

دانشگاه صنعتی شاهرود

دانشکده مهندسی برق

پایان نامه کارشناسی
مهندسی برق-الکترونیک

عنوان:

طراحی و پیاده سازی سیستم بلادرنگ شناسایی چهره در
فریم های ویدئویی مبتنی بر ویژگی های LBP بر روی ُبرد
Odroid XU4

نگارش

رضا آدینه پور

استاد راهنما

جناب آقای دکتر علیرضا احمدی فرد

خرداد ۱۴۰۲

الله اعلم

تصویب‌نامه

به نام خدا

دانشگاه صنعتی شاهرود

دانشکده مهندسی برق

پایان‌نامه کارشناسی

این پایان‌نامه به عنوان تحقیق بخشی از شرایط دریافت درجه کارشناسی است.

عنوان: طراحی و پیاده‌سازی سیستم بلادرنگ شناسایی چهره در فریم‌های ویدئویی مبتنی بر
ویژگی‌های LBP بر روی ُبرد Odroid XU4

نگارش: رضا آدینه‌پور

کمیته ممتحنین

استاد راهنما: جناب آقای دکتر علیرضا احمدی فرد امضاء:

تاریخ:

اطهارنامه

(اصالت متن و محتوای پایان نامه کارشناسی)

عنوان پایان نامه: طراحی و پیاده سازی سیستم بلادرنگ شناسایی چهره در فریم های ویدئویی مبتنی بر ویژگی های Odroid XU4 LBP بر روی بُرد

استاد راهنما: جناب آقای دکتر علیرضا احمدی فرد

این جانب رضا آدینه‌پور اظهار می‌دارم:

۱. متن و نتایج علمی ارائه شده در این پایان نامه اصیل بوده و زیرنظر استادان نامبرده شده در بالا تهیه شده است.
۲. متن پایان نامه به این صورت در هیچ جای دیگری منتشر نشده است.
۳. متن و نتایج مندرج در این پایان نامه، حاصل تحقیقات این جانب به عنوان دانشجوی کارشناسی دانشگاه صنعتی شاهروود است.
۴. کلیه مطالبی که از منابع دیگر در این پایان نامه مورد استفاده قرار گرفته، با ذکر مرجع مشخص شده است.

نگارنده: رضا آدینه‌پور

تاریخ:

امضاء:

نتایج تحقیقات مندرج در این پایان نامه و دستاوردهای مادی و معنوی ناشی از آن (شامل فرمول‌ها، توابع کتابخانه‌ای، نرم‌افزارها، سخت‌افزارها و مواردی که قابلیت ثبت اختراع دارد) متعلق به دانشگاه صنعتی شاهروود است. هیچ شخصیت حقیقی یا حقوقی بدون کسب اجازه از دانشگاه صنعتی شاهروود حق فروش و ادعای مالکیت مادی یا معنوی بر آن یا ثبت اختراع از آن را ندارد. همچنین، کلیه حقوق مربوط به چاپ، تکثیر، نسخه‌برداری، ترجمه، اقتباس و نظائر آن در محیط‌های مختلف اعم از الکترونیکی، مجازی یا فیزیکی برای دانشگاه صنعتی شاهروود محفوظ است. نقل مطلب با ذکر مأخذ بلامانع است.

نگارنده: رضا آدینه‌پور

استاد راهنما: جناب آقای دکتر علیرضا احمدی فرد

تاریخ:

تاریخ:

امضاء:

امضاء:

سپاس

از استاد بزرگوارم **جناب آقای دکتر علیرضا احمدی فرد** که به بنده اعتماد کردن و با کمک‌ها و راهنمایی‌های بی‌دریغشان، مرا در به سرانجام رساندن این پروژه یاری داده‌اند، تشکر و قدردانی می‌کنم.

عرض تشکر و قدردانی دارم از استاد گرامی **جناب آقای دکتر حسین خسروی** جهت در اختیار گذاشتن سخت افزار Odroid و همچنین داوری پایان‌نامه.

همچنین از دوستان عزیزم آقایان بیاتی، توکلی، عباسی، علوی، موسوی، مرادی و نوکی و همه دوستانی که در تهیه دیتاست به بنده لطف داشتند و همکاری کردن‌کمال تشکر را دارم.

و در آخر، صمیمانه از خانواده عزیزم که هر چه دارم از آنهاست تشکر ویژه می‌کنم و این پایان‌نامه را در روز تولد پدرم «امروز، ۲۶ ام خرداد ۱۴۰۲» به ایشان تقدیم می‌کنم.

چنان مجنون شوم گویی که شهرآشوب دورانم
نه پندی و نه اندرزی به گوشم پنجه چسباندم
بماند یادگار این شعر
که من از دار این دنیا فقط این شعر می‌دانم.

چکیده

هدف از انجام این پایان نامه، طراحی و پیاده سازی سیستمی بلادرنگ^۱ برای تشخیص و شناسایی چهره افراد مبتنی بر الگوهای باینری محلی^۲ بوده است. این سیستم بر روی کامپیوتر نهفته Odroid-Xu^۳ پیاده سازی شده است. در این طراحی از کتابخانه متن باز^۴ OpenCV^۵ و به طور خاص از طبقه بند آبشری^۶ و K-NN^۷ استفاده شده است. یکی از مزیت های این سیستم قابل حمل^۸ بودن آن است که می توان در محیط های مختلفی که نیازمند این کاربرد هستند، مانند ادارات جهت حضور و غیاب و ثبت ساعات ورود و خروج افراد استفاده نمود.

کلیدواژه ها: طبقه بند، الگوهای باینری محلی، کامپیوتر نهفته

Real-Time^۱
Local Binary Pattern^۲
Embedded Computer^۳
Open Source^۴
Open Source Computer Vision^۵
Cascade Classifier^۶
K-Nearest Neighbor^۷
Portable^۸

فهرست مطالب

۱	۱	مقدمه
۱	۱-۱	تعريف مسئله
۲	۲-۱	اهمیت موضوع
۲	۳-۱	اهداف پروژه
۲	۴-۱	ساختار پایان نامه
۳	۲	طبقه‌بند‌ها
۳	۱-۲	K-NN طبقه‌بند
۴	۱-۱-۲	الگوریتم
۴	۲-۱-۲	انتخاب پارامتر
۵	۳-۱-۲	مزایا و معایب
۶	۲-۲	طبقه‌بند آبشاری و ویژگی‌های LBP
۶	۱-۲-۲	طبقه‌بند آبشاری
۸	۲-۲-۲	الگوهای باینری محلی
۹	۳-۲-۲	الگوهای باینری محلی برای الگوریتم شناسایی چهره
۱۲	۳	راهاندازی سخت‌افزار
۱۲	۱-۳	معرفی بورد Odroid-XU4
۱۳	۲-۳	نصب سیستم عامل

۱۵	روشن کردن بورد	۳-۳
۱۶	نصب پکیج ها و کتابخانه های ضروری	۴-۳
۱۷	آپدیت سیستم	۱-۴-۳
۱۷	نصب git	۲-۴-۳
۱۷	نصب oh-my-zsh	۳-۴-۳
۱۸	نصب پایتون	۴-۴-۳
۱۸	تغییر ورژن پایتون	۵-۴-۳
۱۸	نصب pip	۶-۴-۳
۱۹	نصب Idle IDE	۷-۴-۳
۲۰	نصب کتابخانه Numpy	۸-۴-۳
۲۰	نصب کتابخانه Pandas	۹-۴-۳
۲۰	نصب کتابخانه Pillow	۱۰-۴-۳
۲۱	نصب کتابخانه Tkinter	۱۱-۴-۳
۲۱	نصب کتابخانه OpenCV	۱۲-۴-۳
۲۶	اجرای الگوریتم و نتایج	۴
۲۶	رابط کاربری برنامه	۱-۴
۲۷	فاز آموزش شبکه	۲-۴
۲۸	فاز تست و شناسایی چهرهها	۳-۴
۳۰	نتیجه گیری	۵
۳۱	مراجع	
۳۲	واژه نامه	
۳۴	آ مطالب تکمیلی	

۳۴	۱- آستخراج بردار ویژگی های LBP
۳۵	۲- تست دوربین
۳۶	۳- فاز آموزش
۳۶	۴- فاز تست و شناسایی چهره

فهرست جداول

فهرست تصاویر

۱-۲	استفاده از قضیه فیثاغورس برای محاسبه فاصله اقلیدسی دو بعدی	۴
۲-۲	فیلترهای مورد استفاده برای استخراج ویژگی	۷
۳-۲	استخراج ویژگی های خوب	۷
۴-۲	همسايگی در LBP	۹
۵-۲	عملکرد LBP	۹
۶-۲	پیاده سازی عملی LBP	۹
۷-۲	بردار ویژگی های LBP	۱۰
۸-۲	تقسیم چهره به سلول های 7×7	۱۱
۹-۲	تصویر اصلی چهره «سمت چپ» و وزن های استخراج شده «سمت راست»	۱۱
۱-۲	بلوک دیاگرام Odroid-XU4	۱۳
۲-۲	بورد Odroid-XU4	۱۳
۳-۲	صفحه دانلود سیستم عامل	۱۴
۴-۲	صفحه دانلود WinRAR	۱۴
۵-۲	صفحه دانلود 7-Zip	۱۵
۶-۲	نرم افزار Win32DiskImager	۱۶
۷-۲	صفحه دسکتاپ اوبونتو	۱۶
۸-۲	تغییر وزن پیش فرض پایتون	۱۹
۹-۲	محیط Idle	۲۰

۲۶	۱-۴ رابط کاربری برنامه
۲۷	۲-۴ تهیه دیتاست
۲۸	۳-۴ اطلاعات ذخیره شده کاربران
۲۹	۴-۴ تست سخت افزار و الگوریتم

فصل ۱

مقدمه

سیستم تشخیص و شناسایی چهره^۱ یک تکنولوژی توانا در شناسایی و تأیید هویت یک فرد از یک عکس دیجیتالی یا ویدئو می‌باشد. سیستم تشخیص چهره، سیستمی است که بر اساس تکنولوژی هوش مصنوعی^۲ و قادر به شناسایی چهره افراد با دقت بالا می‌باشد. در بازشناخت تصویر یک چهره، تصویر ورودی با توجه به اطلاعات موجود در بانک اطلاعات، مورد شناسایی قرار می‌گیرد. این بانک شامل مشخصاتی از تصویر چهره افراد شناسایی شده است.

۱-۱ تعریف مسئله

در بسیاری از کاربردها نیازمند شناسایی چهره افراد هستیم. در مقیاس های کوچک شاید بتوان از ناظر انسانی جهت انجام این کار استفاده کرد. اما با گسترش مقیاس مسئله، انجام این کار توسط انسان قدری دشوار خواهد بود. در این نقطه کامپیوتر به کمک انسان می‌آید و انجام کار را آسان می‌کند. همکاری ای که بینایی ماشین^۳ نامیده می‌شود.

