



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)  
دانشکده مهندسی کامپیوتر

پایان نامه کارشناسی

طراحی و پیاده سازی برنامه کاربردی اندرویدی برای حضور و غیاب با  
استفاده از تشخیص چهره و یادگیری ماشین

نگارش  
امیرعلی بلباسی

استاد راهنما  
دکتر علیرضا باقری

مهر ۱۴۰۳



بسمه تعالی  
برگ ارزیابی پروژه کارشناسی

تاریخ دفاع: ۱۴۰۳/۲/۲۸

نام و نام خانوادگی: امیرعلی بلایی  
رشته و گرایش تحصیلی: مهندسی کامپیوتر  
عنوان پایان نامه: طراحی و پیاده سازی برنامه کاربردی اندرویدی برای حضور و غیاب با استفاده از تشخیص چهره و یادگیری ماشین

شماره دانشجویی: ۹۸۳۱۱۰۹  
دانشگاه: مهندسی کامپیوتر

نیعیسال ثبت نامی: ۱۴۰۲ ✓  
تاریخ تصویب: ۱۴۰۳/۳/۹ ✓

تاریخ تحویل فرم پیشنهاد: ۱۴۰۳/۲/۲۵  
تاریخ تحویل پایان نامه: ۱۴۰۳/۲/۱۸

ارزیابی محتوی پایان نامه	نام و نام خانوادگی	کد استاد	نمره	امضاء
استاد راهنمای اول	علیرضا باقری	۱۰۹۲۱	۱۴٫۵	
استاد راهنمای دوم			۱۷	
استاد داور	مهرن سلیمان فلاح	۱۰۵۱۲	۱۴٫۵	

ارزیابی وقت شناسی و رعایت زمان بندی اجرای پایان نامه	نمره پیشنهادی آموزش دانشکده	نمره نهایی هیئت داوران
تحویل به موقع فرم پیشنهاد (نمره ۱/۵)	۱	۲٫۵
تحویل به موقع پایان نامه (نمره ۱/۵)	۱٫۵	

شرح	نمره به عدد	نمره به حروف
ارزیابی محتوی پایان نامه	۱۴٫۵	چهار و نیم
ارزیابی وقت شناسی و رعایت زمان بندی اجرای پایان نامه	۲٫۵	دو و نیم نمره
نمره نهایی	۱۷	هفده

معاون آموزشی دانشکده  
امضاء و مهر  
تاریخ چاپ: ۱۴۰۳/۲/۲۵

۱۴۳۴/۲/۲۸



به نام خدا

تاریخ: ۱۴۰۳/۰۷/۲۸

## تعهدنامه اصالت اثر

اینجانب امیرعلی بلباسی متعهد می‌شوم که مطالب مندرج در این پایان نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی اساتید 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 مآخذ ذکر گردیده است. این پایان نامه قبلاً 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 پایان نامه 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 پایان نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مآخذ بلامانع است.

امیرعلی بلباسی

## تقدیم

این پایان‌نامه را با عشق و احترام به خانواده عزیزم تقدیم می‌کنم، که با حمایت بی‌دریغ، صبر و تشویق‌هایشان در تمام مراحل زندگی و تحصیل همواره در کنارم بودند. همچنین از دوستان گرامی‌ام که با همفکری، همراهی و انگیزه‌بخشیدن‌هایشان در این مسیر دشوار به من کمک کردند، سپاسگزارم. بدون حضور شما، پیمودن این راه ممکن نبود.

با احترام،

امیرعلی بلباسی

## تقدیر و تشکر

با نهایت احترام و قدردانی، از استاد راهنمای محترم، دکتر علیرضا باقری سپاسگزارم که با دانش، تجربه و راهنمایی‌های ارزشمندشان همواره چراغ راه من در این مسیر بودند و با صبر و حوصله، مرا در انجام این پایان‌نامه یاری دادند. همچنین از استاد مشاور گرامی‌ام، دکتر سجاد شیرعلی شهرضا تشکر ویژه دارم که با دیدگاه‌های سازنده و پیشنهادات مفیدشان نقش مهمی در بهبود کیفیت این کار داشتند. حضور و حمایت شما، الهام‌بخش و انگیزه‌بخش این سفر علمی بود.

با احترام،

امیرعلی بلباسی

## چکیده

در این پروژه، یک برنامه اندرویدی طراحی و توسعه داده شد که با استفاده از محیط Android Studio و زبان برنامه‌نویسی Kotlin پیاده‌سازی شده است. هدف این برنامه، تسهیل و تسريع فرآیند حضور و غیاب برای اساتید دانشگاهی است. برای انجام این کار، یک مدل تشخیص چهره با استفاده از کتابخانه Torch و الگوریتم‌های یادگیری ماشین در زبان Python توسعه یافت. مدل مذکور پس از تمرین با داده‌های مربوطه، قادر به تشخیص چهره دانشجویان از طریق یک عکس گرفته شده در محیط برنامه است. پس از شناسایی موفقیت‌آمیز چهره، نام دانشجو به طور خودکار در لیست حضور و غیاب علامت‌گذاری می‌شود. این سیستم نه تنها باعث افزایش سرعت و دقت در ثبت حضور و غیاب می‌شود، بلکه امکان کاهش خطاهای انسانی و بهبود مدیریت کلاس‌ها را نیز فراهم می‌آورد.

## واژه‌های کلیدی:

اندروید، هوش مصنوعی، یادگیری ماشین، حضور و غیاب

## صفحه

## فهرست مطالب

چکیده.....	أ
فصل اول مقدمه.....	۱
فصل دوم بررسی مدل تشخیص چهره.....	۴
۲-۱- مروری بر مفاهیم پایه تشخیص چهره.....	۶
۲-۲- الگوریتم‌های تشخیص چهره.....	۹
۲-۳- کتابخانه torch.....	۱۳
۲-۳-۱- ویژگی‌ها و مزایای کتابخانه Torch.....	۱۴
۲-۳-۲- کاربرد Torch در پروژه‌های تشخیص چهره.....	۱۵
۲-۴- مدل استفاده شده در این پروژه (Siamese).....	۱۶
۲-۴-۱- ساختار مدل Siamese.....	۱۶
۲-۴-۲- نحوه عملکرد مدل در پروژه.....	۱۸
۲-۴-۳- تابع هزینه و آموزش مدل.....	۱۸
۲-۵- پیش‌پردازش داده‌ها.....	۱۹
۲-۵-۱- دیتاست lfw.....	۱۹
۲-۵-۲- مراحل پیش‌پردازش.....	۲۰
۲-۶- آموزش مدل.....	۲۲
۲-۷- ادغام مدل با برنامه اندرویدی.....	۲۵
فصل سوم بررسی برنامه اندرویدی.....	۲۷
۳-۱- محیط Android Studio.....	۲۹
۳-۲- تکنولوژی Jetpack Compose.....	۳۱
۳-۳- ابزار ساخت نرم افزار Gradle.....	۳۳
۳-۴- پیکربندی ساخت نرم افزار.....	۳۴
۳-۵- کتابخانه CameraX.....	۳۶
فصل چهارم کاربرد و محیط نرم افزار.....	۴۲
۴-۱- کاربرد برنامه.....	۴۳
۴-۲- تصاویری از محیط نرم افزار به همراه استفاده.....	۴۴
فصل پنجم جمع‌بندی و نتیجه‌گیری و پیشنهادات.....	۵۳
منابع و مراجع.....	۵۶
Abstract.....	۵۷

شکل ۱: معماری یک siamese model.....	۱۷
شکل ۲: معماری Network 1.....	۱۷
شکل ۳: توصیف تابع loss.....	۱۹
شکل ۴: خروجی‌های تمرین مدل در چهار epoch اول.....	۲۴
شکل ۵: نمودار تمرین مدل.....	۲۵
شکل ۶: تابع برگرداننده میزان شباهت دانشجویان.....	۴۰
شکل ۷: نحوه Navigation کاربر.....	۴۱
شکل ۸: گرفتن Permission استفاده از دوربین.....	۴۵
شکل ۹: گرفتن Permission دسترسی به Media.....	۴۶
شکل ۱۰: محیط خانه برنامه.....	۴۷
شکل ۱۱: نام دانشجو باید وارد شود.....	۴۸
شکل ۱۲: با زدن دکمه اول دوربین باز شده و امکان گرفتن عکس وجود دارد.....	۴۹
شکل ۱۳: دانشجو به همراه چک باکس اضافه شده است.....	۵۰
شکل ۱۴: با زدن دکمه دوم مجدد دوربین باز شده.....	۵۱
شکل ۱۵: با بازگشت به صفحه اصلی، مشاهده می‌شود که نام دانشجو به عنوان حاضر تیک خورده است.....	۵۲



## فصل اول

### مقدمه

## مقدمه

در عصر حاضر، فناوری‌های نوین در حوزه هوش مصنوعی و یادگیری ماشین تحولات چشمگیری را در زمینه‌های مختلف به وجود آورده‌اند. یکی از این زمینه‌ها، سیستم‌های تشخیص چهره است که به دلیل قابلیت‌های منحصر به فرد خود در شناسایی هویت و بررسی حضور افراد، به سرعت در حال تبدیل شدن به یک استاندارد جهانی در حوزه‌های امنیت، نظارت، شناسایی هویت و مدیریت حضور و غیاب است. با پیشرفت‌های اخیر در یادگیری عمیق و الگوریتم‌های هوش مصنوعی، دقت و سرعت این سیستم‌ها به طور قابل توجهی افزایش یافته است.

روش‌های سنتی حضور و غیاب که معمولاً به ثبت دستی یا استفاده از کارت‌های شناسایی متکی هستند، به طور عمده با محدودیت‌هایی مواجه‌اند. این محدودیت‌ها شامل وقت‌گیر بودن، احتمال تقلب، و خطاهای انسانی می‌شود که می‌تواند به ثبت نادرست حضور منجر شود. این مشکلات نه تنها منجر به اتلاف وقت و منابع می‌شوند، بلکه در برخی موارد می‌توانند باعث ایجاد تنش در محیط‌های آموزشی و کاری شوند. به همین دلیل، نیاز به سیستم‌های خودکار و هوشمند که قادر به انجام این فرآیند با دقت و سرعت بالا باشند، بیش از پیش احساس می‌شود.

در این پایان‌نامه، به توسعه و پیاده‌سازی یک سیستم حضور و غیاب مبتنی بر تشخیص چهره پرداخته شده است. این سیستم با استفاده از یک برنامه اندرویدی که در محیط Android Studio و با زبان برنامه‌نویسی Kotlin توسعه داده شده، عمل می‌کند. علاوه بر این، از یک مدل تشخیص چهره بهره برده می‌شود که با استفاده از کتابخانه Torch و تکنیک‌های یادگیری ماشین در زبان پایتون طراحی و تمرین داده شده است. این سیستم به اساتید دانشگاه این امکان را می‌دهد که تنها با گرفتن یک عکس از هر دانشجو، حضور آن‌ها را به صورت خودکار ثبت کنند.

هدف اصلی این پروژه، تسهیل و تسریع فرآیند حضور و غیاب است. با به‌کارگیری فناوری‌های پیشرفته، علاوه بر کاهش خطاهای انسانی، بهینه‌سازی زمان و افزایش دقت در ثبت حضور نیز محقق

می‌شود. این سیستم نه تنها می‌تواند به کاهش بار اداری کمک کند، بلکه با ارائه اطلاعات دقیق و به موقع، می‌تواند فرآیندهای تصمیم‌گیری در محیط‌های آموزشی و سازمانی را بهبود بخشد.

در این پایان‌نامه، به تفصیل در مورد روش‌ها، ابزارها و الگوریتم‌های مورد استفاده در توسعه این سیستم بحث شده و نتایج نهایی آن مورد ارزیابی قرار خواهد گرفت. این ارزیابی‌ها نشان می‌دهد که چگونه استفاده از تشخیص چهره می‌تواند به یک راه‌حل موثر و کارآمد برای مشکلات موجود در روش‌های سنتی حضور و غیاب تبدیل شود و به ایجاد یک سیستم هوشمند و خودکار در این زمینه کمک کند.

## فصل دوم

### بررسی مدل تشخیص چهره

## بررسی مدل تشخیص چهره

تشخیص چهره یکی از شاخه‌های مهم و پرکاربرد در حوزه بینایی ماشین و هوش مصنوعی است که طی سال‌های اخیر به دلیل پیشرفت‌های چشمگیر در زمینه یادگیری عمیق و پردازش تصاویر دیجیتال، مورد توجه ویژه‌ای قرار گرفته است. این فناوری توانایی شناسایی و تشخیص هویت افراد را بر اساس ویژگی‌های ظاهری چهره دارد و در طیف گسترده‌ای از کاربردها مانند امنیت، پرداخت الکترونیکی، تعامل انسان و کامپیوتر، نظارت و پایش، و سیستم‌های حضور و غیاب مورد استفاده قرار می‌گیرد. پیشرفت‌های اخیر در این حوزه باعث شده تا سیستم‌های تشخیص چهره با دقت و سرعت بالاتری نسبت به گذشته عمل کنند و به عنوان یکی از ابزارهای کلیدی در سیستم‌های هوشمند شناخته شوند.

