



دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکده مهندسی کامپیووتر

پایان نامه کارشناسی

پیاده‌سازی قفل هوشمند با استفاده از روش شناسایی چهره مبتنی بر هوش مصنوعی

نگارنده

امیرحسین علی‌بخشی

استاد راهنما

دکتر حمیدرضا زرندی

۱۴۰۱



تعهدنامه اصالت اثر

این‌جانب امیرحسین علی‌بخشی متعهد می‌شوم که مطالب مندرج در این پایان‌نامه حاصل کار پژوهشی این‌جانب تحت نظارت و راهنمایی استادی دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آن‌ها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مأخذ ذکر گردیده است. این پایان‌نامه قبلًا برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان‌نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان‌نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مأخذ بلامانع است.

امیرحسین علی‌بخشی

امضا

سپاس‌گزاری

در اینجا لازم می‌دانم که از استاد گرانقدر، جناب آقای دکتر حمیدرضا زرندی، که بدون راهنمایی‌های ارزشمند ایشان انجام این پروژه میسر نمی‌شد، نهایت تشکر را داشته باشم. همچنین از استاد گرانمایه، جناب آقای دکتر مسعود صبایی، که زحمت داوری این پایان‌نامه را برعهده داشتند، نهایت تشکر را دارم. از جناب آقای کیخا برای تمامی راهنمایی‌ها و زحماتشان سپاس‌گزاری می‌کنم.

امیرحسین علی‌بخشی

۱۴۰۱ فروردین

چکیده

احراز هویت^۱ یکی از مباحث کلیدی و چالش برانگیز تلقی می‌شود؛ از آن جا که در روش‌هایی مانند استفاده از رمز عبور^۲ ممکن است رمز به هر دلیلی فاش شود، یا در روش استفاده از کارت‌های شناسایی^۳ امکان به سرقت رفتن کارت و استفاده ای یک فرد غیرمجاز از آن وجود دارد. روش‌های بیومتریک^۴ در راستای احراز هویت مطرح شدند که یکی از نمونه‌های معروف آن شناسایی چهره می‌باشد. می‌توان به کمک الگوریتم‌های مبتنی بر هوش مصنوعی، چهره‌ها را در عکس‌ها تشخیص داد و شناسایی کرد که عکس، متعلق به چه شخصی می‌باشد. در این پروژه هدف تشخیص چهره با استفاده از شبکه‌های عصبی پیچشی^۵ یا هیستوگرام شبکه‌های جهت‌دار^۶ و سپس شناسایی چهره تشخیص داده شده به کمک بردارسازی برای قفل درب ورودی یک مکان می‌باشد و بر اساس این‌که چهره از قبل در مجموعه چهره‌های ذخیره شده در سامانه وجود دارد یا خیر، اجازه‌ی ورود شخص به آن مکان را صادر می‌شود. نحوه پیاده‌سازی به گونه‌ای خواهد بود که پروژه هم در رایانه‌های شخصی و هم روی بورد Raspberry Pi قابل اجرا باشد. کاربرانی که در این سامانه نقش "مدیر" دارند، قادر به افزودن کاربر جدید به سامانه یا حذف آن‌ها خواهند بود. در پایان، به کمک مجموعه داده‌ی "چهره‌های برچسب‌دار دنیای واقعی"، ۶۵۷۷ تصویر از ۱۶۸۰ شخص متمایز مورد شناسایی سامانه قرار گرفت و طی این ارزیابی، صحت^۷ ۹۶.۵۰٪ درصد، دقت^۸ ۹۸.۲۱٪ درصد و امتیاز F1^۹ برابر با ۹۹.۹۲٪ درصد محاسبه شدند.

واژه‌های کلیدی:

احراز هویت، تشخیص چهره، شناسایی چهره، شبکه‌های عصبی پیچشی، الگوریتم هیستوگرام شبکه‌های جهت‌دار

¹ Authentication

² Password

³ ID Cards

⁴ Biometric

⁵ Convolutional Neural Networks

⁶ Histogram of Oriented Gradients

⁷ Accuracy

⁸ Precision

⁹ F1 score

فهرست مطالب

صفحه	عنوان
۱.....	۱ - مقدمه
۴.....	۲ - معرفی مفاهیم پایه و ابزارهای مورد استفاده در پروژه
۵.....	۲ - ۱ معرفی مفاهیم پایه
۵.....	۲ - ۱ - ۱ تشخیص چهره
۵.....	۲ - ۱ - ۲ شناسایی چهره
۵.....	۲ - ۱ - ۳ شبکه‌های عصبی مصنوعی
۱۰.....	۲ - ۱ - ۴ معرفی دو الگوریتم مشهور تشخیص چهره
۱۹.....	۲ - ۲ معرفی ابزارها و فناوری‌های استفاده شده در پروژه
۱۹.....	۲ - ۲ - ۱ ابزارهای سخت‌افزاری
۲۵.....	۲ - ۲ - ۲ ابزارهای نرم‌افزاری
۳۰.....	۳ - طراحی سامانه
۳۱.....	۳ - ۱ معرفی مفهوم کاربر
۳۱.....	۳ - ۱ - ۱ شناسه کاربری
۳۱.....	۳ - ۱ - ۲ نام کاربر
۳۱.....	۳ - ۱ - ۳ عکس‌های ذخیره شده از هر کاربر
۳۲.....	۳ - ۱ - ۴ نامگذاری فایل‌های عکس
۳۲.....	۳ - ۲ طراحی معماری سامانه
۳۳.....	۳ - ۲ - ۱ مجموعه داده‌ها
۳۳.....	۳ - ۲ - ۲ مازول کنترلی سامانه
۳۳.....	۳ - ۲ - ۳ مازول تشخیص چهره
۳۴.....	۳ - ۲ - ۴ مازول شناسایی چهره

۳۹.....	۲ - ۵ - طراحی جریان اجرایی برنامه
۴۷.....	۴ - پیاده‌سازی سامانه
۴۸.....	۴ - ۱ - توسعه بخش نرم‌افزاری
۴۸.....	۴ - ۱ - ۱ - ساختار فایل‌های کد پروژه
۴۸.....	۴ - ۱ - ۲ - پیاده‌سازی کد تشخیص چهره
۴۹.....	۴ - ۱ - ۳ - پیاده‌سازی کد شناسایی چهره
۵۱.....	۴ - ۱ - ۴ - پیاده‌سازی جریان اجرایی برنامه
۵۶.....	۴ - ۱ - ۵ - ثبت وقایع برنامه به کمک ذخیره رخدادها
۵۸.....	۴ - ۱ - ۶ - کنترل سامانه به کمک فایل پیکربندی
Error! Bookmark not defined.....	۴ - ۱ - ۷ - توسعه تابلوی کنترلی مدیر
۵۹.....	۴ - ۱ - ۸ - توسعه تابلوی کنترلی مدیر
۶۳.....	۴ - ۱ - ۹ - اجرای سامانه در حالت اشکال‌زدایی
۶۴.....	۴ - ۱ - ۱۰ - کنترل ماظول‌های سخت‌افزاری در کد پروژه
۶۴.....	۴ - ۲ - پیاده‌سازی مدار پروژه
۶۵.....	۴ - ۲ - ۱ - راه اندازی بورد Raspberry Pi
۷۱.....	۴ - ۲ - ۲ - اجرای کد پروژه روی بورد Raspberry Pi
۷۳.....	۵ - ارزیابی پروژه
۷۴.....	۵ - استفاده از تحلیل مولفه‌های اصلی برای بررسی کیفیت شناسایی چهره
۷۵.....	۵ - تاثیر پیش‌پردازش پیش از بردارسازی در زمان تمرین مدل شناسایی چهره
۷۶.....	۵ - ۳ - مقایسه الگوریتم‌های تشخیص چهره
۷۶.....	۵ - ۴ - بررسی دقت شناسایی چهره‌ها
۷۸.....	۵ - ۵ - تاثیر مقدار ۷ مدت زمان هر قاب
۷۹.....	۶ - جمع‌بندی، نتیجه‌گیری و پیشنهادها

۸۰	۱ - جمع‌بندی و نتیجه‌گیری
۸۰	۲ - پیشنهادها
۸۲	۳ - مراجع

فهرست شکل‌ها

عنوان	صفحه
شکل ۲ - ۱ ساختار یک Perceptron	۸
شکل ۲ - ۲ نمونه‌ای از یک شبکه عصبی با دو لایه میانی	۹
شکل ۲ - ۳ شبکه کد الگوریتم تمرین شبکه عصبی	۱۰
شکل ۲ - ۴ تقسیم رنگ به سه کanal رنگی قرمز، سبز و آبی	۱۲
شکل ۲ - ۵ نحوه عملکرد لایه پیچشی برای یک کanal رنگی	۱۲
شکل ۲ - ۶ استفاده از ماتریس هسته سه بعدی برای محاسبه ماتریس پیچشی	۱۲
شکل ۲ - ۷ مقایسه ادغام حداکثری و ادغام میانگین	۱۳
شکل ۲ - ۸ مدل کلی یک شبکه عصبی پیچشی	۱۴
شکل ۲ - ۹ پیش‌پردازش در الگوریتم هیستوگرام شبکه‌های جهت‌دار	۱۵
شکل ۲ - ۱۰ مقادیر اندازه و جهت شبکه برای هر سلول	۱۷
شکل ۲ - ۱۱ تشریح شیوه ساخت هیستوگرام یک سلول	۱۷
شکل ۲ - ۱۲ هیستوگرام نهایی یک سلول	۱۸
شکل ۲ - ۱۳ نرمال سازی بلوک‌ها در هیستوگرام	۱۸
شکل ۲ - ۱۴ خروجی الگوریتم هیستوگرام شبکه‌های جهت‌دار	۱۹
شکل ۲ - ۱۵ تصویر بورد Raspberry Pi	۲۰
شکل ۲ - ۱۶ پایه‌های بورد Raspberry Pi	۲۱
شکل ۲ - ۱۷ ولتاژ پایه‌های Raspberry Pi	۲۲
شکل ۲ - ۱۸ مازول دوربین Raspberry Pi	۲۲
شکل ۲ - ۱۹ قفل Solenoid	۲۳
شکل ۲ - ۲۰ صفحه کلید عددی	۲۳
شکل ۲ - ۲۱ دیود نشده نور	۲۴
شکل ۲ - ۲۲ آژیر	۲۴

شکل ۲-۲۳ مژول Sim900	۲۵
شکل ۲-۲۴ نماد Python	۲۶
شکل ۲-۲۵ نماد OpenCV	۲۶
شکل ۲-۲۶ نماد Dlib	۲۷
شکل ۲-۲۷ نماد NumPy	۲۷
شکل ۲-۲۸ نماد matplotlib	۲۸
شکل ۲-۲۹ نماد Flask	۲۸
شکل ۲-۳۰ نماد PyCharm	۲۹
شکل ۳-۱ معماری سامانه	۳۲
شکل ۳-۲ ساختار مژول تشخیص چهره	۳۴
شکل ۳-۳ خروجی مژول تشخیص چهره برای عکس ورودی حاوی دو چهره	۳۴
شکل ۳-۴ پیش‌پردازش عکس‌های سامانه	۳۵
شکل ۳-۵ فرایند بردارسازی چهره‌ها	۳۶
شکل ۳-۶ شیوه محاسبه بردار به شرط $j = 5$	۳۶
شکل ۳-۷ بردارسازی تمام عکس‌های سامانه	۳۷
شکل ۳-۸ خلاصه فرایند شناسایی چهره	۳۹
شکل ۳-۹ نگاشت حروف انگلیسی در صفحه کلید عددی برای وارد کردن متن در سامانه	۴۳
شکل ۳-۱۰ نمودار جریان اجرایی برنامه	۴۵
شکل ۴-۱ کد بردارسازی چهره‌ها	۵۰
شکل ۴-۲ کد مقایسه چهره‌ها	۵۰
شکل ۴-۳ محاسبه فاصله میان چهره‌ها در کد	۵۱
شکل ۴-۴ تصویر ترمینال شروع اجرای پروژه	۵۲
شکل ۴-۵ تصویری از پروژه در صفحه اصلی	۵۲
شکل ۴-۶ تصویر سامانه در حالتی که درب با موفقیت باز شده است	۵۳
شکل ۴-۷ تصویری از سامانه که درب به دلیل ناشناس بودن چهره باز نمی‌شود	۵۳

شکل ۴-۸ تصویر سامانه در حالتی که درب به علت عدم تشخیص چهره باز نمی‌شود ۵۴
شکل ۴-۹ حالات مختلف وارد کردن شناسه مدیر. سمت راست: موفق، سمت چپ: ناموفق ۵۴
شکل ۴-۱۰ حالت وارد کردن رمز عبور اشتباه ۵۵
شکل ۴-۱۱ عکسبرداری از کاربر جدید ۵۵
شکل ۴-۱۲ تصویر نمایش داده شده از سامانه در حالت تمرین ۵۶
شکل ۴-۱۳ نمونه‌ای از رخدادهای سامانه ۵۷
شکل ۴-۱۴ نمایی از سایت EpochConvertoer در حال نشان داده "۱۶۷۶۶۲۸۹۵۴" timestamp ۵۸
شکل ۴-۱۵ صفحه اصلی تابلوی کنترلی مدیر ۶۱
شکل ۴-۱۶ صفحه رخدادها در تابلوی کنترلی مدیر ۶۲
شکل ۴-۱۷ صفحه شاهده اطلاعات کاربران در تابلوی کنترلی مدیر ۶۲
شکل ۴-۱۸ مشاهده اطلاعات سامانه در تابلوی کنترلی مدیر ۶۳
شکل ۴-۱۹ مشاهده تاثیر افزایش ۷ بر مدت زمان هر قاب در حالت اشکال‌زدایی ۶۴
شکل ۴-۲۰ کد مربوط به کنترل آژیر ۶۵
شکل ۴-۲۱ تصویر وبسایت مربوط به راهاندازی سیستم‌عامل Raspbian ۶۶
شکل ۴-۲۲ مراحل نصب سیستم‌عامل بر روی Raspberry Pi ۶۷
شکل ۴-۲۳ تصویر سیستم‌عامل Raspberry Pi روی بورد Raspbian ۶۸
شکل ۴-۲۴ اتصالات اولیه بورد ۶۸
شکل ۴-۲۵ بخشی از توضیحات تکمیلی مربوط به اجرای پروژه Raspberry Pi روی بورد ۶۹
شکل ۴-۲۶ شیوه اتصال دوربین ۶۹
شکل ۴-۲۷ نحوه اتصال صفحه کلید عددی ۷۰
شکل ۴-۲۸ نحوه اتصال آژیر ۷۰
شکل ۴-۲۹ نحوه اتصال به دیود نشردنه نور ۷۱
شکل ۴-۳۰ شمای مدار پروژه ۷۲
شکل ۴-۳۱ تصویری از اجرای پروژه Raspberry Pi روی بورد ۷۲
شکل ۵-۱ تحلیل مولفه‌های اصلی چهره‌ها در دو و سه بعد ۷۴

- شکل ۵-۲ تاثیر پیش‌پردازش عکس‌ها بر روی زمان تمرین مدل شناسایی چهره ۷۵
- شکل ۵-۳ تعدادی از چهره‌های یک شخص در مجموعه داده "چهره‌های برگسبدار در دنیای واقعی" ۷۷
- شکل ۵-۴ میانگین زمان ۲۰ قاب متوالی را برای ۷ با مقادیر ۱ تا ۵ ۷۸

فهرست جدول‌ها

عنوان	صفحه
جدول ۲- ۱- چند نمونه تابع فعال‌سازی	۷
جدول ۲- ۲- مشخصات فنی بورد Raspberry Pi	۲۱
جدول ۳- ۱- حالات نمایش چهره‌های شناسایی شده در نمایشگر	۴۲
جدول ۳- ۲- نحوه جابجایی میان صفحه‌ها سامانه	۴۶
جدول ۴- ۱- توضیح برخی از فایل‌ها و پوشش‌های مهم پروژه	۴۸
جدول ۴- ۲- توضیحات فایل پیکربندی تنظیمات	۶۰
جدول ۴- ۳- نحوه اتصال قطعات به بورد	۶۶
جدول ۵- ۱- ماتریس سردرگمی	۷۷

١ - مقدمه

یکی از روش‌های احراز هویت که روز به روز در حال پیشرفت است، احراز هویت بر اساس چهره انسان می‌باشد. مقالات زیادی در حوزه پردازش تصویر و تشخیص چهره بیان شده‌اند. به عنوان مثال در [۱] به کمک متلب^۱ از شبکه‌های عصبی برای فرایند تشخیص چهره استفاده شده است. علاوه بر آن، پژوهشگران در [۲] به کمک تشخیص چهره‌ی عمیق متراکم^۲ آن را پیاده‌سازی کرده‌اند که نیازشان به استفاده از ماشین بردار پشتیبان^۳، بخش‌بندی^۴ و لزوم عکس‌برداری از زوایای مختلف را رفع می‌کند. در [۳] از شبکه‌های چندلایه‌ی Perceptron^۵ استفاده شده است و در [۴] به کمک چند شبکه‌ی عصبی مستقل از یکدیگر که با عکس‌های مجزا آموزش داده شده‌اند فرایند تشخیص چهره انجام می‌شود. پژوهشگران در [۵] از مدل ترکیبی شبکه‌های عصبی و Adaboost و همچنین در [۶] پیش از استفاده از شبکه عصبی انتشار معکوس^۶، از تجزیه و تحلیل مولفه‌های اصلی^۷ استفاده می‌کنند.

حال ممکن است هر کدام دارای نقایصی باشند؛ به عنوان مثال در [۴]-[۱] می‌توان عمل کاهش بعد^۸ را داشت که برای نمونه می‌توان به روش تجزیه و تحلیل مؤلفه‌های اصلی^۹ اشاره کرد. همچنین برخی مقالات مانند [۶]-[۳] به امکان وجود زوایای مختلف یک تصویر ورودی مربوط به یک چهره واحد توجهی نشان نداده و به کلی حرفي از این موضوع بیان نکرده‌اند.

هدف این پژوهه، پیاده‌سازی یک سامانه‌ی شناسایی چهره مبتنی بر الگوریتم‌های هوش می‌باشد، به طوری که بتواند به صورت مستقل از سکو^{۱۰}، هم روی رایانه‌های شخصی و هم بوردهای Raspberry Pi اجرا شود. معماری این پژوهه به صورت مازولار^{۱۱} بوده و هر مازول^{۱۲} به صورت مستقل از سایر مازول‌ها می‌تواند فعالیت کند. پس از طراحی جریان اجرایی برنامه و پیاده‌سازی آن به کمک زبان Python، سامانه آماده‌ی استفاده و ارزیابی می‌شود. بار ارزیابی آن، دقت ۹۶.۴۸۷ درصد بدست می‌آید که برای چنین سامانه‌هایی، قابل قبول است.

در ادامه، در فصل دوم به بررسی مفاهیم پایه پژوهه و ابزار و فناوری‌های استفاده شده در توسعه‌ی، در فصل سوم

¹ MATLAB

² Deep Dense Face Detector

³ Support Vector Machine (SVM)

⁴ Segmentation

⁵ Multi-Layer Perceptron

⁶ Back-Propagation Neural Network

⁷ Principal Component Analysis (PCA)

⁸ Dimensionality reduction

⁹ Principal Component Analysis (PCA)

¹⁰ Platform

¹¹ Modular

¹² Module

به معرفی طرح ارائه شده برای پیاده‌سازی سامانه و در فصل چهارم به مراحل پیاده‌سازی پروژه پرداخته می‌شود. در فصل پنجم آزمایش‌های انجام داده شده شرح داده می‌شوند و نتایج بدست آمده از این آزمایش‌ها تحلیل می‌شوند. در نهایت در فصل ششم جمع‌بندی مطالب بیان شده صورت می‌پذیرد و پیشنهادهایی برای کارهایی که در آینده می‌توانند در این زمینه انجام شوند ارائه شده است.

۲ - معرفی مفاهیم پایه و ابزارهای مورد استفاده در پروژه

معرفی مفاهیم پایه و ابزارهای مورد استفاده در پروژه

پیش از شرح پروژه، به معرفی مفاهیم استفاده شده در پروژه و سپس به ابزارهایی که در پیاده‌سازی پروژه از آن‌ها استفاده می‌شود اشاره می‌کنیم.

۲ - ۱ معرفی مفاهیم پایه

در این بخش، مفاهیمی که در پروژه از آن‌ها استفاده شده است را معرفی می‌کنیم.

۲ - ۱ - ۱ تشخیص چهره

تشخیص چهره یک نوع الگوریتم تشخیص شیء^۱ محسوب می‌شود که مشخص می‌کند در یک قاب^۲ تصویر یا ویدیو، چهره انسان وجود دارد یا خیر. در این صورت فرقی نمی‌کند که چهره متعلق به چه کسی باشد و فقط وجود آن بررسی می‌شود. این الگوریتم مکان چهره‌های موجود در تصویر ورودی را در خروجی خود باز می‌گرداند .[۷]

۲ - ۱ - ۲ شناسایی چهره

شناسایی چهره الگوریتمی است که امکان شناسایی چهره انسان را بر اساس شاخص‌های بیومتریک یکتای موجود در چهره افراد فراهم می‌کند. این الگوریتم در ابتدا نیازمند یک الگوریتم تشخیص چهره است؛ در واقع خود الگوریتم شناسایی چهره، چهره‌های تصویر را پیدا نمی‌کند، بلکه مشخص می‌کند یک چهره متعلق به چه کسی است. در نتیجه باید از قبل با تعدادی چهره آشنا شده باشد تا بتواند هویت یک چهره را شناسایی کند [۸].

۲ - ۱ - ۳ شبکه‌های عصبی مصنوعی

الگوریتم‌های هوش مصنوعی بسیاری از مدلی به نام شبکه‌های عصبی مصنوعی استفاده می‌کنند تا بر اساس تجزیه و تحلیل داده‌ها، برای گرفتن تصمیم خاصی، رفتار انسان را تقلید کنند. طرح این ساختار لایه‌ای، برگرفته از

¹ Object Detection

² Frame

ساختار مغز انسان است. همان‌طور که مغز انسان به شناسایی الگوهای مختلف داده‌ها و دسته‌بندی انواع اطلاعات می‌پردازد، می‌توان شبکه‌های عصبی را به شیوه‌ای مشابه با رفتار مغز انسان آموزش داد تا به تشخیص الگوها پرداخته و دسته‌بندی داده‌ها به صورت با نظارت^۱ را انجام دهد.

زمانی که مغز انسان با اطلاعات جدیدی روبرو می‌شود، در تلاش است که این اطلاعات را با مفاهیم قبلی و شناخته‌شده خود مقایسه کند تا به درک بهتر از اطلاعات جدید برسد. هدف شبکه‌های عصبی نیز تشخیص الگوها و دسته‌بندی اطلاعات جدید بر پایه دانش قبلی خود است. در این مدل، یادگیری آنی الگوها در قالب بردارهای عددی انجام می‌شود؛ به عبارتی، تمامی داده‌های دنیای واقعی مانند تصاویر، صوت و متن باید به بردارهای عددی تبدیل شوند و به عنوان ورودی، در اختیار شبکه عصبی قرار گیرند تا مدل هوش مصنوعی بتواند آن‌ها را درک کند [۹].

