



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

پایان نامه کارشناسی

طراحی و پیاده سازی سامانه شناسایی تصاویر کارت ملی جهت
استقرار در مرورگر وب

نگارش

علی نوروزبیگی

استاد راهنما

جناب آقای دکتر احمد نیک آبادی

شهریور ۱۴۰۳



به نام خدا

تاریخ: ۱۴۰۳/۶/۲۸

تعهدنامه اصالت اثر

اینجانب علی نوروزبیگی 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.

علی نوروزبیگی

- متن تقدیم

تقدیم به پدر و مادر عزیز و مهربانم، که در سختی‌ها و دشواری‌های زندگی همواره یآوری دلسوز و فداکار و پشتیبانی محکم و مطمئن برایم بوده‌اند.

- متن تقدیر و تشکر

از جناب آقای دکتر احمد نیک‌آبادی برای زحمات بی‌شائبه و دلسوزانه ایشان در پیشبرد این پروژه کمال تشکر و قدردانی را دارم.

علی نوروزیگی

تابستان ۱۴۰۳

چکیده

در سامانه‌های احراز هویت برخط و غیرحضوری، سامانه استخراج اطلاعات از کارت ملی، نقشی اساسی ایفا می‌کند. این سامانه که خود متشکل از چند مرحله است، در گام اول، باید بتواند خود کارت را به درستی مکان‌یابی کند. در غیر این صورت استخراج اطلاعات از کارت در گام‌های بعدی با اشکال جدی مواجه خواهد شد و باعث بروز اشکلاتی نظیر درخواست‌های مکرر از کاربر برای نمایش کارت ملی و تجربه بد کاربری می‌گردد. به همین دلیل، سامانه باید دقت قابل قبولی در مکان‌یابی کارت ملی داشته باشد. وجود معضلاتی نظیر دوران، شدت نور، نحوه در دست گرفتن کارت و... بر دقت این سامانه تاثیرگذار می‌باشد.

مکان‌یابی کارت ملی در تصاویر در سمت سرور بسیار پرهزینه می‌باشد، چرا که در این صورت حجم زیادی داده و تصویر به سمت سرور ارسال می‌شود و عملکرد سرور را در سرویس‌رسانی به کاربران دیگر را با اختلال جدی مواجه خواهد کرد. به همین دلیل، نیاز است که یک پردازش اولیه در سمت کاربر انجام شود تا از ارسال داده‌های بی‌مورد به سمت سرور جلوگیری شود و پردازش‌های آن کاهش یابد. این پیش پردازش شامل تشخیص کارت در سمت کاربر و اصلاح زاویه آن می‌باشد.

در این پژوهش راه‌حلی برای غلبه بر دو چالش مذکور ارائه می‌گردد و سامانه‌ای برای مکان‌یابی و اصلاح زاویه کارت ملی در مرورگر وب طراحی و پیاده‌سازی می‌شود. استفاده از تکنولوژی‌های هوش مصنوعی و یادگیری عمیق، به ویژه شبکه عصبی YOLOv5 به عنوان اصلی‌ترین ابزار، برای تشخیص شناسایی کارت‌های ملی به کار گرفته شده است. این شبکه یکی از مطرح‌ترین شبکه‌های تشخیص دهنده اشیا است و قادر است با دقت بالا و به صورت بی‌درنگ، کارت ملی را تشخیص دهد. فرآیند آموزش، شامل جمع‌آوری داده، برچسب‌زنی، و تنظیم دقیق بر روی دو مدل معمولی و سبک‌تر YOLO، به تفصیل شرح داده شده است. نتایج ارزیابی‌ها نشان می‌دهد که شبکه‌های پیشنهادی دارای دقت بالا در تشخیص کارت ملی می‌باشد و استفاده از مدل سبک‌تر فرق قابل توجهی با استفاده از مدل معمولی آن ندارد. به دلیل رسیدن به دقت بالای مدنظر، به استفاده از شبکه YOLOv5 بسنده شده است.

به منظور استقرار مدل آموزش دیده در سمت کاربر، از زبان جاوا اسکریپت و جهت اجرای مدل آموزش دیده در مرورگر از کتابخانه TensorFlow.js استفاده شده است. واسط کاربری توسعه داده شده به خوبی قادر است کارت را تشخیص داده و زاویه آن را در صورت نیاز اصلاح کند. در نهایت با توجه به وجود عدم تفاوت چشم‌گیر بین دقت دو مدل آموزش داده شده و اختلاف قابل توجه در عملکرد زمان اجرا، استفاده از مدل سبک به عنوان گزینه قابل قبول مطرح شده است.

واژه‌های کلیدی:

تشخیص اشیا، YOLOv5، TensorFlow.js، کارت ملی، یادگیری عمیق

صفحه	فهرست مطالب
أ	چکیده.....
۱	فصل اول مقدمه.....
۸	فصل دوم مرور کارهای مرتبط و شبکه‌های عصبی تشخیص اشیا.....
۹	۱-۲- مرور سامانه‌ها و شبکه‌های مرتبط.....
۱۷	۲-۲- شبکه YOLOv5.....
۱۷	۱-۲-۲ معماری شبکه YOLOv5.....
۱۸	۲-۲-۲ مراحل تشخیص اشیا.....
۱۹	۳-۲-۲ ویژگی‌های YOLOv5 در آموزش.....
۲۰	۴-۲-۲ اطلاعات مدل‌های از پیش آموخته شده YOLOv5.....
۲۰	۳-۲- نتیجه‌گیری.....
۲۱	فصل سوم تشخیص کارت ملی.....
۲۲	۱-۳- مجموعه داده.....
۲۲	۱-۱-۳ جمع آوری داده.....
۲۶	۲-۱-۳ برچسب‌زنی داده‌ها.....
۳۰	۲-۳- آموزش مدل.....
۳۰	۱-۲-۳ یادگیری انتقالی.....
۳۰	۲-۲-۳ مدل‌های از پیش آموخته شده YOLOv5.....
۳۱	۳-۲-۳ فرآیند آموزش.....
۳۳	۳-۳- آزمون.....
۳۳	۱-۳-۳ معیارهای ارزیابی.....
۳۵	۲-۳-۳ نتایج اعمال شبکه‌های آموزش دیده بر روی داده‌های آزمون.....
۴۰	۳-۳-۳ مقایسه و تحلیل.....
۴۳	۴-۳- نتیجه‌گیری.....
۴۴	فصل چهارم واسط کاربری.....
۴۶	۱-۴- صدور مدل‌های آموزش دیده به فرمت مناسب.....
۴۷	۲-۴- توسعه واسط کاربری.....
۴۷	۱-۲-۴ معرفی.....
۴۹	۲-۲-۴ توضیح اجزای واسط کاربری.....
۵۷	۳-۴- نتیجه‌گیری.....
۵۸	فصل پنجم جمع‌بندی، نتیجه‌گیری، و پژوهش‌های آتی.....

۵-۱ - جمع‌بندی و نتیجه‌گیری.....	۵۹
۵-۲ - پیشنهادهای آتی.....	۶۰
منابع و مراجع.....	۶۱

صفحه	فهرست اشکال
۳	شکل ۱-۱: مراحل استخراج اطلاعات از کارت ملی.....
۵	شکل ۱-۲: کارت ملی به همراه نمادهای آن.....
۱۳	شکل ۱-۲: تشخیص اشیا به همراه زاویه.....
۱۵	شکل ۲-۲: تشخیص کارت ملی به همراه نمادها و اصلاح زاویه.....
۱۶	شکل ۳-۲: روند استخراج اطلاعات در مقاله [۷].....
۱۶	شکل ۴-۲: استخراج نوشته‌ها در مقاله [۷] و مرتب‌سازی بر اساس موقعیت.....
۱۹	شکل ۵-۲: تابع Loss آموزش شبکه YOLOv5.....
۲۴	شکل ۱-۳: حضور کارت در زوایا و فاصله‌های مختلف.....
۲۵	شکل ۲-۳: حضور کارت در شرایط نورپردازی متفاوت.....
۲۵	شکل ۳-۳: حالت‌های مختلف نگهداری کارت در دست.....
۲۶	شکل ۴-۳: تصاویر کارت با دوربین‌های مختلف.....
۲۶	شکل ۵-۳: نمونه داده‌های منفی جمع‌آوری شده.....
۲۷	شکل ۶-۳: شمای نرم افزار labeling.....
۲۸	شکل ۷-۳: نمونه‌ای از برچسب یک تصویر با فرمت YOLO.....
۲۸	شکل ۸-۳: فایل classes.txt.....
۲۹	شکل ۹-۳: ساختار پوشه‌بندی مجموعه داده آموزشی.....
۳۴	شکل ۱۰-۳: تعریف امتیاز F1.....
۳۵	شکل ۱۱-۳: تعریف معیار IoU.....
۳۶	شکل ۱۲-۳: نمونه داده‌های آزمون.....
۳۷	شکل ۱۳-۳: نمودار دقت - اطمینان برای مدل yolov5s.....
۳۷	شکل ۱۴-۳: نمودار فراخوانی - اطمینان برای مدل yolov5s.....

شکل ۳-۱۵: نمودار امتیاز F1 - اطمینان برای مدل yolov5s.....	۳۸
شکل ۳-۱۶: نمودار دقت - اطمینان برای مدل yolov5n.....	۳۹
شکل ۳-۱۷: نمودار فراخوانی - اطمینان برای مدل yolov5n.....	۳۹
شکل ۳-۱۸: نمودار امتیاز F1 - اطمینان برای مدل yolov5n.....	۴۰
شکل ۳-۱۹: دسته اول از تشخیص‌های ناموفق.....	۴۱
شکل ۳-۲۰: دسته دوم از تشخیص‌های ناموفق.....	۴۲
شکل ۴-۱: فایل‌های باینری مدل‌های صادر شده.....	۴۷
شکل ۴-۲: نمای آغازین واسط کاربری.....	۴۸
شکل ۴-۳: اتصال سامانه به دوربین کاربر.....	۴۹
شکل ۴-۴: قطعه کد اتصال به دوربین کاربر.....	۵۰
شکل ۴-۵: قطعه کد تابع تشخیص اشیا.....	۵۱
شکل ۴-۶: لیست تودرتو نتایج.....	۵۱
شکل ۴-۷: مکان‌یابی کارت توسط سامانه در حالت تشخیص.....	۵۲
شکل ۴-۸: مکان‌یابی کارت توسط سامانه در حالت ذخیره.....	۵۵
شکل ۴-۹: تصویر کارت ملی به همراه زاویه اصلاح شده.....	۵۵
شکل ۴-۱۰: عدم تشخیص گواهینامه رانندگی توسط سامانه.....	۵۶
شکل ۴-۱۱: عدم تشخیص کارت دانشجویی توسط سامانه.....	۵۶

صفحه

فهرست جداول

جدول ۱-۲	اطلاعات مدل‌های پایه YOLOv5	۲۰
جدول ۱-۳	اطلاعات داده‌های آموزشی	۲۳
جدول ۲-۳	اطلاعات داده‌های آزمون	۲۴
جدول ۳-۳	نتایج آزمون برای مدل yolov5s	۳۶
جدول ۴-۳	نتایج آزمون برای مدل yolov5n	۳۸

فصل اول

مقدمه

در دنیای امروز، تعامل انسان با کامپیوترها به یکی از مهم‌ترین جنبه‌های زندگی روزمره تبدیل شده است. کاربردهای ساده‌ای چون جستجوی اطلاعات در اینترنت تا سیستم‌های پیچیده‌ای نظیر تشخیص چهره و تعاملات صوتی، حاکی از اهمیت این تعاملات می‌باشد. یکی از نمونه‌های بارز این تعاملات، سامانه‌های استخراج اطلاعات از کارت‌ها می‌باشد که به‌طور گسترده در بانک‌ها، سازمان‌های دولتی و بخش‌های مختلف به کار می‌روند. این سامانه‌ها، با استفاده از فناوری‌های پیشرفته، قادر به خواندن و استخراج اطلاعات از کارت‌ها بوده و فرآیندهای دستی و وقت‌گیر را که پیشتر توسط نیروی انسانی انجام می‌شد، به‌طور قابل توجهی تسریع می‌کنند.

در این میان، سامانه تشخیص کارت ملی به‌طور خاص دارای اهمیت ویژه‌ای است. کارت ملی، به‌عنوان یک مدرک هویتی رسمی، نقش حیاتی در فرآیندهای احراز هویت و تصدیق اطلاعات فردی دارد. سامانه‌های تشخیص کارت ملی با هدف تسهیل و تسریع فرآیندهای احراز هویت طراحی شده‌اند و کاربردهای گسترده‌ای در بانک‌ها، سازمان‌های دولتی، و خدمات آنلاین دارند. این سامانه‌ها نه تنها زمان انجام کارها را کاهش می‌دهند، بلکه دقت و امنیت فرآیندهای احراز هویت را نیز بهبود می‌بخشند.

یک سامانه استخراج اطلاعات از کارت ملی به صورت کلی سه گام را برای ثبت اطلاعات طی می‌کند:

۱- در گام اول، سامانه باید تصاویر کارت ملی را در صحنه شناسایی و مکان‌یابی کند.

۲- در گام دوم، سامانه اطلاعات موجود در کارت ملی را استخراج می‌کند.

۳- در گام سوم نیز، سامانه اطلاعات استخراج شده را به کمک یک ماژول نویسه‌خوان^۱ می‌خواند.

این فرآیند در شکل ۱-۱ قابل مشاهده می‌باشد.