۱ Face Detect and Recognition System

۲ Artifitial Intelligence

۳ Computer Vision

۲-۱ اهمیت موضوع

از گذشته تا به امروز، مسئله شناسایی چهره از اهمیت زیادی برخوردار بوده است. چرا که با شناسایی افراد، هویت آنها فاش نمیشود. محل های زیادی وجود دارد که شناسایی چهره در آنجا اهمیت ویژه ای دارد مانند:

- فرودگاهها

- اداره‌ها

- دانشگاهها

- مکان‌های عمومی

۳-۱ اهداف پروژه

هدف از انجام این پایان‌نامه، طراحی و ساخت سیستمی قابل حمل^۴، بلادرنگ^۵ با قابلیت اطمینان بالا به کمک هوش مصنوعی و طبقه‌بندی‌های مختلف بوده است که تا حد زیادی انتظارات ما در انجام این پروژه برآورده شده است.

۴-۱ ساختار پایان‌نامه

این پایان‌نامه در ۵ فصل و ۱ پیوست به شرح زیر ارائه می‌شود:

فصل دوم به معرفی ساختار طبقه‌بندی‌های مهم همچون K-NN، آبشاری، کاربرد‌ها و ویژگی‌های آنها می‌پردازد. فصل سوم به معرفی بُرد Odroid، نصب سیستم عامل و پکیج‌های مورد نیاز برای انجام پروژه اختصاص دارد. در فصل چهارم، پیاده‌سازی سیستم و نتایج بدست آمده از انجام این پروژه، ارائه می‌شود. فصل پنجم به جمع‌بندی کارهای انجام شده در این پروژه و ارائه‌ی پیشنهادهایی برای انجام کارهای آتی خواهد پرداخت. در پیوست شماره ۱، برخی از کدهای مهم استفاده شده در پروژه آورده شده است^۶.

Portable^۴
Real Time^۵

github.com/rezaAdinepour/Bachelors-Project: تمامی کدها از این لینک قابل دریافت است.

فصل ۲

طبقه‌بند ها

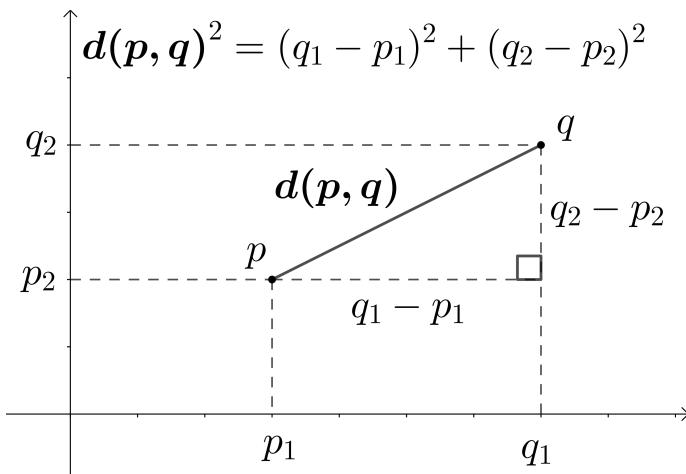
در این فصل ابتدا به معرفی چند طبقه‌بند مهم می‌پردازیم و ویژگی‌ها، کاربردها و نحوه عملکرد و پیاده‌سازی آنها را بررسی می‌کنیم.

۱-۲ طبقه‌بند K-NN

در شناسایی الگو^۱ K -نزدیکترین همسایه^۲ یک متادامار ناپارامتری است که برای طبقه‌بندی آماری^۳ و رگرسیون^۴ استفاده می‌شود. در هر دو حالت K شامل نزدیکترین مثال آموزشی در فضای داده‌ای می‌باشد و خروجی آن بسته به نوع مورد استفاده در طبقه‌بندی و رگرسیون متغیر است. در حالت طبقه‌بندی با توجه به مقدار مشخص شده برای K ، به محاسبه فاصله نقطه‌ای که می‌خواهیم برچسب آن را مشخص کنیم با نزدیکترین نقاط می‌پردازد و با توجه به تعداد رای حداکثری این نقاط همسایه، در رابطه با برچسب نقطه مورد نظر تصمیم‌گیری می‌کنیم.

برای محاسبه این فاصله می‌توان از روش‌های مختلفی استفاده کرد که یکی از مطرح‌ترین این روش‌ها، فاصله اقلیدسی «شکل ۱-۲» است. در حالت رگرسیون نیز میانگین مقادیر به دست آمده از کی خروجی آن می‌باشد. از آنجا که محاسبات این الگوریتم بر اساس فاصله است نرمال‌سازی داده‌ها می‌تواند به بهبود عملکرد آن کمک کند. [۱] و [۲]

Pattern Recognition^۱
k-Nearest Neighbor^۲
Statistical classification^۳
Regression^۴



شکل ۱-۲: استفاده از قضیه فیثاغورس برای محاسبه فاصله اقلیدسی دو بعدی

۱-۱-۲ الگوریتم

داده‌های اولیه، بردارهایی در یک فضای چند بعدی هستند که هر کدام شامل برچسبی^۵ به نام دسته می‌باشد.

فاز یادگیری^۶ الگوریتم، شامل ذخیره‌سازی بردارهای ویژگی و برچسب دسته نمونه‌های اولیه است.

در فاز طبقه‌بندی، K یک ثابت توسط کاربر تعریف می‌شود و بردار بدون برچسب «نقطه تست» از دسته ای است که بیشترین تعداد را در K نزدیک‌ترین همسایه آن نقطه داشته باشد. به این ترتیب برچسب نقطه تست نیز مشخص می‌شود. معیار فاصله برای متغیرهای پیوسته معمولاً فاصله اقلیدسی است.

اگر $K\text{-NN}$ را با استفاده از الگوریتم‌های تخصصی مانند تجزیه و تحلیل اجزای همسایه^۷ یا حاشیه بزرگ نزدیک‌ترین همسایه^۸ پیاده‌سازی کرد، می‌توان دقت اندازه‌گیری را به شدت بهبود داد.

مراحل الگوریتم $K\text{-NN}$ شامل موارد زیر خواهد بود:

۲-۱-۲ انتخاب پارامتر

بهترین انتخاب K بستگی به داده‌ها دارد. به طور کلی، مقادیر بزرگ K باعث کاهش خطای طبقه‌بندی می‌شود، اما وضوح مرز بین کلاس‌ها را کمتر می‌کند. K مناسب را می‌توان با استفاده از تکنیک‌های مختلف انتخاب

Label ^۵	
Training Phase ^۶	
Neighbourhood Components Analysis ^۷	
Large Margin Nearest Neighbor ^۸	

الگوریتم ۱ K-NN

وروودی: ماتریس

خروجی: نزدیکترین همسایه

۱: داده‌های را بارگیری کنید

۲: K به عنوان تعداد نزدیکترین همسایگان انتخاب کنید.

۳: برای هر یک از داده‌های اولیه:

۴: فاصله بین داده مورد سؤال و هر یک داده‌ای اولیه را محاسبه کنید

۵: فاصله و انداز نمونه را به یک مجموعه اضافه کنید

۶: مجموعه را بر اساس فاصله از کوچک به بزرگ مرتب کنید

۷: نقاط K عضو اول مجموعه مرتب شده را انتخاب کنید

۸: بسته به حالت یا حالت طبقه‌بندی، خروجی را اعلام کنید

کرد. مورد خاص زمانی است که دسته پیش‌بینی شده برای عضو جدید، همان دسته نزدیکترین نمونه باشد. « به ازای $1 = K$ ». در این صورت، الگوریتم نزدیکترین همسایه نامیده می‌شود.

دقت الگوریتم K-NN می‌تواند در نتیجه وجود خطأ، یا ویژگی‌های غیر مرتبط یا اگر مقیاس داده‌ها با اهمیت آن‌ها تطابق نداشته باشد، به شدت تضعیف می‌شود. برای بهبود طبقه‌بندی، تلاش‌های زیادی در زمینه انتخاب یا مقیاس بندی داده‌ها شده است. یک رویکرد بسیار مشهور، استفاده از الگوریتم‌های تکاملی برای بهینه‌سازی مقیاس بندی داده‌ها است.

در دسته‌بندی‌های دو کلاس، بهتر است که K یک عدد فرد انتخاب شود، زیرا این امر از گره خوردن آرا جلوگیری می‌کند.

۳-۱-۲ مزایا و معایب

مزایا:

- هیچ پیش فرضی در مورد داده‌ها وجود ندارد - مثال برای داده‌های غیر خطی

- الگوریتم ساده

- دقیق نسبتاً بالا

• مقایسه مدل‌های یادگیری تحت نظارت بهتر

• چند منظوره - برای طبقه‌بندی و رگرسیون

مضرات:

• محاسبه گران

• نیاز به حافظه بالا - چرا که الگوریتم، تمام داده‌های قبلی را ذخیره می‌کند

• ذخیره تقریباً یا همه داده‌های اولیه

• مرحله پیش‌بینی ممکن است کند باشد «با N بزرگ»

• حساس به ویژگی‌های نامناسب و مقیاس داده

۲-۲ طبقه‌بند آبشاری و ویژگی‌های LBP

۱-۲-۲ طبقه‌بند آبشاری

برای اولین بار، ایده تشخیص اشیاء مبتنی بر طبقه‌بند آبشاری^۹ و با استفاده از ویژگی‌های هار^{۱۰} توسط پاول ویولا^{۱۱} و مایکل جونز^{۱۲} در مقاله‌ای تحت عنوان: Rapid Object Detection using a Boosted Cascade of Simple Features مطرح شد. رویکرد این مقاله بر اساس یادگیری ماشین^{۱۳} است که در آن یکتابع آبشاری با تعداد زیادی تصویر مثبت^{۱۵} و منفی^{۱۶} آموزش داده می‌شود.

[۳]

به دلیل آنکه مسئله ما تشخیص چهره است، در ابتدا برای آموزش شبکه به تعداد زیادی تصاویر مثبت و منفی نیاز داریم. در گام دوم باید از تصاویر ویژگی استخراج کنیم. برای استخراج ویژگی، از ویژگی‌های «هار» که در «شکل ۲-۲» نشان داده شده است استفاده می‌شود. این ویژگی‌ها دقیقاً مانند هسته‌های فیلترهای کانولوشنی هستند.