۲ - ۱ - ۳ - ۱ - مفهوم Perceptron یا گره

کوچک‌ترین واحد سازنده یک شبکه عصبی، Perceptron یا گره می‌باشد که رفتاری شبیه به رفتار نورون‌های مغز انسان دارد. هر گره داده‌های ورودی را دریافت می‌کند و با اعمال محاسبات بر روی آن‌ها، مقداری را در خروجی برمی‌گرداند. ساختار درونی گره شامل اجزای مختلفی به شرح زیر است:

۲ - ۱ - ۳ - ۱ - ۱ - ورودی گره

در لایه نخست شبکه عصبی، ورودی گره‌ها در قالب بردارهای عددی با طول یکسان هستند؛ به عنوان مثال، در مسئله تشخیص اشیاء، ورودی گره‌های لایه نخست، لیستی از مقادیر پیکسل^۲‌های تصاویر است. ورودی گره‌ها در لایه‌های پنهان و لایه‌های خروجی، برابر مقادیر خروجی حاصل از لایه‌های قبلی آن‌هاست [۹].

¹ Supervised

² Neuron

³ Pixel

۱ - ۲ - ۳ - وزن‌های ورودی گره

وزن‌ها میزان اهمیت ورودی‌های گره‌ها را مشخص می‌کنند. به عبارتی، با اعمال ضرب داخلی^۱ مقدار ورودی گره و ماتریس وزن، میزان اهمیت هر ویژگی ورودی مشخص می‌شود. در هر مرحله از روال پس‌انتشار^۲، مقادیر وزن‌ها به روزرسانی شده تا مقدار نهایی تابع زیان^۳ کاهش پیدا کند [۹].

۱ - ۲ - ۳ - تابع فعال‌سازی

ورودی تابع فعال‌سازی^۴، ترکیب خطی مقادیر ورودی، وزن‌ها و مقادیر اریب^۵ است. وظیفه تابع فعال‌سازی در شبکه‌های عصبی، تبدیل ترکیب خطی ورودی‌ها به غیرخطی است و مقادیر ورودی را براساس نوع تابع فعال‌سازی به فضا با بازه مشخصی نگاشت می‌کند. به بیان دیگر، این توابع تعیین می‌کنند آیا خروجی گره برای شبکه عصبی اهمیت دارد یا باید آن را نادیده گرفت. در جدول ۲-۱، چند نمونه تابع فعال‌سازی معروف آورده شده است [۹].

جدول ۲ - چند نمونه تابع فعال‌سازی [۹]

محدوده	رابطه ریاضی	نمودار	نام تابع فعال‌سازی
$(-\infty, +\infty)$	x		همانی
$\{0, 1\}$	$\begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x \geq 0 \end{cases}$		پله واحد
$(0, 1)$	$\sigma(x) = \frac{1}{1 + e^{-x}}$		سیگموئید ^۶
$(-1, 1)$	$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$		تانژانت هایپرboleیک
$[0, +\infty)$	$\max(0, x)$		ReLU ^۷

¹ Inner Product

² Backpropagation

³ Loss Function

⁴ Activation Function

⁵ Bias

⁶ Sigmoid

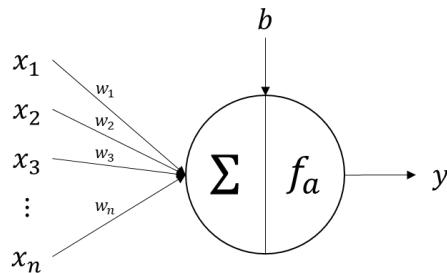
⁷ ReLU

۱-۲-۳-۴- اریب گره

نقش اریب در شبکه عصبی آن است که خروجی نهایی تابع فعال سازی را تغییر دهد. نقش اریب مشابه نقش مقدار ثابت در تابع خطی است و مقدار نهایی تابع فعال سازی را در فضای برداری به چپ و راست منتقل می‌کند تا تابع نهایی بر روی داده‌ها بهتر منطبق شود و مدل در نهایت پیش‌بینی دقیق‌تری داشته باشد.

ساختار کلی یک Perceptron در شکل ۲-۱ قابل مشاهده می‌باشد که در آن b اریب، x_i ورودی i ام، w_i وزن ورودی i ام، f_a تابع فعال سازی و y برابر خروجی Perceptron می‌باشد. فرمول Perceptron نیز در رابطه ۱-۲ آورده شده است [۹].

$$y = f_a \left(\sum w_i x_i + b \right) \quad (1-2)$$



شکل ۲-۱ ساختار یک Perceptron

۱-۳-۲-۳-۱- انواع لایه‌ها در شبکه‌های عصبی

هر لایه شبکه عصبی شامل مجموعه‌ای از Perceptron ها است. در ادامه انواع لایه‌های یک شبکه عصبی معرفی شده‌اند.

۱-۳-۲-۳-۱- لایه ورودی

هر شبکه عصبی دارای یک لایه ورودی است که داده‌های دنیای بیرون را دریافت می‌کند و به منظور اعمال محاسبات مختلف بر روی داده‌ها، آن‌ها را به لایه بعد خود می‌فرستد. به عبارتی، این لایه هیچ گونه عملیات محاسباتی و پردازشی بر روی داده‌ها اعمال نمی‌کند و صرفاً مسئولیت دریافت داده‌ها و ارسال آن‌ها را به لایه‌های بعدی بر عهده دارد. داده‌های ورودی لایه نخست، می‌توانند شامل اطلاعات متنی، صوتی و تصویری باشد [۹].

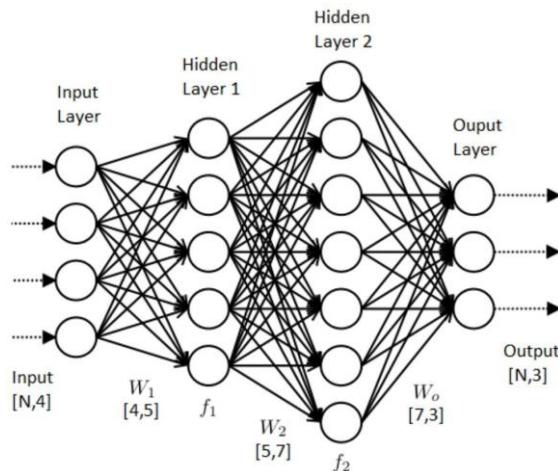
۱-۲-۳-۲- لایه پنهان

لایه‌ای که در میان لایه ورودی و لایه خروجی قرار می‌گیرد، لایه پنهان یا لایه میانی نام دارد. حداقل تعداد لایه‌های میانی شبکه عصبی، یک لایه است و بر اساس پیچیدگی و نوع مسئله می‌توان به تعداد لایه‌های میانی شبکه اضافه کرد. هر چقدر تعداد لایه‌های میانی شبکه بیشتر باشد، شبکه، عمیق‌تر بوده و بار محاسباتی آن بیشتر می‌شود. لایه‌های میانی مسئولیت اعمال عملیات محاسباتی بر روی داده‌های دریافتی از لایه پیشین خود را برعهده دارند [۹].

۱-۳-۲- ۳- لایه خروجی

آخرین لایه شبکه عصبی، لایه خروجی شبکه است که مقادیر خروجی شبکه عصبی را محاسبه می‌کند. به عبارتی، لایه خروجی، با دریافت ورودی خود از لایه قبلی (لایه پنهان)، محاسباتی بر روی آن انجام داده و در نهایت خروجی شبکه را تولید می‌کند.

در شکل ۲-۲، نمونه‌ای از یک شبکه عصبی چند لایه با ۲ لایه میانی آورده شده است [۹].



شکل ۲-۲ نمونه‌ای از یک شبکه عصبی با دو لایه میانی [۹]

۱-۳- ۳- یادگیری در شبکه عصبی

پارامترهایی که در ساخته شدن خروجی مناسب از ورودی یک شبکه عصبی تاثیر مستقیم دارند، وزن‌ها و اریب‌ها می‌باشند. این دو مورد ابتدا به صورت تصادفی مقداردهی می‌شوند و خروجی بر اساس رابطه ۱-۲ محاسبه می‌شود.

از جایی که شبکه عصبی در الگوریتم‌های یادگیری با نظارت استفاده می‌شود، در مرحله تمرین، پاسخ واقعی ورودی شبکه عصبی مشخص است. در نتیجه می‌توان خروجی حاصل از شبکه عصبی را با خروجی واقعی مقایسه کرد و با تغییر مقادیر اریب‌ها و وزن‌ها تلاش کرد تفاوت میان خروجی شبکه عصبی و خروجی واقعی که همان تابع زیان می‌باشد را کاهش داد. شبه کد این الگوریتم در شکل ۲-۳ قابل مشاهده می‌باشد [۹].

```

Given training set  $\{(\mathbf{x}_i, y_i)\}_1^n$  and learning rate  $\eta \in \mathbb{R}$ 
Initialize  $\mathbf{w}$  and  $b$  to small random values
Repeat
    For  $i = 1$  to  $n$ 
        If  $y_i f(\mathbf{x}_i) \leq 0$  (if misclassification)
             $\mathbf{w} \leftarrow \mathbf{w} + \eta y_i \mathbf{x}_i$ 
             $b \leftarrow b + \eta y_i$ 
    End For
    Until no misclassifications made within the For loop
Return  $\mathbf{w}, b$ 

```

شکل ۲-۳- شبکه کد الگوریتم تمرین شبکه عصبی [۹]

در این الگوریتم پس از محاسبه تابع زیان که مربع تفاضل خروجی مورد انتظار از خروجی شبکه عصبی می‌باشد، در شبکه از لایه‌ی خروجی به سمت عقب یعنی به سمت لایه‌ی ورودی حرکت می‌شود و ضمن پس‌انتشار، وزن‌ها به گونه‌ای به روزرسانی می‌شوند که تابع زیان کمینه شود. هنگامی که مقدار این معیار از تمامی داده‌های آموزشی عبور کند و مقدار آن به اندازه‌ی کافی کوچک شود، شبکه همگرا شده است [۹].

۲-۱-۴ معرفی دو الگوریتم مشهور تشخیص چهره

الگوریتم‌های زیادی برای تشخیص چهره ارائه شده است. در این بخش، دو مورد از مشهورترین این الگوریتم‌ها مورد بررسی قرار می‌گیرند.

۲ - ۱ - ۴ - ۱- الگوریتم شبکه عصبی پیچشی^۱

اولین روشی که قرار است به عنوان یک الگوریتم تشخیص چهره ارائه شود، شبکه عصبی پیچشی می‌باشد. این الگوریتم، یک الگوریتم متبنی بر هوش مصنوعی است که می‌تواند یک تصویر را به عنوان ورودی دریافت کند، به جنبه‌ها یا اشیاء مختلف درون تصویر وزن اختصاص دهد تا بتواند یکی را از دیگری تمایز کند. پیش‌پردازش موردنیاز در شبکه عصبی پیچشی در مقایسه با سایر الگوریتم‌های طبقه‌بندی^۲ بسیار کمتر است. در حالی که در روش‌های ابتدایی، فیلترها به صورت دستی مهندسی می‌شوند، با آموزش کافی، شبکه عصبی پیچشی توانایی یادگیری این فیلترها و ویژگی‌ها را دارد [۱۰].

معماری یک شبکه عصبی پیچشی مشابه الگوی اتصال نورون‌ها در مغز انسان است و از سازماندهی قشر بینایی الهام گرفته شده است. نورون‌های منفرد به محرک‌ها فقط در یک ناحیه محدود از میدان بینایی به نام میدان پذیرنده پاسخ می‌دهند. مجموعه‌ای از این زمینه‌ها با هم هم‌پوشانی دارند تا کل ناحیه بصری را پوشش دهند [۱۰].

یک شبکه عصبی پیچشی می‌تواند وابستگی‌های مکانی و زمانی^۳ را در یک تصویر از طریق اعمال فیلترهای مربوط به این وابستگی‌ها با موفقیت ثبت کند. این معماری به دلیل کاهش تعداد پارامترهای درگیر و قابلیت استفاده مجدد وزن‌ها، برازش^۴ بهتری با مجموعه داده تصویر انجام می‌دهد. به عبارت دیگر، شبکه را می‌توان آموزش داد تا پیچیدگی تصویر را بهتر درک کند [۱۰].

۲ - ۱ - ۴ - ۱- تفکیک عکس ورودی

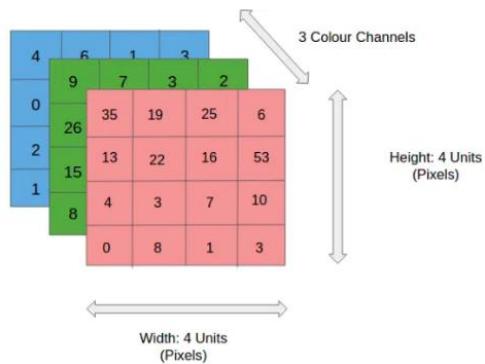
یک تصویر را می‌توان ماتریسی از مقادیر پیکسلی در نظر گرفت. این مقادیر پیکسلی در اصل بیانگر رنگ هر پیکسل می‌باشند که می‌توانند در فضاهای رنگی مختلفی باشند. یکی از رایج‌ترین فضاهای رنگ، فضای رنگ RGB می‌باشد. در این فضا، مطابق شکل ۲-۴ هر عکس از سه کanal قرمز، سبز و آبی تشکیل شده و این مقادیر در کنار هم، رنگ پیکسل‌های عکس مورد نظر را تشکیل می‌دهند [۱۰].

¹ Convolutional Neural Network

² Classification

³ Spatial and Temporal dependencies

⁴ Fitting

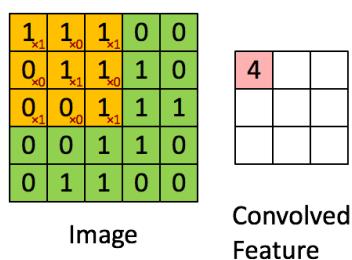


شکل ۲ - ۴ - تقسیم رنگ به سه کanal رنگی قرمز، سبز و آبی [۱۰]

۲ - ۱ - ۴ - ۱ - ۲ - لایه پیچشی

در این مرحله یک ماتریس به نام هسته^۱ تعریف می‌شود که روی پیکسل‌های عکس از چپ به راست و از بالا به پایین حرکت می‌کند و در مقادیر آن‌ها ضرب می‌شود. حاصل این ضرب در ماتریس جدیدی ذخیره می‌شود که ماتریس پیچشی نام دارد که با توجه به قوانین ماتریس‌ها، ابعاد کوچک‌تری نسبت به ماتریس عکس ورودی دارد. در نتیجه پردازش آن ساده‌تر خواهد بود. این ماتریس در واقع بیانگر ویژگی‌های پیچشی عکس ورودی می‌باشد [۱۰].

به عنوان مثال در شکل ۲ - ۵، نشان داده شده که ماتریس هسته (زرد رنگ) دارای ابعاد 3×3 می‌باشد، در تمام پیکسل‌های یکی از کanal‌های عکس ورودی (بخش سبز زنگ) ضرب شده و مقادیر آن در ماتریس جدید (قرمز رنگ) ذخیره می‌شود [۱۰].

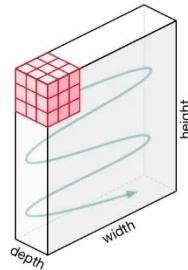


شکل ۲ - ۵ - نحوه عملکرد لایه پیچشی برای یک کanal رنگی [۱۰]

¹ Kernel

می‌توان به جای این که این کار را تنها روی یک کانال رنگی انجام داد، از ماتریس هسته سه بعدی با ابعاد $3 \times 3 \times 3$ استفاده کرد که ماتریس پیچشی ساخته شده یک جا ساخته شود. این فرآیند در شکل ۲-۶ نشان داده شده است

[۱۰]

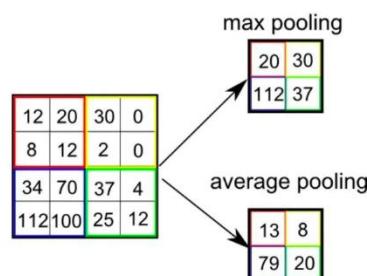


شکل ۲-۶- استفاده از ماتریس هسته سه بعدی برای محاسبه ماتریس پیچشی [۱۰]

۱ - ۲ - ۱ - ۴ - ۳ - لایه ادغام^۱

مشابه لایه پیچشی، لایه ادغام مسئول کاهش اندازه فضایی ویژگی پیچشی است. در پیاده‌سازی الگوریتم شبکه عصبی پیچشی، دو نوع ادغام وجود دارد: «ادغام حداکثری^۲» و «ادغام میانگین^۳».

ادغام حداکثری، حداکثر مقدار را از قسمت تصویر پوشانده شده توسط لایه هسته برمی‌گرداند. از سوی دیگر، ادغام میانگین، میانگین تمام مقادیر را از قسمتی از تصویر تحت پوشش هسته برمی‌گرداند. در شکل ۲-۷ تفاوت میان این دو نوع ادغام شرح داده شده است [۱۰].



شکل ۲-۷- مقایسه ادغام حداکثری و ادغام میانگین [۱۰]

^۱ Pooling Layer

^۲ Max Pooling

^۳ Average Pooling

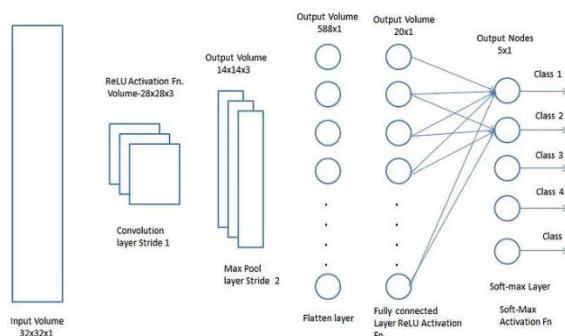
برای مقایسه این دو نوع ادغام می‌توان نشان داد که ادغام حداکثری به عنوان یک سرکوب‌کننده نویز^۱ عمل می‌کند، فعال‌سازی‌های دارای نویز را به کلی حذف می‌کند و همچنین نویز زدایی را همراه با کاهش ابعاد انجام می‌دهد. از این رو، می‌توان گفت که ادغام حداکثری بسیار بهتر از ادغام میانگین عمل می‌کند [۱۰].

لایه پیچشی و ادغام با هم لایه n -ام یک شبکه عصبی پیچشی را تشکیل می‌دهند. بسته به پیچیدگی‌های موجود در تصاویر، ممکن است تعداد این لایه‌ها برای ثبت جزئیات سطح پایین افزایش یابد، ولی این کار باعث افزایش سربار محاسباتی می‌شود [۱۰].

۲ - ۱ - ۴ - ۱ - ۴ - ۱ - ۲ لایه کاملاً متصل^۲

افزودن یک لایه کاملاً متصل، روشی به صرفه برای یادگیری ترکیب‌های غیرخطی ویژگی‌های سطح بالا می‌باشد که در حال یادگیری یکتابع احتمالاً غیرخطی در آن فضا است [۱۰].

اکنون که تصویر ورودی به فرمی مناسب برای Perceptron چند سطحی تبدیل شده است، به یک بردار ستونی مسطح ۱۲۸ بعدی تبدیل می‌شود و خروجی مسطح شده به یک شبکه عصبی پیش‌خور^۳ داده می‌شود و پس انتشار برای هر تکرار تمرین اعمال می‌شود. در طی یک سری از دور^۴ها، این مدل قادر است بین ویژگی‌های غالب و برخی از ویژگی‌های سطح پایین در تصاویر تمایز قائل شود و آن‌ها را طبقه‌بندی کند. مدل کلی یک شبکه عصبی پیچشی در شکل ۲-۸ نمایش داده شده است [۱۰].



شکل ۲-۸ مدل کلی یک شبکه عصبی پیچشی [۱۰]

¹ Noise Supressant

² Fully Connected Layer (FC Layer)

³ Feed-forward

⁴ Epoch

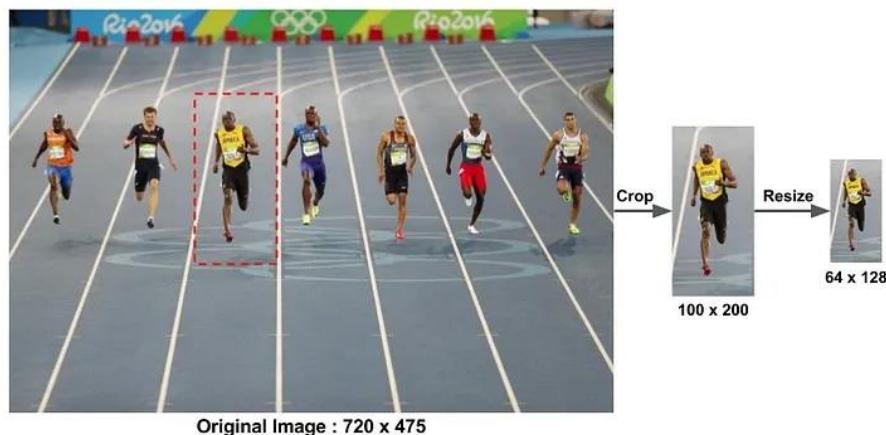
۲ - ۱ - ۴ - ۲ الگوریتم هیستوگرام شبکه های جهت دار

این مدل یک الگوریتم سریع در فرآیند تشخیص چهره به حساب می آید، اما نسبت به الگوریتم شبکه های عصبی پیچشی دقت کمتری خواهد داشت. یکی دیگر از اشکالات این الگوریتم، حساس بودن آن به چرخش عکس ورودی و زاویه دید آن می باشد. هیستوگرام شبکه های جهت دار، یک توصیف گر ویژگی است و در بینایی کامپیوترا و پردازش تصویر به منظور تشخیص اشیاء استفاده می شود. این روش، جهتگیری شبکه را در قسمت محلی یک تصویر شمارش می کند. این توصیف گر بر ساختار یا شکل یک شی تمرکز می کند و از بزرگی و همچنین زاویه شبکه برای محاسبه ویژگی ها استفاده می کند و برای مناطق تصویر، هیستوگرام هایی را با استفاده از بزرگی و جهتگیری شبکه تولید می کند [۱۰].

گام های استخراج ویژگی توسط این الگوریتم در این بخش معرفی می شوند.

۲ - ۱ - ۴ - ۳ - ۱ پیش پردازش

برای شروع، باید اندازه عکس مورد نظر نسبت ۲ به ۱ داشته باشد. به این منظور، اندازه تصویر ورودی به یک تصویر 64×128 پیکسل تغییر داده می شود. این نسبت طی تحقیق های مداوم پژوهشگران بینایی ماشین با هدف به دست آوردن نتایج بهتر در کار تشخیص عابر پیاده پیشنهاد شد [۱۰]. مراحل پیش پردازش عکس در این الگوریتم در شکل ۲ - ۹ آورده شده است.