^۱ Optical Character Recognition (OCR)



شکل ۱-۱: مراحل استخراج اطلاعات از کارت ملی

هدف از اجرای این پروژه، تنها پیاده‌سازی گام اول بوده است و گام‌های بعدی مورد بررسی قرار نگرفته‌اند.

دقت بالا برای یک سامانه تشخیص کارت ملی ضروری است، زیرا مکان‌یابی دقیق کارت ملی به عنوان اولین گام، مسیر استخراج اطلاعات از کارت را در گام‌های بعدی هموار می‌سازد و از بروز مشکلاتی نظیر درخواست‌های مکرر از کاربر جهت بارگذاری مجدد تصویر و در نتیجه تجربه بد کاربری جلوگیری می‌کند. از عوامل تاثیرگذار در دقت مکان‌یابی کارت می‌توان به تغییر فاصله کارت از دوربین، چرخش کارت، تغییرات شدت نور محیط، تفاوت‌های جنسیتی در عکس‌های کارت، و همچنین حضور کارت‌های متفرقه در صحنه اشاره کرد.

برای حل این مسئله، نیاز به استفاده از شبکه‌های تشخیص اشیاء مطمئن و مطرح وجود دارد. از بین شبکه‌های موجود، شبکه عصبی YOLO به ویژه نسخه پنجم آن YOLOv5 که به‌طور خاص برای تشخیص اشیاء طراحی شده است، انتخاب شده است. این شبکه‌ی معروف به دلیل متن‌باز بودن و مشارکت‌های مداوم در جامعه توسعه‌دهندگان، انتخابی قابل قبول می‌باشد.

شبکه YOLOv5 یکی از نسخه‌های مطرح در حوزه تشخیص اشیاء مبتنی بر شبکه‌های عصبی عمیق است که به دلیل سرعت و دقت بالا در کاربردهای مختلفی نظیر پردازش تصویر، امنیت، و رباتیک به کار گرفته می‌شود. این شبکه به‌طور خاص برای تشخیص سریع و همزمان چندین شیء در یک تصویر طراحی

شده است و به دلیل متن‌باز بودن و به‌روزرسانی‌های منظم، یکی از انتخاب‌های محبوب در میان توسعه‌دهندگان و پژوهشگران محسوب می‌شود.

یکی از دلایل اصلی انتخاب YOLOv5 به جای نسخه‌های جدیدتر آن مانند YOLOv7 یا YOLOv8، سادگی، بهینه‌سازی و پایداری این نسخه است. YOLOv5 به عنوان نسخه‌ای که به خوبی آزمایش شده و مستندات کاملی دارد، همواره به‌روزرسانی‌های مداوم از سوی جامعه توسعه‌دهندگان دریافت می‌کند. این شبکه به دلیل پشتیبانی گسترده از ابزارهای مختلف مانند PyTorch برای اجرا در سیستم‌های مختلف، همچنان انتخاب قابل قبولی برای پروژه‌هایی است که نیازمند کارایی بالا و انعطاف‌پذیری هستند.

علاوه بر این، YOLOv5 دارای مدل‌های متنوعی است که به‌ویژه در کاربردهایی که محدودیت منابع پردازشی وجود دارد، بسیار مفید هستند. این تنوع مدل‌ها امکان انتخاب مدل مناسب بسته به نیاز پروژه را فراهم می‌کند، به‌طوری‌که می‌توان با توجه به محدودیت‌های سخت‌افزاری یا نیاز به سرعت بیشتر، از مدل‌های سبک‌تر یا در کاربردهای نیازمند دقت بالاتر، از مدل‌های پیچیده‌تر بهره گرفت.

همچنین، با وجود نسخه‌های جدیدتر مانند YOLOv7 و YOLOv8، بسیاری از پروژه‌ها به دلیل ثبات، پشتیبانی وسیع و گستردگی جامعه کاربران همچنان از YOLOv5 استفاده می‌کنند، چرا که در صورت وقوع مشکل در کار کردن با این شبکه، می‌توان از کمک‌های این جامعه استفاده کرد.

رفع پردازش‌های غیرضروری نیز از دیگر ملزومات این سامانه می‌باشد. در صورت اجرای مکان‌یابی کارت در سمت سرور، حجم زیادی از داده‌های اضافی به سمت سرور روانه می‌شود. همین امر باعث ایجاد پردازش‌های غیرضروری در سرور می‌گردد و سرور را از خدمت‌رسانی مناسب و سریع به کاربرها باز می‌دارد. اجرای پردازش‌های اولیه در سمت کاربر، باعث کاهش بار پردازشی سرور می‌شود. به همین دلیل نیاز است که این سامانه بتواند بدون ارتباط با سرور، کارت را مکان‌یابی و زاویه آن را اصلاح کند و سپس تصویر بریده شده کارت را به سرور بفرستد. به همین دلیل استفاده از محاسبات لبه و اجرای نرم‌افزار در سمت کاربر، راه‌حلی کارآمد برای کاهش محاسبات غیرضروری به‌نظر می‌رسد.

همانطور که اشاره شد، از مزایای شبکه YOLOv5، وجود مدل‌های از پیش آموخته شده‌ی متنوع می‌باشد. این مدل‌ها، در تعداد پارامترهای شبکه عصبی پیاده‌سازی شده با یکدیگر تفاوت دارند. این تفاوت به استفاده‌کنندگان اجازه انتخاب مدل‌های متناسب با کاربرد خاص خودشان را می‌دهد. با توجه به اجرای

برنامه در سمت کاربر و وجود محدودیت های پردازشی احتمالی، دو مدل ساده تر YOLOv5 یعنی yolov5s (مدل اصلی) و مدل yolov5n (ساده ترین مدل) جهت بررسی و مقایسه انتخاب شده اند.

همانطور که در شکل ۱-۱ هم مشاهده می شود، گام بعدی سامانه استخراج اطلاعات از کارت ملی، برای مکان یابی و استخراج اطلاعات آن، باید کارت را به صورت مستقیم و بدون دوران داشته باشد، چرا که در غیر این صورت نمی تواند اطلاعات را به درستی استخراج کند و برای نویسه خوانی به گام بعدی تحویل دهد. به همین دلیل نیاز است که زاویه کارت را نیز پیش از ارسال به گام بعدی تصحیح کنیم.

روشی که برای اصلاح زاویه کارت در پیش گرفته شده است، استفاده از نمادهای کارت ملی می باشد. این نمادها همواره همراه کارت ملی حضور دارند و می توان با مکان یابی آنها و سپس محاسبه شیب بین مراکز دو نماد بالایی، زاویه کارت را اصلاح کرد. به دلیل اینکه تشخیص اشیا بیشتر تاثیری بر زمان استنتاج شبکه YOLOv5 ندارد، از این روش به جای روش هایی نظیر تشخیص لبه و... که بار محاسباتی اضافی را به سامانه تحمیل می کنند استفاده شده است. تصویر کارت ملی به همراه نمادهای آن در شکل ۱-۲ قابل مشاهده است.



شکل ۱-۲: کارت ملی به همراه نمادهای آن

استقرار سامانه در سمت کاربر می تواند به شکل های نرم افزارهای دسکتاپ، نرم افزارهای موبایل، و یا اجرا در مرورگر و از طریق وب صورت پذیرد. اجرای نرم افزار در مرورگر به دلیل در دسترس بودن آن در سیستم عامل ها و دستگاه های مختلف انتخابی معقولانه به نظر می رسد، چرا که کاربران را از نصب نرم افزارهای متعدد بی نیاز می کند و تنها با داشتن یک مرور، می توانند از خدمات این سامانه استفاده کنند.

برای توسعه سامانه در سمت کاربر و استقرار آن در مرورگر، از زبان جاوا اسکریپت بهره‌گیری شده است. این زبان امکانات فوق العاده زیادی را جهت توسعه واسط کاربری در اختیار توسعه‌دهندگان قرار می‌دهد و اجازه نوشتن واسط‌های کاربری پیشرفته و جذاب را در اختیار برنامه‌نویسان می‌گذارد. وجود کتابخانه قدرتمند TensorFlow.js در این زبان، قابلیت استفاده از مدل‌های یادگیری عمیق در مرورگر را فراهم می‌کند.

با توجه به نیازمندی‌های تشریح شده، برای توسعه سامانه‌ای با دقت بالا جهت تشخیص کارت ملی، ابتدا نیاز است که یک مجموعه داده متنوع و جامع از تصاویر کارت ملی تحت شرایط محیطی مختلف تهیه شود. با توجه به چالش‌های متعددی که در فرآیند تشخیص دقیق کارت ملی وجود دارد، ضروری است که این چالش‌ها در جمع‌آوری و آماده‌سازی مجموعه داده به‌خوبی در نظر گرفته شوند. مجموعه داده فراهم شده باید به دقت برچسب زده شود تا فرآیند آموزش مدل‌ها به بهترین شکل صورت پذیرد.

مدل‌های انتخاب‌شده در این پروژه، نسخه‌های ساده و سبک yolov5n و yolov5s هستند، چرا که این مدل‌ها با وجود دقت کافی، به منابع پردازشی کمتری نیاز دارند و قابلیت اجرا در دستگاه‌هایی با توان پردازشی محدود را دارند. به دلیل اینکه هدف اصلی این پژوهش، اجرای سامانه در سمت کاربر بدون نیاز به ارسال حجم زیادی از داده به سرور می‌باشد، استفاده از مدل‌های بهینه و سبک‌تر، ضروری به نظر می‌رسد.

پس از آموزش مدل‌ها، استفاده از آنها در مرورگر کاربر به‌عنوان یک چالش کلیدی مطرح می‌شود. با این راهکار، مدل‌های آموزش‌دیده به‌طور مستقیم در مرورگر بارگذاری و اجرا می‌شوند. این روش مزیت‌های زیادی را به دنبال دارد. اولاً حجم داده‌های ارسالی به سرور به حداقل می‌رسد و در نتیجه، زمان پردازش کاهش می‌یابد. ثانیاً کاربران بدون نیاز به نصب نرم‌افزارهای سنگین، تنها با مرورگر خود می‌توانند از سامانه استفاده کنند.

در فصل دوم این گزارش، به معرفی و بررسی سامانه‌های مرتبط و همچنین شبکه‌های تشخیص اشیا به ویژه شبکه عصبی YOLOv5 و کاربرد آن در این پروژه پرداخته می‌شود. معماری این شبکه و برخی ویژگی‌های آن تشریح خواهند شد و با مراحلی که شبکه برای تشخیص اشیا طی می‌کند آشنا می‌شویم.

پس از آشنا شدن با شبکه، در فصل سوم جمع‌آوری داده‌ها، نحوه آماده‌سازی مجموعه داده و برچسب‌زنی آن، فرآیند آموزش و تنظیم دقیق^۲ مدل جهت رسیدن به یک مدل تشخیص کارت ملی مورد بحث قرار می‌گیرند. در انتهای فصل نیز، عملکرد مدل‌های آموزش دیده بر داده‌های آزمون مورد بررسی قرار می‌گیرد و مقایسه‌ای بین نتایج صورت خواهد پذیرفت.

در فصل چهارم، به شرح تکنولوژی مورد استفاده و ویژگی‌های موجود در واسط کاربری سامانه پرداخته خواهد شد. در ابتدا، با نحوه صدور مدل‌های yolov5 به فرمت مناسب برای استفاده در مرورگر آشنا خواهیم شد. سپس چالش‌هایی که در طی توسعه واسط کاربری به وجود آمده‌اند مرور می‌شوند. در ادامه نیز، عملکرد سامانه در تشخیص و مکان‌یابی کارت‌های ملی و دیگر کارت‌های منفی نشان داده می‌شود و در آخر نیز مقایسه‌ای بین عملکرد مدل‌ها در زمان اجرا صورت می‌گیرد و نکاتی در خصوص تصمیم‌گیری استقرار نهایی سامانه ذکر خواهد شد.

در فصل پنجم و آخر نیز جمع‌بندی و نتیجه‌گیری از موضوعات مطرح شده صورت می‌گیرد. همچنین با توجه به برخی چالش‌های مواجه شده در روند طراحی و توسعه سامانه، پیشنهادهای جهت پژوهش‌های آتی ارائه می‌گردد.

^۲ Fine Tune

فصل دوم

مرور کارهای مرتبط و شبکه‌های عصبی تشخیص اشیا

در بخش اول این فصل، مروری بر سامانه‌های احراز هویت مبتنی بر استخراج اطلاعات از کارت و همچنین شبکه‌های تشخیص اشیای معروف صورت خواهد پذیرفت. با انجام این بررسی، شبکه YOLOv5 به عنوان شبکه موردنظر برای پیاده‌سازی پروژه معرفی خواهد شد. ویژگی‌ها و مزایای این شبکه تشریح می‌شود تا خواننده با دلایل انتخاب این شبکه برای این پروژه‌ی آشنا شود. همچنین، به برخی از تکنیک‌های استفاده شده در YOLOv5 که منجر به دقت و سرعت بالای آن شده است، اشاره خواهد شد.

در زیرفصل دوم، ویژگی‌های شبکه YOLOv5 به تفصیل بررسی خواهد شد و خواننده با روش کار و معماری درونی این شبکه به صورت کامل آشنا خواهد شد. در انتهای فصل نیز از مطالب عنوان شده نتیجه‌گیری خواهد شد.

