروژکتور جاسازی شده روی بدن ربات و امکان کنترل فعال یا غیرفعال کردن پروژکتور به صورت خودکار، از طریق سیستم روبوداک
- اضافه نمودن قابلیت پردازش‌های موازی تحت سیستم عامل لینوکس، با استفاده از یک بُرد رسپری که روی بدن ربات قرار گرفته و امکان کنترل آن از طریق سیستم روبوداک
- طراحی یک نمای جدید برای نائو برای جاسازی پرویکتور و رسپری، یا استفاده از ابزارهای نرم‌افزاری طراحی، یک پوشش کلاه مانند و کیف مانند جدید برای نائو طراحی شد و با استفاده از فناوری چاپ سه بعدی، این پوشش‌ها به صورت سه بعدی پرینت گرفته شد و در بدن ربات جاسازی شد.

مهمترین دستاورد این مرحله، یکپارچه‌سازی سیستم روبوداک برای ارائه خدمات مورد نیاز در این مرحله در قالب سیستم واحد است. بنابراین علاوه بر تغییرات و نوآوری‌های فوق روی بدنه فیزیکی ربات، عملیات زیر در قالب نوآوری‌های سیستمی و عملکردی در این مرحله توسعه یافته است:

- یکپارچه‌سازی خدمات جستجو در پایگاه گنج، خواندن از روی متن، تشخیص گفتار فارسی و انگلیسی، نمایش مصور نتایج جستجو، نمایش نتایج در نمایشگر پروژکتور و ارائه حرکات فیزیکی هماهنگ در ربات، در قالب سیستم هوشمند روبات‌ک
- طراحی سناریوها و مدل تعاملی روبات‌ک با پژوهشگر برای پیاده‌سازی خدمات این مرحله
- ایجاد ارتباط موثر بین چند سیستم با سیستم عامل‌های متنوع در سیستم روبات‌ک
- ارائه مدلی برای نمایش مصور نتایج جستجو براساس نمایه‌ها و موضوعات مشخص شده برای رکوردهای گنج
- طراحی و پیاده‌سازی الگوریتمی برای تشخیص حالت حرکت در فرد تعامل‌کننده با روبات‌ک در دو جهت بالا-پایین و چپ-راست

علاوه بر موارد فوق، ارائه مقاله مربوط به مدل مفهومی توسعه روبات‌ک در مهمترین کنفرانس بین‌المللی ربات‌های اجتماعی<sup>۱۱۲</sup> (Mohebi et al., 2016)، ایجاد اولین آزمایشگاه پژوهشی در ایراندک، تشکیل گروه پژوهشی میان‌رشته‌ای برای محقق نمودن خدمات، و برقراری ارتباط پژوهشی با چند موسسه و دانشگاه، جزو دستاوردهای دیگر این طرح پژوهشی تاکنون محسوب می‌شود. برنامه توسعه روبات‌ک در آینده نیز در فصلی جداگانه بررسی شد و ۲۱ پروژه اصلی برای توسعه قابلیت‌های روبات‌ک معرفی شد. پروژه‌ها براساس معیارهای مختلفی اولویت‌بندی شدند که از بین آنها، سه پروژه با بیشترین امتیاز به شرح زیر مشخص شدند:

- ایجاد قابلیت جستجوی هم‌زمان در موتورهای علمی و پایگاه دانش روبات‌ک
- ایجاد قابلیت بازنمایی نتایج جستجو از طریق دسته‌بندی، تحلیل ساختاری، مصورسازی و سازماندهی متون بازیابی شده
- توسعه کیفیت جستجوی کاربر از طریق ایجاد قابلیت پیشنهاد کلیدواژه براساس اصطلاح‌نامه یا آنتولوژی

در کنار توسعه این پروژه‌ها و یکپارچه‌سازی آنها با روبات‌ک، لازم است که مدل تعاملی روبات‌ک با پژوهشگر نیز در قالب مدل‌های تعامل انسان و ماشین، توسعه یابد تا روبات‌ک بتواند پاسخگوی خدمات حاصل از این پروژه‌ها برای پژوهشگر باشد.

---

<sup>112</sup> International Conference on Social Robotics (ICSR 2016)

## ٩. منابع

- Ailamaki, A., Kantere, V., & Dash, D. (2010). Managing scientific data. *Communications of the ACM* ,3586 ,–.87
- Aldebaran. (2015). Who is Nao?
- Alemi, M., Meghdari, A., Basiri, N. M., & Taheri, A. (2015). The Effect of Applying Humanoid Robots as Teacher Assistants to Help Iranian Autistic Pupils Learn English as a Foreign Language. In *International Conference on Social Robotics* (pp. 1–.01)
- Bainbridge, W. A., Hart, J. W., Kim, E. S., & Scassellati, B. (2011). The benefits of interactions with physically present robots over video-displayed agents. *International Journal of Social Robotics* ,314 ,(1)–.25
- Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* ,91117 ,(7)–.027
- Bilaniuk, O., Fazl-Ersi, E., Laganier, R., Xu, C., Laroche, D., & Moulder, C. (2014). Fast lbp face detection on low-power simd architectures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 616–.226)
- Breazeal, C., Brooks, A., Chilongo, D., Gray, J., Hoffman, G., Kidd, C., ... Lockerd, A. (2004). Working collaboratively with humanoid robots. In *Humanoid Robots, 2004 4th IEEE/RAS International Conference on* (Vol. 1, pp. 253–.272)
- Chang-yeon, J. (2008). Face Detection using LBP features. *Stanford.Edu1* ,–.4
- Embedded system. (n.d.).
- Face Detection using Haar Cascades. (n.d.).
- Face recognition homepage. (n.d.).
- Feil-Seifer, D., & Mataric, M. J. (2005). Defining socially assistive robotics. In *9th International Conference on Rehabilitation Robotics*.
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics* ,72)971 ,(–.881
- Fridin, M. (2014). Kindergarten social assistive robot: First meeting and ethical issues. *Computers in Human Behavior* ,03262 ,–.272
- Gonzalez-Pacheco, V. (2011). Maggie: A social robot as a gaming platform. *International Journal of Social Robotics* ,3173 ,–.183
- Gorostiza, J. F., & Salichs, M. A. (2011). End-user programming of a social robot by



- dialog. *Robotics and Autonomous Systems* ,952011 ,(21)–.4111
- Hartnett, T. (2011). *Consensus-oriented decision-making: the codm model for facilitating groups to widespread agreement*. new society publishers.
- Jaiswal, S. (2011). Comparison between face recognition algorithm-eigenfaces, fisherfaces and elastic bunch graph matching. *Journal of Global Research in Computer Science* ,2781 ,(7)–.391
- Kanda, T. (2008). Who will be the customer?: a social robot that anticipates people’s behavior from their trajectories. In *Proceedings of the 10th international conference on Ubiquitous computing*. ACM.
- Kidd, C. D., & Breazeal, C. (2004). Effect of a robot on user perceptions. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on* (Vol. 4, pp. 3559–.(4653
- Kidd, C. D., Taggart, W., & Turkle, S. (2006). A sociable robot to encourage social interaction among the elderly. In *Proceedings 2006 IEEE International Conference on Robotics and Automation*.
- Klingspor, V., Demiris, J., & Kaiser, M. (1997). Human-robot communication and machine learning. *Applied Artificial Intelligence* ,11917 ,(7)–.647
- Leite, I., Martinho, C., & Paiva, A. (2013). Social robots for long-term interaction: a survey. *International Journal of Social Robotics* ,5192 ,(2)–.803
- Lienhart, R., Kuranov, A., & Pisarevsky, V. (2003). Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *Joint Pattern Recognition Symposium* (pp. 297–.(403
- Malik, N. A., Yussof, H., & Hanapiah, F. A. (2014). Development of Imitation Learning through Physical Therapy Using a Humanoid Robot. *Procedia Computer Science* ,24191 ,–.791
- Meltzoff, A. N., Brooks, R., Shon, A. P., & Rao, R. P. N. (2010). “Social” robots are psychological agents for infants: A test of gaze following. *Neural Networks* , 32669 ,(8)–.279
- Michaud, F. (2007). Telepresence Robot for Home Care Assistance. *AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics*.
- Mitsunaga, N. (2006). Robovie-IV: a communication robot interacting with people daily in an office. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Mohebi, A., Golshaie, R., Ganjefar, S., Jalalimanesh, A., Afshar, P., Hosseini, A. A., ... Badamchi, A. (2016). Conceptual Framework for RoboDoc: A New Social Robot for Research Assistantship. In A. Agah, J.-J. Cabibihan, A. M. Howard, M. A. Salichs, & H. He (Eds.), *Social Robotics: 8th International Conference, ICSR 2016, Kansas City, MO, USA, November 1-3, 2016 Proceedings* (pp. 808–.(818 Cham: Springer International Publishing. [http://doi.org/10.1007/978-3-913-73474-97\\_3](http://doi.org/10.1007/978-3-913-73474-97_3)
- Nejat, G., & Ficocelli, M. (2008). Can I be of assistance? The intelligence behind an assistive robot. In *ICRA 2008. IEEE International Conference on*.

- Newell, A., & Simon, H. A. (1976). Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, 9(13), 611–621.
- Pacchierotti, E., Christensen, H. I., & Jensfelt, P. (2006). Design of an office-guide robot for social interaction studies. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Pfeifer, R., & Bongard, J. (2006). *How the body shapes the way we think: a new view of intelligence*. MIT press.
- Robertson, S. (2010). *The Probabilistic Relevance Framework: BM25 and Beyond . Foundations and Trends® in Information Retrieval* (Vol. 3). <http://doi.org/10.1561/15000000019>
- Robertson, S., Zaragoza, H., & Taylor, M. (2004). Simple BM25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management* (pp. 42–). (94
- Rodriguez-Lizundia, E., Marcos, S., Zalama, E., Gómez-García-Bermejo, J., & Gordaliza, A. (2015). A bellboy robot: Study of the effects of robot behaviour on user engagement and comfort. *International Journal of Human-Computer Studies*, 2838, –.59
- Salichs, M. A., Barber, R., Khamis, A. M., Malfaz, M., Gorostiza, J. F., Pacheco, R., ... García, D. (2006). Maggie: A robotic platform for human-robot social interaction. In *2006 IEEE Conference on Robotics, Automation and Mechatronics* (pp. 1–). (7
- Tzafestas, S. G. (2014). Mobile Robots at Work-.51
- Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (Vol. 1, pp. 1–). (115
- Wainer, J., Feil-Seifer, D. J., Shell, D. A., & Mataric, M. J. (2006). The role of physical embodiment in human-robot interaction. In *ROMAN 2006-The 15th IEEE International Symposium on Robot and Human Interactive Communication* (pp. 711–). (221
- Yan, H., Jr, M. H. A., & Poo, A. N. (2014). A survey on perception methods for human–robot interaction in social robots. In *International Journal of Social Robotics* (pp. 85–). (911
- Yumakulov, S., Yergens, D., & Wolbring, G. (2012). Imagery of disabled people within social robotics research. In *International Conference on Social Robotics. Springer Berlin Heidelberg*.
- Zafeiriou, S., Zhang, C., & Zhang, Z. (2015). A survey on face detection in the wild: Past, present and future. *Computer Vision and Image Understanding*, 83(1)(March), 1–24. <http://doi.org/10.1016/j.cviu.2015.03.015>

پیوست‌ها

پیوست ۱:

**جداول امتیازدهی خدمات روبوداک**

## امتیازهای خدمات براساس معیارها

| امتیاز نهایی | پیچیدگی اجرا | هزینه | محوریت | تطابق با الزامات مرحله اول | الزامات مرحله آغازین                                                              |                  |
|--------------|--------------|-------|--------|----------------------------|-----------------------------------------------------------------------------------|------------------|
|              |              |       |        |                            | خدمات روبوداک                                                                     |                  |
| ۰/۶۷         | ۲            | ۲     | ۱      | ۰                          | دریافت و سازماندهی ایده های پژوهشگر                                               | خلق ایده         |
| ۰/۲۲         | ۳            | ۳     | ۱      | ۰                          | کمک به توسعه پژوهش و بهبود ایده                                                   |                  |
| ۱/۴۴         | ۱            | ۱     | ۱      | ۱                          | کمک در انتخاب متدولوژی و ابزارهای پژوهش                                           | طراحی            |
| ۱/۰۰         | ۲            | ۲     | ۱      | ۱                          | کمک در انتخاب و طراحی شیوه جمع آوری اطلاعات                                       | پژوهش            |
| ۲/۳۳         | ۲            | ۱     | ۳      | ۳                          | جستجو در پایگاه وب و گنج                                                          | جمع آوری اطلاعات |
| ۱/۵۶         | ۲            | ۲     | ۲      | ۲                          | جستجوی هم زمان در موتورهای علمی و پایگاه دانش روبوداک                             |                  |
| ۲/۲۲         | ۱/۵          | ۲     | ۳      | ۳                          | توسعه کلیدواژه ها براساس اصطلاح نامه یا آنتولوژی                                  |                  |
| ۲/۰۰         | ۲            | ۲     | ۲/۵    | ۳                          | قابلیت دسته بندی و سازماندهی متون بازیابی شده                                     |                  |
| ۱/۲۲         | ۱/۵          | ۲     | ۱/۵    | ۱                          | ذخیره سازی رکوردهای بازیابی شده و ترجیحات پژوهشگر به صورت همزمان (داده های محیطی) |                  |
| ۰/۴۴         | ۲/۵          | ۲/۵   | ۱      | ۰                          | ذخیره سازی معنایی به صورت شبکه ای                                                 |                  |
| ۲/۰۵         | ۱/۵          | ۲     | ۳      | ۲/۵                        | تحلیل ساختاری و مصورسازی نتایج جستجو                                              |                  |
| ۱/۳۳         | ۱/۵          | ۲     | ۲      | ۱                          | ارزیابی و اعتبارسنجی رکوردهای علمی                                                |                  |

| خدمات روبوداک             |                                                             |        |       |              |              |
|---------------------------|-------------------------------------------------------------|--------|-------|--------------|--------------|
| الزامات مرحله آغازین      | تطابق با الزامات مرحله اول                                  | محوریت | هزینه | پیچیدگی اجرا | امتیاز نهایی |
| سازماندهی و تحلیل اطلاعات | مدیریت منابع و ارجاعات پژوهش                                | ۰      | ۱/۵   | ۱/۵          | ۱/۰۰         |
|                           | ارائه جعبه ابزار تحلیل اطلاعات                              | ۰      | ۱     | ۲            | ۰/۶۷         |
|                           | تحلیل شبکه مولفان و واژگانی پایگاه دانش (ابزارهای علم‌سنجی) | ۰      | ۱     | ۱/۵          | ۱/۳۳         |
|                           | مصورسازی اطلاعات و ترسیم نمودار                             | ۱      | ۱     | ۱/۵          | ۰/۷۸         |
| تولید گزارش               | کمک در ویرایش ادبی و علمی متن                               | ۰      | ۱/۵   | ۲            | ۱/۱۱         |
|                           | تولید گزارش اولیه به همراه نمودار                           | ۰      | ۱     | ۲            | ۰/۸۹         |
|                           | کنترل اصالت گزارش (سرقت علمی)                               | ۰      | ۲/۵   | ۱            | ۰/۶۷         |
|                           | مدیریت ارجاعات گزارش                                        | ۰      | ۱/۵   | ۱            | ۱/۴۴         |
|                           | تایپ گزارش                                                  | ۲      | ۲/۵   | ۱            | ۱/۱۱         |
|                           | ذخیره گزارش به صورت ساخت یافته                              | ۰      | ۱     | ۱/۷۵         | ۲/۰۰         |
| انتشار                    | ردیابی مقالات پژوهشگر و اطلاع رسانی                         | ۰      | ۱     | ۱            | ۰/۷۲         |
|                           | ردیابی حوزه‌ها و موضوعات پژوهشی تازه (ترجیحات پژوهشگر)      | ۱      | ۲     | ۱/۵          | ۱/۱۱         |
|                           | کمک در انتخاب مجله                                          | ۱      | ۱     | ۱            | ۱/۵۶         |
|                           | تبدیل قالب مقاله                                            | ۰      | ۱     | ۲/۵          | ۱/۴۴         |

**پیوست ۲:**

**کدهای برنامه**

## ۱- کدهای برنامه اصلی

```
#!/usr/bin/env python3
#-*- coding: cp1256 -*-

import sys
from naoqi import ALProxy
import time
import almath
import os
import socket
import subprocess
import shutil
import ftplib
import codecs
import speech_recognition as sr
from sys import byteorder
from array import array
from struct import pack
from os import path
import pyaudio
import wave
import urllib
import contextlib
from shutil import copyfile

saving a wav file based on silence detection

THRESHOLD = 100
CHUNK_SIZE = 1024
FORMAT = pyaudio.paInt16
RATE = 44100

def is_silent(snd_data):
 "Returns 'True' if below the 'silent' threshold"
 return max(snd_data) < THRESHOLD

def normalize(snd_data):
 "Average the volume out"
 MAXIMUM = 16384
 times = float(MAXIMUM)/max(abs(i) for i in snd_data)

 r = array('h')
```



```

for i in snd_data:
 r.append(int(i*times))
return r

def trim(snd_data):
 "Trim the blank spots at the start and end"
 def _trim(snd_data):
 snd_started = False
 r = array('h')

 for i in snd_data:
 if not snd_started and abs(i)>THRESHOLD:
 snd_started = True
 r.append(i)

 elif snd_started:
 r.append(i)
 return r

 # Trim to the left
 snd_data = _trim(snd_data)

 # Trim to the right
 snd_data.reverse()
 snd_data = _trim(snd_data)
 snd_data.reverse()
 return snd_data

def add_silence(snd_data, seconds):
 "Add silence to the start and end of 'snd_data' of length 'seconds' (float)"
 r = array('h', [0 for i in xrange(int(seconds*RATE))])
 r.extend(snd_data)
 r.extend([0 for i in xrange(int(seconds*RATE))])
 return r

def record():
 """
 Record a word or words from the microphone and
 return the data as an array of signed shorts.

 Normalizes the audio, trims silence from the
 start and end, and pads with 0.5 seconds of
 blank sound to make sure VLC et al can play
 it without getting chopped off.
 """

```

```

p = pyaudio.PyAudio()
stream = p.open(format=FORMAT, channels=1, rate=RATE,
 input=True, output=True,
 frames_per_buffer=CHUNK_SIZE)

num_silent = 0
snd_started = False

r = array('h')

while 1:
 # little endian, signed short
 snd_data = array('h', stream.read(CHUNK_SIZE))
 if byteorder == 'big':
 snd_data.byteswap()
 r.extend(snd_data)

 silent = is_silent(snd_data)

 if silent and snd_started:
 num_silent += 1
 elif not silent and not snd_started:
 snd_started = True

 if snd_started and num_silent > 30:
 break

sample_width = p.get_sample_size(FORMAT)
stream.stop_stream()
stream.close()
p.terminate()

r = normalize(r)
r = trim(r)
r = add_silence(r, 0.5)
return sample_width, r

def record_to_file(path):
 "Records from the microphone and outputs the resulting data to 'path'"
 sample_width, data = record()
 data = pack('<' + ('h'*len(data)), *data)

 wf = wave.open(path, 'wb')
 wf.setnchannels(1)
 wf.setsampwidth(sample_width)

```

```
wf.setframerate(RATE)
wf.writeframes(data)
wf.close()
```

```
google speech to text
```

```
def english():
 print("please speak a word into the microphone")
 record_to_file('demo.wav')
 print("done - result written to demo.wav")
```

```
AUDIO_FILE = path.join(path.dirname(path.realpath(__file__)), "demo.wav")
```

```
r = sr.Recognizer()
with sr.AudioFile(AUDIO_FILE) as source:
 audio = r.record(source) # read the entire audio file
```

```
Text= r.recognize_google(audio)
print("Text: " + Text)
return Text
```

```
#Sending search statement to raspberry
```

```
def search():
 global GooyeshSock
 global sock
 Pactus("search1")
 SendToGooyesh("start")
 word=GooyeshSock.recv(128)
 while ("فارسی" not in word) & ("انگلیسی" not in word) & ("انگلیسی" not in word)
```

```
 & ("انگلیسی" not in word):
```

```
 word=GooyeshSock.recv(128)
```

```
Pactus("search")
```

```
if "فارسی" in word:
```

```

SendToGooyesh("search")
data = GooyeshSock.recv(128)
data=data.decode('cp1256')
data=data.encode('utf-8')
SendToServer((
S = "searchfarsi ," + data
sock.sendall(S)
else:
 data=english()
 S = "searchenglish ," + data
 SendToServer()
 sock.sendall(S)
reading a text using pactus
def read():
 global sock
 SendToServer()
 sock.sendall('read')
 time.sleep(2)

Pactus('enter')
time.sleep(2)
IP = '192.168.1.18' #set your robot's IP here
PORT = 9559

#text downloading
link = "http://10.7.6.172/files/Text-utf8.txt"
text = open(r"C:\Users\Nao\Downloads\test.txt", 'r')
text2=text.read()

text2=text2.decode('utf8')
text.close()

while True:
 f = urllib.urlopen(link)
 myfile1 = f.read()
 myfile= myfile1.decode('utf8')
 #checking for new text
 if text2 != myfile:
 text = open("C:/Users/Nao/Downloads/test.txt", 'w')
 text.write(myfile.encode("utf-8"))
 text.close()
 link2 = "http://10.7.6.172/files/Text-unicode.txt"
 f2 = urllib.urlopen(link2)
 myfile2 = f2.read()

```

```

p = open("D:/AutoText2Wave/TXT/read.txt",'w')
p.write(myfile2)
p.close()
time.sleep(2)
fname1 = "C:/Users/Nao/Desktop/controler/read.wav"
with contextlib.closing(wave.open(fname1,'r')) as f:
 frames = f.getnframes()
 rate = f.getframerate()
 duration1 = frames / float(rate)
 print 1
 print(duration1)

#checking for new audio
while True:
 fname2 = "D:/AutoText2Wave/Wave/read.wav"
 with contextlib.closing(wave.open(fname2,'r')) as f:
 frames = f.getnframes()
 rate = f.getframerate()
 duration2 = frames / float(rate)
 print 2
 print(duration2)

 if duration1 != duration2:
 #sending audio to robot
 s=ftplib.FTP(IP,'nao','nao')
 file2=open(r"D:\AutoText2Wave\WAVE\read.wav",'rb')
 s.storbinary('STOR behaviors/play_sound/1.wav',file2)

 file2.close()
 s.quit()
 b = ALProxy("ALBehaviorManager", IP, PORT)
 b.runBehavior('play_sound')
 copyfile("D:/AutoText2Wave/WAVE/read.wav",
"C:/Users/Nao/Desktop/controler/read.wav")

 break
 time.sleep(1)

break

time.sleep(2)

```

```
#connecting to nevisa
```

```
def ConnectToGooyesh():
 # Create a TCP/IP socket
 global GooyeshSock
 GooyeshSock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

 # Connect the socket to the port where the server is listening
 server_address = ('192.168.1.201', 9998)
 print >>sys.stderr, 'connecting to %s port %s' % server_address
 GooyeshSock.connect(server_address)
```

```
Send data
#message = 'This is the message. It will be repeated.'
print >>sys.stderr, 'sending "salam"'
```

```
def SendToGooyesh(status):
 global GooyeshSock
 if status=="start":
 GooyeshSock.sendall("startt")

 elif status=="search":
 GooyeshSock.sendall("search")
 else:
 GooyeshSock.sendall("stoppp")
```

```
robot speacks using paktus
```

```
def Pactus(text1):
 print text1

 if text1=="Gomar":
 text1="سلام. خانم گمار. حال شما چطور است؟"

 elif text1=="Afshar":
 text1="سلام. خانم افشار. حال شما چطور است؟"
 elif text1=="Ramezani":
 text1="سلام. آقای روضانی. حال شما چطور است؟"
 elif text1=="NoDetect":
 text1="من کسی را نمی بینم."
```

```

elif text1=="NotDetect":
 text1="من کسی را نمی بینم".
elif text1=="wrong":
 text1="خیلی معذرت می خواهم"
elif text1=="true":
 text1="خیلی خوشحال شدم. چه کاری می خواهید برایتان انجام دهم؟"
elif text1=="enter":
 text1="لطفا متن مورد نظر خود را وارد کنید".
elif text1=="search1":
 text1="عبارت جستجوی شما فارسی است یا انگلیسی؟"
elif text1=="search":
 text1="چه عبارتی را برایتان جستجو کنم؟"
elif text1=="repeat":
 text1="لطفا پاسخ خود را تکرار کنید".

else:
 text1="من شما را نمی شناسم. من، دستیار پژوهشی آقای رضایی هستم".

file = open("1.txt", "w")#create text file
file.write(text1)
file.close()

shutil.copy2("1.txt", r"D:\AutoText2Wave\TXT")
time.sleep(1)

#connecting with robot-----

IP = '192.168.1.18' #set your robot's IP here
PORT = 9559

#send wav file to robot with ftp-----
s=ftplib.FTP(IP,'nao','nao')
file2=open(r"D:\AutoText2Wave\WAVE\1.wav",'rb')
s.storbinary('STOR behaviors/play_sound/1.wav',file2)

#this is the dir for play_sound behavior in robot
file2.close()

```

```
s.quit()
```

```
#active play behavior-----
b = ALProxy("ALBehaviorManager", IP, PORT)
b.runBehavior('play_sound')
#now robot must read the text
```

```
def setStiffnesses (robotIP):
 PORT = 9559
```

```
 try:
 motionProxy = ALProxy("ALMotion", robotIP, PORT)
 except Exception,e:
 print "Could not create proxy to ALMotion"
 print "Error was: ",e
 sys.exit(1)
```

```
 motionProxy.setStiffnesses("Head", 1.0)
```

```
def Stiffnessesoff(robotIP):
 PORT = 9559
```

```
 try:
 motionProxy = ALProxy("ALMotion", robotIP, PORT)
 except Exception,e:
 print "Could not create proxy to ALMotion"
 print "Error was: ",e
 sys.exit(1)
```

```
 motionProxy.setStiffnesses("Head", 0.0)
```

```
def HeadYaw (robotIP,angles):
 PORT = 9559
```

```
 try:
 motionProxy = ALProxy("ALMotion", robotIP, PORT)
 except Exception,e:
 print "Could not create proxy to ALMotion"
 print "Error was: ",e
 sys.exit(1)
 names = "HeadYaw"
```



```

fractionMaxSpeed = 0.1
motionProxy.setStiffnesses("Head", 1.0)
motionProxy.changeAngles(names,angles,fractionMaxSpeed)

def HeadYawset (robotIP):
 PORT = 9559
 try:
 postureProxy = ALProxy("ALRobotPosture", robotIP, PORT)
 except Exception,e:
 print "Could not create proxy to ALMotion"
 print "Error was: ",e
 sys.exit(1)
 postureProxy.goToPosture("Stand", 1.(0
def HeadPitch (robotIP,angles):
 PORT = 9559
 try:
 motionProxy = ALProxy("ALMotion", robotIP, PORT)
 except Exception,e:
 print "Could not create proxy to ALMotion"
 print "Error was: ",e
 sys.exit(1)
 names = "HeadPitch"
 fractionMaxSpeed = 0.1

 motionProxy.setStiffnesses("Head", 1.0)
 motionProxy.changeAngles(names,angles,fractionMaxSpeed)

#connecting to raspberry

def SendToServer():
 # Create a TCP/IP socket
 global sock
 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

 # Connect the socket to the port where the server is listening
 server_address = ('192.168.1.55', 9998)
 print >>sys.stderr, 'connecting to %s port %s' % server_address
 sock.connect(server_address)
 print >>sys.stderr, 'sending salam'

#when some one is recognized:
def knownrec():
 #robot turns head

```

```

global x ,y , name, GooyeshSock
robotIp = "192.168.1.18"
motionProxy = ALProxy("ALMotion", "192.168.1.18", 9559)
motionProxy.post.moveTo(X, Y, Theta)
time.sleep(2.0)
angles1= (x)*almath.TO_RAD
angles2= (-y)*almath.TO_RAD
HeadYaw(robotIp,angles1)
HeadPitch(robotIp,angles2)
#robot says hello to the person
Pactus(name)
#nevisa waits for predetermined answers
while True:
 gest=""
 SendToServer()
 sock.sendall("gesture")
 SendToGooyesh("start")

 #raspberry runs gesture recognition program

 data = GooyeshSock.recv(32)

 print data
 if ("ممنون" in data)| ("خوب" in data)| ("نیست" in data):
 break

 SendToGooyesh("stop")

 if ("ممنون" not in data)& ("خوب" not in data)& ("نیست" not in data) :
 gest=sock.recv(128)
 print gest
 if (gest=="No") | (gest=="Yes"):
 break

 Pactus("repeat")

 if ("نیست" in data) | (gest=="No"):
 Pactus("wrong("

```

```

elif ("خوب" in data) | ("ممنون" in data) |(gest=="Yes"):
 Pactus("true")
 count = 0
 while True:
 count = count+1
 SendToGooyesh("start")
 #user wants to search something or not

 data = GooyeshSock.recv(32)
 while ("جستجو" not in data) & ("بخوان" not in data) & ("تمام" not in data) &
("ممنون" not in data):

 data = GooyeshSock.recv(32)
 print data

 SendToGooyesh("stop")
 if ("تمام" in data) | ("ممنون" in data):
 if count != 1:

 SendToServer()
 sock.sendall("off")
 managerProxy.runBehavior("return")
 break

 elif ("جستجو" in data):

 if count ==1 :
 robotIp = "192.168.1.18"
 postureProxy = ALProxy("ALRobotPosture", robotIp, 9559)
 time.sleep(1.0)
 postureProxy.goToPosture("Stand", 1.0)
 managerProxy = ALProxy("ALBehaviorManager", "192.168.1.18",
(9559)

 SendToServer()
 sock.sendall("on")
 managerProxy.runBehavior("projector")

 search()

 else:

 if count ==1 :

```

```

robotIp = "192.168.1.18"
postureProxy = ALProxy("ALRobotPosture", robotIp, 9559)
time.sleep(1.0)
postureProxy.goToPosture("Stand", 1.0)
managerProxy = ALProxy("ALBehaviorManager", "192.168.1.18",
(9559
 SendToServer()
 sock.sendall("on")
 managerProxy.runBehavior("projector")
read()

```

#when robot wants to recognize user

def unknownrec(angles1):

```

 global sock
 robotIp = "192.168.1.18"
 postureProxy = ALProxy("ALRobotPosture", robotIp, 9559)
 time.sleep(1.0)
 postureProxy.goToPosture("Stand", 1.0)
 motionProxy = ALProxy("ALMotion", "192.168.1.18", 9559)
 motionProxy.setWalkArmsEnabled(True, True)

 motionProxy.setMotionConfig([["ENABLE_FOOT_CONTACT_PROTECTION",
 True]])

 X = 0
 Y = 0
 Theta = angles1
 print Theta
 motionProxy.post.moveTo(X, Y, Theta)
 time.sleep(2.0)
 X = 0.2
 Y = 0.0
 Theta = angles1
 print Theta
 motionProxy.post.moveTo(X, Y, Theta)
 time.sleep(2.0)
 SendToServer()
 detectresult="false"
 sock.sendall("Detect")

```

```

detectresult = sock.recv(128)
print >>sys.stderr, 'received "%s"' % detectresult
if detectresult=="false":
 detectresult=TurningHead(detectresult)
print detectresult

SendToServer()
sock.sendall("Recognition")
total=sock.recv(128)
return total

```

```

def TurningHead(detectresult):
 rod=0
 rod1=0
 y=0
 while rod1<1:

 if detectresult=="false":
 while rod<=1:
 robotIp = "192.168.1.18"
 print rod
 rod2=rod
 rod3=rod1
 x=20
 setStiffnesses("192.168.1.18")
 angles1= (x)*almath.TO_RAD
 angles2= (y)*almath.TO_RAD
 HeadPitch(robotIp,angles2)
 HeadYaw(robotIp,angles1)
 time.sleep(0.5)
 SendToServer()
 sock.sendall("Detect")
 detectresult = sock.recv(128)
 if detectresult=="true":
 break

 rod=rod+1
 rod=0
 if detectresult=="false":
 while rod<=3:
 robotIp = "192.168.1.18"
 print rod
 rod2=rod
 rod3=rod1
 x=-02