در این فصل، به بررسی اصول و روش‌های مختلف تشخیص چهره، از جمله الگوریتم‌ها و ابزارهای مرتبط، و چگونگی به کارگیری آن‌ها در پروژه حاضر پرداخته می‌شود. در این پروژه، هدف اصلی توسعه یک مدل تشخیص چهره است که بتواند به طور خودکار افراد را از روی تصاویر شناسایی کرده و در برنامه‌های کاربردی اندرویدی، به ویژه در سیستم‌های حضور و غیاب دانشگاهی، به کار رود.

تشخیص چهره به معنای توانایی یک سیستم کامپیوتری در شناسایی یا تأیید هویت یک فرد از طریق تحلیل تصاویر یا ویدیوهای دیجیتال است. این سیستم‌ها به دلیل امکان شناسایی سریع و غیرتماسی، به عنوان جایگزینی برای روش‌های سنتی احراز هویت نظیر استفاده از کارت شناسایی، رمز عبور یا اثر انگشت مورد استقبال قرار گرفته‌اند. فناوری تشخیص چهره به دلیل دقت بالا و سهولت در استفاده، در حوزه‌هایی مانند امنیت و نظارت، دسترسی کنترل شده، احراز هویت در سامانه‌های مالی و بانکی، و بهینه‌سازی فرآیندهای سازمانی، به ویژه حضور و غیاب، کاربرد گسترده‌ای پیدا کرده است.

## ۲-۱- مروری بر مفاهیم پایه تشخیص چهره

تشخیص چهره (Face Recognition) یکی از حوزه‌های مهم در بینایی ماشین و یادگیری ماشین است که هدف آن شناسایی و تأیید هویت افراد بر اساس ویژگی‌های صورت آن‌ها است. این فناوری با استفاده از تصاویر دیجیتال یا ویدیوها، قادر به شناسایی چهره یک فرد در میان مجموعه‌ای از تصاویر است. تشخیص چهره از این نظر مهم است که برخلاف سایر روش‌های شناسایی، مانند استفاده از کارت‌های شناسایی یا رمز عبور، نیازی به تعامل مستقیم فرد ندارد و به شکل غیرمستقیم هویت فرد را از روی تصویر شناسایی می‌کند.

تشخیص چهره به عنوان یک فرآیند پیچیده به سه مرحله اصلی تقسیم می‌شود:

- شناسایی چهره (Face Detection):

اولین مرحله در تشخیص چهره، شناسایی ناحیه چهره در تصویر یا ویدیو است. این فرآیند به این معناست که سیستم باید بتواند یک ناحیه مشخص از تصویر را که حاوی چهره است، شناسایی کند. در این مرحله از الگوریتم‌های خاصی برای تفکیک نواحی چهره از سایر بخش‌های تصویر استفاده می‌شود. برخی از الگوریتم‌های معروف در این زمینه عبارتند از:

روش‌های مبتنی بر آشکارساز هار (Haar Cascade Classifiers): یک روش قدیمی و کلاسیک که از ویژگی‌های هار برای تشخیص چهره استفاده می‌کند. این روش سریع است، اما در شرایط نوری نامناسب یا برای زوایای مختلف چهره دقت آن کاهش می‌یابد.

شبکه‌های عصبی عمیق (Deep Neural Networks - DNN): این روش‌ها در سال‌های اخیر با استفاده از یادگیری عمیق توسعه یافته‌اند و دقت بسیار بالایی در شناسایی چهره در شرایط پیچیده‌تر دارند.

- استخراج ویژگی‌ها (Feature Extraction):

پس از شناسایی ناحیه چهره، مرحله بعدی استخراج ویژگی‌های کلیدی از آن است. هدف این مرحله شناسایی ویژگی‌های منحصر به فرد هر چهره است که امکان تمایز چهره‌ها از یکدیگر را فراهم می‌کند. این ویژگی‌ها شامل مواردی مانند فاصله بین چشم‌ها، شکل بینی، دهان، زاویه چانه، و سایر ویژگی‌های هندسی چهره است. تکنیک‌های مختلفی برای استخراج این ویژگی‌ها وجود دارد:

تحلیل مؤلفه‌های اصلی (Principal Component Analysis - PCA): یک روش آماری است که ویژگی‌های کلیدی چهره را با کاهش ابعاد داده و تمرکز بر مهم‌ترین متغیرها استخراج می‌کند.

تحلیل تفکیک خطی (Linear Discriminant Analysis - LDA): این روش نیز برای استخراج ویژگی‌ها استفاده می‌شود و بر تفکیک چهره‌ها بر اساس تفاوت‌های آماری تمرکز دارد.

امروزه بیشتر از روش‌های یادگیری عمیق مانند شبکه‌های عصبی پیچشی (CNN) برای استخراج خودکار و بهینه ویژگی‌ها استفاده می‌شود. در این روش‌ها، مدل به طور خودکار الگوهای پیچیده و خاص چهره را از تصویر استخراج می‌کند، بدون اینکه نیازی به مداخله انسانی برای تعریف ویژگی‌های مشخص باشد.

- تطبیق و شناسایی (Matching and Recognition):

مرحله نهایی در فرآیند تشخیص چهره، تطبیق ویژگی‌های استخراج‌شده از چهره جدید با یک پایگاه داده از چهره‌های شناخته شده است. در این مرحله، سیستم باید با استفاده از یک متریک یا الگوریتم تطبیق، فاصله یا شباهت بین ویژگی‌های چهره جدید و چهره‌های موجود در پایگاه داده را محاسبه کند. الگوریتم‌های مختلفی برای این کار وجود دارد:

روش‌های مبتنی بر فاصله اقلیدسی (Euclidean Distance): این روش‌ها فاصله بین بردارهای ویژگی چهره‌ها را محاسبه کرده و نزدیک‌ترین چهره را به عنوان تطبیق می‌شناسند.

شبکه‌های عصبی سیامی (Siamese Networks): در این روش‌ها، از شبکه‌های عصبی خاصی استفاده می‌شود که دو ورودی تصویر چهره را گرفته و یاد می‌گیرند تا فاصله ویژگی‌های آن‌ها را به گونه‌ای محاسبه کنند که چهره‌های مشابه به هم نزدیک‌تر و چهره‌های متفاوت دورتر باشند.

تشخیص چهره با چالش‌های متعددی روبروست که می‌تواند عملکرد مدل‌های تشخیصی را تحت تأثیر قرار دهد. برخی از این چالش‌ها عبارتند از:

- **تغییرات زاویه و نورپردازی:** زوایای مختلف صورت و نورهای متنوع می‌توانند تأثیر زیادی بر دقت تشخیص چهره داشته باشند.

- **تغییرات احساسی چهره:** تغییرات احساسی مانند خنده، غم، یا عصبانیت نیز می‌توانند ویژگی‌های چهره را به شکلی تغییر دهند که مدل به اشتباه بیفتد.

- **چهره‌های مشابه:** تشخیص چهره‌های بسیار شبیه به هم یکی از چالش‌های اصلی در این فناوری است.

- **انسدادهای جزئی چهره:** گاهی اوقات چهره افراد به دلیل استفاده از عینک، ماسک یا کلاه به طور کامل دیده نمی‌شود، که تشخیص را دشوار می‌کند.

فناوری تشخیص چهره در حوزه‌های متعددی کاربرد دارد، از جمله:



- **امنیت:** استفاده در سیستم‌های نظارت و امنیت برای شناسایی افراد مشکوک یا احراز هویت در دسترسی به مکان‌های مهم.

- **پرداخت الکترونیکی:** استفاده در سیستم‌های پرداخت با شناسایی چهره به عنوان یک روش احراز هویت.

- **سیستم‌های حضور و غیاب:** استفاده در محیط‌های آموزشی یا سازمان‌ها برای تسریع و بهینه‌سازی فرآیند حضور و غیاب.

تشخیص چهره به عنوان یکی از تکنولوژی‌های پیشرفته در بینایی ماشین و هوش مصنوعی با بهره‌گیری از الگوریتم‌های مختلف و تکنیک‌های یادگیری ماشین توانسته است به سطح بالایی از دقت و کارایی دست یابد. این فناوری با وجود چالش‌هایی که در تشخیص دقیق چهره‌ها در شرایط مختلف دارد، به طور گسترده در صنایع مختلف مورد استفاده قرار گرفته و به تدریج نقش پررنگ‌تری در زندگی روزمره انسان‌ها ایفا می‌کند.

## ۲-۲- الگوریتم‌های تشخیص چهره

تشخیص چهره یکی از پیچیده‌ترین و کاربردی‌ترین شاخه‌های بینایی ماشین و هوش مصنوعی است که از الگوریتم‌های مختلف برای شناسایی و تطبیق چهره‌ها در تصاویر و ویدیوها استفاده می‌کند. این الگوریتم‌ها تلاش می‌کنند ویژگی‌های منحصر به فرد چهره‌های مختلف را استخراج کرده و برای شناسایی و تأیید هویت به کار ببرند. در این بخش، به معرفی و بررسی مهم‌ترین الگوریتم‌های تشخیص چهره می‌پردازیم که به طور گسترده در حوزه‌های مختلف به کار گرفته می‌شوند.

## ۱. الگوریتم تحلیل مؤلفه‌های اصلی (PCA: Principal Component Analysis)

تحلیل مؤلفه‌های اصلی (PCA) یکی از اولین و پرکاربردترین الگوریتم‌ها در حوزه تشخیص چهره است. این الگوریتم از تکنیک‌های کاهش ابعاد استفاده می‌کند تا ویژگی‌های اصلی چهره را استخراج و الگوهای به نام «چهره‌های اصلی» (Eigenfaces) را ایجاد کند. PCA با کاهش ابعاد داده‌ها، به دنبال پیدا کردن ترکیبی از ویژگی‌هایی است که بیشترین واریانس بین داده‌ها را داشته باشد، که این ویژگی‌ها برای شناسایی و تفکیک چهره‌ها استفاده می‌شوند.

مزایا: سرعت و کارایی بالا در شناسایی چهره‌ها در شرایط کنترل‌شده.

معایب: حساسیت به تغییرات نور و زاویه تصویر.

## ۲. الگوریتم تحلیل تفکیک خطی (LDA: Linear Discriminant Analysis)

تحلیل تفکیک خطی (LDA) یک روش یادگیری ماشین نظارت‌شده است که تلاش می‌کند ویژگی‌هایی را استخراج کند که به بهترین نحو بتوانند چهره‌ها را از یکدیگر تفکیک کنند. LDA بر خلاف PCA که صرفاً به دنبال حداکثر کردن واریانس است، تلاش می‌کند تا ویژگی‌هایی را پیدا کند که کلاس‌های مختلف (چهره‌های متفاوت) را با بیشترین دقت از هم جدا کند.

مزایا: دقت بالاتر نسبت به PCA در شرایطی که داده‌ها برچسب‌گذاری شده‌اند.

معایب: نیاز به داده‌های آموزشی با برچسب و حساسیت به نویز در داده‌ها.

## ۳. الگوریتم‌های شبکه‌های عصبی پیچشی (CNN: Convolutional Neural Networks)

شبکه‌های عصبی پیچشی (CNN) در سال‌های اخیر به یکی از موفق‌ترین روش‌ها برای تشخیص چهره تبدیل شده‌اند. این شبکه‌ها از لایه‌های پیچشی برای استخراج ویژگی‌های چهره در سطوح مختلف استفاده می‌کنند. در هر لایه، ویژگی‌های پیچیده‌تری از تصویر استخراج می‌شود که در نهایت به شناسایی چهره منجر می‌گردد. CNN‌ها به دلیل توانایی یادگیری ویژگی‌های مختلف تصویر به‌طور خودکار، در شرایط پیچیده‌ای مانند تغییرات زاویه، نور، و بیان چهره بسیار دقیق عمل می‌کنند.

مزایا: دقت بسیار بالا در شرایط متغیر، مقاومت به تغییرات نور و زاویه.  
معایب: نیاز به داده‌های بزرگ و منابع محاسباتی سنگین.

#### ۴. الگوریتم شبکه‌های عصبی سیامی (Siamese Networks)

شبکه‌های عصبی سیامی برای مسائل شناسایی شباهت‌ها بین دو تصویر طراحی شده‌اند و در تشخیص چهره نیز کاربرد دارند. این شبکه‌ها به‌ویژه برای تأیید هویت و تطبیق چهره‌ها در شرایطی که داده‌های کمی در دسترس است، مناسب هستند. شبکه‌های سیامی با مقایسه دو تصویر ورودی، فاصله یا شباهت بین ویژگی‌های آن‌ها را محاسبه کرده و نتیجه می‌گیرند که آیا دو چهره یکسان هستند یا خیر.  
مزایا: مناسب برای تطبیق چهره‌ها و تأیید هویت.  
معایب: نیاز به معماری‌های پیچیده‌تر و تنظیم دقیق.