شکل ۲ - ۹ پیش پردازش در الگوریتم هیستوگرام شبکه های جهت دار [۱۰]

۱ - ۲ - ۴ - ۲ - محاسبه شیب

شیب از ترکیب بزرگی و زاویه از عکس ورودی محاسبه می‌شود. برای یک بلوک 3×3 پیکسلی، ابتدا مقادیر G_x و G_y طبق رابطه ۲-۲ و ۳-۲ برای هر پیکسل محاسبه می‌شود که r و c به ترتیب به سطر و ستون هر بلوک اشاره می‌کنند [۱۰].

$$G_x(r, c) = I(r, c + 1) - I(r, c - 1) \quad (۳-۲)$$

$$G_y(r, c) = I(r - 1, c) - I(r + 1, c) \quad (۳-۲)$$

سپس به کمک رابطه ۴-۲ و ۵-۲، به ترتیب بزرگی و زاویه هر پیکسل محاسبه می‌شود [۱۰].

$$\text{Magnitude}(\mu) = \sqrt{G_x^2 + G_y^2} \quad (۴-۲)$$

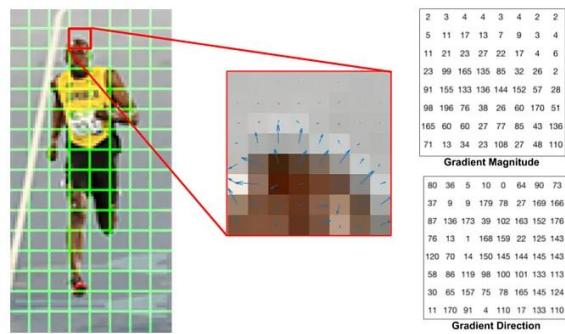
$$\text{Angle}(\theta) = \left| \tan^{-1} \left(\frac{G_y}{G_x} \right) \right| \quad (۵-۲)$$

۱ - ۲ - ۴ - ۳ - ساخت هیستوگرام از شیب

برای شروع این مرحله تصویر سلول‌هایی تقسیم می‌شود تا هیستوگرام شیب برای هر سلول محاسبه شود. به عنوان مثال، برای تصویری با ابعاد 64×128 ، به سلول‌های 8×8 تقسیم می‌شود. یک سلول 8×8 را می‌توان به سادگی با استفاده از 128 عدد توضیح داد ($2 \times 8 \times 8$ که در آن 2 مورد آخر مربوط به مقدار گرادیان و مقادیر جهت هستند). با تبدیل بیشتر این اعداد به محاسبه هیستوگرام، هر وصله^۱ تصویری نسبت به نویز بسیار قوی‌تر عمل می‌کند [۱۰].

هر هیستوگرام باید به 9 بخش جداگانه تقسیم شود، به‌طوری که هر کدام مربوط به زوایای 0° تا 160° درجه با افزایش 20° درجه است. در شکل ۲-۱۰، مقادیر شیب برای یک سلول نمایش داده شده است. اندازه هر پیکان در این شکل بیانگر اندازه شیب هر پیکسل می‌باشد [۱۰].

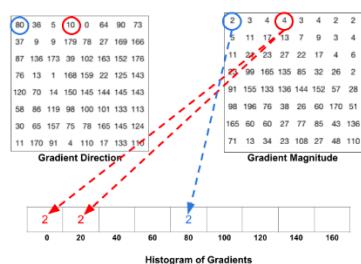
¹ patch



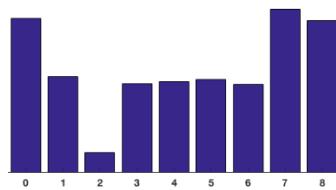
شکل ۲-۱۰ مقادیر اندازه و جهت شیب برای هر سلول [۱۰]

یک بخش از هیستوگرام بسته به جهت انتخاب شده انتخاب می‌شود و مقداری که بعداً در داخل بخش قرار می‌گیرد به بزرگی آن بستگی دارد. اگر یک پیکسل در نیمه راه بین دو بخش باشد، بسته به فاصله هر پیکسل از هر بخش مربوطه، مقدار بزرگی آن‌ها را بر اساس آن تقسیم می‌کند. در شکل ۲-۱۱ می‌توان نحوه قرارگیری بزرگی دو پیکسل قرمز و آبی را در هیستوگرام مشاهده کرد [۱۰].

پس از انجام این فرآیند می‌توان یک هیستوگرام تشکیل داد که هیستوگرام نهایی در شکل ۲-۱۲ نمایش داده شده است [۱۰].



شكل ۲-۱۱ تشریح شیوه ساخت هیستوگرام یک سلول



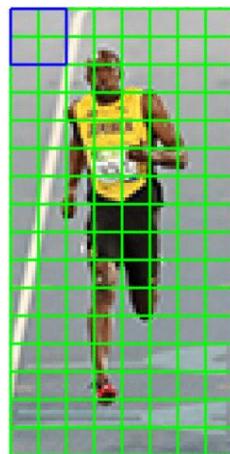
شکل - ۲ - ۱۰ هیستوگرام نهایی یک سلوول [۱۰]

۲ - ۱ - ۴ - ۲ - ۴ - ۱ - ۲ نرمال سازی بلوکها (16×16)

یک بلوک 16×16 دارای ۴ هیستوگرام است که می‌توان آن‌ها را به هم متصل کرد تا یک بردار 1×36 عنصری را تشکیل دهند و که قابل نرمال‌سازی است. سپس پنجره با ۸ پیکسل (عرض یک سلوول) جابه‌جا می‌شود و یک بردار عادی 1×36 بر روی این پنجره محاسبه و این روند تکرار می‌شود. این کار در شکل ۲ - ۱۳ نمایش داده شده است که بلوک 16×16 با رنگ آبی مشخص شده است [۱۰].

۲ - ۱ - ۴ - ۲ - ۵ محاسبه بردار ویژگی هیستوگرام شبیه جهت‌دار

برای محاسبه بردار ویژگی هیستوگرام شبیه جهت‌دار باید بردارهای 1×36 را به هم وصل شوند و یک بردار بزرگ ساخته شود. با توجه به این‌که ۷ بردار افقی و ۱۵ بردار عمودی 16×16 پیکسلی داریم، ۱۰۵ موقعیت در این حالت وجود دارد. در نتیجه طول این بردار بزرگ برابر است با $105 \times 36 = 3780$.



شکل ۲ - ۱۳ نرمال سازی بلوک‌ها در هیستوگرام [۱۰]

معمولًاً هیستوگرام‌های نرمال شده 1×9 در سلوول‌های 8×8 تجسم می‌شوند که خروجی آن در شکل ۲ - ۱۴ قابل مشاهده است.



[۱۰] شکل ۲-۱۴ خروجی الگوریتم هیستوگرام شبکهای جهت‌دار

۲-۲ معرفی ابزارها و فناوری‌های استفاده شده در پروژه

در این پروژه از ابزارهایی استفاده شده که در ادامه معرفی خواهند شد.

۲-۲-۱ ابزارهای سختافزاری

در این بخش، قطعاتی که برای پیاده‌سازی مدار پروژه استفاده شده است معرفی می‌شوند.

۲-۲-۱-۱ بورد Raspberry Pi

سختافزارهای متنوعی در بازار وجود دارند که به کمک آن‌ها می‌توان پروژه‌های متعددی را اجرا کرد. از آن‌جا که هدف اصلی این، پروژه پردازش تصویر می‌باشد، باید سختافزاری را انتخاب کنیم که از پس این کار به خوبی برآید. یکی از رایانه‌های تک-بورد^۱ قوی و پر فروش در بازار، بورد Raspberry Pi می‌باشد که به اندازه یک کف دست است و بنیاد Raspberry Pi آن را ساخته است. در این پروژه از ۴ Raspberry Pi مدل B با RAM یک گیگابایت استفاده شده است که در شکل ۲-۱۵ قابل مشاهده می‌باشد. این دستگاه مجهز به پردازنده Cortex-A72 مشتمل بر چهار هسته ۶۴ بیتی ۱.۵ گیگاهرتزی است که روی سیستم روی چیپ^۲ BCM2837 قرار دارد. Raspberry Pi، به یک جفت درگاه^۳ micro-HDMI مجهز شده است که پشتیبانی از ویدیوهای باکیفیت را فراهم کرده و

¹ Single-board computer

² System on a Chip

³ Port

قادر است در پروژه‌های جریان‌سازی^۱ ویدئو با کیفیت بالا مورد استفاده قرار گیرد. در جدول ۲-۲، خلاصه‌ای از مشخصات این بورد آورده شده است [۱۱].

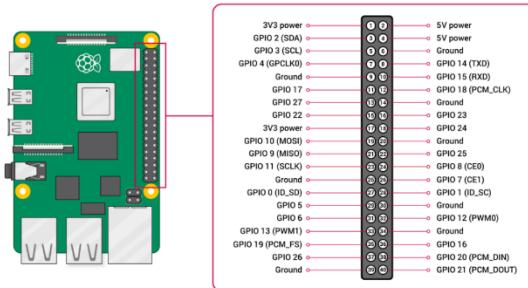


شکل ۲-۱۵ تصویر بورد Raspberry Pi

در شکل ۲-۱۶ پایه‌های بورد Raspberry Pi نمایش داده شده است [۱۱]. دو پایه ۵ ولت و دو پایه ۳.۳ ولت روی بورد و همچنین تعدادی پایه زمین (۰ ولت) وجود دارد که قابل تنظیم نیستند. پایه‌های باقی‌مانده همگی پایه‌های ۳.۳ ولتی هستند، به این معنی که خروجی‌ها روی ۳.۳ ولت تنظیم شده‌اند و ورودی‌ها ۳.۳ ولت تحمل دارند. در شکل ۲-۱۷ مشخص شده هر پایه Raspberry Pi دارای چه ولتاژی می‌باشد [۱۱].

دو پایه ۵ ولت و دو پایه ۳.۳ ولت روی بورد و همچنین تعدادی پایه زمین (۰ ولت) وجود دارد که قابل تنظیم نیستند. پایه‌های باقی‌مانده همگی پایه‌های ۳.۳ ولتی هستند، به این معنی که خروجی‌ها روی ۳.۳ ولت تنظیم شده‌اند و ورودی‌ها ۳.۳ ولت تحمل دارند. در شکل ۲-۱۷ مشخص شده که هر پایه Raspberry Pi دارای چه ولتاژی می‌باشد [۱۱].

¹ Streaming Pin

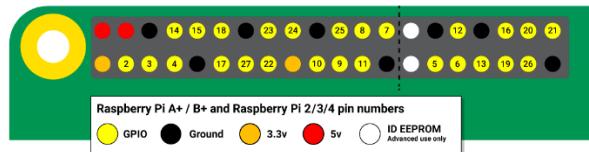


شکل ۲- ۱۶ پایه‌های بورد Raspberry Pi

جدول ۲- مشخصات فنی بورد Raspberry Pi

نام ویژگی	توضیحات
پردازنده	Cortex A-72- ۱.۵- گیگاهرتز
حافظه RAM	۱ گیگابایت ^۱ DDR4
پردازنده گرافیکی	پردازنده VideoCore VI با فرکانس ۵۰۰ مگاهرتز
خروجی ویدیو	دو درگاه micro-HDMI
درگاههای USB	دو درگاه USB مدل ۲ و دو درگاه USB مدل ۳
درگاه شارژ	یک درگاه USB مدل C
جریان موردنیاز	۳ آمپر
ولتاژ موردنیاز	۵ ولت
ابعاد	۱۹.۵ × ۵۸ × ۸۸ میلی‌متر
وزن	۴۶ گرم
انرژی مصرفی	از ۵۴۰ میلی‌آمپر (۲.۷ وات) تا ۱۲۸۰ میلی‌آمپر (۶.۴ وات)

^۱ Double Data Rate 4 Synchronous Dynamic Random-Access Memory



شکل ۲-۱۷ ولتاژ پایه‌های Raspberry Pi

۲-۱-۲-۲ مازول دوربین

دوربین ۵ مگاپیکسل Raspberry Pi، مازولی مخصوص بوردهای Raspberry Pi می‌باشد که توسط همین شرکت تولید شده است. این مازول بر روی یکی از ۲ سوکت رابط سریالی دوربین^۱ موجود بر روی بورد نصب خواهد شد که قابلیت پردازش بالایی دارند. ابعاد این بورد ۹×۲۰×۲۵ میلی‌متر می‌باشد که نسبتاً کوچک می‌باشد و حدود ۳ گرم وزن دارد که آن را برای پروژه‌هایی که سایز و وزن در آن‌ها مهم می‌باشند، بسیار مناسب نموده است. این دوربین توسط یک کابل ربانی به بورد متصل می‌شود و قابلیت دریافت تصاویر با کیفیت ۵ مگاپیکسل با حداکثر ابعاد ۱۹۴۴×۲۵۹۲ دارد [۱۲]. شکل ۲-۱۸ نمایی از این مازول را نشان می‌دهد.



شکل ۲-۱۸ مازول دوربین Raspberry Pi

۲-۱-۳-۲ قفل Solenoid

Solenoid ها یک راه حل ساده و موثر برای کنترل سوپاپ‌ها و سوئیچ‌های الکترومغناطیسی یا قفل مکانیکی هستند. اصل عملکرد و واکنش آنی، آن‌ها را به یک راه حل خوب برای کاربردهایی تبدیل کرده که نیاز به مقدار زیادی نیرو در فضای کوچک دارند و در آن‌جا نیاز به عملکرد سریع و قوی وجود دارد. نمونه‌ای از این قبل در شکل ۲-۱۹ قابل مشاهده می‌باشد. قفل‌های Solenoid در ولتاژ‌های کاری متفاوتی وجود دارند که دو نوع ۱۲ و ۵ ولتی آن، از

^۱ Camera Serial Interface (CSI)

محبوب ترین انواع آن محسوب می‌شود. از آن جا که ولتاژ کاری پایه‌های بورد ۳.۳ Raspberry Pi ۵ ولت می‌باشد، در صورت استفاده از قفل Solenoid ۱۲ ولتی باید از یک رله^۱ و یک منبع تغذیه ۱۲ ولتی استفاده شود.



شکل ۲-۱۹- قفل Solenoid

۲-۲-۱-۴ صفحه کلید عددی

علاوه بر صفحه کلید رایانه، می‌توان از صفحه کلید ماتریسی به عنوان ورودی دستورات کاربران در سامانه استفاده کرد. در این پروژه از یک صفحه کلید 4×4 استفاده می‌شود که همان‌طور که در شکل ۲-۲۰ نشان داده شده است، شامل ارقام ۰ تا ۹، حروف "A", "B", "C", "D" و همچنین حروف "*" و "#" می‌باشد. برای اتصال این صفحه کلید به بورد Raspberry Pi باید ۸ پایه از بورد به آن اختصاص داده شود. نمایی از یک صفحه کلید عددی در شکل ۲-۲۰ قابل مشاهده است.



شکل ۲-۲۰ صفحه کلید عددی

۲-۲-۱-۵ دیود نشردهنده نور

دیود نشردهنده نور^۲ (LED) یک منبع نور نیمرسانا^۳ است که با عبور جریان الکتریکی از خود، نور تابش می‌کند. الکترون‌های موجود در ماده نیمرسانا با حفره‌های الکترون بازترکیب^۴ می‌شوند و انرژی خود را به شکل فوتون آزاد

¹Relay

²Light-Emitting Diode

³Semiconductor

⁴Recombine

می‌کند. در این پروژه از این قطعه برای اطلاع‌رسانی به کاربر استفاده می‌شود. یک نمونه دیود نشرده‌نده نور در شکل ۲-۲۱ آورده شده است.



شکل ۲-۲۱ دیود نشرده‌نده نور

۲-۱-۶ آژیر

آژیر وسیله‌ای است که می‌تواند سیگنال‌های صوتی را به صدا تبدیل کند و معمولاً با ولتاژ DC کار می‌کند. آژیرها به دو دسته پیزوالکتریک و الکترومغناطیسی تقسیم می‌شوند و با توجه به طراحی و کاربردهای مختلف، آژیر می‌تواند صدای مختلفی مانند موسیقی، آژیر و زنگ هشدار را منتشر کند. در شکل ۲-۲۲ نمونه‌ای از یک آژیر قابل مشاهده است.



شکل ۲-۲۲ آژیر

۲-۱-۷ ماژول Sim۹۰۰

ماژول Sim۹۰۰ یک مودم^۱ کوچک با قابلیت‌هایی چون دریافت و ارسال پیام، دریافت و ایجاد تماس‌های تلفنی، خدمات GPRS^۲ و انواع دیگر از کاربردها در شبکه‌های اینترنت اشیاء^۳ است. به طور ویژه از ماژول Sim۹۰۰ برای اتصال دستگاه‌های خانگی و صنعتی به اینترنت و ایجاد بستر کنترل از راه دور استفاده می‌شود. اصطلاح فنی GSM/GPRS شبیه یک پروتکل برای تعریف تمام خدمات موبایلی مانند ارسال پیامک، خدمات تماس و مکان‌یابی

¹ Modem

² General Packet Radio Services

³ Internet of Things (IoT)

است. این خدمات پایه‌ای ترین دسترسی‌هایی هستند که یک دستگاه بی‌سیم یا یک ماشین متصل به یک شبکه ارتباطی اینترنتی می‌تواند در اختیار کاربر خود قرار دهد.

این مژول‌ها، قطعات الکترونیکی کوچکی هستند که با صرف پایین‌ترین مقادیر انرژی، خدمات پایه GSM/GPRS را در اختیار کاربر خود قرار می‌دهند. نمایی از یک مژول Sim900 در شکل ۲-۲ قابل مشاهده است که در این پروژه، از آن برای ارسال پیامک استفاده می‌شود.



شکل ۲-۲ مژول Sim900

۲-۲-۲ ابزارهای نرم‌افزاری

برای پیاده‌سازی منطق و نرم‌افزار پروژه، از ابزارهایی استفاده شده است که در ادامه به صورت خلاصه معرفی شده‌اند.

۲-۲-۱ زبان برنامه‌نویسی Python

Python یک زبان برنامه‌نویسی شیء‌گرا، تفسیری، سطح بالا و همه‌منظوره^۱ است و اولین بار در سال ۱۹۹۱ توسط Guido van Rossum منتشر شده است. فلسفه اصلی طراحی Python خوانایی بالایی کد است. ساختار زبانی و دیدگاه شیء‌گرا^۲ در Python به گونه‌ای طراحی شده است که برای برنامه‌نویس امکان نوشتن کد منطقی و واضح برای پروژه‌های کوچک و بزرگ فراهم می‌کند^[۱۳]. نماد این زبان برنامه‌نویسی در شکل ۲-۲ آورده شده است. از آنجا که کتابخانه‌های مفیدی در حیطه هوش مصنوعی و همچنین کتابخانه‌های مربوط کار با بوردهای Raspberry Pi وجود دارد، در این پروژه از زبان Python استفاده می‌شود.

¹ General-purpose
² Object oriented



[۱۳] شکل ۲-۲ نماد Python

۲-۲-۱-۱-۱-۲-۲ کتابخانه‌های مورد استفاده برای توسعه پروژه

کتابخانه‌ها به برنامه‌نویسان کمک می‌کنند تا با قرار دادن تمرکز خود روی منطق اصلی برنامه‌شان، سرعت توسعه خود را افزایش دهند. در ادامه به معرفی چند کتابخانه اصلی مورد استفاده در این پروژه پرداخته خواهد شد.

۲-۲-۱-۱-۲-۱-۱-۲-۲ کتابخانه‌ی OpenCV

کتابخانه opencv یک کتابخانه متن باز^۱ بینایی ماشین است که سال ۲۰۰۰ میلادی توسط شرکت اینتل^۲ معرفی شد و با زبان‌های C و C++ توسعه داده شده است[۱۴]. از این کتابخانه در این پروژه برای پیاده‌سازی جریان اجرایی برنامه استفاده می‌شود؛ به این صورت که تصویر دوربین تعیین شده برای برنامه را دریافت کرده و با قابلیت افزودن متون و اشکال مختلف، آن را در نمایش‌گر نشان می‌دهد. این دوربین می‌تواند دوربین رایانه شخصی، دوربین مخصوص بورد Raspberry Pi یا هر دوربین دیگری باشد که به بستر اجرایی برنامه متصل است. در بخش‌های بعدی در مورد تنظیمات دوربین این سامانه توضیحات بیشتری ارائه شده است [۱۴]. نماد این کتابخانه در شکل ۲-۲۵ قابل مشاهده می‌باشد.



[۱۴] شکل ۲-۲۵ نماد OpenCV

^۱ Open-source
^۲ Intel

۲ - ۱ - ۱ - ۲ - ۲ - ۲ کتابخانه dlib

کتابخانه dlib یک کتابخانه چند منظوره است که در سال ۲۰۰۲ معرفی شده است. این کتابخانه با زبان ^۱ C توسعه داده شده است و در بسیاری از زمینه‌ها از جمله شبکه، ریسمان‌ها^۲، رابطه‌های کاربر گرافیکی^۳، ساختارهای داده، جبر خطی^۴، یادگیری ماشین، پردازش تصویر^۵، داده‌کاوی^۶، بهینه‌سازی عددی و شبکه‌های بیزی^۷، ابزارهایی در اختیار کاربران خود قرار می‌دهد که بتوانند کارهای خود را راحت‌تر و بهینه‌تر انجام دهند [۱۵]. در این پروژه از این کتابخانه جهت تسهیل فرآیند تشخیص چهره و استفاده از ابزارهایی جهت بهبود کیفیت شناسایی چهره استفاده شده است. نماد مربوط به این کتابخانه در شکل ۲-۲۶ قابل مشاهده است.



شکل ۲-۲۶ نماد Dlib [۱۵]

۲ - ۱ - ۱ - ۲ - ۳ کتابخانه NumPy

کتابخانه NumPy یک کتابخانه برای زبان برنامه‌نویسی Python است که پشتیبانی از آرایه‌ها و ماتریس‌های بزرگ و چند بعدی را به همراه مجموعه بزرگی از توابع ریاضی سطح بالا برای کار بر روی این آرایه‌ها ارائه می‌کند [۱۶]. از این کتابخانه در محاسبات مربوط به شناسایی چهره استفاده می‌شود و نماد آن در شکل ۲-۲۷ قرار دارد.



شکل ۲-۲۷ نماد NumPy [۱۶]

^۱ Cross-platform
^۲ Threads

^۳ Graphical User Interface (GUI)

^۴ Linear algebra

^۵ Image processing

^۶ Data mining

^۷ Bayesian networks

۲ - ۱ - ۱ - ۴ - کتابخانه matplotlib

کتابخانه Matplotlib یک کتابخانه ترسیم نمودار برای زبان برنامه‌نویسی Python می‌باشد که در دل خود از کتابخانه Numpy استفاده می‌کند. به کمک این نمودار می‌توان انواع نمودارهای رایج را به راحتی رسم کرد [۱۷]. تمامی نمودارهای موجود در بخش ارزیابی پروژه، توسط کتابخانه matplotlib رسم شده‌اند. در شکل ۲-۲۸، نماد این کتابخانه آورده شده است.



[۱۷] شکل ۲-۲۸ نماد matplotlib

۲ - ۱ - ۱ - ۵ - کتابخانه RPi.GPIO

برای اجرا برنامه روی بورد Raspberry Pi، می‌توان به کمک کتابخانه RPi.GPIO پایه‌های بورد را کنترل کرد و از مقادیر هر پایه استفاده کرد. در صورتی که Raspberry Pi از سیستم‌عامل Raspbian استفاده کند، این کتابخانه به‌طور پیش‌فرض بر روی آن نصب شده است و نیازی به نصب مجدد آن نیست.