۲-۱- مرور سامانه‌ها و شبکه‌های مرتبط

احراز هویت در سامانه‌های برخط و دیجیتال به عنوان یک چالش کلیدی در دنیای امروز مطرح است. در ایران، کارت ملی به عنوان یکی از اصلی‌ترین مدارک هویتی شناخته می‌شود و استفاده از آن در فرایندهای احراز هویت، به دلیل اعتبار بالا و گستردگی استفاده، به طور وسیعی مورد توجه قرار گرفته است. سامانه‌های تجاری مختلفی در ایران برای خوانش کارت ملی توسعه یافته‌اند که در ادامه به برخی آنها می‌پردازیم:

- احراز هویت دیجیتال ویدا^۳: در ویدا، انواع خدمات احراز هویت دیجیتال مبتنی بر پارامترهای بیومتریک مانند احراز هویت مبتنی بر ویدئو و تصویر، خدمات تکمیلی آن مانند سرویس‌های تبدیل تصویر به متن (OCR) و همچنین امضای دیجیتال ارائه می‌شود. با استفاده از سرویس OCR ویدا می‌توان انواع اسناد هویتی مانند کارت ملی، شناسنامه، گذرنامه، عابر بانک، گواهینامه رانندگی و غیره را تشخیص داده و سپس متون آن اسناد بدون آنکه نیاز باشد کاربران به صورت دستی اطلاعات هویتی یا مالی خود را وارد کنند، استخراج کرد.

³ <https://vidaverify.ir>

- احراز هویت آنالین لوانت^۴: لوانت با استفاده از راه‌کارهای مبتنی بر هوش مصنوعی، در حوزه احراز هویت و ثبت‌نام مشتریان محصولات کاربردی و متنوعی تولید کرده که در مقیاس بالا مورد بهره‌برداری قرار گرفته‌اند. این مجموعه، سرویس‌های متنوعی مانند خوانش متن کارت بانکی، خوانش متن کارت ملی، تطبیق چهره، سرویس نویسه‌خوان چک صیاد و... را در اختیار مشتریان خود قرار می‌دهد.

- احراز هویت بیومتریک سپید سیستم^۵: شرکت سپید سیستم با فعالیت در حوزه راه‌اندازی سامانه‌های مبتنی با بیومتریک، امکان ذخیره و بازیابی سنج‌های هویتی مانند چهره، اثر انگشت و امضا به صورت برخط (اسکن) و یا برون خط (مانند اثر انگشت برداشته شده از صحنه جرم و یا به صورت جوهری و یا با داشتن یک عکس از چهره یک فرد) را فراهم می‌کند. در این سامانه از روش‌های بهینه شناسایی بیومتریکی شامل تشخیص اثر انگشت، تشخیص چهره و تشخیص امضا بهره گرفته می‌شود که سرعت و دقت بالایی را برای جستجو بین تعداد زیادی الگوی بیومتریکی ارائه می‌کند. علاوه بر این، این شرکت امکان احراز هویت به وسیله برخی داده‌هایی که در کارت‌های هوشمند (نظیر کارت ملی، گواهینامه، و یا گذرنامه هوشمند) ثبت شده‌اند نیز فراهم می‌کند.

با توجه به اینکه یکی از بخش‌های اصلی سامانه، تشخیص کارت می‌باشد، در ادامه مروری بر شبکه‌های عصبی تشخیص اشیا خواهیم داشت:

- شبکه YOLO^۶ [۱]: شبکه YOLO یکی از معروف‌ترین و پرکاربردترین شبکه‌های عصبی در زمینه تشخیص اشیاء است. ویژگی برجسته YOLO این است که برخلاف روش‌های سنتی که

^۴ <https://levants.io>

^۵ <https://sepidsystem.com/biometric-authentication-solution>

^۶ You Only Look Once

از پنجره‌های لغزان^۷ یا پیشنهاد مناطق^۸ استفاده می‌کردند، تنها در یک مرحله^۹ قادر به تشخیص اشیاء است. این ویژگی باعث شده تا YOLO به عنوان یکی از سریع‌ترین مدل‌های تشخیص اشیاء شناخته شود. YOLO تصویر ورودی را به یک شبکه عصبی پیچشی^{۱۰} می‌دهد و در خروجی، مستقیماً مختصات جعبه‌های دربرگیرنده^{۱۱} و احتمال دسته‌های مختلف اشیاء را تولید می‌کند. این رویکرد باعث شده تا YOLO بتواند به صورت بی‌درنگ^{۱۲} تشخیص اشیاء را انجام دهد و در کاربردهایی که نیازمند سرعت بالا هستند، به شدت کارآمد باشد [۱]. نسخه‌های مختلفی از YOLO ارائه شده است که هر کدام بهبودهایی در سرعت و دقت داشته‌اند. در این میان، YOLOv5 که از آن به عنوان یک مدل بهینه‌تر یاد می‌شود، به طور گسترده‌ای در کاربردهای مختلف مورد استفاده قرار گرفته است.

- شبکه SSD^{۱۳} [۲]: شبکه SSD یکی دیگر از شبکه‌های معروف در زمینه تشخیص اشیاء است. SSD به عنوان یک مدل تک‌مرحله‌ای طراحی شده که مشابه YOLO، بدون نیاز به فرآیند پیشنهاد مناطق، به طور مستقیم مختصات جعبه‌های مرزی و برجسب‌های مربوط به هر شیء را پیش‌بینی می‌کند. SSD از یک شبکه عصبی پیچشی به همراه یک سری از نقشه‌های ویژگی^{۱۴} چندگانه استفاده می‌کند که هر کدام اندازه و دقت متفاوتی دارند. این ویژگی به SSD امکان می‌دهد که اشیاء با اندازه‌های مختلف را با دقت بالاتری تشخیص دهد. مزیت اصلی SSD نسبت به YOLO در توانایی آن برای تشخیص اشیاء با مقیاس‌های مختلف و حفظ تعادل بین سرعت و دقت است [۲]. با این حال، SSD در شرایطی که نیاز به سرعت بسیار بالا مطرح باشد، عملکرد YOLO را ندارد و ممکن است در کاربردهای بی‌درنگ به عنوان گزینه دوم در نظر گرفته

⁷ Sliding Windows

⁸ Region Proposal

⁹ Single Stage

¹⁰ Convolutional Neural Network

¹¹ Bounding Box

¹² Real Time

¹³ Single Shot MultiBox Detector

¹⁴ Feature Maps

شود [۳]. با این وجود، SSD به دلیل قابلیت تشخیص چندین شیء با اندازه‌های مختلف در یک تصویر، همچنان یکی از گزینه‌های محبوب در بسیاری از پروژه‌های تشخیص اشیا است.

شبکه YOLOv5 یکی از پیشرفته‌ترین و محبوب‌ترین مدل‌های تشخیص اشیا است که توسط Ultralytics توسعه داده شده است [۴]. این شبکه به دلیل سرعت و دقت بالا، در کاربردهای متنوعی از جمله تشخیص اشیا در تصاویر و ویدئوها به کار می‌رود. YOLOv5 بر اساس اصول اولیه‌ی نسخه‌های قبلی YOLO [۱] طراحی شده، اما با بهبودهای قابل توجهی که در معماری شبکه و الگوریتم‌های یادگیری همراه شده، عملکرد آن به طرز چشم‌گیری ارتقا داده شده است.

ویژگی دیگر YOLOv5، سهولت در آموزش و پیاده‌سازی برنامه با آن است. این شبکه با استفاده از زبان برنامه‌نویسی Python و کتابخانه‌ی محبوب PyTorch [۵] توسعه یافته است که امکان سفارشی‌سازی و بهینه‌سازی مدل را برای نیازهای خاص فراهم می‌کند. همچنین، YOLOv5 دارای مجموعه‌ای از ابزارها و اسکریپت‌های از پیش آماده شده برای آموزش و ارزیابی مدل است که فرآیند استفاده از آن را برای توسعه‌دهندگان بسیار ساده‌تر می‌سازد. با این تفاسیر، انتخاب شبکه عصبی YOLOv5 برای این پروژه انتخابی مناسب به نظر می‌رسد، چرا که دقت بالا و مناسبی را در اختیار ما قرار می‌دهد.

یکی از محدودیت‌های اصلی الگوریتم‌های تشخیص اشیا، استخراج جعبه دربرگیرنده شیء تنها به صورت موازی با محورهای مختصات است. بدین معنی که YOLO مانند بسیاری از الگوریتم‌های تشخیص اشیا، جعبه‌ای مستطیلی و افقی را برای مکان‌یابی اشیا در نظر می‌گیرد. بنابراین چنانچه کارت ملی در تصویر چرخیده باشد و دوران کرده باشد، YOLO به‌تنهایی نمی‌تواند میزان دوران کارت را تشخیص دهد.

چند رویکرد برای رفع مشکل اصلاح زاویه کارت وجود دارد:

- استفاده از مدل‌های تشخیص دهنده شیء و زاویه: در این رویکرد از مدل‌هایی نظیر YOLOv5_OBB¹⁵ [۶] استفاده می‌گردد. برای این منظور کارت‌های موجود در تصویر به صورت باکس‌های زاویه‌دار برچسب زده می‌شوند و مدل‌ها بر اساس این داده‌ها آموزش می‌بینند. مهمترین مزیت این مدل‌ها تشخیص زاویه کارت به همراه کارت تشخیص داده شده است اما غیراستاندارد بودن مدل پایه استفاده شده، پشتیبانی ضعیف و سرعت پردازش پایین باعث می‌شود که این رویکرد، رویکرد قابل اطمینانی برای اجرای پروژه نباشد.



شکل ۲-۱: تشخیص اشیا به همراه زاویه [۶]

- استفاده از تصویر چهره روی کارت: در این روش، از ویژگی‌های تصویر برای مکان‌یابی چشم‌ها روی کارت استفاده می‌شود. در صورت مکان‌یابی چشم‌ها، می‌توان زاویه دوران کارت را تشخیص داد. اما این روش نیز بسیار به وضوح تصویر چهره روی کارت وابسته است و نسبت به تنوع تصاویر اشخاص نیز حساس است. به همین دلیل این رویکرد نیز نمی‌تواند برای اجرای پروژه استفاده شود.

¹⁵ YOLOv5 Oriented Bounding Box

- استفاده از نمادهای ثابت روی کارت: این رویکرد، بر پایه استخراج سه لوگوی حاشیه ای کارت ملی و تشخیص زاویه دوران به کمک آن ها است. به دلیل اینکه YOLOv5 در یکبار استنتاج تصویر، می‌تواند همه اشیا تعریف شده را تشخیص دهد، استفاده از این رویکرد سرعتی معادل تشخیص کارت به تنهایی خواهد داشت. به همین علت، مسئله تشخیص و مکان‌یابی کارت ملی و نمادهای آن به یک مسئله تشخیص اشیا چهار کلاسه فرموله می‌شود. این چهار کلاس در شکل ۲-۲ نشان داده شده‌اند.

در نهایت با توجه به مزایا و معایب ذکر شده، از رویکرد اخیر به عنوان رویکرد پیشنهادی جهت اصلاح زاویه در این پروژه استفاده می‌گردد. در این حالت اگر کارت به صورتی در مقابل دوربین قرار گرفته شده باشد که دوران زیاد حول محورهای کناری کارت که باعث دیده نشدن تصویر روی کارت می‌شوند، نداشته باشد و تنها حول محور عمود بر کارت دوران کرده باشد و نمادهای حاشیه‌ای آن در تصویر قابل مشاهده باشند، امکان تشخیص و اصلاح زاویه دوران کارت و استخراج اطلاعات آن وجود خواهد داشت. با استخراج خود کارت ملی و نمادهای حاشیه‌ای آن توسط جعبه‌های دربرگیرنده، زاویه دوران تقریب زده می‌شود و معکوس این دوران به تصویر اعمال می‌گردد.

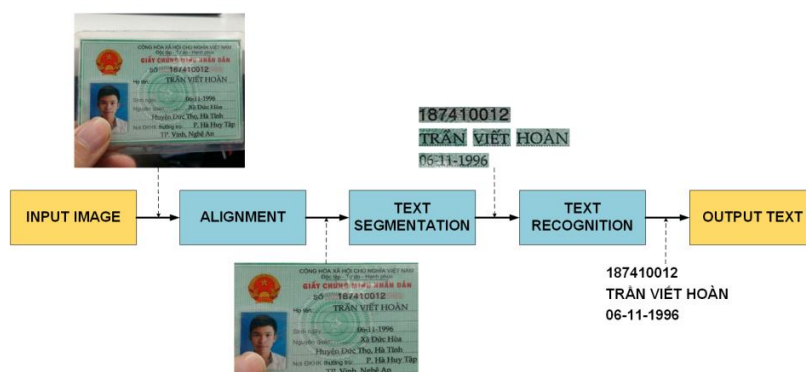
با فرضیات فوق می‌توان از رویکرد استاندارد YOLOv5 برای تشخیص کارت استفاده نمود و در عین حال دوران کارت را با توجه به محتوای آن (نمادهای حاشیه‌ای) تخمین زد و کارت را به صورت افقی در مرکز جعبه تقریب زده شده، قرار داد.



شکل ۲-۲: تشخیص کارت ملی به همراه نمادها و اصلاح زاویه

در ادامه این بخش، مقالاتی که از تشخیص اشیا برای کاربردهای مختلف استفاده کرده‌اند بررسی می‌شوند.

در مقاله [۷]، یک سامانه استخراج اطلاعات از کارت شناسایی ویتنامی معرفی شده است. در ابتدا، سیستم کارت شناسایی را بر اساس تشخیص گوشه‌ها و نشانه ملی ویتنام تراز می‌کند. در گام بعدی، این سامانه از یک شبکه عصبی پیچشی عمیق برای مکان‌یابی نوشته‌ها استفاده می‌کند. این نوشته‌ها بر اساس موقعیت آنها دسته‌بندی شده و در نهایت یک نویسه‌خوان مبتنی بر Attention برای خوانش نوشته‌ها به کار گرفته می‌شود. این فرآیند در شکل ۲-۳ قابل مشاهده است.