#### ۵. الگوریتم‌های مبتنی بر ویژگی‌های محلی (LBP: Local Binary Patterns)

الگوریتم LBP یکی از روش‌های ساده و سریع برای تشخیص چهره است که بر اساس تحلیل بافت چهره کار می‌کند. در این روش، چهره به چندین منطقه کوچک تقسیم می‌شود و در هر منطقه، الگوی باینری محلی برای هر پیکسل محاسبه می‌شود. سپس این الگوها با هم ترکیب شده و یک بردار ویژگی نهایی تشکیل می‌شود که می‌توان از آن برای تطبیق و شناسایی چهره‌ها استفاده کرد.  
مزایا: سرعت بالا و پیاده‌سازی ساده.  
معایب: حساسیت به تغییرات نور و زاویه.

#### ۶. الگوریتم فیلترهای گابور (Gabor Filters)

فیلترهای گابور از الگوریتم‌های دیگری هستند که برای استخراج ویژگی‌های محلی چهره استفاده می‌شوند. این فیلترها به‌خصوص در شناسایی الگوهای پیچیده مانند لبه‌ها و بافت‌های چهره بسیار کارآمد

هستند و می‌توانند تغییرات نور، زاویه، و حالت‌های چهره را بهتر مدیریت کنند. با استفاده از فیلترهای گابور، ویژگی‌های سطح پایین چهره مانند جهت‌گیری لب‌ها و شدت روشنایی استخراج می‌شود و سپس برای تطبیق استفاده می‌شود.

مزایا: توانایی استخراج ویژگی‌های پیچیده محلی.

معایب: نیاز به محاسبات سنگین‌تر نسبت به روش‌های ساده‌تر.

## ۷. الگوریتم‌های یادگیری عمیق (Deep Learning Algorithms)

الگوریتم‌های یادگیری عمیق به‌ویژه شبکه‌های عصبی چندلایه و شبکه‌های بازگشتی (RNN) نقش کلیدی در پیشرفت تشخیص چهره داشته‌اند. این الگوریتم‌ها با استفاده از معماری‌های عمیق و داده‌های بزرگ، می‌توانند به‌طور خودکار ویژگی‌های مختلف چهره را یاد بگیرند و دقت بسیار بالایی در شناسایی چهره‌ها حتی در شرایط پیچیده مانند تغییرات نور، زاویه، و بیان چهره داشته باشند.

مزایا: دقت بسیار بالا و تطبیق‌پذیری با شرایط مختلف.

معایب: نیاز به داده‌های بزرگ و منابع محاسباتی زیاد.

الگوریتم‌های مختلف تشخیص چهره هر کدام مزایا و محدودیت‌های خاص خود را دارند و انتخاب الگوریتم مناسب به شرایط خاص پروژه، نوع داده‌ها، و نیازهای پردازشی بستگی دارد. الگوریتم‌های ساده‌تر مانند PCA و LDA در شرایط کنترل‌شده و زمانی که داده‌های آموزشی محدود هستند، عملکرد خوبی دارند. از سوی دیگر، الگوریتم‌های پیچیده‌تر مانند CNN ها و شبکه‌های عصبی سیامی به دلیل قدرت یادگیری ویژگی‌های پیچیده چهره و تطبیق‌پذیری بالا با شرایط پیچیده، دقت بیشتری در تشخیص چهره دارند و در پروژه‌های بزرگ و پیچیده بیشتر مورد استفاده قرار می‌گیرند.

من در پروژه خود برای مقایسه بین دو تصویر چهره و دریافت شباهت آن‌ها از مدل Siamese استفاده کردم، زیرا این مدل به‌طور خاص برای مسائل شناسایی شباهت و تفاوت بین دو نمونه طراحی

شده است. در فرآیند حضور و غیاب خودکار، نیاز است که چهره فرد با چهره‌های موجود در دیتابیس مقایسه شده و تعیین شود آیا این دو چهره متعلق به یک شخص هستند یا خیر. مدل Siamese با ساختاری که از دو شبکه عصبی مشابه تشکیل شده، قادر است ویژگی‌های منحصربه‌فرد هر چهره را استخراج کرده و سپس فاصله یا تفاوت بین این ویژگی‌ها را محاسبه کند.

یکی از مهم‌ترین مزایای استفاده از این مدل، توانایی آن در کار کردن با تعداد کمی داده آموزشی است، زیرا به جای نیاز به یادگیری ویژگی‌های چندین کلاس (چهره‌های مختلف)، تنها به یادگیری ویژگی‌های فردی برای تمایز بین دو چهره پرداخته می‌شود. همچنین، مدل‌های Siamese بسیار کارآمد هستند و می‌توانند با دقت بالایی شباهت بین چهره‌ها را حتی در شرایطی که تصاویر دچار تغییراتی در نور، زاویه، یا بیان چهره شده‌اند، شناسایی کنند. به همین دلیل، این مدل برای مسئله تشخیص هویت در پروژه من انتخاب شد تا بتواند به‌طور دقیق و بهینه فرایند مقایسه و شناسایی چهره‌ها را انجام دهد.

## ۲-۳- کتابخانه torch

Torch یکی از قدرتمندترین و پرکاربردترین کتابخانه‌های متن‌باز برای یادگیری ماشین و شبکه‌های عصبی است که به‌طور گسترده در پروژه‌های هوش مصنوعی و به‌ویژه در پردازش داده‌های تصویری، مانند تشخیص چهره، استفاده می‌شود. این کتابخانه در ابتدا به زبان Lua توسعه داده شد، اما نسخه محبوب‌تر آن به نام PyTorch به زبان Python ایجاد شده است و توسط شرکت فیسبوک (Meta) پشتیبانی می‌شود. PyTorch به دلیل انعطاف‌پذیری بالا، سهولت در استفاده، و سرعت اجرای بسیار بالا در بین محققان و توسعه‌دهندگان یادگیری ماشین جایگاه ویژه‌ای دارد.

## ۲-۳-۱ - ویژگی‌ها و مزایای کتابخانه Torch

- پشتیبانی از شبکه‌های عصبی عمیق (Deep Learning):

Torch امکان پیاده‌سازی و آموزش مدل‌های پیچیده شبکه عصبی را فراهم می‌کند، از جمله مدل‌های CNN، RNN، و Siamese Networks که برای تشخیص چهره و مقایسه بین تصاویر استفاده می‌شوند. این کتابخانه از محاسبات موازی بر روی واحدهای پردازش گرافیکی (GPU) پشتیبانی می‌کند که باعث سرعت بالای پردازش می‌شود.

- پویایی در ساختار گراف محاسباتی:

برخلاف برخی از کتابخانه‌های دیگر، PyTorch از گراف‌های پویا برای ساختاردهی محاسبات استفاده می‌کند. این بدان معناست که در حین آموزش مدل، ساختار گراف به‌صورت لحظه‌ای (دینامیک) ساخته می‌شود که به انعطاف بیشتری در توسعه و تست مدل‌های پیچیده کمک می‌کند.

- سهولت در یادگیری و استفاده:

یکی از مهم‌ترین مزایای PyTorch نسبت به سایر کتابخانه‌ها، ساده‌سازی فرآیندهای پیچیده یادگیری عمیق است. این کتابخانه با ساختاری شفاف و کاربرپسند، برای توسعه‌دهندگان این امکان را فراهم می‌کند تا مدل‌های خود را به راحتی تعریف، آموزش و ارزیابی کنند.

- پشتیبانی گسترده از عملیات تنسوری:

کتابخانه Torch به‌طور گسترده از عملیات‌های ریاضیاتی و محاسباتی بر روی تنسورها پشتیبانی می‌کند. این تنسورها، مشابه آرایه‌های چندبعدی، برای ذخیره‌سازی داده‌ها و پردازش موازی روی GPU به کار می‌روند. این قابلیت به‌ویژه برای پردازش تصاویر و داده‌های بزرگ مانند دیتاست‌های تشخیص چهره بسیار کاربردی است.

### • کتابخانه‌های پیش‌ساخته و مدل‌های آماده:

PyTorch شامل بسیاری از توابع پیش‌ساخته و مدل‌های از پیش آموزش‌داده‌شده است که باعث می‌شود توسعه‌دهندگان بتوانند به سرعت پروژه‌های خود را آغاز کرده و مدل‌های موجود را برای پروژه‌های خود سفارشی کنند. این ویژگی برای کار با داده‌های تصویری و مدل‌های تشخیص چهره بسیار ارزشمند است.

### ۲-۳-۲- کاربرد Torch در پروژه‌های تشخیص چهره

کتابخانه Torch به دلیل توانمندی‌های بالای خود در یادگیری عمیق، به‌ویژه در پروژه‌های تشخیص چهره کاربردهای زیادی دارد. مدل‌های عصبی پیچیده مانند شبکه‌های عصبی پیچشی (CNN) و شبکه‌های عصبی سیامی (Siamese Networks) که در تشخیص و مقایسه چهره‌ها استفاده می‌شوند، به راحتی در Torch قابل پیاده‌سازی هستند. این کتابخانه ابزارهای متنوعی برای پردازش و تحلیل داده‌های تصویری فراهم می‌کند و به دلیل پشتیبانی از موازی‌سازی و اجرای سریع مدل‌ها بر روی GPU، برای پروژه‌های بزرگ و نیازمند پردازش سریع، انتخاب ایده‌آلی است.

کتابخانه Torch به عنوان یکی از برترین ابزارهای یادگیری ماشین و شبکه‌های عصبی، امکان پیاده‌سازی و آموزش مدل‌های تشخیص چهره را با دقت و کارایی بالا فراهم می‌کند. پشتیبانی از محاسبات گرافیکی، ساختار پویا، و ابزارهای کارآمد در حوزه پردازش تصویر، آن را به گزینه‌ای مناسب برای توسعه‌دهندگان و محققانی که به دنبال پیاده‌سازی سیستم‌های پیشرفته هوش مصنوعی هستند، تبدیل کرده است. در پروژه‌های تشخیص چهره، از جمله پروژه حاضر، Torch به عنوان هسته اصلی پیاده‌سازی و آموزش مدل‌های تشخیص چهره استفاده شده و نتایج قابل اعتمادی ارائه می‌دهد.

## ۲-۴- مدل استفاده شده در این پروژه (Siamese)

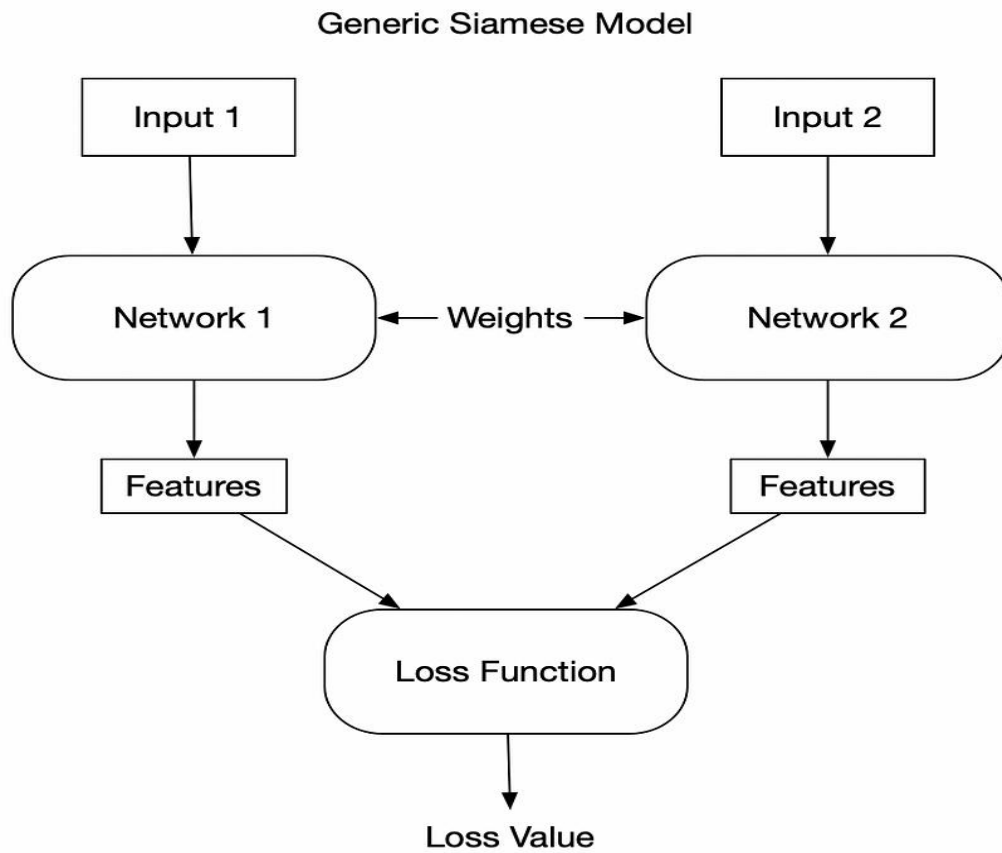
در این پروژه، از مدل Siamese برای مقایسه تصاویر چهره‌ها و تشخیص شباهت یا تفاوت آن‌ها استفاده شده است. مدل سیامی یکی از مدل‌های یادگیری عمیق است که به‌طور خاص برای حل مسائلی که نیاز به مقایسه بین دو داده ورودی دارند، طراحی شده است. در این پروژه، از این مدل برای تشخیص چهره و تعیین حضور دانشجویان بر اساس شباهت چهره آن‌ها به تصاویری که در دیتابیس وجود دارد، استفاده شده است.