Cascade Classifier^۹
Haar^{۱۰}

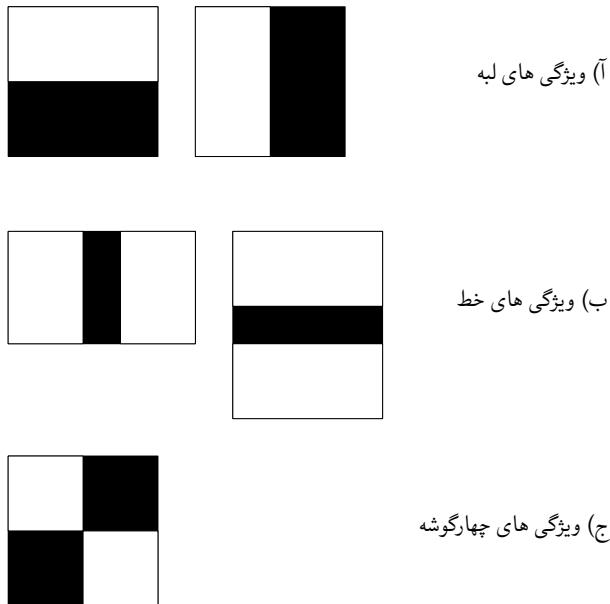
Paul Viola^{۱۱}
Michael Jones^{۱۲}

b2n.ir/reference_Haar نسخه اصلی مقاله از لینک زیر قابل دریافت است:

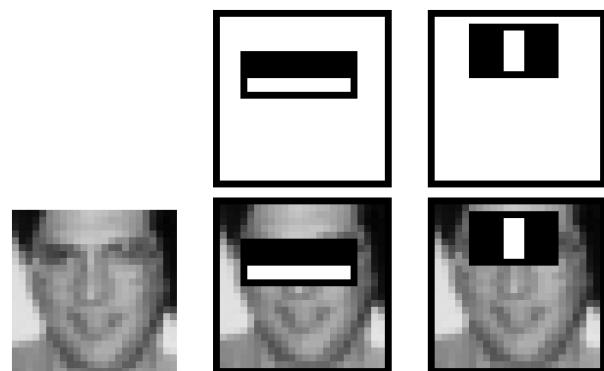
Machine Learning^{۱۴}

^{۱۵} تصویر چهره

^{۱۶} تصویر بدون چهره



شکل ۲-۲: فیلترهای مورد استفاده برای استخراج ویژگی



شکل ۳-۲: استخراج ویژگی های خوب

با استخراج ویژگی‌ها، تعداد زیادی از ویژگی‌ها مطلوب ما نیستند و اطلاعات مفیدی به ما نمی‌دهند. برای مثال، «شکل ۳-۲» دو نمونه ویژگی خوب را نشان می‌دهد. معمولاً ناحیه فورفتگی چشم‌ها تیره‌تر از ناحیه بالای بینی و گونه‌ها است. ویژگی اول به همین موضوع اشاره دارد. ویژگی دوم، بر این خاصیت متکی است که چشم‌ها تیره‌تر از ناحیه بینی است. اما اگر همین فیلترها در محل‌های دیگری قرار گیرند، ویژگی خوبی استخراج نمی‌کنند.

برای آموزش، تک تک ویژگی‌ها را روی تمام تصاویر آموزشی اعمال می‌کنیم. برای هر ویژگی، بهترین آستانه را پیدا می‌کنیم به طوری که چهره‌ها به دو دسته مثبت و مفی طبقه‌بندی کنند. بدیهی‌سیت که ممکن است طبقه‌بندی‌های اشتباهی صورت بگیرد که موجب ایجاد خطأ می‌شود. ما باید ویژگی‌هایی با کمترین میزان خطأ را انتخاب کنیم، یعنی ویژگی‌هایی را انتخاب کنیم که تصاویر چهره و غیر چهره را با دقت بیشتری طبقه‌بندی

کنند.

فاز اول فرایند آموزش بدنی صورت است که ابتدا به هر تصویر وزن‌های برابر داده می‌شود. پس از هر دوره آموزش وزن تصاویر طبقه‌بندی اشتباه افزایش می‌یابد. فرایند تا زمانی تکرار می‌شود که خطا یا دقت به مقدار مورد نظر ما برسد و بهترین ویژگی‌ها را پیدا کنیم.

طبقه‌بند نهایی، مجموع تعدادی طبقه بند ضعیف است که در مراحل قبل بدست آمده است. می‌گوییم ضعیف چون به تنها یک نمی‌توانند تصویر را طبقه‌بندی کنند اما اگر با سایر طبقه‌بندها ترکیب شوند، طبقه‌بندی قوی تشکیل می‌دهند به طوری که با ترکیب 200 ویژگی، دقت تشخیص به حدود 95% می‌رسد.

شرکت OpenCV بهترین ویژگی‌های بدست آمده^{۱۷} را برای تشخیص چهره، چشم، بدن و ... در قالب فایل‌هایی با پسوند `.xml`. ارائه کرده است.

در این پروژه ما از فایل `haarcascade_frontalcatface.xml` استفاده کردیم. در فصل بعد به بررسی مقدمات و الزامات انجام پروژه مانند معرفی سخت‌افزار، نصب سیستم‌عامل و ... می‌پردازیم.

۲-۲-۲ الگوهای باینری محلی

الگوهای باینری محلی^{۱۸} یا به طور اختصار LBP اولین بار در سال 2004 توسط آهنن^{۱۹} در مقاله‌ای تحت عنوان Face Recognition with Local Binary Patterns^{۲۰} ارائه شد.

نحوه عملکرد LBP بدنی صورت است که یک پیکسل از تصویر طبق یک آستانه^{۲۱} با پیکسل‌های همسایه خود مقایسه می‌شود و مقدار جدید پیکسل‌ها ساخته می‌شود.

اولین مرحله در LBP تبدیل تصویر RGB به Grayscale است. برای تمامی پیکسل‌های تصویر خاکستری، یک همسایگی به اندازه شعاع r از پیکسل مرکزی انتخاب می‌کنیم. انتخاب می‌کنیم. «شکل ۴-۲» سپس یک مقدار جدید LBP برای پیکسل مرکزی محاسبه می‌شود و در آرایه 2 بعدی خروجی با همان عرض و ارتفاع تصویر ورودی ذخیره می‌شود. «شکل ۵-۲» [۲۲]

مقدار جدید بدست آمده برای پیکسل وسط جایگزین مقدار قبلی آن می‌شود. این عمل برای تمامی پیکسل‌های تصویر تکرار شده و مقادیر پیکسل‌ها آپدیت می‌شود. در نهایت تمامی ویژگی‌های بدست آمده در یک بردار^{۲۳} کنار هم چیده می‌شود و بردار ویژگی‌های LBP را می‌سازند. [۵]

^{۱۷} فایل‌های نام بردۀ از این لینک قابل دریافت است: github.com/opencv/opencv/tree/master/data/haarcascades

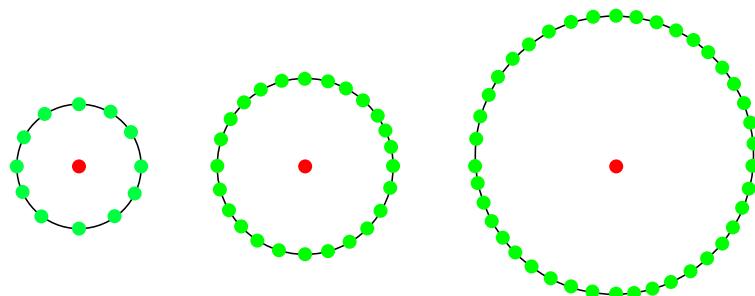
Local Binary Pattern^{۱۸}

Ahonen^{۱۹}

^{۲۰} فایل مقاله از این لینک قابل دریافت است: b2n.ir/reference_Local

Threshold^{۲۱}

Vector^{۲۲}



شکل ۲-۴: همسایگی در LBP

28	92	87
49	60	75
32	93	81

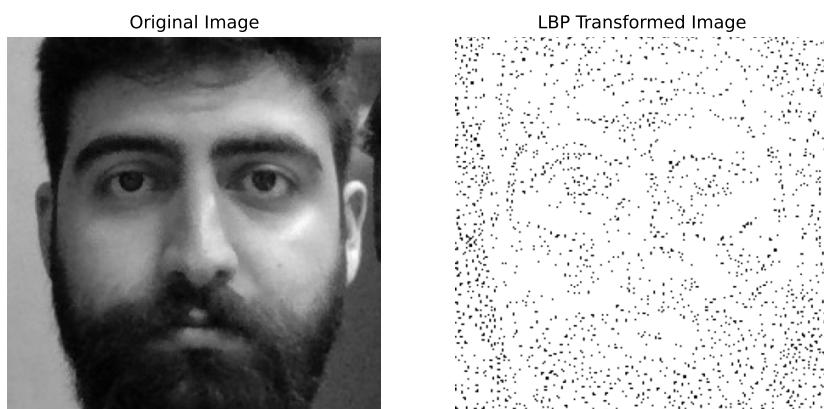
Thresholding →

0	1	1
0		1
0	1	1

Coding → (11111000) = 242

شکل ۲-۵: عملکرد LBP

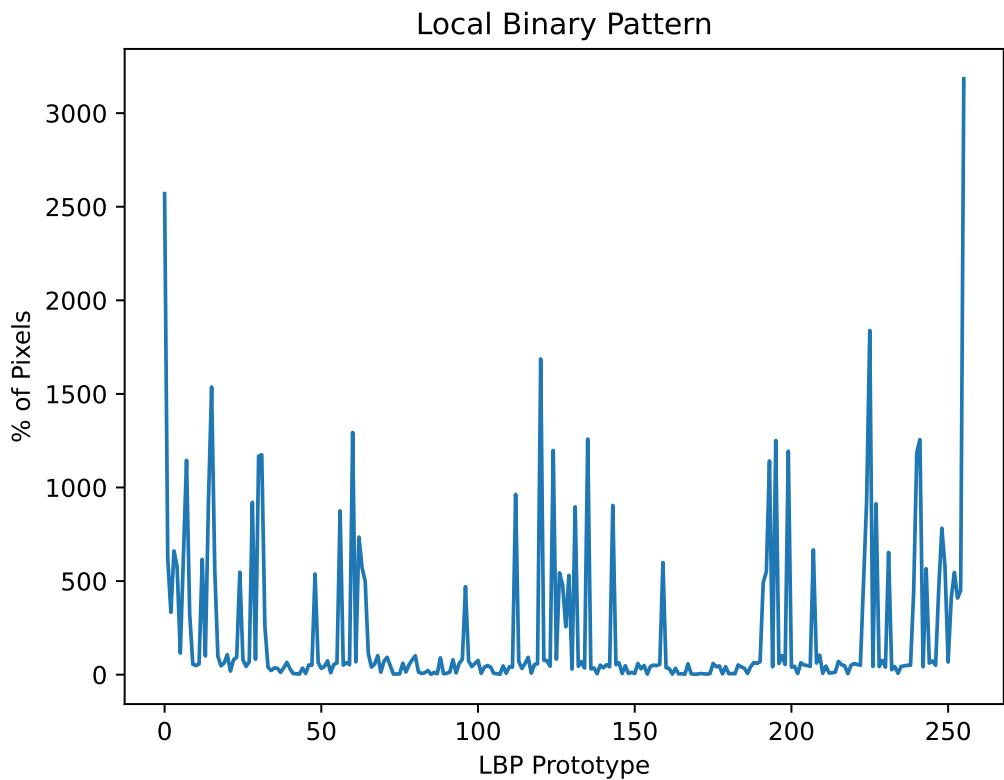
در شکل‌های «۶-۲ و ۷-۲» نمونه‌ای عملی از استخراج ویژگی‌های LBP با همسایگی ۸ پیکسل و شعاع همسایگی ۱ آورده شده است: «جهت بررسی کد این بخش، به پیوست آ مراجعه کنید»