شکل ۲-۳: روند استخراج اطلاعات در مقاله [۷]

به جهت تشخیص گوشه‌ها و نماد ملی، از شبکه عصبی سبک SSD نسخه MobileNet V2 استفاده شده است. برای مکان‌یابی نوشته‌ها، همه آنها یک کلاس در نظر گرفته می‌شوند و به جهت دسته‌بندی آنها، از موقعیت آنها استفاده شده است. این موضوع در شکل ۲-۴ قابل مشاهده است. شبکه تشخیص دهنده نوشته‌ها نیز همان شبکه MobileNet V2 است که توانسته دقت بالایی همراه با زمان پردازشی کمی داشته باشد. در این پژوهش، برای تشخیص گوشه‌ها به دقت ۹۸ درصد و برای مکان‌یابی نوشته‌ها به دقت ۸۸.۵ درصد رسیده‌اند.



شکل ۲-۴: استخراج نوشته‌ها در مقاله [۷] و مرتب‌سازی بر اساس موقعیت

در مقاله [۸]، برای تشخیص پلاک، از شبکه عصبی YOLOv7 استفاده شده است. در این پژوهش، تصاویر از دوربین‌های مدار بسته بزرگراه‌ها دریافت شده و در ابتدا پلاک ماشین توسط این شبکه تشخیص داده می‌شود. در ادامه، به کمک شبکه U-Net، تصویر پلاک قطعه‌بندی می‌شود و در نهایت اعداد و حروف قطعه شده، با استفاده از روش Template Matching خوانده می‌شوند. نتایج بدست آمده از این پژوهش نشان می‌دهد که شبکه YOLOv7 توانسته به دقت ۹۷.۰۵ درصد برسد، و دقتی که برای شبکه YOLOv5 ۹۵.۴۷ درصد عنوان شده است را بهبود ببخشد.

۲-۲- شبکه YOLOv5

توضیحات ارائه شده در این بخش، خلاصه مقاله [۹] می‌باشد.

در YOLO، تشخیص اشیا به عنوان یک مسئله‌ی رگرسیون تلقی می‌شود. به عبارت دیگر، به جای تقسیم تصویر به نواحی مختلف و اعمال تشخیص چندگانه مانند روش‌های مبتنی بر دسته‌بندی، مدل YOLO به کل تصویر به عنوان ورودی نگاه کرده و در یک گام، مختصات جعبه دربرگیرنده^{۱۶} و احتمالات مربوط به اشیا مختلف را پیش‌بینی می‌کند. این بدان معناست که YOLO همزمان موقعیت و نوع شیء را به‌طور مستقیم پیش‌بینی می‌کند، که این فرآیند مشابه رویکرد رگرسیون برای پیش‌بینی مقادیر پیوسته است.

۲-۲-۱ معماری شبکه YOLOv5

YOLO یک معماری سرتاسری^{۱۷} برای تشخیص اشیا معرفی کرده است که وظایف تشخیص جعبه دربرگیرنده و دسته‌بندی آن را با یک شبکه عصبی انجام می‌دهد. معماری شبکه YOLO به صورت کلی از سه قسمت اصلی تشکیل شده است:

¹⁶ Bounding Box

¹⁷ End To End

- **Backbone** (شبکه پایه): یک شبکه عصبی می‌باشد که وظیفه رمز کردن^{۱۸} اطلاعات موجود در تصویر به ویژگی‌ها را بر عهده دارد. در شبکه YOLOv5، شبکه CSPDarknet53 به عنوان Backbone استفاده شده است. این شبکه یک نسخه بهبود یافته از Darknet است که باعث افزایش بهره‌وری در استخراج ویژگی‌ها می‌شود. استفاده از قابلیت Cross Stage Partial connections یا (CSP) باعث بهبود کارایی در استخراج ویژگی‌ها می‌شود.
- **Neck** (بخش میانی): یک سری لایه‌هایی هستند که برای یکپارچه‌سازی و بهبود ویژگی‌های بدست آمده در مرحله قبل طراحی شده‌اند. خروجی این قسمت، سه نقشه است که برای تشخیص اشیا کوچک، متوسط و بزرگ استفاده می‌شوند. شبکه‌های Feature Pyramid Path و Aggregation در این بخش استفاده می‌شوند.
- **Head** (بخش انتهایی): قسمتی از سامانه است که جعبه‌ها و اسم کلاس‌ها را بر اساس ویژگی‌های پردازش شده تولید می‌کند. با به‌کارگیری چند حد آستانه، اطلاعات نامطمئن را کنار می‌گذارد و با اجرای فرآیند سرکوب غیر بیشینه‌ها^{۱۹} (NMS) اطلاعات نهایی را خروجی می‌دهد. این فرآیند در بخش بعدی توضیح داده شده است.

۲-۲-۲ - مراحل تشخیص اشیا

- در YOLOv5، فرآیند تشخیص اشیا به صورت زیر انجام می‌شود:
- تقسیم تصویر به **Grid** (شبکه‌بندی تصویر): تصویر ورودی به یک شبکه (Grid) تقسیم می‌شود. هر سلول این شبکه مسئول پیش‌بینی یک یا چند ناحیه به همراه احتمال حضور شیء در آن ناحیه می‌باشد.
 - پیش‌بینی جعبه‌ها و کلاس‌ها: هر سلول، مختصات ناحیه‌ای که در آن اشیا را تشخیص داده اعلام می‌کند.
 - اعمال **Non Maximum Supression (NMS)**: هر شیء ممکن است توسط چندین سلول مختلف تشخیص داده شود. با اعمال NMS، تنها جعبه‌ای انتخاب می‌شود که بیشترین اطمینان را داشته باشد. این تکنیک باعث حذف جعبه‌های دارای اشتراک برای یک شیء خاص می‌شود.

¹⁸ Encoding

¹⁹ Non Max Suppression

جعبه‌های دارای اشتراک به وسیله معیار اشتراک بر اجتماع^{۲۰} (IoU) تشخیص داده می‌شوند و اگر جعبه‌هایی دارای IoU بیش از یک حد آستانه باشند، تنها جعبه با اطمینان بیشتر نگه داشته خواهد شد.

۲-۲-۳ ویژگی‌های YOLOv5 در آموزش

موثر بودن یک شبکه تشخیص اشیا، علاوه بر معماری زیرین آن، به روش آموزش آن نیز بستگی دارد. دو ویژگی خیلی مهم در آموزش شبکه YOLOv5 وجود دارد:

- افزونگی داده^{۲۱}: در این روش، شبکه با به وجود آوردن دگرگونی^{۲۲}های مختلف نظیر چرخش بر روی اعضای مجموعه داده، به قوام^{۲۳} و عمومیت بخشیدن شبکه کمک می‌کند. به همین دلیل، مدل آموزش دیده بر روی داده‌های جهان واقعی که شرایط متفاوتی نسبت به مجموعه داده دارند مقاوم خواهد شد.
- تابع Loss: از مجموع سه جزء اصلی محاسبه می‌شود. IoU عمومی، شیء بودن^{۲۴} و خطای دسته‌بندی کلاس‌ها. تابع Loss، مجموع وزن دار این سه ویژگی می‌باشد که در شکل ۲-۵ آورده شده است.

$$\text{Loss} = \lambda_1 \cdot L_{\text{cls}} + \lambda_2 \cdot L_{\text{obj}} + \lambda_3 \cdot L_{\text{loc}}$$

شکل ۲-۵: تابع Loss آموزش شبکه YOLOv5

L_{cls} خطای پیش‌بینی کلاس، L_{obj} خطای شیء بودن و L_{loc} خطای مکان‌یابی با توجه به IoU می‌باشد.

²⁰ Intersection over Union

²¹ Data Augmentation

²² Transformation

²³ Robustness

²⁴ Objectness

۲-۲-۴ - اطلاعات مدل‌های از پیش آموخته شده YOLOv5

در جدول ۱-۲، اطلاعاتی در مورد مدل‌های پایه و از پیش آموخته شده YOLOv5 آورده شده است.

جدول ۱-۲ اطلاعات مدل‌های پایه YOLOv5

مدل	تعداد پارامترها (میلیون)	دقت	زمان CPU (ms)	زمان GPU (ms)
YOLOv5n	۱.۹	۴۵.۷	۴۵	۶.۳
YOLOv5s	۷.۲	۵۶.۸	۹۸	۶.۴
YOLOv5m	۲۱.۲	۶۴.۱	۲۲۴	۸.۲
YOLOv5l	۴۶.۵	۶۷.۳	۴۳۰	۱۰.۱
YOLOv5x	۸۶.۷	۶۸.۹	۷۶۶	۱۲.۱

۲-۳ - نتیجه‌گیری

YOLOv5 به عنوان یک پیشرفت قابل توجه در تشخیص اشیا ظاهر شده است و تعادل قانع کننده‌ای از سرعت، دقت و کاربرپسندی را از خود نشان می‌دهد. در حالی که معماری اصلی مبتنی بر اصول قبلی YOLO تعیین شده است، پیاده‌سازی مدل در چارچوب PyTorch، یک جهش قابل توجه به جلو است که هم کارایی و هم قابلیت‌های استقرار را افزایش می‌دهد. در دسترس بودن انواع مدل‌های متعدد متناسب با محدودیت‌های محاسباتی متنوع، کاربرد YOLOv5 را در حوزه‌های مختلف گسترش می‌دهد.

در دسترس بودن مدل، همراه با سازگاری آن با ابزارها و پلتفرم‌های موجود، YOLOv5 را به عنوان یک راه حل همه کاره برای کاربردهای تحقیقاتی و عملی معرفی می‌کند. از آنجایی که حوزه بینایی کامپیوتر به سرعت در حال تکامل است، YOLOv5 به عنوان پایه‌ای قوی برای پیشرفت‌های آینده در تشخیص اشیا و بلادرنگ و وظایف مرتبط عمل می‌کند.

فصل سوم

تشخیص کارت ملی

هدف از ارائه این فصل، بررسی روند طی شده جهت آموزش مدل تشخیص کارت ملی می‌باشد. با آشنا شدن با شبکه‌های تشخیص دهنده اشیا و ویژگی آنها در فصل قبل، اکنون خواننده قابل است تا با مراحل آموزش و تنظیم دقیق یک مدل تشخیص اشیا بر روی اشیا دلخواه آشنا شود.

۳-۱- مجموعه داده

۳-۱-۱- جمع آوری داده

در این پروژه سعی شده است که مجموعه داده متنوعی از کارت‌های ملی مختلف جمع‌آوری شود که همه چالش‌های ذکر شده موثر بر دقت شبکه را پوشش دهد. برای جمع‌آوری داده، فیلم برداری از کارت‌های ملی افراد (با رضایت آنها) انجام گرفته است. در فیلم برداری از کارت‌ها، به چرخش حول زوایای مختلف، شدت نورهای متفاوت، پس زمینه‌های متنوع، پوشیدگی‌های جزئی و کلی کارت، فواصل مختلف از دوربین، محل قرار گرفتن کارت در تصویر، و حالت‌های مختلف در دست گرفتن کارت توجه شده است. همچنین، با توجه به اینکه ممکن است در صحنه کارت‌های دیگری نظیر گواهینامه رانندگی، کارت پایان خدمت، کارت دانشجویی و... نیز حضور داشته باشند، سیستم و شبکه نباید آنها را تشخیص دهند. به همین جهت نیاز است داده‌های منفی شامل این قبیل کارت‌ها در مجموعه داده حضور داشته باشند. داده‌های منفی نیازی به برچسب خوردن ندارند و تنها حضور آنها کفایت می‌کند.

داده‌های جمع‌آوری شده به دو بخش آموزش و آزمون تقسیم شده‌اند. برای تولید داده‌های برچسب‌دار، از کارت ملی ۲۸ نفر فیلم‌برداری انجام شده است و به هر فرد یک عدد اختصاص داده شده است. هر فرد می‌تواند چندین ویدئو از کارت ملی خود داشته باشد که هر ویدئو نیز یک عدد به خود می‌گیرد. بعد از فیلم‌برداری، ویدیوها به کمک یک کد کوتاه^{۲۵} به زبان پایتون، به فریم‌ها تبدیل شدند تا داده‌ها در قالب عکس ذخیره شوند. نام فریم‌ها در فرمت خاصی ذخیره می‌شود، در قسمت اول، شمارنده

²⁵Script

فرد، در قسمت دوم، شمارنده ویدئو و در قسمت سوم شماره آن فریم ذخیره می‌شود. برای مثال، فریم 001_3_0025، نشان دهنده فریم ۲۵م از سومین ویدئو ضبط شده از فرد اول است. در جدول ۱-۳ اطلاعات داده‌های آموزشی و در جدول ۲-۳ اطلاعات مربوط به داده‌های آزمون به صورت خلاصه بیان شده‌اند.

جدول ۱-۳ اطلاعات داده‌های آموزشی

عنوان	تعداد
تعداد افراد	۲۳
تعداد ویدیوهای ضبط شده	۴۵
تعداد کارت ملی آقایان	۱۲
تعداد کارت ملی بانوان	۱۱
تعداد فریم‌های استخراج شده	۱۲۱۸۴۸
تعداد فیلم‌های داده‌های منفی	۱۱
تعداد عکس‌های داده‌های منفی	۱۴۶
تعداد کل تصاویر منفی	۲۱۱۱۳

جدول ۳-۲ اطلاعات داده‌های آزمون

عنوان	تعداد
تعداد افراد	۵
تعداد ویدیوهای ضبط شده	۱۰
تعداد کارت ملی آقایان	۳
تعداد کارت ملی بانوان	۲
تعداد فریم‌های استخراج شده	۸۲۲۵
تعداد فیلم‌های داده‌های منفی	۱
تعداد کل تصاویر منفی	۶۲۶

با توجه به نیاز به یک مدل مقاوم و با دقت بالا در شرایط مختلف، در تولید مجموعه داده‌ها عوامل مختلف محیطی در نظر گرفته شده است. در ادامه برخی از این شرایط به همراه نمونه‌هایی از تصاویر مجموعه داده که چالش مورد نظر در آنها وجود دارد، نمایش داده شده‌اند.