### ۲-۴-۱ ساختار مدل Siamese

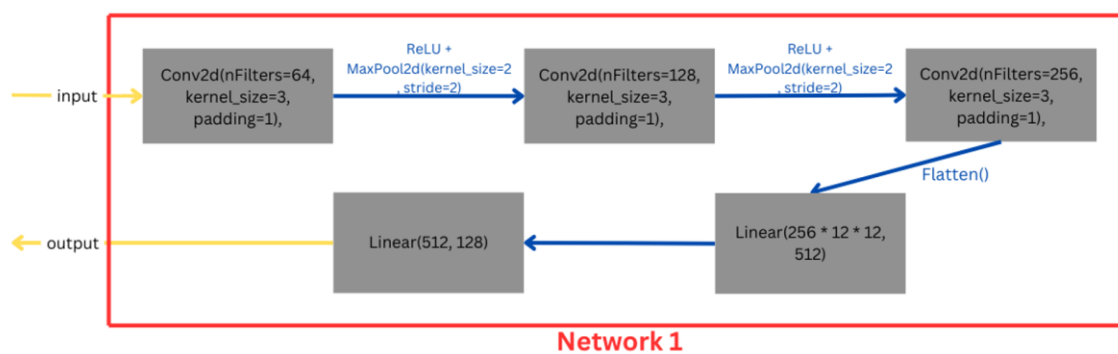
در این پروژه، مدل سیامی از دو شبکه عصبی مشابه تشکیل شده که به‌طور موازی بر روی دو تصویر ورودی کار می‌کنند. ساختار این شبکه‌ها به گونه‌ای طراحی شده که ویژگی‌های منحصربه‌فرد هر تصویر را استخراج کنند. خروجی هر شبکه، بردار ویژگی‌های استخراج شده از تصویر است که نمایانگر ویژگی‌های مهم چهره می‌باشد. سپس این دو بردار با استفاده از یک تابع فاصله (معمولاً فاصله اقلیدسی) مقایسه می‌شوند. هرچه این فاصله کمتر باشد، به معنای شباهت بیشتر بین دو چهره و بالعکس است.

در کد این پروژه، شبکه عصبی پایه‌ای که برای هر کدام از دو شاخه مدل سیامی استفاده شده، شامل سه لایه کانولوشنی (Conv2d) و لایه‌های فعال‌سازی ReLU است. بعد از این لایه‌ها، از لایه‌های Fully Connected برای استخراج ویژگی‌های نهایی استفاده می‌شود که به برداری با ابعاد کوچک‌تر تبدیل می‌شوند. خروجی نهایی شبکه، یک بردار ۱۲۸ بعدی است که نشان‌دهنده ویژگی‌های فشرده‌شده هر تصویر (Embedding) است.





شکل ۱: معماری یک siamese model



شکل ۲: معماری Network 1

معماری Network 2 نیز مشابه Network 1 است.

## ۲-۴-۲- نحوه عملکرد مدل در پروژه

فرایند کار مدل سیامی به این صورت است که دو تصویر ورودی (تصاویر دانشجویان) به مدل داده می‌شود. این تصاویر ابتدا توسط شبکه عصبی پایه (BaseNetwork یا همان Network 1 و Network 2 در شکل ۱ و شکل ۲) پردازش می‌شوند و ویژگی‌های آن‌ها استخراج می‌شود. سپس بردارهای ویژگی دو تصویر با استفاده از فاصله اقلیدسی مقایسه می‌شوند. اگر فاصله بین دو بردار کم باشد، مدل تصمیم می‌گیرد که دو تصویر متعلق به یک فرد است و در غیر این صورت، تصاویر متفاوت در نظر گرفته می‌شوند.

## ۲-۴-۳- تابع هزینه و آموزش مدل

برای آموزش مدل، از تابع هزینه کانتراستیو (Contrastive Loss) استفاده شده است که مخصوص مدل‌های سیامی طراحی شده است. این تابع هزینه، بر اساس فاصله بین بردارهای ویژگی محاسبه می‌شود و سعی می‌کند مدل را به گونه‌ای آموزش دهد که بردارهای ویژگی تصاویر مشابه به هم نزدیک‌تر و تصاویر غیرمشابه دورتر باشند.

تابع کانتراستیو در کد به این شکل عمل می‌کند که اگر دو تصویر به یک فرد تعلق داشته باشند، فاصله بین بردارهای ویژگی آن‌ها باید کم باشد و اگر به افراد مختلفی تعلق داشته باشند، این فاصله باید بزرگ باشد. این تابع با محاسبه فاصله اقلیدسی بین خروجی‌های دو شبکه و اعمال محدودیتی به نام margin، هزینه کلی را محاسبه می‌کند.

در شکل ۳ توصیف فرمول Contrastive Loss مدل در زبان Python آمده است. در آن euclidean\_distance فاصله اقلیدسی دو خروجی Network 1 و Network 2 است.

```
loss_contrastive = torch.mean((1 - label) * torch.pow(euclidean_distance, 2) +
                               (label) * torch.pow(torch.clamp(self.margin - euclidean_distance, min=0.0), 2))
```

شکل ۳: توصیف تابع loss

مدل سیامی به دلیل توانایی‌اش در مقایسه مستقیم بین دو تصویر و استخراج ویژگی‌های مرتبط، انتخاب مناسبی برای تشخیص چهره در این پروژه بوده است. با استفاده از این مدل، می‌توان به‌طور خودکار و دقیق حضور و غیاب دانشجویان را با مقایسه تصاویر ورودی و دیتابیس مرجع انجام داد. مزیت دیگر این مدل، کارایی بالا در مسائل تطبیق چهره حتی با داده‌های آموزشی کمتر است، چراکه به جای یادگیری تمامی ویژگی‌ها، تنها به مقایسه شباهت یا تفاوت بین دو تصویر می‌پردازد.

## ۲-۵- پیش‌پردازش داده‌ها

### ۲-۵-۱- دیتاست lfw

دیتاست LFW (Labeled Faces in the Wild) یکی از معروف‌ترین و پرکاربردترین مجموعه داده‌ها برای تشخیص و شناسایی چهره در حوزه بینایی ماشین و یادگیری ماشین است. این دیتاست به‌طور ویژه برای ارزیابی سیستم‌های تشخیص هویت از طریق چهره طراحی شده است و شامل تصاویر واقعی و متنوع از چهره افراد است که در شرایط نوری و پس‌زمینه‌های مختلف گرفته شده‌اند.

LFW شامل بیش از ۱۳,۰۰۰ تصویر از ۵۷۴۹ فرد مختلف است که از اینترنت جمع‌آوری شده‌اند. تصاویر این مجموعه داده به صورت طبیعی گرفته شده‌اند و تنوع زیادی از نظر نور، زاویه، اندازه و حالات مختلف چهره دارند. همین تنوع، این دیتاست را به یکی از چالش‌برانگیزترین مجموعه داده‌ها برای مدل‌های تشخیص چهره تبدیل کرده است، زیرا تصاویر در شرایط واقعی و بدون تنظیمات کنترل‌شده گرفته شده‌اند.

ویژگی مهم این دیتاست این است که به صورت جفت تصاویر برای مقایسه چهره‌ها و ارزیابی مدل‌ها تنظیم شده است. در آزمایش‌های تشخیص چهره، از این جفت تصاویر برای تعیین میزان شباهت بین دو چهره استفاده می‌شود؛ به طوری که یک برچسب (۱ یا ۰) مشخص می‌کند که آیا دو تصویر متعلق به یک فرد هستند یا خیر. به دلیل تنوع بالای چهره‌ها و تغییرات مختلف، دیتاست LFW برای ارزیابی مدل‌های مبتنی بر شبکه‌های عصبی و به ویژه مدل‌های سیامی (Siamese Networks) بسیار مناسب است.

LFW به عنوان یک معیار استاندارد در تحقیقات تشخیص چهره استفاده می‌شود و مدل‌های مختلفی که برای این حوزه توسعه داده شده‌اند، بر اساس عملکردشان در این دیتاست مقایسه و ارزیابی می‌شوند. این مجموعه داده به توسعه روش‌های نوین در تشخیص هویت چهره کمک کرده است و همچنان یکی از مهم‌ترین ابزارها در این زمینه محسوب می‌شود.

## ۲-۵-۲ - مراحل پیش‌پردازش

پیش‌پردازش داده‌ها یکی از مراحل حیاتی در هر پروژه یادگیری ماشین است که به بهبود عملکرد مدل و افزایش دقت پیش‌بینی‌ها کمک می‌کند. در پروژه حاضر که از مدل سیامی برای تشخیص چهره استفاده شده است، تصاویر ورودی ابتدا تحت فرآیند پیش‌پردازش قرار می‌گیرند تا به فرمتی مناسب برای شبکه عصبی تبدیل شوند. این فرآیند شامل مراحل مختلفی از جمله بارگذاری و تغییر اندازه تصاویر، نرمال‌سازی داده‌ها، و تنظیم داده‌ها در قالب مناسب برای مدل است.

یکی از اولین گام‌ها در این پیش‌پردازش، بارگذاری تصاویر است. تصاویر از دیتاست مربوطه استخراج شده و سپس به ابعاد استاندارد تغییر اندازه داده می‌شوند. در این پروژه، تمامی تصاویر به

اندازه ۱۰۰ پیکسل در ۱۰۰ پیکسل تغییر داده شده‌اند. این کار باعث می‌شود تمامی داده‌های ورودی همگن باشند و مدل بتواند به درستی آن‌ها را پردازش کند.

پس از تغییر اندازه، مرحله نرمال‌سازی بر روی تصاویر اعمال می‌شود. در این مرحله، مقادیر پیکسل‌های تصاویر که معمولاً بین ۰ تا ۲۵۵ است، به مقادیر بین ۰ تا ۱ نرمال‌سازی می‌شوند. این نرمال‌سازی باعث می‌شود که داده‌ها با دامنه کوچکتری به مدل وارد شوند و به این ترتیب، فرآیند آموزش مدل بهبود پیدا می‌کند و سرعت و دقت شبکه عصبی افزایش می‌یابد.

گام بعدی، تبدیل تصاویر به فرمتی مناسب برای شبکه‌های عصبی است. برای این منظور، تصاویر به صورت داده‌های چندبعدی (کانال، عرض، ارتفاع) تغییر داده می‌شوند که فرمت مورد نیاز شبکه‌های عصبی پیچشی (Convolutional Neural Networks) است. این فرمت به شبکه کمک می‌کند تا به درستی اطلاعات تصویری را پردازش و ویژگی‌های مهم هر تصویر را استخراج کند.

علاوه بر این، در مرحله پیش‌پردازش، داده‌ها به دو دسته تقسیم می‌شوند: جفت تصاویر مشابه (دو تصویر از یک فرد) و جفت تصاویر غیرمشابه (تصاویر مربوط به دو فرد مختلف). جفت‌های مشابه به مدل کمک می‌کنند تا شباهت بین تصاویر را تشخیص دهد، در حالی که جفت‌های غیرمشابه به آن اجازه می‌دهند تفاوت بین چهره‌های مختلف را بیاموزد. این داده‌ها در نهایت در قالب دو مجموعه تصویر و یک برچسب برای هر جفت (که نشان می‌دهد تصاویر متعلق به یک فرد هستند یا نه) به مدل وارد می‌شوند.

این فرآیند پیش‌پردازش، با تنظیم دقیق داده‌های ورودی و همگن‌سازی آن‌ها، مدل سیامی را برای یادگیری بهتر آماده می‌کند و به آن کمک می‌کند تا ویژگی‌های چهره‌های مختلف را با دقت بیشتری شناسایی و مقایسه کند.

## ۲-۶-آموزش مدل

برای آموزش مدل سیامی، ابتدا مجموعه‌ای از جفت تصاویر ایجاد می‌شود که شامل دو نوع جفت هستند:

- جفت‌های مشابه که شامل دو تصویر از یک فرد است
- جفت‌های غیرمشابه که از دو فرد مختلف انتخاب شده‌اند

به منظور انجام فرآیند یادگیری، ابتدا تصاویر از دیتاست LFW بارگذاری شده و پس از انجام مراحل پیش‌پردازش، به دو مجموعه تصویر و برچسب‌های مربوطه تقسیم می‌شوند. برچسب‌ها نشان می‌دهند که آیا دو تصویر متعلق به یک فرد هستند (برچسب ۱) یا خیر (برچسب ۰).