شکل ۲-۶: پیاده‌سازی عملی LBP

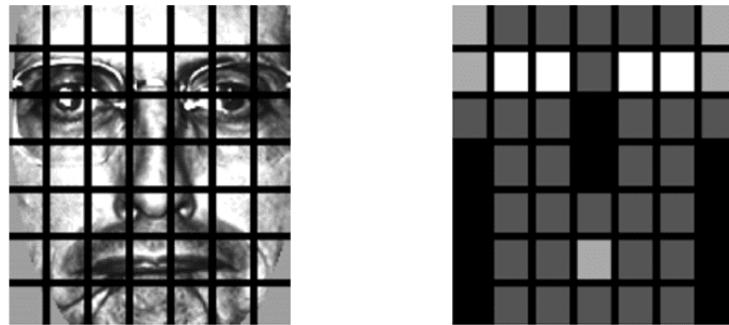
۳-۲-۲ الگوهای باینری محلی برای الگوریتم شناسایی چهره

با توجه به دیتاست تصاویر موجود، در اولین مرحله الگوریتم، چهره را به سلول‌های 7×7 تقسیم می‌کنیم. «شکل ۲-۸» سپس برای هرکدام از این سلول‌ها هیستوگرام LBP را محاسبه می‌کنیم. با محاسبه هیستوگرام





شکل ۲-۸: تقسیم چهره به سلول های 7×7



شکل ۲-۹: تصویر اصلی چهره «سمت چپ» و وزن های استخراج شده «سمت راست»

عمل تشخیص چهره با استفاده از فاصله χ^2 و طبقه بند نزدیک ترین همسایه^{۲۳} به صورت زیر انجام می شود:

- چهره به سیستم داده می شود
- LBP ها به روی که توضیح داده شد استخراج شده و به وزن تبدیل می شوند و به هم متصل شده تا بردار ویژگی ها را بسازند.
- KNN با $K = 1$ و با فاصله χ^2 برای یافتن نزدیک ترین چهره در دیتا ست انجام می شود.
- نام شخص مرتبط با چهره با کمترین فاصله χ^2 به عنوان طبقه بندی نهایی انتخاب می شود.

Nearest Neighbor^{۲۴}

فصل ۳

راهاندازی سخت افزار

در این فصل به بررسی مشخصات سخت افزار مورد استفاده، نصب سیستم عامل می پردازیم.

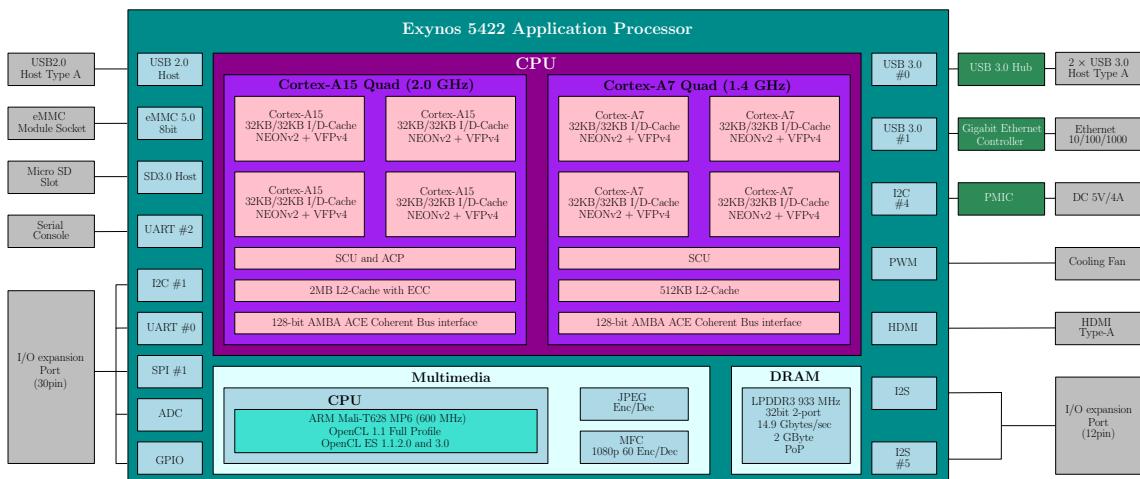
۱-۳ معرفی بورد Odroid-XU4

بُرد او دروید مینی کامپیوتری است که می توان بر روی آن سیستم عامل هایی مثل لینوکس «او بونتو^۱، فدورا^۲، آرج^۳ و دبین^۴» اندروید^۵ را نصب کرد. در این پروژه از سیستم عامل لینوکس و توزیع Ubuntu استفاده شده است.

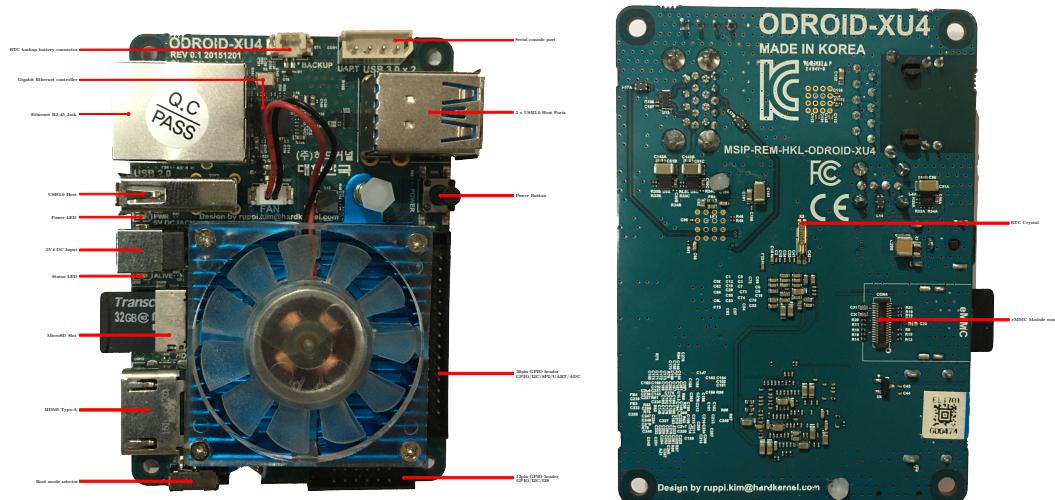
بُرد او دروید مشخصاتی همانند سایر کامپیوترهای معمولی را دارد. RAM CPU، پورت های استاندارد ورودی و خروجی و... که بلوک دیاگرام آن در «شکل ۱-۳» آورده شده است.

تغذیه بُرد، ۵ ولت ۴ آمپر پیشنهاد شده است که ما از یک آدپتور DC، ۵ ولت ۶ آمپر استفاده کردیم. برای نمایش تصویر هم از یک مانیتور معمولی استفاده شده. از یک عدد موس و کیبورد معمولی هم برای دسترسی کاربر استفاده شده است. ضمن اینکه از یک عدد MicroSD کارت ۳۲ گیگابایتی برای نصب سیستم عامل بر روی آن استفاده شده است «می توان از حافظه ای با حجم کمتر هم استفاده نمود»

Ubuntu^۱
Fedora^۲
ArchLinux^۳
Debian^۴
Android^۵



شکل ۱-۳: بلوک دیاگرام Odroid-XU4



شکل ۲-۳: بورد Odroid-XU4

۲-۳ نصب سیستم عامل

ابتدا از این لینک^۶ آخرین ورژن اوبونتو را دانلود می‌کنیم. «در این پروژه از ۱۶.۰۴ Ubuntu استفاده شده است.

فایل دانلود شده با پسوند img. را باید با یکی از نرم افزار های فشرده ساز مانند WinRAR (شکل ۴-۳)^۷ یا 7-Zip (شکل ۵-۳)^۸ استخراج و به فایل قابل Boot تبدیل کرد.

^۶ لینک دانلود: odroid.com/dokuwiki/doku.php?id=en:xu3_ubuntu_release_note_20161011 می‌توانید از این لینک نرم افزار را دانلود کنید: soft98.ir/software/compress/21-winrar-full.html

^۷ می‌توانید از این لینک نرم افزار را دانلود کنید: 7-zip.org/download.html

WE'RE NO LONGER UPDATING THIS WIKI!! [Go to New Wiki](#)

Trace: • [xu3_ubuntu_release_note_20161011](#) • [xu3_release_linux_ubuntu](#)

[Show pagesource](#) [Search](#)

Product

- ODROID-XU4
- ODROID-C2
- ODROID-XU4/CO
- Available display resolutions
- Use the USB-UART
- eMMC Bootloader Recovery
- eMMC Reference Chart
- SmartPower2
- USA IO BOARD
- WEATHER-BOARD
- ODROID-SHOW
- WEATHER-BOARD
- ODROID-UPS3

Software Release for Linux/Ubuntu on XU4/XU3

Ubuntu 16.04 (20161011)

- Ubuntu 16.04 Image - Download from [Main server](#) or [Mirror server](#)
- MD5SUM - Download from [Main server](#) or [Mirror server](#)
- Release Note (20161011)

Ubuntu 16.04 (20160708)

- Ubuntu 16.04 Image - Download from [Main Server](#) or [Mirror server](#)
- MD5SUM - Download from [Main server](#) or [Mirror server](#)
- Release Note (20160708)

Ubuntu 15.10 (20160114)

- Ubuntu 15.10 Image - Download from [Main server](#) or [Mirror server](#)
- MD5SUM - Download from [Main server](#) or [Mirror server](#)
- Release Note (20160114)

Ubuntu 15.04 (20150710)

- Ubuntu 15.04 Image - Download from [Main server](#) or [Mirror server](#)
- MD5SUM - Download from [Main server](#) or [Mirror server](#)
- Release Note (20150710)

Table of Contents

- Software Releases for Linux/Ubuntu on XU4/XU3
- Ubuntu 16.04 [20161011]
- Ubuntu 16.04 (20160708)
- Ubuntu 15.10 [20160114]
- Ubuntu 15.04 [20150710]
- Ubuntu 15.04 (20150528)
- Ubuntu 14.04 [20140522]
- Ubuntu 14.04.03 [20141105]

شکل ۳-۳: صفحه دانلود سیستم‌عامل

RARLAB® WinRAR®

Search

Language: English

Download WinRAR

If you don't know what you are looking for then you are probably looking for this WinRAR 64 bit version:

[!\[\]\(ca438df2c6b32a99f967c8e131a97780_img.jpg\)](#)

★★★★★ USER VOTES 4.5 stars by CNET

If you are looking for the WinRAR 32 bit version [click here](#), or did not find what you were looking for, please search below...