```

```
setStiffnesses("192.168.1.18")
angles1= (x)*almath.TO_RAD
angles2= (y)*almath.TO_RAD
HeadPitch(robotIp,angles2)
HeadYaw(robotIp,angles1)
time.sleep(0.5)
SendToServer()
sock.sendall("Detect")
detectresult = sock.recv(128)
if detectresult=="true":
 break
rod=rod+1
```

```
if detectresult=="true":
 break
rod=0
rod1=rod1+1
y=y-1
print detectresult
return detectresult
```

```
if __name__ == '__main__':
 SendToServer()
 global sock
 SendToServer()
 global GooyeshSock
 ConnectToGooyesh()

 #waiting for user to say hello
 SendToGooyesh("start")
 data = GooyeshSock.recv(32)
 a="سلام"

 while a not in data:
 time.sleep(1)
 data = GooyeshSock.recv(32)
 print data
 SendToGooyesh("stop")
```

```
#opening pactus
subprocess.Popen("D:\Pactos\Gooya\PacTxt2WavFile.exe")
global x , y , name
robotIp = "192.168.1.18"
rod=0
rod1=0
y=0
detectresult="false"
sock.sendall("Detect")
detectresult = sock.recv(128)
print >>sys.stderr, 'received "%s"' % detectresult
if detectresult=="false":
 detectresult=TurningHead(detectresult)
print detectresult

if detectresult=="true":
 SendToServer()
 sock.sendall("Recognition")
 total = sock.recv(128)
 print total

total2=total.split()
print total2
name=total2[2]
x=0
y=0
if name!='NotDetect':
 y1=total2[6]
 y1=y1[:1]
 print y1
 c=total2[5]
 x1=c[1:]
 print x1
 x=-(int(x1)-11)/(023
 y=-(int(y1)-11)/(091

angles1= (x)*almath.TO_RAD
angles2= (y)*almath.TO_RAD
X = 0
Y = 0
Theta = angles1
if (name!='unknown') & (name!='NotDetect'):
 knownrec()
```

```

elif name=='unknown':
 total=unknownrec(angles1)
 total2=total.split()
 name=total2[2]
 x=0
 y=0
 if (name!='NotDetect'):
 y1=total2[6]
 y1=y1[:1]
 c=total2[5]
 x1=c[1:]
 x=-((int(x1)-11)/(023))
 y=-((int(y1)-11)/(091))
 angles1= (x)*almath.TO_RAD
 angles2= (y)*almath.TO_RAD
 X = 0
 Y = 0
 Theta = angles1

if (name!='unknown') & (name!='NotDetect'):
 knownrec()
elif name=='unknown':
 Pactus(name)
 robotIp = "192.168.1.18"
 postureProxy = ALProxy("ALRobotPosture", robotIp, 9559)
 time.sleep(1.0)
 postureProxy.goToPosture("Sit", 1)

if (name=='NotDetect'):
 Pactus(name)
 robotIp = "192.168.1.18"
 postureProxy = ALProxy("ALRobotPosture", robotIp, 9559)
 time.sleep(1.0)
 postureProxy.goToPosture("Sit", 1)

else:
 name='NotDetect'
 Pactus(name)
 robotIp = "192.168.1.18"
 postureProxy = ALProxy("ALRobotPosture", robotIp, 9559)
 time.sleep(1.0)
 postureProxy.goToPosture("Sit", 1)

```



else:

```
name='NoDetect'
Pactus(name)
robotIp = "192.168.1.18"
postureProxy = ALProxy("ALRobotPosture", robotIp, 9559)
time.sleep(1.0)
postureProxy.goToPosture("Sit", 1)
```

Stiffnessesoff("192.168.1.18")

#### ۱- کدهای بخش بینایی ماشین

لیست تمام کدهای استفاده شده برای پردازش تصویر در این پژوهش به شرح زیر است:

- فایل‌های مورد استفاده برای آموزش و تست بازشناسی چهره<sup>۱۱۳</sup> به همراه کد تشخیص صورت<sup>۱۱۴</sup>:

Main\_train.cpp (۱)  
Main\_predict.cpp (۲)

detect\_face.cpp ✓  
detectObject.cpp ✓  
preprocessFace.cpp ✓  
recognition.cpp ✓  
ImageUtiles\_0.7.cpp ✓

فایل‌های سرآمد<sup>۱۱۵</sup>:

detectObject.h ○  
preprocessFace.h ○  
recognition.h ○  
ImageUtiles\_0.7.h ○  
Time.h : فایل خود opencv است. ○  
Cv.h : فایل خود opencv است. ○

- فایل تشخیص حرکت<sup>۱۱۶</sup>:

GestureRecognition.cpp (۳)  
Object.cpp ✓

---

<sup>۱۱۳</sup> Face Recognition

<sup>۱۱۴</sup> Face Detection

<sup>۱۱۵</sup> Header

<sup>۱۱۶</sup> Gesture

Object.h ✓

(۱-۱) کد آموزش بازشناسی چهره Main\_train.cpp:

برای انجام بازشناسی چهره در کتابخانه‌ی opencv، سه الگوریتم Eigenfaces، Fisherfaces و Local Binary Patterns Histograms تعبیه شده است. در کد پیاده‌سازی شده در این پروژه نیز امکان استفاده از هر یک از این الگوریتم‌ها وجود دارد. برای انتخاب هر یک از این الگوریتم‌ها کافی است که خط مربوط به آن را از حالت کامنت خارج کرده و سپس بعد از make کردن دوباره برنامه آن را اجرا کنید.

|                                                              |                      |
|--------------------------------------------------------------|----------------------|
| const char *facerecAlgorithm = "FaceRecognizer.Fisherfaces;" | الگوریتم Fisherfaces |
| const char *facerecAlgorithm = "FaceRecognizer.Eigenfaces;"  | الگوریتم Eigenfaces  |
| const char *facerecAlgorithm = "FaceRecognizer.LBPH;"        | الگوریتم LBPH        |

برای تشخیص بخشی از تصویر به عنوان صورت انسان، الگوریتم‌های مختلفی با دقت تشخیص‌های متفاوتی وجود دارد. برای این کار باید یکی از الگوریتم‌های مورد نیاز را از حالت کامنت بودن خارج کرد.

|                                                                                                                                                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| const char *faceCascadeFilename = "lbpcascade_frontalface.xml";<br>این دسته‌بند، برای زمانی است که می‌خواهیم تشخیص چهره را با الگوریتم LBP انجام دهیم.                     |
| const char *faceCascadeFilename = "haarcascade_frontalface_alt_tree.xml"<br>این دسته‌بند، برای تشخیص چهره با روش Haar است.                                                 |
| //const char *eyeCascadeFilename ۱ = "haarcascade_lefteye_۲splits.xml"<br>این دسته‌بند، برای تشخیص چشم چپ با دقت بالا است که آن را در حالت‌های باز و بسته تشخیص دهد.       |
| //const char *eyeCascadeFilename ۲ = "haarcascade_righteye_۲splits.xml";<br>این دسته‌بند، برای تشخیص چشم راست با دقت بالا است که آن را در حالت‌های باز و بسته تشخیص دهد.   |
| //const char *eyeCascadeFilename ۱ = "haarcascade_mcs_lefteye.xml";<br>این دسته‌بند، برای تشخیص چشم چپ با دقت خوب است که آن را در حالت‌های باز و بسته تشخیص دهد.           |
| //const char *eyeCascadeFilename ۲ = "haarcascade_mcs_righteye.xml";<br>این دسته‌بند، برای تشخیص چشم راست با دقت خوب است که آن را در حالت‌های باز و بسته تشخیص دهد.        |
| const char *eyeCascadeFilename ۱ = "haarcascade_eye.xml";<br>این دسته‌بند، برای تشخیص هر دو چشم فقط در حالتی که باز باشند، است.                                            |
| const char *eyeCascadeFilename ۲ = "haarcascade_eye_tree_eyeglasses.xml";<br>این دسته‌بند، برای تشخیص هر دو چشم فقط در حالتی که باز باشند و عینک هم استفاده شده باشد، است. |

در تنظیمات بعدی، ابعاد چهره دلخواه را تنظیم می‌کنیم. باید توجه داشت که تابع "getPreprocessedFace()" پس از تشخیص چهره، قطعه‌ای از تصویر را که به عنوان چهره تشخیص داده شده است را به صورت مربعی برمی‌گرداند. پس باید طول و عرض تعیین شده یکسان باشد.

```
const int faceWidth = 7;
const int faceHeight = faceWidth;
```

در تنظیمات بعدی، باید رزولوشن تصویر را متناسب با دوربین و ابزار پردازشی مان تنظیم کنیم.

```
const int DESIRED_CAMERA_WIDTH = 0291;
const int DESIRED_CAMERA_HEIGHT = 0801;
```

تنظیمات بعدی مربوط به نحوه دریافت داده‌های آموزشی است. در این تنظیمات تعیین شده است که تصویر بعدی که به عنوان داده آموزش از کاربر گرفته می‌شود حداقل یک ثانیه بعد از تصویر ضبط شده قبلی باشد. همچنین میزان تفاوت آن با تصویر قبلی به اندازه‌ی 3.0 باشد. این تنظیمات برای این قرار داده شده است که تصاویر بسیار مشابه که دارای اطلاعات یکسانی هستند در داده‌های آموزش قرار نگیرند. همچنین کاربر فرصت کافی برای تغییر حالت صورت خود را داشته باشد.

```
const double CHANGE_IN_IMAGE_FOR_COLLECTION = 0.3;
const double CHANGE_IN_SECONDS_FOR_COLLECTION = 0.1;
```

تنظیمات بعدی، تنظیمات مربوط به نحوه نمایش پنجره تصویر در هنگام دریافت تصاویر آموزش است.

```
const char *windowName = "WebcamFaceRec"; // Name shown in the GUI window.
const int BORDER = 8; // Border between GUI elements to the edge of the image.
```

پارامتر بعدی برای این است که تنظیم کنیم که آیا می‌خواهیم که سمت چپ و راست صورت به طور جداگانه‌ای بررسی شوند و یا این که همه‌ی صورت به یکباره مورد مقایسه و بررسی قرار گیرد. بدیهی است که بررسی جداگانه دو سمت صورت، دقت بهتری را به ما خواهد داد.

```
const bool preprocessLeftAndRightSeparately = true;
```

این پارامتر برای زمان دیباگ کردن مورد استفاده قرار می‌گیرد و تصاویر مراحل میانی پردازش را نشان می‌دهد.

```
bool m_debug = false;
```

بعد از انجام تنظیمات مورد نظر، برنامه‌ها و کتابخانه‌های مورد نیاز به برنامه الحاق می‌شوند.

```
#include <stdio.h>
#include <vector>
#include <string>
#include <iostream>

// Include OpenCV's C++ Interface
#include "opencv2/opencv.hpp"

// Include the rest of our code!
#include "detectObject.h" // Easily detect faces or eyes (using LBP or Haar Cascades).
#include "preprocessFace.h" // Easily preprocess face images, for face recognition.
#include "recognition.h" // Train the face recognition system and recognize a person from
an image.
#include "ImageUtils.h" // Shervin's handy OpenCV utility functions.
using namespace cv;
using namespace std;
```

تنظیمات بعدی مربوط به GUI زمان دریافت تصاویر برای آموزش مدل است. در این برنامه، در ابتدای اجرا، برنامه در موقعیت شروع (MODE\_STARTUP) قرار دارد. سپس با کلیک بر روی دکمه Detect در GUI مربوطه، برنامه به موقعیت شناسایی صورت (MODE\_DETECTION) رفته و در تصاویر گرفته شده به جستجوی صورت می‌پردازد. با کلیک بر روی صفحه، برنامه به موقعیت جمع‌آوری صورت (MODE\_COLLECT\_FACES) رفته و شروع به ذخیره تصاویر برای یک شخص مورد نظر می‌کند. این کار تا زمانی ادامه می‌یابد که یا بر روی دکمه add person کلیک کرده باشیم، که در این صورت شروع به ذخیره تصاویر برای فرد دیگری می‌کند، و یا اینکه بر روی جایی از محیط پنجره کلیک کرده باشیم که در این صورت به موقعیت آموزش (MODE\_TRAINING) می‌رود. در موقعیت آموزش، تصاویر ذخیره شده از مرحله قبل را به تابع آموزشی OPENCV می‌دهد و مدل آموزش دیده را ذخیره می‌کند. در صورتی که بخواهیم تصاویر ذخیره شده حذف شوند و ضبط تصاویر را از ابتدا شروع کنیم، بر روی دکمه Delete All کلیک کرده و به موقعیت MODE\_DELETE\_ALL می‌رود.

```
enum MODES {MODE_STARTUP=0, MODE_DETECTION,
MODE_COLLECT_FACES, MODE_TRAINING, MODE_DELETE_ALL,
```

```
MODE_END};
const char* MODE_NAMES[] = {"Startup", "Detection", "Collect Faces", "Training",
"Delete All", "ERROR!"};
MODES m_mode = MODE_STARTUP; موقعیت ابتدایی برنامه

int m_selectedPerson = -1
این متغیر شماره فرد مورد نظر که در حال جمع آوری تصویر آموزش برای آن هستیم را نشان می-
دهد.

int m_numPersons = 0; این متغیر تعداد افرادی را که تا به حال داده آموزش برای آن‌ها ذخیره شده
است را نشان می‌دهد.

vector<int> m_latestFaces; این متغیر برداری است که تصاویر گرفته شده از افراد را در خود ذخیره
می‌کند.

هستند GUI این متغیرها مربوط به شکل ظاهری
// Position of GUI buttons:
Rect m_rcBtnAdd;
Rect m_rcBtnDel;
Rect m_rcBtnDebug;
int m_gui_faces_left = -1;
int m_gui_faces_top = -1;
```

تابع toString برای تبدیل عدد به رشته نوشته شده است.

```
template <typename T> string toString(T t)
{
 ostringstream out;
 out << t;
 return out.str();
}
```

تابع fromString، برای تبدیل رشته به عدد صحیح نوشته شده است.

```
template <typename T> T fromString(string t)
{
 T out;
 istringstream in(t);
 in >> out;
 return out;
}
```

تابع initDetectors برای بارگذاری فایل‌های دسته‌بندی XML. مورد نیاز برای تشخیص صورت و چشم‌ها به برنامه است.

```
void initDetectors(CascadeClassifier &faceCascade, CascadeClassifier &eyeCascade1,
CascadeClassifier &eyeCascade2){
```

در این بخش سعی بر لود کردن فایل XML برای تشخیص صورت است.

```
try { // Surround the OpenCV call by a try/catch block so we can give a useful error
message!
 faceCascade.load(faceCascadeFilename);
} catch (cv::Exception &e) {}
```

در این بخش اگر متغیر faceCascade تهی باشد، به این معنا است که فایل XML برای ما لود نشده است و در نتیجه یک خطا در خروجی نشان داده می‌شود و برنامه به پایان می‌رسد.

```
if (faceCascade.empty()) {
 cerr << "ERROR: Could not load Face Detection cascade classifier [" <<
faceCascadeFilename << "]" << endl;
 cerr << "Copy the file from your OpenCV data folder (eg:
'C:\\OpenCV\\data\\lbpcascades') into this WebcamFaceRec folder." << endl;
 exit(1);
}
```

در صورتی که فایل به درستی لود شود، این عمل در خروجی اعلام می‌شود.

```
cout << "Loaded the Face Detection cascade classifier [" << faceCascadeFilename << "].
<< endl;
```

در این بخش فایل XML دسته بند چشم در برنامه لود می‌شود.

```
try {
 eyeCascade1.load(eyeCascadeFilename1);
} catch (cv::Exception &e) {}
```

اگر عملیات لود به درستی انجام نشده باشد، در خروجی خطایی نمایش داده می‌شود و اجرای برنامه متوقف می‌شود.

```
if (eyeCascade1.empty()) {
 cerr << "ERROR: Could not load 1st Eye Detection cascade classifier [" <<
eyeCascadeFilename1 << "]" << endl;
 cerr << "Copy the file from your OpenCV data folder (eg:
'C:\\OpenCV\\data\\haarcascades') into this WebcamFaceRec folder." << endl;
 exit(1);
}
```

```

 cout << "Loaded the 1st Eye Detection cascade classifier [" << eyeCascadeFilename1 <<
 "]" << endl;
 // Load the Eye Detection cascade classifier xml file.
 try { // Surround the OpenCV call by a try/catch block so we can give a useful error
 message!
 eyeCascade2.load(eyeCascadeFilename2);
 } catch (cv::Exception &e) {}
 if (eyeCascade2.empty()) {
 cerr << "Could not load 2nd Eye Detection cascade classifier [" <<
 eyeCascadeFilename2 << "]" << endl;
 // Dont exit if the 2nd eye detector did not load, because we have the 1st eye detector at
 least.
 //exit(1);
 {
 else
 cout << "Loaded the 2nd Eye Detection cascade classifier [" << eyeCascadeFilename2
 << "]" << endl;
 {

```

در تابع `initWebcam`، سعی در ایجاد ارتباط با دوربین دارد. در صورتی که نتواند دوربین مورد نظر را پیدا کند، با نمایش یک خطا در خروجی، از برنامه خارج می‌شود.

```

void initWebcam(VideoCapture &videoCapture, int cameraNumber)
{
 // Get access to the default camera.
 try { // Surround the OpenCV call by a try/catch block so we can give a useful error
 message!
 videoCapture.open(cameraNumber);
 } catch (cv::Exception &e) {}
 if (!videoCapture.isOpened()) {
 cerr << "ERROR: Could not access the camera!" << endl;
 exit(1);
 {
 cout << "Loaded camera " << cameraNumber << "." << endl;
 {

```

تابع `drawString`، برای نمایش نوشته‌ها و اشکال مورد نیاز بر روی پنجره ی GUI است.

```

Rect drawString(Mat img, string text, Point coord, Scalar color, float fontScale = 0.6f, int
thickness = 1, int fontFace = FONT_HERSHEY_COMPLEX)
{
 // Get the text size & baseline.
 int baseline=0;
 Size textSize = getTextSize(text, fontFace, fontScale, thickness, &baseline);
 baseline += thickness;
 // Adjust the coords for left/right-justified or top/bottom-justified.
 if (coord.y >= 0) {
 // Coordinates are for the top-left corner of the text from the top-left of the image, so
 move down by one row.
 coord.y += textSize.height;
 }
 else {
 // Coordinates are for the bottom-left corner of the text from the bottom-left of the
 image, so come up from the bottom.
 coord.y += img.rows - baseline + 1;
 }
 // Become right-justified if desired.
 if (coord.x < 0) {
 coord.x += img.cols - textSize.width + 1;
 }
 // Get the bounding box around the text.
 Rect boundingRect = Rect(coord.x, coord.y - textSize.height, textSize.width, baseline +
textSize.height);
 // Draw anti-aliased text.
 putText(img, text, coord, fontFace, fontScale, color, thickness, CV_AA);
 // Let the user know how big their text is, in case they want to arrange things.
 return boundingRect;
}

```

تابع drawButton برای رسم دکمه‌های استفاده شده در GUI است.



```

Rect drawButton(Mat img, string text, Point coord, int minWidth = 0)
{
 int B = BORDER;
 Point textCoord = Point(coord.x + B, coord.y + B);
 // Get the bounding box around the text.
 Rect rcText = drawString(img, text, textCoord, CV_RGB(0,0,0));
 // Draw a filled rectangle around the text.
 Rect rcButton = Rect(rcText.x - B, rcText.y - B, rcText.width + 2*B, rcText.height + 2*B);
 // Set a minimum button width.
 if (rcButton.width < minWidth)
 rcButton.width = minWidth;
 // Make a semi-transparent white rectangle.
 Mat matButton = img(rcButton);
 matButton += CV_RGB(90, 90, 90);
 // Draw a non-transparent white border.
 rectangle(img, rcButton, CV_RGB(200,200,200), 1, CV_AA);
 // Draw the actual text that will be displayed, using anti-aliasing.
 drawString(img, text, textCoord, CV_RGB((0,55,01
 return rcButton;
}

```

تابع `isPointInRect` نیز مربوط به بخش GUI است. این تابع تعیین می‌کند که آیا موقعیت کلیک شده بر روی پنجره GUI بر روی دکمه‌های درون پنجره بوده است یا خیر.

```

bool isPointInRect(const Point pt, const Rect rc)
{
 if (pt.x >= rc.x && pt.x <= (rc.x + rc.width - ((1
 if (pt.y >= rc.y && pt.y <= (rc.y + rc.height - ((1
 return true;
 return false;
}

```

تابع `onMouse`، برای تشخیص کلیک چپ انجام شده بر روی پنجره GUI است که زمانی که کاربر بر روی پنجره کلیک می‌کند، به طور اتوماتیک به وسیله `OPENCV` فراخوانی می‌شود.

```

void onMouse(int event, int x, int y, int, void*)
{
 if (event != CV_EVENT_LBUTTONDOWN)
 return;
 اگر هیچ کلیک چپی انجام نشده بود که بدون انجام کاری و برگرداندن مقداری از تابع خارج می-

```

شود.

```
Point pt = Point(x,y);
```

اگر کلیک چپ انجام شده بود، موقعیت کلیک شده را در متغیری با نوع Point ذخیره می کند.

در این بخش از کد، چک می شود که آیا بر روی دکمه اضافه کردن شخص دیگر (Add Person) کلیک شده است یا خیر.

```
if (isPointInRect(pt, m_rcBtnAdd)) {
 cout << "User clicked [Add Person] button when numPersons was " << m_numPersons
 << endl;
```

در صورتی که بر روی دکمه اضافه کردن شخص کلیک شده باشد، تعداد افراد را یکی زیاد می کند.

```
if ((m_numPersons == 0) || (m_latestFaces[m_numPersons-1] == 0)) {
 // Add a new person.
 m_numPersons++;
 m_latestFaces.push_back(0); // Allocate space for an extra person.
 cout << "Num Persons: " << m_numPersons << endl;
}
```

```
// Use the newly added person. Also use the newest person even if that person was empty.
m_selectedPerson = m_numPersons - 1;
m_mode = MODE_COLLECT_FACES;
{
```

اگر کلیک بر روی دکمه حذف همه (Delete All) باشد، برنامه به موقعیت MODE\_DELETE\_ALL می رود.

```
else if (isPointInRect(pt, m_rcBtnDel)) {
 cout << "User clicked [Delete All] button." << endl;
 m_mode = MODE_DELETE_ALL;
}
```

اگر کلیک بر روی دکمه Debug بوده باشد، مقدار متغیر m\_debug تغییر می کند.

```
else if (isPointInRect(pt, m_rcBtnDebug)) {
 cout << "User clicked [Debug] button." << endl;
 m_debug = !m_debug;
 cout << "Debug mode: " << m_debug << endl;
}
```

در صورتی که بر روی هیچ یک از دکمه ها کلیک نشده باشد، یعنی بر روی بخشی از تصویر کلیک شده که در این صورت، یا بر روی صورت افراد های قبلی کلیک شده که هدف انتخاب افراد قبلی برای افزایش داده های مربوط به آنها است.

در این قطعه کد، سعی شده است که با توجه به موقعیت تصاویر صورت های قبلی در پنجره ی GUI، موقعیت کلیک شده و شخص انتخاب شده مشخص شود.

```
else {
 cout << "User clicked on the image" << endl;
 // Check if the user clicked on one of the faces in the list.
 int clickedPerson = -1;
 for (int i=0; i<m_numPersons; i++) {
 if (m_gui_faces_top >= 0) {
 Rect rcFace = Rect(m_gui_faces_left, m_gui_faces_top + i * faceHeight,
 faceWidth, faceHeight);
 if (isPointInRect(pt, rcFace)) {
 clickedPerson = i;
 break;
 }
 }
 }
}
```

اگر بر روی یکی از تصاویر کلیک شده باشد، متغیر m\_selectedPerson، به شماره شخص انتخاب شده تغییر می کند. همچنین موقعیت برنامه به MODE\_COLLECT\_FACES تغییر می کند.

```
// Change the selected person, if the user clicked on a face in the GUI.
if (clickedPerson >= 0) {
 // Change the current person, and collect more photos for them.
 m_selectedPerson = clickedPerson; // Use the newly added person.
 m_mode = MODE_COLLECT_FACES;
}
```

در صورتی که کلیک انجام شده نه بر روی هیچ دکمه ای بوده باشد و نه بر روی هیچ صورتی، پس بر روی بخش خالی از تصویر کلیک شده است که در این صورت، اگر در موقعیت MODE\_COLLECT\_FACES بوده ایم، حال موقعیت تغییر کرده و برنامه به موقعیت آموزش (MODE\_TRAINING) می رود.

```
// Otherwise they clicked in the center.
else {
 // Change to training mode if it was collecting faces.
 if (m_mode == MODE_COLLECT_FACES) {
 cout << "User wants to begin training." << endl;
```

```
m_mode = MODE_TRAINING;
{
{
{
{
```

تابع `recognizeAndTrainUsingWebcam` تابع اصلی در این برنامه است که تمام عملیات‌های مربوط به پردازش تصویر را انجام می‌دهد.

```
void recognizeAndTrainUsingWebcam(VideoCapture &videoCapture, CascadeClassifier
&faceCascade, CascadeClassifier &eyeCascade1, CascadeClassifier &eyeCascade2)
{
```

```
Ptr<FaceRecognizer> model;
vector<Mat> preprocessedFaces;
vector<int> faceLabels;
Mat old_preprocessedFace;
double old_time = 0;
```

زمانی که وارد این تابع می‌شویم، موقعیت برنامه در حالت شناسایی صورت قرار می‌گیرد.

```
m_mode = MODE_DETECTION;
```

سپس در یک حلقه فریم به فریم تصاویر را از دوربین دریافت می‌کنیم و مورد آنالیز قرار می‌دهیم. این حلقه زمانی پایان می‌یابد که دکمه `Escape` روی صفحه کلید فشار داده شود.

```
while (true) {
Mat cameraFrame;
videoCapture >> cameraFrame; دریافت فریم بعدی از دوربین
```

در صورتی که فریمی از دوربین دریافت نشود، خطایی در خروجی نمایش داده می‌شود و برنامه پایان می‌پذیرد

```
if(cameraFrame.empty()) {
cerr << "ERROR: Couldn't grab the next camera frame." << endl;
exit(1);
{
```

بر روی آن گرفته می‌شود. GUI یک کپی از فریم دریافت شده برای نمایش

```
Mat displayedFrame;
cameraFrame.copyTo(displayedFrame);
```

```
int identity = -1;
// Find a face and preprocess it to have a standard size and contrast & brightness.
```

این متغیر، مکان صورت تشخیص داده شده در تصویر را برمی گرداند. Rect faceRect;

این دو متغیر موقعیت چپ و راست بالای صورت را نشان می دهند. از این دو متغیر برای جستجوی چشم ها استفاده می شود.

Rect searchedLeftEye, searchedRightEye;

مکان چشم های چپ و راست تشخیص داده شده را نشان می دهند. Point leftEye, rightEye;

در خط بعدی تابع getPreprocessedFace فراخوانی می شود. این تابع بعد از دریافت فریم ضبط شده و تنظیمات انجام شده، به دنبال تشخیص صورت در فریم می گردد. در صورت تشخیص صورت، اندازه ی صورت، روشنایی و کانتراست آن را به حالت استاندارد تغییر می دهد.

```
Mat preprocessedFace = getPreprocessedFace(displayedFrame, faceWidth, faceCascade,
eyeCascade1, eyeCascade2, preprocessLeftAndRightSeparately, &faceRect, &leftEye,
&rightEye, &searchedLeftEye, &searchedRightEye);
```

```
bool gotFaceAndEyes = false;
if (preprocessedFace.data)
 gotFaceAndEyes = true;
// Draw an anti-aliased rectangle around the detected face.
```

اگر صورتی تشخیص داده شده باشد، دور صورت یک مربع سبز رسم می کند. و در صورت تشخیص هر یک از چشم ها، یک دایره آبی روی هر کدام رسم می کند.

```
if (faceRect.width > 0) {
 rectangle(displayedFrame, faceRect, CV_RGB(255, 255, 0), 2, CV_AA);
 // Draw light-blue anti-aliased circles for the 2 eyes.
 Scalar eyeColor = CV_RGB(0,255,255);

 if (leftEye.x >= 0) { // Check if the eye was detected
 circle(displayedFrame, Point(faceRect.x + leftEye.x, faceRect.y + leftEye.y), 6,
eyeColor, 1, CV_AA);
 }
```

```
 if (rightEye.x >= 0) { // Check if the eye was detected
 circle(displayedFrame, Point(faceRect.x + rightEye.x, faceRect.y + rightEye.y), 6,
eyeColor, 1, CV_AA);
 }
}
```

در ادامه کد متناسب با این که موقعیت برنامه در چه حالتی قرار دارد، کار متناسب با آن را انجام می-دهد.

```
if (m_mode == MODE_DETECTION) {
 // Don't do anything special. در این حالت نیاز به انجام کار خاصی در اینجا نیست.
}
```

```
else if (m_mode == MODE_COLLECT_FACES) {
```

در این حالت، ابتدا چک می‌شود که آیا صورت در فریم تشخیص داده شده است یا خیر. در صورتی که تشخیص داده نشده بود، کاری انجام نمیدهد و برنامه به ابتدای حلقه برمی‌گردد و فریم بعدی را می‌خواند. اما اگر تشخیص داده شده بود ادامه کد اجرا می‌شود.

```
if (gotFaceAndEyes) {
```

در صورت تشخیص صورت در فریم جاری، میزان تفاوت آن با صورت تشخیص داده شده در فریم قبلی محاسبه می‌شود.

```
double imageDiff = 10000000000.0;
if (old_preprocessedFace.data) {
 imageDiff = getSimilarity(preprocessedFace, old_preprocessedFace);
}
```

همچنین میزان زمان بین دریافت فریم جاری با فریم قبلی که در آن صورت تشخیص داده شده بود را محاسبه می‌کند.

```
double current_time = (double)getTickCount();
double timeDiff_seconds = (current_time - old_time)/getTickFrequency();
```

تصویر صورت تشخیص داده شده، در صورتی پردازش می‌شود که میزان تفاوت تصویر (imageDiff) و میزان اختلاف زمان دریافت فریم‌ها (timeDiff\_seconds) از حد مشخصی بیشتر باشد.

```
if ((imageDiff > CHANGE_IN_IMAGE_FOR_COLLECTION) &&
(timeDiff_seconds > CHANGE_IN_SECONDS_FOR_COLLECTION)) {
```

در این برنامه تصویر چرخش آینه‌ای صورت نیز محاسبه و در مجموعه داده‌های آموزش قرار داده می‌شود. این کار به منظور غنی‌تر کردن مجموعه داده‌های آموزش است.

```
Mat mirroredFace;
flip(preprocessedFace, mirroredFace, 1);
```

سپس تصویر صورت تشخیص داده شده به همراه تصویر آینه‌ای از آن، به لیست تصاویر تشخیص داده شده برای فردی که در حال حاضر روبروی دوربین قرار دارد، اضافه می‌شوند.

```
preprocessedFaces.push_back(preprocessedFace);
preprocessedFaces.push_back(mirroredFace);
faceLabels.push_back(m_selectedPerson);
faceLabels.push_back(m_selectedPerson);
```

یک اشاره‌گر به آخرین تصویر غیر آینه‌ای که از هر فرد گرفته شده است، ذخیره می‌شود. این اشاره‌گر برای بررسی با فریم‌های بعدی، مورد استفاده قرار می‌گیرد.

```
m_latestFaces[m_selectedPerson] = preprocessedFaces.size() - 2; // Point to the non-
mirrored face.
```

کد زیر صرفاً برای نمایش تعداد تصاویر گرفته شده از فرد مورد نظر در GUI برنامه است. این عدد بدون احتساب تصاویر آینه‌ای است.

```
cout << "Saved face " << (preprocessedFaces.size()/2) << " for person >> "
m_selectedPerson << endl;
```