در مرحله آموزش، مدل از شبکه پایه یا BaseNetwork برای استخراج ویژگی‌های چهره استفاده می‌کند. این شبکه شامل لایه‌های پیچشی (Convolutional) و لایه‌های کاملاً متصل است که ویژگی‌های چهره را از تصاویر ورودی استخراج و آن‌ها را به یک بردار ویژگی کوچک تبدیل می‌کند. این بردار ویژگی برای مقایسه دو تصویر توسط شبکه سیامی استفاده می‌شود.

- **ورود تصاویر به مدل:** هر جفت تصویر به‌طور همزمان وارد دو شبکه پایه می‌شود که به صورت اشتراکی وزن دارند. این دو شبکه ویژگی‌های هر تصویر را استخراج کرده و به بردارهای عددی تبدیل می‌کنند.

- **محاسبه فاصله:** پس از عبور تصاویر از شبکه، مدل با استفاده از تابع فاصله اقلیدسی (Euclidean distance) شباهت بین دو بردار ویژگی را محاسبه می‌کند.

- استفاده از تابع خطای **Contrastive**: این تابع خطا که با نام Contrastive Loss شناخته می‌شود، تفاوت بین پیش‌بینی مدل و برچسب واقعی را محاسبه می‌کند. در اینجا، هدف مدل این است که برای جفت‌های مشابه، فاصله بین بردارهای ویژگی را کم و برای جفت‌های غیرمشابه، این فاصله را زیاد کند.

- به‌روزرسانی وزن‌ها: در هر مرحله از آموزش، پس از محاسبه خطا، وزن‌های شبکه به کمک الگوریتم بهینه‌سازی Adam به‌روز می‌شوند تا مدل به تدریج بهتر شود.

آموزش مدل به صورت تکراری (epochs) انجام می‌شود و در هر دوره، مدل بر اساس جفت‌های جدید تصاویر بهبود می‌یابد. در هر دوره، مدل به دنبال به حداقل رساندن خطای محاسبه شده برای جفت‌های تصاویر است، به طوری که در پایان، مدل قادر باشد به دقت بالا شباهت یا تفاوت بین چهره‌ها را شناسایی کند.

پس از پایان آموزش، مدل نهایی ذخیره می‌شود تا در محیط اندرویدی به کار رود و فرآیند حضور و غیاب را با تشخیص چهره‌های مختلف به‌طور خودکار انجام دهد. مدل تحت عنوان و فرمت `siamese_model.pth` ذخیره می‌شود.

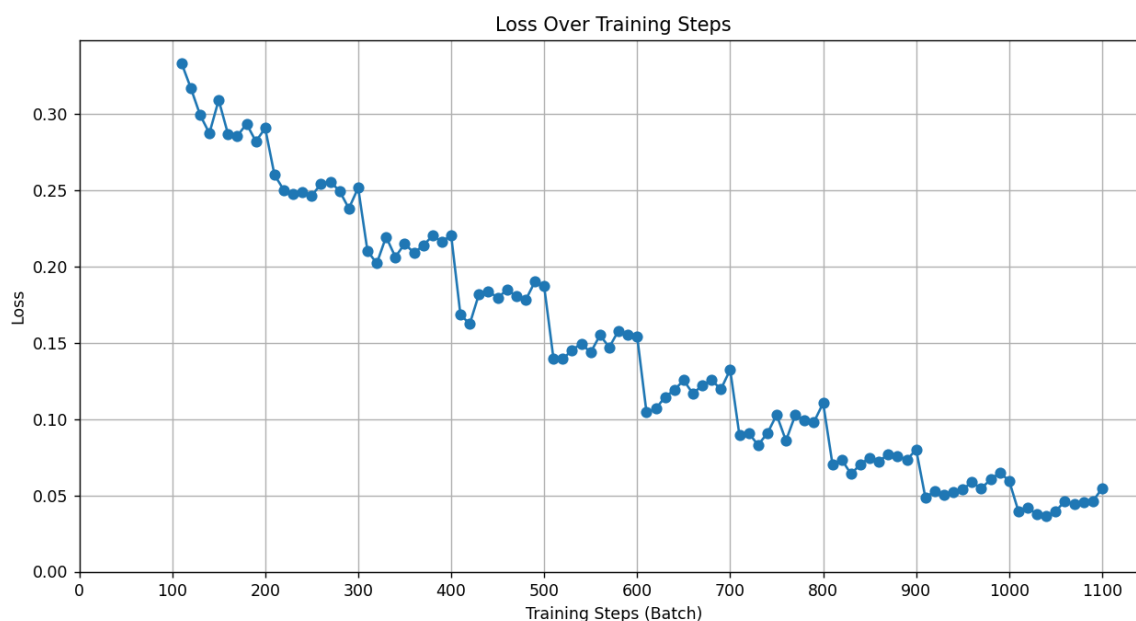
در شکل ۴ می‌تواند Log آموزش چهار epoch اول را مشاهده کنید.

```
[1, 10] loss: 0.3332
[1, 20] loss: 0.3168
[1, 30] loss: 0.2992
[1, 40] loss: 0.2871
[1, 50] loss: 0.3089
[1, 60] loss: 0.2867
[1, 70] loss: 0.2854
[1, 80] loss: 0.2933
[1, 90] loss: 0.2818
[1, 100] loss: 0.2905
[2, 10] loss: 0.2599
[2, 20] loss: 0.2500
[2, 30] loss: 0.2473
[2, 40] loss: 0.2485
[2, 50] loss: 0.2463
[2, 60] loss: 0.2539
[2, 70] loss: 0.2555
[2, 80] loss: 0.2493
[2, 90] loss: 0.2379
[2, 100] loss: 0.2514
[3, 10] loss: 0.2101
[3, 20] loss: 0.2019
[3, 30] loss: 0.2193
[3, 40] loss: 0.2061
[3, 50] loss: 0.2148
[3, 60] loss: 0.2088
[3, 70] loss: 0.2137
[3, 80] loss: 0.2200
[3, 90] loss: 0.2162
[3, 100] loss: 0.2203
[4, 10] loss: 0.1682
[4, 20] loss: 0.1623
[4, 30] loss: 0.1820
[4, 40] loss: 0.1837
[4, 50] loss: 0.1794
[4, 60] loss: 0.1846
[4, 70] loss: 0.1805
[4, 80] loss: 0.1778
[4, 90] loss: 0.1899
[4, 100] loss: 0.1869
```

شکل ۴: خروجی‌های تمرین مدل در چهار epoch اول



در شکل ۵ نیز می‌توانید نمودار تمرین مدل را مشاهده کنید.



شکل ۵: نمودار تمرین مدل

## ۲-۷-۲ ادغام مدل با برنامه اندرویدی

در این پروژه، مدل سیامی تشخیص چهره که با استفاده از کتابخانه Torch در پایتون توسعه داده و آموزش دیده است، با اپلیکیشن اندرویدی ادغام شده تا فرآیند تشخیص چهره‌ها و حضور و غیاب به صورت خودکار انجام شود. برای ارتباط بین اپلیکیشن اندرویدی و کدهای پایتون، از کتابخانه Chaquopy استفاده شده است. Chaquopy امکان اجرای کدهای پایتون در محیط اندرویدی را فراهم می‌کند و به اپلیکیشن اجازه می‌دهد تا مدل آموزش‌دیده را بارگذاری و از آن برای تشخیص چهره‌ها استفاده کند.

در این مرحله، مدل پس از آموزش در محیط پایتون، به اپلیکیشن اندرویدی منتقل شده و به صورت مستقیم از طریق Chaquopy اجرا می‌شود. این ارتباط باعث می‌شود که اپلیکیشن بتواند تصاویر

چهره را از طریق دوربین دستگاه دریافت کرده، آن‌ها را به مدل ارسال کند، و سپس نتیجه شباهت چهره‌ها را به صورت بلادرنگ برای تعیین حضور دانشجویان در کلاس استفاده کند.

جزئیات فنی دقیق‌تر و نحوه پیاده‌سازی کامل این ادغام در فصل چهارم به طور جامع‌تری بررسی خواهد شد.

## فصل سوم

### بررسی برنامه اندرویدی

## بررسی برنامه اندرویدی

این فصل به معرفی و تشریح جنبه‌های کلیدی برنامه اندرویدی توسعه‌یافته برای سیستم حضور و غیاب مبتنی بر تشخیص چهره اختصاص دارد. در عصر حاضر، با پیشرفت فناوری‌های هوش مصنوعی و یادگیری ماشین، نیاز به سیستم‌های هوشمند و خودکار در حوزه‌های مختلف، به‌ویژه در فرآیندهای آموزشی، بیش از پیش احساس می‌شود. سیستم‌های سنتی حضور و غیاب، که به ثبت دستی و یا استفاده از کارت‌های شناسایی متکی هستند، با مشکلاتی از قبیل زمان‌بر بودن، امکان تقلب و خطاهای انسانی مواجه‌اند.

در این فصل، به بررسی معماری و اجزای کلیدی اپلیکیشن پرداخته خواهد شد. برنامه اندرویدی ما، که با استفاده از زبان Kotlin و ابزار Android Studio توسعه داده شده، با هدف تسهیل فرآیند ثبت حضور و غیاب طراحی شده است. این برنامه به اساتید دانشگاه امکان می‌دهد تا تنها با گرفتن یک عکس از هر دانشجو، حضور آنان را به‌صورت خودکار ثبت کنند.

همچنین در این فصل، پیاده‌سازی دوربین و الگوریتم‌های تشخیص چهره، مدیریت داده‌ها و تجربه کاربری اپلیکیشن به‌طور مفصل مورد بررسی قرار خواهند گرفت. این بررسی‌ها به ما کمک می‌کنند تا درک بهتری از چالش‌ها و محدودیت‌های موجود در توسعه نرم‌افزار داشته باشیم و نقاط قوت و ضعف برنامه را شناسایی کنیم. در نهایت، نتایج و ارزیابی‌های انجام‌شده بر روی عملکرد برنامه به تحلیل و نتیجه‌گیری نهایی در این فصل منجر خواهد شد. این فصل، به خواننده دیدی جامع از فرآیند توسعه و پیاده‌سازی اپلیکیشن می‌دهد و اهمیت فناوری‌های نوین در بهینه‌سازی فرآیندهای آموزشی را به تصویر می‌کشد.

## ۳-۱- محیط Android Studio

Android Studio یک محیط توسعه یکپارچه (IDE) است که به طور خاص برای توسعه برنامه‌های اندرویدی طراحی شده است. این محیط به برنامه‌نویسان این امکان را می‌دهد تا با ابزارهای متنوع و امکانات پیشرفته، برنامه‌های کاربردی را به طور کارآمد و سریع ایجاد کنند.

ویژگی‌های محیط Android Studio:

- **رابط کاربری**

رابط کاربری اندروید استودیو به گونه‌ای طراحی شده است که کار با آن آسان و کاربرپسند باشد. این محیط شامل نوار ابزار، پنل پروژه، و ویرایشگر کد است که به توسعه‌دهندگان اجازه می‌دهد به راحتی به منابع مختلف برنامه دسترسی داشته باشند.

- **ویرایشگر کد**

ویرایشگر کد اندروید استودیو از قابلیت‌های متنوعی مانند تکمیل خودکار کد، تشخیص خطا، و قابلیت‌های بازنویسی کد پشتیبانی می‌کند که به تسهیل روند کدنویسی کمک می‌کند.

- **ابزارهای طراحی**

اندروید استودیو شامل ابزارهای طراحی بصری است که به توسعه‌دهندگان اجازه می‌دهد تا واسط کاربری برنامه‌های خود را به صورت گرافیکی طراحی کنند. این ابزارها شامل قابلیت پیش‌نمایش در زمان واقعی و قابلیت طراحی بر اساس اندازه‌های مختلف صفحه نمایش هستند.

- مدیریت منابع

این IDE به توسعه‌دهندگان این امکان را می‌دهد که به راحتی منابع مختلف پروژه مانند تصاویر، فایل‌های صوتی و XML را مدیریت کنند. این امر باعث می‌شود که توسعه‌دهندگان بتوانند به راحتی منابع خود را در پروژه سازماندهی کنند.

- سیستم‌عامل‌های چندگانه

اندروید استودیو به‌طور کامل از سیستم‌عامل‌های مختلف مانند ویندوز، macOS و لینوکس پشتیبانی می‌کند. این ویژگی به توسعه‌دهندگان اجازه می‌دهد تا در هر سیستمی که راحت‌تر هستند، برنامه‌نویسی کنند.

- ابزارهای تست

اندروید استودیو ابزارهای متنوعی برای تست برنامه‌ها در اختیار توسعه‌دهندگان قرار می‌دهد. این ابزارها شامل شبیه‌ساز اندروید و تست‌های خودکار برای بررسی عملکرد برنامه هستند.