Select for download

Language	English	Version	All	Platform	All	Architecture	All
Latest WinRAR and RAR Versions English							
WinRAR 6.21 English 64 bit		3501 KB		Windows			
WinRAR 6.21 English 32 bit		3237 KB		Windows			

Download WinRAR and RAR Latest Version	Size	Platform
WinRAR 6.21 English 64 bit	3501 KB	Windows

شکل ۴-۳: صفحه دانلود WinRAR

بعد از استخراج فایل ubuntu-16.04-mate-odroid-xu3-20161011 نوبت به بوت کردن Mi-croSD می‌رسد. برای این کار از نرم‌افزار Win32DiskImager_v13^۹ استفاده شده است.

با قرار دادن MicroSD در کامپیوتر و انتخاب مسیر فایل و انتخاب دکمه Write Boot آغاز می‌شود. این کار چند دقیقه طول خواهد کشید و در نهایت اگر عملیات به درستی پایان یابد، پیغام Successful بر روی صفحه نمایان می‌شود و می‌توانید MicroSD را از کامپیوتر جدا کرده و آماده روشن کردن بُرد شوید.

^۹ می‌توانید از لینک زیر نرم‌افزار را دانلود کنید: bit.ly/1LVPcbF

The screenshot shows the official 7-Zip download page. At the top right is a large "Download" button. Below it is a table titled "Download 7-Zip 22.01 (2022-07-15)". This table lists various download links categorized by system type (Windows, Linux, macOS) and file type (.exe, .msi, .tar.xz, .7z). The second table, titled "Download 7-Zip 21.07 (2021-12-26)", shows a single link for macOS. On the left side of the page, there's a sidebar with links to Home, 7z Format, LZMA SDK, Download, FAQ, Support, and Links. Below the sidebar is a list of languages: English, Chinese Simpl., Chinese Trad., Esperanto, French, German, Indonesian, Japanese, Persian, Portuguese Brazil, Spanish, Thai, and Vietnamese.

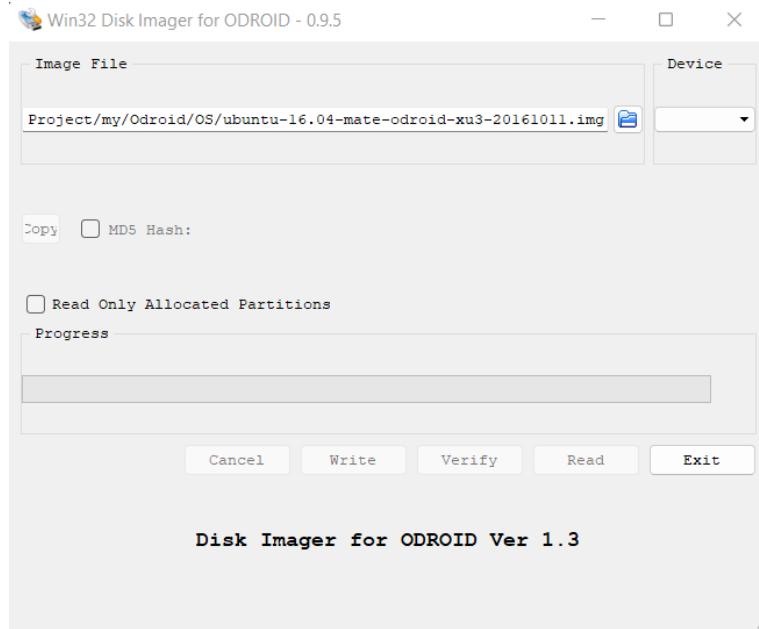
شکل ۵-۳: صفحه دانلود 7-Zip

۳-۳ روش نگاردن بورد

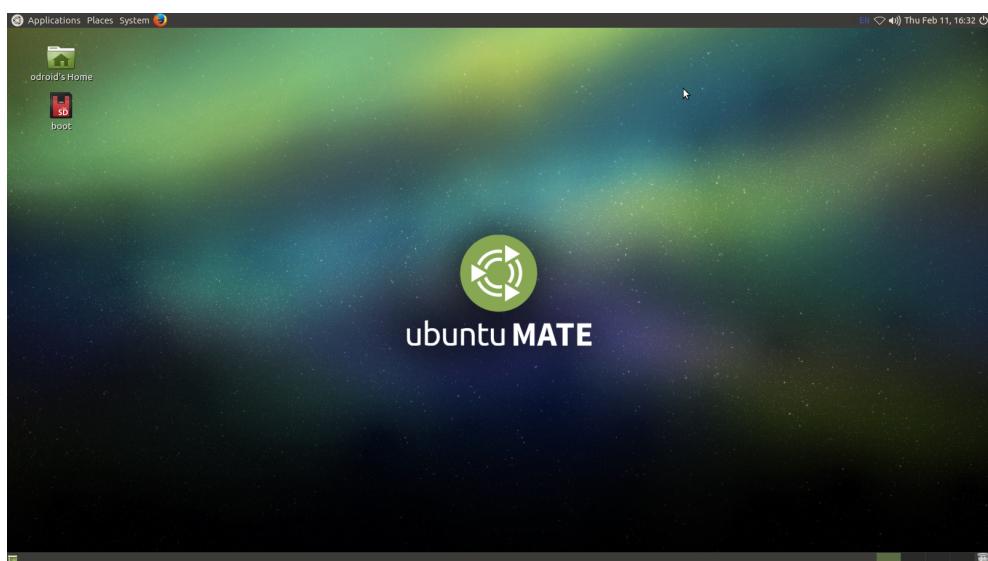
پس از آنکه عملیات Boot کردن سیستم عامل با موفقیت انجام شد، نوبت به روشن کردن بُرد می‌رسد. MicroSD را در محل خود طبق «شکل ۲-۳» قرار دهید. با استفاده از کابل HDMI بُرد را به مانیتور متصل می‌کنیم. موس و کیبورد راهم به خروجی‌های USB وصل می‌کنیم.

پس از آنکه از اتصال تمام لوازم جانبی به بُرد مطمئن شدیم، تغذیه بُرد را وصل می‌کنیم. اگر تغذیه در مدار جاری شود، یک LED قرمز رنگ روشن می‌شود. چند ثانیه پس از روشن شدن LED قرمز رنگ، LED آبی رنگی شروع به چشمک زدن می‌کند «مانند ضربان قلب» که این نشان دهنده Boot شدن سیستم عامل است. اگر LED آبی رنگ چشمک نزند، احتمالاً فرآیند Boot کردن سیستم عامل به درستی انجام نشده است و باید مجدداً سیستم عامل بر روی MicroSD Card نصب شود.

اگر سیستم به درستی Boot شود، صفحه ورود به سیستم عامل را مشاهده می‌کنید که باید رمز ورود سیستم وارد شود. رمزعبور سیستم به صورت پیش‌فرض odroid است. اگر به درستی رمزعبور را وارد نمایید، سیستم روشن شده و به طور کامل در اختیار شما قرار می‌گیرد.



شکل ۳-۶: نرم افزار Win32DiskImager



شکل ۷-۳: صفحه دسکتاپ اوبونتو

۴-۳ نصب پکیج‌ها و کتابخانه‌های ضروری

در ابتدا با زدن **Ctrl+Alt+T** یا کلیک راست بر روی صفحه و انتخاب گوینه **Open new terminal** یک صفحه ترمینال باز کرده و دستورات زیر را به ترتیب اجرا می‌کنیم: «در هر مرحله اگر نیاز به وارد کردن رمز عبور بود، **odroid** وارد شود.»

در هنگام نصب پکیج ها حتما از دسترسی سیستم به اینترنت اطمینان حاصل کنید. از آنجایی که در ۰droid-XU4 مژول Wifi وجود ندارد، نمی‌توان به صورت بی‌سیم^{۱۰} به اینترنت وصل شد. روشی USB که پیشنهاد می‌شود، اتصال از طرق نقطه‌اتصال^{۱۱} است. بدین صورت که با استفاده از یک کابل USB به اینترنت موبایل وصل شده و از تنظیمات موبایل، USB Hotspot Tethering را فعال کرده و به اینترنت وصل می‌شویم

۱-۴-۳ آپدیت سیستم

جهت دانلود و آپدیت Package ها و Dependency های قدیمی سیستم دستورات زیر وارد شود:

1. `$ sudo apt update`
2. `$ sudo apt-get upgrade`

۲-۴-۳ نصب git

جهت دانلود و نصب git دستورات زیر وارد شود:

1. `$ sudo apt install git`
2. `$ git --version`
`output: git version 2.7.4`

۳-۴-۳ نصب oh-my-zsh

این پکیج جهت طبقه‌بندی و جلوه بهتر ترمینال نصب می‌شود:

1. `$ sh -c "$(wget https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh -O -)"`

Wireless^{۱۰}
Hotspot tethering^{۱۱}

۴-۴-۳ نصب پایتون ۳

به صورت پیشفرض بر روی سیستم پایتون ۲.۷ نصب است اما بهجت ارتباط بهتر پکیج ها با پایتون، ورژن ۳ آن نصب شده است.

1. `$ sudo apt install software-properties-common`
2. `$ sudo add-apt-repository ppa:deadsnakes/ppa`
3. `$ sudo apt update`
4. `$ sudo apt install python3.5`
5. `$ python --version`
`output: python 3.5.2`

۵-۴-۳ تغییر ورژن پایتون

پس از نصب پایتون ۳ مراحل زیر را برای تغییر ورژن پیشفرض پایتون انجام دهید:

1. `$ sudo update-alternatives --install /usr/bin /python python /user/bin/python2.7 1`
2. `$ sudo update-alternatives --install /usr/bin /python python /user/bin/python3.5 2`
`output: update-alternatives: using /usr/bin/python3.5 to provide /usr/bin/python (python) in auto mode`
3. `$ sudo update-alternatives --config python`
`output: «۸-۳»`

۶-۴-۳ pip نصب

pip سیستم نصب و مدیریت پکیج و کتابخانه های پایتون است که به صورت زیر نصب می شود:

```

odroid@odroid:~ 
File Edit View Search Terminal Help
→ ~ sudo update-alternatives --install /usr/bin/python python /usr/bin/python^
2.7
update-alternatives: --install needs <link> <name> <path> <priority>
Use 'update-alternatives --help' for program usage information.
→ ~ sudo update-alternatives --install /usr/bin/python python /usr/bin/python2.
7 1
update-alternatives: using /usr/bin/python2.7 to provide /usr/bin/python (python
) in auto mode
→ ~ sudo update-alternatives --install /usr/bin/python python /usr/bin/python3.
5 2
update-alternatives: using /usr/bin/python3.5 to provide /usr/bin/python (python
) in auto mode
→ ~ sudo update-alternatives --config python
There are 2 choices for the alternative python (providing /usr/bin/python).

Selection    Path          Priority   Status
-----*----- /usr/bin/python3.5      2         auto mode
           /usr/bin/python2.7     1         manual mode
           /usr/bin/python3.5     2         manual mode

Press <enter> to keep the current choice[*], or type selection number: 2
→ ~

```

شکل ۸-۳: تغییر ورژن پیشفرض پایتون

اسکریپت نصب را از این لینک^{۱۲} دانلود کنید.