- تصاویر دارای زاویه‌های مختلف (مستقیم، چرخیده) و فاصله‌های دور و نزدیک هستند



شکل ۳-۱: حضور کارت در زوایا و فاصله‌های مختلف

- تصاویر در شرایط نورپردازی متنوع (روز و شب، هر کدام با شدت نورهای مختلف) گرفته شده‌اند.



شکل ۳-۲: حضور کارت در شرایط نورپردازی متفاوت

- تصاویر دارای زمینه‌های متنوع هستند که در عکس‌های موجود در دیگر بخش‌ها قابل مشاهده هستند.
- حالت‌های مختلف نگه داشتن کارت در دست انسان در مجموعه داده وجود دارد.



شکل ۳-۳: حالت‌های مختلف نگهداری کارت در دست

- فیلم برداری از کارت ها به وسیله دوربین های مختلف و با کیفیت های متفاوت (وبکم، دوربین جلو موبایل، دوربین اصلی موبایل) انجام شده است.



شکل ۳-۴: تصاویر کارت با دوربین‌های مختلف

- داده‌های منفی نیز جهت آموزش بهتر مدل جمع آوری شده است.



شکل ۳-۵: نمونه داده‌های منفی جمع‌آوری شده

۳-۱-۲- برچسب‌زنی داده‌ها

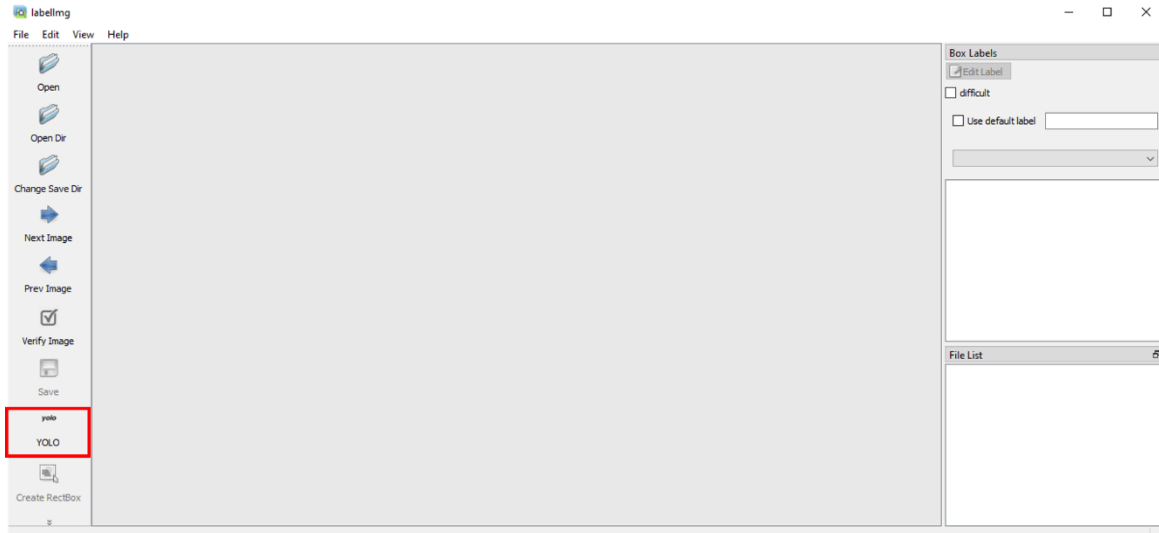
برچسب‌زنی داده‌ها برای آموزش شبکه با استفاده از نرم افزار ²⁶LabelImg انجام شده است. در یک محیط مجازی²⁷ پایتون، می‌توان با استفاده از دستورات زیر، این برنامه را نصب و اجرا کرد:

```
`pip install labelimg`  
`labelimg`
```

²⁶ <https://github.com/tzutalin/labelImg>

²⁷ Virtual Environment

شمای کلی محیط نرم افزار LabelImg در شکل ۳-۶ نشان داده شده است.



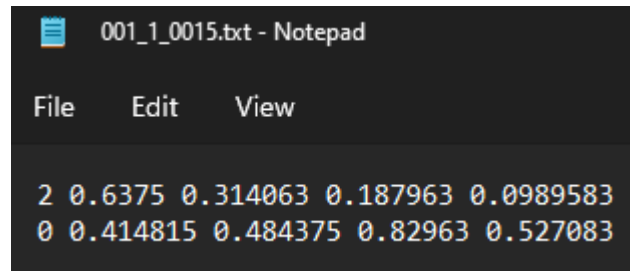
شکل ۳-۶: شمای نرم افزار *labelimg*

در نرم افزار، فرمت ذخیره سازی برچسبها را YOLO قرار می دهیم. این مورد با مستطیل قرمز در شکل ۳-۶ نشان داده شده است. برچسب زنی با کشیدن یک جعبه دور شیء مورد نظر اجرا می شود. هر برچسب یک فایل متنی است که به ازای هر شیء در آن، یک خط مانند زیر در آن فایل ذخیره می شود.

Class x_center y_center width height

مقادیر با کاراکتر فاصله از یکدیگر جدا شده اند. متغیر class، نوع شیء مورد نظر در ناحیه مورد علاقه^{۲۸} را نشان می دهد. کلاسها به صورت عددی در برچسب ذخیره می شوند. یک نمونه از برچسب یک تصویر در شکل ۳-۷ آورده شده است.

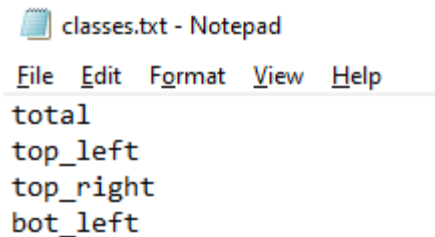
²⁸ Region of Interest



```
001_1_0015.txt - Notepad
File Edit View
2 0.6375 0.314063 0.187963 0.0989583
0 0.414815 0.484375 0.82963 0.527083
```

شکل ۳-۷: نمونه‌ای از برچسب یک تصویر با فرمت *YOLO*

برای برچسب زنی تصاویر در این نرم افزار، نیاز است که یک فایل متنی به نام `classes.txt` داشته باشیم که هر خط آن نام یکی از کلاس‌ها می‌باشد. ترتیب این اسامی مهم است، زیرا در برچسب‌ها بر اساس ترتیب این کلاس‌ها عددگذاری انجام می‌شود. با توجه به این موضوع که تشخیص کارت ملی را یک مسئله چهار کلاسه فرموله کردیم، فایل `classes.txt` باید مانند شکل ۳-۸ ایجاد شود و در پوشه اصلی مجموعه داده قرار بگیرد.



```
classes.txt - Notepad
File Edit Format View Help
total
top_left
top_right
bot_left
```

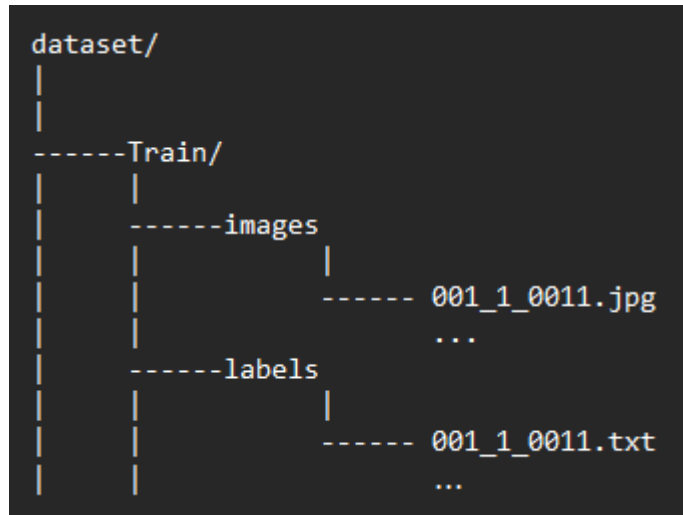
شکل ۳-۸: فایل `classes.txt`

مقادیر `x_center` و `y_center`، مختصات مرکز جعبه دربرگیرنده شیء را ذخیره می‌کنند. همچنین، مقادیر `width` و `height` نیز طول و عرض جعبه را نگهداری می‌کنند. در نتیجه، با استفاده از این پنج مقدار، می‌توان جعبه دربرگیرنده و نوع کلاس آن را در کنار هم ذخیره کرد تا شبکه بر اساس آنها آموزش ببیند.

برای برچسب‌زنی داده‌ها کفایت پوشه تصاویر را درون نرم‌افزار بارگذاری کنیم. با اتمام برچسب‌زنی، به ازای هر تصویری که در آن شیئی وجود داشته باشد، یک فایل متنی متناظر وجود خواهد داشت و اگر

در یک تصویر هیچ شیئی حضور نداشته باشد (مانند داده‌های منفی)، نباید فایل متنی وجود داشته باشد، چرا که در غیر این صورت با مواجهه فایل خالی، آموزش شبکه دچار اختلال خواهد شد.

در شکل ۳-۹، یک نمای کلی از نحوه پوشه‌بندی مجموعه داده آموزشی نشان داده شده است. این ساختار برای مجموعه داده آزمون نیز مشابه می‌باشد.



شکل ۳-۹: ساختار پوشه‌بندی مجموعه داده آموزشی

به جهت برچسب‌زنی کامل داده‌ها، نیاز است که یک مدل ابتدایی روی ۱۰۰۰ تصویر انتخاب شده آموزش ببیند. این ۱۰۰۰ داده به صورت دستی برچسب خورده‌اند. جزئیات آموزش مدل در بخش ۳-۲-۳ بررسی شده است. بعد از آموزش این مدل ابتدایی، یکبار آن را بر تمامی داده‌های مجموعه داده اجرا می‌کنیم. نتایج استنتاج این مدل فرمتی مشابه با فرمت برچسب YOLO دارد، به همین دلیل باقی مجموعه داده، به صورت خودکار برچسب می‌خورند. با توجه به اینکه مدل ابتدایی بسیار ساده است، نیاز است که برچسب‌های تمام داده‌ها به صورت دستی بررسی شوند. بعد از اطمینان از صحت برچسب خوردن تمامی داده‌ها، آموزش جهت ایجاد مدل نهایی آغاز می‌گردد.

۳-۲- آموزش مدل

۳-۲-۱ یادگیری انتقالی

یادگیری انتقالی^{۲۹} یک روش موثر در آموزش مدل‌های یادگیری عمیق است که از مدل‌های از پیش آموزش دیده برای حل مسائل جدید استفاده می‌کند. در این روش، مدل‌هایی که قبلاً بر روی مجموعه داده‌های بزرگ و متنوع آموزش دیده‌اند، به عنوان نقطه شروع برای آموزش مدل‌های جدید به کار می‌روند. لایه‌های اولیه و میانی این مدل‌های از پیش آموزش دیده شده، برای استخراج ویژگی‌های عمومی اشیاء استفاده می‌شوند. برای ویژگی‌های خاص هر شیء، نیاز است که لایه‌های انتهایی این شبکه‌های عصبی مورد آموزش مجدد بر روی همان داده‌های خاص قرار بگیرند تا ویژگی‌های منحصر به فرد اشیاء جدید آموخته شود. این فرآیند تنظیم دقیق نامیده می‌شود. این تکنیک به طور قابل توجهی زمان آموزش و نیاز به داده‌های گسترده را کاهش می‌دهد و دقت مدل نهایی را افزایش می‌دهد. YOLOv5 یکی از شبکه‌هایی است که از یادگیری انتقالی بهره می‌برد و دارای مدل‌های از پیش آموزش دیده متنوعی برای تسهیل و تسریع فرآیند آموزش است.

۳-۲-۲ مدل‌های از پیش آموخته شده YOLOv5

YOLOv5 شامل چندین مدل یا وزن از پیش آموخته شده است که هر کدام برای نیازها و کاربردهای مختلف بهینه شده‌اند. مدل‌های اصلی YOLOv5 شامل YOLOv5s، YOLOv5m، YOLOv5l و YOLOv5x هستند. مدل YOLOv5s سریع‌ترین مدل و مدل اصلی است، در حالی که YOLOv5x بزرگ‌ترین و دقیق‌ترین مدل به شمار می‌آید. علاوه بر این مدل‌ها، مدل YOLOv5n

²⁹ Transfer Learning

نیز وجود دارد که به عنوان نسخه‌ای فوق‌العاده سبک و سریع برای کاربردهایی که نیاز به محاسبات کم و سرعت بالا دارند، طراحی شده است.

برای پیاده‌سازی و استقرار مدل‌های YOLOv5 در مرورگر و اجرای آنها در سمت کاربر، انتخاب مدل‌های ساده‌تر و سبک‌تر مانند YOLOv5s و YOLOv5n اهمیت ویژه‌ای دارد. این مدل‌ها با حجم کمتر و سرعت بیشتر، امکان اجرای کارآمد در محیط‌های با محدودیت منابع مانند مرورگرهای وب را فراهم می‌کنند. با توجه به این موضوع، دو وزن YOLOv5s و YOLOv5n برای آموزش در این پروژه انتخاب شده‌اند.