برای اینکه کاربر متوجه شود که چه زمانی تصویر صورت آن تشخیص داده شده و آن فریم در مجموعه آموزش قرار داده شده است، بعد از تشخیص صورت، در GUI در مستطیل صورت یک فلش سفید نمایش داده می‌شود.

```
Mat displayedFaceRegion = displayedFrame(faceRect);
displayedFaceRegion += CV_RGB(90,90,90);
// Keep a copy of the processed face, to compare on next iteration.
```

برای تشخیص میزان اعتبار تصویر صورت در فریم بعدی، تصویر صورت در فریم جاری و زمان

کنونی در دو متغیر زیر نگه‌داری می‌شوند.

```
old_preprocessedFace = preprocessedFace;
old_time = current_time;
{
{
{
```

```
else if (m_mode == MODE_TRAINING) {
```

اگر موقعیت در حالت یادگیری (MODE\_TRAINING) قرار داشته باشد، ابتدا چک می‌شود که آیا تعداد داده‌های آموزشی ما به تعداد مناسبی هستند یا خیر. در صورتی که از Eigenfaces استفاده کنیم، میتوانیم مدل را فقط برای یک فرد هم آموزش دهیم. اما در صورتی که از Fisherfaces استفاده کنیم، برای آموزش مدل حداقل باید داده‌های آموزش برای دو نفر را در اختیار داشته باشیم.

```
bool haveEnoughData = true;
```

چک کردن تعداد افراد برای Fisherfaces :

```
if (strcmp(facerecAlgorithm, "FaceRecognizer.Fisherfaces") == 0) {
 if ((m_numPersons < 2) || (m_numPersons == 2 && m_latestFaces[1] < 0)) {
 cout << "Warning: Fisherfaces needs atleast 2 people, otherwise there is nothing
to differentiate! Collect more data ..." << endl;
 haveEnoughData = false;
 }
}
```

چک کردن وجود یا عدم وجود داده برای آموزش:

```
if (m_numPersons < 1 || preprocessedFaces.size() <= 0 || preprocessedFaces.size() !=
faceLabels.size()) {
 cout << "Warning: Need some training data before it can be learnt! Collect more
data ..." << endl;
 haveEnoughData = false;
}
```

اگر تعداد داده‌های آموزشی به اندازه کافی موجود بود، تابع learnCollectedFaces برای آموزش مدل، فراخوانی می‌شود.

```
if (haveEnoughData) {
```

```
 model = learnCollectedFaces(preprocessedFaces, faceLabels, facerecAlgorithm);
```

در بخش زیر، اگر متغیر m\_debug مقدار true داشته باشد، تصاویر مورد استفاده برای آموزش را به ما نشان می‌دهد. این بخش صرفاً برای دیباگ کردن برنامه برای برنامه‌نویس است.

```
if (m_debug)
```



|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <pre>showTrainingDebugData(model, faceWidth, faceHeight);</pre> <p>حال آموزش تمام شده است و برنامه دوباره به موقعیت تشخیص چهره (MODE_DETECTION) منتقل می‌شود.</p> <pre>m_mode = MODE_DETECTION;</pre> <pre>{</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |  |
| <p>اگر تعداد داده‌های آموزشی به اندازه کافی موجود نبود، موقعیت برنامه به حالت جمع‌آوری داده (MODE_COLLECT_FACES) برمیگردد.</p> <pre>else {</pre> <pre>    // Since there isn't enough training data, go back to the face collection mode!</pre> <pre>    m_mode = MODE_COLLECT_FACES;</pre> <pre>{</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                      |  |
| <p>کلیک شده باشد، در موقعیت GUI) Delete all اگر بر روی دکمه حذف همه (MODE_DELETE_ALL قرار می‌گیرد و کدهای بخش زیر اجرا می‌شوند. در این بخش همه متغیرها MODE_DELETE_ALL به حالت اولیه برمی‌گردند و داده‌های ضبط شده قبلی از لیست حذف می‌شوند و موقعیت برنامه دوباره قرار می‌گیرد. MODE_DETECTION در حالت</p> <pre>} else if (m_mode == MODE_DELETE_ALL) {</pre> <pre>    // Restart everything!</pre> <pre>    m_selectedPerson = -1;</pre> <pre>    m_numPersons = 0;</pre> <pre>    m_latestFaces.clear();</pre> <pre>    preprocessedFaces.clear();</pre> <pre>    faceLabels.clear();</pre> <pre>    old_preprocessedFace = Mat();</pre> <pre>    // Restart in Detection mode.</pre> <pre>    m_mode = MODE_DETECTION;</pre> <pre>{</pre> |  |
| <p>اگر موقعیت برنامه چیزی غیر از موارد قبلی باشد، در خروجی خطایی را نمایش می‌دهد و برنامه پایان می‌پذیرد.</p> <pre>else {</pre> <pre>    cerr &lt;&lt; "ERROR: Invalid run mode " &lt;&lt; m_mode &lt;&lt; endl;</pre> <pre>    exit(1);</pre> <pre>{</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |  |

کدهایی که در ادامه این بخش آمده‌اند، برای نمایش نوشته‌هایی بر روی GUI است که به کاربر موقعیت برنامه و همچنین تعداد تصاویر گرفته شده از فرد را نمایش می‌دهد. به عبارت دیگر کدهای زیر برای کمک به کاربر و در بخش GUI هستند.

```
Rect rcHelp;
if (m_mode == MODE_DETECTION)
 help = "Click [Add Person] when ready to collect faces.";
else if (m_mode == MODE_COLLECT_FACES)
 help = "Click anywhere to train from your " + toString(preprocessedFaces.size()/2) +
" faces of " + toString(m_numPersons) + " people.";
else if (m_mode == MODE_TRAINING)
 help = "Please wait while your " + toString(preprocessedFaces.size()/2) + " faces of "
+ toString(m_numPersons) + " people builds.";
if (help.length() > 0) {
 // Draw it with a black background and then again with a white foreground.
 // Since BORDER may be 0 and we need a negative position, subtract 2 from the
border so it is always negative.
 float txtSize = 0.4;
 drawString(displayedFrame, help, Point(BORDER, -BORDER-2), CV_RGB(0,0,0),
txtSize); // Black shadow.
 rcHelp = drawString(displayedFrame, help, Point(BORDER+1, -BORDER-1),
CV_RGB(255,255,255), txtSize); // White text.
}
```

```
// Show the current mode.
if (m_mode >= 0 && m_mode < MODE_END) {
 string modeStr = "MODE: " + string(MODE_NAMES[m_mode]);
 drawString(displayedFrame, modeStr, Point(BORDER, -BORDER-2-
rcHelp.height), CV_RGB(0,0,0)); // Black shadow
 drawString(displayedFrame, modeStr, Point(BORDER+1, -BORDER-1-
rcHelp.height), CV_RGB(0,255,0)); // Green text
}
```

بخش زیر برای نمایش تصاویر پیش‌پردازش شده در GUI است.

```
// Show the current preprocessed face in the top-center of the display.
int cx = (displayedFrame.cols - faceWidth) / 2;
if (preprocessedFace.data) {
 // Get a BGR version of the face, since the output is BGR color.
 Mat srcBGR = Mat(preprocessedFace.size(), CV_8UC3);
 cvtColor(preprocessedFace, srcBGR, CV_GRAY2BGR);
 // Get the destination ROI (and make sure it is within the image!).
 //min(m_gui_faces_top + i * faceHeight, displayedFrame.rows - faceHeight);
 Rect dstRC = Rect(cx, BORDER, faceWidth, faceHeight);
 Mat dstROI = displayedFrame(dstRC);
 // Copy the pixels from src to dst.
 srcBGR.copyTo(dstROI);
}
```

```
// Draw an anti-aliased border around the face, even if it is not shown.
rectangle(displayedFrame, Rect(cx-1, BORDER-1, faceWidth+2, faceHeight+2),
CV_RGB(200,200,200), 1, CV_AA);
```

رسم دکمه‌های GUI بر روی تصویر displayedFrame .

```
// Draw the GUI buttons into the main image.
m_rcBtnAdd = drawButton(displayedFrame, "Add Person", Point(BORDER,
BORDER));
m_rcBtnDel = drawButton(displayedFrame, "Delete All", Point(m_rcBtnAdd.x,
m_rcBtnAdd.y + m_rcBtnAdd.height), m_rcBtnAdd.width);
m_rcBtnDebug = drawButton(displayedFrame, "Debug", Point(m_rcBtnDel.x,
m_rcBtnDel.y + m_rcBtnDel.height), m_rcBtnAdd.width);
```

در بخش زیر آخرین تصویر خیره شده از هر فرد را در بالای پنجره‌ی GUI سمت راست نشان می‌دهد.

```
// Show the most recent face for each of the collected people, on the right side of the display.
m_gui_faces_left = displayedFrame.cols - BORDER - faceWidth;
m_gui_faces_top = BORDER;
for (int i=0; i<m_numPersons; i++) {
 int index = m_latestFaces[i];
 if (index >= 0 && index < (int)preprocessedFaces.size()) {
 Mat srcGray = preprocessedFaces[index];
 if (srcGray.data) {
 // Get a BGR version of the face, since the output is BGR color.
 Mat srcBGR = Mat(srcGray.size(), CV_8UC3);
 cvtColor(srcGray, srcBGR, CV_GRAY2BGR);
 // Get the destination ROI (and make sure it is within the image!).
 int y = min(m_gui_faces_top + i * faceHeight, displayedFrame.rows -
faceHeight);
 Rect dstRC = Rect(m_gui_faces_left, y, faceWidth, faceHeight);
 Mat dstROI = displayedFrame(dstRC);
 // Copy the pixels from src to dst.
 srcBGR.copyTo(dstROI);
 }
 }
}
```

در این بخش، اگر در موقعیت دریافت تصاویر برای آموزش (MODE\_COLLECT\_FACES) باشیم، دور تصویر فردی که در حال دریافت تصویر برای آن هستیم یک مربع قرمز رسم می‌کند.

```
// Highlight the person being collected, using a red rectangle around their face.
if (m_mode == MODE_COLLECT_FACES) {
 if (m_selectedPerson >= 0 && m_selectedPerson < m_numPersons) {
 int y = min(m_gui_faces_top + m_selectedPerson * faceHeight,
displayedFrame.rows - faceHeight);
 Rect rc = Rect(m_gui_faces_left, y, faceWidth, faceHeight);
 rectangle(displayedFrame, rc, CV_RGB(255,0,0), 3, CV_AA);
 }
}
```

در این بخش دور صورت فرد که تشخیص داده شده است یک مستطیل سبز رسم می‌کند. این کار برای اطلاع به کاربر است که متوجه شود چه زمانی صورت تشخیص داده شده است و آن تصویر برای آموزش ذخیره شده است.

```
// Highlight the person that has been recognized, using a green rectangle around their face.
if (identity >= 0 && identity < 1000) {
 int y = min(m_gui_faces_top + identity * faceHeight, displayedFrame.rows - faceHeight);
 Rect rc = Rect(m_gui_faces_left, y, faceWidth, faceHeight);
 rectangle(displayedFrame, rc, CV_RGB(0,255,0), 3, CV_AA);
}
```

نمایش تصویر displayedFrame همراه با تمام تنظیمات GUI ای که بر روی آن انجام شده است.

```
// Show the camera frame on the screen.
imshow(windowName, displayedFrame);
```

اگر کاربر درخواست نمایش تصاویر دیباگ را داده باشد، یعنی مقدار متغیر m\_debug برابر true باشد، این بخش اجرا می‌شود. و در نتیجه‌ی اجرای این بخش، تصاویر مربوط به چشم‌های چپ و راستی که تشخیص داده شده است در پنجره‌های جداگانه‌ای نمایش داده می‌شوند.

```
if (m_debug) {
 Mat face;
 if (faceRect.width > 0) {
 face = cameraFrame(faceRect);
 if (searchedLeftEye.width > 0 && searchedRightEye.width > 0) {
 Mat topLeftOfFace = face(searchedLeftEye);
 Mat topRightOfFace = face(searchedRightEye);
 imshow("topLeftOfFace", topLeftOfFace);
 imshow("topRightOfFace", topRightOfFace);
 }
 {
 if (!model.empty())
 showTrainingDebugData(model, faceWidth, faceHeight);
 }
 }
}
```

در انتهای این تابع ۲۰ میلی ثانیه صبر می‌کند که چک کند آیا کاربر کلید Escape را زده است یا نه. در صورت فشردن کلید Escape توسط کاربر، برنامه پایان می‌یابد. در غیر این صورت برنامه به ابتدای حلقه while رفته و تکرار عملیات‌های قبلی انجام می‌شود.

```
char keypress = waitKey(20); // This is needed if you want to see anything!
if (keypress == VK_ESCAPE) { // Escape Key
 // Quit the program!
 break;
}
} //end while
{
```

تابع main، تابع اصلی برنامه است که ترتیب اجرای توابع دیگر را کنترل می‌کند.

```
int main(int argc, char *argv[])
{
 CascadeClassifier faceCascade;
 CascadeClassifier eyeCascade1;
 CascadeClassifier eyeCascade2;
 VideoCapture videoCapture;
 // Load the face and 1 or 2 eye detection XML classifiers.
 initDetectors(faceCascade, eyeCascade1, eyeCascade2);
 cout << endl;
 cout << "Hit 'Escape' in the GUI window to quit." << endl;

 در این بخش می‌توان شماره دوربین مورد نظر خود را تنظیم کنید.

 int cameraNumber = 0; // Change this if you want to use a different camera device.
 if (argc > 1) {
 cameraNumber = atoi(argv[1]);
 }
 // Get access to the webcam.
 initWebcam(videoCapture, cameraNumber);

 // Try to set the camera resolution. Note that this only works for some cameras on
 // some computers and only for some drivers, so don't rely on it to work!
 videoCapture.set(CV_CAP_PROP_FRAME_WIDTH, DESIRED_CAMERA_WIDTH);
 videoCapture.set(CV_CAP_PROP_FRAME_HEIGHT, DESIRED_CAMERA_HEIGHT);

 // Create a GUI window for display on the screen.
 namedWindow(windowName); // Resizable window, might not work on Windows.
 // Get OpenCV to automatically call my "onMouse()" function when the user clicks in the
 GUI window.
 setMouseCallback(windowName, onMouse, 0);

 // Run Face Recognition interactively from the webcam. This function runs until the user
 quits.
 recognizeAndTrainUsingWebcam(videoCapture, faceCascade, eyeCascade1,
 eyeCascade2);
 return 0;
}
```

## ۲-۱ توضیح کد Main\_predict

این کد، بازشناسی چهره را در فریم ورودی بر اساس مدلی که در Main\_train.cpp آموزش داده شده است، انجام می‌دهد. در ۲۰ فریم سعی در شناسایی فرد دیده شده در تصویر دارد. در صورت شناسایی، اسم و مکان صورت مورد نظر را در فایل rec.txt می‌نویسد و در صورت شناسایی نشدن،

مکان صورت را به همراه رشته unknown در فایل rec.txt می‌نویسد. همچنین در صورتی که طی ۵ ثانیه صورتی را در تصویر تشخیص ندهد، رشته NotDetect را در فایل rec.txt می‌نویسد. در بخش زیر، الگوریتم بازشناسی چهره را تعیین می‌کنیم. باید همان الگوریتمی که برای آموزش مدل استفاده شده است را استفاده کرد.

```
//FaceRecognizer.Eigenfaces": Eigenfaces, also referred to as PCA (Turk and Pentland, ۱۹۹۱).
//FaceRecognizer.Fisherfaces": Fisherfaces, also referred to as LDA (Belhumeur et al, 7991).
//FaceRecognizer.LBPH": Local Binary Pattern Histograms (Ahonen et al, 6002).
const char *facerecAlgorithm = "FaceRecognizer.Fisherfaces;"
//const char *facerecAlgorithm = "FaceRecognizer.Eigenfaces;"
//const char *facerecAlgorithm = "FaceRecognizer.LBPH ;"
```

در کد پایین، یک حد آستانه برای تشخیص افراد آموزش دیده و افراد ناآشنا تنظیم می‌شود. مقدار مناسب برای الگوریتم eigenface برابر ۰٫۵ و برای الگوریتم Fisherface برابر ۰٫۷ است. هرچه مقدار این حد آستانه کم‌تر شود، احتمال برچسب‌زنی افراد به عنوان فرد ناشناس بیشتر می‌شود.

```
const float UNKNOWN_PERSON_THRESHOLD = ۰٫۷f ;
```

بخش زیر کدهای مشترک با فایل Main\_train.cpp هستند که قبلاً توضیح داده شده است. برای خواندن توضیحات بیشتر پیرامون این قطعه کد، به تابع Main\_train.cpp رجوع شود.

```
// Cascade Classifier file, used for Face Detection.
const char *faceCascadeFilename = "lbpcascade_frontalface.xml"; // LBP face detector.
//const char *faceCascadeFilename = "haarcascade_frontalface_alt_tree.xml"; // Haar face detector.
//const char *eyeCascadeFilename1 = "haarcascade_lefteye_2splits.xml"; // Best eye detector for open-or-closed eyes.
//const char *eyeCascadeFilename2 = "haarcascade_righteye_2splits.xml"; // Best eye detector for open-or-closed eyes.
//const char *eyeCascadeFilename1 = "haarcascade_mcs_lefteye.xml"; // Good eye detector for open-or-closed eyes.
//const char *eyeCascadeFilename2 = "haarcascade_mcs_righteye.xml"; // Good eye detector for open-or-closed eyes.
const char *eyeCascadeFilename1 = "haarcascade_eye.xml"; // Basic eye detector for open eyes only.
const char *eyeCascadeFilename2 = "haarcascade_eye_tree_eyeglasses.xml"; // Basic eye detector for open eyes if they might wear glasses.

// Set the desired face dimensions. Note that "getPreprocessedFace()" will return a square face.
```

```

const int faceWidth = 70;
const int faceHeight = faceWidth;

// Try to set the camera resolution. Note that this only works for some cameras on
// some computers and only for some drivers, so don't rely on it to work!
const int DESIRED_CAMERA_WIDTH = 1920;//640;1920;
const int DESIRED_CAMERA_HEIGHT = 1080;//480;1080;

const bool preprocessLeftAndRightSeparately = true; // Preprocess left & right sides of the
face separately, in case there is stronger light on one side.
// Set to true if you want to see many windows created, showing various debug info. Set to 0
otherwise.
bool m_debug = false;

#include <stdio.h>
#include <vector>
#include <string>
#include <iostream>
#include <fstream>
#include <time.h>

// Include OpenCV's C++ Interface
#include "opencv2/opencv.hpp"

// Include the rest of our code!
#include "detectObject.h" // Easily detect faces or eyes (using LBP or Haar Cascades).
#include "preprocessFace.h" // Easily preprocess face images, for face recognition.
#include "recognition.h" // Train the face recognition system and recognize a person from
an image.
#include "ImageUtils.h" // Shervin's handy OpenCV utility functions.

using namespace cv;
using namespace std;
#ifdef VK_ESCAPE
#define VK_ESCAPE 0x1B // Escape character (27)
#endif
string label;
// C++ conversion functions between integers (or floats) to std::string.
template <typename T> string toString(T t)
{
 ostringstream out;
 out << t;
 return out.str();
}

// Load the face and 1 or 2 eye detection XML classifiers.
void initDetectors(CascadeClassifier &faceCascade, CascadeClassifier &eyeCascade1,
CascadeClassifier &eyeCascade2)
{
 // Load the Face Detection cascade classifier xml file.
 try { // Surround the OpenCV call by a try/catch block so we can give a useful error

```

```

message!
 faceCascade.load(faceCascadeFilename);
} catch (cv::Exception &e) {}
if (faceCascade.empty()) {
 cerr << "ERROR: Could not load Face Detection cascade classifier [" <<
faceCascadeFilename << "]" << endl;
 cerr << "Copy the file from your OpenCV data folder (eg:
'C:\\OpenCV\\data\\lbpcascades') into this WebcamFaceRec folder." << endl;
 exit(1);
}
 cout << "Loaded the Face Detection cascade classifier [" << faceCascadeFilename << "]"
<< endl;
 // Load the Eye Detection cascade classifier xml file.
 try { // Surround the OpenCV call by a try/catch block so we can give a useful error
message!
 eyeCascade1.load(eyeCascadeFilename1);
 } catch (cv::Exception &e) {}
 if (eyeCascade1.empty()) {
 cerr << "ERROR: Could not load 1st Eye Detection cascade classifier [" <<
eyeCascadeFilename1 << "]" << endl;
 cerr << "Copy the file from your OpenCV data folder (eg:
'C:\\OpenCV\\data\\haarcascades') into this WebcamFaceRec folder." << endl;
 exit(1);
 }
 cout << "Loaded the 1st Eye Detection cascade classifier [" << eyeCascadeFilename1 <<
"]." << endl;

 // Load the Eye Detection cascade classifier xml file.
 try { // Surround the OpenCV call by a try/catch block so we can give a useful error
message!
 eyeCascade2.load(eyeCascadeFilename2);
 } catch (cv::Exception &e) {}
 if (eyeCascade2.empty()) {
 cerr << "Could not load 2nd Eye Detection cascade classifier [" <<
eyeCascadeFilename2 << "]" << endl;
 // Dont exit if the 2nd eye detector did not load, because we have the 1st eye detector at
least.
 //exit(1);
 {
 else
 cout << "Loaded the 2nd Eye Detection cascade classifier [" << eyeCascadeFilename2
<< "]" << endl;
 }
 }
}

```

تابع زیر، تابع اصلی بازشناسی چهره است که طی دریافت ۲۰ فریم سعی در بازشناسی چهره فرد موجود در تصویر را دارد.



```
void recognizeAndTrainUsingWebcam(CascadeClassifier &faceCascade, CascadeClassifier
&eyeCascade), CascadeClassifier &eyeCascade){
}
Ptr<FaceRecognizer> model;
```

در کد زیر سعی در تعریف و اتصال به دوربین است. در صورتی که دوربینی یافت نشود، خطایی چاپ می‌کند و برنامه پایان می‌پذیرد. در غیر این صورت ادامه کد اجرا می‌شود.

```
VideoCapture cam(*);
if(!cam.isOpened()){
 cout<<"ERROR not opened "<< endl;
 // return -1;
}else{
```

قبل ادامه کد، زمان شروع ذخیره می‌شود. این زمان برای محاسبه مدت زمانی است که هیچ صورتی در تصویر دیده نشده باشد.

```
clock_t t_start_video,t_duration_video;
t_start_video=clock();
```

شروع حلقه دریافت فریم‌ها، تلاش برای بازشناسی چهره تا ۲۰ فریم انجام می‌شود.

```
float seconds=0;
while (seconds<20) {

Mat cameraFrame;
bool b=cam.read(cameraFrame);
if(!b){
 cout<<"ERROR : cannot read"<<endl;
 //return -1;
}
```

تلاش برای برقراری ارتباط با دوربین

انجام پیش‌پردازش بر روی تصویر ورودی و اعمال الگوریتم تشخیص صورت.

```
int identity = -1;
// Find a face and preprocess it to have a standard size and contrast & brightness.
Rect faceRect; // Position of detected face.
Rect searchedLeftEye, searchedRightEye; // top-left and top-right regions of the face, where
eyes were searched.
Point leftEye, rightEye; // Position of the detected eyes.
Mat preprocessedFace = getPreprocessedFace(cameraFrame, faceWidth, faceCascade,
eyeCascade1, eyeCascade2, preprocessLeftAndRightSeparately, &faceRect, &leftEye,
&rightEye, &searchedLeftEye, &searchedRightEye);
```

اگر در تصویر صورت فردی تشخیص داده شده بود، متغیر `preprocessedFace.data` مقدار خواهد داشت و متغیر `gotFaceAndEyes` را با مقدار `true` مقداردهی می‌کند.

```
bool gotFaceAndEyes = false;
if (preprocessedFace.data)
 gotFaceAndEyes = true;
```

سپس زمان کنونی را اندازه می‌گیرید. اگر مدت زمان طی شده بیشتر از ۵ ثانیه باشد (۵ ثانیه هیچ صورتی در تصویر تشخیص داده نشده باشد)، در فایل `rec.txt` رشته `NotDetect` را چاپ می‌کند و برنامه پایان می‌یابد.

```
t_duration_video=clock();
float diff_detect ((float)t_duration_video-(float)t_start_video);
float seconds_detect = diff_detect / CLOCKS_PER_SEC;
if (seconds_detect > 5){
 ofstream myfile;
 myfile.open("rec.txt", std::ofstream::out | std::ofstream::trunc);
 myfile << "person : "<< "NotDetect" <<endl;
 myfile.close();
 break;
}
```

در صورتی که صورت فردی در فریم ورودی تشخیص داده شود، ادامه برنامه وارد بخش زیر می‌شود.

```
if (gotFaceAndEyes) {
 قبل از هر کاری، زمان شروع شمارنده برای تشخیص صورت را به روز رسانی می‌کند.
 t_start_video=clock();
 سپس مدل آموزش داده شده در بخش آموزش را در برنامه بارگذاری می‌کند.
```

```
Mat reconstructedFace;
model = Algorithm::create<FaceRecognizer>(facerecAlgorithm);
model->load("/home/pi/Desktop/shervin/build/new_face_model.yml");
```

سپس بر اساس مدل بارگذاری شده و تصویر پیش پردازش شده، میزان شباهت تصویر ورودی را به هر یک از کلاس‌های موجود محاسبه می‌کند و مقدار بیشترین شباهت را در متغیر `similarity` ذخیره می‌کند. (هر چه شباهت کم‌تر باشد مقدار عددی موجود در متغیر `similarity` کم‌تر می‌شود.)

```
reconstructedFace = reconstructFace(model, preprocessedFace);
double similarity = getSimilarity(preprocessedFace, reconstructedFace);
```

سپس در صورتی که میزان شباهت از حد آستانه‌ای که برای اعلام فرد ناشناس تعیین شده بود کم‌تر باشد، یعنی این تصویر می‌تواند متعلق به یکی از افراد آموزش دیده باشد. در این صورت تصویر را به تابع `predict` ارسال می‌کند و برچسب فرد تخمین زده شده را برمی‌گرداند. در غیر این صورت، برچسب را با رشته `"Unknown"` مقداردهی می‌کند.

```
string outputStr;
if (similarity < UNKNOWN_PERSON_THRESHOLD) {
 // Identify who the person is in the preprocessed face image.
```

```

identity = model->predict(preprocessedFace);
outputStr = toString(identity);
}else {
 // Since the confidence is low, assume it is an unknown person.
 outputStr = "Unknown";
 {
 cout << "Identity: " << outputStr << ". Similarity: " << similarity << endl;

```

در بخش زیر در متغیر label اسم فرد تخمین زده شده را ذخیره می‌کند. و نام و موقعیت صورت فرد را در فایل rec.txt ذخیره می‌کند.

```

label="";
if (outputStr=="0"){
 label = "Afshar";
}else if (outputStr=="1"){
 label = "Ramezani";
}else if (outputStr=="2"){
 label = "Gomar";
}else if (outputStr=="Unknown"){
 label = "unknown";
 {
 ofstream myfile;
 myfile.open ("rec.txt");
 myfile << "person : " << label << endl;
 myfile << "position: " << Point(faceRect.x
+(leftEye.x+rightEye.x)/2,faceRect.y+(leftEye.y+rightEye.y)/2) << endl;
 myfile.close();
 }
}
در صورتی که فردی را تشخیص دهد، برنامه پایان می‌یابد.
if(label!=" " && label!="unknown"){
 break;
}
}
} //end if
} //end while
} //end else
{

```

تابع Main نیز تابع اصلی برنامه است که ترتیب اجرای توابع را کنترل می‌کند.

```

int main(int argc, char *argv[])
{
 CascadeClassifier faceCascade;
 CascadeClassifier eyeCascade1;
 CascadeClassifier eyeCascade2;
 VideoCapture videoCapture;
 // Load the face and 1 or 2 eye detection XML classifiers.
 initDetectors(faceCascade, eyeCascade1, eyeCascade2);

 // Try to set the camera resolution. Note that this only works for some cameras on

```

```
// some computers and only for some drivers, so don't rely on it to work!
videoCapture.set(CV_CAP_PROP_FRAME_WIDTH,
DESIRED_CAMERA_WIDTH);
videoCapture.set(CV_CAP_PROP_FRAME_HEIGHT ,
DESIRED_CAMERA_HEIGHT);

// Run Face Recogintion interactively from the webcam. This function runs until the
user quits.
recognizeAndTrainUsingWebcam(faceCascade,eyeCascade1, eyeCascade2);
return 0;
{
```

### ۳-۱ توضیح کد detect\_face.cpp

این کد در واقع بخشی از کد Main\_predict.cpp است که فقط به منظور اعلام تشخیص یا عدم تشخیص صورت در تصویر نوشته شده است. به همین دلیل از توضیح مباحث تکراری اجتناب شده است و فقط بخش‌های جدید توضیح داده شده است.

انتخاب نوع دسته‌بند تشخیص صورت

```
const char *faceCascadeFilename = "lbpcascade_frontalface.xml" ;
//const char *faceCascadeFilename = "haarcascade_frontalface_alt_tree.xml" ;
```

انتخاب نوع دسته‌بند تشخیص چشم

```
//const char *eyeCascadeFilename1 ="haarcascade_lefteye_2splits.xml" ;
//const char *eyeCascadeFilename2 ="haarcascade_righteye_2splits.xml" ;
//const char *eyeCascadeFilename1 ="haarcascade_mcs_lefteye.xml" ;
//const char *eyeCascadeFilename2 ="haarcascade_mcs_righteye.xml" ;
const char *eyeCascadeFilename1 ="haarcascade_eye.xml
const char *eyeCascadeFilename2 ="haarcascade_eye_tree_eyeglasses.xml" ;
```

تنظیم پارامترهای مورد نیاز (توضیح راجع به این پارامترها در کدهای قبلی توضیح داده شده است).

```
// Set the desired face dimensions. Note that "getPreprocessedFace()" will return a square
face.
```

```
const int faceWidth = 70;
const int faceHeight = faceWidth;
```

```
// Try to set the camera resolution. Note that this only works for some cameras on
// some computers and only for some drivers, so don't rely on it to work!
const int DESIRED_CAMERA_WIDTH = 1920;
const int DESIRED_CAMERA_HEIGHT = 1080;
```

```

const bool preprocessLeftAndRightSeparately = true;
#include <fstream>

// Include OpenCV's C++ Interface
#include "opencv2/opencv.hpp"

// Include the rest of our code!
#include "preprocessFace.h" // Easily preprocess face images, for face recognition.

using namespace cv;
using namespace std;

 بارگزاری دسته‌بندی‌های تشخیص صورت و چشم

// Load the face and 1 or 2 eye detection XML classifiers.
Void initDetectors(CascadeClassifier &faceCascade, CascadeClassifier &eyeCascade1,
CascadeClassifier &eyeCascade2)
{
 // Load the Face Detection cascade classifier xml file.
 Try { // Surround the OpenCV call by a try/catch block so we can give a useful error
message!
 faceCascade.load(faceCascadeFilename);
 } catch (cv::Exception &e) {}
 if (faceCascade.empty()) {
 cerr << "ERROR: Could not load Face Detection cascade classifier [" <<
faceCascadeFilename << "]" << endl;
 cerr << "Copy the file from your OpenCV data folder (eg:
'C:\\OpenCV\\data\\lbpcascades') into this WebcamFaceRec folder." << endl;
 exit(1);
 }
 cout << "Loaded the Face Detection cascade classifier [" << faceCascadeFilename << "]"
<< endl;

 // Load the Eye Detection cascade classifier xml file.
 Try { // Surround the OpenCV call by a try/catch block so we can give a useful error
message!
 eyeCascade1.load(eyeCascadeFilename1);
 } catch (cv::Exception &e) {}
 if (eyeCascade1.empty()) {
 cerr << "ERROR: Could not load 1st Eye Detection cascade classifier [" <<
eyeCascadeFilename1 << "]" << endl;
 cerr << "Copy the file from your OpenCV data folder (eg:
'C:\\OpenCV\\data\\haarcascades') into this WebcamFaceRec folder." << endl;
 exit(1);
 }
 cout << "Loaded the 1st Eye Detection cascade classifier [" << eyeCascadeFilename1 <<
"]" << endl;

 // Load the Eye Detection cascade classifier xml file.
 Try { // Surround the OpenCV call by a try/catch block so we can give a useful error
message!
 eyeCascade2.load(eyeCascadeFilename2);
 } catch (cv::Exception &e) {}

```

```

if (eyeCascade2.empty()) {
 cerr << "Could not load 2nd Eye Detection cascade classifier [" <<
eyeCascadeFilename2 << "]" << endl;
 // Don't exit if the 2nd eye detector did not load, because we have the 1st eye detector at
least.
 //exit(1);
 {
 else
 cout << "Loaded the 2nd Eye Detection cascade classifier [" << eyeCascadeFilename2
<< "]" << endl;
 {

```

این تابع، تابع اصلی تشخیص صورت است که در یک حلقه ۱۰ فریم را می‌گیرد و در آن‌ها به جستجوی صورت می‌پردازد. اگر در یک فریم صورت را تشخیص دهد، از حلقه خارج شده و بعد از نوشتن کلمه true در فایل detect.txt برنامه پایان می‌یابد. در غیر این صورت، بعد تلاش در ۱۰ فریم، بعد از نوشتن کلمه false در فایل detect.txt برنامه پایان می‌یابد.

```

Void Detect(CascadeClassifier &faceCascade, CascadeClassifier &eyeCascade1,
CascadeClassifier &eyeCascade2)
{

```

ایجاد فایل detect.txt و مقداردهی اولیه آن با مقدار false.

```

ofstream myfile;
myfile.open ("detect.txt");
myfile << "false"<<endl;
myfile.close();

```

اتصال به دوربین

```

VideoCapture cam(0);
if(!cam.isOpened()){
 cout<<"ERROR not opened "<< endl;
}else{

```

شروع حلقه دریافت فریم از دوربین

```

int time=0;
bool gotFaceAndEyes = false;
while (time<10) {
 time=time+1;
 // Grab the next camera frame.