۲ - ۱ - ۶ - کتابخانه Flask

Flask یک چارچوب وب^۱ ساده و سبک و در عین حال قدرتمند برای زبان برنامه‌نویسی Python است. عموماً به عنوان یک ریزچارچوب شناخته می‌شود؛ به این معنی که خصوصیاتی نظیر نگاشت شیء-رابطه‌ای، اعتبارسنجی فرم‌ها و دیگر ویژگی‌هایی که در چارچوب‌های بزرگ‌تر مانند Django یافت می‌شوند را ندارد؛ اما در عوض توسعه‌دهنده می‌تواند برای اعمال پیاده‌سازی مورد نظرش به راحتی عمل کند [۱۸]. از این کتابخانه که نماد آن در شکل ۲-۲۹ آورده شده است، یک تابلوی کنترلی^۲ مدیر برای مشاهده وضعیت سامانه معرفی شده که از Flask استفاده می‌کند.



[۱۸] شکل ۲-۲۹ نماد Flask

¹ Web framework
² Panel

۲ - ۲ - ۱ استفاده از محیط یکپارچه توسعه^۱ PyCharm

محیط توسعه یکپارچه، یک برنامه نرم‌افزاری است که به برنامه‌نویسان کمک می‌کند تا کد نرم‌افزار را به‌طور موثر توسعه دهند. بهره‌وری توسعه‌دهندگان را با ترکیب قابلیت‌هایی مانند ویرایش نرم‌افزار، ساخت، آزمایش و بسته‌بندی در یک برنامه کاربردی آسان افزایش می‌دهد [۱۹].

با توجه به محبوبیت زبان برنامه‌نویسی Python در سال‌های اخیر، ویرایش‌گرهای بسیاری توسط شرکت‌های مختلف توسعه داده شده‌اند و برنامه‌نویسان می‌توانند برای سهولت توسعه برنامه خود از آن‌ها استفاده کنند. یکی از محبوب‌ترین آن‌ها، PyCharm می‌باشد که توسط شرکت JetBrains ارائه شده و ابزارهای بسیار مفیدی را در اختیار برنامه‌نویس می‌گذارد [۲۰]. در پیاده‌سازی این پژوهه نیز از PyCharm استفاده شده است که نماد آن در شکل ۲ - ۳۰ قابل مشاهده است.



شکل ۲ - ۳۰ نماد PyCharm [۲۰]

^۱ Integrated Development Environment (IDE)

۳ - طراحی سامانه

پیش از شروع پیاده‌سازی سامانه، باید جنبه‌های مختلف آن را طراحی شود. در این فصل، به توضیح بخش‌های مختلفی پرداخته می‌شود که در ادامه، پیاده‌سازی خواهد شد.

۳ - ۱ معرفی مفهوم کاربر

به هر شخصی که قرار است از سامانه استفاده کند، یک کاربر گفته می‌شود. در این بخش، نکات لازم برای هر کاربر بررسی می‌شود.

۳ - ۱ - ۱ شناسه کاربری

هر شخص که قرار است در مقابل دوربین این سامانه قرار بگیرد و احراز هویت شود، یک کاربر می‌باشد. هر کاربر در صورتی که در سامانه ثبت شده باشد، دارای یک شناسه^۱ یکتا می‌باشد که در واقع یک عدد چهار رقمی بوده و با اضافه شدن شخص جدید به سامانه، این شناسه‌ها به صورت صعودی به اندازه یک واحد افزایش پیدا می‌کنند.

۳ - ۱ - ۲ نام کاربر

هر کاربر می‌تواند در سامانه یک نام داشته باشد که این نام در بالای تصویر گرفته شده توسط دوربین سامانه از کاربر نمایش داده خواهد شد. البته نام یک بخش اختیاری برای هر کاربر می‌باشد و در تنظیمات سامانه که در ادامه به آن پرداخته خواهد شد، می‌توان از ذخیره‌ی آن صرف‌نظر کرد. نام دو کاربر می‌تواند یکتا نباشد و یکسان بودن نام دو کاربر، اشکالی ایجاد نمی‌کند.

۳ - ۱ - ۳ عکس‌های ذخیره شده از هر کاربر

از آن جا که قرار است سامانه چهره افرادی که نام آن‌ها در سامانه ثبت شده را شناسایی کند، لازم است عکس‌هایی از چهره هر شخص در این سامانه ذخیره شود. علت این‌که به ازای هر چهره چندین عکس می‌تواند وجود داشته باشد این است که سامانه بتواند حالات مختلف چهره اشخاص را به خاطر بسپارد و عمل احراز هویت را راحت‌تر انجام دهد.

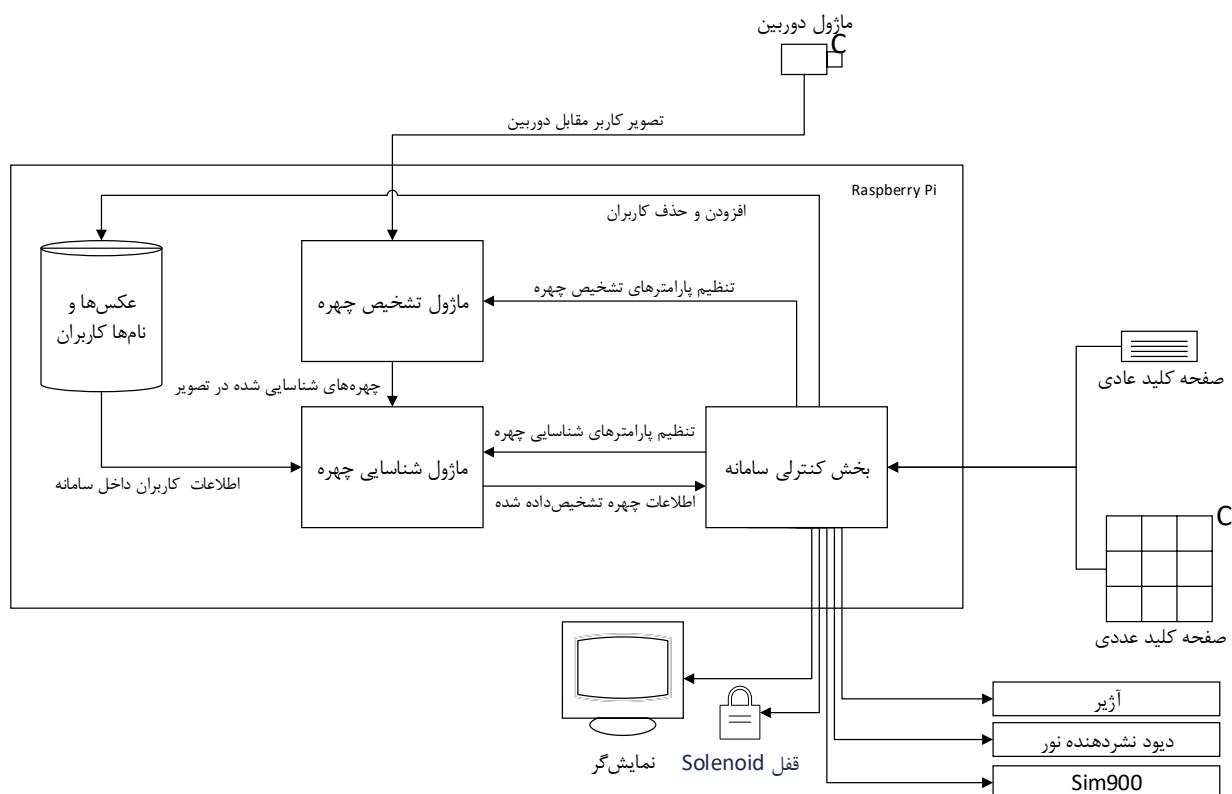
¹ ID

۳ - ۱ - ۴ نامگذاری فایل‌های عکس

فایل‌هایی که به عنوان تصاویر کاربران سامانه ذخیره می‌شوند، دارای نامی با قالب $id_n.jpg$ می‌باشد که id همان شناسه کاربری و n شماره عکس مربوط به آن کاربر می‌باشد.

۳ - ۲ - ۳ طراحی معماری سامانه

معماری سامانه ارائه شده در این پروژه در شکل ۳ - ۱ قابل مشاهده است که در ادامه، به توضیح بخش‌های اصلی آن می‌پردازیم:



شکل ۳ - ۱ - معماری سامانه

۳ - ۲ - ۱ مجموعه داده‌ها

عکس کاربران و همچنین نام آن‌ها در این بخش ذخیره می‌شود. این اطلاعات در اختیار مازول شناسایی چهره قرار داده می‌شود که بتوان به کمک آن، فرایند شناسایی را انجام داد. اطلاعات این بخش می‌تواند توسط بخش کنترلی سامانه تغییر پیدا کند.

۳ - ۲ - ۲ مازول کنترلی سامانه

مدیریت اتفاق‌ها سامانه در بخش کنترلی انجام خواهد شد. این بخش می‌تواند از کاربر به کمک صفحه کلید عددی یا صفحه کلید دستور بگیرد. همچنین توانایی ایجاد تغییر در بخش‌های دیگر سامانه را دارد؛ به عنوان مثال می‌تواند پیکربندی^۱ سایر بخش‌ها را تغییر دهد، به کاربران سامانه اضافه کند یا آن‌ها را حذف کند. همچنین کنترل جریان اجرایی برنامه نیز بر عهده این بخش می‌باشد.

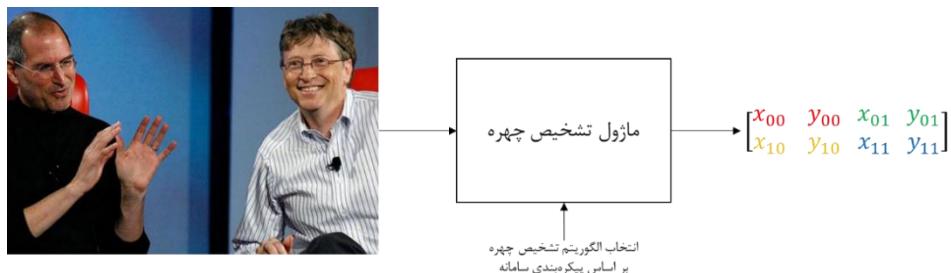
۳ - ۲ - ۳ مازول تشخیص چهره

وظیفه این مازول، استفاده از دوربین برای دریافت تصویر مقابله آن است که پس از تشخیص چهره‌های موجود در این تصویر، آن را در اختیار مازول شناسایی چهره قرار می‌دهد. پارامترهای این بخش از مازول کنترلی سامانه کنترل می‌شود.

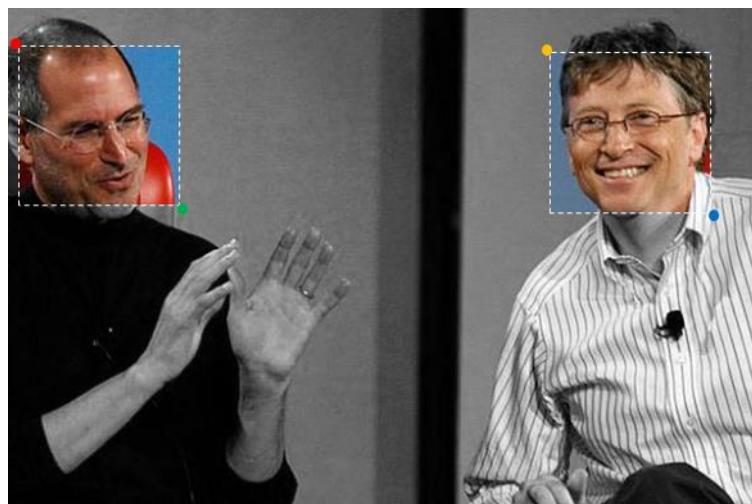
در این پروژه از دو الگوریتم شبکه‌های عصبی پیچشی و همچنین الگوریتم هیستوگرام شبکه‌های جهت‌دار برای تشخیص چهره‌ها استفاده می‌شود. این دو الگوریتم به صورت مجزا از یکدیگر کار می‌کنند و در تنظیمات سامانه مشخص می‌شود که عکس ورودی، از کدام الگوریتم برای فرایند تشخیص چهره استفاده می‌کند. ورودی این بخش یک عکس می‌باشد و خروجی آن، ماتریسی با ۴ ستون به تعداد چهره‌های موجود در عکس ورودی سطر است که مولفه‌های هر آرایه متناظر با مختصات دو گوشی چهره‌های تشخیص داده شده در عکس ورودی می‌باشد. ساختار مفهومی این مازول در شکل ۳-۲ قابل مشاهده می‌باشد. همچنین در شکل ۳-۳ مشخص شده که چگونه به کمک خروجی مازول تشخیص چهره می‌توان به چهره‌ها دسترسی پیدا کرد. طبق این شکل، به ازای هر چهره، دو عنصر اول آرایه بیان‌گر فاصله گوش به بالا سمت چپ از بالا و چپ تصویر، و دو عنصر دوم نیز بیان‌گر فاصله

¹ Configuration

گوشه پایین سمت راست چهره تشخیص داده شده می‌باشد. همان‌طور که مشاهده می‌شود، رنگ نقاط موجود در حاشیه چهره‌های شکل ۳-۳، متناظر با رنگ مولفه‌های آرایه شکل ۲-۳ می‌باشد.



شکل ۳-۲ ساختار ماژول تشخیص چهره



شکل ۳-۳ خروجی ماژول تشخیص چهره برای عکس ورودی حاوی دو چهره

۳ - ۲ - ۴ ماژول شناسایی چهره

این ماژول، وظیفه شناسایی عکس ورودی که از ماژول تشخیص چهره در اختیار دارد را بر عهده دارد که این کار را بر اساس داده‌های ذخیره شده در سامانه انجام می‌دهد. پارامترهای این بخش نیز از ماژول کنترلی سامانه کنترل می‌شود.

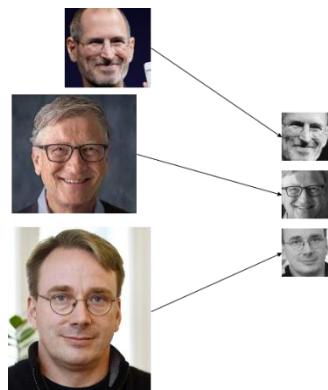
در ادامه، گام‌های لازم برای فرایند تشخیص چهره آورده شده است.

۳ - ۲ - ۱ - تبدیل هر چهره به بردار متناظر

رویکرد اصلی برای شناسایی چهره، تبدیل عکس چهره‌ها به بردار است. می‌توان هر عکس را ماتریسی از مقادیر پیکسل‌های آن عکس درنظر گرفت. هدف تبدیل این ماتریس به یک بردار می‌باشد.

۳ - ۲ - ۱ - پیش‌پردازش

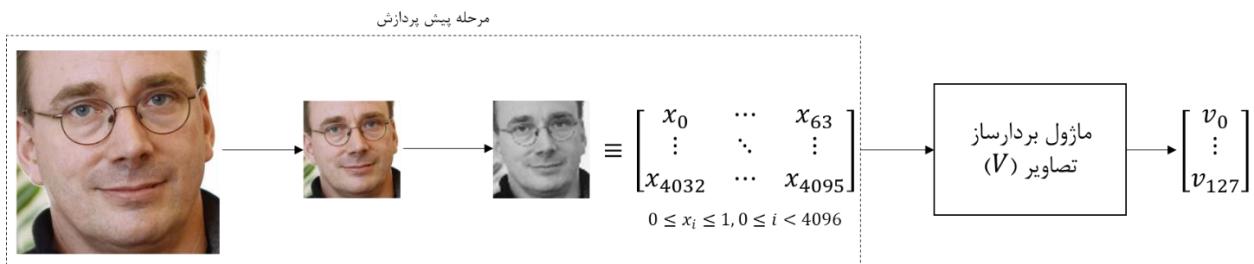
برای سادگی و کوچک‌تر شدن این ماتریس، ابتدا ابعاد عکس‌های ورودی به 64×64 کاهش داده می‌شود و سپس سیاه و سفید می‌شود که مولفه رنگ از آن حذف شود. در نتیجه ماتریس هر عکس، یک ماتریس 64×64 عضوی با درایه‌هایی با مقادیر بین صفر و یک متناظر با میزان روشنایی هر پیکسل می‌باشد؛ مقدار صفر معادل سیاه و یک معادل سفید می‌باشد. در شکل ۳-۴، فرایند این کار قابل مشاهده است.



شکل ۳-۴ - پیش‌پردازش عکس‌های سامانه

۳ - ۲ - ۱ - تبدیل ماتریس پیش‌پردازش شده به بردار

پس از آماده شدن ماتریس عکس پیش‌پردازش شده، این ماتریس در اختیار مازول بردارساز تصاویر قرار داده می‌شود که قرار است ورودی 40×96 مولفه‌ای را به برداری 128 مولفه‌ای تبدیل کند. این مازول از شبکه‌ی عصبی استفاده می‌کند و لایه‌ی نهایی آن دارای 128 خروجی است که مقدار هر Perceptron بین ۱ و ۰ است. در واقع با این نگاشت که، هر چهره، نقطه‌ای واقع در یک فضای 128 بعدی خواهد بود؛ در نتیجه می‌توان به کمک محاسبات برداری، چهره‌ها را با هم مقایسه نمود. این عمل توسطتابع بردارساز (V) انجام می‌شود. در شکل ۳-۵، خلاصه‌ای از فرایند بردار سازی چهره‌ها آورده شده است.

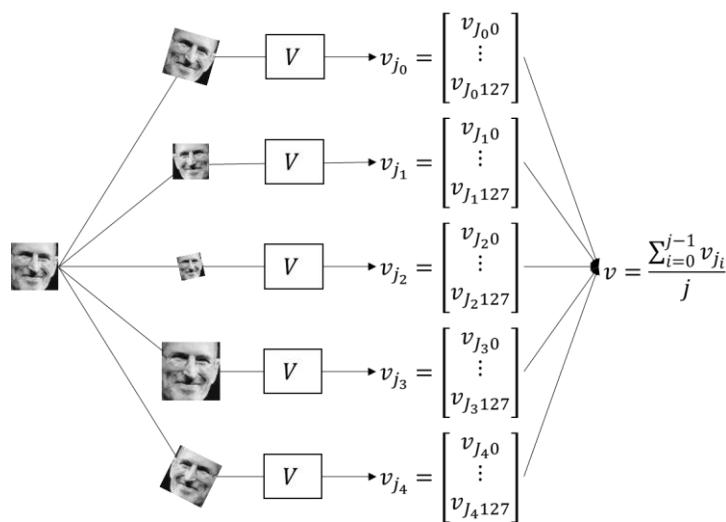


شکل ۳-۵ فرایند بردارسازی چهره‌ها

۳-۲-۱-۴-۳ نمونه‌برداری مجدد از تصویر برای بهبود کیفیت شناسایی چهره

برای بردارسازی چهره‌ها، پارامتری عددی تحت عنوان γ تعریف می‌شود. روند کار به این صورت است که به تعداد این پارامتر، عکس‌هایی از عکس اصلی که اعمال چرخش، جابه‌جایی، برگردانده شدن و بزرگنمایی به صورت تصادفی روی آن‌ها اعمال شده است را در نظر می‌گیریم، هر یک را به صورت هم‌رونده بردارسازی کرده و میانگین این بردارها به عنوان بردار نهایی در نظر گرفته می‌شود.

در شکل ۳-۶، نحوه محاسبه بردار یک چهره برای $j = 5$ نمایش داده شده است.

شکل ۳-۶ شیوه محاسبه بردار به شرط $j = 5$

۳-۲-۱-۴-۴ تشخیص چهره‌های کوچک‌تر

پارامتری عددی تحت عنوان γ تعریف می‌شود که تعیین می‌کند بزرگنمایی عکس ورودی برابر با چه نسبتی باشد. به عنوان مثال اگر $\gamma = 2$ باشد، ابتدا اندازه عکس ورودی دو برابر می‌شود و سپس پردازش می‌شود. این کار می‌تواند به یافتن چهره‌های کوچک‌تر در تصویر کمک کند، اما باعث می‌شود سامانه بسیار کندتر اجرا شود. رابطه

۱-۳ محاسبه بردار نهایی را توصیف می‌کند که در آن R تابع اعمال تغییرات تصادفی روی تصویر ورودی پیش‌پردازش شده، P تابع بزرگ‌نمایی تصویر ورودی به اندازه γ و V تابع بردارساز می‌باشد.

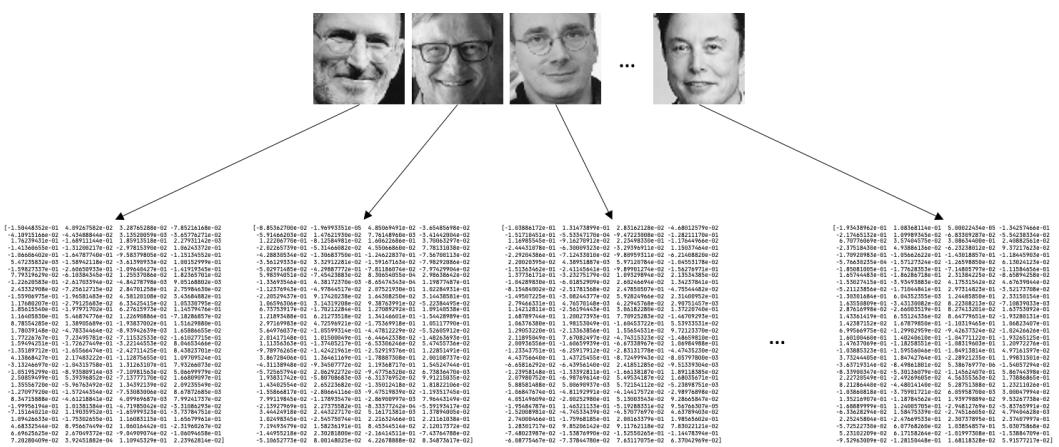
$$v = \frac{\sum_{i=0}^{j-1} V(R(P(img, \gamma)))}{j} \quad (1-3)$$

۲-۴-۲-۳ فرایند شناسایی چهره

در ادامه، فرایند شناسایی چهره را توصیف خواهد شد.

۳-۲-۴-۱ محاسبه و ذخیره بردار متناظر تصاویر کاربران سامانه

برای شناسایی چهره بر اساس کاربران سامانه، ابتدا نیاز است عمل بردارسازی روی تصاویر ذخیره شده از کاربران سامانه انجام شود. این مرحله تمرین مدل شناسایی چهره نامیده می‌شود و در آن تابع V را روی تمام عکس‌ها فراخوانی شده و بردارهای حاصل را ذخیره می‌شود. در شکل ۳-۷، این فرایند قابل مشاهده است.



شکل ۷-۳ بردارسازی تمام عکس‌های سامانه

زمانی که این اطلاعات در سامانه ذخیره شد، می‌توان چهره‌های جدید را شناسایی کرد.