۳-۲-۳- فرآیند آموزش

در این پروژه، آموزش شبکه دو بار انجام شده است. بار اول، تعداد ۱۰۰۰ تصویر که به صورت دستی برچسب زده شده بودند، مورد آموزش قرار گرفتند. در ادامه با برچسب خوردن تمامی داده‌ها، آموزش اصلی اجرا می‌گردد. برای آموزش شبکه، نیاز به سیستمی مجهز به پردازنده گرافیکی^{۳۰} می‌باشد. پردازنده‌های گرافیکی به دلیل داشتن تعداد بسیار زیادی نخ^{۳۱} در قالب تار^{۳۲}های متعدد، امکان پردازش به مراتب سریع‌تر از پردازنده‌های اصلی^{۳۳} را فراهم می‌کنند. برای استفاده از قابلیت‌های پردازنده گرافیکی، نیاز به نصب لوازم توسعه^{۳۴} برنامه نویسی به زبان CUDA برای اجرا روی پردازنده‌های Nvidia می‌باشد. پس از نصب این لوازم (به کمک دستور nvcc)، می‌توانیم نسخه همگام با CUDA کتابخانه PyTorch را نیز نصب کنیم. با نصب این نسخه، کتابخانه PyTorch می‌تواند با پردازنده گرافیکی ارتباط برقرار کند و پردازش‌های مورد نیاز برای آموزش شبکه را به صورت موازی اجرا کند. نسخه همگام با پردازنده گرافیکی این کتابخانه در

³⁰ GPU

³¹ Thread

³² Warp

³³ CPU

³⁴ Kit

فایل نیازمندی‌های YOLOv5 موجود است. این قابلیت باعث می‌شود زمان آموزش به‌طور قابل توجهی کوتاه‌تر از پردازش با پردازنده اصلی شود. پردازنده گرافیکی استفاده شده در این پژوهش پردازنده گرافیکی Nvidia با حافظه ۸ گیگابایت بوده است.

برای شروع فرآیند آموزش، از دستور زیر استفاده می‌شود:

```
`python train.py --img 320 --batch 64 --epochs 10 --data dataset.yaml --weights yolov5n.pt`
```

این دستور شامل چندین گزینه است که هر کدام نقش مهمی در تنظیمات آموزش دارند:

- `img 320`: این پارامتر اندازه تصاویر ورودی به مدل را تعیین می‌کند. در اینجا، اندازه تصاویر به ۳۲۰ در ۳۲۰ پیکسل تنظیم شده است که تعادلی بین دقت و سرعت پردازش برقرار می‌کند.
- `batch 64`: این پارامتر اندازه دسته^{۳۵}ها را تعیین می‌کند که نشان‌دهنده تعداد نمونه‌هایی است که در هر تکرار به مدل ارائه می‌شود. اندازه دسته ۶۴ انتخاب شده است که به تعادل بین استفاده از حافظه و کارایی آموزش کمک می‌کند. این مقدار با توجه به میزان حافظه GPU متفاوت است.
- `epochs 10`: این پارامتر تعداد دوره‌های تکرار الگوریتم را تعیین می‌کند. در اینجا، مدل به مدت ۱۰ دوره آموزش داده می‌شود که به مدل امکان می‌دهد تا الگوهای موجود در داده‌ها را به‌خوبی یاد بگیرد.
- `data dataset.yaml`: این پارامتر مسیر فایل YAML داده‌ها را مشخص می‌کند که شامل اطلاعاتی در مورد مجموعه داده، نظیر لیست کلاس‌ها و مسیر تصاویر و برچسب‌های هر کدام از مجموعه داده‌های آموزشی و آزمون است. این فایل تنظیمات مربوط به مجموعه داده را فراهم می‌کند و به برنامه آموزش کمک می‌کند تا داده‌های صحیح را برای آموزش و ارزیابی مدل استفاده کند.

³⁵ Batch

- `weights yolov5n.pt`: این پارامتر مدل ابتدایی جهت شروع آموزش را مشخص می‌کند. در این پروژه، از دو مدل از پیش آموخته شده `yolov5n` و `yolov5s` استفاده می‌شود. با اتمام آموزش، مدل نهایی در پوشه‌ای که در خروجی دستور آموزش مشخص است ذخیره می‌شود و می‌توان از آن برای آزمون و اجرا در برنامه‌های دیگر استفاده کرد. استفاده از این پارامترها و تنظیمات مناسب به ما این امکان را می‌دهد تا فرآیند آموزش را به‌طور بهینه انجام دهیم و مدل‌هایی با دقت و کارایی بالا برای تشخیص کارت ملی ایجاد کنیم. مدل‌های آموزش دیده پس از این مرحله آماده می‌شوند تا در مرورگر پیاده‌سازی شده و برای تشخیص کارت ملی به‌کار گرفته شوند.

۳-۳- آزمون

در این بخش، ابتدا مروری بر معیارهای ارزیابی متداول برای شبکه‌های تشخیص اشیا انجام خواهد شد. در ادامه نیز، نتایج اعمال دو شبکه آموزش دیده بر داده‌های آزمون مورد بررسی قرار خواهد گرفت. در انتهای این بخش نیز، مقایسه‌ای بین دو شبکه بر اساس نتایج به‌دست آمده صورت خواهد گرفت و شرایطی که شبکه در آن دچار اشکال در تشخیص می‌شود را مرور خواهیم کرد.

۳-۳-۱- معیارهای ارزیابی

معیارهای ارزیابی برای شبکه‌های تشخیص اشیا در مسئله تشخیص کارت به صورت زیر تعریف می‌شوند:

- دقت^{۳۶}: از بین تمامی تصاویری که در آنها کارت ملی تشخیص و مکان‌یابی شده است، چه درصدی واقعا کارت ملی بودند.

³⁶ Precision

- فراخوانی^{۳۷}: از بین تمامی تصاویری که حاوی کارت ملی بودند، چه تعداد توسط شبکه به درستی تشخیص داده شده و کارت موجود در آنها به درستی مکان‌یابی شده است.
- امتیاز F1: معیار ارزیابی F1 یک میانگین هماهنگ بین دقت و فراخوانی است که به‌طور همزمان به تعادل بین نرخ تشخیص صحیح و نرخ مثبت‌های کاذب^{۳۸} توجه دارد. تعریف این معیار در شکل ۳-۱۰ آورده شده است. هرچه معیار F1 به یک نزدیک‌تر باشد، به این معنی است که مصالحه بین دقت و فراخوانی به نحو بهتری برقرار است و هر دو مقدار بالا هستند.

$$F1 \text{ Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

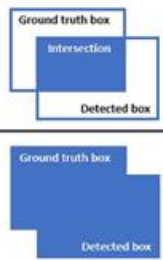
شکل ۳-۱۰: تعریف امتیاز *F1*

- اشتراک نسبت به اجتماع (IoU)^{۳۹}: این معیار، دقت مکان‌یابی را مشخص می‌کند و دو جعبه تقریب زده شده توسط الگوریتم و جعبه اصلی آن (برچسب داده‌های آزمون) را با یکدیگر مقایسه می‌کند. به عبارت دیگر، این معیار میزان همپوشانی دو جعبه را محاسبه می‌کند. این معیار در شکل ۳-۱۱ تعریف شده است. در ارزیابی‌ها، چنانچه میزان IoU از ۰.۵ بیشتر باشد، یک پیش‌بینی درست و در غیر این صورت یک پیش‌بینی نادرست در نظر گرفته می‌شود. لازم به ذکر است که این حد آستانه قابل تنظیم می‌باشد، هرچه مقدار آن بیشتر باشد، سخت‌گیری بیشتری در صحیح بودن پیش‌بینی اعمال می‌شود.

³⁷ Recall

³⁸ False Positives

³⁹ Intersection Over Union

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{\text{Intersection}}{\text{Ground truth box} \cup \text{Detected box}}$$


شکل ۳-۱۱: تعریف معیار IoU

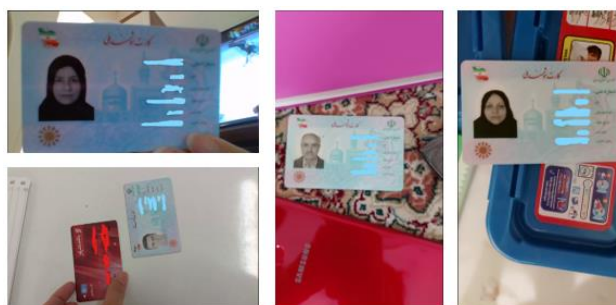
هر چه مقدار این کسر به عدد یک نزدیکتر باشد، مکان‌یابی بهتر انجام شده است و مدل دقیق‌تر است.

- $mAP50$: سطح زیر نمودار منحنی دقت – فراخوانی یک معیار رایج در ارزیابی الگوریتم‌های دسته‌بندی است که به آن دقت میانگین (AP)^{۴۰} گفته می‌شود. هرچه این مقدار به یک نزدیک‌تر باشد، دسته‌بند به حالت ایده‌آل و بدون خطا نزدیک‌تر می‌شود. $mAP50$ ، میانگین AP را برای تمام کلاس‌ها و $IoU=0.5$ محاسبه می‌کند.
- $mAP50-95$: میانگین AP را برای $IoU = [0.5:0.05:0.95]$ محاسبه می‌کند.

۳-۳-۲ - نتایج اعمال شبکه‌های آموزش دیده بر روی داده‌های آزمون

داده‌های آزمون شامل ۸۲۲۵ فریم از کارت‌های ملی افرادی که در داده‌های آموزشی حضور نداشتند، در شرایط محیطی مختلف جمع‌آوری شدند. اطلاعات مجموعه داده آزمون در جدول ۳-۲ و نمونه‌هایی از داده‌های آزمون در شکل ۳-۱۲ آورده شده است. شبکه‌های آموزش دیده بر روی آنها مورد آزمون قرار گرفتند. نتایج به دست آمده برای هر شبکه را به صورت جداگانه بررسی می‌کنیم.

⁴⁰ Average Precision



شکل ۳-۱۲: نمونه داده‌های آزمون

در ابتدا، نتایج آزمون برای مدل yolov5s بررسی می‌شود. این نتایج در جدول ۳-۳ خلاصه شده است.

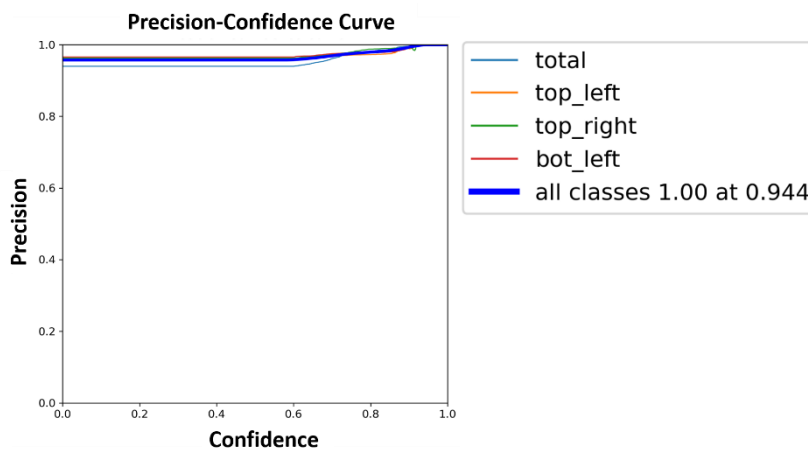
جدول ۳-۳ نتایج آزمون برای مدل yolov5s

کلاس	تعداد تصویر	تعداد نمونه‌ها	دقت	فراخوانی	mAP50	mAP50-95
All	8225	29289	0.963	0.925	0.956	0.871
Total	8225	7502	0.949	0.933	0.963	0.909
Top_Left	8225	7360	0.965	0.926	0.954	0.846
Top_Right	8225	7173	0.968	0.928	0.956	0.88
Bottom_Left	8225	7254	0.97	0.912	0.95	0.85

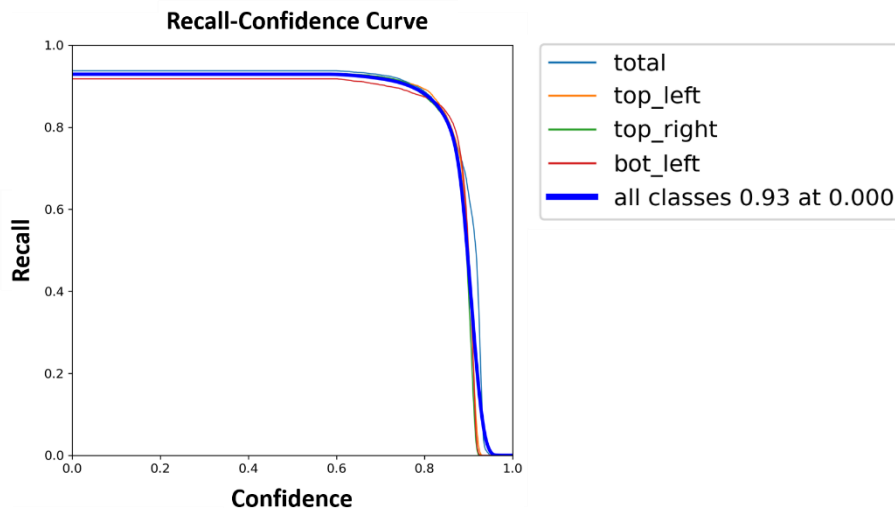
نمودار دقت به ازای مقادیر مختلف اطمینان^{۴۱} در شکل ۳-۱۳ آورده شده است. همانطور که مشاهده می‌شود، با افزایش اطمینان، میزان دقت افزایش می‌یابد، چراکه با اطمینان بالاتر، اشیا تشخیص داده شده واقعاً اشیا مدنظر هستند. به طور مشابه، نمودار فراخوانی به ازای مقادیر مختلف اطمینان در شکل ۳-