```

دریافت فریم بعدی

```

Mat cameraFrame;
bool b=cam.read(cameraFrame);
if(!b){
 cout<<"ERROR : cannot read"<<endl;
 //return -;1
 {

```

انجام پیش‌پردازش بر روی فریم دریافت شده و اعمال الگوریتم تشخیص صورت

```

// Find a face and preprocess it to have a standard size and contrast & brightness.
Rect faceRect; // Position of detected face.
Rect searchedLeftEye, searchedRightEye; // top-left and top-right regions of the
face, where eyes were searched.
Point leftEye, rightEye; // Position of the detected eyes.
Mat preprocessedFace = getPreprocessedFace(cameraFrame, faceWidth,
faceCascade, eyeCascade1, eyeCascade2, preprocessLeftAndRightSeparately, &faceRect,
&leftEye, &rightEye, &searchedLeftEye, &searchedRightEye);

```

اگر در متغیر `preprocessedFace.data` مقداری وجود داشته باشد، به معنی تشخیص صورت در تصویر ورودی بوده است. پس در فایل `detect.txt` عبارت `true` نوشته می‌شود و با `reak` از حلقه `while` و در نهایت از کل برنامه خارج می‌شود. در غیر این صورت، عبارت برای دریافت فریم بعدی و سعی مجدد تلاش می‌کند.

```

gotFaceAndEyes = false;
if (preprocessedFace.data)
 gotFaceAndEyes = true;

if (gotFaceAndEyes) {
 ofstream myfile;
 myfile.open ("detect.txt");
 myfile << "true"<<endl;
 myfile.close();
 break;
}
} //end while

```

در صورتی که ۱۰ فریم دریافت شود و در هیچ یک از آن‌ها صورت تشخیص داده نشود، عبارت `false` در فایل `detect.txt` نوشته می‌شود.

```

if (!gotFaceAndEyes) {
 ofstream myfile;
 myfile.open ("detect.txt");
 myfile << "false"<<endl;
 myfile.close();
}
} //end else

} //end detect function

```

تابع زیر، تابع اصلی برنامه است که توابع مورد نیاز را به ترتیب فراخوانی می‌کند.

```

int main(int argc, char *argv[])
{
 CascadeClassifier faceCascade;
 CascadeClassifier eyeCascade1;
 CascadeClassifier eyeCascade2;

```

```
VideoCapture videoCapture;

// Load the face and 1 or 2 eye detection XML classifiers.
initDetectors(faceCascade, eyeCascade1, eyeCascade2);

// Try to set the camera resolution. Note that this only works for some cameras on
// some computers and only for some drivers, so don't rely on it to work!
videoCapture.set(CV_CAP_PROP_FRAME_WIDTH, DESIRED_CAMERA_WIDTH);
videoCapture.set(CV_CAP_PROP_FRAME_HEIGHT, DESIRED_CAMERA_HEIGHT);

Detect(faceCascade, eyeCascade1, eyeCascade2);
return 0;
{
```

#### ۴-۱ توضیح کد detectObject.cpp

در این فایل سه تابع `detectObjectsCustom`، `detectLargestObject` و `detectManyObjects` برای تشخیص صورت در تصویر وجود دارد.

- تابع `detectObjectsCustom` با توجه به پارامترهای دریافتی، در جستجوی **همه‌ی صورت و چشم‌ها** در تصویر دریافتی می‌گردد. و به ازای هر صورتی که تشخیص دهد، یک مستطیل در متغیر `object` قرار می‌دهد.
- تابع `detectManyObjects`، موقعیت تمام صورت‌های یافت شده در تصویر ورودی را برمی‌گرداند. در این تابع بعد از تنظیم پارامترها تابع `detectObjectsCustom` را فراخوانی می‌کند.
- تابع `detectLargestObject`، فقط موقعیت **بزرگ‌ترین صورت** یا به عبارت دیگر نزدیک‌ترین صورت را در تصویر دریافتی برمی‌گرداند. این تابع بسیار سریع‌تر از تابع `detectManyObjects` عمل می‌کند.



```
#include "detectObject.h" // Easily detect faces or eyes (using LBP or Haar Cascades).
```

```
void detectObjectsCustom(const Mat &img, CascadeClassifier &cascade, vector<Rect>
&objects, int scaledWidth, int flags, Size minFeatureSize, float searchScaleFactor, int
minNeighbors)
```

```
}
```

الگوریتم‌های تشخیص صورت بر روی تصاویر مقیاس خاکستری<sup>۱۱۷</sup> اعمال می‌شوند. به همین دلیل اگر تصاویر از نوع مقیاس خاکستری نباشند، ابتدا این تبدیل بر روی آن انجام می‌شود.

```
Mat gray;
```

```
if (img.channels() == 3){
```

```
 cvtColor(img, gray, CV_BGR2GRAY);
```

```
{ else if (img.channels() == 4){
```

```
 cvtColor(img, gray, CV_BGRA2GRAY);
```

```
{ else }
```

```
 // Access the input image directly, since it is already grayscale.
```

```
 gray = img;
```

```
{
```

در این بخش، با توجه به پارامتر ورودی scaledWidth سعی می‌شود که در صورت بزرگ بودن اندازه تصویر ورودی، به منظور افزایش سرعت الگوریتم، آن را کوچک کند.

```
//Possibly shrink the image, to run much faster.
```

```
Mat inputImg;
```

```
float scale = img.cols / (float)scaledWidth;
```

```
if (img.cols > scaledWidth){
```

```
 // Shrink the image while keeping the same aspect ratio.
```

```
 int scaledHeight = cvRound(img.rows / scale);
```

```
 resize(gray, inputImg, Size(scaledWidth, scaledHeight));
```

```
{ else}
```

```
 // Access the input image directly, since it is already small.
```

```
 inputImg = gray;
```

```
{
```

در بخش زیر، میزان کنتراست و روشنایی تصویر به حد استاندارد تغییر می‌کند. این کار باعث می‌شود که کیفیت تصویرهای تاریک بهتر شوند.

```
Mat equalizedImg;
```

<sup>117</sup> Grayscale

```
equalizeHist(inputImg, equalizedImg);
```

در بخش زیر، تصویری را که پیش‌پردازش‌های بالا بر روی آن اعمال شده است را برای تشخیص صورت به تابع `detectMultiScale` ارسال می‌کند.

```
//Detect objects in the small grayscale image.
cascade.detectMultiScale(equalizedImg, objects, searchScaleFactor, minNeighbors, flags,
minFeatureSize);
```

اگر قبل تشخیص صورت تصویر به طور موقت کوچک شده باشد، اکنون آن را دوباره بزرگ می‌کنیم.

```
// Enlarge the results if the image was temporarily shrunk before detection.
if (img.cols > scaledWidth){
 for (int i = 0; i < (int)objects.size(); i++) {
 objects[i].x = cvRound(objects[i].x * scale);
 objects[i].y = cvRound(objects[i].y * scale);
 objects[i].width = cvRound(objects[i].width * scale);
 objects[i].height = cvRound(objects[i].height * scale);
 }
}
```

در بخش زیر چک می‌شود که تمام صورت تشخیص داده شده درون تصویر قرار داشته باشد.

```
// Make sure the object is completely within the image, in case it was on a border.
for (int i = 0; i < (int)objects.size(); i++) {
 if (objects[i].x < 0(
 objects[i].x = 0;
 if (objects[i].y < 0(
 objects[i].y = 0;
 if (objects[i].x + objects[i].width > img.cols)
 objects[i].x = img.cols - objects[i].width;
 if (objects[i].y + objects[i].height > img.rows)
 objects[i].y = img.rows - objects[i].height;
}
// Return with the detected face rectangles stored in "objects."
} //end function
```

این تابع بزرگ‌ترین تصویر صورت یا به عبارتی نزدیک‌ترین صورت را برمی‌گرداند.

```
void detectLargestObject(const Mat &img, CascadeClassifier &cascade, Rect
&largestObject, int scaledWidth)
{
 // Only search for just ۱ object (the biggest in the image).
 int flags = CASCADE_FIND_BIGGEST_OBJECT; // |
 CASCADE_DO_ROUGH_SEARCH;
 // Smallest object size.
 Size minFeatureSize = Size(۲۰, ۲۰);
 // How detailed should the search be. Must be larger than ۱,۰.
 float searchScaleFactor = ۱,۱f;
 // How much the detections should be filtered out. This should depend on how bad false
 detections are to your system.
 // minNeighbors= ۲ means lots of good+bad detections, and minNeighbors= ۴ means
 only good detections are given but some are missed.
 int minNeighbors = ۴;

 // Perform Object or Face Detection, looking for just ۱ object (the biggest in the image).
 vector<Rect> objects;
 detectObjectsCustom(img, cascade, objects, scaledWidth, flags, minFeatureSize,
 searchScaleFactor, minNeighbors);
 if (objects.size() > ۰ {
 // Return the only detected object.
 largestObject = (Rect)objects.at(۰);
 // cout<< "largestObject"<< largestObject<< endl;
 } else {
 // Return an invalid rect.
 largestObject = Rect(۱-, ۱-, ۱-, ۱-);
 }
}
```

این تابع تمام صورت‌های تشخیص داده شده در تصویر ورودی را برمی‌گرداند.

```
void detectManyObjects(const Mat &img, CascadeClassifier &cascade, vector<Rect>
&objects, int scaledWidth)
{
 // Search for many objects in the one image.
 int flags = CASCADE_SCALE_IMAGE;

 // Smallest object size.
 Size minFeatureSize = Size(۲۰, ۲۰)
 // How detailed should the search be. Must be larger than ۱,۰.
 float searchScaleFactor = ۱,۱f;
 // How much the detections should be filtered out. This should depend on how bad false
 detections are to your system.
 // minNeighbors=2 means lots of good+bad detections, and minNeighbors= ۶ means only
 good detections are given but some are missed.
 int minNeighbors = 4;

 // Perform Object or Face Detection, looking for many objects in the one image.
 detectObjectsCustom(img, cascade, objects, scaledWidth, flags, minFeatureSize,
 searchScaleFactor, minNeighbors);
}
```

#### ۵-۱) فایل preprocessFace.cpp

در این فایل عملیات‌های پیش‌پردازشی برای بهبود تصویر دریافتی انجام می‌شود. از جمله‌ی این عملیات‌ها می‌توان به حذف پس‌زمینه و موهای فرد مورد نظر اشاره کرد. این کار باعث می‌شود که وابستگی الگوریتم آموزش به این دو مورد که همواره قابل تغییر هستند، از بین برود. در این فایل چندین تابع وجود دارد که در ادامه به توضیح هر یک پرداخته می‌شود.

- `getPreprocessedFace`: یک تصویر grayscale با اندازه و میزان روشنایی و کنتراست استاندارد از تصویر ورودی ایجاد می‌کند. اگر متغیر `doLeftAndRightSeparately` با مقدار `true` مقداردهی شده باشد، هر یک از چشم‌ها را به طور جداگانه بررسی می‌کند. این کار باعث می‌شود که اگر یک سمت صورت روشن و سمت دیگر تاریک باشد، دقت تشخیص چشم‌ها بیشتر شود.

به طور خلاصه عملیات پیش‌پردازش به صورت زیر انجام می‌شود:

- جابه‌جایی و چرخش و `scale` کردن هندسی با استفاده از تشخیص چشم

- حذف نویز با استفاده از فیلتر Bilateral
  - استاندارد کردن میزان روشنایی در هر دو سمت صورت به طور مستقل با استفاده از الگوریتم Histogram Equalization
  - حذف پس زمینه و موها با استفاده از یک ماسک بیضوی
- چیزی که این تابع برمی گرداند، یا یک مربع از صورت است و یا اینکه null برمی گرداند. اگر صورت تشخیص داده شود، مکان صورت و چشم چپ و راست را به ترتیب در متغیرهای storeFaceRect، storeLeftEye و storeRightEye ذخیره می کند.
- scaleRectFromCenter: این تابع پس زمینه و حرکات را حذف می کند و بر اساس متغیر scale و مرکز صورت تشخیص داده شده، تصویر صورت را scale می کند.
  - detectBothEyes: این تابع موقعیت دو چشم را جستجو می کند و مرکز چشم چپ و راست را در صورت تشخیص برمی گرداند. امکان استفاده از دو نوع دسته بند چشم برای تشخیص چشم چپ و راست وجود دارد.
  - equalizeLeftAndRightHalves: در این تابع به طور جداگانه الگوریتم Histogram Equalize بر روی دو سمت صورت اعمال می شود. به طور معمول، در تصاویر گرفته شده یک سمت صورت تاریک تر از سمت دیگر است. اگر الگوریتم Histogram Equalize بر روی کل صورت اعمال شود، آنگاه یک سمت صورت روشن و سمت دیگر همچنان تیره تر خواهد بود. به همین دلیل این الگوریتم به طور جداگانه بر هر دو سمت صورت اعمال می شود. این کار باعث می شود که هر دو سمت صورت به طور میانگین یکسان شوند. البته این کار باعث می شود که به دلیل تفاوت در دو سمت صورت یک لبه تیز در وسط صورت ایجاد شود. به همین دلیل یک Histogram Equalize نیز دوباره بر روی کل تصویر اعمال می شود. این کار باعث می شود که لبه وسط صورت کمی محو شود. در آخر هم برای تصویر بهتر داشتن، هر سه تصویر با هم ترکیب می شوند.

۲-۶) تنظیمات اولیه برنامه

```
const double DESIRED_LEFT_EYE_X = 61.0; // Controls how much of the face is visible
after preprocessing.
const double DESIRED_LEFT_EYE_Y = 0.14;
const double FACE_ELLIPSE_CY = 0.04;
```

```
const double FACE_ELLIPSE_W = 05.0; // Should be atleast 5.0
const double FACE_ELLIPSE_H = 08.0; // Controls how tall the face mask is.

#include "detectObject.h" // Easily detect faces or eyes (using LBP or Haar Cascades).
#include "preprocessFace.h" // Easily preprocess face images, for face recognition.
#include "ImageUtils.h" // Shervin's handy OpenCV utility functions.
```

```
Mat getPreprocessedFace(Mat &srcImg, int desiredFaceWidth, CascadeClassifier
&faceCascade, CascadeClassifier &eyeCascade\, CascadeClassifier &eyeCascade\, bool
doLeftAndRightSeparately, Rect *storeFaceRect, Point *storeLeftEye, Point
*storeRightEye, Rect *searchedLeftEye, Rect *searchedRightEye)
{
```

```
// Use square faces.
```

```
int desiredFaceHeight = desiredFaceWidth;
```

مقداردهی اولیه‌ی متغیرهای مربوط به صورت تشخیص داده شده با مقادیر نامعتبر

```
// Mark the detected face region and eye search regions as invalid, in case they aren't
detected.
```

```
if (storeFaceRect)
```

```
storeFaceRect->width = -;
```

```
if (storeLeftEye)
```

```
storeLeftEye->x = -;
```

```
if (storeRightEye)
```

```
storeRightEye->x = -;
```

```
if (searchedLeftEye)
```

```
searchedLeftEye->width = -;
```

```
if (searchedRightEye)
```

```
searchedRightEye->width = -;
```

با فراخوانی تابع detectLargestObject، مکان بزرگ‌ترین صورت (نزدیک‌ترین صورت) را بدست می‌آورد.

```
// Find the largest face.
```

```
Rect faceRect;
```

```
detectLargestObject(srcImg, faceCascade, faceRect);
```

در صورتی که صورت فردی را در تصویر تشخیص داده باشد، وارد بخش زیر می‌شود.

```
// Check if a face was detected.
```

```
if (faceRect.width > 0)
```

```
// Give the face rect to the caller if desired.
```

```
if (storeFaceRect)
```

```
* storeFaceRect = faceRect;
```

```
Mat faceImg = srcImg(faceRect); // Get the detected face image.
// If the input image is not grayscale, then convert the BGR or BGRA color image to grayscale.
```

در صورت رنگی بودن تصویر ورودی آن را به قالب grayscale می‌برد.

```
Mat gray;
if (faceImg.channels() == 3){
 cvtColor(faceImg, gray, CV_BGR2GRAY);
}
else if (faceImg.channels() == 4){
 cvtColor(faceImg, gray, CV_BGRA2GRAY);
} else{
// Access the input image directly, since it is already grayscale.
 gray = faceImg;
}

جستجو برای پیدا کردن چشم‌ها در بالاترین رزولوشن ممکن

// Search for the 2 eyes at the full resolution, since eye detection needs max resolution possible!
Point leftEye, rightEye;
detectBothEyes(gray, eyeCascade1, eyeCascade2, leftEye, rightEye,
searchedLeftEye, searchedRightEye);

// Give the eye results to the caller if desired.
if (storeLeftEye)
* storeLeftEye = leftEye;
if (storeRightEye)
* storeRightEye = rightEye;

در صورتی که هر دو چشم تشخیص داده شدند، تغییر اندازه و چرخش و انتقال صورت بر اساس آن-
ها انجام می‌شود. اندازه‌ی صورت را به اندازه صورت‌های ذخیره شده در مرحله آموزش می‌رساند.

// Check if both eyes were detected.
if (leftEye.x >= * && rightEye.x >= * (}

// Make the face image the same size as the training images.
// Since we found both eyes, lets rotate & scale & translate the face so that the 2
eyes
// line up perfectly with ideal eye positions. This makes sure that eyes will be
horizontal,
// and not too far left or right of the face, etc.
```

مرکز دو چشم و زاویه‌ی آن‌ها محاسبه می‌شود.

```

// Get the center between the 2 eyes.
Point2f eyesCenter = Point2f((leftEye.x + rightEye.x) * 0.5f, (leftEye.y +
rightEye.y) * 0.5f);
// Get the angle between the 2 eyes.
double dy = (rightEye.y - leftEye.y);
double dx = (rightEye.x - leftEye.x);
double len = sqrt(dx*dx + dy*dy);
double angle = atan2(dy, dx) * 180./CV_PI; // Convert from radians to
degrees.
// Hand measurements shown that the left eye center should ideally be at roughly
(0.19, 0.14) of a scaled face image.
const double DESIRED_RIGHT_EYE_X = (1.0f - DESIRED_LEFT_EYE_X);
// Get the amount we need to scale the image to be the desired fixed size we want.
double desiredLen = (DESIRED_RIGHT_EYE_X - DESIRED_LEFT_EYE_X) *
desiredFaceWidth;
double scale = desiredLen / len;
// Get the transformation matrix for rotating and scaling the face to the desired
angle & size.
Mat rot_mat = getRotationMatrix2D(eyesCenter, angle, scale);
// Shift the center of the eyes to be the desired center between the eyes.
rot_mat.at<double>(0,2) += desiredFaceWidth * 0.5f - eyesCenter.x;
rot_mat.at<double>(1,2) += desiredFaceHeight * DESIRED_LEFT_EYE_Y -
eyesCenter.y;

// Rotate and scale and translate the image to the desired angle & size & position!
// Note that we use 'w' for the height instead of 'h', because the input face has 1:1
aspect ratio.
Mat warped = Mat(desiredFaceHeight, desiredFaceWidth, CV_8U, Scalar(821));
// Clear the output image to a default grey.
warpAffine(gray, warped, rot_mat, warped.size());
// imshow("warped", warped);

// Give the image a standard brightness and contrast, in case it was too dark or had
low contrast.
if (!doLeftAndRightSeparately)
// Do it on the whole face.
 equalizeHist(warped, warped);
{ Else}
// Do it separately for the left and right sides of the face.
 equalizeLeftAndRightHalves(warped);
{
// imshow("equalized", warped);

```



```
// Use the "Bilateral Filter" to reduce pixel noise by smoothing the image, but
keeping the sharp edges in the face.
 Mat filtered = Mat(warped.size(), CV_8U);
 bilateralFilter(warped, filtered, ۰, ۲۰, ۰, ۲, ۰);
// imshow("filtered", filtered);

// Filter out the corners of the face, since we mainly just care about the middle
parts.
// Draw a filled ellipse in the middle of the face-sized image.
 Mat mask = Mat(warped.size(), CV_8U, Scalar(۰)); // Start with an empty mask.
 Point faceCenter = Point(desiredFaceWidth/۲, cvRound(desiredFaceHeight *
FACE_ELLIPSE_CY));
 Size size = Size(cvRound(desiredFaceWidth * FACE_ELLIPSE_W),
cvRound(desiredFaceHeight * FACE_ELLIPSE_H));
 ellipse(mask, faceCenter, size, ۰, ۰, ۳۶۰, Scalar(۲۵۵), CV_FILLED);
// imshow("mask", mask);

// Use the mask, to remove outside pixels.
 Mat dstImg = Mat(warped.size(), CV_8U, Scalar(۱۲۸)); // Clear the output
image to a default gray.
 */

 namedWindow("filtered");
 imshow("filtered", filtered);
 namedWindow("dstImg");
 imshow("dstImg", dstImg);
 namedWindow("mask");
 imshow("mask", mask);

 /*
// Apply the elliptical mask on the face.
 filtered.copyTo(dstImg, mask); // Copies non-masked pixels from filtered to
dstImg.
// imshow("dstImg", dstImg);

 return dstImg;
{
 */
 else {
// Since no eyes were found, just do a generic image resize.
 resize(gray, tmpImg, Size(w,h));
 }
 /*
```

```
{
 return Mat;()
}
```

## ۷-۲) تابع equalizeLeftAndRightHalves

```
// Histogram Equalize separately for the left and right sides of the face.
void equalizeLeftAndRightHalves(Mat &faceImg)
{
 int w = faceImg.cols;
 int h = faceImg.rows;

 // 1) First, equalize the whole face.
 Mat wholeFace;
 equalizeHist(faceImg, wholeFace);

 // 2) Equalize the left half and the right half of the face separately.
 int midX = w/2;
 Mat leftSide = faceImg(Rect(0,0, midX,h));
 Mat rightSide = faceImg(Rect(midX,0, w-midX,h));
 equalizeHist(leftSide, leftSide);
 equalizeHist(rightSide, rightSide);

 // 3) Combine the left half and right half and whole face together, so that it has a smooth
 transition.
 for (int y=0; y<h; y++) {
 for (int x=0; x<w; x++) {
 int v;
 if (x < w/4) { // Left 25%: just use the left face.
 v = leftSide.at<uchar>(y,x);
 }
 else if (x < w*2/4) { // Mid-left 25%: blend the left face & whole face.
 int lv = leftSide.at<uchar>(y,x);
 int wv = wholeFace.at<uchar>(y,x);
 // Blend more of the whole face as it moves further right along the face.
 float f = (x - w*1/4) / (float)(w*0.25f);
 v = cvRound((1.0f - f) * lv + (f) * wv);
 }
 else if (x < w*3/4) { // Mid-right 25%: blend the right face & whole face.
 int rv = rightSide.at<uchar>(y,x-midX);
 int wv = wholeFace.at<uchar>(y,x);
 // Blend more of the right-side face as it moves further right along the face.
 float f = (x - w*2/4) / (float)(w*0.25f);
 v = cvRound((1.0f - f) * wv + (f) * rv);
 }
 else { // Right 25%: just use the right face.
 v = rightSide.at<uchar>(y,x-midX);
 }
 faceImg.at<uchar>(y,x) = v;
 }
 }
}
```

```

 } // end x loop
 } // end y loop
{

```

## تابع detectBothEyes (۸-۲)

```

void detectBothEyes(const Mat &face, CascadeClassifier &eyeCascade1, CascadeClassifier
&eyeCascade2, Point &leftEye, Point &rightEye, Rect *searchedLeftEye, Rect
*searchedRightEye)
{
 // Skip the borders of the face, since it is usually just hair and ears, that we don't care
 about.
 // For default eye.xml or eyeglasses.xml: Finds both eyes in roughly 40% of detected
 faces, but does not detect closed eyes.
 const float EYE_SX = 0.16f;
 const float EYE_SY = 0.26f;
 const float EYE_SW = 0.30f;
 const float EYE_SH = 0.28f;

 int leftX = cvRound(face.cols * EYE_SX);
 int topY = cvRound(face.rows * EYE_SY);
 int widthX = cvRound(face.cols * EYE_SW);
 int heightY = cvRound(face.rows * EYE_SH);
 int rightX = cvRound(face.cols * (1.0-EYE_SX-EYE_SW)); // Start of right-eye corner

 Mat topLeftOfFace = face(Rect(leftX, topY, widthX, heightY));
 Mat topRightOfFace = face(Rect(rightX, topY, widthX, heightY));
 Rect leftEyeRect, rightEyeRect;

 // Return the search windows to the caller, if desired.
 if (searchedLeftEye)
 *searchedLeftEye = Rect(leftX, topY, widthX, heightY);
 if (searchedRightEye)
 *searchedRightEye = Rect(rightX, topY, widthX, heightY);

 // Search the left region, then the right region using the 1st eye detector.
 detectLargestObject(topLeftOfFace, eyeCascade1, leftEyeRect, topLeftOfFace.cols);
 detectLargestObject(topRightOfFace, eyeCascade1, rightEyeRect, topRightOfFace.cols);

 //If the eye was not detected, try a different cascade classifier.
 if (leftEyeRect.width <= 0 && !eyeCascade2.empty()) {
 detectLargestObject(topLeftOfFace, eyeCascade2, leftEyeRect, topLeftOfFace.cols);
 //if (leftEyeRect.width > 0)
 // cout << "2nd eye detector LEFT SUCCESS" << endl;
 //else
 // cout << "2nd eye detector LEFT failed" << endl;
 }
 //else
 // cout << "1st eye detector LEFT SUCCESS" << endl;

 // If the eye was not detected, try a different cascade classifier.
 if (rightEyeRect.width <= 0 && !eyeCascade2.empty()) {

```

```

 detectLargestObject(topRightOfFace, eyeCascade2, rightEyeRect,
topRightOfFace.cols);
 //if (rightEyeRect.width > 0)
 // cout << "2nd eye detector RIGHT SUCCESS" << endl;
 //else
 // cout << "2nd eye detector RIGHT failed" << endl;
 {
 //else
 // cout << "1st eye detector RIGHT SUCCESS" << endl;

 if (leftEyeRect.width > 0) { // Check if the eye was detected.
 leftEyeRect.x += leftX; // Adjust the left-eye rectangle because the face border was
removed.
 leftEyeRect.y += topY;
 leftEye = Point(leftEyeRect.x + leftEyeRect.width/2, leftEyeRect.y +
leftEyeRect.height/2);
 {
 else {
 leftEye = Point(-1,-1); // Return an invalid point
 }

 if (rightEyeRect.width > 0) { // Check if the eye was detected.
 rightEyeRect.x += rightX; // Adjust the right-eye rectangle, since it starts on the right
side of the image.
 rightEyeRect.y += topY; // Adjust the right-eye rectangle because the face border was
removed.
 rightEye = Point(rightEyeRect.x + rightEyeRect.width/2, rightEyeRect.y +
rightEyeRect.height/2);
 {
 else {
 rightEye = Point(-1,-1); // Return an invalid point
 }
 }
 }
 }

```

۲-۹) تابع scaleRectFromCenter:

```

Rect scaleRectFromCenter(const Rect wholeFaceRect, float scale)
{
 float faceCenterX = wholeFaceRect.x + wholeFaceRect.width * 0.5f;
 float faceCenterY = wholeFaceRect.y + wholeFaceRect.height * 0.5f;
 float newWidth = wholeFaceRect.width * scale;
 float newHeight = wholeFaceRect.height * scale;
 Rect faceRect;
 faceRect.width = cvRound(newWidth); // Shrink the region
 faceRect.height = cvRound(newHeight);
 faceRect.x = cvRound(faceCenterX - newWidth * 0.5f); // Move the region so that the
center is still the same spot.
 faceRect.y = cvRound(faceCenterY - newHeight * 0.5f);

 return faceRect;
}

```

## ۲-۱۰) فایل recognition.cpp

توابع موجود در این فایل برای انجام آموزش مدل بازشناسی چهره با استفاده از مجموعه تصاویر آموزش و همچنین تخمین تشابه تصویر ورودی با تصاویر آموزش داده شده است. این فایل شامل توابع زیر است:

۱) learnCollectedFaces: این تابع تصویر پیش پردازش شده، برچسب افرادی که تصاویر آنها

برای آموزش استفاده شده است و الگوریتم مورد نظر برای آموزش را دریافت می کند. و

مدل آموزش داده شده را برمی گرداند. همچنین یک نسخه از مدل آموزش داده شده را برای

استفاده های بعدی، بر روی سیستم خیره می کند.

۲) getImageFrom1DFloatMat: یک ماتریس سطری یا ستونی را به صورت دو بعدی و قابل

نمایش در قالب تصویر در می آورد. همچنین مقادیر را به مقادیر بین ۰ تا ۲۵۵ تبدیل می کند.

۳) showTrainingDebugData: این تابع مراحل میانی بازشناسی چهره را به کاربر نشان می -

دهد. و صرفاً جهت کمک رسانی به کاربر برای دیباگ کردن بهتر استفاده می شود.

۴) reconstructFace: تولید تخمینی از صورت بازسازی شده توسط eigenvectors و

eigenvalues از تصویر صورت دریافت شده.