۳-۲-۴-۲ محاسبه فاصله دو بردار

سامانه برای بررسی این‌که آیا دو چهره مطابقت دارند یا خیر، نیاز به یک معیاری برای مقایسه دو چهره دارد. از آنجا که در بخش‌های قبل گفته شد که می‌توان هر چهره را به یک بردار تبدیل کرد، برای مقایسه دو چهره کافی

است فاصله بردارهای متناظر دو چهره محاسبه شوند. برای این کار از معیار Norm-1 استفاده می‌شود که نحوه محاسبه آن در رابطه ۲-۳ آورده شده است.

$$\|W\|_1 = \sum_i^n |w_i| \quad (2-3)$$

همان‌طور که از رابطه فوق پیدا است، این معیار برابر جمع مولفه‌های یک بردار می‌باشد. برای محاسبه اختلاف دو بردار، کافی است نُرم ۱ اختلاف دو بردار را طبق رابطه ۳-۳ محاسبه شود.

$$D(X, Y) = \|X - Y\| \quad (3-3)$$

در نتیجه می‌توان گفت فاصله بردار متناظر با چهره موجود در عکس ورودی (V_i) و بردار متناظر با چهره n -ام ثبت شده در سامانه (V_n) برابر است با $\|V_i - V_n\|$

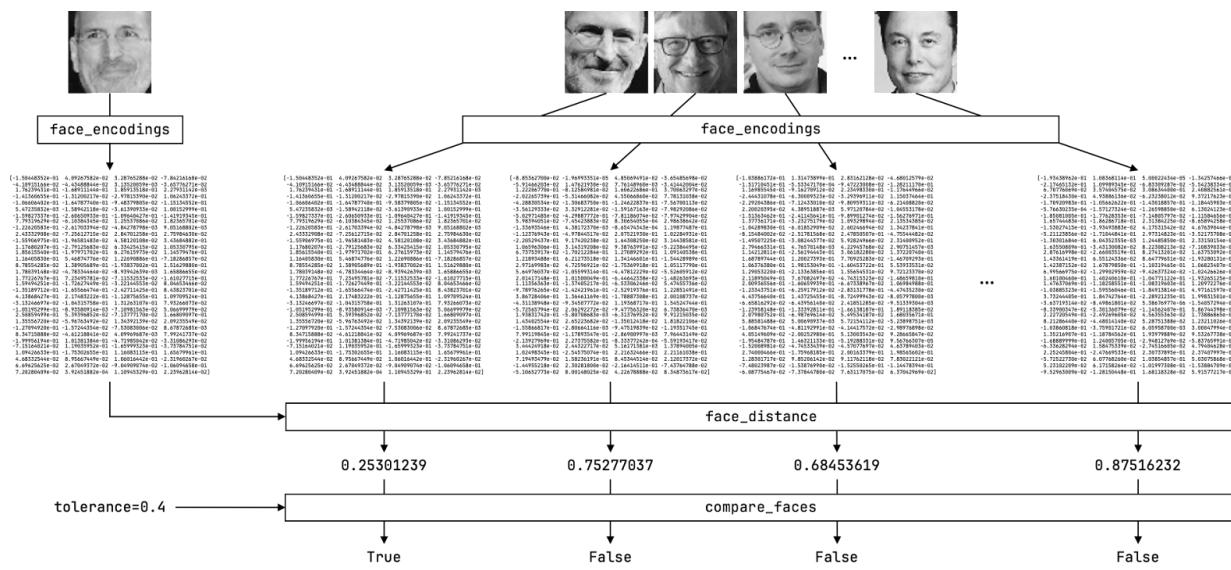
۳ - ۲ - ۴ - ۲ - ۳ شناسایی چهره بر اساس پارامتر t

در مرحله قبل نحوه محاسبه فاصله بردار متناظر چهره موجود در عکس ورودی و بردار متناظر با تمام چهره‌های ثبت شده در سامانه بیان شد. حال کافیست معیاری تعریف شود که مرز میان تطبیق چهره ورودی و چهره‌های سامانه را تعیین کند. این پارامتر t نامیده می‌شود. در صورتی که از میان تمام فواصل بردارهای محاسبه شده، کمترین فاصله از مقدار t کمتر باشد، ادعا می‌شود که چهره ورودی با چهره‌ی کاربر ثبت شده در سامانه که دارای کمترین فاصله است تطابق دارد. بدیهی است هرچه t کمتر باشد، احراز هویت سخت‌گیرانه‌تر خواهد بود. معمولاً این مقدار برای شناسایی چهره برابر 60° درنظر گرفته می‌شود، به این صورت که با این مقدار کاربر حتی با ماسک هم قابل شناسایی می‌باشد ولی در این پروژه، این مقدار را به 40° کاهش داده شده تا احراز هویت سخت‌گیرانه‌تر و دقیق‌تری داشته باشیم.

برای بررسی این موضوع تابع `isRecognized` به صورت رابطه ۴-۴ تعریف می‌شود که در این رابطه i همان چهره عکس ورودی و u بردار متناظر با چهره کاربر u می‌باشد.

$$isRecognized(i) = \begin{cases} u, \text{if } \min_{u \in U} D(V(i), V_u) < t \\ \text{none}, \text{if } \min_{u \in U} D(V(i), V_u) > t \end{cases} \quad (4-3)$$

خلاصه‌ای از فرآیند شناسایی چهره در شکل ۳-۸- قابل مشاهده می‌باشد. در این شکل t برابر با 0.4 می‌باشد و از آن جا که عکس اول از میان چهره‌های ثبت شده در سامانه فاصله‌ای کمتر از مقدار t با چهره مقابله دوربین دارد، سامانه مطابقت میان این چهره با چهره مقابله دوربین را تایید می‌کند. اگر یک عکس ورودی با چندین عکس از سامانه مطابقت داشته باشد، چهره با کمترین فاصله به عنوان چهره مطابقت داده شده شناسایی می‌شود.



شکل ۳-۸ خلاصه فرایند شناسایی چهره

۳-۲-۵-۱- تمرین مدل شناسایی چهره

هدف این بخش، طراحی ترتیب صفحه‌ها مشاهده شده توسط کاربر سامانه با توجه به اعمال او می‌باشد. اولین کار پس از شروع اجرای سامانه، تمرین مدل شناسایی چهره است. همان‌طور که قبل تر گفته شد، در این مرحله، بردارهای تمام تصاویر کاربران سامانه محاسبه شده و به همراه نام آن‌ها در سامانه ذخیره می‌شود.

۳-۲-۵-۲- معرفی صفحه‌ها موجود در سامانه

کاربر هنگام استفاده از سامانه با صفحاتی مواجه خواهد شد که در این بخش به توصیف آن‌ها پرداخته می‌شود.

۳ - ۲ - ۵ - صفحه اصلی

اولین صفحه که کاربر به محض شروع اجرای سامانه مشاهده می‌کند، صفحه اصلی می‌باشد. در این صفحه کاربر تصویر ضبط شده توسط دوربین را مشاهده می‌کند که در صورت وجود چهره‌ای در تصویر، اطراف آن را با یک مستطیل مشخص می‌کند. همان‌طور که گفته شد، این کار توسط مازول تشخیص چهره انجام می‌گردد. بالای این مستطیل، برای هر کاربر یک متن نمایش داده می‌شود که بر اساس مازول شناسایی چهره مشخص می‌شود. رنگ مستطیل و متن بالای آن نیز دارای معنی خاصی هستند که در جدول ۳ - ۱ مشخص شده است. طبق این جدول، اگر مستطیل به رنگ سبز یا زرد باشد، کاربر اجازه دارد درب سامانه را باز کند.

از این صفحه، کاربران به کمک صفحه کلید عددی یا صفحه کلید رایانه، می‌توانند عمل‌هایی روی سامانه انجام دهند. این عمل‌ها عبارتند از:

- اقدام برای باز کردن درب و وارد شدن به اتاق
- اقدام برای افزودن کاربر جدید به سامانه
- اقدام برای حذف کاربر موجود از سامانه

۳ - ۲ - ۵ - ۱ - صفحه ورود به اتاق

وقتی کاربر در صفحه اصلی قرار داشته باشد، با اقدام به ورودی به اتاق، صفحه‌ای را تحت عنوان صفحه ورود به اتاق مشاهده می‌کند. در این صفحه در صورتی که فقط یک کاربر در تصویر مقابل دوربین شناسایی شده باشد و آن کاربر یکی از کاربران تایید شده‌ی سامانه باشد، پیام خوش‌آمدگویی برای او ظاهر می‌شود و درب باز می‌گردد؛ در غیر این صورت، با پیام خطأ مواجه می‌شود. در هر دو حالت گفته شده، کاربر پس از مدتی به صفحه اصلی برگردانده می‌شود.

۳ - ۲ - ۵ - ۲ - صفحه وارد کردن شناسه مدیر

وقتی کاربر در صفحه اصلی قرار دارد، هنگامی که اقدام برای افزودن کاربر جدید به سامانه کند یا بخواهد یکی از کاربرهای موجود در سامانه را حذف کند، ابتدا به "صفحه وارد کردن شناسه مدیر" وارد می‌شود. در این صفحه در بالای صفحه از کاربر خواسته می‌شود که شناسه کاربری خود را وارد کند؛ در صورتی که کاربر عبارتی را وارد کند، می‌تواند آن را در صفحه ببیند. در صورتی که شناسه وارد شده، شناسه یک کاربر مدیر باشد، کاربر اجازه دارد وارد مرحله بعد یعنی "صفحه وارد کردن رمز عبور مدیر" شود، در غیر این صورت، با پیام خطأ روبرو می‌شود. از آن جا

که احتمال دارد شناسه کاربری یک مدیر برای یک کاربر غیر مدیر به هر دلیلی فاش شود، این امکان وجود دارد که یک کاربر غیر مدیر با داشتن شناسه یک کاربر مدیر، بتواند احراز هویت موفقی داشته باشد، که این ممکن است امنیت سامانه را دچار اشکال کند. برای رفع این مشکل، یک شرط در "صفحه وارد کردن شناسه مدیر" ذکر می‌شود به این صورت که "کاربر مدیری که اقدام به وارد کردن شناسه خود می‌کند، باید در همان لحظه مقابل دوربین سامانه قرار بگیرد و هویتش توسط مژول شناسایی چهره تایید شود تا بتواند وارد مراحل بعدی شود". با این موضوع، در صورت فاش شدن رمز عبور یک کاربر مدیر، شخص دیگری نمی‌تواند از هویت او سوءاستفاده کند.

۳ - ۲ - ۵ - ۳ صفحه وارد کردن رمز عبور مدیر

وقتی کاربر مدیر بتواند مرحله "وارد کردن شناسه مدیر" را با موفقیت پشت سر بگذارد، وارد "صفحه وارد کردن رمز عبور مدیر" خواهد شد. عملکرد این موضوع دقیقاً مانند صفحه قبل می‌باشد با این تفاوت که اولاً بجای شناسه کاربری، کاربر باید رمز عبور مدیرها را وارد کند، و دوماً عبارت وارد شده توسط کاربر روی صفحه مشاهده نمی‌شود و به ازای هر کارکتر ورودی، عبارت ^{**}"نمایش داده می‌شود. در این صفحه کاربر می‌تواند به صورت پیش‌فرض تا سه مرتبه رمز عبور اشتباه وارد کند و با پیام خطأ روبرو شود. پس از ورود رمز اشتباه چهارم، پیام خطای احراز هویت ناموفق نمایش داده می‌شود و صفحه اصلی نمایش داده می‌شود. تعداد دفعات مجاز برای وارد کردن رمز عبور اشتباه از طریق پیکربندی سامانه قابل تنظیم می‌باشد.

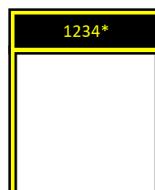
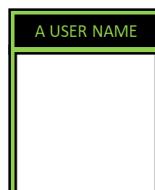
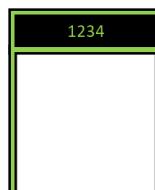
۳ - ۲ - ۵ - ۴ - ۳ صفحه وارد کردن نام (مخصوص افزودن کاربر جدید)

این امکان در سامانه وجود دارد که نام کاربران ذخیره شود. در صورتی که این اقدام در پیکربندی سامانه فعال باشد، کاربر پس از احراز هویت موفق، وارد "صفحه وارد کردن نام" می‌شود.

دو حال برای ورود نام کاربران جدید توسط مدیر سامانه وجود خواهد داشت:

- استفاده از صفحه کلید عادی: در این حالت از یک صفحه کلید عادی (که در رایانه‌ها از آن‌ها استفاده می‌شود) استفاده می‌شود که حروف زیادی مانند اعداد، حروف انگلیسی و سایر نمادها به عنوان ورودی قابل استفاده هستند.

جدول ۳-۱ حالات نمایش چهره‌های شناسایی شده در نمایش گر

سطح دسترسی			نمونه تصویر	رنگ کادر مستطیلی	متن نمایش داده شده در بالای کادر	دسته‌بندی چهره‌ها
حذف کاربر موجود	افزودن کاربر جدید	باز کردن درب				
✓	✓	✓	 	زرد رنگ	نام کاربر در کنار یک ستاره (*)	کاربر مدیر که نام او در سامانه ذخیره شده
					شناسه کاربری در کنار یک ستاره (*)	کاربر مدیر که نام او در سامانه ذخیره نشده
x	x	✓	 	سبز رنگ	نام کاربر	کابر عادی که نام او در سامانه ذخیره شده
					شناسه کاربری	کابر عادی که نام او در سامانه ذخیره نشده
x	x	x		قرمز رنگ	عبارت "UNKNOWN"	چهره فرد ناشناس

- استفاده از صفحه کلید عددی: در این حالت برای ورودی دادن به سامانه از صفحه کلیدهای ماتریسی عددی استفاده می‌شود که امکان ورود حروف انگلیسی در آن‌ها وجود نخواهد داشت. برای زمانی که کاربر قرار است از صفحه کلید عددی برای وارد کردن نام استفاده کند، باید این قابلیت را از طریق پیکربندی سامانه فعال کرد. با این کار شخص می‌تواند به کمک صفحه کلید عددی، حروف مورد نظر خود را نیز وارد سامانه کند. منطق مواجهه سامانه با ورودی‌های عددی در این حالت، مانند تلفن‌های همراه دکمه‌دار قدیمی می‌باشد که شیوه استفاده از آن در شکل زیر وجود دارد. به عنوان مثال، مدیر برای وارد کردن عبارت «main admin» باید کلیدها را به ترتیب زیر بفشارد: *#۲۳۶۴۴۴۶۶#۶۴۴۶۶۴۲۳

در شکل ۳-۹، نگاشت حروف انگلیسی در صفحه کلید عددی آورده شده است.



شکل ۳-۹ نگاشت حروف انگلیسی در صفحه کلید عددی برای وارد کردن متن در سامانه

۳ - ۲ - ۵ - ۳ - ۵ صفحه گرفتن عکس (مخصوص افزودن کاربر جدید)

در این صفحه، باید از کاربر جدید عکس گرفته شود. برای گرفتن عکس، باید حتماً چهره‌ای در مقابل دوربین قرار بگیرد که سامانه عکس‌برداری خود را انجام دهد، در غیر این صورت، پیامی نمایش داده می‌شود که به کاربر هشدار دهد چهره‌ای تشخیص داده نشده است.

در مواردی لازم است از هر کاربر چندین عکس گرفته شود که سامانه بتواند عملکرد بهتری داشته باشد، به عنوان مثال یک شخص عینکی می‌تواند یک عکس با عینک و عکس دیگری بدون عینک در سامانه ذخیره کند که بدون استفاده از عینک نیز بتواند توسط سامانه تشخیص داده شود. تعداد عکس‌هایی که از هر کاربر گرفته می‌شود به صورت پیش‌فرض برابر با ۲ است و این مقدار می‌تواند در پیکربندی سامانه تنظیم شود.

پس از اتمام فرایند تصویربرداری از کاربر، پیامکی برای مدیر سامانه ارسال شده و سپس سامانه وارد مرحله "تمرین مدل شناسایی چهره" می‌شود؛ یک صفحه نمایش داده می‌شود که از کاربر می‌خواهد صبر کند تا سامانه تمرین را انجام دهد و مجدداً صفحه اصلی نمایش داده شود.

۳ - ۲ - ۵ - ۶ صفحه وارد کردن شناسه (مخصوص حذف کاربر موجود)

برای حذف یک کاربر، باید او را به کمک یک مقدار یکتا پیدا کرد و سپس اطلاعات او را از سامانه حذف کرد. از آن جا که شناسه کاربری، تنها پارامتر یکتا کاربران سامانه می‌باشد، آن را به عنوان واسطه‌ای برای حذف کاربران در نظر گرفته می‌شود. هنگامی که در این مرحله، شناسه وارد شده متعلق به یکی از کاربران سامانه باشد، او از سامانه حذف خواهد شد و در غیر این صورت، پیام خطای نمایش داده می‌شود.

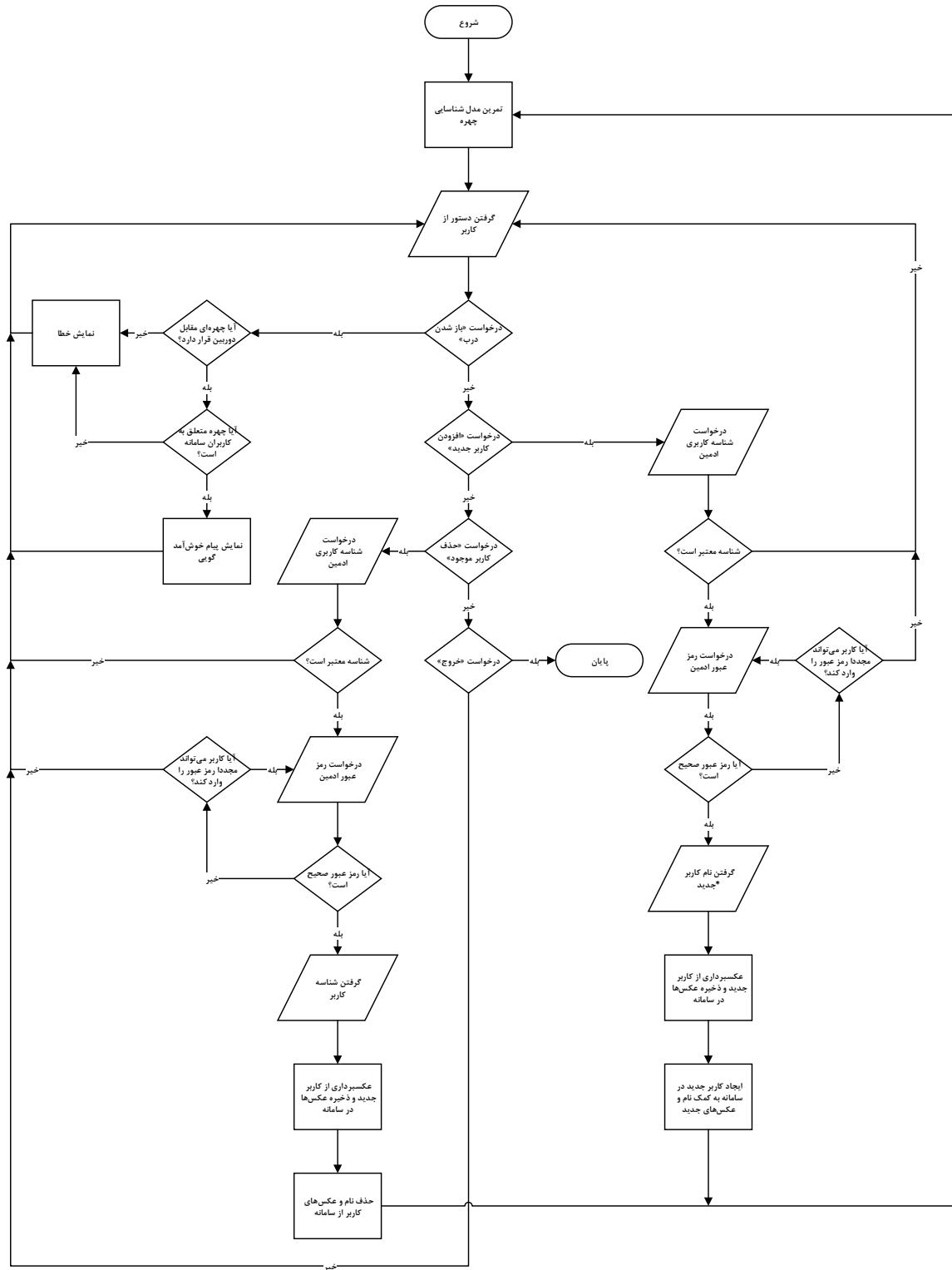
در صورت معتبر بودن شناسه وارد شده نیز پس از ارسال پیامک به مدیر، سامانه وارد مرحله "تمرین مدل شناسایی چهره" می‌شود که پس از نمایش یک صفحه برای نشان دادن تمرین سامانه، کاربر به صفحه اصلی برده می‌شود.

۳ - ۲ - ۵ - ۴ رسم نمودار جریان برنامه

در شکل ۳-۱۰، نمودار جریان این برنامه قابل مشاهده است که خلاصه‌ای از تمام مراحل گفته شده در مورد جریان اجرایی سامانه می‌باشد.

۳ - ۲ - ۵ - ۵ نحوه جابه‌جایی بین صفحه‌ها سامانه

کاربر توسط اقداماتی که در جدول ۳-۲ آورده شده است، می‌تواند بین صفحه‌ها جابه‌جا شود. در این جدول، مشخص شده در هر صفحه، با فشردن هر کلید، چه اتفاقی رخ می‌دهد.



شکل ۳ - ۱۰ نمودار جریان اجرایی برنامه

جدول ۳-۲ نحوه جابجایی میان صفحه‌ها سامانه

نتیجه	کلید فشرده شده		صفحه فعلی
	صفحه کلید عددی	صفحه کلید ریانه	
اقدام برای وارد شدن به اتاق	*	enter	صفحه اتاق صفحه ورود به اتاق
اقدام برای افزودن کاربر جدید به سامانه	A	a يا A	
اقدام برای حذف کاربر موجود از سامانه	D	d يا D	
خروج از برنامه	#	escape يا Q	
-	-	-	صفحه ورود به اتاق
بازگشت به صفحه اصلی	#	escape	صفحه وارد کردن صفحه وارد کردن مدیر
وارد کردن شناسه کاربری	اعداد	اعداد	
حذف آخرین کاراکتر وارد شده	D	backspace	
رفتن به صفحه وارد کردن رمز عبور مدیر	*	enter	
بازگشت به صفحه وارد کردن شناسه مدیر	#	escape	صفحه وارد کردن رمز صفحه وارد کردن مدیر
وارد کردن شناسه کاربری	اعداد	اعداد	
حذف آخرین کاراکتر وارد شده	D	backspace	
تلاش برای احرار هویت	*	enter	
بازگشت به صفحه اصلی	#	escape	صفحه وارد کردن نام صفحه وارد کردن نام (مجموعه افراد)
وارد کردن نام کاربر جدید	اعداد	اعداد و حروف	
حذف آخرین کاراکتر وارد شده	D	backspace	
رفتن به صفحه گرفتن عکس	*	enter	
گرفتن عکس / افزودن کاربر جدید	*	enter	صفحه گرفتن
بازگشت به صفحه اصلی	#	escape	صفحه وارد کردن صفحه وارد کردن (مجموعه افراد) صفحه وارد کردن (مجموعه افراد) مخصوص
وارد کردن شناسه کاربر موجود	اعداد	اعداد و حروف	
حذف آخرین کاراکتر وارد شده	D	backspace	
حذف کاربر	*	enter	

۴ - پیاده‌سازی سامانه

۴ - ۱ توسعه بخش نرم‌افزاری

در این بخش، شیوه توسعه نرم‌افزار پروژه توضیح داده خواهد شد.

۴ - ۱ - ۱ ساختار فایل‌های کد پروژه

در جدول ۴ - ۱، لیستی از فایل‌ها و پوشش‌های مهم پروژه توضیح داده شده است. لازم به ذکر است که علاوه بر موارد ذکر شده در این جدول، فایل‌های دیگری نیز وجود دارند اما تلاش شده تا در این جدول، تنها به بخش‌های کلیدی اشاره شود.