- مستندات و منابع آموزشی

اندروید استودیو شامل مستندات جامع و منابع آموزشی است که به توسعه‌دهندگان کمک می‌کند تا با مفاهیم و ابزارهای جدید آشنا شوند و به راحتی مشکلات خود را حل کنند.

این محیط به عنوان یکی از ابزارهای اصلی در توسعه برنامه‌های اندروید شناخته می‌شود و به توسعه‌دهندگان این امکان را می‌دهد که برنامه‌های باکیفیت و کاربرپسند ایجاد کنند.

## ۲-۳- Jetpack Compose تکنولوژی

تکنولوژی Jetpack Compose یک ابزار مدرن و کارآمد برای توسعه واسط کاربری (UI) در برنامه‌های اندروید است که بر پایه کدنویسی کاملاً واکنشی (declarative programming) بنا شده است. این تکنولوژی به توسعه‌دهندگان این امکان را می‌دهد که واسط کاربری خود را به صورت ساده و مستقیم با استفاده از کد Kotlin طراحی کنند. در Jetpack Compose، همه چیز به صورت کامپوننت‌های قابل ترکیب (Composable) تعریف می‌شود که می‌توانند با یکدیگر ترکیب شوند تا واسط کاربری پیچیده‌تری ایجاد کنند.

به طور سنتی طراحی UI در این محیط با استفاده از یک فایل XML انجام می‌شد. در اینجا یک مقایسه با روش سنتی استفاده از فایل XML خواهیم داشت:

### • کدنویسی

#### Jetpack Compose:

با استفاده از کد Kotlin، توسعه‌دهندگان می‌توانند به طور مستقیم واسط کاربری را در کد خود تعریف کنند. این کار باعث می‌شود که کد و UI به طور همزمان و هماهنگ نوشته شوند. به عنوان مثال، یک دکمه می‌تواند به راحتی با استفاده از تابع Button تعریف شود و می‌توان به سادگی به آن رفتار (مثلاً کلیک کردن) اضافه کرد.

#### روش سنتی (XML):

در این روش، طراحی واسط کاربری به صورت جداگانه در فایل‌های XML انجام می‌شود. این موضوع می‌تواند باعث سردرگمی شود زیرا UI و کد مرتبط با آن در دو مکان مختلف قرار دارد. برای تعریف یک دکمه، باید از تگ‌های XML استفاده کرد و رفتار آن باید در کد جاوا یا کاتلین نوشته شود، که می‌تواند پیچیدگی بیشتری ایجاد کند.

- مشاهده تغییرات

### Jetpack Compose

به دلیل ماهیت واکنشی کد، هر بار که داده‌ها تغییر می‌کنند، UI به صورت خودکار به روزرسانی می‌شود. این امر به توسعه‌دهندگان این امکان را می‌دهد که به راحتی با تغییرات در داده‌ها کار کنند.

روش سنتی (XML):

در این روش، به روزرسانی UI به صورت دستی و با فراخوانی متدهایی مانند `notifyDataSetChanged()` یا `invalidate()` انجام می‌شود. این فرآیند می‌تواند پیچیده و زمان‌بر باشد.

- انعطاف پذیری و مقیاس پذیری

### Jetpack Compose

توسعه‌دهندگان می‌توانند کامپوننت‌های قابل استفاده مجدد (reusable) ایجاد کنند و از آن‌ها در بخش‌های مختلف برنامه استفاده کنند. این کار باعث می‌شود که کد تمیزتر و قابل نگهداری‌تر باشد.

روش سنتی (XML):

در این روش، ایجاد کامپوننت‌های قابل استفاده مجدد نیاز به ایجاد View های سفارشی دارد که می‌تواند زمان‌بر باشد و در برخی موارد، پیچیدگی‌هایی ایجاد کند.



## • یادگیری و تسلط

### Jetpack Compose

با توجه به نوآوری در طراحی و شیوه کدنویسی، ممکن است یادگیری آن برای توسعه‌دهندگان جدید دشوار باشد، اما به‌زودی با تسلط بر آن، توسعه‌دهندگان می‌توانند به‌سرعت برنامه‌های باکیفیت بالا ایجاد کنند.

### روش سنتی (XML):

روش سنتی ممکن است برای توسعه‌دهندگان با تجربه در کدنویسی در XML آشنا باشد، اما با گسترش پیچیدگی‌های UI، به مرور زمان نگهداری و توسعه آن دشوارتر می‌شود.

تکنولوژی Jetpack Compose به‌عنوان یک ابزار مدرن و نوآورانه در توسعه واسط کاربری اندروید، قابلیت‌هایی را ارائه می‌دهد که به‌طور قابل توجهی فرآیند توسعه را تسهیل می‌کند و آن را به گزینه‌ای جذاب برای توسعه‌دهندگان تبدیل می‌کند. با این حال، انتخاب بین Jetpack Compose و روش سنتی بستگی به نیازهای پروژه و تجربه تیم توسعه‌دهنده دارد.

## ۳-۳- ابزار ساخت نرم افزار Gradle

Gradle یک ابزار ساخت مدرن و قدرتمند است که به‌ویژه برای پروژه‌های اندروید و JVM طراحی شده است. این ابزار به توسعه‌دهندگان این امکان را می‌دهد که به‌صورت خودکار فرآیندهای ساخت، تست و بسته‌بندی نرم‌افزار را مدیریت کنند. Gradle از زبان Groovy یا Kotlin برای نوشتن اسکریپت‌های ساخت استفاده می‌کند که این امر به توسعه‌دهندگان اجازه می‌دهد تا به‌طور انعطاف‌پذیری ساختار پروژه خود را تعریف و سفارشی کنند.

از ویژگی‌های بارز Gradle می‌توان به موارد زیر اشاره کرد:

- **مدیریت وابستگی‌ها:** Gradle به خوبی با مخازن مختلف (مانند Maven Central و JCenter) کار می‌کند و می‌تواند وابستگی‌های پروژه را به‌طور خودکار دانلود و مدیریت کند.
  - **ساخت چند پروژه‌ای:** Gradle به توسعه‌دهندگان این امکان را می‌دهد که به‌سادگی پروژه‌های چندگانه را در یک ساختار واحد مدیریت کنند.
  - **تسریع در ساخت:** با استفاده از قابلیت‌های کش و تجزیه و تحلیل تغییرات، Gradle می‌تواند زمان ساخت را به‌طور قابل توجهی کاهش دهد.
  - **سفارشی‌سازی:** توسعه‌دهندگان می‌توانند وظایف سفارشی ایجاد کنند و با توجه به نیازهای پروژه، مراحل مختلف ساخت را به راحتی تغییر دهند.
- در نهایت، Gradle به عنوان یک ابزار ساخت پیشرفته، نقش مهمی در تسهیل فرآیند توسعه نرم افزار و بهبود کارایی تیم‌های توسعه دارد.

### ۳-۴- پیکربندی ساخت نرم افزار

در این بخش، پیکربندی Gradle برای ساخت برنامه بررسی می‌شود. این پیکربندی شامل استفاده از پلاگین‌های ضروری، تنظیمات مربوط به Chaquopy برای ادغام با Python، و پیکربندی کلی Android و وابستگی‌ها است.

- پلاگین‌ها

`android.application`: این پلاگین به منظور ساخت اپلیکیشن‌های اندروید استفاده می‌شود.

`kotlin.android`: این پلاگین به منظور پشتیبانی از زبان برنامه‌نویسی Kotlin در پروژه به کار می‌رود.

`com.chaquo.python`: این پلاگین برای ادغام Python با پروژه اندرویدی استفاده می‌شود و امکان استفاده از کتابخانه‌های Python را فراهم می‌کند.

- پیکربندی Chaquopy

`defaultConfig`: در این بخش، نسخه Python مشخص شده و پیکربندی کتابخانه‌ها از طریق `pip` انجام می‌شود. کتابخانه‌های مورد استفاده شامل `numpy`، `opencv-python`، `torch` و `dlib` هستند که در توسعه الگوریتم‌های یادگیری ماشین و پردازش تصویر کاربرد دارند.

- پیکربندی Android

`compileSdk`، `minSdk`، `targetSdk`: این مقادیر به ترتیب نسخه SDK مورد استفاده برای ساخت، حداقل و حداکثر نسخه‌های پشتیبانی شده را مشخص می‌کنند. این نرم افزار با استفاده از SDK 34 ساخته شده است و نسخه اندروید هدف آن اندروید ۱۴ است.

`buildTypes`: در این قسمت، نوع ساخت پروژه (Release) مشخص شده و تنظیمات مربوط به ProGuard برای کاهش اندازه برنامه و بهینه‌سازی کد اعمال می‌شود.

`compileOptions` و `kotlinOptions`: تنظیمات مربوط به نسخه‌های Java و Kotlin مشخص شده‌اند تا اطمینان حاصل شود که کد با نسخه‌های انتخاب شده سازگار است.

- ویژگی‌های ساخت

`buildFeatures`: قابلیت Jetpack Compose فعال شده است که به توسعه‌دهندگان اجازه می‌دهد تا از ابزارهای UI مدرن استفاده کنند.

`composeOptions`: نسخه کامپایلر برای Jetpack Compose تعیین شده است.

- وابستگی‌ها

لیست وابستگی‌ها شامل کتابخانه‌های مختلف برای پشتیبانی از Jetpack Compose، CameraX، و ابزارهای دیگر برای توسعه اپلیکیشن است. به‌طور خاص، CameraX برای استفاده از دوربین دستگاه و Jetpack Compose برای توسعه رابط کاربری مدرن انتخاب شده‌اند.

### ۳-۵- کتابخانه CameraX

کتابخانه CameraX یک چارچوب مدرن و آسان برای استفاده در توسعه اپلیکیشن‌های اندرویدی است که به توسعه‌دهندگان امکان می‌دهد تا قابلیت‌های دوربین را به راحتی و با عملکرد بالا در اپلیکیشن‌های خود ادغام کنند. CameraX به‌خصوص برای تسهیل استفاده از دوربین‌های دستگاه‌های مختلف و کاهش پیچیدگی‌های مرتبط با پیاده‌سازی دوربین در اندروید طراحی شده است.

## • ویژگی‌ها

سادگی استفاده: CameraX با ارائه یک API ساده و قابل فهم، به توسعه‌دهندگان این امکان را می‌دهد که به سرعت و بدون نیاز به درک عمیق از جزئیات فنی دوربین، قابلیت‌های آن را در اپلیکیشن‌های خود پیاده‌سازی کنند.

سازگاری با دستگاه‌های مختلف: یکی از مزایای اصلی CameraX، قابلیت سازگاری آن با طیف وسیعی از دستگاه‌های اندرویدی است. این کتابخانه به‌طور خودکار با ویژگی‌های دوربین دستگاه تطابق پیدا می‌کند و از مشکلات رایج مربوط به سازگاری جلوگیری می‌کند.

عملکرد بالا: CameraX از ویژگی‌های پردازش تصویر پیشرفته‌ای پشتیبانی می‌کند و بهینه‌سازی‌های لازم را برای اطمینان از عملکرد بالای دوربین انجام می‌دهد.

## • معماری

CameraX بر اساس سه مؤلفه اصلی طراحی شده است: Use Cases، Lifecycle و Image Analysis.

Use Cases شامل توابع مختلفی مانند Camera Preview، Image Capture و Image Analysis است که به توسعه‌دهندگان اجازه می‌دهد تا به‌طور خاص به نیازهای خود پاسخ دهند. Lifecycle به CameraX کمک می‌کند تا به‌طور خودکار با چرخه حیات اپلیکیشن هماهنگ شود و از مشکلات مربوط به منابع دوربین جلوگیری کند.

## • مزایا

پشتیبانی از قابلیت‌های جدید: CameraX به‌روزرسانی‌های مکرر را دریافت می‌کند و از ویژگی‌های جدید دوربین مانند قابلیت‌های تصویربرداری چندگانه و ML Kit Vision Integration پشتیبانی می‌کند.

مدیریت آسان منابع: با استفاده از CameraX، توسعه‌دهندگان می‌توانند به‌طور موثری منابع دستگاه را مدیریت کنند و از استفاده نادرست از باتری و پردازنده جلوگیری کنند.

#### • استفاده در پروژه‌ها

CameraX به‌طور گسترده در اپلیکیشن‌هایی مانند اپلیکیشن‌های عکاسی، واقعیت افزوده، و پردازش تصویر استفاده می‌شود. این کتابخانه به توسعه‌دهندگان کمک می‌کند تا تجربه کاربری بهتری را از طریق تعاملات دوربینی آسان و بدون خطا فراهم کنند.

در مجموع، CameraX یک ابزار قدرتمند و منعطف برای توسعه‌دهندگان اندروید است که به آنها امکان می‌دهد تا به راحتی و با اطمینان از قابلیت‌های دوربین در اپلیکیشن‌های خود استفاده کنند. این کتابخانه به عنوان یک راه‌حل جامع، نیازهای مختلف توسعه‌دهندگان را برآورده می‌کند و آنها را در پیاده‌سازی سریع و مؤثر امکانات دوربینی یاری می‌کند.