در محل دانلود یک صفحه ترمینال باز کنید و دستور زیر را تایپ کنید

3. `$ python get-pip.py`

4. `$ python -m pip install --upgrade pip`

5. `$ python -m pip --version`

`output: pip 20.3.4 from /home/odroid/.local/lib/python3.5/site-packages
(python 3.5)`

۷-۴-۳ نصب Idle IDE

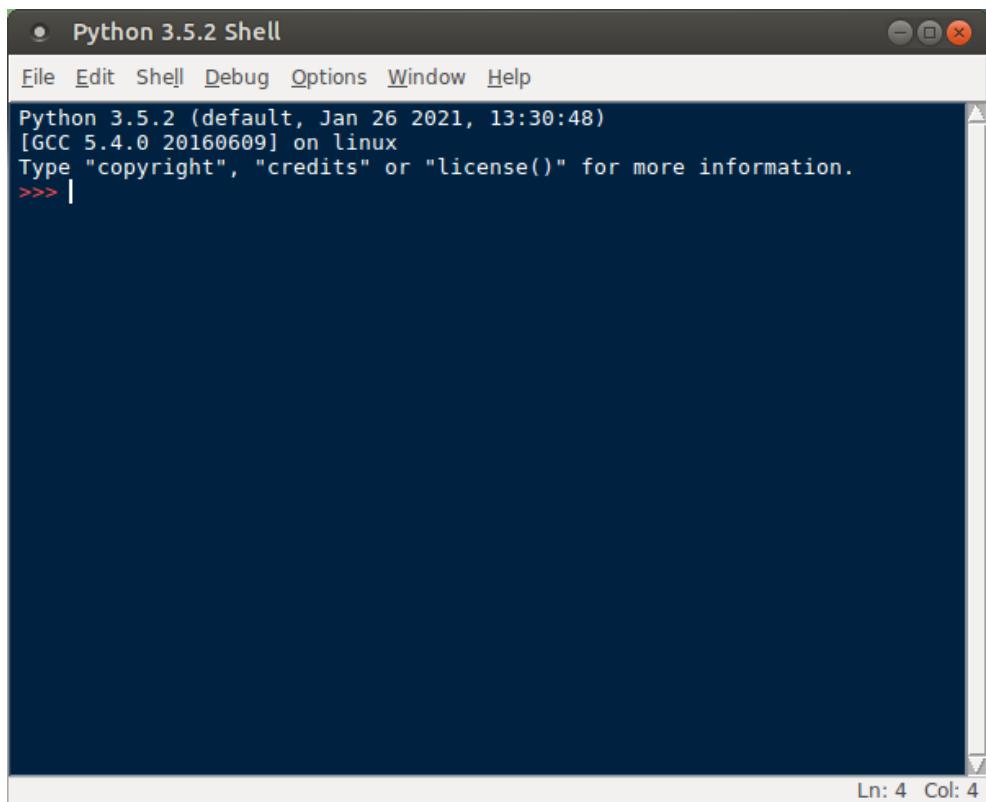
پیشفرض پایتون برای ادیت و اجرای کدهاست «شکل ۹-۳» که به صورت زیر نصب می‌شود:

1. `$ sudo apt-get update`

2. `$ sudo apt-get install idle3`

¹²bootstrap.pypa.io/get-pip.py

Integrated Development Environment^{۱۳}



شکل ۹-۳: محیط Idle

۸-۴-۳ نصب کتابخانه Numpy

کتابخانه ایست برای انجام محاسبات جبری و ماتریسی که به صورت زیر نصب می‌شود:

1. `$ pip3 install numpy`

۹-۴-۳ نصب کتابخانه Pandas

کتابخانه ایست جهت کار و استفاده از دیتاست‌ها که به صورت زیر نصب می‌شود:

1. `$ pip3 install pandas`

۱۰-۴-۳ نصب کتابخانه Pillow

کتابخانه ایست برای تصویر برداری که به صورت زیر نصب می‌شود:

1. `$ pip3 install Pillow`

۱۱-۴-۳ نصب کتابخانه Tkinter

کتابخانه ایست برای نوشتن رابط کاربری GUI^{۱۴} که به صورت زیر نصب می‌شود:

1. `$ pip3 install tk`

۱۲-۴-۳ نصب کتابخانه OpenCV

OpenCV^{۱۵}، کتابخانه ایست برای انجام کارهای پردازش تصویر، یادگیری ماشین و ... که به زبان C++ نوشته شده است. این کتابخانه رابط برنامه نویسی کاملی برای زبان‌های پایتون، جاوا و متلب دارد. در سیستم‌های مبتنی بر لینوکس و ویندوز، ورژن پایتون این کتابخانه با دستور `$ pip install opencv-python` نصب می‌شود. اما در بُرد ما این امکان فراهم نیست و نمی‌توان بدین صورت کتابخانه را نصب کرد.

برای نصب باید به صورت زیر سورس‌کد کتابخانه را برای سیستم‌عامل مورد نظر خودمان کامپایل کنیم.

۱. بهروز رسانی سیستم

(a) `$ sudo apt update`

(b) `$ sudo apt upgrade`

۲. ابزار زیر را نصب کنید

(a) `$ sudo apt -y install build-essential cmake gfortran
pkg-config unzip software-properties-common doxygen`

۳. بهروز رسانی Numpy، Pip، Python

(a) `$ sudo apt -y install python-dev python-pip python3-dev
python3-pip python3-testresources`

(b) \$ sudo apt -y install python-numpy python3-numpy

۴. موارد زیر را جهت تضمین نصب کامل OpenCV نصب کنید

- (a) \$ sudo apt -y install libblas-dev libblas-test liblapack-dev
libatlas-base-dev libopenblas-base libopenblas-dev
 - (b) \$ sudo apt -y install libjpeg-dev libpng-dev libtiff-dev
libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
 - (c) \$ sudo apt -y install libxvidcore-dev libx264-dev
 - (d) \$ sudo apt -y install libgtk2.0-dev libgtk-3-dev libcanberra-gtk*
 - (e) \$ sudo apt -y install libtiff5-dev libeigen3-dev libtheora-dev
libvorbis-dev sphinx-common libtbb-dev yasm libopencore-amrwb-dev
 - (f) \$ sudo apt -y install libopenexr-dev libgstreamer-plugins-base1.0-dev
libgstreamer1.0-dev libavutil-dev libavfilter-dev
 - (g) \$ sudo apt -y install libavresample-dev ffmpeg
libdc1394-22-dev libwebp-dev
 - (h) \$ sudo apt -y install libjpeg8-dev libxine2-dev
libfaac-dev libmp3lame-dev libopencore-amrnb-dev libprotobuf-dev
 - (i) \$ sudo apt -y install protobuf-compiler libgoogle-glog-dev
libgflags-dev libgphoto2-dev libhdf5-dev
 - (j) \$ sudo apt -y install qt5-default v4l-utils
 - (k) \$ sudo apt -y install libtbb2
- این مرحله کمی طول می کشد، صبور باشید...

۵. کتابخانه های قدیمی Ubuntu را نصب کنید

- (a) \$ sudo add-apt-repository "deb http://ports.ubuntu.com/ubuntu-ports
xenial-security main"

(b) `$ sudo apt -y update`

(c) `$ sudo apt -y install libjasper-dev libjasper`

۶. ایجاد پوشه جهت کامپایل OpenCV

(a) `$ mkdir opencv_package`

شما می‌توانید هر مسیری دیگری را برای کامپایل انتخاب کنید.

۷. دانلود فایل‌های OpenCV

(a) `$ wget -O opencv.zip`

<https://github.com/opencv/opencv/archive/4.1.2.zip>

(b) `$ wget -O opencv_contrib.zip`

https://github.com/opencv/opencv_contrib/archive/4.1.2.zip

۸. خارج کردن فایل‌های دانلود شده از حالت فشرده

(a) `$ unzip opencv.zip`

(b) `$ unzip opencv_contrib.zip`

۹. برای راحتی کار، پوشه‌های دانلود شده را تغییر نام دهید

(a) `$ mv opencv-4.1.2 opencv`

(b) `$ mv opencv_contrib-4.1.2 opencv_contrib`

۱۰. به محل پوشه ایجاد شده می‌رویم

- (a) `$ mv cd opencv`
- (b) `$ mv mkdir build`
- (c) `$ cd build`

۱۱. عملیات کامپایل را با دستورات زیر پیکر ہندی کنید

```
(a) $ cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D OPENCV_ENABLE_NONFREE=ON \
-D OPENCV_EXTRA_MODULES_PATH=/home/odroid/Desktop/opencv_contrib/modules \
\
-D PYTHON_EXECUTABLE=/usr/bin/python3.6 \
-D PYTHON2_EXECUTABLE=/usr/bin/python2.7 \
-D PYTHON3_EXECUTABLE=/usr/bin/python3.6 \
-D PYTHON_PACKAGES_PATH=/usr/lib/python3/dist-packages \
-D PYTHON_LIBRARY=/usr/lib/python3.6/config-3.6m-arm-linux-gnueabihf/
libpython3.6m.so \
-D PYTHON_INCLUDE_DIR=/usr/include/python3.6 \
-D PYTHON3_NUMPY_INCLUDE_DIRS=/usr/lib/python3
/dist-packages/numpy/core/include \
-D OPENCV_GENERATE_PKGCONFIG=ON \
-D OPENCV_PYTHON3_INSTALL_PATH=/home/odroid/Desktop/opencv_package \
\
-D INSTALL_PYTHON_EXAMPLES=OFF \
-D INSTALL_C_EXAMPLES=OFF \
-D BUILD_DOCS=NO \
-D BUILD_TIFF=ON \
-D WITH_FFMPEG=ON \
-D WITH_GSTREAMER=ON \
```

```
-D WITH_TBB=ON \
-D BUILD_TBB=ON \
-D WITH_V4L=ON \
-D WITH_LIBV4L=ON \
-D WITH_VTK=OFF \
-D WITH_OPENGL=ON \
-D BUILD_NEW_PYTHON_SUPPORT=ON \
-D BUILD_TESTS=OFF \
-D BUILD_EXAMPLES=OFF ..
```

۱۲. با دستور زیر، فرایند کامپایل را آغاز کنید

(a) `$ make -j4`

بورد Odroid-XU4 دارای ۸ هسته است. با اعمال کد بالا از نصف ظرفیت بورد برای کامپایل استفاده می‌کنیم. ضمن اینکه عملیات کامپایل چندین ساعت طول می‌کشد. صبور باشید...