جدول ۴ - ۱ توضیح برخی از فایل‌ها و پوشش‌های مهم پروژه

نام فایل یا پوشش	توضیحات
admin_panel/	فایل سرور مربوط به تابلوی کنترلی مدیر و همچنین قالب صفحه‌ها این تابلوی کنترلی در این پوشش قرار دارند.
data/	اطلاعات مربوط به کاربران سامانه مانند نام و عکس‌های مربوط به هر شخص در این پوشش ذخیره می‌شوند.
utils/	مجموعه‌ای از توابع رایج استفاده شده در پروژه در این پوشش جمع آوری شده است. از انواع این توابع می‌توان به توابع مربوط به احراز هویت، توابع نمایش نوشه‌ها بر روی صفحه نمایش، کنترل قطعات سخت‌افزاری و مدیریت کاربران اشاره کرد.
configs.json	تنظیمات پیکربندی سامانه در این فایل قابل مشاهده و تغییر می‌باشد.
main.py	این فایل، فایل اصلی پروژه می‌باشد که سامانه را اجرای آن شروع به کار می‌کند.
requirement.txt	لیستی از کتابخانه‌های لازم برای این پروژه در این فایل جمع آوری شده که با اجرای دستور pip install -r requirements.txt، این کتابخانه‌ها نصب می‌شوند.

۴ - ۲ پیاده‌سازی کد تشخیص چهره

بسیاری از سازمان‌های بزرگ در حیطه فناوری که در زمینه هوش مصنوعی فعالیت می‌کنند، مدل‌های از پیش تمرین داده شده را در اختیار عموم قرار می‌دهند. برای تشخیص چهره از دو مدل از پیش تمرین داده شده استفاده می‌شود.

۴ - ۱ - ۲ - هیستوگرام شبکه‌های جهت‌دار

مدل‌های از پیش تمرین شده‌ی بسیاری وجود دارد که می‌توانند چهره‌های عکس‌ها را تشخیص دهند. برای تمرین این مدل‌ها روزها زمان صرف شده است که بتوانند چهره‌هایی با انواع رنگ پوست در زوایای مختلف تشخیص دهند؛ حتی در برخی از مدل‌ها چهره‌هایی که دارای عینک یا ماسک هستند نیز قابل تشخیص هستند. یکی از این مدل‌ها، مدل ارائه شده توسط opencv است که "haarcascade_frontalface_default.xml" نام دارد و در قالب فایل xml ذخیره شده است [۲۱]. برای استفاده از هیستوگرام شبکه‌های جهت‌دار در این مدل استفاده می‌شود.

۴ - ۱ - ۲ - شبکه عصبی پیچشی

برای استفاده از شبکه عصبی پیچشی در این پروژه از کتابخانه MTCNN استفاده شده است که به کمک کتابخانه TensorFlow پیاده‌سازی شده است [۲۲].

۴ - ۱ - ۳ - پیاده‌سازی کد شناسایی چهره

در این بخش، مختصری در مورد نحوه پیاده‌سازی شناسایی چهره در این پروژه توضیح داده شده است.

۴ - ۱ - ۳ - ۱ - تبدیل چهره‌ها به بردار

در این پروژه، به کمک تابع `face_encodings` که از مدل "compute_face_descriptor" در کتابخانه `dlib` استفاده می‌کند، هر چهره به یک بردار ۱۲۸ مولفه‌ای تبدیل می‌شود؛ درواقع هر چهره را به عنوان یک نقطه در یک فضای موهومی ۱۲۸ بعدی در نظر می‌گیرد. کد این تابع در شکل ۴ - ۱ آورده شده است که پارامتر "face_image" یا عکس ورودی، پارامتر "known_face_locations" که می‌تواند آرایه‌ای از چهره‌های تشخیص‌داده شده در عکس باشد و همچنین پارامتر "num_jitter" که همان \mathcal{Z} است را به عنوان ورودی خود دارا است. خروجی این تابع، آرایه‌ای از بردارهای ۱۲۸ بیتی با مقادیر موجود در بازه $[1, 1]$ که متناظر با چهره‌های موجود در عکس ورودی هستند می‌باشد. هنگامی که برنامه در مرحله «تمرین مدل شناسایی چهره» قرار دارد، این تابع برای تمامی عکس‌های ذخیره شده‌ی کاربران موجود در سامانه فراخوانی می‌شود و سامانه آن‌ها را به خاطر می‌سپارد. هنگام اتمام تمرین، در صورتی که چهره یک شخص مقابل دوربین سامانه قرار گیرد، برای این چهره نیز این تابع فراخوانی شده تا بتوان آن را با عکس‌های موجود در سامانه مقایسه کرد.

```
def face_encodings(face_image, known_face_locations=None, num_jitters=1):
    raw_landmarks = _raw_face_landmarks(face_image, known_face_locations)
    return [
        np.array(
            face_encoder.compute_face_descriptor(face_image, raw_landmark_set, num_jitters)
        ) for raw_landmark_set in raw_landmarks
    ]
```

شکل ۴-۱ کد بردارسازی چهره‌ها

۴-۱-۳-۲ مقایسه چهره‌ها

همان‌گونه که گفته شد، سامانه برای بررسی تطابق دو چهره، کافی است بردارهای متناظر دو چهره را مقایسه کند. برای این کار از تابع `compare_faces` که کد آن در شکل ۴-۲ آورده شده است استفاده می‌شود.

```
def compare_faces(known_face_encodings, face_encoding_to_check, tolerance=0.6):
    return list(face_distance(known_face_encodings, face_encoding_to_check) <= tolerance)
```

شکل ۴-۲ کد مقایسه چهره‌ها

این تابع سه پارامتر ورودی دارد:

۱. آرایه‌ای از بردارهای متناظر با چهره‌های موجود در سامانه که در مرحله تمرین مدل شناسایی چهره محاسبه می‌شوند،
۲. بردار متناظر با چهره‌ای که مقابل دوربین قرار گرفته و قرار است شناسایی شود
۳. پارامتر t

خروجی این تابع، آرایه‌ای به طول عکس‌های موجود در سامانه با مقادیر "True" یا "False" است که مشخص می‌کند فاصله عکس مقابل دوربین با عکس کاربران سامانه با t چه نسبتی دارد. به عنوان مثال، اگر فاصله بردار متناظر با چهره مقابل دوربین با بردار متناظر با اولین عکس موجود میان کاربران سامانه از مقدار t کمتر باشد، اولین مقدار موجود در آرایه‌ی خروجی تابع `compare_faces` برابر "True" است.

۴ - ۱ - ۳ - ۳ محاسبه فاصله میان بردارهای چهره‌ها

برای محاسبه فاصله میان دو چهره نیز باید از تابع `face_distance` که کد آن در شکل ۴ - ۳ آورده شده است استفاده شود که بردار کل چهره‌های ثبت شده در سامانه و بردار عکس مقابل دوربین را به عنوان ورودی می‌گیرد و فاصله آن‌ها را بر اساس نُرم ۱ برمی‌گرداند.

```
def face_distance(_face_encodings, face_to_compare):
    if len(_face_encodings) == 0:
        return np.empty(0)
    return np.linalg.norm(_face_encodings - face_to_compare, axis=1)
```

شکل ۴ - ۳ محاسبه فاصله میان چهره‌ها در کد

۴ - ۱ - ۴ - ۱ پیاده‌سازی جریان اجرایی برنامه

پس از تکمیل پیاده‌سازی ماثوله‌های تشخیص و شناسایی چهره، نوبت به پیاده‌سازی جریان اجرایی برنامه می‌رسد.

۴ - ۱ - ۴ - ۱ شروع برنامه

برای اجرای پروژه باید فایل `main.py` را از طریق دستور `python main.py` اجرا کرد. پس از اجرای این دستور، شکل ۴ - ۴ در ترمینال ظاهر می‌شود که تعداد عکس‌های سامانه و وضعیت تمرین مدل شناسایی چهره را نشان می‌دهد. پس از اتمام فرایند تمرین مدل، مدت زمان تمرین در ترمینال نمایش داده شده و سپس پنجره‌ای مشابه شکل ۴ - ۵ باز می‌شود که صفحه اصلی سامانه است و علاوه بر تصویر، زمان سامانه نیز در گوشه پایین سمت راست تصویر قابل مشاهده است. طبق جدول ۳ - ۱، شخص حاضر در شکل ۴ - ۵، مدیر این سامانه است و علاوه بر باز کردن درب، قادر است کاربران جدیدی در سامانه تعریف کند یا کاربران موجود در سامانه را حذف کند.

۴ - ۱ - ۴ - ۲ باز کردن درب در صورت احراز هویت موفق

کاربر برای اقدام به باز کردن درب، طبق جدول ۳ - ۲، باید کلید "enter" در صفحه کلید رایانه یا کلید "*" در صفحه کلید ماتریسی را فشار دهد و وارد صفحه ورود به اتاق شود. در این صفحه سه حالت ممکن است رخ دهد:



شکل ۴-۴ تصویر ترمینال شروع اجرای پروژه



شکل ۴-۵ تصویری از پروژه در صفحه اصلی

۱. کاربرانی که نام آن‌ها در سامانه ثبت شده است، در صورتی که اقدام باز کردن درب کنند پیام موقبیت برای آن‌ها نمایش داده می‌شود و درب برای آن‌ها باز می‌گردد. سپس پس از مدت سه ثانیه مجدداً صفحه اصلی نمایش داده می‌شود. تصویری از این حالت در شکل ۴-۶ آورده شده است.



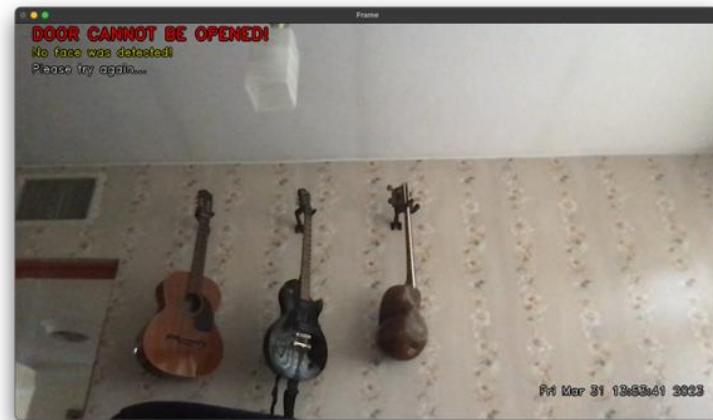
شکل ۴-۶ تصویر سامانه در حالتی که درب با موفقیت باز شده است

۲. در صورتی که شخص مورد نظر در سامانه شناخته نشود، با پیغام خطا مواجه می‌شود و درب برای او باز نمی‌شود. این خطا به مدت سه ثانیه در نمایش گر باقی می‌ماند و سپس به صفحه اصلی می‌رود. این حالت در شکل ۴-۷ نمایش داده شده است.



شکل ۴-۷ تصویری از سامانه که درب به دلیل ناشناس بودن چهره باز نمی‌شود

۳. در صورتی که هیچ چهره‌ای در مقابل سامانه نباشد، سامانه پیام «هیچ چهره‌ای تشخیص داده نشد!» را نمایش می‌دهد و درب را باز نمی‌کند. پس از سه ثانیه، سامانه صفحه اصلی نمایش می‌دهد. این حالت در شکل ۴-۸ قابل مشاهده است.



شکل ۴-۸ تصویر سامانه در حالتی که درب به علت عدم تشخیص چهره باز نمی‌شود

۴-۱-۳-۴ افزودن کاربر جدید

کاربر مدیر برای اقدام به افزودن کاربر جدید، طبق جدول ۳-۲، باید کلید "a" در صفحه کلید رایانه یا کلید "A" در صفحه کلید ماتریسی را فشار دهد و وارد صفحه وارد کردن شناسه مدیر شود. در صورتی که چهره مدیر با شناسه‌ای که وارد کرده هم خوانی داشته باشد، کاربر می‌تواند با استفاده از اطلاعات جدول ۳-۲ به صفحه وارد کردن رمز عبور برود. در غیر این صورت با پیام خطاب روبرو می‌شود. در شکل ۴-۹، حالات موفق و ناموفق وارد کردن شناسه مدیر قبل مشاهده است.



شکل ۴-۹-۹ حالات مختلف وارد کردن شناسه مدیر. سمت راست: موفق، سمت چپ: ناموفق

در صفحه وارد کردن رمز عبور مدیر نیز شرایط مشابه صفحه وارد کردن شناسه می‌باشد که در شکل ۴-۱۰، حالات مختلف این صفحه نیز آورده شده است.



شکل ۴-۱۰ حالت وارد کردن رمز عبور اشتباه

پس از احراز هویت موفق مدیر، او می‌تواند از کاربر بخواهد که مقابل سامانه قرار بگیرد و عکس خود را ثبت کند. در این فرایند، در بالای تصویر تعداد عکس‌های گرفته شده و تعداد کل عکس‌هایی که باید گرفته شوند وجود دارد. در شکل ۴-۱۱، نمونه‌ای از زمانی که سامانه در حالت عکس‌برداری است نمایش داده شده که طبق این عکس، کاربر در حال گرفتن عکس دوم خود می‌باشد.



شکل ۴-۱۱ عکس‌برداری از کاربر جدید

پس از ثبت تمام تصاویر، طبق شکل ۴-۱۲، از کاربر خواسته می‌شود که صبر کند تا اطلاعاتش در سامانه ذخیره شود.



شکل ۴-۱۲ تصویر نمایش داده شده از سامانه در حالت تمرین

۴-۱-۴ - حذف کاربران

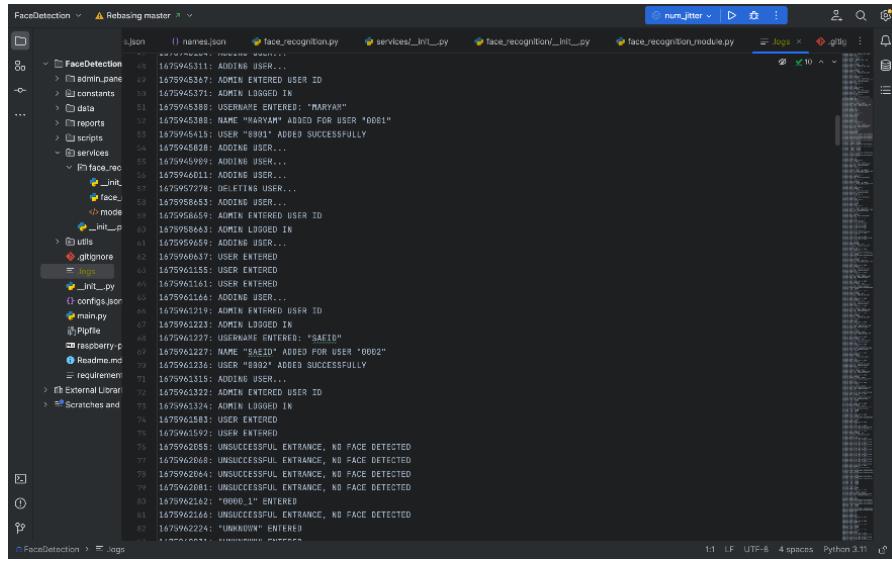
فرآیند حذف کاربران موجود نیز شباهت زیادی به افزودن آن‌ها به سامانه دارد. بدین صورت که ابتدا مدیر شناسه کاربری خود را وارد کرده و سپس رمز عبور مربوط به کاربران مدیر را وارد می‌کند. در صورت صحت اطلاعات وارد شده، مدیر شناسه کاربری مربوط به کاربری که باید حذف شود را وارد می‌کند. پس از حذف اطلاعات این کاربر، مدل شناسایی چهره مجدداً تمرین داده می‌شود و سامانه کار خود را از سر می‌گیرد.

۴-۱-۵ - ثبت وقایع برنامه به کمک ذخیره رخدادها^۱

برای پیگیری راحت‌تر ورود و خروج و همچنین افزودن و حذف کاربران توسط مدیرها، می‌توان از ذخیره رخداد برای تمام اتفاق‌ها مربوط به سامانه استفاده کرد. نمونه‌ای از یک فایل رخداد در شکل ۴-۱۳ نمایش داده شده است.

هر سطر از این رخدادها با یک timestamp معماري عددی برای مشخص کردن زمان شروع می‌شود. timestamp است که در واقع تعداد ثانیه‌های سپری شده از ساعت ۱۲ بامداد اول ژانویه سال ۱۹۷۰ میلادی به وقت گرینویچ می‌باشد. به دو دلیل به جای استفاده از یک متن که در آن توضیح تاریخ نوشته شده است، از timestamp استفاده می‌کنیم:

^۱ Logs



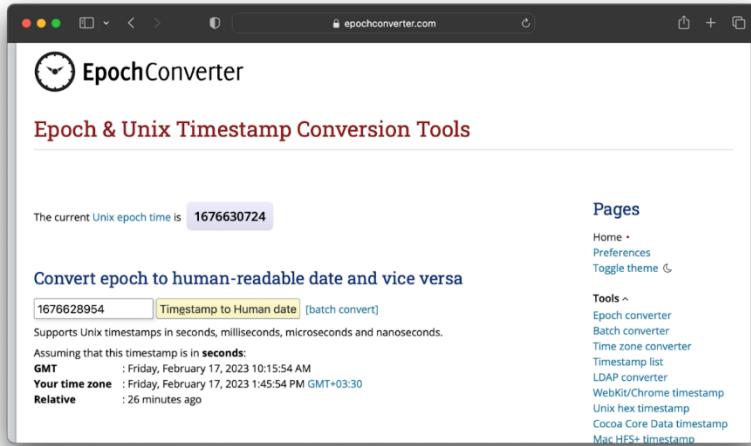
شکل ۴-۱۳ نمونه‌ای از رخدادهای سامانه

- ۱ - timestamp فضای کمتری اشغال می‌کند و در نتیجه فایل رخداد حجم کمتری خواهد داشت.
 - ۲ - دقت timestamp در مرتبه ثانیه می‌باشد.

برای بازیابی زمان مربوط به timestamp می‌توان از وبسایت www.epochconverter.com استفاده کرد.
به عنوان مثال زمان مربوط به timestamp با مقدار "۱۶۷۶۶۲۸۹۵۴" در شکل ۴-۱۴ آورده شده است.

در ادامه timestamp، هر سطر توضیحی از رویداد اتفاق افتاده آورده شده است. مواردی که رخداد آن‌ها نوشه می‌شوند عبارتند از:

- باز کردن موفق در توسط کاربر شناخته شده توسط سامانه
 - باز نشدن در پس از اقدام ناموفق کاربر ناشناس
 - اقدام برای ورود به سیستم به عنوان مدیر
 - وارد کردن رمز عبور اشتباه توسط مدیر
 - افزودن کاربر توسط مدیر
 - حذف کاربر توسط مدیر



شکل ۴-۴ نمایی از سایت EpochConvertoer در حال نشان داده "۱۶۷۶۶۲۸۹۵۴" timestamp

۱-۶ کنترل سامانه به کمک فایل پیکربندی

برای پیکربندی تنظیمات سامانه یک فایل با فرمت JSON^۱ با نام configs.json در ریشه‌ی پروژه وجود دارد که به کمک این فایل می‌توان تنظیمات سامانه را تغییر داد. در شکل زیر نمونه‌ای از محتوای این فایل نمایش داده شده است.

توضیحات هر یک از موارد موجود در این فایل، در

۱-۷ توسعه تابلوی کنترلی مدیر

برای مشاهده راحت اطلاعات موجود در سامانه، پنلی طراحی شده که به کمک آن می‌توان مواردی نظری تنظیمات سامانه، کاربران و همچنین رخدادهای ثبت شده را در آن مشاهده کرد. این تابلوی کنترلی توسط سروری که با کتابخانه Flask توسعه داده شده ایجاد شده و می‌توان این سرور را به کمک دستور flask run فعال و از آن استفاده کرد. این دستور، سرور را در آدرس آی‌پی ۱۲۷.۰.۰.۱ و درگاه ۵۰۰۰ ایجاد می‌کند و می‌توان به رفتن به این آدرس آن را مشاهده کرد. عکس صفحه اصلی این تابلوی کنترلی در شکل ۴-۱۵ آورده شده است.

^۱ JavaScript Object Notation

جدول ۴-۲ آورده شده است.

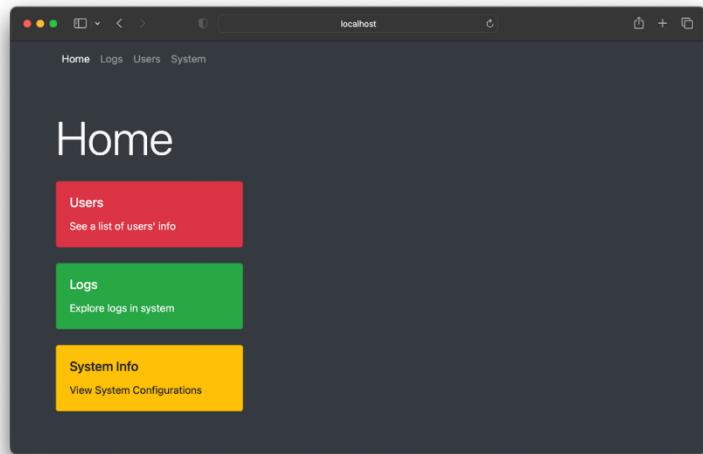
۴-۱-۸ توسعه تابلوی کنترلی مدیر

برای مشاهده راحت اطلاعات موجود در سامانه، پتلی طراحی شده که به کمک آن می‌توان مواردی نظیر تنظیمات سامانه، کاربران و همچنین رخدادهای ثبت شده را در آن مشاهده کرد. این تابلوی کنترلی توسط سروری که با کتابخانه Flask توسعه داده شده ایجاد شده و می‌توان این سرور را به کمک دستور flask run فعال و از آن استفاده کرد. این دستور، سرور را در آدرس آی‌پی^۱ ۱۲۷.۰.۰.۱ و درگاه ۵۰۰۰ ایجاد می‌کند و می‌توان به رفتن به این آدرس آن را مشاهده کرد. عکس صفحه اصلی این تابلوی کنترلی در شکل ۴-۱۵ آورده شده است.