۵) getSimilarity: مقایسه دو تصویر با استفاده از معیار خطای  $L_2$  (square-root of sum of

squared error)

تنظیمات اولیه ی کد:

```
#include "recognition.h" // Train the face recognition system and recognize a person from
an image.
#include "ImageUtils.h" // Shervin's handy OpenCV utility functions.
```

## ۲-۱۱) تابع learnCollectedFaces

```
Ptr<FaceRecognizer> learnCollectedFaces(const vector<Mat> preprocessedFaces, const
vector<int> faceLabels, const string facerecAlgorithm)
{
 Ptr<FaceRecognizer> model;
 cout << "Learning the collected faces using the [" << facerecAlgorithm << "] algorithm
... " << endl;

 // Make sure the "contrib" module is dynamically loaded at runtime.
 // Requires OpenCV v ۲,۴,۱ or later (from June ۲۰۱۲), otherwise the FaceRecognizer
 will not compile or run!
```

```
bool haveContribModule = initModule_contrib();
if (!haveContribModule){
 cerr << "ERROR: The 'contrib' module is needed for FaceRecognizer but has not been
loaded into OpenCV!" << endl;
 exit(1);
}

// Use the new FaceRecognizer class in OpenCV's "contrib" module:
// Requires OpenCV v ۲,۴,۱ or later (from June ۲۰۱۲), otherwise the FaceRecognizer
will not compile or run!
model = Algorithm::create<FaceRecognizer>(facerecAlgorithm);
if (model.empty()) {
 cerr << "ERROR: The FaceRecognizer algorithm [" << facerecAlgorithm << "] is not
available in your version of OpenCV. Please update to OpenCV v ۲,۴,۱ or newer." << endl;
 exit(۱)
}

// Do the actual training from the collected faces. Might take several seconds or minutes
depending on input!
model->train(preprocessedFaces, faceLabels);
model->save("new_face_model.yml");
return model;
}
```

تابع `DFloatMat\getImageFrom` (۱۲-۲)

```
Mat getImageFrom1DFloatMat(const Mat matrixRow, int height)
{
 // Make it a rectangular shaped image instead of a single row.
 Mat rectangularMat = matrixRow.reshape(1, height);
 // Scale the values to be between 0 to 255 and store them as a regular 8-bit uchar image.
 Mat dst;
 normalize(rectangularMat, dst, 0, 255, NORM_MINMAX, CV_8UC1);
 return dst;
}
```

تابع `showTrainingDebugData` (۱۳-۲)

```
void showTrainingDebugData(const Ptr<FaceRecognizer> model, const int faceWidth, const
int faceHeight)
{
 try { // Surround the OpenCV calls by a try/catch block so we don't crash if some model
parameters aren't available.
 // Show the average face (statistical average for each pixel in the collected images).
 Mat averageFaceRow = model->get<Mat>("mean");
 printMatInfo(averageFaceRow, "averageFaceRow");
 }
```

```

// Convert the matrix row (1D float matrix) to a regular 8-bit image.
Mat averageFace = getImageFrom1DFloatMat(averageFaceRow, faceHeight);
printMatInfo(averageFace, "averageFace");
imshow("averageFace", averageFace);
// Get the eigenvectors
Mat eigenvectors = model->get<Mat>("eigenvectors");
printMatInfo(eigenvectors, "eigenvectors");

// Show the best 20 eigenfaces
for (int i = 0; i < min(20, eigenvectors.cols); i++) {
 // Create a column vector from eigenvector #i.
 // Note that clone() ensures it will be continuous, so we can treat it like an array,
 otherwise we can't reshape it to a rectangle.
 // Note that the FaceRecognizer class already gives us L2 normalized eigenvectors, so
 we don't have to normalize them ourselves.
 Mat eigenvectorColumn = eigenvectors.col(i).clone();
 //printMatInfo(eigenvectorColumn, "eigenvector");
 Mat eigenface = getImageFrom1DFloatMat(eigenvectorColumn, faceHeight);
 //printMatInfo(eigenface, "eigenface");
 imshow(format("Eigenface%d", i), eigenface);
}
// Get the eigenvalues
Mat eigenvalues = model->get<Mat>("eigenvalues");
printMat(eigenvalues, "eigenvalues");
//int ncomponents = model->get<int>("ncomponents");
//cout << "ncomponents = " << ncomponents << endl;
vector<Mat> projections = model->get<vector<Mat>>("projections");
cout << "projections: " << projections.size() << endl;
for (int i = 0; i < (int)projections.size(); i++) {
 printMat(projections[i], "projections");
}
//labels = model->get<Mat>("labels");
//printMat(labels, "labels");
} catch (cv::Exception e) {
 //cout << "WARNING: Missing FaceRecognizer properties." << endl;
}
{

```

## ۲-۱۴) تابع reconstructFace

```

Mat reconstructFace(const Ptr<FaceRecognizer> model, const Mat preprocessedFace)
{
 // Since we can only reconstruct the face for some types of FaceRecognizer models (ie:
 Eigenfaces or Fisherfaces),
 // we should surround the OpenCV calls by a try/catch block so we don't crash for other
 models.
 try {

 // Get some required data from the FaceRecognizer model.
 Mat eigenvectors = model->get<Mat>("eigenvectors");
 Mat averageFaceRow = model->get<Mat>("mean");

```

```

int faceHeight = preprocessedFace.rows;

// Project the input image onto the PCA subspace.
Mat projection = subspaceProject(eigenvectors, averageFaceRow,
preprocessedFace.reshape(1,1));
//printMatInfo(projection, "projection");

// Generate the reconstructed face back from the PCA subspace.
Mat reconstructionRow = subspaceReconstruct(eigenvectors, averageFaceRow,
projection);
//printMatInfo(reconstructionRow, "reconstructionRow");

// Convert the float row matrix to a regular 8-bit image. Note that we
// shouldn't use "getImageFrom1DFloatMat()" because we don't want to normalize
// the data since it is already at the perfect scale.

// Make it a rectangular shaped image instead of a single row.
Mat reconstructionMat = reconstructionRow.reshape(1, faceHeight);
// Convert the floating-point pixels to regular 8-bit uchar pixels.
Mat reconstructedFace = Mat(reconstructionMat.size(), CV_8U);
reconstructionMat.convertTo(reconstructedFace, CV_8U, 1, 0);
//printMatInfo(reconstructedFace, "reconstructedFace");

return reconstructedFace;

} catch (cv::Exception e) {
//cout << "WARNING: Missing FaceRecognizer properties." << endl;
return Mat();
}
{
{

```

## ۲-۱۵ تابع getSimilarity

```

double getSimilarity(const Mat A, const Mat B)
{
if (A.rows > 0 && A.rows == B.rows && A.cols > 0 && A.cols == B.cols) {
// Calculate the L2 relative error between the 2 images.
double errorL2 = norm(A, B, CV_L2);
// Convert to a reasonable scale, since L2 error is summed across all pixels of the image.
double similarity = errorL2 / (double)(A.rows * A.cols);
return similarity;
}
else {
//cout << "WARNING: Images have a different size in 'getSimilarity()'." << endl;
return 100000000.0; // Return a bad value
}
}
{

```



```

*****/

*Face Recognition using Eigenfaces or Fisherfaces

*by Shervin Emami, ۵th Dec ۲۰۱۲
*http://www.shervinemami.info/openCV.html

*Ch ۱ of the book "Mastering OpenCV with Practical Computer Vision Projects"
*Copyright Packt Publishing . ۲۰۱۲
*http://www.packtpub.com/cool-projects-with-opencv/book

/*****
*/

* ImageUtils
* Handy utility functions for dealing with images in OpenCV (desktop or Android).
* by Shervin Emami (shervin.emami@gmail.com), ۲۷th May . ۲۰۱۲
* The most recent version of this will always be available from
"http://shervinemami.info/openCV.html"
/*

#define USE_HIGHGUI // Enable this to display graph windows using OpenCV's
HighGUI. (Supports Windows, Linux & Mac, but not iPhone).

#include "ImageUtils.h"
using namespace std;

```

```

//Print the label and then some text info about the IplImage properties, to LOG() for easy
debugging.

void printImageInfo(const IplImage *image, const char *label)
{

```

```

string s; "" =

char buff;[۱۰۲۴]

if (label)
 s = label + string;(":")

if (image){
 snprintf(buff, sizeof(buff), "[Image] = %dw%dh, %d channels of %dbit depth,
widthStep=%d, origin=%d", image->width, image->height, image->nChannels, image-
>depth, image->widthStep, image->origin);
 s += buff;
 if (image->roi)
 snprintf(buff, sizeof(buff), " ROI=[at %d,%d of size %dx%d, COI=%d].\n", image-
>roi->xOffset, image->roi->yOffset, image->roi->width, image->roi->height, image->roi-
>coi);
 else
 snprintf(buff, sizeof(buff), " ROI=<null>.\n");
 s += buff;
 { else }
 s = "[Image] = <null>\n;"
 {
// LOG can be printf or similar.

LOG(s.c_str());
{

//Print the pixel values of the IplImage, to LOG() for easy debugging.
void printImagePixels(const IplImage *image, const char *label, int maxElements)
}

string s;
char buff;[۳۲]
if (label)
 s = label + string;(":")
else
 s = "Image;" :
if (image){
 if (maxElements == *(
 maxElements = image->width * image->height;
 sprintf(buff, "(%dw%dh):\n", image->width, image->height);
 s += string(buff);
 LOG(s.c_str());

```

```

int totalElements = ۰;

int depth = image->depth & ۲۵۵;

for (int row=۰; row < image->height; row++) {

 string s;"" =

 int element = ۰;

 if (image->nChannels > ۱ && image->height > ۱ {

 snprintf(buff, sizeof(buff), "row%d: ", row);
 {
 for (int col=۰; col < image->width; col++){

 if (image->nChannels > ۱(

 s;"]" =+

 for (int ch=۰; ch <= image->nChannels-۱; ch++) {

 if (ch > ۰ || (image->nChannels == ۱ && col != ۰)) // Add a separator, except
for the first element of this pixel or row.

 s;" ," =+

// Allow to print just part of the image.
 totalElements++;
 if (totalElements > maxElements){
// LOG can be printf or similar.

 LOG("%s ... <just displaying the ۱st %d entries from %d!>", s.c_str(),
maxElements, image->width * image->height * image->nChannels);
 return;
 }

 if (depth == ۱ // (۱-bit UCHAR image.
 snprintf(buff, sizeof(buff), "%d", image->imageData[(row * (image-
>widthStep)) + (col*image->nChannels) + ch]);
 else if (depth == ۱۶ // (۱۶-bit short image.
 snprintf(buff, sizeof(buff), "%d", *(short*)(uchar*)&image->imageData[(row
* image->widthStep) + ((col*image->nChannels) + ch) * sizeof(short)]);
 else if (depth == ۳۲ // (۳۲-bit float image.
 snprintf(buff, sizeof(buff), "%.۳f", *(float*)(uchar*)&image-
>imageData[(row * image->widthStep) + ((col*image->nChannels) + ch) * sizeof(float)]);

```

```

else if (depth == ۶۴ // (۶۴-bit double image.

 snprintf(buff, sizeof(buff), "%.۴lf", *(double*)(uchar*)&image-
>imageData[(row * image->widthStep) + ((col*image->nChannels) + ch) * sizeof(double)]);
 s += buff;

 const int MAX_ELEMENTS_PER_LINE = ۶۰; // Only print upto ۳۰ numbers
per LOG() statement, since Android can only handle about ۲۵۰ characters per log line!

 if (element > MAX_ELEMENTS_PER_LINE){

// LOG can be printf or similar.

 LOG(s.c_str());
 s;"" =
 element = ;۰
 {
 element; ++
 }
 if (image->nChannels > ۱(
 s;" [" =+
 {
// LOG can be printf or similar.

 LOG(s.c_str());
 {
 { else }
 LOG("[%s] = <null>!", s.c_str());
 {
 {

//Return the number of bits in each channel of the given Mat. ie: ۸ , ۱۶ , ۳۲ or ۶۴
int getBitDepth(const cv::Mat M)
}
switch (CV_MAT_DEPTH(M.type())){
 case CV_۸U:

 case CV_۸S:

 return ;۸

 case CV_۱۶U:

 case CV_۱۶S:

```

```

 return ;۱۶

 case CV_۳۲S:

 case CV_۳۲F:

 return ;۳۲

 case CV_۶۴F:

 return ;۶۴

 {
 return -;۱
 }

//Print the contents of a multi-channel array (using "LOG()") for easy debugging.
//If 'maxElements' is ۰ ,it will print the whole array. If it is -۱, it will not print the array at
all.

void printArray۲D(const uchar *data, int cols, int rows, int channels, int depth_type, int step,
int maxElements)
{
 char buff;[۳۲]

 if (data != ۰&& cols > ۰&& rows > ۰&& channels > ۰&& step > ۰({

// Show the actual data values

 if (maxElements >= ۰){

 if (maxElements == ۰(

 maxElements = rows * cols;
 int totalElements = ;۰

// int step = step;

 for (int row=۰; row < rows; row++){

 string s;"" =

 int element = ;۰

 if (channels > ۱&& rows > ۱){

 snprintf(buff, sizeof(buff), "row%d: ", row);

 {

```

```

for (int col=۰; col < cols; col++) {

 if (channels > ۱(

 s;""] +=

 for (int ch=۰; ch <= channels-۱; ch++){

 if (ch > ۰|| (channels == ۱&& col != ۰)) // Add a separator, except for the
first element of this pixel or row.

 s;"", " +=

 buff[۰] = '?'; // Initialize the string to "?" if something goes wrong.

 buff[۱] = [;۰

// Allow to print just part of the image.

 totalElements++;
 if (totalElements > maxElements) }
// LOG can be printf or similar.

 LOG("%s ... <just displaying the ۱st %d entries from %d!>", s.c_str(),
maxElements, rows * cols * channels);

 return;
 {

 switch (depth_type) {
 case CV_AU:

 case CV_AS: //۸-bit UCHAR Mat.

 snprintf(buff, sizeof(buff), "%d", data[(row * step) + (col*channels) + ch]);
 break;
 case CV_۱۶U:

 case CV_۱۶S: // ۱۶-bit short Mat.

 snprintf(buff, sizeof(buff), "%d", *(short*)(uchar*)&data[(row * step) +
((col*channels) + ch) * sizeof(short)]);
 break;
 case CV_۳۲S: // ۳۲-bit int Mat.

 snprintf(buff, sizeof(buff), "%d", *(int*)(uchar*)&data[(row * step) + ((col*channels) + ch)
* sizeof(int)]);
 break;

```

```

case CV_32F: // ۳۲-bit float Mat.
 snprintf(buff, sizeof(buff), "%.3f", *(float*)(uchar*)&data[(row * step) + ((col*channels) +
ch) * sizeof(float)]);
 break;
case CV_64F: // ۶۴-bit double Mat.
 snprintf(buff, sizeof(buff), "%.3lg", *(double*)(uchar*)&data[(row * step) + ((col*channels)
+ ch) * sizeof(double)]);
 break;
default:
 snprintf(buff, sizeof(buff), "UNKNOWN DEPTH OF %d!", depth_type);
}
s += buff;

const int MAX_ELEMENTS_PER_LINE = ۳۰; // Only print upto ۳۰ numbers per LOG()
statement, since Android can only handle about ۲۵۰ characters per log line!

if (element > MAX_ELEMENTS_PER_LINE){
// LOG can be printf or similar.

LOG(s.c_str());
s, "" =
element = ;
{
 Element++;
{
 if (channels > ۱(
 s, " [" +=
{
// LOG can be printf or similar.
LOG(s.c_str());
{
// { end if (maxElements>=0)
{
{
//Print the label and then contents of a cv::Mat from the C++ interface (using "LOG()") for
easy debugging.

```

//If 'maxElements' is ۰, it will print the whole array. If it is -۱, it will not print the array at all.

```
void printMat(const cv::Mat M, const char *label, int maxElements)
{
```

```
 string s;
```

```
 char buff;[۳۲]
```

```
 if (label)
```

```
 s = label + string;(" :")
```

```
 else
```

```
 s = "Mat;" :
```

```
 if (!M.empty())}
```

```
 int channels = CV_MAT_CN(M.type());
```

```
 int depth_bpp = getBitDepth(M); // eg: ۸ , ۱۶ , ۳۲
```

```
 int depth_type = CV_MAT_DEPTH(M.type()); // eg: CV_۳۲S, CV_۳۲F
```

```
// Show the dimensions & data type
```

```
 sprintf(buff, "%dw%dh %dch %dbpp", M.cols, M.rows, channels, depth_bpp);
```

```
 s += string(buff);
```

```
// Show the data range for each channel
```

```
 s += ", range;"
```

```
 for (int ch=۰; ch<channels; ch++)}
```

```
 cv::Mat arr = cv::Mat(M.rows, M.cols, depth_type);
```

```
// Extract one channel at a time, to show it's range.
```

```
 int from_to;[۲]
```

```
 from_to[۰] = ch;
```

```
 from_to[۱] = [;۰
```

```
 cv::mixChannels(&M, ۱, &arr, ۱, from_to, ۱);
```

```
// Show it's range.
```

```
 double minVal, maxVal;
```

```
 cv::minMaxLoc(arr, &minVal, &maxVal);
```

```
 snprintf(buff, sizeof(buff), "[%lg,%lg]", minVal, maxVal);
```

```
 s += buff;
```

```
 {
```

```
 LOG(s.c_str());
```



```
// Show the actual data values

 printArray%D(M.data, M.cols, M.rows, channels, depth_type, M.step, maxElements);
 {
 else }
 LOG("%s empty Mat", s.c_str());
 {
 {

//Print the label and info of a cv::Mat from the C++ interface (using "LOG()") for easy
debugging.
void printMatInfo(const cv::Mat M, const char *label)
}
 printMat(M, label, -1);
{

//Print the label and then contents of a cvMat from the C interface (using "LOG()") for easy
debugging.
void printMatrix(const CvMat *M, const char *label, int maxElements)
}
 string s;
 char buff;[۳۲]
 if (label)
 s = label + string;(":")
 else
 s = "Matrix;" :
 if (M) }
 if (maxElements == 0)
 maxElements = M->rows * M->cols;
 sprintf(buff, "[%drows x %dcols]:\n", M->rows, M->cols);
 s += string(buff);
 LOG(s.c_str());
 int channels = CV_MAT_CN(M->type);
 int depth = CV_MAT_DEPTH(M->type);
 int totalElements = ;
 uchar *data = (uchar*)M->data.ptr;
 int step = M->step;

 for (int row=0; row < M->rows; row++) }
```

```

string s;"" =
int element = ۰;
if (channels > ۱ && M->rows > ۱){
 snprintf(buff, sizeof(buff), "row%d: ", row);
{
 for (int col=۰; col < M->cols; col++) {
 if (channels > ۱(
 s;"]" =+
 for (int ch=۰; ch <= channels-۱; ch++) {
 if (ch > ۰ || (channels == ۱ && col != ۰)) // Add a separator, except for the
first element of this pixel or row.
 s;" ," =+

 buff[۰ // ;'? = [Initialize the string to "?" if something goes wrong.
 buff[۱ = [; ۰

// Allow to print just part of the image.
 totalElements; ++
 if (totalElements > maxElements){
// LOG can be printf or similar.
 LOG("%s ... <just displaying the ۱st %d entries from %d!>", s.c_str(),
maxElements, M->rows * M->cols * channels);
 return;
{
 switch (depth) {
 case CV_8U:
 case CV_8S: // ۸-bit UCHAR Mat.
 buff[۰] = 'C; '
// snprintf(buff, sizeof(buff), "%d", data[(row * step) + (col*channels) + ch]);
// ; UNTESTED!
 break;
 case CV_16U:

```

```

 case CV_16S: // ۱۶-bit short Mat.

 buff[۰] = 'S';

 //snprintf(buff, sizeof(buff), "%d", *(short*)(uchar*)&data[(row * step) + ((col*channels) +
 ch) * sizeof(short)]);
 // ; UNTESTED!

 break;

 case CV_32S: // ۳۲-bit int Mat.

 buff[۰] = 'I';

 snprintf(buff, sizeof(buff), "%d", *(int*)(uchar*)&data[(row * step) + ((col*channels) +
 ch) * sizeof(int)]);
 // ; UNTESTED!

 break;

 case CV_32F: // ۳۲-bit float Mat.

 snprintf(buff, sizeof(buff), "%.۳f", *(float*)(uchar*)&data[(row * step) + ((col*channels)
 + ch) * sizeof(float)]);
 break;

 case CV_64F: // ۶۴-bit double Mat.

 buff[۰] = 'D';

 // snprintf(buff, sizeof(buff), "%.۴lf", *(double*)(uchar*)&data[(row * step) +
 ((col*channels) + ch) * sizeof(double)]);
 // ; UNTESTED!

 break;

 default:

 snprintf(buff, sizeof(buff), "UNKNOWN DEPTH OF %d!", depth);
 {
 s += buff;
 const int MAX_ELEMENTS_PER_LINE = ۶۰; // Only print upto ۳۰ numbers
 per LOG() statement, since Android can only handle about ۲۵۰ characters per log line!

 if (element > MAX_ELEMENTS_PER_LINE){
 // LOG can be printf or similar.

 LOG(s.c_str());
 s, "" =

 element = ;

```

```

{
 element; ++
 {
 if (channels > ۱(
 s; " [" =+
 {
 // LOG can be printf or similar.
 LOG(s.c_str());
 {
 { else }
 LOG("[%s] = <null>!", s.c_str());
 }
 {
 }

 //Print the x & y coords of the given point. If 'label' is supplied, prints that first, and also
 prints a newline character on the end.
 void printPoint(const CvPoint pt, const char *label)
 {
 if (label)
 cout << label; " : " >>

 cout << "(" << pt.x << ", " << pt.y << ");"
 if (label)
 cout << endl;
 }

 //Print the start & end coords of the given line. If 'label' is supplied, prints that first, and also
 prints a newline character on the end.
 void printLine(const CvPoint ptA, const CvPoint ptB, const char *label)
 {
 if (label)
 cout << label; " : " >>

 cout << "(" << ptA.x << ", " << ptA.y << ")-(" << ptB.x << ", " << ptB.y << ");"
 if (label)
 cout << endl;
 }

 //Just for debugging float images & matrices.
 void printDataRange(const CvArr *src, const char *msg)
 {
 if (((IplImage*)src)->nChannels == ۲ // } (۲-ch (Complex) input

```

```

double min_val[2] = {0};
double max_val[2] = {0};
IplImage * imgComplexSrcRe = cvCreateImage(cvGetSize(((IplImage*)src)),
IPL_DEPTH_32F, 1);
IplImage * imgComplexSrcIm = cvCreateImage(cvGetSize(((IplImage*)src)),
IPL_DEPTH_32F, 1);

cvSplit(src, imgComplexSrcRe, imgComplexSrcIm, NULL, NULL);
cvMinMaxLoc(imgComplexSrcRe, &min_val[0], &max_val[0], NULL, NULL,
NULL);
cvMinMaxLoc(imgComplexSrcIm, &min_val[1], &max_val[1], NULL, NULL,
NULL);
LOG("\t` ` %s Range: Complex MIN = %lf %lfj, MAX = %lf %lfj", msg, min_val[0],
min_val[1], max_val[0], max_val[1]);

cvReleaseImage(&imgComplexSrcRe);
cvReleaseImage(&imgComplexSrcIm);
{
else if (((IplImage*)src)->nChannels == 1 // } (1-ch (Real) input
double min_val[1] = {0};
double max_val[1] = {0};

cvMinMaxLoc(src, &min_val[0], &max_val[0], NULL, NULL, NULL);

LOG("\t` ` %s Range: Real MIN = %lf, MAX = %lf", msg, min_val[0], max_val[0]);
{
else}

LOG("\t` ` %s Range: UNKNOWN because nChannels == %d != 1 or 2", msg,
((IplImage*)src)->nChannels);
{
{
-----//

//Graphing functions

```

```

-----//
const CvScalar BLACK = CV_RGB;(۰,۰,۰)
const CvScalar WHITE = CV_RGB;(۲۵۵,۲۵۵,۲۵۵)
const CvScalar GREY = CV_RGB;(۱۵۰,۱۵۰,۱۵۰)

int countGraph = ۰; // Used by 'getGraphColor'()
CvScalar customGraphColor;
int usingCustomGraphColor = ;۰

//Get a new color to draw graphs. Will use the latest custom color, or change between blue,
green, red, dark-blue, dark-green and dark-red until a new image is created.
CvScalar getGraphColor(void)
{
 if (usingCustomGraphColor) {
 usingCustomGraphColor = ;۰
 return customGraphColor;
 }

 countGraph++;
 switch (countGraph){
 case ۱: return CV_RGB(۶۰,۶۰,۲۵۵); // light-blue
 case ۲: return CV_RGB(۶۰,۲۵۵,۶۰); // light-green
 case ۳: return CV_RGB(۲۵۵,۶۰,۴۰); // light-red
 case ۴: return CV_RGB(۰,۲۱۰,۲۱۰); // blue-green
 case ۵: return CV_RGB(۱۸۰,۲۱۰,۰); // red-green
 case ۶: return CV_RGB(۲۱۰,۰,۱۸۰); // red-blue
 case ۷: return CV_RGB(۰,۰,۱۸۵); // dark-blue
 case ۸: return CV_RGB(۰,۱۸۵,۰); // dark-green
 case ۹: return CV_RGB(۱۸۵,۰,۰); // dark-red
 default:
 countGraph = ۰; // start rotating through colors again.
 }
}

```

```

 return CV_RGB(۲۰۰,۲۰۰,۲۰۰); // grey
{
{
//Call 'setGraphColor()' to reset the colors that will be used for graphs.
void setGraphColor(int index)
}

 countGraph = index;
 usingCustomGraphColor = ۰; // dont use a custom color.
{

//Specify the exact color that the next graph should be drawn as.
void setCustomGraphColor(int R, int B, int G)
}
 customGraphColor = CV_RGB(R, G, B);
 usingCustomGraphColor = ۱; // show that it will be used.
{

//Draw the graph of an array of floats into imageDst or a new image, between minV &
maxV if given.

//Remember to free the newly created image if imageDst is not given.
IplImage* drawFloatGraph(const float *arraySrc, int nArrayLength, IplImage *imageDst,
float minV, float maxV, int width, int height, char *graphLabel, bool showScale)
}
 int w = width;
 int h = height;
 int b = ۱۰; // border around graph within the image

 if (w <= ۲۰(
 w = nArrayLength + b*۲; // width of the image

 if (h <= ۲۰(
 h = ;۲۲۰

 int s = h - b*۲;// size of graph height

 float xscale = ;۱,۰

 if (nArrayLength > ۱(
 xscale = (w - b*۲) / (float)(nArrayLength-۱); // horizontal scale

```

```

IplImage *imageGraph; // output image

// Get the desired image to draw into.
if (!imageDst){
// Create an RGB image for graphing the data
imageGraph = cvCreateImage(cvSize(w,h), ۸, ۳);

// Clear the image
cvSet(imageGraph, WHITE);
{
else}

// Draw onto the given image.
imageGraph = imageDst;
{
if (!imageGraph){
cerr << "ERROR in drawFloatGraph(): Couldn't create image of " << w << " x " << h
<< endl;
exit;(۱)
{
CvScalar colorGraph = getGraphColor(); // use a different color each time.

// If the user didnt supply min & max values, find them from the data, so we can draw it at
full scale.

if (fabs(minV) < ۰,۰۰۰۰۰۰۱f && fabs(maxV) < ۰,۰۰۰۰۰۰۱f){
for (int i=۰; i<nArrayLength; i++){
float v = (float)arraySrc[i];
if (v < minV)
minV = v;
if (v > maxV)
maxV = v;
{
{
float diffV = maxV - minV;
if (diffV == ۰(

```



```

diffV = ۰,۰۰۰۰۰۰۰۱f; // Stop a divide-by-zero error

float fscale = (float)s / diffV;

// Draw the horizontal & vertical axis

int y = cvRound(minV*fscale);

cvLine(imageGraph, cvPoint(b,h-(b-y)), cvPoint(w-b, h-(b-y)), BLACK);

cvLine(imageGraph, cvPoint(b,h-(b)), cvPoint(b, h-(b+s)), BLACK);

// Write the scale of the y axis

CvFont font;
cvInitFont(&font,CV_FONT_HERSHEY_PLAIN,۰,۵۵,۰,۷, ۰,۱,CV_AA); // For
OpenCV ۱,۱
if (showScale){
// cvInitFont(&font,CV_FONT_HERSHEY_PLAIN,۰,۵۰,۰,۶, ۰,۱, CV_AA); // For
OpenCV ۲,۰

CvScalar clr = GREY;
char text[۱۶]

snprintf(text, sizeof(text)-۱,"%",\f", maxV);

cvPutText(imageGraph, text, cvPoint(۱, b+۴), &font, clr);

// Write the scale of the x axis

snprintf(text, sizeof(text)-۱, "%d", (nArrayLength-۱((;

cvPutText(imageGraph, text, cvPoint(w-b+۵-۴*strlen(text), (h/۲+(۱۰), &font, clr);
{

// Draw the values

CvPoint ptPrev = cvPoint(b,h-(b-y)); // Start the lines at the ۱st point.

for (int i=۰; i<nArrayLength; i++){

int y = cvRound((arraySrc[i] - minV) * fscale); // Get the values at a bigger scale
int x = cvRound(i * xscale);
CvPoint ptNew = cvPoint(b+x, h-(b+y));
cvLine(imageGraph, ptPrev, ptNew, colorGraph, ۱, CV_AA); // Draw a line from the
previous point to the new point
ptPrev = ptNew;

```

```

{

// Write the graph label, if desired
if (graphLabel != NULL && strlen(graphLabel) > ۰){
// cvInitFont(&font,CV_FONT_HERSHEY_PLAIN, ۰,۵,۰,۷,۰,۱,CV_AA);
cvPutText(imageGraph, graphLabel, cvPoint(۳۰, ۱۰), &font, CV_RGB(۰,۰,۰) // black text
black text
{

return imageGraph;
}

//Draw the graph of an array of ints into imageDst or a new image, between minV & maxV
if given.

//Remember to free the newly created image if imageDst is not given.
IplImage* drawIntGraph(const int *arraySrc, int nArrayLength, IplImage *imageDst, int
minV, int maxV, int width, int height, char *graphLabel, bool showScale)
}

int w = width;
int h = height;
int b = ۱۰; // border around graph within the image
if (w <= ۲۰(
w = nArrayLength + b*۲; // width of the image
if (h <= ۲۰(
h = ;۲۲۰

int s = h - b*۲;// size of graph height
float xscale = ;۱,۰
if (nArrayLength > ۱(
xscale = (w - b*۲) / (float)(nArrayLength-۱); // horizontal scale
IplImage *imageGraph; // output image

// Get the desired image to draw into.