⁴¹ Confidence

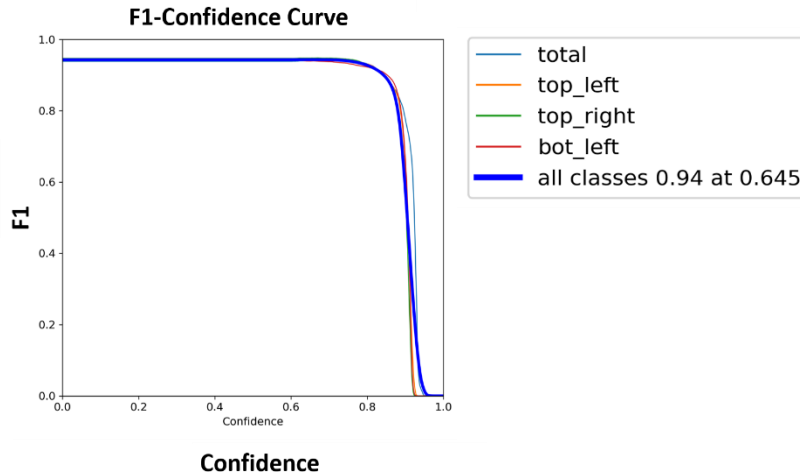
۱۴ آورده شده است. برخلاف حالت قبلی، با افزایش اطمینان، میزان فراخوانی کاهش پیدا می‌کند، چرا که با افزایش اطمینان، تعدادی از اشیا واقعی به اشتباه نادیده گرفته می‌شوند. از آنجا که مقادیر دقت و فراخوانی به تنهایی نمی‌توانند دید درستی در عملکرد شبکه انتقال دهند، نیاز به معیاری داریم که هر دو این معیارها را بهم ربط و مورد بررسی قرار دهد. امتیاز F1 از ادغام این دو معیار به دست می‌آید. نمودار تغییرات F1 نسبت به اطمینان در شکل ۳-۱۵ نشان داده شده است. طبق این نمودار، مقادیر اطمینان بین ۰.۶ تا ۰.۸ بهترین عملکرد و مصالحه را دارد.



شکل ۳-۱۳: نمودار دقت – اطمینان برای مدل *yolov5s*



شکل ۳-۱۴: نمودار فراخوانی – اطمینان برای مدل *yolov5s*



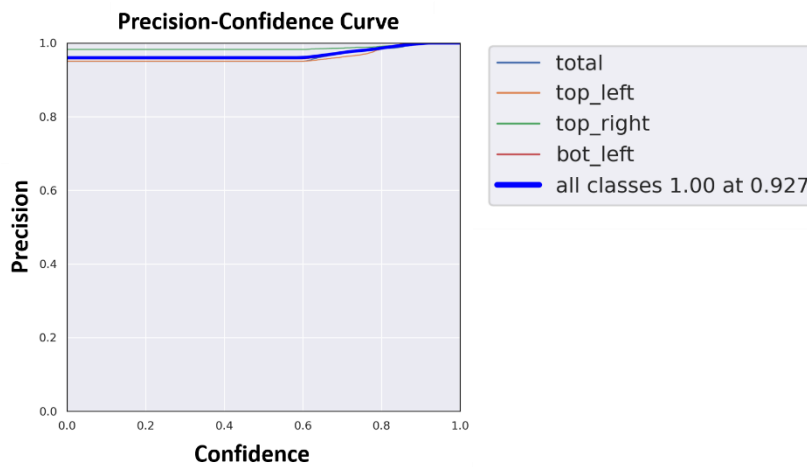
شکل ۳-۱۵: نمودار امتیاز $F1$ – اطمینان برای مدل *yolov5s*

اکنون، نتایج آزمون برای مدل *yolov5n* بررسی می‌شود. مشابه قبل، نتایج آزمون بر روی این مدل در جدول ۳-۴ آورده شده است.

جدول ۳-۴: نتایج آزمون برای مدل *yolov5n*

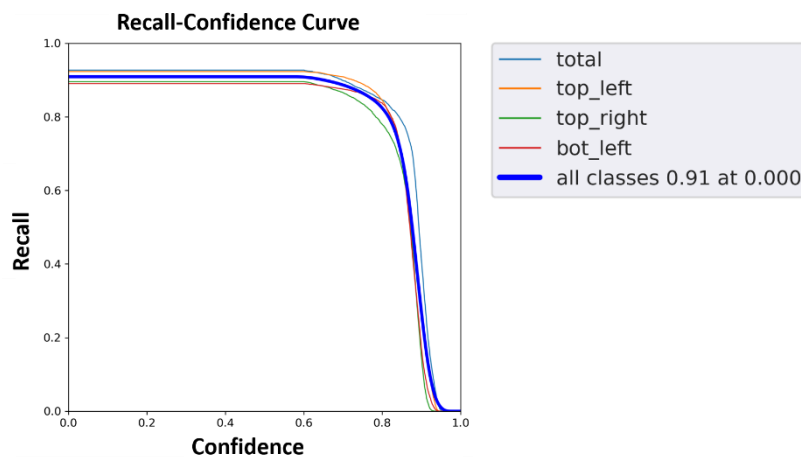
کلاس	تعداد تصویر	تعداد نمونه‌ها	دقت	فراخوانی	mAP50	mAP50-95
All	8225	29289	0.96	0.909	0.949	0.848
Total	8225	7502	0.95	0.927	0.957	0.879
Top_Left	8225	7360	0.951	0.923	0.954	0.85
Top_Right	8225	7173	0.983	0.896	0.943	0.812
Bottom_Left	8225	7254	0.957	0.89	0.94	0.851

نمودار دقت به ازای مقادیر مختلف اطمینان در شکل ۳-۱۶ آورده شده است. همانطور که مشاهده می‌شود، در این نمودار نیز مانند شکل ۳-۱۲ با افزایش اطمینان، میزان دقت بالا می‌رود.



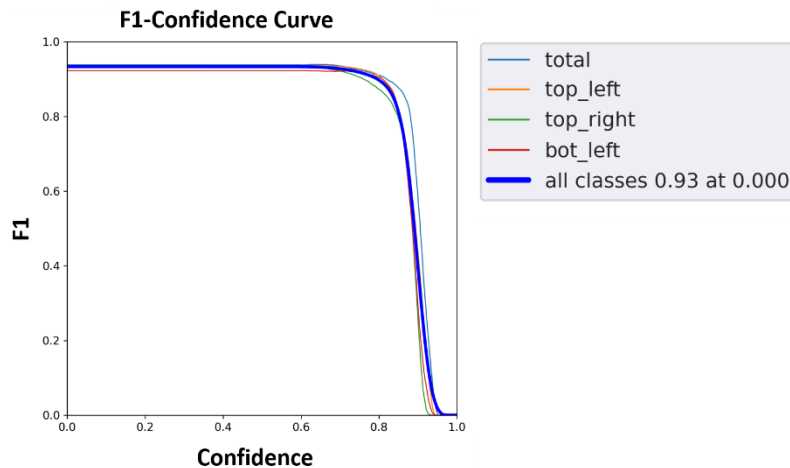
شکل ۳-۱۶: نمودار دقت – اطمینان برای مدل *yolov5n*

شکل ۳-۱۷ نمودار فراخوانی بر اساس مقادیر مختلف اطمینان را نشان می‌دهد. همانطور که مشاهده می‌شود، با افزایش اطمینان، مقدار فراخوانی به صفر نزدیک می‌شود.



شکل ۳-۱۷: نمودار فراخوانی – اطمینان برای مدل *yolov5n*

نمودار تغییرات امتیاز F1 نسبت به اطمینان برای این مدل نیز در شکل ۳-۱۸ آورده شده است.



شکل ۳-۱۸: نمودار امتیاز $F1$ - اطمینان برای مدل *yolov5n*

همانطور که در شکل ۳-۱۸ نیز مشاهده می‌شود، مقدار اطمینان قابل قبول برای این مدل نیز تا ۰.۸ قابل قبول است و می‌توان مقادیر اطمینان بین ۰.۶ تا ۰.۸ را نیز برای این مدل تعمیم داد.

۳-۳-۳ - مقایسه و تحلیل

هر یک از دو مدل به تنهایی، عملکرد قابل قبولی برای تشخیص کارت ملی و نمادهای آن ارائه می‌دهند. وجود مقادیر اطمینان بیش از ۹۶ درصد و مقدار فراخوانی بالای ۹۰ درصد در هر مدل، نشان از یادگیری قابل قبول و همچنین توانایی تعمیم بالای مدل‌ها دارد.

همانطور که مشاهده می‌شود، دو مدل عملکرد نزدیک و تقریباً مشابهی بر روی داده‌های آزمون داشتند. مدل *yolov5s* از نظر دقت، بازیابی، و امتیاز $F1$ ، وضعیت بهتری دارد. چرا که این مدل دارای پارامترها و لایه‌های بیشتری نسبت به مدل *yolov5n* دارد.

به نظر می‌رسد دلیل اصلی این شباهت و دقت بالا، وجود داده‌های آموزشی در حجم زیاد و در شرایط مختلف می‌باشد. یادگیری عمیق، برای آموزش مناسب مدل‌ها، نیاز به جمع‌آوری و حضور انبوهی از داده‌ها دارد. با زیاد بودن داده‌های آموزشی، مدل‌ها ویژگی‌های اشیا مد نظر را به خوبی یاد می‌گیرند و علاوه بر این، می‌توانند ویژگی‌های آموخته شده را به داده‌های جدید تعمیم دهند.

در ادامه، برخی از تصاویری که مدل‌های آموزش‌دیده دچار اشتباه شدند را بررسی می‌کنیم. شبکه YOLOv5، به صورت مستقیم لیستی از این تصاویر را در اختیار کاربر قرار نمی‌دهد. به جهت پیدا کردن این تصاویر، در کد کوتاه `val.py`^{۴۲} که برای اجرای ارزیابی استفاده شد، تغییراتی داده شد تا اسامی تصاویری که دارای برچسب بودند، اما شبکه نتوانسته بود کلاسی برای آن پیش‌بینی کند، در خروجی نمایش داده شود. حال به دو دسته کلی از حالت‌هایی که باعث عدم تشخیص در تصاویر شده است می‌پردازیم.



شکل ۳-۱۹: دسته اول از تشخیص‌های ناموفق

در شکل ۳-۱۹، نمونه‌ای از دسته اول تشخیص‌های ناموفق قابل مشاهده است. علت اصلی تشخیص نادرست در این دسته تصاویر، زاویه بیش از حد عکس کارت ملی می‌باشد. در این گروه از تصاویر، ویژگی‌های اصلی کارت ملی قابل مشاهده نیست و به همین دلیل شبکه قادر به تشخیص شیء نشده است.

⁴² Script



شکل ۳-۲۰: دسته دوم از تشخیص‌های ناموفق

در دسته دوم که در شکل ۳-۲۰ آورده شده است، کارت ملی به‌طور کامل قابل مشاهده است، اما شبکه هیچ شئی در تصویر تشخیص نداده است. علت این امر، شدت نور بسیار زیاد از سمت پشت کارت می‌باشد.

با این تفاسیر، عملکرد شبکه هنگام چرخش بیش از حد و شدت نور بسیار زیاد از پشت دچار اختلال می‌شود. در بقیه حالت‌ها، نظیر وجود کارت‌های منفی، دوران‌های معقول و... شبکه توانسته است به‌خوبی استنتاج کند و اشیا را تشخیص دهد.

۳-۴- نتیجه‌گیری

در این فصل، با سامانه‌های مرتبط و شبکه‌های تشخیص اشیا به‌خصوص شبکه عصبی YOLOv5 آشنا شدیم. نحوه جمع‌آوری داده مورد نیاز برای تشخیص کارت ملی و شیوه برچسب زنی را مرور کردیم. در ادامه نیز فرآیند آموزش و نتایج مدل‌های آموزش دیده را بر روی داده‌های آزمون بررسی کردیم.

همانطور که مشاهده شد، مدل‌ها بر روی داده‌های آزمون رقابت نزدیکی داشتند. مدل yolo5n با اینکه در ساختار خود از لایه‌ها و پارامترهای کمتری تشکیل شده است، اما توانست عملکردی در حدی مدل پیچیده‌تر خود، مدل yolo5s داشته باشد. هر دو مدل نیز دقت بسیار بالایی در بازشناسی تصاویر کارت ملی از خود نشان دادند. با توجه به اینکه دقت شبکه YOLOv5 در آزمون توانست به حد قابل قبول و مناسب برسد و انتظارات را برآورده کند، نیازی به آموزش دیگر شبکه‌های تشخیص اشیا دیده نمی‌شود و به همین شبکه اکتفا می‌شود.

علت بالا بودن دقت و همچنین عملکرد نزدیک دو مدل، ناشی از زیاد بودن حجم داده‌های آموزشی عنوان شد. با زیاد بودن حجم داده، مدل‌ها توانستند ویژگی شیء مدنظر را به خوبی یاد بگیرند و آن را نیز به خوبی تعمیم دهند. سوال اصلی اینجاست که با نزدیک بودن عملکرد دو مدل بر داده‌های آزمون، آیا عملکرد دو مدل در پیاده‌سازی سامانه و اجرای واسط کاربری نیز مشابه می‌باشد یا خیر؟ پاسخ این سوال در فصل آینده بررسی شده است. فصل چهارم پیاده‌سازی واسط کاربری و عملکرد دو مدل را مورد تحلیل قرار می‌دهد.