### ۳-۶- معماری برنامه

معماری این برنامه که به توسعه یک سیستم حضور و غیاب مبتنی بر شناسایی چهره با استفاده از دوربین و کتابخانه‌های مختلف شده، از چندین بخش کلیدی تشکیل شده است. در ادامه، اجزای اصلی معماری این برنامه را بررسی می‌کنیم:

#### • کامپوننت‌ها

MainActivity: این کلاس به عنوان نقطه ورودی برنامه عمل می‌کند. در اینجا، درخواست مجوزهای لازم برای دسترسی به دوربین و ذخیره‌سازی تصاویر انجام می‌شود.

CameraPreviewScreen: در این کامپوننت، نمایش زنده دوربین و دکمه‌ای برای ثبت عکس وجود دارد. این کامپوننت از کتابخانه CameraX برای دسترسی و مدیریت دوربین استفاده می‌کند.

FirstScreen: این صفحه به کاربر امکان می‌دهد نام‌های دانشجویان را وارد کند و لیستی از آن‌ها با قابلیت انتخاب (Checkbox) نمایش دهد. اینجا هم دکمه‌هایی برای اضافه کردن دانشجویان و شروع فرآیند حضور و غیاب وجود دارد.

NavHost: این بخش مسئول مدیریت ناوبری بین صفحات مختلف برنامه است. این ویژگی با استفاده از NavController پیاده‌سازی شده است.

#### • مدیریت وضعیت

SharedPreferences: برای ذخیره و بارگذاری نام‌های دانشجویان از SharedPreferences استفاده شده است. این رویکرد به کاربر امکان می‌دهد تا نام‌های وارد شده را بین راه‌اندازی‌های مختلف برنامه حفظ کند.

#### • منطق شناسایی چهره

در متد captureImage، عکس ثبت شده پردازش می‌شود تا با استفاده از مدل پایتون، شناسایی چهره‌ها انجام شود. نتایج پردازش به لیست دانشجویان اضافه می‌شود و وضعیت آن‌ها در لیست نشان داده می‌شود. عکس دانشجویان در پوشه Students در پوشه Pictures در حافظه داخلی دستگاه ذخیره می‌شود. با اضافه شدن هر دانشجو جدید یک عکس جدید به آن پوشه اضافه می‌شود. ترتیب نامگذاری دانشجویان از 0 است و به ترتیب زیاد می‌شود. مثلاً نام فایل دوم 1.jpg است. در همان Direcotry، یک پوشه به نام Tmp نیز تشکیل می‌شود که مسئولیتش نگه داری عکسی است که برای مقایسه با باقی دانشجویان گرفته شده است. عکس با نام Tmp.jpg در آن پوشه ذخیره

می‌شود. در واقع این همان دانشجویی است که کاربر قصد حضور و غیاب وی را دارد. درون برنامه قطعه کد پایتونی و فایل مدل ذخیره شده قرار دارد (در پوشه python). برنامه با صدا زدن این قطعه کد و پاس دادن آرایه‌ای از آدرس‌های عکس دانشجویان ذخیره شده و عکس دانشجو برای تعیین حضور یا غیاب، آرایه‌ای از اعداد float از قطعه کد پایتون پس می‌گیرد. هر عدد پیشبینی مدل از میزان شباهت عکس دانشجو برای حضور و غیاب با یک از دانشجویان ذخیره شده است. این مقایسه دو به دو و به ترتیب انجام می‌شود. تابعی که این وظیفه را به عهده دارد در شکل ۶ قابل مشاهده است. طبق توضیحات فصل ۲، هر چه عدد برگردانده شده به صفر نزدیک‌تر باشد یعنی آن دو عکس (دانشجو) بیشتر به هم شبیه‌اند. در نتیجه آن شخصی که مقایسه با آن کمترین عدد را برگردانده به عنوان پیشبینی نرم افزار معرفی می‌شود. در نهایت نام وی در لیست دانشجویان تیک می‌خورد.

```
def isSamePerson(ar, img2_path):
    model = SiameseNetwork()
    model.load_state_dict(torch.load(join(dirname(__file__), "weights/siamese_model.pth")))
    model.eval()

    img1s = []
    res = []
    for i in ar:
        img1s.append(torch.tensor([load_image(i)], dtype=torch.float32))
    img2 = torch.tensor([load_image(img2_path)], dtype=torch.float32)

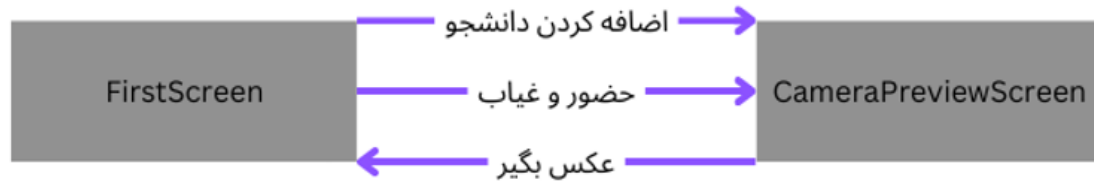
    with torch.no_grad():
        for i in img1s:
            embedding_a, embedding_b = model(i, img2)
            distance = F.pairwise_distance(embedding_a, embedding_b)
            res.append(distance.item())

    return res
```

شکل ۶: تابع برگرداننده میزان شباهت دانشجویان

در شکل ۷ نیز شمایی از نحوه جابجایی (Navigation) کاربر بین صفحه اصلی و صفحه گرفتن عکس نمایش داده شده‌است. فلش‌ها معنی انجام یک action را می‌دهند که در اینجا فشردن دکمه است.





شکل ۷: نحوه Navigation کاربر

در اولین استفاده از نرم افزار توسط کاربر نرم افزار مجوز دسترسی به دوربین و گالری را می گیرد. معماری این برنامه به طور مؤثری از ترکیبی از کتابخانه های مدرن و الگوهای طراحی متداول استفاده کرده و قابلیت های کاربر پسند و کاربردی را فراهم می کند. این برنامه به کاربران امکان می دهد تا به سادگی حضور و غیاب را با استفاده از فناوری شناسایی چهره انجام دهند، و تمام جنبه های برنامه به صورت منظم و قابل مدیریت پیاده سازی شده اند.

## فصل چهارم

### کاربرد و محیط نرم افزار

## کاربرد و محیط نرم افزار

### ۴-۱- کاربرد برنامه

در این پروژه، یک نرم افزار حضور و غیاب مبتنی بر شناسایی چهره توسعه یافته است که می تواند کاربردهای متنوع و موثری در محیط های آموزشی، به ویژه دانشگاه ها، داشته باشد. این نرم افزار به اساتید و مدیران آموزش این امکان را می دهد که به جای استفاده از روش های سنتی و دستی، فرآیند حضور و غیاب دانشجویان را به شکل خودکار و دقیق مدیریت کنند.

برخی از کاربردهای این نرم افزار عبارتند از:

- **بهبود کارایی فرآیند حضور و غیاب:** این نرم افزار با استفاده از فناوری شناسایی چهره، دانشجویان را به طور خودکار از طریق عکس گرفته شده شناسایی می کند و حضور آن ها را ثبت می کند. این قابلیت زمان زیادی را برای اساتید صرفه جویی می کند و از خطاهای انسانی جلوگیری می کند.
- **دقت بالا در شناسایی دانشجویان:** با استفاده از مدل شناسایی چهره که با یادگیری ماشین آموزش داده شده است، این نرم افزار می تواند با دقت بالایی چهره ها را شناسایی کند و به طور خودکار حضور دانشجویان را ثبت نماید.
- **جلوگیری از تقلب:** در روش های سنتی حضور و غیاب، احتمال تقلب وجود دارد (برای مثال، یکی از دانشجویان به جای دیگری پاسخ دهد). این نرم افزار با استفاده از شناسایی چهره، این امکان را از بین می برد و تنها حضور فرد واقعی ثبت می شود.

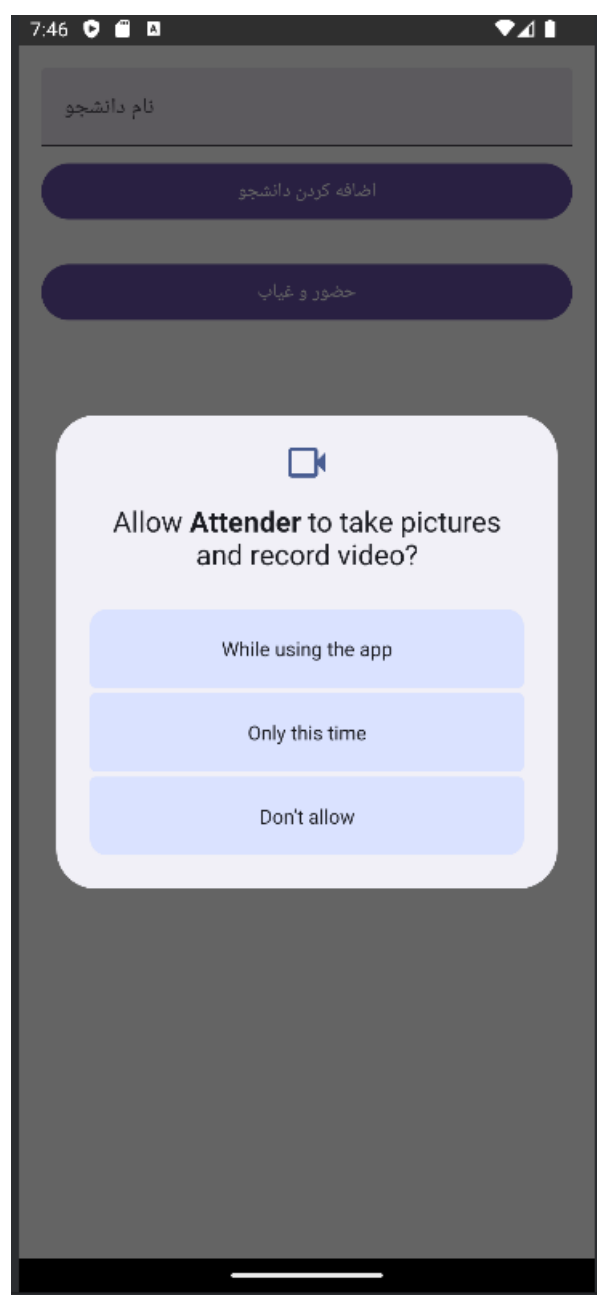
- **ذخیره سازی داده ها و گزارش دهی:** نرم افزار می تواند داده های مربوط به حضور و غیاب دانشجویان را ذخیره کند و در صورت نیاز، گزارش های مربوط به حضور دانشجویان را به اساتید ارائه دهد.

- **کاربرد در محیط های دیگر:** علاوه بر دانشگاه ها، این نرم افزار می تواند در مدارس، مراکز آموزشی و حتی سازمان ها و شرکت ها برای مدیریت حضور و غیاب پرسنل نیز به کار رود.

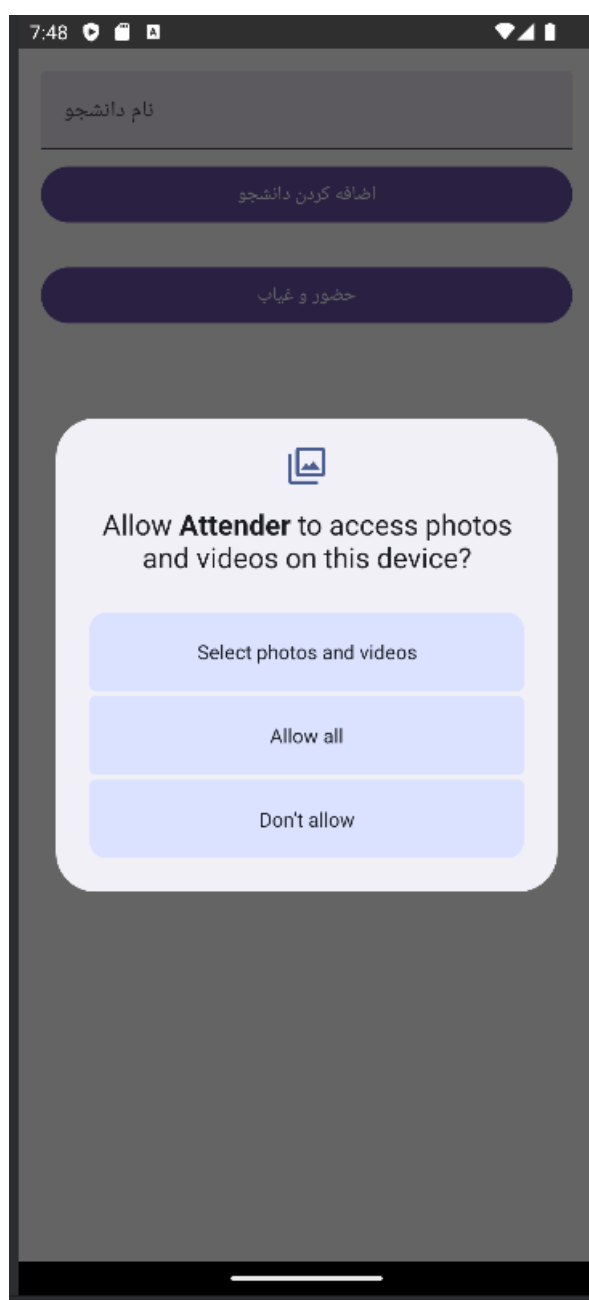
به طور کلی، این نرم افزار می تواند فرآیند حضور و غیاب را به صورت هوشمند و بدون دخالت انسانی مدیریت کند، کارایی را افزایش دهد و دقت بالایی در ثبت اطلاعات ارائه دهد.