```

```

 if (!imageDst){

// Create an RGB image for graphing the data
 imageGraph = cvCreateImage(cvSize(w,h), ۸, ۳);

// Clear the image
 cvSet(imageGraph, WHITE);
 {
 else}

// Draw onto the given image.
 imageGraph = imageDst;
 {
 if (!imageGraph){
 cerr << "ERROR in drawIntGraph(): Couldn't create image of " << w << " x " << h << endl;
 exit;(۱)
 }

 CvScalar colorGraph = getGraphColor(); // use a different color each time.

// If the user didnt supply min & mav values, find them from the data, so we can draw it at
full scale.

 if (minV == ۰ && maxV == ۰){
 for (int i=۰; i<nArrayLength; i++){
 int v = arraySrc[i];
 if (v < minV)
 minV = v;
 if (v > maxV)
 maxV = v;
 }
 }

 int diffV = maxV - minV;
 if (diffV == ۰(
 diffV = ۱; // Stop a divide-by-zero error

 float fscale = (float)s / (float)diffV;

```

```
// Draw the horizontal & vertical axis

int y = cvRound(minV*fscale);

cvLine(imageGraph, cvPoint(b,h-(b-y)), cvPoint(w-b, h-(b-y)), BLACK);

cvLine(imageGraph, cvPoint(b,h-(b)), cvPoint(b, h-(b+s)), BLACK);

// Write the scale of the y axis

CvFont font;
cvInitFont(&font,CV_FONT_HERSHEY_PLAIN, 0.5, 0.5, 0, CV_AA); // For
OpenCV 1.1
if (showScale){
// cvInitFont(&font,CV_FONT_HERSHEY_PLAIN, 0.5, 0.5, 0, CV_AA); // For
OpenCV 2.0

CvScalar clr = GREY;
char text[10]

snprintf(text, sizeof(text)-1, "%f", maxV);

cvPutText(imageGraph, text, cvPoint(1, b+4), &font, clr);

// Write the scale of the x axis

snprintf(text, sizeof(text)-1, "%d", (nArrayLength-1) (;

cvPutText(imageGraph, text, cvPoint(w-b+5-4*strlen(text), (h/2+(1)), &font, clr);

{

// Draw the values

CvPoint ptPrev = cvPoint(b,h-(b-y)); // Start the lines at the 1st point.

for (int i=0; i<nArrayLength; i++){

int y = cvRound((arraySrc[i] - minV) * fscale); // Get the values at a bigger scale
int x = cvRound(i * xscale);

CvPoint ptNew = cvPoint(b+x, h-(b+y));

cvLine(imageGraph, ptPrev, ptNew, colorGraph, 1, CV_AA); // Draw a line from the
previous point to the new point
ptPrev = ptNew;

{
```

```
// Write the graph label, if desired
if (graphLabel != NULL && strlen(graphLabel) > ۰){
// cvInitFont(&font,CV_FONT_HERSHEY_PLAIN, ۰٫۵,۰٫۷,۰٫۱,CV_AA);
 cvPutText(imageGraph, graphLabel, cvPoint(۳۰,۱۰), &font, CV_RGB(۰٫۰٫۰ // ;((
black text
{

 return imageGraph;
}

//Draw the graph of an array of uchars into imageDst or a new image, between minV &
maxV if given..

//Remember to free the newly created image if imageDst is not given.
IplImage* drawUCharGraph(const uchar *arraySrc, int nArrayLength, IplImage *imageDst,
int minV, int maxV, int width, int height, char *graphLabel, bool showScale)
{

 int w = width;
 int h = height;
 int b = ۱۰; // border around graph within the image
 if (w <= ۲۰(
 w = nArrayLength + b*۲; // width of the image
 if (h <= ۲۰(
 h = ;۲۲۰

 int s = h - b*۲;// size of graph height
 float xscale = ;۱٫۰
 if (nArrayLength > ۱(
 xscale = (w - b*۲) / (float)(nArrayLength-۱); // horizontal scale
 IplImage *imageGraph; // output image

// Get the desired image to draw into.
 if (!imageDst)}

// Create an RGB image for graphing the data
```

```

imageGraph = cvCreateImage(cvSize(w,h), ۸, ۳);

// Clear the image
cvSet(imageGraph, WHITE);
{
 else}

// Draw onto the given image.
imageGraph = imageDst;
{
 if (!imageGraph){
 cerr << "ERROR in drawUCharGraph(): Couldn't create image of " << w << " x " << h
 << endl;
 exit;(۱)
 }

 CvScalar colorGraph = getGraphColor(); // use a different color each time.

// If the user didnt supply min & max values, find them from the data, so we can draw it at
full scale.

 if (minV == ۰ && maxV == ۰){
 for (int i=۰; i<nArrayLength; i++){
 int v = arraySrc[i];
 if (v < minV)
 minV = v;
 if (v > maxV)
 maxV = v;
 }
 }

 int diffV = maxV - minV;
 if (diffV == ۰){
 diffV = ۱; // Stop a divide-by-zero error

 float fscale = (float)s / (float)diffV;

// Draw the horizontal & vertical axis

```

```

int y = cvRound(minV*fyscale);

cvLine(imageGraph, cvPoint(b,h-(b-y*)), cvPoint(w-b, h-(b-y*)), BLACK);

cvLine(imageGraph, cvPoint(b,h-(b)), cvPoint(b, h-(b+s)), BLACK);

// Write the scale of the y axis

CvFont font;
cvInitFont(&font,CV_FONT_HERSHEY_PLAIN,۰,۵۵,۰,۷, ۰,۱,CV_AA); // For
OpenCV ۱,۱
if (showScale){
// cvInitFont(&font,CV_FONT_HERSHEY_PLAIN,۰,۵۵,۰,۶, ۰,۱, CV_AA); // For
OpenCV ۲,۰

CvScalar clr = GREY;
char text[۱۶]

snprintf(text, sizeof(text)-۱,"%f", maxV);

cvPutText(imageGraph, text, cvPoint(۱, b+۴), &font, clr);

// Write the scale of the x axis

snprintf(text, sizeof(text)-۱, "%d", (nArrayLength-۱) (;
cvPutText(imageGraph, text, cvPoint(w-b+۵-۴*strlen(text), (h/۲+(۱*)), &font, clr);
{

// Draw the values

CvPoint ptPrev = cvPoint(b,h-(b-y*)); // Start the lines at the ۱st point.

for (int i=۰; i<nArrayLength; i++){
int y = cvRound((arraySrc[i] - minV) * fyscale); // Get the values at a bigger scale
int x = cvRound(i * xscale);
CvPoint ptNew = cvPoint(b+x, h-(b+y));
cvLine(imageGraph, ptPrev, ptNew, colorGraph, ۱, CV_AA); // Draw a line from the
previous point to the new point
ptPrev = ptNew;
}

// Write the graph label, if desired

```

```

 if (graphLabel != NULL && strlen(graphLabel) > ۰){
// cvInitFont(&font,CV_FONT_HERSHEY_PLAIN, ۰,۵,۰,۷,۰,۱,CV_AA);
 cvPutText(imageGraph, graphLabel, cvPoint(۳۰, ۱۰), &font, CV_RGB(۰,۰,۰ // ;((
black text
{

 return imageGraph;
}

//Display a graph of the given float array.

//If background is provided, it will be drawn into, for combining multiple graphs using
drawFloatGraph.()

//Set delay_ms to ۰ if you want to wait forever until a keypress, or set it to ۱ if you want it
to delay just ۱millisecond.
void showFloatGraph(const char *name, const float *arraySrc, int nArrayLength, int
delay_ms, IplImage *background)
}
#ifdef USE_HIGHGUI
// Draw the graph

IplImage *imageGraph = drawFloatGraph(arraySrc, nArrayLength, background);

// Display the graph into a window

cvNamedWindow(name);
cvShowImage(name, imageGraph);

cvWaitKey(۱۰); // Note that cvWaitKey() is required for the OpenCV window to
show!

cvWaitKey(delay_ms); // Wait longer to make sure the user has seen the graph

cvReleaseImage(&imageGraph);
#endif
{

//Display a graph of the given int array.

//If background is provided, it will be drawn into, for combining multiple graphs using

```



```

drawIntGraph.()

//Set delay_ms to 0 if you want to wait forever until a keypress, or set it to 1 if you want it
to delay just 1millisecond.
void showIntGraph(const char *name, const int *arraySrc, int nArrayLength, int delay_ms,
IplImage *background)
{
#ifdef USE_HIGHGUI
// Draw the graph

IplImage *imageGraph = drawIntGraph(arraySrc, nArrayLength, background);

// Display the graph into a window

cvNamedWindow(name);
cvShowImage(name, imageGraph);

cvWaitKey(10); // Note that cvWaitKey() is required for the OpenCV window to
show!

cvWaitKey(delay_ms); // Wait longer to make sure the user has seen the graph

cvReleaseImage(&imageGraph);
#endif
{

//Display a graph of the given unsigned char array.

//If background is provided, it will be drawn into, for combining multiple graphs using
drawUCharGraph.()

//Set delay_ms to 0 if you want to wait forever until a keypress, or set it to 1 if you want it
to delay just 1millisecond.
void showUCharGraph(const char *name, const uchar *arraySrc, int nArrayLength, int
delay_ms, IplImage *background)
{
#ifdef USE_HIGHGUI
// Draw the graph

IplImage *imageGraph = drawUCharGraph(arraySrc, nArrayLength, background);

// Display the graph into a window

cvNamedWindow(name);

```

```

cvShowImage(name, imageGraph);

cvWaitKey(۱۰); // Note that cvWaitKey() is required for the OpenCV window to
show!

cvWaitKey(delay_ms); // Wait longer to make sure the user has seen the graph

cvReleaseImage(&imageGraph);
#endif
{

//Simple helper function to easily view an image, with an optional pause.
void showImage(const IplImage *img, int delay_ms, char *name)
}

*/
#ifdef USE_HIGHGUI
if (!name)
 name = "Image;";
cvNamedWindow(name, CV_WINDOW_AUTOSIZE);
cvShowImage(name, img);
cvWaitKey(delay_ms);
#endif
/*
{

-----//

//Color conversion functions
-----//

//Return a new image that is always greyscale, whether the input image was RGB or
Greyscale.

//Remember to free the returned image using cvReleaseImage() when finished.
IplImage* convertImageToGreyscale(const IplImage *imageSrc)
}

```

```

IplImage *imageGrey;
// Either convert the image to greyscale, or make a copy of the existing greyscale image.
// This is to make sure that the user can always call cvReleaseImage() on the output,
whether it was greyscale or not.

if (imageSrc->nChannels == ۳){
 imageGrey = cvCreateImage(cvGetSize(imageSrc), IPL_DEPTH_۸U, ۱);
 cvCvtColor(imageSrc, imageGrey, CV_BGR۲GRAY);
}
else{
 imageGrey = cvCloneImage(imageSrc);
}

return imageGrey;
}

//Do the color conversion of a single pixel, from RGB to HSV using Hue values between ۰
to ۲۵۵, whereas OpenCV only allows Hues up to ۱۸۰ instead of ۲۵۵
//ref: "http://cs.haifa.ac.il/hagit/courses/ist/Lectures/Demos/ColorApplet۲/t_convert.html"
inline void convertPixelRGBtoHSV_۲۵۶(int bR, int bG, int bB, int &bH, int &bS, int &bV)
{
 float fR, fG, fB;
 float fH, fS, fV;
 const float FLOAT_TO_BYTE = ۲۵۵,۰f;
 const float BYTE_TO_FLOAT = ۱,۰f / FLOAT_TO_BYTE;

 // Convert from ۸-bit integers to floats
 fR = bR * BYTE_TO_FLOAT;
 fG = bG * BYTE_TO_FLOAT;
 fB = bB * BYTE_TO_FLOAT;

 // Convert from RGB to HSV, using float ranges ۰,۰ to ۱,۰
 float fDelta;
 float fMin, fMax;
 int iMax;

```

```
// Get the min & max, but use integer comparisons for slight speedup

if (bB < bG){
 if (bB < bR){
 fMin = fB;
 if (bR > bG){
 iMax = bR;
 fMax = fR;
 }
 else{
 iMax = bG;
 fMax = fG;
 }
 }
 else{
 fMin = fR;
 fMax = fG;
 iMax = bG;
 }
}
else{
 if (bG < bR) {
 fMin = fG;
 if (bB > bR){
 fMax = fB;
 iMax = bB;
 }
 {
 Else{
 fMax = fR;
 iMax = bR;
 }
 }
 }
 else {
 fMin = fR;
 fMax = fB;
 iMax = bB;
 }
}
{
 fDelta = fMax - fMin;
```

```

fV = fMax; // Value (Brightness).

if (iMax != ۰) { // Make sure its not pure black.

 fS = fDelta / fMax; // Saturation.

 float ANGLE_TO_UNIT = ۱,۰f / (۶,۰f * fDelta); // Make the Hues between ۰,۰ to
 ۱,۰ instead of ۶,۰

 if (iMax == bR) { // between yellow & magenta.

 fH = (fG - fB) * ANGLE_TO_UNIT;
 {
 else if (iMax == bG) { // between cyan & yellow.

 fH = (۲,۰f/۶,۰f) + (fB - fR) * ANGLE_TO_UNIT;
 {
 else { // between magenta & cyan.

 fH = (۴,۰f/۶,۰f) + (fR - fG) * ANGLE_TO_UNIT;
 {
// Wrap outlier Hues around the circle.

 if (fH < ۰,۰f)

 fH += ۱,۰f;

 if (fH >= ۱,۰f)

 fH -= ۱,۰f;
 { else }
// color is pure Black.

 fS = ۰,۰;

 fH = ۰; // undefined hue
 {

// Convert from floats to ۸-bit integers

 bH = (int)(۰,۵f + fH * ۲۵۵,۰f);

 bS = (int)(۰,۵f + fS * ۲۵۵,۰f);

 bV = (int)(۰,۵f + fV * ۲۵۵,۰f);

// Clip the values to make sure it fits within the ۸bits

// if (bH > ۲۵۵|| bH < ۰|| bS > ۲۵۵|| bS < ۰|| bV > ۲۵۵|| bV < ۰(}

```

```
// cout << "Warning: HSV pixel(" << x << "," << y << ") is being clipped. " << bH >>
", " << bS << "," << bV << endl;
{//
if (bH > ۲۵۵(
 bH = ;۲۵۵
if (bH < ۰(
 bH = ;۰
if (bS > ۲۵۵(
 bS = ;۲۵۵
if (bS < ۰(
 bS = ;۰
if (bV > ۲۵۵(
 bV = ;۲۵۵
if (bV < ۰(
 bV = ;۰
{

//Create a HSV image from the RGB image using the full ۸-bits, since OpenCV only allows
Hues up to ۱۸۰ instead of ۲۵۵

//ref: "http://cs.haifa.ac.il/hagit/courses/ist/Lectures/Demos/ColorApplet۲/t_convert.html"

//Remember to free the generated HSV image.
IplImage* convertImageRGBtoHSV(const IplImage *imageRGB)
{
 // Create a blank HSV image

 IplImage *imageHSV = cvCreateImage(cvGetSize(imageRGB), ۸, ۳);
 if (!imageHSV || imageRGB->depth != ۸ || imageRGB->nChannels != ۳){
 LOG("ERROR in convertImageRGBtoHSV()! Bad input image.\n");
 exit;(۱)
 }

 int h = imageRGB->height; // Pixel height
 int w = imageRGB->width; // Pixel width
```

```

 int rowSizeRGB = imageRGB->widthStep; // Size of row in bytes, including extra
padding
 char *imRGB = imageRGB->imageData; // Pointer to the start of the image pixels.

 int rowSizeHSV = imageHSV->widthStep; // Size of row in bytes, including extra
padding
 char *imHSV = imageHSV->imageData; // Pointer to the start of the image pixels.

 for (int y=۰; y<h; y++) {
 for (int x=۰; x<w; x++) {

// Get the RGB pixel components. NOTE that OpenCV stores RGB pixels in B,G,R
order.

 uchar *pRGB = (uchar*)(imRGB + y*rowSizeRGB + x*۳);
 int bB = *(uchar*)(pRGB+۰); // Blue component
 int bG = *(uchar*)(pRGB+۱); // Green component
 int bR = *(uchar*)(pRGB+۲); // Red component

// Do the conversion.

 int bH, bS, bV;
 convertPixelRGBtoHSV_۲۵۶(bR,bG,bB, bH,bS,bV);

// Set the HSV pixel components

 uchar *pHSV = (uchar*)(imHSV + y*rowSizeHSV + x*۳);

 * (pHSV+۰) = bH; // H component
 * (pHSV+۱) = bS; // S component
 * (pHSV+۲) = bV; // V component
 }
 }

 return imageHSV;
}

//Do the color conversion of a single pixel, from HSV to RGB using Hue values between ۰
to ۲۵۵, whereas OpenCV only allows Hues up to ۱۸۰ instead of ۲۵۵
//ref: "http://cs.haifa.ac.il/hagit/courses/ist/Lectures/Demos/ColorApplet۲/t_convert.html"

```

```

inline void convertPixelHSVtoRGB_۲۵۶(int bH, int bS, int bV, int &bR, int &bG, int &bB)
{
 float fH, fS, fV;
 float fR, fG, fB;
 const float FLOAT_TO_BYTE = ۲۵۵,۰f;
 const float BYTE_TO_FLOAT = ۱,۰f / FLOAT_TO_BYTE;

 // Convert from ۸-bit integers to floats
 fH = (float)bH * BYTE_TO_FLOAT;
 fS = (float)bS * BYTE_TO_FLOAT;
 fV = (float)bV * BYTE_TO_FLOAT;

 // Convert from HSV to RGB, using float ranges ۰,۰ to ۱,۰
 int iI;
 float fI, fF, p, q, t;

 if(bS == ۰){
 // achromatic (grey)
 fR = fG = fB = fV;
 }
 else{
 // if (bH < ۰ || bH >= ۲۵۵ || bS < ۰ || bS > ۲۵۵ || bV < ۰ || bV > ۲۵۵){
 // cout << "ERROR: HSVi pixel(" << x << "," << y << ") is being clipped. " << bH
 << "," << bS << "," << bV << endl;
 // cout << "ERROR: HSVf pixel(" << x << "," << y << ") is being clipped. " << fH
 << "," << fS << "," << fV << endl;
 }

 // If Hue == ۱,۰, then wrap it around the circle to ۰,۰
 if (fH >= ۱,۰f)
 fH = ۰,۰f;

 fH *= ۶,۰; // sector ۰ to ۵

 fI = floor(fH); // integer part of h (۰,۱,۲,۳,۴,۵ or ۶(

```



```

iI = (int) fH" " " " // ;
fF = fH - fI; // factorial part of h (* to \ (

p = fV * (\, * f - fS);
q = fV * (\, * f - fS * fF);
t = fV * (\, * f - fS * (\, * f - fF));

switch(iI) {
 case :۱
 fR = fV;
 fG = t;
 fB = p;
 break;
 case :۲
 fR = q;
 fG = fV;
 fB = p;
 break;
 case :۳
 fR = p;
 fG = q;
 fB = fV;
 break;
 case :۴
 fR = t;
 fG = p;
 fB = fV;
 break;
 default: // case ۵ (or ۶ (
 fR = fV;
 fG = p;

```

```
 fB = q;
 break;
 {
 {

// Convert from floats to 8-bit integers

 bR = (int)(fR * FLOAT_TO_BYTE);
 bG = (int)(fG * FLOAT_TO_BYTE);
 bB = (int)(fB * FLOAT_TO_BYTE);

// Clip the values to make sure it fits within the 8bits
// if (bR > ۲۵۵|| bR < ۰|| bG > ۲۵۵|| bG < ۰|| bB > ۲۵۵|| bB < ۰){

 // cout << "Warning: RGB pixel(" << x << "," << y << ") is being clipped. " << bR <<
 "<< bG << "," << bB << endl;
 {//

 if (bR > ۲۵۵(
 bR = ;۲۵۵
 if (bR < ۰(
 bR = ;۰
 if (bG > ۲۵۵(
 bG = ;۲۵۵
 if (bG < ۰(
 bG = ;۰
 if (bB > ۲۵۵(
 bB = ;۲۵۵
 if (bB < ۰(
 bB = ;۰
 {

//Create an RGB image from the HSV image using the full 8-bits, since OpenCV only
allows Hues up to ۱۸۰ instead of ۲۵۵
```

```
//ref: "http://cs.haifa.ac.il/hagit/courses/ist/Lectures/Demos/ColorApplet۲/t_convert.html"

//Remember to free the generated RGB image.
IplImage* convertImageHSVtoRGB(const IplImage *imageHSV)
{
 // Create a blank RGB image

 IplImage *imageRGB = cvCreateImage(cvGetSize(imageHSV), ۸, ۳);

 if (!imageRGB || imageHSV->depth != ۸ || imageHSV->nChannels != ۳){
 LOG("ERROR in convertImageHSVtoRGB()! Bad input image.\n");
 exit;(۱)
 }

 int h = imageHSV->height; // Pixel height
 int w = imageHSV->width; // Pixel width
 int rowSizeHSV = imageHSV->widthStep; // Size of row in bytes, including extra
padding
 char *imHSV = imageHSV->imageData; // Pointer to the start of the image pixels.

 int rowSizeRGB = imageRGB->widthStep; // Size of row in bytes, including extra
padding
 char *imRGB = imageRGB->imageData; // Pointer to the start of the image pixels.

 for (int y=۰; y<h; y++){
 for (int x=۰; x<w; x++){

 // Get the HSV pixel components

 uchar *pHSV = (uchar*)(imHSV + y*rowSizeHSV + x*۳);

 int bH = *(uchar*)(pHSV+۰); // H component
 int bS = *(uchar*)(pHSV+۱); // S component
 int bV = *(uchar*)(pHSV+۲); // V component

 // Do the conversion.

 int bB, bG, bR;
 convertPixelHSVtoRGB_۲۵۶(bH,bS,bV, bR,bG,bB);

 // Set the RGB pixel components. NOTE that OpenCV stores RGB pixels in B,G,R
order.
```

```

uchar *pRGB = (uchar*)(imRGB + y*rowSizeRGB + x*۳);

* (pRGB+۰) = bB; // B component
* (pRGB+۱) = bG; // G component
* (pRGB+۲) = bR; // R component
{
{
return imageRGB;
{

//Do the color conversion of a single pixel, from RGB to HSV using Hue values between ۰
to ۱۷۹ just like OpenCV does, instead of Hues upto ۲۵۵

void convertPixelRGBtoHSV_۱۸۰(int bR, int bG, int bB, int &bH, int &bS, int &bV)
{

// Do the conversion with Hues between ۰ to ۲۵۵
convertPixelRGBtoHSV_۲۵۶(bR,bG,bB, bH,bS,bV);

// Convert to a Hue of ۱۸۰ range from ۲۵۶ range.
bH = ۱۷۹* bH / ۲۵۵
{

//Do the color conversion of a single pixel, from HSV to RGB using Hue values between ۰
to ۱۷۹ just like OpenCV does, instead of Hues upto ۲۵۵

void convertPixelHSVtoRGB_۱۸۰(int bH, int bS, int bV, int &bR, int &bG, int &bB)
{

// Convert from a Hue of ۱۸۰ range to ۲۵۶ range.
bH = ۲۵۵* bH / ۱۷۹

// Do the conversion with Hues between ۰ to ۲۵۵
convertPixelHSVtoRGB_۲۵۶(bH,bS,bV, bR,bG,bB);
{

```

```
//Create an RGB image from the YIQ image using an approximation of NTSC
conversion(ref: "YIQ" Wikipedia page).

//Remember to free the generated RGB image.
IplImage* convertImageYIQtoRGB(const IplImage *imageYIQ)
{
 float fY, fI, fQ;
 float fR, fG, fB;
 const float FLOAT_TO_BYTE = ۲۵۵,۰f;

 // const float BYTE_TO_FLOAT = ۱,۰f / FLOAT_TO_BYTE;

 const float MIN_I = -۰,۵۹۵۷f;
 const float MIN_Q = -۰,۵۲۲۶f;

 const float Y_TO_FLOAT = ۱,۰f / ۲۵۵,۰f;
 const float I_TO_FLOAT = -۲,۰f * MIN_I / ۲۵۵,۰f;
 const float Q_TO_FLOAT = -۲,۰f * MIN_Q / ۲۵۵,۰f;

 // Create a blank RGB image
 IplImage *imageRGB = cvCreateImage(cvGetSize(imageYIQ), ۸, ۳);
 if (!imageRGB || imageYIQ->depth != ۸ || imageYIQ->nChannels != ۳){
 LOG("ERROR in convertImageYIQtoRGB()! Bad input image.\n");
 exit;(۱)
 }

 int h = imageYIQ->height; // Pixel height
 int w = imageYIQ->width; // Pixel width
 int rowSizeYIQ = imageYIQ->widthStep; // Size of row in bytes, including extra
padding
 char *imYIQ = imageYIQ->imageData; // Pointer to the start of the image pixels.
 int rowSizeRGB = imageRGB->widthStep; // Size of row in bytes, including extra
padding
 char *imRGB = imageRGB->imageData; // Pointer to the start of the image pixels.

 for (int y=۰; y<h; y++){
 for (int x=۰; x<w; x++){

 // Get the YIQ pixel components
```

```

uchar *pYIQ = (uchar*)(imYIQ + y*rowSizeYIQ + x*۳);

int bY = *(uchar*)(pYIQ+۰); // Y component
int bI = *(uchar*)(pYIQ+۱); // I component
int bQ = *(uchar*)(pYIQ+۲); // Q component

// Convert from ۸-bit integers to floats
fY = (float)bY * Y_TO_FLOAT;
fI = (float)bI * I_TO_FLOAT + MIN_I;
fQ = (float)bQ * Q_TO_FLOAT + MIN_Q;
// Convert from YIQ to RGB
// where R,G,B are ۱-۰, Y is ۱-۰, I is - ۰,۵۹۵۷to +۰,۵۹۵۷, Q is - ۰,۵۲۲۶to +۰,۵۲۲۶
fR = fY + ۰,۹۵۶۳f * fI + ۰,۶۲۱۰f * fQ;
fG = fY - ۰,۲۷۲۱f * fI - ۰,۶۴۷۴f * fQ;
fB = fY - ۱,۱۰۷۰f * fI + ۱,۷۰۴۶f * fQ;

// Convert from floats to ۸-bit integers
int bR = (int)(fR * FLOAT_TO_BYTE);
int bG = (int)(fG * FLOAT_TO_BYTE);
int bB = (int)(fB * FLOAT_TO_BYTE);

// Clip the values to make sure it fits within the ۸bits
// if (bR > ۲۵۵|| bR < ۰|| bG > ۲۵۵|| bG < ۰|| bB > ۲۵۵|| bB < ۰){
// cout << "Warning: RGB pixel(" << x << ", " << y << ") is being clipped. " << bR
<< ", " << bG << ", " << bB << endl;
{
 if (bR > ۲۵۵(
 bR = ;۲۵۵
 if (bR < ۰(
 bR = ;۰
 if (bG > ۲۵۵(
 bG = ;۲۵۵
 if (bG < ۰(

```

```

 bG = ;
 if (bB > ۲۵۵(
 bB = ;۲۵۵
 if (bB < ۰(
 bB = ;

// Set the RGB pixel components. NOTE that OpenCV stores RGB pixels in B,G,R
order.

 uchar *pRGB = (uchar*)(imRGB + y*rowSizeRGB + x*۳(;
 * (pRGB+۰) = bB; // B component
 * (pRGB+۱) = bG; // G component
 * (pRGB+۲) = bR; // R component
 {
 {
 return imageRGB;
 {

-----//

۲ //D Point functions
-----//

CvPoint۲D۳۲f addPointF(const CvPoint۲D۳۲f pointA, const CvPoint۲D۳۲f pointB)
}

 CvPoint۲D۳۲f ret;

 ret.x = pointA.x + pointB.x;
 ret.y = pointA.y + pointB.y;
 return ret;
 {

CvPoint۲D۳۲f subtractPointF(const CvPoint۲D۳۲f pointA, const CvPoint۲D۳۲f pointB)

```

```

}

CvPoint_2Df ret;

ret.x = pointA.x - pointB.x;
ret.y = pointA.y - pointB.y;
return ret;
{

CvPoint_2Df scalePointF(const CvPoint_2Df point, float scale)
}

CvPoint_2Df ret;

ret.x = point.x * scale;
ret.y = point.y * scale;
return ret;
{

CvPoint_2Df rotatePointF(const CvPoint_2Df point, float angleDegrees)
}

CvPoint_2Df ret;

float dx = point.x;
float dy = point.y;
float angleRadians = angleDegrees * ((float)CV_PI / 180, f);

float cosA = cos(angleRadians);
float sinA = sin(angleRadians);
ret.x = (dx * cosA - dy * sinA);
ret.y = (dx * sinA + dy * cosA);
return ret;
{

CvPoint_2Df rotatePointAroundPointF(const CvPoint_2Df point, const CvPoint_2Df
origin, float angleDegrees)
}

CvPoint_2Df ret;

float dx = point.x - origin.x;
float dy = point.y - origin.y;
float angleRadians = angleDegrees * ((float)CV_PI / 180, f);

```



```

float cosA = cos(angleRadians);
float sinA = sin(angleRadians);
ret.x = (dx * cosA - dy * sinA) + origin.x;
ret.y = (dx * sinA + dy * cosA) + origin.y;
return ret;
{

CvPoint_2D32f scalePointAroundPointF(const CvPoint_2D32f point, const CvPoint_2D32f
origin, float scale)
}

CvPoint_2D32f ret;

float dx = point.x - origin.x;
float dy = point.y - origin.y;
ret.x = (dx * scale) + origin.x;
ret.y = (dy * scale) + origin.y;
return ret;
{

//Return (p * s), to a maximum value of maxVal
float scaleValueF(float p, float s, float maxVal)
}

float r = p * s;
if (r > maxVal)
}

r = maxVal;
{

return r;
{

//Return (p * s), to a maximum value of maxVal
int scaleValueI(int p, float s, int maxVal)
}

int r = cvRound(((float)p) * s);
if (r > maxVal)
}

r = maxVal;

```

```

{
 return r;
}

//Calculate the distance between the ۲given points.
float findDistanceBetweenPointsF(const CvPoint۲D۳۲f p۱, const CvPoint۲D۳۲f p۲(
}

// Calc the pythagoras distance.
float dx = (p۱.x - p۲.x);
float dy = (p۱.y - p۲.y);
return sqrt((float) (dx * dx + dy * dy));
{
float findDistanceBetweenPointsI(const CvPoint p۱, const CvPoint p۲(
}

// Calc the pythagoras distance.
int dx = (p۱.x - p۲.x);
int dy = (p۱.y - p۲.y);
return sqrt((float) (dx * dx + dy * dy));
{

//Calculate the angle between the ۲given floating-point points (in degrees).
float findAngleBetweenPointsF(const CvPoint۲D۳۲f p۱, const CvPoint۲D۳۲f p۲(
}

// Calculate the angle of in-plane-rotation of the face based on the angle of the ۲eye
positions.
float dx = (float)(p۲.x - p۱.x);
if (dx == ۰,۰f)
 dx = ۰,۰۰۰۰۰۰۰۱f; // Stop a divide-by-zero error.
float radians = atan۲((float) (p۲.y - p۱.y), dx); // angle = inv_tan(dy / dx).