^۱ IP Address

جدول ۴-۲ توضیحات فایل پیکربندی تنظیمات

تنظیمات	نام تنظیمات در فایل پیکربندی	مقادیر پیش‌فرض	توضیحات تکمیلی
مسیر ذخیره‌سازی تصاویر	images_path	data/images/	-
تعداد عکس موجود از هر کاربر	images_per_user	2	-
ذخیره‌سازی نام کاربران جدید	store_usernames	true	-
استفاده از صفحه کلید عددی برای ذخیره نامها	using_numeric_keypad	false	در صورت true بودن، فرض بر این است که کاربر به کمک صفحه کلید عددی قصد وارد کردن نام را دارد
تعیین دوربین تشخیص چهره	camera_arg	0	می‌تواند یک عدد باشد که شماره دوربین متعلق به رایانه است. همچنین می‌تواند آدرس آی پی یک دوربین تحت شبکه باشد.
مسیر ذخیره‌سازی نام کاربران	names_data	data/names.json	-
لیست مدیرها	admin_users	["0000"]	-
رمز عبور مدیرها	admin_encrypted_password	"==MTIzNA"	به دلایل امنیتی، ابتدا رمز عبور را به کمک الگوریتم بیس ۶۴ کدگذاری کرده و سپس در این فایل قرار می‌دهیم. در عمل بیس ۶۴ داده‌های دودویی ۸ بیتی را به قالب ۷ بیتی تبدیل می‌کند؛ یعنی معمولاً ۳ بایت ۸ بیتی، تبدیل به ۴ بایت ۷ بیتی می‌شوند و برای نمایش تنها از حروف -A- Z و a-z / و .۰-۹ استفاده می‌شود.
تعداد دفعات مجاز ورود رمز عبور	wrong_password_limit	3	-
اشتباه			
احراز هویت سختگیرانه	strict_authentication	true	-
آدرس فایل رخدادها	file_path	true	-
نوشتن رخداد در فایل	log_in_file	true	-
نوشتن رخداد در کنسول	log_in_console	true	-
نوشتن رخداد افزودن کاربر جدید	use_logging_in_add_user	true	-
نوشتن رخداد حذف کاربر موجود	use_logging_in_delete_user	true	-
نوشتن رخداد ورود مدیر به سامانه	use_logging_in_admin_login	true	-
نوشتن رخداد نوشتن رمز عبور	use_logging_in_wrong_password	true	-
اشتباه			
پارامتر آستانه تطبیق (t)	tolerance	0.6	-
تعداد نمونه برداری از چهره (j)	num_jitter	2	-
پارامتر تشخیص چهره‌های کوچک (v)	number_of_times_to_upsample	2	-
مدل تشخیص چهره	model	"hog"	در حال حاضر دو مقدار "hog" و "cnn" پذیرفته می‌شود.
تعیین ارسال پیامک	send_sms	false	-
شماره همراه	phone_number	"989123456789"	-



شکل ۴-۱۵ صفحه اصلی تابلوی کنترلی مدیر

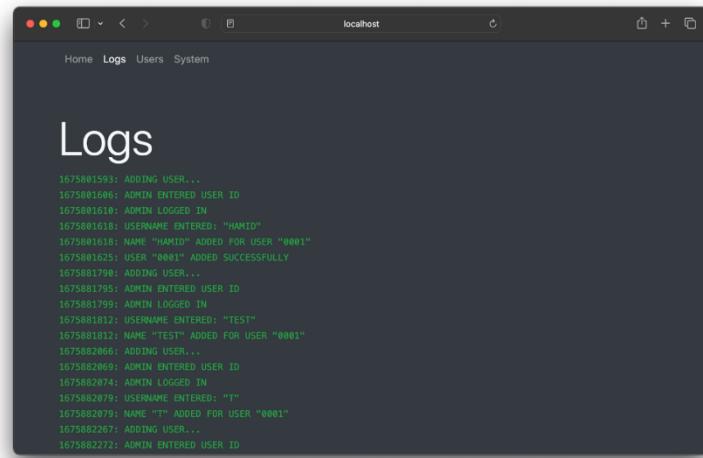
از این صفحه می‌توان به سه صفحه دیگر جایه‌جا شد که در این قسمت این صفحه‌ها را معرفی می‌کنیم.

۴-۱-۸-۱ صفحه مشاهده رخدادهای سامانه

تمام رخدادهای ثبت شده در سامانه در این صفحه قابل مشاهده هستند. تصویر گرفته شده از این صفحه در شکل ۴-۱۶ آورده شده است.

۴-۱-۸-۲ صفحه مشاهده اطلاعات کاربران

اطلاعات کاربران سامانه اعم از شناسه کاربری، نام، نقش و همچنین پیش‌نمایش عکس‌های ذخیره شده از کاربران در این صفحه آورده شده است. تصویر گرفته شده از این صفحه در شکل ۴-۱۷ قابل مشاهده می‌باشد.



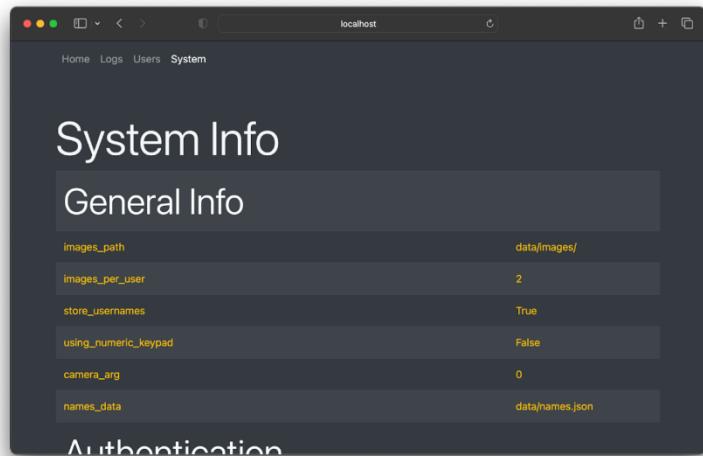
شکل ۱۶ - صفحه رخدادها در تابلوی کنترلی مدیر

id	role	name	images
0000	Administrator	Amirhossein	
0001	User	Elon	
0002	User	Linus	
0003	User	Bill	
0004	User	Mark	

شکل ۱۷ - صفحه مشاهده اطلاعات کاربران در تابلوی کنترلی مدیر

۴ - ۱ - ۸ - ۳ - صفحه مشاهده اطلاعات سامانه

در این صفحه محتوای فایل پیکربندی آورده شده است. تصویر گرفته شده از این صفحه در شکل ۱۸ - ۴ قابل مشاهده می‌باشد.

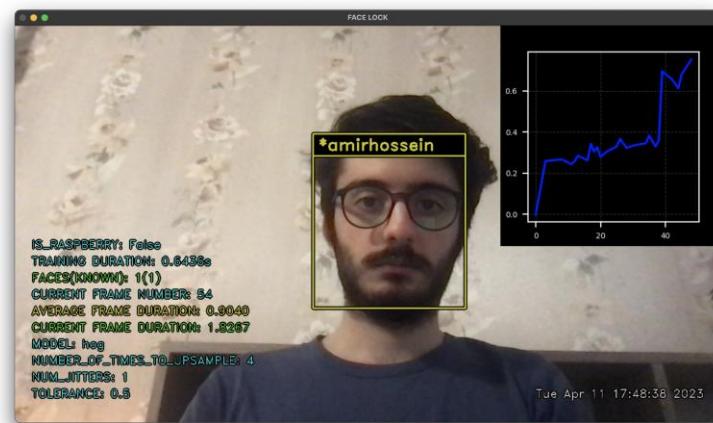


شکل ۱۸-۴ مشاهده اطلاعات سامانه در تابلوی کنترلی مدیر

۱-۹-۱ اجرای سامانه در حالت اشکال‌زدایی

علاوه بر روش گفته شده، می‌توان برنامه را در حالت اشکال‌زدایی اجرا کرد. برای این کار باید برنامه را با دستور `python main.py --debug true` اجرا کرد. با این کار، روی نمایش‌گر سامانه، اطلاعات تکمیلی سامانه در گوشه پایین سمت چپ نمایش‌گر مشاهده می‌شود. در شکل ۱۹-۴، نمونه‌ای از عکس نمایش‌گر سامانه در حالت اشکال‌زدایی نمایش داده شده است. همان‌طور که مشاهده می‌شود، اطلاعات زیر در این صفحه قابل مشاهده هستند:

- مدت زمان تمرین مأذول شناسایی چهره
- تعداد چهره‌های موجود در تصویر و تعداد چهره‌های افراد عضو سامانه
- شماره قاب از زمان اجرای برنامه در حالت اشکال‌زدایی
- میانگین مدت زمان ۲۰ قاب اخیر
- مدت زمان قاب قبلی که می‌تواند در دورنگ نمایش داده شود؛ در صورتی که مدت زمان قاب فعلی نسبت به قاب قبلی کمتر شده باشد، نوشته به رنگ سبز و در غیر این صورت، به رنگ قرمز نمایش داده می‌شود.
- مدل هوش مصنوعی استفاده شده برای تشخیص چهره
- پارامترهای γ ، t و z



شکل ۴-۱۹- مشاهده تاثیر افزایش ۷ بر مدت زمان هر قاب در حالت اشکال‌زدایی

۴-۱-۱۰ کنترل ماژول‌های سخت‌افزاری در کد پروژه

برای کنترل ماژول‌های سخت‌افزاری متصل به بورد Raspberry Pi، از کتابخانه RPi.GPIO استفاده می‌کنیم. برای این کار ابتدا در کد خود باید این کتابخانه را به کمک `import RPi.GPIO as GPIO` اضافه کرد. سپس می‌توان به کمک روش `setup` از `GPIO`، پایه‌های موردنیاز را تعریف کرد. این روش دو پارامتر ورودی دارد. اولین پارامتر، شماره پایه و دومین پارامتر، تعیین می‌کند که پایه ورودی است یا خروجی. برای قطعاتی مانند صفحه کلید عددی که ورودی سامانه هستند، این پارامتر برابر با `GPIO.IN` و برای قطعاتی مانند آژیر که خروجی سامانه هستند، برابر با `GPIO.OUT` قرار داده می‌شوند. برای کنترل مقادیر پایه‌ها، می‌توان از روش `GPIO.output` استفاده کرد که این روش هم دو پارامتر ورودی دارد که اولین پارامتر، شماره پایه و دومین پایه، مقداری است که قرار است به پایه نسبت داده شود. در این پروژه با مقادیر صفر و یک این مقداردهی انجام شده است.

برای نمونه در شکل ۴-۲۰، کد مربوط به کنترل آژیر آورده شده است.

۴-۲ پیاده‌سازی مدار پروژه

در این بخش، به نحوه راهاندازی بورد Raspberry Pi و پیاده‌سازی سخت‌افزار پروژه پرداخته می‌شود.

```

import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(BUZZER_PIN, GPIO.OUT)

def buzz_on():
    GPIO.output(BUZZER_PIN, HIGH)

def buzz_off():
    GPIO.output(BUZZER_PIN, LOW)

def buzz(n):
    for i in range(n):
        buzz_on()
        time.sleep(BUZZER_DELAY)
        buzz_off()
        time.sleep(BUZZER_DELAY)

```

شکل ۴-۲۰ کد مربوط به کنترل آژیر

۴-۲-۱- راه اندازی بورد Raspberry Pi

برای اجرای کد پروژه روی بورد Raspberry Pi، باید اقداماتی پیش از اجرای کد انجام شود. در این بخش این اقدامات شرح داده می‌شوند.

۴-۲-۱-۱- نصب سیستم عامل

برای این پروژه از سیستم عامل Raspbian استفاده شده است که ابتدا باید آن را روی یک کارت حافظه نوشته و این کارت را در بورد قرار داد. برای دریافت این سیستم عامل ابتدا باید فایل آن را از سایت <https://www.raspberrypi.com/software> بارگیری کرد. تصویر این وبسایت در شکل ۴-۲۱ آورده شده است.

جدول ۴-۳ نحوه اتصال قطعات به بورد

نام ماژول	نام متغیر در کد	پایه متناظر در بورد
ماژول دوربین	-	CSI
صفحه کلید عددی	L1	5
	L2	6
	L3	13
	L4	19
	C1	12
	C2	16
	C3	20
	C4	21
آژیر	BUZZER_PIN	4
دیود نشرده‌نده نور	LED_PIN	2
Sim900	-	14(TX)
	-	15(RX)
قفل	LOCK_PIN	24



شکل ۴-۲۱ تصویر وبسایت مربوط به راهاندازی سیستم‌عامل Raspbian

پس از دریافت این برنامه، باید آن را اجرا کرد و پس از انتخاب سیستم‌عامل دلخواه و همچنین کارت حافظه‌ی موردنظر، عمل نوشتن سیستم‌عامل رو کارت حافظه را انجام داد. محیط این برنامه در شکل ۴-۲۲ قابل مشاهده است.



شکل ۴-۲۲ مراحل نصب سیستم‌عامل بر روی Raspberry Pi

پس از اتمام این کار می‌توان کارت حافظه را در بخش پایینی بورد قرار داد و در این صورت با اتصال بورد به رایانه از طریق درگاه USB نوع C به عنوان منبع تغذیه، می‌توان از این سیستم‌عامل استفاده کرد.

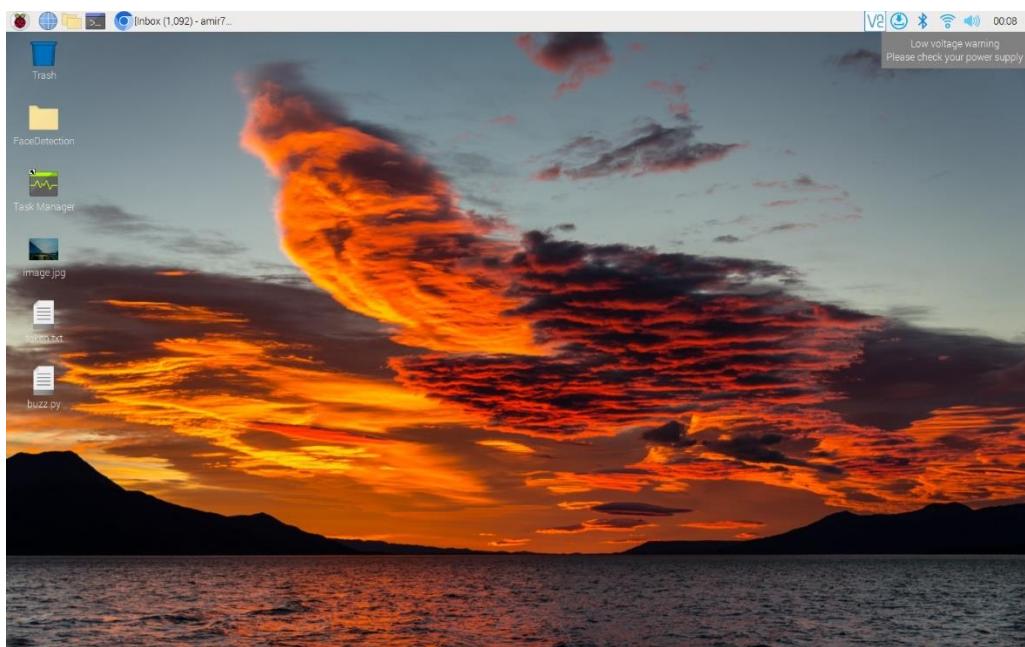
۴ - ۱ - ۲ - اتصال بورد Raspberry Pi به نمایشگر

می‌توان با اتصال بورد به نمایشگر از طریق درگاه micro-HDMI برای نمایش گرافیکی آن و همچنین از موشواره و صفحه کلید جهت کنترل کردن بورد مانند یک رایانه‌ی معمولی بهره بورد. در شکل ۴-۲۳، تصویری از محیط این سیستم‌عامل هنگام اجرا روی بورد Raspberry Pi قابل مشاهده است. همچنین در شکل ۴-۲۴ می‌توان نحوه اتصال موارد لازم برای روشن شدن بورد را مشاهده کرد.

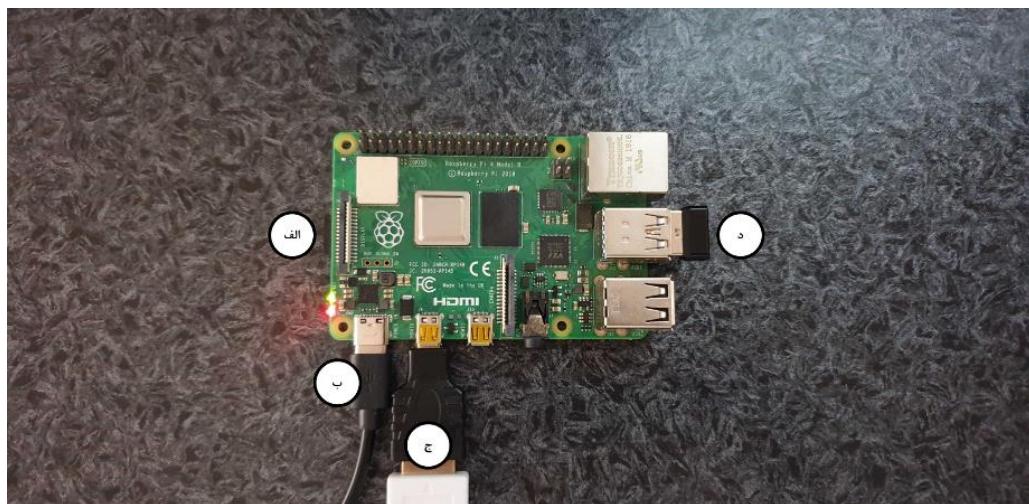
۴ - ۲ - ۳ - نصب کتابخانه‌های موردنیاز برای اجرای کد پروژه روی بورد Raspberry Pi

برای امکان استفاده از کتابخانه OpenCV روی بورد Raspberry Pi، نیاز است بسته^۱‌هایی را روی سیستم‌عامل Raspbian نصب شود که شرایط کار کردن صحیح این کتابخانه را فراهم کند. برای این کار توضیحاتی در فایل raspberry-pi-instruction.md آورده شده است که دستورات موردنیاز به همراه توضیحات مربوط به هر دستور را می‌توان مشاهده کرد. قسمتی از این فایل در شکل ۴-۲۵ قابل مشاهده می‌باشد.

^۱ Package

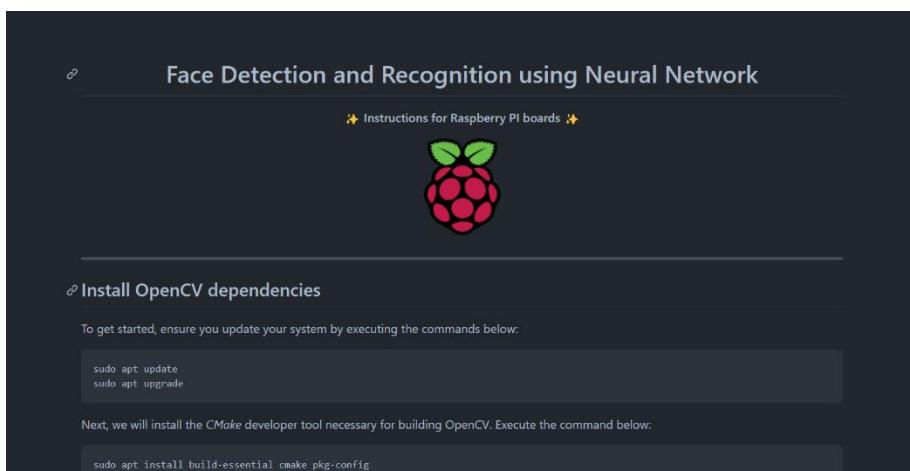


شکل ۴-۲۳ تصویر سیستم‌عامل Raspbian روی بورد Raspberry Pi



شکل ۴-۲۴ اتصالات اولیه بورد

الف: کارت حافظه در زیر بورد برای سیستم‌عامل، ب: اتصال به تغذیه، ج: اتصال به نمایشگر، د: اتصال موشواره و صفحه کلید از طریق درگاه USB



شکل ۴-۲۵ بخشی از توضیحات تکمیلی مربوط به اجرای پروژه روی بورد Raspberry Pi

برای اجرای پشت سر هم این دستورات، می‌توان دستورات موجود در فایل config_raspberry_pi.sh را به کمک دستور sh config_raspberry_pi.sh اجرا کرد.

۴ - ۱ - ۲ - ۳ - ۴ اتصال مژوپهای سختافزاری به بورد Raspberry Pi

در این بخش به شیوه اتصال قطعات مورد استفاده در پروژه به بورد Raspberry Pi طبق جدول ۴-۳ اشاره خواهد شد.

۴ - ۱ - ۲ - ۳ - ۴ اتصال مژول دوربین

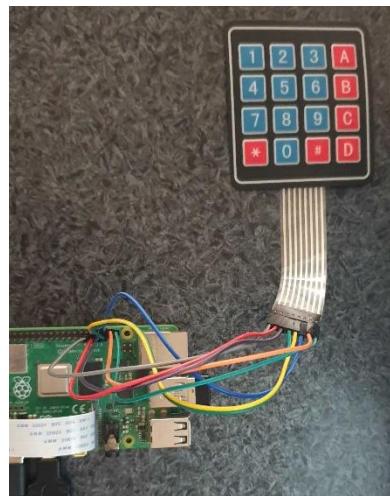
برای اتصال دوربین به بورد، کافی است آن را به یکی از ۲ سوکت رابط سریالی دوربین متصل کنیم. در شکل ۴-۲۶، اتصال مژول دوربین به بورد نمایش داده شده است.



شکل ۴-۲۶ شیوه اتصال دوربین

۴ - ۱ - ۲ - ۴ - اتصال صفحه کلید عددی

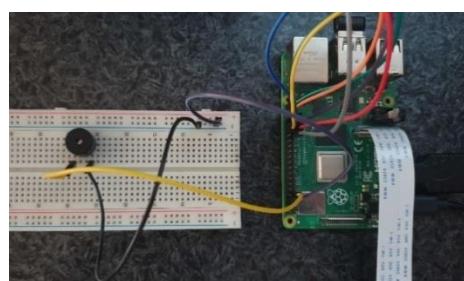
از آن جایی که در این پروژه از صفحه کلید عددی ماتریسی 4×4 استفاده می‌شود، به ۸ پایه از بورد برای کنترل صفحه کلید عددی نیاز است؛ ۴ پایه مخصوص سطرهای صفحه کلید عددی هستند که از نوع ورودی می‌باشند، چهار پایه دیگر از نوع خروجی هستند و برای ستون‌های صفحه کلید عددی مورد استفاده قرار می‌گیرند. در شکل ۴-۲۷، نحوه اتصال صفحه کلید عددی به بورد طبق اطلاعات موجود در جدول ۴-۳ آورده شده است.



شکل ۴-۲۷ - نحوه اتصال صفحه کلید عددی

۴ - ۱ - ۲ - ۳ - اتصال آژیر

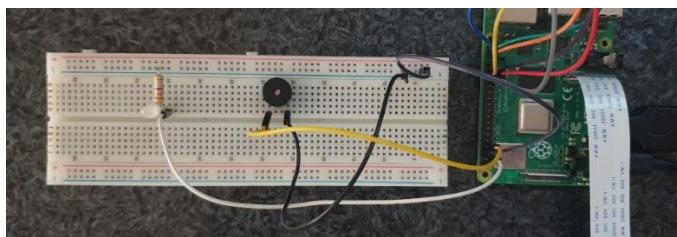
برای اتصال باز به بورد، یک سر آن به پایه در نظر گرفته شده برای آژیر در بورد و سر دیگر آن به پایه زمین بورد متصل می‌شود. در شکل ۴-۲۸، نحوه اتصال آژیر به بورد طبق اطلاعات موجود در جدول ۴-۳ آورده شده است.



شکل ۴-۲۸ - نحوه اتصال آژیر

۴ - ۱ - ۲ - ۴ - اتصال دیود نشرده‌نده نور

برای اتصال دیود نشرده‌نده نور به بورد، کافی است سر مثبت آن را به پایه در نظر گرفته شده برای دیود نشرده‌نده نور در بورد سر دیگر آن را به واسطه یک مقاومت ۳۳۰ اهمی، به پایه زمین بورد متصل شود. در شکل ۴ - ۲۹ نحوه اتصال دیود نشرده‌نده نور به بورد طبق اطلاعات موجود در جدول ۴ - ۳ قابل مشاهده است.