۱۳. با نصب ۴.۱.۲ OpenCV عملیات کامپایل را به پایان برسانید

(a) `$ sudo make install`

(b) `$ sudo ldconfig`

(c) `$ sudo apt update`

۱۴. مرحله آخر

نصب OpenCV ۴.۱.۲ به پایان رسید است. با دستور زیر می‌تواند از نصب آن اطمینان حاصل کنید.

(a) `$ python`

(b) `$ import cv2`

(c) `$ cv2.__version__`

```
output: '4.1.2'
```

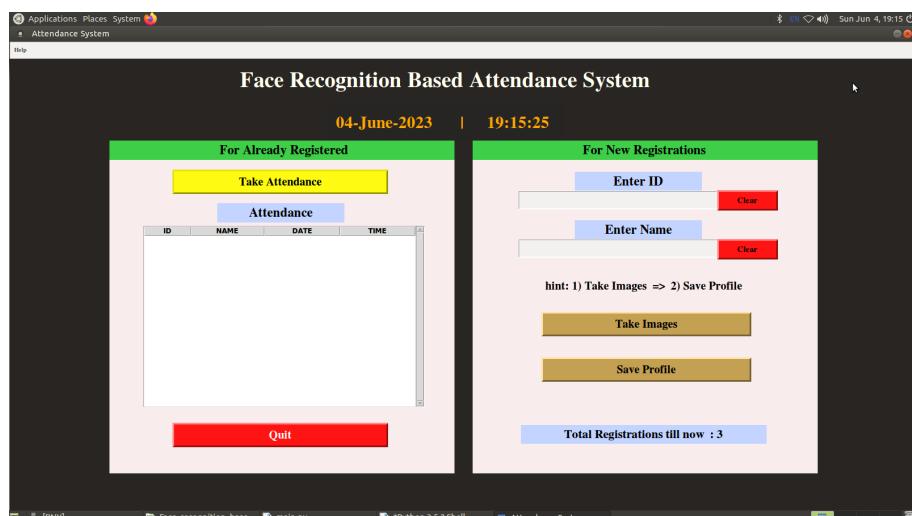
فصل ۴

اجرای الگوریتم و نتایج

در این فصل به بررسی الگوریتم نوشته شده و اجرای آن و بررسی خروجی‌ها می‌پردازیم. هسته اصلی کد از این لینک^۱ دانلود شده و آن را دستخوش تغییرات دادیم و برای آن رابط کاربری نوشتیم.

۱-۴ رابط کاربری برنامه

برای الگوریتم نوشته شده، یک برنامه^۲ به زبان پایتون «بخشی از کد ها در پیوست آ، آورده شده» با رابط کاربری آسان نوشته شده است. «شکل ۱-۴»



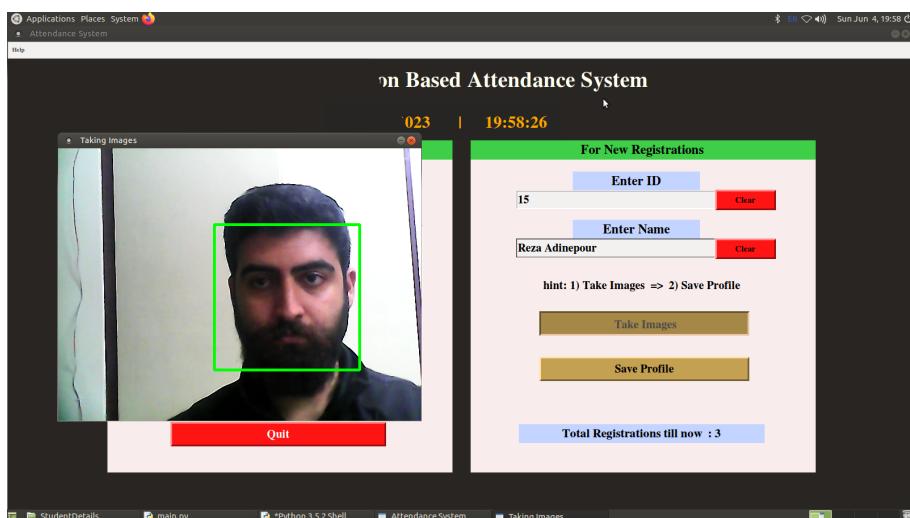
شکل ۱-۴: رابط کاربری برنامه

[github.com/kunalyelne/Face-Recognition-using-Raspberry-Pi^۱](https://github.com/kunalyelne/Face-Recognition-using-Raspberry-Pi)
[Application^۲](#)

این برنامه از ۳ بخش کلی تشکیل شده است:

۱. بخش اول: وارد کردن ID و نام کاربر جهت تهیه عکس از فرد و ذخیره آن در دیتابست برنامه.
۲. بخش دوم: آموزش شبکه بر روی دیتابست تهیه شده.
۳. بخش سوم: تست شبکه آموزش دیده.

با وارد کردن ID و نام کاربر و زدن دکمه Take Image، دوربین متصل به بُرد فعال شده و ۱۰۰ عدد عکس از کاربر می‌گیرد^۳ «شکل ۲-۴» و با نام‌های User.Id و User.Username که به ترتیب Id و اسم و آیدی وارد شده و شماره عکس است. ضمن اینکه تمامی ID ها و Username های وارد شده در فایل اکسل با نام «شکل ۳-۴ StudentDetails ذخیره می‌شوند. «شکل ۳-۴



شکل ۲-۴: تهیه دیتابست

پس از آنکه عکس‌ها گرفته شد، با استفاده از دکمه Save Profile، می‌توان شبکه را Train کرد. با زدن این دکمه باکسی نمایش داده می‌شود که باید Password سیستم را وارد کرد. به صورت پیش‌فرض، پسورد سیستم 9814303 در نظر گرفته شده است. با وارد کردن پسورد و زدن کلید Ok، شبکه Train می‌شود.

۲-۴ فاز آموزش شبکه

پس از تهیه دیتابست، با زدن دکمه Save Profile، باید شبکه را آموزش داد. با وارد کردن Password برنامه، این Password در تابع psw با مقدار ذخیره شده آن در تنظیمات برنامه مقایسه می‌شود و اگر مقدار آن درست بود وارد تابع TrainImages می‌شود و فرایند آموزش آغاز می‌شود.

^۳ از این لینک می‌توانید دیتابست را دانلود کنید: b2n.ir/dataset

شکل ۳-۴: اطلاعات ذخیره شده کاربران

در ابتدای ورود به تابع `TrainImages` چک می‌شود که آیا فایل ضرایب Haar که در فایلی با نام `haarcascade_frontalface_default.xml` ذخیره ضرایب آموزش دیده اطمینان حاصل می‌کنیم. سپس با این دستور `cv2.face_LBPHFaceRecognizer.create()` یک شیء برآش شناسایی چهره‌ها می‌سازیم. سپس همه عکس‌های و آیدی‌های موجود در پوشه دیتاست با نام `TrainingImage` را به ترتیب در تابعی با نام `getImagesAndLabels` به شبکه می‌دهیم و با ماژول `train` ضرایب مطلوب را بدست می‌آوریم و در فایلی با نام `Trainner.yml` ذخیره می‌کنیم.

۳-۴ فاز تست و شناسایی چهره‌ها

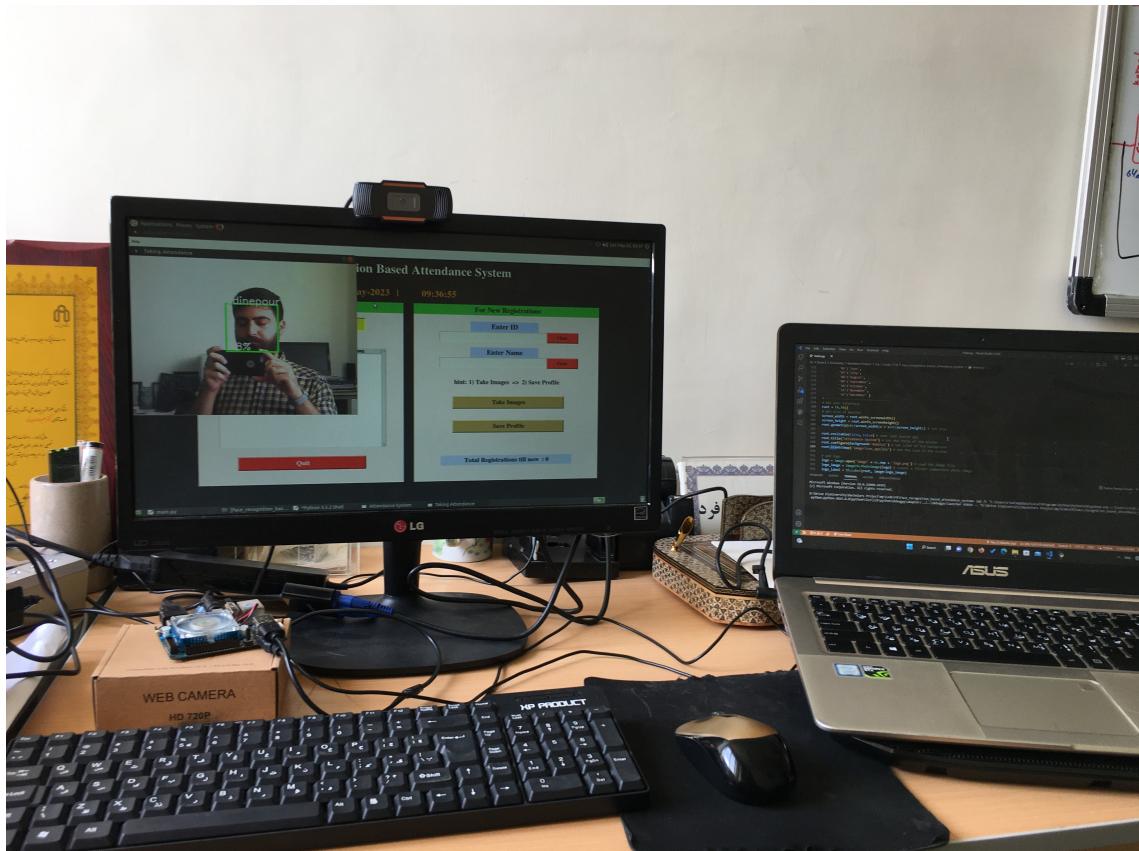
پس از آنکه آموزش شبکه به اتمام رسید، نوبت به تست شبکه می‌رسد. با زدن دکمه `Take Attendance`، فاز شناسایی چهره‌های از پیش آموزش داده شروع می‌شود. «شکل ۴-۴» این فاز در تابعی به نام `TrackImages` انجام شده است. در ابتدای کار از وجود مسیرهای ذخیره‌سازی افراد شناسایی شده و اطلاعات دیتاست اطمینان حاصل می‌کنیم. و همانند فاز آموزش یک شیء با دستور `cv2.face.LBPHFaceRecognizer_create()` می‌سازیم و محتویات فایل `Trainner.yml` ذخیره شده در مرحله قبل را می‌خوانیم.