```

```

return (radians * ۱۸۰,۰f / (float)CV_PI); // convert to degrees from radians.
{
//Calculate the angle between the ۲given integer points (in degrees).
float findAngleBetweenPointsI(const CvPoint p۱, const CvPoint p۲(
}

// Calculate the angle of in-plane-rotation of the face based on the angle of the ۲eye
positions.

float dx = (float)(p۲.x - p۱.x);

if (dx == ۰,۰f)

 dx = ۰,۰۰۰۰۰۰۰۱f; // Stop a divide-by-zero error.

float radians = atan۲((float) (p۲.y - p۱.y), dx); // angle = inv_tan(dy / dx).

return (radians * ۱۸۰,۰f / (float)CV_PI); // convert to degrees from radians.
{

//Print the label and then the rect to the console for easy debugging
void printPoint(const CvPoint pt, const char *label)
}

if (label)
 std::cout << label;":" >>

 std::cout << "[Point] at (" << pt.x << "," << pt.y << ")" << std::endl;
{

//Print the label and then the rect to the console for easy debugging
void printPointF(const CvPoint۲D۳۲f pt, const char *label)
}

if (label)
 std::cout << label;":" >>

 std::cout << "[Point۳۲f] at (" << pt.x << "," << pt.y << ")" << std::endl;
{

-----//

```

```

//Rectangle region functions
-----//

//Enlarge or shrink the rectangle size & center by a given scale.
//If w & h are given, will make sure the rectangle stays within the bounds if it is enlarged
too much.

//Note that for images, w should be (width-۱) and h should be (height-۱).
CvRect scaleRect(const CvRect rectIn, float scaleX, float scaleY, int w, int h)
{
 CvRect rect;
 // Scale the image region position & size
 rect.x = cvRound(scaleX * (float)rectIn.x);
 rect.y = cvRound(scaleY * (float)rectIn.y);
 rect.width = cvRound(scaleX * (float)rectIn.width);
 rect.height = cvRound(scaleY * (float)rectIn.height);
 // Make sure it doesn't go outside the image
 if (w > ۰ && h > ۰) {
 if (rect.x + rect.width > w)
 rect.width = w - rect.x;
 if (rect.y + rect.height > h)
 rect.height = h - rect.y;
 }
 return rect;
}

//Enlarge or shrink the rectangle region by a given scale without moving its center, and
possibly add a border around it.

//If w & h are given, will make sure the rectangle stays within the bounds if it is enlarged
too much.

//Note that for images, w should be (width-۱) and h should be (height-۱).
CvRect scaleRectInPlace(const CvRect rectIn, float scaleX, float scaleY, float borderX, float
borderY, int w, int h)

```

```

}

CvRect rect = rectIn; // Make a local copy
// Scale the image region size and add an extra border to the region size
rect.x -= cvRound((scaleX - ۱,۰f) * (float)rect.width * ۰,۵f + borderX);
rect.y -= cvRound((scaleY - ۱,۰f) * (float)rect.height * ۰,۵f + borderY);
rect.width = cvRound(scaleX * (float)rect.width + ۲,۰f * borderX);
rect.height = cvRound(scaleY * (float)rect.height + ۲,۰f * borderY);

// Make sure it doesn't go outside the image
if (w > ۰ && h > ۰) {
 if (rect.x < ۰) {
 rect.width += rect.x; // shrink the width, since 'x' is negative
 rect.x = ۰;
 }
 if (rect.x + rect.width > w)
 rect.width = w - rect.x;
 if (rect.y < ۰) {
 rect.height += rect.y; // shrink the height, since 'y' is negative
 rect.y = ۰;
 }
 if (rect.y + rect.height > h)
 rect.height = h - rect.y;
}

return rect;
}

//Return a new rect that the same as rectA when shifted by (rectB.x, rectB.y).
CvRect offsetRect(const CvRect rectA, const CvRect rectB)
{
 CvRect rectC;
 rectC.x = rectA.x + rectB.x;
 rectC.y = rectA.y + rectB.y;
 rectC.width = rectA.width;
 rectC.height = rectA.height;
 return rectC;
}

```

```
{

//Return a new rect that the same as rectA when shifted by (pt.x, pt.y).
CvRect offsetRectPt(const CvRect rectA, const CvPoint pt)
{
 CvRect rectC;
 rectC.x = rectA.x + pt.x;
 rectC.y = rectA.y + pt.y;
 rectC.width = rectA.width;
 rectC.height = rectA.height;
 return rectC;
}

//Draw a rectangle around the given object.
void drawRect(IplImage *img, const CvRect rect, const CvScalar color)
{
 CvPoint p1, p2;
 p1.x = rect.x;
 p1.y = rect.y;
 p2.x = min(rect.x + rect.width-1, img->width-1); // Make sure the end-point is within the
image
 p2.y = min(rect.y + rect.height-1, img->height-1(" " " // ;
 cvRectangle(img, p1, p2, color, 1);
}

//Draw a filled rectangle around the given object.
void drawRectFilled(IplImage *img, const CvRect rect, const CvScalar color)
{
 CvPoint p1, p2;
 p1.x = rect.x;
 p1.y = rect.y;
 p2.x = min(rect.x + rect.width-1, img->width-1); // Make sure the end-point is within the
image
 p2.y = min(rect.y + rect.height-1, img->height-1(" " " // ;
```

```

cvRectangle(img, p۱, p۲, color, CV_FILLED);
{

//Draw a crossbar at the given position.
void drawCross(IplImage *img, const CvPoint pt, int radius, const CvScalar color)
{
 CvPoint p۱, p۲, p۳, p۴;
 // Make sure the points are within the image
 p ۱= cvPoint(max(pt.x-radius,۰), pt.y);
 p ۲= cvPoint(min(pt.x+radius,img->width-۱), pt.y);
 p ۳= cvPoint(pt.x, max(pt.y-radius,۰) (;
 p ۴= cvPoint(pt.x, min(pt.y+radius,img->height-۱) (;
 // Draw a horizontal line through the given point
 cvLine(img, p۱, p۲, color);
 // Draw a vertical line through the given point
 cvLine(img, p۳, p۴, color);
}

//Print the label and then the rect to the console for easy debugging
void printRect(const CvRect rect, const char *label)
{
 if (label)
 std::cout << label;" : " >>
 std::cout << "[Rect] at (" << rect.x << "," << rect.y << ") of size " << rect.width << "x" <<
 rect.height>>
 , " where bottom-right corner is at (" << (rect.x + rect.width-۱) << "," << (rect.y +
 rect.height-۱) << ")" << std::endl;
}

//Make sure the given rectangle is completely within the given image dimensions.
CvRect cropRect(const CvRect rectIn, int w, int h)
{

```

```

CvRect roi = CvRect(rectIn);
// Make sure the displayed image is within the viewing dimensions
if (roi.x + roi.width > w) // Limit the bottom-right from past the image
 roi.width = w - roi.x;
if (roi.y + roi.height > h)
 roi.height = h - roi.y;
if (roi.x < ۰) // Limit the top-left from before the image
 roi.x = ۰;
if (roi.y < ۰(
 roi.y = ۰;
if (roi.x > w-۱) // Limit the top-left from after the image
 roi.x = w-۱;
if (roi.y > h-۱(
 roi.y = h-۱;
if (roi.width < ۰) // Limit the negative sizes
 roi.width = ۰;
if (roi.height < ۰(
 roi.height = ۰;
if (roi.width > w) // Limit the large sizes
 roi.width = w - roi.x;
if (roi.height > h)
 roi.height = h - roi.y;
return roi;
{

-----//

//Image transforming functions
-----//

*/

```



```
//Returns a new image that is a cropped version of the original image.
IplImage* slow_cropImage(const IplImage *img, const CvRect region)
{
 IplImage *imageTmp,*imageRGB;
 CvSize size;
 size.height = img->height;
 size.width = img-> width;
 if (img->width <= ۰ || img->height <= ۰ || region.width <= ۰ || region.height <= ۰){
 std::cerr << "ERROR in cropImage(): dimensions of image or region are invalid." <<
endl;
 printRect(region, "crop region");
 printImageInfo(img, "crop image");
 exit;(۱)
 }

 if (img->depth != IPL_DEPTH_8U){
 std::cerr << "ERROR in cropImage(): Unknown image depth of " << img->depth << "
given instead of 8." << std::endl;
 exit;(۱)
 }

 // First create a new (color or greyscale) IPL Image and copy contents of img into it.
 imageTmp = cvCreateImage(size, IPL_DEPTH_8U, img->nChannels);
 cvCopy(img, imageTmp);
 // Create a new image of the detected region
 // printf("Cropping image at x = %d, y = %d...", faces[i].x, faces[i].y);
 // printf("Setting region of interest...");
 // Make sure the crop region is completely within the image region.
 // cropRegion = cropRect(cropRegion, size.width-۱, size.height-۱); // Make sure it fits
within (۰,۰) and (w,h)

 // Set the region of interest to that surrounding the face.
 cvSetImageROI(imageTmp, region);
 // Copy region of interest (i.e. face) into a new iplImage (imageRGB) and return it
 size.width = region.width;
 size.height = region.height;
```

```

imageRGB = cvCreateImage(size, IPL_DEPTH_8U, img->nChannels);

cvCopy(imageTmp, imageRGB); // Copy just the region.

cvReleaseImage(&imageTmp);
return imageRGB;
{
/*

//Returns a new image that is a cropped version (rectangular cut-out) of the original image.
IplImage* cropImage(const IplImage *img, const CvRect region)
{
 IplImage *imageCropped;
 CvSize size;

 if (img->width <= 0 || img->height <= 0 || region.width <= 0 || region.height <= 0){
 LOG("ERROR in cropImage(): dimensions of image or region are invalid.");
 printRect(region, "crop region");
 printImageInfo(img, "crop image");
 exit(1)
 }

 // Set the desired region of interest.
 cvSetImageROI((IplImage*)img, region);
 // Copy region of interest into a new iplImage and return it
 size.width = region.width;
 size.height = region.height;
 imageCropped = cvCreateImage(size, img->depth, img->nChannels);
 cvCopy(img, imageCropped); // Copy just the region.

 cvResetImageROI((IplImage*)img);
 return imageCropped;
{

//Creates a new image copy that is of a desired size. The aspect ratio will be kept constant if
'keepAspectRatio' is true,

//by cropping undesired parts so that only pixels of the original image are shown, instead of

```

adding extra blank space.

```
//Remember to free the new image later.
```

```
IplImage* resizeImage(const IplImage *origImg, int newWidth, int newHeight, bool
keepAspectRatio)
```

```
{
```

```
 IplImage *outImg = ;
```

```
 int origWidth = ;
```

```
 int origHeight = ;
```

```
 if (origImg){
```

```
 origWidth = origImg->width;
```

```
 origHeight = origImg->height;
```

```
{
```

```
 if (newWidth <= 0 || newHeight <= 0 || origImg == 0 || origWidth <= 0 || origHeight <= 0)
```

```
{
```

```
 std::cerr << "ERROR: Bad desired image size of " << newWidth << "x" << newHeight
<< " in resizeImage().\n";
```

```
 if (origImg)
```

```
 std::cerr << "Image was " << origImg->width << "x" << origImg->height << endl;
```

```
 exit(1)
```

```
{
```

```
 if (keepAspectRatio){
```

```
// cerr << "ResizeImage : " << origWidth << "x" << origHeight << " -> " << newWidth
<< "x" << newHeight << " (rOld" << (origWidth / (float)origHeight) << ", rNew" <<
(newWidth / (float)newHeight) << endl;
```

```
// Resize the image without changing its aspect ratio, by cropping off the edges and
enlarging the middle section.
```

```
 CvRect r;
```

```
 float origAspect = (origWidth / (float)origHeight); // input aspect ratio
```

```
 float newAspect = (newWidth / (float)newHeight); // output aspect ratio
```

```
 if (origAspect > newAspect) { // crop width to be origHeight * newAspect
```

```
 int tw = (origHeight * newWidth) / newHeight;
```

```
 r = cvRect((origWidth - tw)/2, 0, tw, origHeight);
```

```
{
```

```
 else { // crop height to be origWidth / newAspect
```

```

 int th = (origWidth * newHeight) / newWidth;
 r = cvRect(0, (origHeight - th)/۲, origWidth, th);
 {
// cerr << "cropping image to (" << r.width << "x" << r.height << " at (" << r.x << ","
<< r.y << ")." << endl;
 IplImage *croppedImg = cropImage(origImg, r);
// Call this function again, but with the new aspect ratio image.
// cerr << "calling resizeImage(" << newWidth << "," << newHeight << "false)" <<
endl;
 outImg = resizeImage(croppedImg, newWidth, newHeight, false); // do a scaled image
resize, since the aspect ratio is correct now.

 cvReleaseImage(&croppedImg);
 {
 else}

// Scale the image to the new dimensions, even if the aspect ratio will be changed.
 outImg = cvCreateImage(cvSize(newWidth, newHeight), origImg->depth, origImg-
>nChannels);
 if (newWidth > origImg->width && newHeight > origImg->height){
// Make the image larger
// printf("Making the image larger\n"); fflush(stdout);

 cvResetImageROI((IplImage*)origImg);
 cvResize(origImg, outImg, CV_INTER_LINEAR); // CV_INTER_CUBIC or
CV_INTER_LINEAR is good for enlarging
 {
 else}

// Make the image smaller
// printf("Making the image smaller\n"); fflush(stdout);

 cvResetImageROI((IplImage*)origImg);
 cvResize(origImg, outImg, CV_INTER_AREA); // CV_INTER_AREA is good for
shrinking / decimation, but bad at enlarging.
 {

```

```
{
 return outImg;
}

//Rotate the image clockwise and possibly scale the image. Use 'mapRotatedImagePoint()'
to map pixels from the src to dst image.
IplImage *rotateImage(const IplImage *src, float angleDegrees, float scale)
{
 // Create a map_matrix, where the left 2x 2 matrix is the transform and the right 2x 1 is the
 dimensions.

 float m[6];

 CvMat M = cvMat(2, 3, CV_32F, m);

 int w = src->width;
 int h = src->height;

 float divscale = 1.0f;

 if (scale != 1.0f && scale > 1e-20f(
 divscale = 1.0f / scale;

 float angleRadians = angleDegrees * ((float)CV_PI / 180.0f);

 m[0] = (float)(cos(angleRadians) * divscale);
 m[1] = (float)(sin(angleRadians) * divscale);
 m[2] = [-m[1];
 m[3] = m[0];
 m[4] = w*0.5f;
 m[5] = h*0.5f;

 // Make a spare image for the result

 CvSize sizeRotated;
 sizeRotated.width = cvRound(scale * w);
 sizeRotated.height = cvRound(scale * h);

 // Rotate and scale

 IplImage *imageRotated = cvCreateImage(sizeRotated, src->depth, src->nChannels);
```

```
// Transform the image
cvGetQuadrangleSubPix(src, imageRotated, &M);

return imageRotated;
{

//Get the position of a pixel in the image after the rotateImage() operation.
CvPoint_2D32f mapRotatedImagePoint(const CvPoint_2D32f pointOrig, const IplImage
*image, float angleDegrees, float scale)
}

// Get the old image center.
CvPoint_2D32f ptImageCenterOrig = cvPoint_2D32f(image->width / ۲,۰f, image->height /
۲,۰f);

// Get the new image center (after the rotation & scale that was performed to the image by
'rotateImage()').
CvPoint_2D32f ptImageCenterNew = scalePointF(ptImageCenterOrig, scale);

// Get the relative coords of the point
CvPoint_2D32f relPOrig = subtractPointF(pointOrig, ptImageCenterOrig); // relative
coords
// Rotate & scale the relative coords of the point
CvPoint_2D32f relPNew = rotatePointF(relPOrig, angleDegrees);
relPNew = scalePointF(relPNew, scale);
// Get the absolute coords of the new point
CvPoint_2D32f ptPNew = addPointF(relPNew, ptImageCenterNew);
return ptPNew;
{

-----//

//Image utility functions
-----//
```

```

//Do Bilateral Filtering to smooth the image noise but preserve the edges.
//A smoothness of ۵ is very little filtering, and ۱۰۰ is very high filtering.
//Remember to free the returned image.
IplImage* smoothImageBilateral(const IplImage *src, float smoothness)
{
 IplImage *imageSmooth = cvCreateImage(cvGetSize(src), src->depth, src->nChannels);
 IplImage *imageOut = cvCreateImage(cvGetSize(src), src->depth, src->nChannels);
 // Do bilateral filtering on the input image
 cvSmooth(src, imageSmooth, CV_BILATERAL, ۵, ۵, smoothness, smoothness);

 // Mix the smoothed image with the original image
 cvAddWeighted(src, ۰.۷, imageSmooth, ۰.۳, ۰, imageOut);

 // saveImage("bilatA.png", src);
 // saveImage("bilatB.png", imageSmooth);
 // saveImage("bilatC.png", imageOut);
 cvReleaseImage(&imageSmooth);
 return imageOut;
}

//Paste multiple images next to each other as a single image, for saving or displaying.
//Remember to free the returned image.
//Sample usage: cvSaveImage("out.png", combineImages(۲, img۱, img۲) (;
//Modified by Shervin from the cvShowManyImages() function on the OpenCVWiki by
Parameswaran.
' //combineImagesResized()' will resize all images to ۳۰۰x۳۰۰, whereas 'combineImages()'
doesn't resize the images at all.
IplImage* combineImagesResized(int nArgs, ...)
{
 // img - Used for getting the arguments

```

```

IplImage *img;

// DispImage - the image in which input images are to be copied
IplImage *DispImage;

int size;
int i;
int m, n;
int x, y;

// w - Maximum number of images in a row
// h - Maximum number of images in a column
int w, h;

// scale - How much we have to resize the image
float scale;
int max;

// If the number of arguments is lesser than ۰ or greater than ۱۲
// return without displaying
if(nArgs <= ۰){
 printf("Number of arguments too small....\n");
 return NULL;
}
else if(nArgs > ۱۲){
 printf("Number of arguments too large....\n");
 return NULL;
}

// Determine the size of the image,
// and the number of rows/cols
// from number of arguments
else if (nArgs == ۱){
 w = h = ۱;
 size = ۳۰۰;
}

```



```

else if (nArgs == ۲){
 w = ۲; h = ;۱
 size = ;۳۰۰
{
 else if (nArgs == ۳|| nArgs == ۴){
 w = ۲; h = ;۲
 size = ;۳۰۰
 {
 else if (nArgs == ۵|| nArgs == ۶){
 w = ۳; h = ;۲
 size = ;۲۰۰
 {
 else if (nArgs == ۷|| nArgs == ۸){
 w = ۴; h = ;۲
 size = ;۲۰۰
 {
 else{
 w = ۴; h = ;۳
 size = ;۱۵۰
 {
 // Create a new ۳channel image
 DispImage = cvCreateImage(cvSize(۱۰۰+ size*w, ۶۰+ size*h), ۸ , ۳(;

 // Used to get the arguments passed
 va_list args;
 va_start(args, nArgs);

 // Loop for nArgs number of arguments
 for (i = ۰, m = ۲۰, n = ۲۰; i < nArgs; i++, m += (۲۰+ size))}

```

```
// Get the Pointer to the IplImage
img = va_arg(args, IplImage*);

// Make sure a proper image has been obtained
if(img){

// Find the width and height of the image
x = img->width;
y = img->height;

// Find whether height or width is greater in order to resize the image
max = (x > y)? x: y;

// Find the scaling factor to resize the image
scale = (float) ((float) max / size);

// Used to Align the images
if(i % w == ۰ && m != ۲) {
 m = ۲;
 n += ۲ + size;
}

// Make sure we have a color image. If its greyscale, then convert it to color.
IplImage *colorImg = ۰;
IplImage *currImg = img;
if (img->nChannels == ۱) {
 colorImg = cvCreateImage(cvSize(img->width, img->height), ۸, ۳);
// std::cout << "[Converting greyscale image " << greyImg->width << "x" <<
greyImg->height << "px to color for combineImages()]" << std::endl;
 cvCvtColor(img, colorImg, CV_GRAY2BGR);
 currImg = colorImg; // Use the greyscale version as the input.
}

// Set the image ROI to display the current image
```

```

cvSetImageROI(DispImage, cvRect(m, n, (int)(x/scale), (int)(y/scale)));

// Resize the input image and copy it to the Single Big Image
cvResize(currImg, DispImage, CV_INTER_CUBIC);

// Reset the ROI in order to display the next image
cvResetImageROI(DispImage);

if (colorImg)
 cvReleaseImage(&colorImg);
{
 else { // This input image is NULL
// printf("Error in combineImages(): Bad image%d given as argument\n", i);
// cvReleaseImage(&DispImage); // Release the image and return
// return NULL;
{
{

// End the number of arguments
va_end(args);

return DispImage;
{

//Paste multiple images next to each other as a single image, for saving or displaying.
//Remember to free the returned image.
//Sample usage: cvSaveImage("out.png", combineImages(۲, img۱, img۲) (;
//Modified by Shervin from the cvShowManyImages() function on the OpenCVWiki by
Parameswaran.

' //combineImagesResized()' will resize all images to ۳۰۰x۳۰۰, whereas 'combineImages()'
doesn't resize the images at all.
IplImage* combineImages(int nArgs, ...)
}

```

```

const int MAX_COMBINED_IMAGES = ۶۰;

int col\Width, col۲Width;

int row\Height, row۲Height, row۳Height;

IplImage *imageArray[MAX_COMBINED_IMAGES];
int xPos[MAX_COMBINED_IMAGES];
int yPos[MAX_COMBINED_IMAGES];
int wImg[MAX_COMBINED_IMAGES] = {۰/}; // image dimensions are assumed to be
۰, if they dont exist.

int hImg[MAX_COMBINED_IMAGES]; {۰} =

// int rows, columns; // number of rows & cols of images.

int wP, hP; // dimensions of the combined image.

int i;
int nGoodImages = ۰;

IplImage *combinedImage;
int B = ۵; // Border size, in pixels

// Load all the images that were passed as arguments
va_list args; // Used to get the arguments passed
va_start(args, nArgs);
for (i = ۰; i < nArgs; i++)}

// Get the Pointer to the IplImage
IplImage *img = va_arg(args, IplImage*);
// Make sure a proper image has been obtained, and that there aren't too many images
already.

if ((img != ۰&& img->width > ۰&& img->height > ۰) && (nGoodImages <
MAX_COMBINED_IMAGES))}

// Add the new image to the array of images

imageArray[nGoodImages] = img;
wImg[nGoodImages] = img->width;
hImg[nGoodImages] = img->height;
nGoodImages++;

{

```

```

{

// If the number of arguments is lesser than ۱ or greater than ۱۲
// return without displaying
if(nGoodImages <= ۱ || nGoodImages > MAX_COMBINED_IMAGES) {
 printf("Error in combineImages(): Cant display %d of %d images\n", nGoodImages,
nArgs);
 return NULL;
}

// Determine the size of the combined image & number of rows/cols.
// columns = MIN(nGoodImages, ۲ // ۱ ; \ or ۲ columns
// rows = (nGoodImages-۱ / (۲ // ۱ ; \ or ۲ or ۳ or ... rows
col\Width = MAX(wImg[۱], MAX(wImg[۲], wImg[۳]([;
col۲Width = MAX(wImg[۱], MAX(wImg[۳], wImg[۵]([;
row\Height = MAX(hImg[۱], hImg[۱]([;
row۲Height = MAX(hImg[۲], hImg[۳]([;
row۳Height = MAX(hImg[۴], hImg[۵]([;
wP = B + col\Width + B + (col۲Width ? col۲Width + B : ۰(;
hP = B + row\Height + B + (row۲Height ? row۲Height + B : ۰) + (row۳Height ?
row۳Height + B : ۰(;
xPos[۱] = B;
yPos[۱] = B;
xPos[۱] = B + col\Width + B;
yPos[۱] = B;
xPos[۲] = B;
yPos[۲] = B + row\Height + B;
xPos[۳] = B + col\Width + B;
yPos[۳] = B + row\Height + B;
xPos[۴] = B;

```

```

yPos[۴] = B + row\Height + B + row۲Height + B;
xPos[۵] = B + col\Width + B;
yPos[۵] = B + row\Height + B + row۲Height + B;

// Create a new RGB image
combinedImage = cvCreateImage(cvSize(wP, hP), ۸ , ۳(;
if (!combinedImage)
 return NULL;

// Clear the background
cvSet(combinedImage, CV_RGB(۵۰, ۵۰, ۵۰)((;

for (i=۰; i < nGoodImages; i++)}
 IplImage *img = imageArray[i];

// Make sure we have a color image. If its greyscale, then convert it to color.
IplImage *colorImg = ;۰
if (img->nChannels == ۱){
 colorImg = cvCreateImage(cvSize(img->width, img->height), ۸ , ۳(;
 cvCvtColor(img, colorImg, CV_GRAY۲BGR);
 img = colorImg; // Use the greyscale version as the input.
}

// Set the image ROI to display the current image
cvSetImageROI(combinedImage, cvRect(xPos[i], yPos[i], img->width, img->height));
// Draw this image into its position
cvCopy(img, combinedImage);
// Reset the ROI in order to display the next image
cvResetImageROI(combinedImage);

if (colorImg)
 cvReleaseImage(&colorImg);
{

```

```
// End the number of arguments
va_end(args);

return combinedImage;
{

//Blend color images 'image\1' and 'image\2' using an 8-bit alpha-blending mask channel.
//Equivalent to this operation on each pixel: imageOut = image\1 * (imageAlphaMask/255) + image\2 * (imageAlphaMask/255)
//So if a pixel in imageAlphaMask is 255, then that pixel in imageOut will be image\1, or if
imageAlphaMask is 0 then imageOut is image\2
//or if imageAlphaMask was 128 then imageOut would be: image\1 * 0.5 + image\2 * 0.5
//Returns a new image, so remember to call 'cvReleaseImage()' on the result.
IplImage* blendImage(const IplImage* image\1, const IplImage* image\2, const IplImage*
imageAlphaMask)
{
// Make sure that image\1 & image\2 are RGB UCHAR images, and imageAlphaMask is an
8-bit UCHAR image.

if (!image\1 || image\1->width <= 0 || image\1->height <= 0 || image\1->depth != 8 || image\1-
>nChannels != 3){

std::cout << "Error in blendImage(): Bad parameter 'image\1'." << std::endl;
printImageInfo(image\1, "image\1");
return NULL;
}

if (!image\2 || image\2->width <= 0 || image\2->height <= 0 || image\2->depth != 8 || image\2-
>nChannels != 3){

std::cout << "Error in blendImage(): Bad parameter 'image\2'." << std::endl;
printImageInfo(image\2, "image\2");
return NULL;
}
}
```

```

if (!imageAlphaMask || imageAlphaMask->width <= ۰ || imageAlphaMask->height <= ۰ ||
imageAlphaMask->depth != ۸ || imageAlphaMask->nChannels != ۱){
 std::cout << "Error in blendImage(): Bad parameter 'imageAlphaMask'." << std::endl;
 printImageInfo(imageAlphaMask, "imageAlphaMask");
 return NULL;
}
// Make sure that image ۱& image ۲& imageAlphaMask are all the same dimensions
if ((image۱->width != image۲->width || image۱->width != imageAlphaMask->width || (
 (image۱->height != image۲->height || image۱->height != imageAlphaMask->height) (
}
 std::cout << "Error in blendImage(): Input images aren't the same dimensions." <<
std::endl;
 printImageInfo(image۱, "image۱(");
 printImageInfo(image۲, "image۲(");
 printImageInfo(imageAlphaMask, "imageAlphaMask");
 return NULL;
}

// Combine the ۲images using the alpha channel mask
int width = image۱->width;
int height = image۱->height;
IplImage *imageBlended = cvCreateImage(cvSize(width, height), image۱->depth,
image۱->nChannels);

int widthStep ۱= image۱->widthStep;
int widthStep ۲= image۲->widthStep;
int widthStepM = imageAlphaMask->widthStep;
int widthStepBlended = imageBlended->widthStep;
// Start from the beginning of the image and move through the image, row by row.
UCHAR *p ۱= (UCHAR*)image۱->imageData;
UCHAR *p ۲= (UCHAR*)image۲->imageData;
UCHAR *pM = (UCHAR*)imageAlphaMask->imageData;
UCHAR *pBlended = (UCHAR*)imageBlended->imageData;

```



```

for (int y=۰; y < height; y++){
 for (int x=۰; x < width; x++){
 int m = *pM++; // Get the alpha mask value (۰ to ۲۵۵) and move to its next pixel.
 int part\B, part\G, part\B;
 int part۲B, part۲G, part۲B;
 int pixel\B = *p۱++; // Get the B value and move to its next pixel.
 int pixel\G = *p۱++; // Get the G value and move to its next pixel.
 int pixel\R = *p۱++; // Get the R value and move to its next pixel.

 // Instead of dividing slowly by ۲۵۵: multiply image ۲with (alpha mask + ۱), then
 divide it by ۲۵۶

 // It is similar to dividing by ۲۵۵, but much faster.
 // But note that if m was ۰, it would be slightly off, so m == ۰ is treated separately.
 if (m) {
 int f \= m + ۱;

 int pixel۲B = *p۲++; // Get the B value and move to its next pixel.
 int pixel۲G = *p۲++; // Get the G value and move to its next pixel.
 int pixel۲R = *p۲++; // Get the R value and move to its next pixel.

 part۲B = (pixel۲B * f \ << (;^
 part۲G = (pixel۲G * f \ << (;^
 part۲R = (pixel۲R * f \ << (;^

 // Multiply image ۱with (۲۵۵- alpha mask), then divide it by ۲۵۶. Similar to
 dividing by ۲۵۵,۰, but much faster.

 int f ۲ = ۲۵۵- m;
 part\B = (pixel\B * f ۲ << (;^
 part\G = (pixel\G * f ۲ << (;^
 part\R = (pixel\R * f ۲ << (;^
 }
 else{

 // If the alpha mask was ۰, then just use image ۱and not image.۲

```

```

 part\B = pixel\B;
 part\G = pixel\G;
 part\R = pixel\R;
 part\B = ;
 part\G = ;
 part\R = ;
 p += 3; // move the pointer to the next pixel.
{
 // Combine the part of image \ with the part of image, and then move to the next
 pixel.
 * pBlended++ = part\B + part\B;
 * pBlended++ = part\G + part\G;
 * pBlended++ = part\R + part\R;
 {
 // Move everything to the next row by adding widthStep and subtracting how much of
 the row was already processed.
 p += widthStep - width * 3; // Move to the next RGB row, ignoring any row
 padding.
 p += widthStep - width * 3; // Move to the next RGB row, ignoring any row
 padding.
 pM += widthStepM - width; // Move to the next Greyscale row, ignoring any row
 padding.
 pBlended += widthStepBlended - width * 3; // Move to the next RGB row, ignoring
 any row padding.
 {
 return imageBlended;
 {

```

```
//Save the given image to a JPG or BMP file, even if its format isn't an 1-bit image, such as
a 32bit float image.
int saveImage(const char *filename, const IplImage *image)
{
 int ret = -1;
#ifdef USE_HIGHGUI
 IplImage *image1Bit = cvCreateImage(cvSize(image->width,image->height),
IPL_DEPTH_1U, image->nChannels); // 1-bit greyscale image.

 if (image1Bit)
 cvConvert(image, image1Bit); // Convert to an 1-bit image instead of potentially
16,24,32or 64bit image.

 if (image1Bit)
 ret = cvSaveImage(filename, image1Bit);

 if (image1Bit)
 cvReleaseImage(&image1Bit);
#endif
 return ret;
}

//Store a greyscale floating-point CvMat image into a BMP/JPG/GIF/PNG image,
//since cvSaveImage() can only handle 1bit images (not 32bit float images).
void saveFloatMat(const char *filename, const CvMat *srcMat)
{
 // cout << "in Saving Image(" << ((CvMat*)src)->width << "," << ((CvMat*)src)->height
<< ") " << filename << "." << endl;
 if (CV_MAT_CN(srcMat->type) != 1 || CV_MAT_DEPTH(srcMat->type) != 32){
 LOG("ERROR in saveFloatMat(): Matrix is not single-channel 32-bit float matrix!");
 return;
 }

 // Fill the Matrix's float data as a float image into this temporary image, since it wont be
needed after this function.
```

```

IplImage srcIplImg;
cvGetImage(srcMat, &srcIplImg);
// Store the float image
saveFloatImage(filename, &srcIplImg);
{

//Get an 8-bit equivalent of the 32-bit Float Matrix.

//Returns a new image, so remember to call 'cvReleaseImage()' on the result.
IplImage* convertMatrixToUcharImage(const CvMat *srcMat)
}

// Fill the Matrix's float data as a float image into this image.

IplImage srcIplImg;
cvGetImage(srcMat, &srcIplImg);

// Convert the float image into a normal Uchar image.
return convertFloatImageToUcharImage(&srcIplImg);
{

//Get an 8-bit equivalent of the 32-bit Float image.

//Returns a new image, so remember to call 'cvReleaseImage()' on the result.
IplImage* convertFloatImageToUcharImage(const IplImage *srcImg)
}

IplImage *dstImg = ;
if ((srcImg) && (srcImg->width > 0 && srcImg->height > 0) {

 if (srcImg->nChannels != 1 || srcImg->depth != 32) {
 LOG("ERROR in convertFloatImageToUcharImage(): Image is not single-channel
 32-bit float image!");
 return ;
 }

// Spread the 32bit floating point pixels to fit within 8bit pixel range.
double minVal, maxVal;
cvMinMaxLoc(srcImg, &minVal, &maxVal);

```

```

LOG("FloatImage:(minV=%lf, maxV=%lf)", minVal, maxVal);

// Deal with NaN and extreme values, since the DFT seems to give some NaN results.
if (cvIsNaN(minVal) || minVal < -1e30(
 minVal = -1e30;
if (cvIsNaN(maxVal) || maxVal > 1e30(
 maxVal = 1e30;
if (maxVal-minVal == 0,0f)
 maxVal = minVal + 0,001; // remove potential divide by zero errors.

// Convert the format
dstImg = cvCreateImage(cvSize(srcImg->width, srcImg->height), 8, 1(
 cvConvertScale(srcImg, dstImg, 255,0/ (maxVal - minVal), - minVal * 255,0 /
(maxVal-minVal));
{
 return dstImg;
}

//Store a greyscale floating-point CvMat image into a BMP/JPG/GIF/PNG image,
//since cvSaveImage() can only handle 8bit images (not 32bit float images).
void saveFloatImage(const char *filename, const IplImage *srcImg)
{
#ifdef USE_HIGHGUI
// LOG("Saving Float Image '%s' (%dx%d)", filename, srcImg->width, srcImg->height);
if (srcImg->nChannels != 1 || srcImg->depth != 32){
 LOG("ERROR in saveFloatImage(): Image is not single-channel 32-bit float image!");
 return;
}

IplImage *byteImg = convertFloatImageToUcharImage(srcImg);
cvSaveImage(filename, byteImg);
cvReleaseImage(&byteImg);
// cout << "done saveFloatImage()" << endl;
#endif

```

```
{

//Draw some text onto an image using printf() format.
void drawText(IplImage *img, CvPoint position, CvScalar color, char *fmt, ...)
{

// Write the given arguments to 'szMsg' using printf() style formatting.

va_list marker;
char szMsg[۱۰۲۴]

va_start(marker, fmt);
vsprintf(szMsg, fmt, marker);
va_end(marker);
// marker = marker; // stop warning messages

// Display the text onto the image.

CvFont font;
cvInitFont(&font, CV_FONT_HERSHEY_SIMPLEX, ۰٫۳،۰٫۳،۰٫۱, CV_AA);

cvPutText(img, szMsg, position, &font, color);
{
```

در ابتدا برنامه و کتابخانه‌های مورد نیاز به برنامه اضافه می‌شود.

```
#include <sstream>
#include <string>
#include <iostream>
#include <fstream>
#include <vector>
#include "Object.cpp"
```

اندازه فریم دریافتی از دوربین تنظیم می‌شود. (برای این فایل پردازش فریم با اندازه بزرگ در توان ریسپری نبود به همین دلیل اندازه فریم را به ۶۴۰\*۴۸۰ کاهش دادیم.)

```
//default capture width and height.
const int FRAME_WIDTH = 640;
const int FRAME_HEIGHT = 480;
```

با توجه به اینکه ممکن است نویزهایی از محیط هم به عنوان جسم مورد نظر در نظر گرفته شود، یک آستانه برای اندازه کمینه و بیشینه برای جسم مورد نظرمان در نظر می‌گیریم. باید توجه داشت که این اندازه باید با توجه به اندازه فریم مورد نظر که در بالا تنظیم شد در نظر گرفته شود.

```
//minimum and maximum object area
const int MIN_OBJECT_AREA = 50*50;
const int MAX_OBJECT_AREA = FRAME_HEIGHT*FRAME_WIDTH/1.5;
```

مقداردهی اولیه متغیرها

```
Mat src;
int posX=0;
int posY=0;
int counter_R_L=0;
int counter_U_D=0;
int counter_NoMove;0=
Mat current_frame;
```

۲-۱۶) تابع drawObject:

در واقع، معیار موقعیت شی در هر فریم را موقعیت میانگین بین همه‌ی اشیاء یافت شده در آن فریم در نظر می‌گیریم. این کار با این فرض انجام می‌شود که به جز آرم مورد نظر، هیچ شی سبز متحرکی در پس زمینه وجود ندارد. و در صورتی که شی سبزی هم بود و ثابت بود در تصمیم‌گیری انجام شده نسبت به جهت حرکت آرم تاثیر نگذارد. زیرا اگر همه‌ی شی‌ها ثابت باشند و فقط یک جسم متحرک باشد، جهت حرکت موقعیت میانگین همه‌ی اشیاء در همه‌ی فریم‌ها، همان جهت حرکت شی متحرک است.