شکل ۴ - ۲۹ نحوه اتصال به دیود نشرده‌نده نور

Sim900 - ۱ - ۲ - ۴ - اتصال

برای اتصال Sim900 به بورد، باید ابتدا یک سیم‌کارت در آن قرار داده و پایه‌های TX و RX آن را مطابق جدول ۴ - ۳ به بورد متصل شود. سپس با اتصال آن به پایه‌های زمین و ۵ ولت بورد، می‌توان از آن استفاده کرد. باید توجه داشت که برای استفاده از این مژول در بورد Raspberry Pi، باید در تنظیمات بورد، رابط سریال آن را فعال کرد.

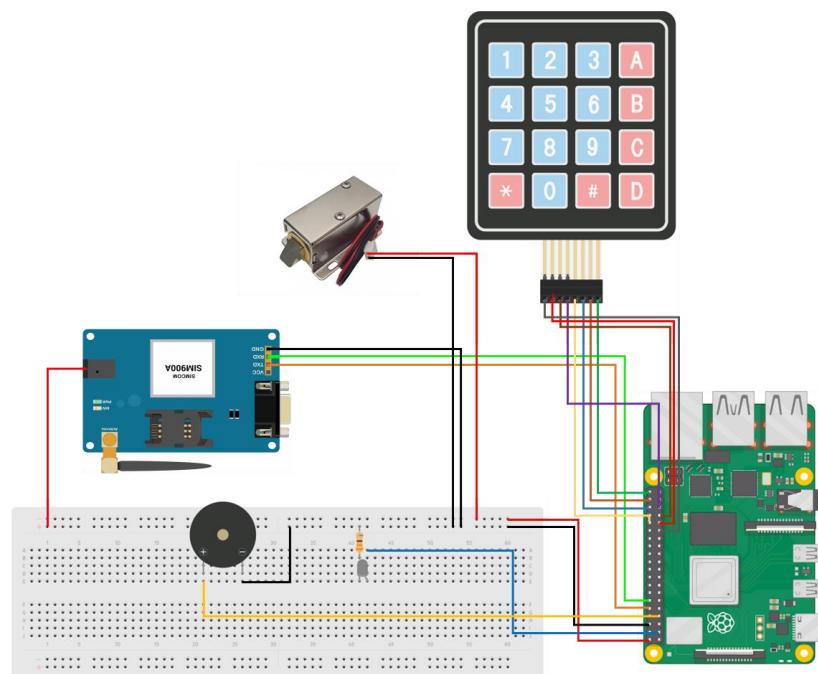
۴ - ۱ - ۲ - ۶ - اتصال قفل Solenoid

در صورت استفاده از قفل ۵ ولتی، می‌توان به راحتی یک سر آن را به پایه زمین بورد و سر دیگر آن را به پایه اختصاص داده شده به بورد طبق جدول ۴ - ۳ متصل کرد. در صورت استفاده از قفل با ولتاژ کاری ۱۲ ولت، باید به کمک یک رله این کار را انجام داد. شمای مدار پروژه در شکل ۴ - ۳۰ قابل مشاهده است.

۴ - ۲ - اجرای کد پروژه روی بورد Raspberry Pi

پس از مراحل گفته شده، می‌توان کد پروژه که در یک مخزن¹ GitHub² ذخیره شده است را به کمک دستور git clone دریافت کرد و پس از نصب کتابخانه‌های موردنیاز آن، سامانه را با کمک دستور python main.py اجرا

¹ Repository
² GitHub



شکل ۴-۴ شمای مدار پروژه

کرد. برای مشاهده تابلوی کنترلی مدیر نیز باید دستور Flask run را اجرا نمود. در شکل ۴-۳۱، تصویری از نمایشگر متصل به بورد Raspberry Pi مشاهده می‌شود که در حال اجرای سامانه می‌باشد.

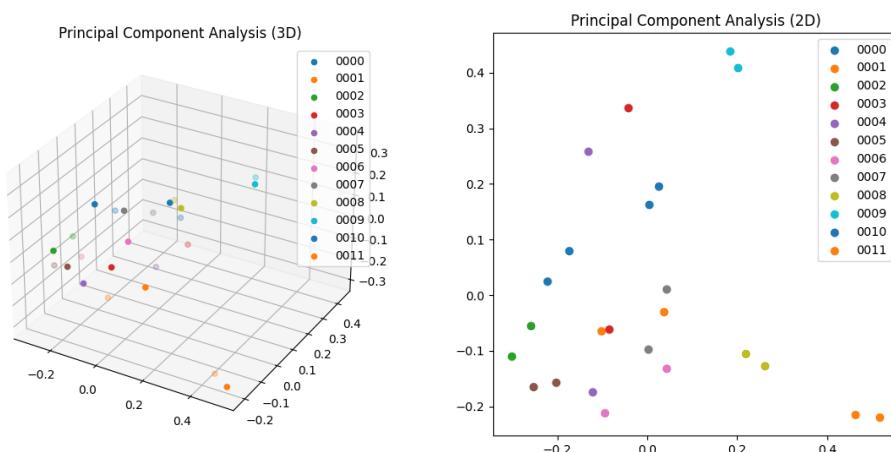


شکل ۴-۳۱ تصویری از اجرای پروژه روی بورد Raspberry Pi

۵ - ارزیابی پروژه

۵ - استفاده از تحلیل مولفه‌های اصلی برای بررسی کیفیت شناسایی چهره

همان‌طور که گفته شد، هر چهره یک نقطه در فضای ۱۲۸ بعدی می‌باشد که انتظار می‌رود نقاط مربوط به یک شخص در این فضا در نزدیکی یکدیگر قرار داشته باشند. برای مصورسازی و درک بهتر این موضوع، نیاز است داده‌ها را به نحوی تغییر داده شوند که قابل رسم باشد. برای این کار از تحلیل مؤلفه‌های اصلی^۱ استفاده می‌شود که تبدیلی در فضای برداری است و می‌تواند مجموعه داده‌های بزرگ با تعداد بعد یا ویژگی زیاد را تحلیل کند. تفسیرپذیری داده‌ها با حفظ حداقل مقدار اطلاعات انجام دهد و تجسم داده‌های چند بعدی را فراهم می‌کند. تحلیل مؤلفه‌های اصلی، در واقع یک روش آماری برای کاهش ابعاد یک مجموعه داده است. این کار با تبدیل خطی داده‌ها به یک سیستم مختصات جدید انجام می‌شود که بیشتر تغییرات در داده‌ها را می‌توان با ابعاد کمتری نسبت به داده‌های اولیه توصیف کرد. در شکل ۵-۱، خروجی روش تحلیل مؤلفه‌های اصلی برای دو و سه بعد آورده شده است. همان‌طور که در این شکل مشاهده می‌شود، عکس‌هایی که از هر شخص گرفته می‌شوند با یک رنگ خاص نمایش داده شده‌اند و همان‌طور که انتظار می‌رود، نقاط متناظر با چهره‌های هر شخص در این فضاهای برداری در مجاورت یکدیگر قرار دارند.



شکل ۵-۱ تحلیل مولفه‌های اصلی چهره‌ها در دو و سه بعد

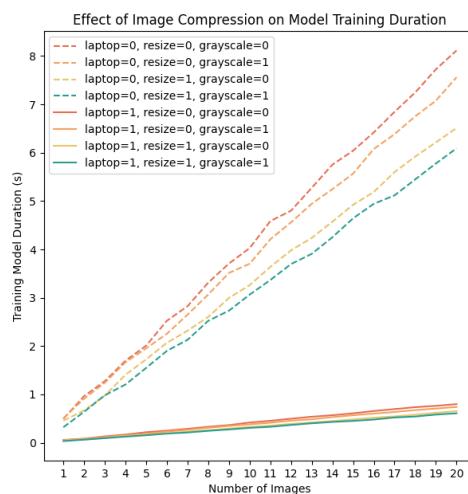
^۱ Principal Component Analysis (PCA)

۵ - ۲ - تاثیر پیش‌پردازش عکس‌ها بر زمان تمرین مدل شناسایی چهره

همان‌طور که در فصل سه و بخش ۱-۱-۴-۲-۳ گفته شد، دو تغییر روی عکس‌های چهره‌هایی که در سامانه ذخیره می‌شوند اعمال می‌شود:

- تغییر ابعاد عکس‌ها به مربع‌های 64×64 پیکسلی
- حذف مولفه رنگ و ذخیره عکس‌ها به صورت سیاه و سفید

در نمودار شکل ۵-۴، تاثیر این کار در نمودار مدت زمان تمرین مدل شناسایی چهره بر حسب تعداد عکس‌های ثبت‌شده در سامانه هم در رایانه‌ی عادی و هم بورد Raspberry Pi قابل مشاهده است. همان‌طور که در راهنمای این نمودار گفته شده، منحنی‌های خط‌چین مربوط به بورد Raspberry Pi و خطوط صاف مربوط به لپ‌تاپ عادی هستند و می‌توان مشاهده کرد که سرعت تمرین مدل شناسایی چهره روی بورد Raspberry Pi کمتر است. همچنین در این نمودار می‌توان مشاهده کرد که اگر قرار باشد فقط یکی از این دو تغییر را روی عکس‌ها اعمال شود، تغییر اندازه عکس‌ها تاثیر بیش‌تری نسبت به سیاه و سفید کردن آن‌ها در کاهش سرعت تمرین مدل دارد.



شکل ۵-۲ - تاثیر پیش‌پردازش عکس‌ها بر روی زمان تمرین مدل شناسایی چهره

۵ - ۳ مقایسه الگوریتم‌های تشخیص چهره

برای تشخیص چهره دقیق‌تر، می‌توان از شبکه عصبی پیچشی استفاده کرد. این مدل در برابر تغییرات در چرخش صورت و زاویه دید بسیار دقیق‌تر و قوی‌تر از الگوریتم هیستوگرام شبیه‌های جهت دار عمل می‌کند ولی از آن جا که به محاسبات بسیار بیشتری نیاز دارد، بسیار کندر عمل می‌کند و در تصویر نمایش داده شده روی نمایش‌گر، پرش مشاهده می‌شود؛ مخصوصاً در صورتی که این برنامه روی بورد Raspberry Pi که منابع ضعیف‌تری دارد اجرا شود.

برای پردازنده‌های ضعیف‌تر پیشنهاد می‌شود که از الگوریتم هیستوگرام شبیه‌های جهت‌دار استفاده شود، زیرا الگوریتمی کاراتر محسوب می‌شود و دقیق‌تر است که در تشخیص چهره دارد قابل قبول است. با توجه به این که در این پروژه قرار است افراد روبروی دوربین بایستند و صورت خود را به سمت دوربین قرار دهند، نیازی به استفاده از الگوریتم شبکه عصبی پیچشی برای تشخیص چهره‌های چرخیده وجود ندارد. برای استفاده از الگوریتم شبکه عصبی پیچشی، می‌توان از واحد پردازش گرافیک^۱ کمک گرفت که در کنار دقت بالای این الگوریتم، بتوان پرش کمتری در تصویر دوربین مشاهده کرد.

۵ - ۴ بررسی دقت شناسایی چهره‌ها

برای سنجیدن این که مدل شناسایی چهره تا چه حد می‌تواند خوب عمل کند، آن روی مجموعه داده‌ی "چهره‌های برچسب‌دار در دنیای واقعی"^۲ آزمایش می‌کنیم. این مجموعه داده که توسط آزمایشگاه بینایی دانشگاه ماساچوست^۳ ارائه شده، شامل ۱۳۲۳۳ عکس از ۵۷۴۹ شخص منحصر به فرد است که ۱۶۸۰ نفر از این اشخاص، دارای ۲ عکس یا بیش‌تر می‌باشند که ۸۲۵۷ عکس را به خود اختصاص می‌دهند [۲۳]. نمونه‌ای از تصاویر موجود از یک شخص در این مجموعه داده در شکل ۵-۳ قابل مشاهده است.

¹ GPU

² Labeled Faces in the Wild

³ Massachusetts



شکل ۵- ۳ تعدادی از چهره‌های یک شخص در مجموعه داده "چهره‌های برق‌ساز در دنیا واقعی"

برای ارزیابی مأذول شناسایی چهره، مجموعه تمرین^۱ خود را به این صورت تعریف می‌شود که شامل یکی از تصویرهای آن ۱۶۸۰ نفری باید که بیش از یک تصویر از آن‌ها وجود دارد، و سایر تصاویر را در مجموعه تست^۲ قرار داده می‌شود؛ بدین ترتیب، مجموعه تمرین شامل ۱۶۸۰ تصویر و مجموعه تست شامل ۶۵۷۷ تصویر می‌باشد. ابتدا مدل با مجموعه تمرین آموزش داده می‌شود و سپس به ترتیب، تمام داده‌های مجموعه تست توسط مدل دریافت می‌شود؛ سپس نام برگردانده شده توسط مدل با نام واقعی تصاویر ورودی مقایسه می‌شود. ماتریس سردرگمی^۳ این آزمایش در جدول فلان نمایش داده شده است. به کمک جدول ۱-۵ می‌توان مقادیر صحت، دقت و امتیاز F1 را برای این آزمایش به ترتیب طبق روابط ۱-۵، ۲-۵ و ۳-۵ بدست آورد.

جدول ۱-۵ ماتریس سردرگمی

		خروجی مدل شناسایی چهره	
		6577	$P'(6346)$
نحوه نمونه‌برداری	مثبت	$P(6566)$	$TP(6341)$
	منفی	$N(11)$	$FP(5)$
			$TN(6)$

¹ Training set

² Testing set

³ Confusion matrix

$$Accuracy = \frac{TP + TN}{P + N} = \frac{6341 + 6}{6577} = \frac{6347}{6577} = 96.502\% \quad (1-5)$$

$$precision = \frac{TP}{TP + FP} = \frac{6341}{6341 + 5} = \frac{6341}{6346} = 99.921\% \quad (2-5)$$

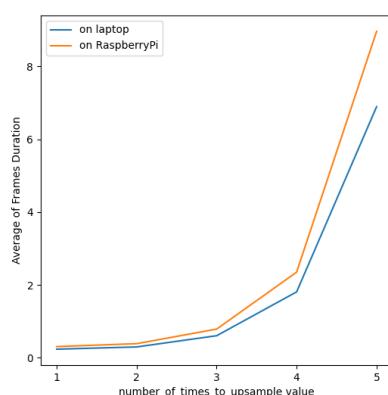
$$F1 Score = \frac{2TP}{2TP + FP + FN} = \frac{2 \times 6341}{2 \times 6341 + 5 + 225} = \frac{12682}{12312} = 98.218\% \quad (3-5)$$

در نتیجه طی این آزمایش، برای صحت، مقدار ۹۶.۵۰۲ درصد، برای دقت مقدار ۹۹.۹۲۱ درصد و برای امتیاز F_1 مقدار ۹۸.۲۱۸ محاسبه شد که مقدار قابل قبولی است.

۵ - ۵ تاثیر مقدار γ مدت زمان هر قاب

برای بررسی تاثیر مقدار γ مدت زمان هر قاب، میانگین زمان ۲۰ قاب متوالی را برای γ با مقادیر ۱ تا ۵ محاسبه کرده و آن را در نمودار موجود در شکل ۴-۵ نشان می‌دهیم. همان‌طور که مشاهده می‌شود، با افزایش مقدار γ جهت تشخیص بهتر چهره‌های کوچکتر، مدت زمان هر قاب به صورت نمایی افزایش پیدا می‌کند و باعث مشاهده پرش در تصویر نمایش داده شده روی نمایش‌گر می‌شود. در نتیجه پیشنهاد می‌شود در تنظیمات سامانه که از طریق فایل پیکربندی قابل کنترل است، مقدار γ را بیشتر از عدد "۲" قرار داده نشود.

Effect of number_of_times_to_upsample on Frame Duration

شکل ۴-۵ میانگین زمان ۲۰ قاب متوالی را برای γ با مقادیر ۱ تا ۵

٦ - جمع‌بندی، نتیجه‌گیری و پیشنهادها

۶ - ۱ جمع‌بندی و نتیجه‌گیری

شناسایی چهره، یکی از چالش‌برانگیزترین مسائل دنیای امروز محسوب می‌شود چرا که معمولاً از آن برای احراز هویت افراد استفاده می‌شود و احراز هویت غلط ممکن است در برخی موارد، موجب به وجود آمدن خسارت شود. اولین گام فرایند شناسایی چهره، تشخیص چهره می‌باشد. دو الگوریتم تشخیص چهره در این پژوهه بیان شده است: الگوریتم هیستوگرام شبکه‌های جهت‌دار و الگوریتم شبکه عصبی پیچشی.

طبق بررسی‌های انجام شده از عملکرد سامانه، نتایج زیر حاصل شد:

- الگوریتم هیستوگرام شبکه‌های جهت‌دار نسبت به الگوریتم شبکه عصبی پیچشی در تشخیص چهره‌های با زاویه نسبت به دوربین عملکرد خوبی ندارد ولی سربار کمتری دارد و برای اجرا روی بورد Raspberry Pi که منابع محدودی دارد مناسب‌تر است.
- بر اساس ارزیابی‌های انجام شده روی "چهره‌های برچسبدار در دنیای واقعی"، صحت ۹۶.۵۰۲ درصد، دققت ۹۸.۲۱۸ درصد و امتیاز F1 برابر با ۹۹.۹۲۱ درصد محاسبه شدند که در صدهای قابل قبولی می‌باشند.
- زیاد کردن پارامترهایی که به دققت سامانه کمک می‌کنند، باعث کند شدن تصویر سامانه می‌شوند که نامطلوب است، در نتیجه باید از زیاد کردن بی‌رویه آن جلوگیری نمود.

۶ - ۲ پیشنهادها

• یکی از بزرگترین ابراهای فرایند احراز هویت به کمک چهره، تشخیص مثبت نادرست^۱ است. به عنوان مثال، ممکن است تصویر یک شخص که عضو سامانه است در تلفن همراه یک شخص ناشناس وجود داشته باشد و چهره کاربر واقعی سامانه از این طریق مقابله دوربین سامانه قرار بگیرد. در این صورت، سامانه متوجه این موضوع نخواهد شد و هویت او را تایید می‌کند. یک پیشنهاد، افزودن حسگری برای سنجیدن فاصله چهره تا دوربین است. اگر از کاربران خواسته شود که در فاصله‌ای معین از دوربین سامانه قرار گیرند، می‌توان گفت اندازه چهره‌های تشخیص داده شده در محدوده معینی قرار خواهد گرفت. اگر

^۱ False Positive (FP)

- احراز هویت در صورتی انجام شود که فاصله چهره از سامانه و همچنین اندازه چهره تشخیص داده شده معین باشد، در این صورت می‌توان از برخی موارد مثبت نادرست جلوگیری کرد.
- پیشنهاد می‌شود از مدل‌های قوی‌تر Raspberry Pi برای این سامانه استفاده شود. در این حالت، میزان حافظه برابر یک گیگابایت می‌باشد و برنامه ممکن است در مواردی که تعداد چهره‌های سامانه زیاد شود خیلی کند عمل کند.
 - در این پروژه پس از افزودن یا حذف کاربران، مجدداً تمام عکس‌های سامانه بررسی می‌شوند. اگر این اطمینان وجود داشته باشد که فرایند افزودن و حذف کاربران تنها از طریق سامانه انجام می‌شود، می‌توان از بررسی عکس‌های اضافی صرف‌نظر کرد.
 - در پیاده‌سازی این پروژه فرض بر این بوده که تنظیمات پیکربندی سامانه با نرخ کمی تغییر می‌کنند. اگر در عمل، نرخ این تغییرات افزایش پیدا کرد، خوب است امکان ویرایش پیکربندی را از طریق تابلوی کنترلی مدیر فراهم نمود.

٧ - مراجع

- [1] B. T. V. Bhandiwad, "Face recognition and detection using neural networks," *International Conference on Trends in Electronics and Informatics (ICEI)*, p. 879–882, 2017.
- [2] M. J. S. L.-J. L. S. S. Farfade, "Multi-view Face Detection Using Deep Convolutional Neural Networks," Shanghai China, 2015, p. 643–650.
- [3] H. A. B. M. H. Ahmad Fadzil, "Human face recognition using neural networks," *1st International Conference on Image Processing*, vol. 3, p. 936–939, 1994.
- [4] D.-S. H.-Y. S. Z.-Q. Zhao, "Human face recognition based on multi-features using neural networks committee," *Pattern Recognition Letters*, vol. 25, p. 1351–1358, 2004.
- [5] T. H. Le, "Applying Artificial Neural Networks for Face Recognition," *Advances in Artificial Neural Systems*, 2011.
- [6] P. Latha, "Face Recognition using Neural Network," *Signal Processing*, p. 8.
- [7] T. i, "Face Detection vs. Facial Recognition – What's the Difference?," Two i, 2022. [Online]. Available: <https://www.two-i.com/blog/face-detection-vs-facial-recognition-whats-the-difference>. [Accessed 2023].
- [8] NEC, "Face Detection vs Facial Recognition – what's the difference?," NEC, 1 June 2022. [Online]. Available: <https://www.nec.co.nz/market-leadership/publications-media/face-detection-vs-facial-recognition-whats-the-difference/>. [Accessed 2023].
- [9] IBM, "Introduction to Neural Networks," IBM, 2021. [Online]. Available: <https://www.ibm.com/topics/neural-networks>. [Accessed 2023].
- [10] A. Rosebrock, "Face detection with dlib," pyimagesearch, 2021 April 2021. [Online]. Available: <https://pyimagesearch.com/2021/04/19/face-detection-with-dlib-hog-and-cnn/>. [Accessed 2023].

-
- [11] R. Pi, "Raspberry Pi Documentation," Raspberry Pi, 2022. [Online]. Available: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>. [Accessed 2023].
 - [12] R. Pi, "About the Camera Modules," Raspberry Pi, 2022. [Online]. Available: <https://www.raspberrypi.com/documentation/accessories/camera.html>. [Accessed 2023].
 - [13] A. B. Downey, Think Python: How to Think Like a Computer Scientist, O'Reilly, 2020.
 - [14] OpenCV, "OpenCV website," OpenCV, [Online]. Available: <https://opencv.org/>. [Accessed 2023].
 - [15] Dlib, "Dlib Web Page," Dlib, [Online]. Available: <http://dlib.net>. [Accessed 2023].
 - [16] NumPy, "NumPy Webpage," NumPy, [Online]. Available: <https://numpy.org>. [Accessed 2023].
 - [17] Matplotlib, "Matplotlib Web Page," Matplotlib, [Online]. Available: <https://matplotlib.org>. [Accessed 2023].
 - [18] Flask, "Flask Web Framework," Flask, [Online]. Available: <https://Flask.palletsprojects.com/en/2.2.x>. [Accessed 2023].
 - [19] AWS, "What is an IDE?," Amazon, [Online]. Available: <https://aws.amazon.com/what-is/ide/>. [Accessed 2023].
 - [20] "PyCharm Guide," RealPython, [Online]. Available: <https://realpython.com/pycharm-guide/>. [Accessed 2023].
 - [21] OpenCV, "HOG pre-trained model," OpenCV, [Online]. Available: <https://github.com/opencv/opencv/tree/master/data/haarcascades>. [Accessed 2023].
 - [22] Z. Z. Z. L. Y. Q. K Zhang, "Face Detection and alignment using multitask cascaded convolutional networks," *IEEE*, 2016.
 - [23] U. o. Massachusetts, "Labeled Faces in the Wild," University of Massachusetts, 2018. [Online]. Available: <http://vis-www.cs.umass.edu/lfw/>. [Accessed 2023].