شیء دیگری برای طبقه‌بند با دستور `cv2.CascadeClassifier(harcascadePath)` ایجاد می‌کنیم و با دستور `cv2.VideoCapture(0)` به دوربین متصل شده به بُرد وصل می‌شویم و در حلقه‌ی بینهایت به طور متوالی فریم‌های دوربین را دریافت می‌کنیم و در تصویر دریافت شده ابتدا چهره‌ها را در تصویر پیدا

^۴ Directory

می‌کنیم و دور آنها با عرض و ارتفاع مناسب مستطیل می‌کشیم.

سپس با استفاده از ماژول predict چهره موجود در تصویر و ضریب اطمینان^۵ آن را که عددی بین ۰ تا ۱۰۰ است بدست می‌آوریم. اگر ضریب اطمینان بدست آمده بین ۵۰ تا ۵۰ «در این کد آستان اطمینان ۵۰ در نظر گرفته شده است» باشد چهره به درستی شناسایی شده است. هر چقدر مقدار این آستانه را بالاتر ببریم خطای در شناسایی افزایش می‌یابد.



شکل ۴-۴: تست سخت‌افزار و الگوریتم

فصل ۵

نتیجه‌گیری

در این پایان‌نامه به بررسی و پیاده سازی سیستمی بلادرنگ برای تشخیص چهره مبتنی بر ویژگی‌های LBP و طبقه‌بند آبشاری پرداخته شد که در فصل‌های ۱، ۲ به معرفی ساختار طبقه‌بند‌ها و تئوری مورد نیاز پایان‌نامه پرداخته شد. در فصل ۳ سخت‌افزار استفاده شده برای ساخت معرفی شد و نحوه راه‌اندازی آن بررسی شد و در فصل ۴ نتایج بدست آمده گزارش شده است.

با گسترش تکنولوژی و هوش‌مصنوعی، امید است سیستم‌های سنتی حضور و غیاب در ادارت، سیستم‌های شناسایی چهره در محیط‌های عمومی و ... دسخوش تغییراتی بشوند، از این رو در این پایان‌نامه در جهت حرکت به این سوگام برداشتیم. سیستم طراحی شده در این پایان‌نامه می‌تواند جایگزین مناسبی برای سیستم حضور و غیاب در اداره‌ها و دانشگاه‌ها باشد.

هسته اصلی سخت‌افزار این پایان‌نامه مینی کامپیوتر Odroid است که به دلیل سهولت در جابه‌جایی از آن استفاده شده است. دلیل دوم استفاده از این بُرد سرعت پردازش بالای آن در مقایسه با بُردهای هم‌چون Raspberry pi است.

در پیاده‌سازی این پروژه از یک عدد Webcam و مانیتور استفاده شده است. استفاده از مانیتور‌های معمولی سهولت در جابه‌جایی و Portable بودن سیستم را از بین می‌برد از این جهت پیشنهاد می‌شود از مانیتور‌های کوچک با قابلیت تاچ خازنی استفاده شود تا به طور کامل Portable بودن سیستم تضمین شود.

و در نهایت از تمام کسانی که این پایان‌نامه را مطالعه کردند عرض تشکر دارم و درخواست دارم تا نقطه‌نظراتون را با ایمیل زیر با بنده درمیان بگذارید:

reza_adinepour@shahroodut.ac.ir

Bibliography

- [1] S. M. Piryonesi and T. E. El-Diraby. Role of data analytics in infrastructure asset management: Overcoming data size and quality problems. *Transportation Engineering*, 2020.
- [2] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. 2001.
- [3] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *IEEE*, 2001.
- [4] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray scale and rotation invariant texture classification with local binary patterns. *IEEE*, 2002.
- [5] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. *Transportation Engineering*, 2004.

واژه‌نامه

function	تابع	الف
density	تراکم	آستانه
approximation	تقریب	انتقال
partition	تقسیم‌بندی	الگو
theory	تئوری	آبشاری
adaptive	تطبیقی	آموزش
transform	تبدیل	

ب

black box	جعبه سیاه	online	برخط
		optimum	بهینه
		maximum	بیشینه
		computer vision	بینایی ماشین
		real time	بلادرنگ
		unsupervised	بدون ناظارت
		vector	بردار
		wireless	بی‌سیم
		application	برنامه

پ

data	داده	complexity	پیچیدگی
------------	------------	------------------	---------------

ت

operational system	سیستم عامل
--------------------------	------------------

ک

efficient کارا

minimum کمینه

م

set مجموعه

local محلی

open source متن باز

artifitial مصنوعی

hidden مخفی

directory مسیر

ش

شبه کد

pseudo code شبکه کد

object شیء

ض

confidence ضریب اطمینان

ط

classifier طبقه بند

ن

embedded نهفته

nearest neighbor نزدیک ترین همسایه

غ

dominate غلبه

و

weight وزن

ف

distance فاصله

space فضا

activation فعال ساز

ه

artificial intelligence هوش مصنوعی

converge همگرا

ق

deterministic قطعی

portable قابل حمل

پیوست آ

مطالب تکمیلی

در این قسمت، بخشی از کدهای مهم پروژه آورده شده است. فایل‌های اصلی پروژه را می‌توانید از این لینک^۱ دریافت کنید.

آ-۱ استخراج بردار ویژگی‌های LBP

در این قسمت برنامه‌ای ساده برای استخراج ویژگی‌های LBP و تشکیل بردار ویژگی‌ها نوشته شده است.

- برنامه استخراج ویژگی‌های LBP

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 from skimage.feature import local_binary_pattern
4 from skimage.color import gray2rgb, rgb2gray
5 from skimage.io import imread
6
7 # Load the image
8 img = imread('reza.jpg')
9
10 # Convert the image to grayscale if it is in RGB format
11 if (img.ndim == 3):
12     img = rgb2gray(img)
13
14 # Compute the LBP transformation
15 lbp = local_binary_pattern(img, 8, 1.5)
16
17 # Convert the LBP image to a color image for display purposes
18 lbp_color = gray2rgb(lbp)
```

^۱ لینک دریافت: github.com/rezaAdinepour/Bachelors-Project

```

19
20 # Display the original and LBP transformed images
21 fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(10, 5))
22
23 ax1.imshow(img, cmap=plt.cm.gray)
24 ax1.set_title('Original Image')
25
26 ax2.imshow(lbp_color)
27 ax2.set_title('LBP Transformed Image')
28
29 for ax in (ax1, ax2):
30     ax.axis('off')
31
32 plt.show()
33
34 # Plot the pixel values of the LBP image
35 fig, ax = plt.subplots()
36 ax.plot(np.arange(0, 256), np.histogram(lbp, bins=np.arange(0, 257))[0])
37 ax.set_xlabel('LBP Prototype')
38 ax.set_ylabel('% of Pixels')
39 ax.set_title('Local Binary Pattern')
40
41 plt.show()
42
43 ax.hist(lbp, 5, density=True, facecolor='g', alpha=0.75)
44 plt.show()

```

آ-۲ تست دوربین

در این قسمت ابتدا از اتصال دوربین به بورد اطمینان حاصل می‌کنیم.

- برنامه اتصال دوربین به بورد

```

1 import numpy as np
2 import cv2
3
4 cap = cv2.VideoCapture(0)
5 cap.set(3, 640) # set Width
6 cap.set(4, 480) # set Height
7
8 while(True):
9     ret, frame = cap.read()
10    # frame = cv2.flip(frame, -1) # Flip camera vertically
11    if (ret):
12        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

```

```

13     cv2.imshow('frame', frame)
14     cv2.imshow('gray', gray)
15
16     k = cv2.waitKey(30) & 0xff
17     if (k == 27): # press 'ESC' to quit
18         break
19 cap.release()
20 cv2.destroyAllWindows()

```

آ-۳ فاز آموزش

- تابع نوشته شده در بخش آموزش برنامه به صورت زیر است:

```

1 # train phase
2 def TrainImages():
3     check_haarcascadefile()
4     assure_path_exists('TrainingImageLabel' + os.sep)
5     recognizer = cv2.face_LBPHFaceRecognizer.create()
6     harcascadePath = 'haarcascade_frontalface_default.xml'
7     detector = cv2.CascadeClassifier(harcascadePath)
8     faces, ID = getImagesAndLabels('TrainingImage')
9     try:
10         recognizer.train(faces, np.array(ID))
11     except:
12         mess._show(title='No Registrations', message='Please Register someone
13             first!!!!')
14     return
15     recognizer.save('TrainingImageLabel' + os.sep + 'Trainer.yml')
16     res = 'Profile Saved Successfully'
17     message1.configure(text=res)
18     message.configure(text='Total Registrations till now : ' + str(max(ID)))

```

آ-۴ فاز تست و شناسایی چهره

در بحث شناسایی چهره، مهمترین گام تشخیصی محل چهره در تصویر است که در این قسمت آن را تست کردیم.

- برنامه تشخیص چهره

```

1 import numpy as np
2 import cv2
3

```

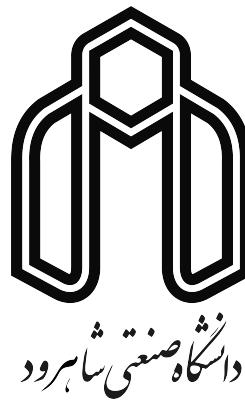
```
4 faceCascade =
5     cv2.CascadeClassifier(r'/home/odroid/Desktop/Face-Recognition-using-Odroid/
6                             haarcascades/haarcascade_frontalface_default.xml')
7 cap = cv2.VideoCapture(0)
8 cap.set(3,640) # set Width
9 cap.set(4,480) # set Height
10
11 while (True):
12     ret, img = cap.read()
13     #img = cv2.flip(img, -1)
14     gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
15     faces = faceCascade.detectMultiScale(
16         gray,
17         scaleFactor = 1.2,
18         minNeighbors = 5,
19         minSize=(20, 20) )
20     for (x, y, w, h) in faces:
21         cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
22         roi_gray = gray[y:y+h, x:x+w]
23         roi_color = img[y:y+h, x:x+w]
24         cv2.imshow('video', img)
25         k = cv2.waitKey(30) & 0xff
26         if (k == 27): # press 'ESC' to quit
27             break
28 cap.release()
29 cv2.destroyAllWindows()
```

Abstract

The purpose of this thesis was to design and implement a real-time system for detect and recognition people's faces based on local binary patterns. This system is implemented on Odroid embedded computer.

In this design, the OpenCV open source library and specifically the cascade and K-NN classifier are used. One of the advantages of this system is its portability, which can be used in various environments that require this application, such as offices for attendance and absence and recording the hours of arrival and departure of people.

Keywords: Classifier, Local Binary Pattern, Embedded Computer



Shahrood University of Technology

Department of Electrical Engineering

B.Sc. Thesis

Design Real Time Face Recognition Systems Based on LBP Features on Odroid XU4 Embedded Computer Board

By:

Reza Adinepour

Supervisor:

Dr. Alireza Ahmadyfard

Jun 2023