در این تابع تمام objectهایی که به رنگ مورد نظر (در این برنامه رنگ سبز لیمویی) را دارا هستند را دریافت می‌کند.

```
void drawObject(vector<Object> theObjects,Mat &frame, Mat &temp, vector<vector<Point>> contours, vector<Vec4i> hierarchy){
```

در هر فریم، بین مقدار x و y همه‌ی objectهای یافت شده میانگین گرفته می‌شود.

```
int NewAveragePosX=;
```

```
int NewAveragePosY=;
```

```
for(int i = ۰; i<theObjects.size(); i++){
```

```
 NewAveragePosX+=NewAveragePosX+theObjects.at(i).getXPos();
```

```
 NewAveragePosY+=NewAveragePosY+theObjects.at(i).getYPos();
```

```
{
```

با توجه به اینکه موقعیت در تصویر به معنای شماره پیکسل در سطر و ستون است، باید این عدد یک مقدار صحیح باشد.

```
 NewAveragePosX=floor(NewAveragePosX/theObjects.size());
```

```
 NewAveragePosY=floor(NewAveragePosY/theObjects.size());
```

متغیر  $posX$  و  $posY$  نشان‌دهنده‌ی موقعیت میانگین شی‌های دیده‌شده در فریم قبلی هستند. اگر مقدار آن‌ها برابر صفر باشد به این معنا است که فریم جاری اولین فریم دریافت شده است. پس تنها کاری که در این حالت انجام می‌شود این است که این دو متغیر با مقدار  $NewAveragePosX$  و  $NewAveragePosY$  مقداردهی می‌شوند.

```
 if(posX== ۰ && posY==۰(
```

```
{
```

```
 posX=NewAveragePosX ;
```

```
 posY=NewAveragePosY;
```

```
 }else{
```

در صورتی که مقادیر متغیرهای  $posX$  و  $posY$  غیر صفر باشند، با مقایسه‌ی موقعیت میانگین محاسبه شده در این فریم جاری با موقعیت میانگین در فریم قبلی، جهت و راستای حرکت جسم مورد نظر را محاسبه می‌کنیم.

```
 int xdir=posX-NewAveragePosX ;
```

```
 int ydir=posY-NewAveragePosY;
```

در صورتی که مقدار حرکت جسم حداقل در یکی از راستاهای  $X$  و یا  $Y$ ، بیشتر از ۱۰ پیکسل باشد، آن را به عنوان حرکت جسم قبول می‌کنیم. در غیر این صورت آن را جزء خطای محاسباتی و ضبط فریم در نظر می‌گیریم. (این مقدار ۱۰ پیکسل بر اساس آزمایش و همچنین با توجه به اندازه فریم دریافتی بدست آمده است و هیچ فرمول محاسباتی ندارد!)

```
 if(abs(xdir)>10 || abs(ydir)>10){
```

برای بدست آوردن راستای حرکت شی مورد نظر، تفاضل قدرمطلق اندازه حرکت در جهت  $X$  و  $Y$  را محاسبه می‌کنیم.

```
 int dirupdown=abs(xdir)-abs(ydir);
```



اگر مقدار dirupodown مثبت باشد، به این معنا است که حرکت در راستای X بیشتر بوده است به همین دلیل یک واحد به متغیر counter\_R\_L اضافه می‌شود. متغیر counter\_R\_L شمارنده تعداد دفعاتی است که جسم در راستای X حرکت کرده است.

```
if(dirupodown>0){
 counter_R_L+=1;
```

اگر مقدار dirupodown منفی باشد، به این معنا است که حرکت در راستای Y بیشتر بوده است به همین دلیل یک واحد به متغیر counter\_U\_D اضافه می‌شود. متغیر counter\_U\_D شمارنده تعداد دفعاتی است که جسم در راستای Y حرکت کرده است.

```
}else if(dirupodown<0){
 counter_U_D+=1;
{
```

مقدار متغیرهای posX و posY را با مقدارهای NewAveragePosX و NewAveragePosY محاسبه شده در فریم جاری مقداردهی می‌کنیم.

```
posX=NewAveragePosX;
posY=NewAveragePosY;
```

```
}else{
```

اگر مقدار حرکت در هر دو راستای X و Y کم‌تر از ۱۰ پیکسل باشد، آن را بدون حرکت در نظر می‌گیریم و متغیر counter\_NoMove که نشان‌دهنده تعداد فریم‌های ثابت است را یک واحد زیاد می‌کنیم.

```
counter_NoMove+=1;
posX=NewAveragePosX;
posY=NewAveragePosY;
```

```
{
{
```

```
{
```

## ۲-۱۷) تابع morphOps:

این تابع برای انجام پیش پردازش بر روی فریم جاری است و به این منظور نوشته شده است که نویز-های کوچک تر از مقدار خاصی را در تصویر را حذف کند. همچنین شی های بزرگ تر را صاف تر و یکدست تر کند.

```
void morphOps(Mat &thresh){

 //create structuring element that will be used to “dilate” and “erode” image.
 //the element chosen here is a ELLIPSE
 Mat erodeElement = getStructuringElement(MORPH_ELLIPSE ,Size(8,8)((;

 //dilate with larger element so make sure object is nicely visible
 Mat dilateElement = getStructuringElement(MORPH_ELLIPSE ,Size(21,21)((;

 erode(thresh,thresh,erodeElement);
 erode(thresh,thresh,erodeElement);

 dilate(thresh,thresh,dilateElement);
 dilate(thresh,thresh,dilateElement);

 {
```

## ۲-۱۸) تابع trackFilteredObject

```
void trackFilteredObject(Object theObject,Mat threshold,Mat HSV, Mat ¤t_frame){

 vector <Object> objects;
 Mat temp;
 threshold.copyTo(temp);

 //these two vectors needed for output of findContours
 vector< vector<Point> > contours;
 vector<Vec۴i> hierarchy;

 //find contours of filtered image using openCV findContours function
 findContours(temp,contours,hierarchy,CV_RETR_CCOMP,CV_CHAIN_APPROX
_SIMPLE);

 //use moments method to find our filtered object
 bool objectFound = false;
```

```

if (hierarchy.size() > 0 {
 int numObjects = hierarchy.size();
 for (int index = 0; index < numObjects; index++) {
 Moments moment = moments((cv::Mat)contours[index]);
 double area = moment.m00;

 //if the area is less than MIN_OBJECT_AREA px then it is probably just
 //if the area is the same as the 1/3 of the image size, probably just a bad
 //we only want the object with the largest area so we save a reference area
 //iteration and compare it to the area in the next iteration.
 if(area>MIN_OBJECT_AREA){

 Object object;

 object.setXPos(moment.m10/area);
 object.setYPos(moment.m01/area);
 cout<< "x"<<moment.m10/area << endl;
 cout<< "y"<<moment.m01/area << endl;
 object.setType(theObject.getType());
 object.setColor(theObject.getColor());
 objects.push_back(object);
 objectFound = true;

 }
 {else objectFound = false;
 {
 //let user know you found an object
 if(objectFound ==true){
 drawObject(objects,current_frame,temp,contours,hierarchy);
 }
 }
 }
}

```

تابع main (۱۹-۲)

```

int main(int argc, char* argv[])
{

 //Matrix to store each frame of the webcam feed

 Mat threshold;
 Mat HSV;

```

```
//video capture object to acquire webcam feed
VideoCapture capture;

//open capture object at location zero (default location for webcam)
capture.open(0);

//set height and width of capture frame
capture.set(CV_CAP_PROP_FRAME_WIDTH,FRAME_WIDTH);
capture.set(CV_CAP_PROP_FRAME_HEIGHT,FRAME_HEIGHT);

// create and open "GestureResult.txt" file for write result in it.
ofstream f;
f.open ("GestureResult.txt");
// we just get 15frame. It can be change.
for(int counter=0;counter<15;counter++){

 //store image to matrix
 capture.read(current_frame);
 src = current_frame ;
 if(!src.data)
 } return -1;

 //convert frame from BGR to HSV colorspace
 cvtColor(current_frame,HSV,COLOR_BGR2HSV);

 //create some temp fruit objects so that

 //we can use their member functions/information
 Object green("green");
 cvtColor(current_frame,HSV,COLOR_BGR2HSV);

 inRange(HSV,green.getHSVmin(),green.getHSVmax(),threshold);
 morphOps(threshold);
 trackFilteredObject(green,threshold,HSV,current_frame);
}
if(counter_R_L == 0 && counter_U_D == 0 && counter_NoMove == 0){
 f << "NoGusture;"
 cout<< "NoGusture"<< endl;
}else if((counter_R_L > counter_U_D) && (counter_R_L > counter_NoMove)){
 f << "No;"
```

```

 cout<< "No"<< endl;
 }else if(counter_U_D > counter_NoMove){
 f << "Yes;"
 cout<< "Yes"<< endl;
 }else{
 f << "Bad Gusture;"
 cout<< "Bad Gusture"<< endl ;
 }
 f.close;()
 return 0;
}

```

### ۳. کدهای بخش جستجو و بازیابی اطلاعات

```

public static DataSet SearchByBM25(string Keyword)
{
 DataSet dsSearchResults = new DataSet();
 string preProcessedKeyword = Stemming.preProcess(Keyword);
 List<string> QueryWords = preProcessedKeyword.Split(' ').ToList();
 int QueryWordsCount = QueryWords.Count;
 Dictionary<TF, float>[] QueryWordsTFValues = new Dictionary<TF,
float>[QueryWordsCount];
 int[] FDValues = new int[QueryWordsCount];
 double[] IDFValues = new double[QueryWordsCount];
 float ArticlesLengthAverage = ArticlesLength.Average(kvp => kvp.Value);
 int SelectedArticlesCount = SelectedArticles.Count;
 for (int i = 0; i < QueryWordsCount; i++)
 {
 Dictionary<TF, float> TFValues = CalculateTF(QueryWords[i]);
 QueryWordsTFValues[i] = TFValues;
 int FD = TFValues.Where(kvp => kvp.Value != 0).Count();
 FDValues[i] = FD;
 IDFValues[i] = Math.Log((SelectedArticlesCount - FD + 0.5) / (FD + 0.5));
 }
 List<Score> ArticlesScore = new List<Score>();
 // Calculate Score
 string CurrentFileName = @"Files\" + Keyword + ".csv";
 string CurrentFilePath = HttpContext.Current.Server.MapPath(CurrentFileName);
 foreach (Article CurrentArticle in SelectedArticles)
 {
 float ArticleLength = ArticlesLength.Where(kvp => kvp.Key.ArticleID ==
CurrentArticle.ID).FirstOrDefault().Value;

 double Score = 0;
 }
}

```

```

 for (int i = 0; i < QueryWordsCount; i++)
 {
 float TFValue = QueryWordsTFValues[i].Where(kvp => kvp.Key.ArticleID
== CurrentArticle.ID).FirstOrDefault().Value;
 double IDFValue = IDFValues[i];
 double CurrentScore = (((K1Value+1)*TFValue) / (K1Value * ((1 - BValue)
+ (BValue * ArticleLength) / ArticlesLengthAverage)) + TFValue) * IDFValue;
 Score += CurrentScore;
 }
 Score CurrentArticleScore = new Score { ArticleID = CurrentArticle.ID,
ScoreValue = Score };
 ArticlesScore.Add(CurrentArticleScore);
 }

 var SortedArticleIDScore = from entry in ArticlesScore orderby entry.ScoreValue
descending select entry;

 DataTable tblSearchResult = new DataTable("articles");
 tblSearchResult.Columns.Add("ID");
 tblSearchResult.Columns.Add("Title");
 tblSearchResult.Columns.Add("Abstract");
 tblSearchResult.Columns.Add("FirstSubject");
 tblSearchResult.Columns.Add("SecondSubject");
 tblSearchResult.Columns.Add("Score");

 foreach (var item in SortedArticleIDScore)
 {
 int CurrentItemArticleID = item.ArticleID;
 Article CurrentArticle = SelectedArticles[SelectedArticles.FindIndex(o => o.ID
== CurrentItemArticleID)];
 if (tblSearchResult.Rows.Count < 100)
 tblSearchResult.Rows.Add(CurrentArticle.ID, CurrentArticle.Title,
CurrentArticle.Abstract, CurrentArticle.FirstSubject, CurrentArticle.SecondSubject,
item.ScoreValue);
 else
 break;
 }
 dsSearchResults.Tables.Add(tblSearchResult);
 return dsSearchResults;
}

public static Dictionary<WordCount, float> CalculateArticleLength()
{
 Dictionary<WordCount, float> Result = new Dictionary<WordCount, float>();
 foreach (Article CurrentArticle in SelectedArticles)
 {
 WordCount CurrentWordCount = new WordCount();
 CurrentWordCount.ArticleID = CurrentArticle.ID;
 CurrentWordCount.TitleWordCount = CurrentArticle.TitleWordCount;
 CurrentWordCount.AbstractWordCount = CurrentArticle.AbstractWordCount;
 }
}

```

```

 CurrentWordCount.TagsWordCount = CurrentArticle.TagsWordCount;
 CurrentWordCount.WeightedTitleWordCount =
CurrentWordCount.TitleWordCount * TitleWeight;
 CurrentWordCount.WeightedAbstractWordCount =
CurrentWordCount.AbstractWordCount * AbstractWeight;
 CurrentWordCount.WeightedTagsWordCount =
CurrentWordCount.TagsWordCount * TagWeight;
 float ArticleLengthValue = CurrentWordCount.WeightedTitleWordCount +
CurrentWordCount.WeightedAbstractWordCount +
CurrentWordCount.WeightedTagsWordCount;
 Result.Add(CurrentWordCount, ArticleLengthValue);
 }
 return Result;
}

public static Dictionary<TF, float> CalculateTF(string Keyword)
{
 Dictionary<TF, float> Result = new Dictionary<TF, float>();
 foreach (Article CurrentArticle in SelectedArticles)
 {
 TF CurrentTF = new TF();
 CurrentTF.ArticleID = CurrentArticle.ID;
 CurrentTF.Keyword = Keyword;
 CurrentTF.FreqInTitle = OccurrencesCount(CurrentArticle.Title, Keyword);
 CurrentTF.FreqInAbstract = OccurrencesCount(CurrentArticle.Abstract,
Keyword);
 CurrentTF.FreqInTags = OccurrencesCount(CurrentArticle.Tags, Keyword);
 CurrentTF.WeightedFreqInTitle = CurrentTF.FreqInTitle * TitleWeight;
 CurrentTF.WeightedFreqInAbstract = CurrentTF.FreqInAbstract *
AbstractWeight;
 CurrentTF.WeightedFreqInTags = CurrentTF.FreqInTags * TagWeight;
 float TFValue = CurrentTF.WeightedFreqInTitle +
CurrentTF.WeightedFreqInAbstract + CurrentTF.WeightedFreqInTags;
 Result.Add(CurrentTF, TFValue);
 }
 return Result;
}

public static int OccurrencesCount(string SearchString, string StringToFind)
{
 int Count = 0;
 int Position = 0;
 while ((Position < SearchString.Length) && (Position =
SearchString.IndexOf(StringToFind, Position)) != -1)
 {
 Count++;
 Position += StringToFind.Length;
 }
 return Count;
}

```

#### ۴. کدهای برنامه اتصال به زیرسیستم تشخیص گفتار

```
using AsrGooyeshPardaz.NevisaModule;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Net.NetworkInformation;
using System.ServiceProcess;
using System.Diagnostics;
using System.Threading;
using System.Text;
using System.Windows.Forms;
using System.Net;
using System.Net.Sockets;
using Lemmatizer;

namespace Nevisa01
{
 public partial class frmMain : Form
 {
 string NevisaPath;
 bool NevisaInitialized;

 private Socket Listener;
 private Socket Client;
 private int MaxConnections = 30;
 private int MessageLength = 6;
 private byte[] ReceivedBuffer;
 private bool IsClientConnect = false;
 Thread RobotInteractionThread;
 public bool ProcessRecognizedText = false;
 StemmingTools st;
 string AllRecognizedText = "";
 string PreviuosRecognizedText = "";

 static char[] splitChars = { ' ', '\', '\r', '\n' };
 static string stoptexts;
 static List<string> beforeWords;
 public static string[] stop;

 public frmMain()
```



```

 }
 InitializeComponent();
 {

private void frmMain_Load(object sender, EventArgs e)
 {
 InitializeStemmingTools();
 try
 {
 // Load from setting
 txtNevisaPath.Text = Properties.Settings.Default.NevisaPath;
 txtServerAddress.Text = Properties.Settings.Default.ServerAddress;
 txtInterfaceIP.Text = Properties.Settings.Default.InterfaceIP;
 txtInterfacePort.Text = Properties.Settings.Default.InterfacePort;
 btnCheckNevisaPath_Click(sender, e);
 //btnSetServerAddress_Click(sender, e);
 //btnStartServer_Click(sender, e);
 btnInitializeNevisa_Click(sender, e);
 }
 catch (Exception exp)
 {
 MessageBox.Show("Error occured in reading default setting!\\n" + exp.Message);
 }
 }

private void btnBrowseNevisaPath_Click(object sender, EventArgs e)
 {
 if (fbdBrowseNevisaPath.ShowDialog() == DialogResult.OK)
 {
 txtNevisaPath.Text = fbdBrowseNevisaPath.SelectedPath;
 NevisaPath = txtNevisaPath.Text;
 }
 }

private void btnCheckNevisaPath_Click(object sender, EventArgs e)
 {
 bool result = true;
 NevisaPath = txtNevisaPath.Text;
 if (File.Exists(NevisaPath + @"\StartNevisaModuleService.cmd"))
 if (File.Exists(NevisaPath + @"\StopNevisaModuleService.cmd"))
 result = true;

 if (result)
 {
 txtNevisaPath.Enabled = false;
 btnBrowseNevisaPath.Enabled = false;
 btnStartServer.Enabled = true;
 btnStopServer.Enabled = true;
 btnInitializeNevisa.Enabled = true;
 }
 }

```

```

{
 else {
 txtNevisaPath.Enabled = true;
 btnBrowseNevisaPath.Enabled = true;
 btnStartServer.Enabled = false;
 btnStopServer.Enabled = false;
 btnInitializeNevisa.Enabled = false;
 }
}

private void btnSetServerAddress_Click(object sender, EventArgs e)
{
 Ping ping = new Ping();
 PingReply pingResult = null;
 try
 {
 pingResult = ping.Send(txtServerAddress.Text);
 }
 catch (Exception exp)
 {
 MessageBox.Show("Server address is not correct");
 return;
 }
 if (pingResult.Status.ToString() == "Success")
 {
 txtServerAddress.ReadOnly = true;
 //SimpleNevisaInterface.ServerAddress = txtServerAddress.Text;
 }
 else
 {
 MessageBox.Show("Server is not available!");
 }
}

private void btnStartServer_Click(object sender, EventArgs e)
{
 ServiceController service = new ServiceController("NevisaModuleService");
 try
 {
 if (service.Status != ServiceControllerStatus.Running)
 service.Start();
 service.WaitForStatus(ServiceControllerStatus.Running);
 }
 catch {
 MessageBox.Show("Error in start service");
 }
}

private void btnStopServer_Click(object sender, EventArgs e)
{
}

```

```

ServiceController service = new ServiceController("NevisaModuleService");
try
{
 if (service.Status != ServiceControllerStatus.Stopped)
 service.Stop();
 service.WaitForStatus(ServiceControllerStatus.Stopped);
}
catch
{
 MessageBox.Show("Error in stop service");
}
{

private void btnInitializeNevisa_Click(object sender, EventArgs e)
{
 bool serverStatus= true;
 bool lockStatus = false;
 //if (SimpleNevisaInterface.TestServer())
 // serverStatus = true;
 if (SimpleNevisaInterface.IsClientLockExists())
 lockStatus = true;
 if (serverStatus && lockStatus)
 {
 SimpleNevisaInterface.Initialize(NevisaPath);
 MessageBox.Show("Initialize Done!");
 }
 else
 MessageBox.Show("Server Status: " + serverStatus + "\nLock Status: " +
lockStatus);
}

private void btnShowNevisaForm_Click(object sender, EventArgs e)
{
 SimpleNevisaInterface.ShowNevisaForm();
}

private void btnStartRecognition_Click(object sender, EventArgs e)
{
 SimpleNevisaInterface.StartRecognition();
 txtNevisaRecognizedText.Text = "";
 NevisaInterface.NewSpeechHypothesis += NevisaInterface_NewSpeechHypothesis;
 NevisaInterface.DoTransferHypothesis += NevisaInterface_DoTransferHypothesis;
}

void NevisaInterface_DoTransferHypothesis()
{
 if (PrevuiosRecognizedText == AllRecognizedText)
 return;
 string Result = AllRecognizedText;
}

```

```

 if (ProcessRecognizedText)
 Result = RemoveStopWords(Result);
 txtNevisaRecognizedText.Text = Result;
 if ((IsClientConnect))
 {
 PreviuosRecognizedText = Result;
 byte[] RecognizedText = Encoding.Default.GetBytes(Result);
 //string s = Encoding.UTF8.GetString(RecognizedText);
 rtxtSendData.AppendText(Result + "\r\n");
 rtxtReceivedData.ScrollToCaret();
 Client.Send(RecognizedText);
 }
 }

 void NevisaInterface_NewSpeechHypothesis(string hypothesis)
 {
 AllRecognizedText = hypothesis;
 }

 private void InitializeStemmingTools()
 {
 new
 DependencyBasedSentenceAnalyzer.SentenceAnalyzer(@"D:\Nevisa01\Nevisa01\Res\VerbList.txt");
 string RootWordsPath = @"D:\Nevisa01\Nevisa01\Res\root words.txt"; //root words
 st = new StemmingTools(RootWordsPath);
 stoptexts = System.IO.File.ReadAllText(@"D:\Nevisa01\Nevisa01\StopWord.txt");
 beforeWords = stoptexts.ToLower().Split(splitChars).ToList<string>();
 stop = beforeWords.ToArray();
 }

 private static bool isStopWord(String s)
 {
 if (s.Trim().Length < 3 || stop.Contains(s.Trim()) || s.Contains('1') || s.Contains('2') ||
 s.Contains('3') || s.Contains('4') || s.Contains('5') || s.Contains('6') || s.Contains('7') ||
 s.Contains('8') || s.Contains('9') || s.Contains('0'))
 {
 return true;
 }
 else
 {
 return false;
 }
 }

 private string RemoveStopWords(string Result)
 {
 string RawText = Result;
 char[] splitChars = { ' ' };
 }

```

```

List<string> AfterWords = RawText.ToLower().Split(splitChars).ToList<string>();
AfterWords.RemoveAll(isStopWord);
string AfterText = string.Join(" ", AfterWords.ToArray());
string AfterText2 = "";
try
{
 AfterText2 = st.LemmatizeText(AfterText, LevelOfLemmatizer.First).ToString();
}
catch (Exception exp)
{
 MessageBox.Show(exp.Message);
}
AfterText2 = st.LemmatizeText(AfterText, LevelOfLemmatizer.First).ToString();
AfterWords = AfterText2.Split(splitChars).ToList<string>();
string ProcessedText = string.Empty;
foreach (var item in AfterWords)
{
 if (item.Contains('+'))
 {
 int int1 = item.IndexOf('+');
 ProcessedText = ProcessedText + " " + item.Substring(0, int1);
 }
 else
 {
 ProcessedText = ProcessedText + " " + item;
 }
}
return AfterText;
}

private void btnStopRecognition_Click(object sender, EventArgs e)
{
 SimpleNevisaInterface.StopRecognition();
}

private void btnInterfaceStartServer_Click(object sender, EventArgs e)
{
 RobotInteractionThread = new Thread(RobotoInteraction) { IsBackground=true };
 RobotInteractionThread.Start();
}

private void RobotoInteraction()
{
 IPAddress ipAddress = IPAddress.Parse(txtInterfaceIP.Text);
 int portNumber = int.Parse(txtInterfacePort.Text);
 IPEndPoint localEndPoint = new IPEndPoint(ipAddress, portNumber);
 Listener = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
ProtocolType.Tcp);
 Listener.Bind(localEndPoint);
 if (Listener.IsBound)

```

```

 }
 Listener.Listen(MaxConnections);
 Client = Listener.Accept();
 ReceivedBuffer = new byte[MessageLength];
 AsyncCallback callback = new AsyncCallback(OnReceivedData);
 Client.BeginReceive(ReceivedBuffer, 0, MessageLength, SocketFlags.None,
callback, null);
 IsClientConnect = true;
 {

 {

private void OnReceivedData(IAsyncResult ar)
}
 //rtxtReceivedData.Invoke((MethodInvoker)delegate
 }//
 // rtxtReceivedData.AppendText("ReceivedData\r\n");
 // rtxtReceivedData.ScrollToCaret();

 ;{//
 //SocketError errorCode;
 //int nBytesRec = Client.EndReceive(ar, out errorCode);
 //if (errorCode != SocketError.Success)
 }//
 // nBytesRec = 0;
 // Client.Close();
 // Listener.Close();
 // return;
 {//

 int TextLength = 0;
 for (int i = 0; i < ReceivedBuffer.Length; i++)
 {
 if (ReceivedBuffer[i] != 0)
 TextLength++;
 }
 byte[] Text = new byte[TextLength];
 Array.Copy(ReceivedBuffer, 0, Text, 0, TextLength);
 string Message = Encoding.ASCII.GetString(Text);
 Message = Message.Trim();
 if (Message == "startt")
 {
 ProcessRecognizedText = false;
 SimpleNevisaInterface.StartRecognition();
 NevisaInterface.NewSpeechHypothesis +=
NevisaInterface_NewSpeechHypothesis;
 NevisaInterface.DoTransferHypothesis+=NevisaInterface_DoTransferHypothesis;
 }
 else if (Message == "stoppp")

```

```

 }
 if (ProcessRecognizedText)
 ProcessRecognizedText = false;
 SimpleNevisaInterface.StopRecognition();
 {
 else if (Message == "search")
 }
 SimpleNevisaInterface.StartRecognition();
 ProcessRecognizedText = true;
 {
 rtxtReceivedData.Invoke((MethodInvoker)delegate
 {
 rtxtReceivedData.AppendText(Message + "\r\n");
 rtxtReceivedData.ScrollToCaret();
 };
 AsyncCallback callback = new AsyncCallback(OnReceivedData);
 Client.BeginReceive(ReceivedBuffer, 0, MessageLength, SocketFlags.None,
callback, null);
 {

private void btnInterfaceStopServer_Click(object sender, EventArgs e)
{
 //Listener.Disconnect(true);
 //Listener.Shutdown(SocketShutdown.Receive);
 Listener.Close();
 {

private void btnSpeakerTraining_Click(object sender, EventArgs e)
{
 SimpleNevisaInterface.ShowSpeakerTraining();
 {

private void btnClear_Click(object sender, EventArgs e)
{
 txtNevisaRecognizedText.Text = "";
 {

private void btnUsers_Click(object sender, EventArgs e)
{
 SimpleNevisaInterface.ShowManageUsers();
 {

private void btnVocabulary_Click(object sender, EventArgs e)
{
 SimpleNevisaInterface.ShowManageVocabulary();
 {

```

```
private void btnSetting_Click(object sender, EventArgs e)
{
 SimpleNevisaInterface.ShowSettings();
}

private void btnAudioSetting_Click(object sender, EventArgs e)
{
 SimpleNevisaInterface.ShowAudioSetup();
}
{
{
```



## Abstract

A researcher usually undertakes different types of activities for conducting a research, such as identifying and analyzing a research problem, generating ideas for a possible solution to the problem, gathering and organizing proper research data, analyzing data, and preparing and publishing technical research report. Conducting a research becomes a challenging task when a large number of scientific data is available and needs to be organized. While, many scientific data management tools are available, they are not usually accessible through an integrated manner. Moreover, learning the tools and integrating them requires additional efforts, while sometimes the tools may not be compatible with each other. Also in the case of access to hi-tech facilities by researchers, some advancement has been made, e.g., to help surgeons operating by tele-surgery, but these solutions have not been extended to other conventional research environments and disciplines. In such cases, the availability of an integrated and intelligent tool or device for research assistantship would be beneficial for researchers. Such assistant should be tailored and personalized to researcher's need and expertise .

In this project, we propose to tackle the challenge of having a personalized research assistant in a more unified and human-friendly manner by having social robots that can communicate with researchers and assist them in addressing their research needs. Having a physical embodiment, such a robot can be considered an effective, credible and informative interactive partner. Among the computerized or physical personalized intelligent assistant, social robots are identified as one of the most influential and beneficial choices. Social robots are intelligent self-control robots that can interact with human naturally and in a social way.

In this project, we present a conceptual framework for a social research assistant robot, that we name it as RoboDoc. This robot is aimed to provide assistance to novice and experienced researchers to plan and organize various research stages. It is also envisioned to provide researchers with remote access to materials and facilities those are not easily accessible. The project aims to create synergy among various scientific disciplines including artificial linguistics, computational linguistics, computer science, and robotics.

Developing RoboDoc is a long-term project with various phases. This technical report is about the first phase of developing RoboDoc in which a set of services that this robot can offer for a researcher is developed. As we are not going to develop a new robotic platform at this phase of the project, we adopt Nao platform as the preliminary platform to develop RoboDoc services. However, due to the requirements at the first phase we have customized the physical appearance of Nao and add some devices to the body to customize Nao for RoboDoc. RoboDoc can interact with the user through voice and video. RoboDoc can recognize and talk to the user. It has Persian and English speech recognition module, and gesture and face detection module for interacting with the user. The user can provide a query for RoboDoc, and the robot can search for it in Ganj (Persian scientific database), and Google scholar, organize and visualize the search results. RoboDoc can also read a Persian and English text for the user. All of these services are integrated through a unified robotic platform.

At the end of this report a set of projects are also defined and prioritized for developing RoboDoc in the future.

**Keywords:** Social robotics, Speech processing and recognition, Face recognition, Information retrieval, Data visualization



**Iranian Research Institute  
for Information Science and Technology  
(I r a n D o c)**

**First Phase of  
Developing RoboDoc:  
A new Social Robot for Research Assistanship**

**Principal Investigator:  
Azadeh Mohebi**

**Co-Investigators:  
Soheil Ganjefar, Ramin Golshaie, Ammar Jalalimanesh**

**February 2017**