## ۴-۲- تصاویری از محیط نرم افزار به همراه استفاده

در ادامه تصاویری از نحوه کارکرد برنامه در شبیه ساز اندروید ۱۴ در محیط Android Studio را می بینیم.



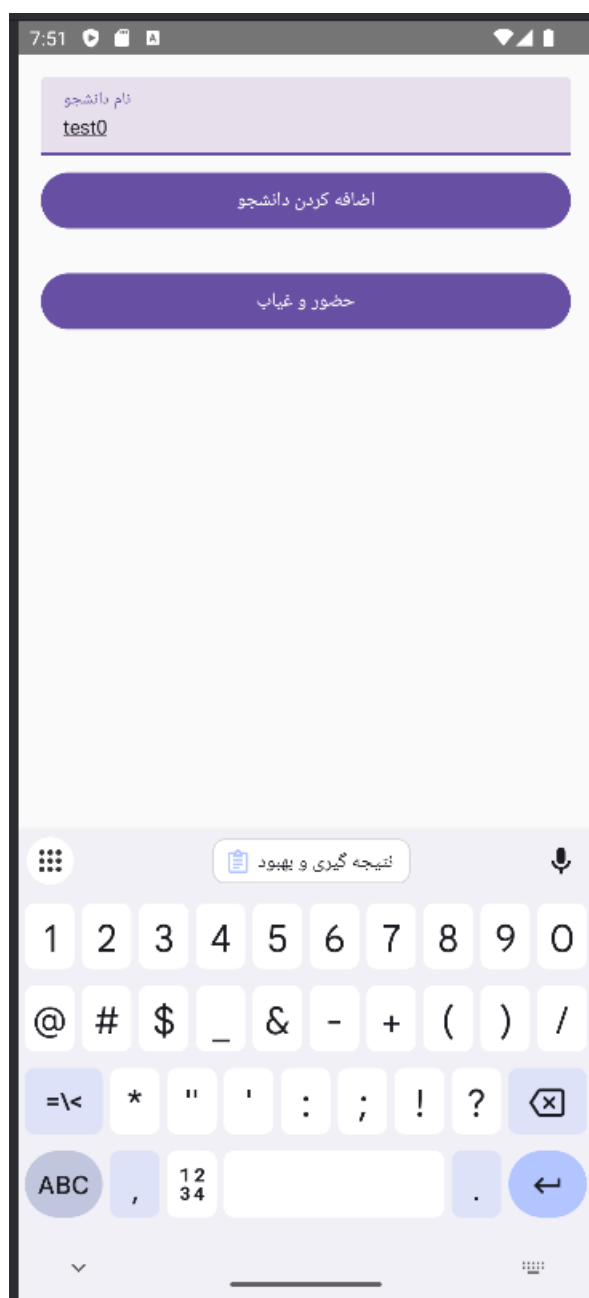
شکل ۸: گرفتن Permission استفاده از دوربین



شکل ۹: گرفتن Permission دسترسی به Media



شکل ۱۰: محیط خانه برنامه



شکل ۱۱: نام دانشجو باید وارد شود





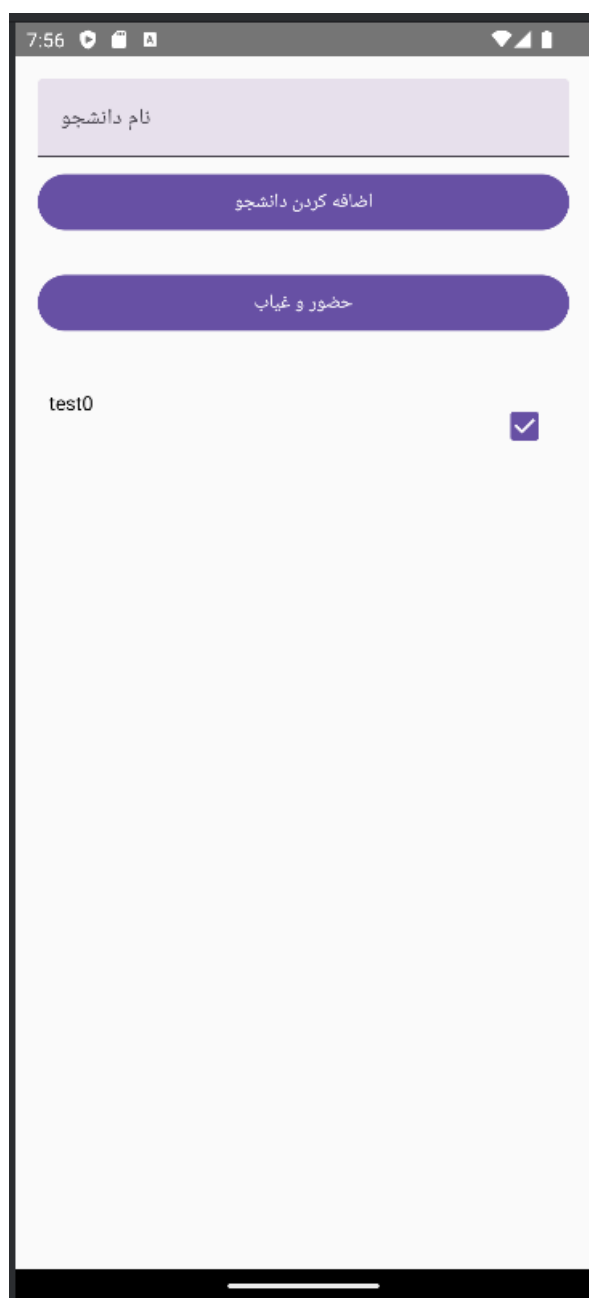
شکل ۱۲: با زدن دکمه اول دوربین باز شده و امکان گرفتن عکس وجود دارد

The screenshot shows a mobile application interface with a status bar at the top displaying the time 7:53 and various icons. The main content area has a light purple header with the text "نام دانشجو" (Student Name). Below this header are two purple buttons: "اضافه کردن دانشجو" (Add Student) and "حضور و غیاب" (Attendance). Under the buttons, there is a list item with the text "test0" and a small square checkbox to its right. The bottom of the screen shows a white home indicator bar.

شکل ۱۳: دانشجو به همراه چک باکس اضافه شده است



شکل ۱۴: با زدن دکمه دوم مجدد دوربین باز شده



شکل ۱۵: با بازگشت به صفحه اصلی، مشاهده می‌شود که نام دانشجو به عنوان حاضر تیک خورده‌است

## فصل پنجم

### جمع‌بندی و نتیجه‌گیری و پیشنهادات

## جمع‌بندی و نتیجه‌گیری و پیشنهادات

در این پروژه، یک سیستم مدیریت حضور و غیاب مبتنی بر شناسایی چهره طراحی و پیاده‌سازی شده است که از فناوری‌های مدرن مانند یادگیری ماشین و هوش مصنوعی استفاده می‌کند. این نرم‌افزار با بهره‌گیری از دوربین گوشی و کتابخانه CameraX تصاویر دانشجویان را ثبت کرده و با استفاده از یک مدل شناسایی چهره که با کتابخانه Torch آموزش داده شده، به طور خودکار چهره دانشجویان را شناسایی و حضور آن‌ها را ثبت می‌کند.

این سیستم با دقت بالا و سرعت مناسبی که در شناسایی چهره‌ها دارد، فرآیند حضور و غیاب را از حالت دستی و زمان‌بر به فرآیندی سریع، دقیق و خودکار تبدیل کرده است. علاوه بر این، ویژگی‌هایی مانند کاهش احتمال تقلب و نگهداری داده‌های حضور و غیاب در حافظه دستگاه، از مزایای مهم این پروژه به شمار می‌آید.

موارد زیر تعدادی از پیشنهادات هستند:

- **بهبود دقت مدل شناسایی چهره:** با استفاده از مجموعه داده‌های گسترده‌تر و متنوع‌تر می‌توان مدل شناسایی چهره را بیشتر آموزش داد و دقت آن را افزایش داد. به‌ویژه در مواقعی که نور محیط یا کیفیت تصویر پایین باشد، این بهبودها می‌توانند اثرگذار باشند.
- **افزافه کردن قابلیت‌های جدید:** امکان افزودن قابلیت‌هایی مانند تشخیص زمان ورود و خروج دانشجویان، یا تولید خودکار گزارش‌های هفتگی و ماهانه برای استادان و مدیران می‌تواند ارزش افزوده بیشتری به نرم‌افزار بدهد.
- **افزایش امنیت:** برای افزایش امنیت و حفاظت از داده‌های کاربران، می‌توان از روش‌های رمزنگاری پیشرفته و مدیریت دقیق‌تر دسترسی به داده‌های حضور و غیاب استفاده کرد تا

اطلاعات به‌صورت امن نگهداری شوند. همچنین دسترسی نرم افزار به گالری می‌تواند محدودتر باشد.

- **پشتیبانی از چند دستگاه:** با اضافه کردن امکاناتی برای همگام‌سازی داده‌ها بین چند دستگاه یا بین سیستم‌های مختلف، این نرم‌افزار می‌تواند به شکل گسترده‌تری در سازمان‌ها و موسسات بزرگ مورد استفاده قرار گیرد.

- **پشتیبانی از چند زبان:** افزودن قابلیت پشتیبانی از زبان‌های مختلف می‌تواند این نرم‌افزار را برای استفاده در مناطق و کشورهای مختلف مناسب‌تر کند و کاربرد آن را در محیط‌های آموزشی متنوع‌تر افزایش دهد.

به‌طور کلی، این پروژه پایه‌ای محکم برای توسعه سیستم‌های هوشمند مدیریت حضور و غیاب فراهم کرده و با بهبودهای پیشنهادی می‌تواند به ابزاری کاربردی‌تر و گسترده‌تر تبدیل شود.

## منابع و مراجع

- [1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. 2003. Face recognition: A literature survey. *ACM Comput. Surv.* 35, 4 (December 2003), 399–458. <https://doi.org/10.1145/954339.954342>
  - [2] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* 521, 436–444 (2015). <https://doi.org/10.1038/nature14539>
  - [3] J. Liu and J. Yu, "Research on Development of Android Applications," 2011 4th International Conference on Intelligent Networks and Intelligent Systems, Kuming, China, 2011, pp. 69-72, doi: 10.1109/ICINIS.2011.40.
  - [4] Dospinescu, Octavian & Popa, Iulian. (2016). Face Detection and Face Recognition in Android Mobile Applications. *Informatica Economica*. 20. 20-28. 10.12948/issn14531305/20.1.2016.02.
  - [5] A. Salihbašić and T. Orehovački, "Development of Android Application for Gender, Age and Face Recognition Using OpenCV," 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 2019, pp. 1635-1640, doi: 10.23919/MIPRO.2019.8756700.
  - [6] J. Hu, L. Peng and L. Zheng, "XFace: A Face Recognition System for Android Mobile Phones," 2015 IEEE 3rd International Conference on Cyber-Physical Systems, Networks, and Applications, Hong Kong, China, 2015, pp. 13-18, doi: 10.1109/CPSNA.2015.12.
- Dwi Sunaryono, Joko Siswantoro, Radityo Anggoro, An android based course attendance system using face recognition, *Journal of King Saud University - Computer and Information Sciences*, Volume 33, Issue 3, 2021, Pages 304-312, ISSN 1319-1578, <https://doi.org/10.1016/j.jksuci.2019.01.006>.



### Abstract

In this project, an Android application was designed and developed using Android Studio and the Kotlin programming language. The purpose of this application is to facilitate and expedite the attendance process for university professors. To achieve this, a facial recognition model was developed using the Torch library and machine learning algorithms in Python. After being trained with the relevant data, the model can recognize students' faces through a photo taken within the application. Upon successful face recognition, the student's name is automatically marked in the attendance list. This system not only increases the speed and accuracy of attendance tracking but also reduces human errors and improves classroom management.

**Key Words:** Android, Artificial Intelligence, Machine Learning, Attendance



**Amirkabir University of Technology  
(Tehran Polytechnic)**

**Department of Computer Engineering**

**BSc Thesis**

**Design and development of an Android application  
for attendance using face recognition and machine  
learning**

**By  
Amir Ali Belbasi**

**Supervisor  
Dr. Alireza Bagheri**

**October 2024**