فصل چهارم

واسط کاربری

در این فصل به بررسی واسط کاربری پروژه و استقرار آن در مرورگر می‌پردازیم. واسط کاربری بخش اساسی از هر سیستم تشخیص و شناسایی مبتنی بر هوش مصنوعی است که تجربه کاربر را شکل می‌دهد و امکان تعامل کاربر با سیستم را فراهم می‌کند.

همانطور که پیشتر هم عنوان شده است، نیاز است که پردازش اولیه کارت ملی شامل تشخیص و اصلاح زاویه آن در سمت کاربر و بدون نیاز به ارسال به سمت سرور انجام شود. با انجام این کار، از ارسال انبوهی از داده‌ها به سمت سرور جلوگیری می‌شود و از بار پردازشی آن کاسته خواهد شد. به همین جهت، واسعه کاربری بر مرورگر مستقر می‌شود تا کاربران قادر باشند با هر دستگاهی از خدمات این سامانه بهره‌مند شوند.

برای اجرای مدل آموزش دیده‌شده در مرورگر، از کتابخانه TensorFlow.js [۱۰] استفاده شده است. TensorFlow.js یک کتابخانه قدرتمند برای اجرای مدل‌های یادگیری ماشین در مرورگر است که امکان بارگذاری و اجرای مدل‌های TensorFlow [۱۱] را به زبان جاوا اسکریپت فراهم می‌کند. با این حال، مدل YOLOv5 به صورت اولیه برای استفاده در TensorFlow طراحی نشده است و نیاز به صادر کردن آن به قالب مناسب دارد.

YOLOv5 قابلیت صدور مدل‌های خود را به انواع مختلف فراهم کرده است که یکی از این قالب‌ها، قالب tfjs می‌باشد. با استفاده از این قابلیت، می‌توانیم مدل آموزش دیده را به فرمت tfjs صادر کنیم و سپس آن را در کد جاوا اسکریپت بارگذاری کنیم. این فرآیند شامل تبدیل مدل به فرمت مناسب، ذخیره آن و بارگذاری مجدد در محیط مرورگر است.

پس از بارگذاری مدل، مرحله بعدی دریافت جریان دوربین کاربر است. استفاده از API‌های مرورگر اجازه دسترسی به دوربین کاربر را می‌دهد و تصاویر زنده از آن دریافت می‌شود. هر فریمی که از وبکم دریافت می‌شود، به مدل بارگذاری شده ارسال می‌شود تا تشخیص‌های لازم انجام گیرد. این فرآیند به صورت بی‌درنگ انجام می‌شود و نتایج تشخیص به کاربر نمایش داده می‌شود.

در این فصل به تفصیل مراحل مختلف پیاده‌سازی واسط کاربری، از بارگذاری مدل تا دریافت و پردازش جریان^۱ دوربین و نمایش نتایج، پرداخته خواهد شد.

^۱ Stream

۴-۱- صدور مدل‌های آموزش دیده شده به فرمت مناسب

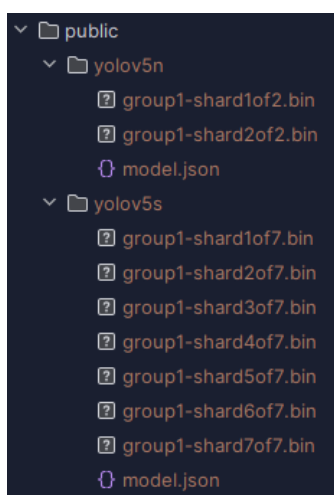
شبکه YOLOv5، مدل‌ها را با فرمت PyTorch (با پسوند pt) آموزش می‌دهد. این مدل‌ها را نمی‌توان در مرورگر (و زبان جاوا اسکریپت) به کار برد. به همین دلیل، نیاز است که مدل‌ها پیش از استقرار در مرورگر به فرمت مناسب صادر شده باشند.

TensorFlow.js [۱۰] کتابخانه‌ای به زبان جاوا اسکریپت است که برای استفاده و استقرار مدل‌های یادگیری ماشین به کار برده می‌شود. به وسیله این کتابخانه، می‌توانیم مدل مورد نظر خود را در مرورگر بارگذاری کنیم و داده‌های ورودی را به وسیله آن استنتاج کنیم.

خوشبختانه، YOLOv5 قابلیت صدور مدل‌های خود را به انواع مختلفی از فرمت‌های دیگر نظیر onnx، openvino، و tfjs فراهم کرده است. به جهت صدور مدل، دستور زیر اجرا شده است:

```
`python export.py --weights yolov5s.pt --include tfjs`
```

با اجرای این دستور، تعدادی فایل باینری (با پسوند bin) و یک فایل json ایجاد خواهد شد. تعداد فایل‌های باینری بسته به حجم مدل اولیه، متفاوت خواهد بود. برای مثال، مدل yolov5n به دو فایل باینری و مدل yolov5s به هفت فایل باینری صادر شده‌اند. به همین دلیل، مدل اول، در نهایت حجمی معادل هفت مگابایت و مدل دوم حجمی معادل ۲۷ مگابایت خواهد گرفت. با توجه به این موضوع که برنامه قرار است در سمت کاربر اجرا شود، این موضوع می‌تواند یک عامل تاثیرگذار بر تجربه کاربری باشد، چرا که مدل با حجم بالا، می‌تواند فرآیند تشخیص را با کندی مواجه کند.



شکل ۴-۱: فایل‌های باینری مدل‌های صادر شده

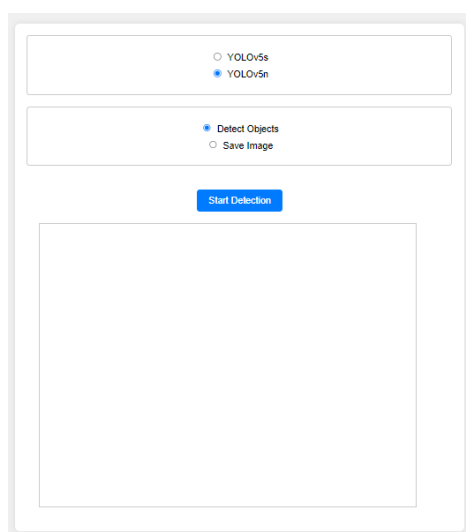
۴-۲- توسعه واسط کاربری

۴-۲-۱- معرفی

همانطور که پیشتر هم عنوان شد، مدل آموزش دیده به جهت انجام پیش‌پردازش‌ها باید در مرورگر و سمت کاربر اجرا شود، به همین دلیل برای توسعه واسط کاربری، از زبان جاوا اسکریپت و یکی از چارچوب‌های آن به نام React استفاده شده است.

React یک کتابخانه قدرتمند و محبوب جاوا اسکریپت است که برای ساخت رابط‌های کاربری تعاملی و پویا بسیار کارآمد است. React، قابلیت ایجاد عناصر گرافیکی مجزا و قابل استفاده مجدد را فراهم می‌کند که این امر، باعث نگهداری و توسعه ساده‌تر کد می‌شود. نمایی از واسط کاربری آغازین در شکل ۴-۲ آورده شده است.

² Framework



شکل ۴-۲: نمای آغازین واسط کاربری

یکی از قابلیت‌های مهم چارچوب React، وجود متغیر از نوع State می‌باشد. به وسیله این متغیرها، کاربر می‌تواند مقادیر متغیرها را تغییر دهد. از این ویژگی، در این جهت استفاده شده که واسط به کاربر اجازه تغییر مدل بارگذاری شده را بدهد. به همین جهت، کاربر می‌تواند مدل‌های مختلف را تنها با انتخاب یک دکمه امتحان کند. دو مدل آموزش داده شده از این طریق در اختیار کاربر قرار خواهند گرفت.

علاوه بر این، در واسط کاربری، با کمک State، دو حالت^۳ برای رفتار سامانه تعریف شده است:

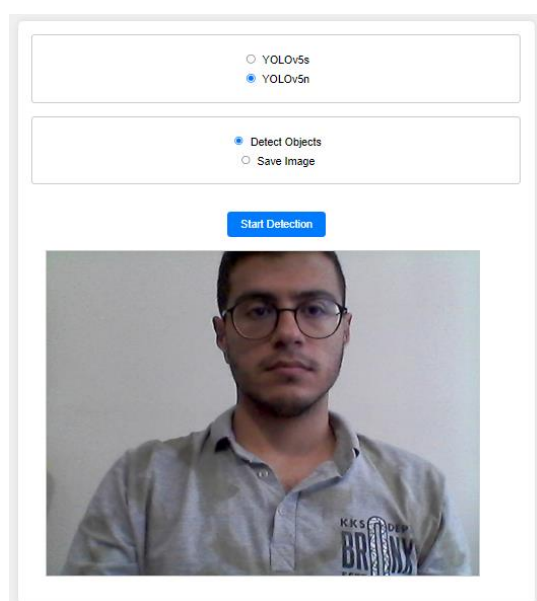
- حالت تشخیص: در این حالت، سامانه تنها کارت ملی و نمادهای آن را مکان‌یابی می‌کند. اصلاح زاویه در این حالت صورت نمی‌گیرد و اشیاء پیدا شده دنبال خواهند شد.
- حالت ذخیره: در این حالت، سامانه علاوه بر مکان‌یابی کارت ملی و نمادها، اصلاح زاویه را نیز انجام می‌دهد.

توضیحات و نحوه کارکرد سامانه در بخش بعدی به تفصیل بررسی خواهد شد.

³ Mode

۴-۲-۲- توضیح اجزای واسط کاربری

در ابتدای توسعه، باید مدل‌هایی را که در قسمت پیش صادر کردیم، در برنامه بارگذاری کنیم. از آنجا که نیاز است این مدل‌ها به سمت کاربر ارسال شوند، باید آنها را در پوشه فایل‌های عمومی پروژه^۴ قرار دهیم. برنامه، مدل‌ها را از روی فایل json تشخیص می‌دهد، زیرا در این فایل، اطلاعاتی در مورد تعداد و مکان فایل‌های باینری مدل وجود دارد. پس تنها نیاز است که مسیر فایل‌های json در برنامه داده شود. همانطور که در شکل ۲-۴ هم مشاهده می‌شود، دکمه Start Detection در مرکز صفحه حضور دارد. با کلیک بر روی این دکمه، برنامه به دوربین کاربر متصل خواهد شد و تصویر دوربین را در مرکز صفحه نمایش خواهد داد. این موضوع در شکل ۳-۴ قابل مشاهده است.



شکل ۳-۴: اتصال سامانه به دوربین کاربر

قطعه کد اتصال به دوربین کاربر در شکل ۴-۴ قابل مشاهده است.

^۴ Public Folder

```

startWebcam():void {
  navigator.mediaDevices.getUserMedia( constraints: { video: true })
    .then(stream: MediaStream => {
      const video: HTMLVideoElement = document.getElementById( "video");
      video.srcObject = stream;
      video.onloadedmetadata = () :void => {
        video.play();
        this.detectObjects();
      };
    })
    .catch(error => {
      console.error("Error accessing webcam:", error);
    });
}

```

شکل ۴-۴: قطعه کد اتصال به دوربین کاربر

با قرار گرفتن کارت ملی در برابر دوربین، سامانه شروع به تشخیص کارت می‌کند. سامانه در هر دو حالت، کارت ملی و نمادهای موجود در تصویر را تشخیص می‌دهد، با این تفاوت که در حالت ذخیره، کاربر ملزم است کارت را به گونه‌ای در برابر دوربین بگیرد که همه نمادهای آن مشخص باشد. اگر چرخش تنها در محور عمود بر کارت صورت پذیرفته باشد، سامانه قادر خواهد بود تا زاویه آن را نیز اصلاح کند.

برای استنتاج و تشخیص کارت، باید یک تصویر به مدل داده شود، به همین جهت نیاز است که دوربین کاربر را به عکس‌های متوالی تبدیل کنیم. برای تبدیل کردن جریان دوربین به فریم‌ها و سپس ارسال آن به مدل، نیاز است که از بازه زمانی^۵ استفاده کنیم. در این سامانه، بازه زمانی به صورت ۳۰ / ۱۰۰۰ میلی ثانیه یا به عبارت دیگر، ۳۰ فریم در ثانیه تعریف شده است، به این معنی که بین دریافت هر فریم از جریان دوربین کاربر، باید یک سی‌ام ثانیه وقفه وجود داشته باشد. با افزایش تعداد گرفتن فریم‌ها در ثانیه، حجم پردازش برنامه بیشتر می‌شود و به همین دلیل تشخیص کارت و به‌روز شدن نمایش دوربین با تاخیر همراه می‌گردد. تابع `detectObjects` که در شکل ۴-۴ استفاده شده است، در شکل ۴-۵ آورده شده است. میزان وقفه زمانی استفاده شده در این شکل نیز قابل مشاهده است.

⁵ Time Interval

```
detectObjects(): void {
  const video: HTMLVideoElement = document.getElementById( "video");
  const canvas: HTMLCanvasElement = document.getElementById( "canvas");
  const ctx = canvas.getContext( "2d");

  const modelWidth = this.state.model.inputs[0].shape[1];
  const modelHeight = this.state.model.inputs[0].shape[2];

  const confidenceThreshold: number = 0.75;
  let objectCount: number = 0;

  let c1_x, c1_y, c2_x, c2_y, c3_x, c3_y;

  const DROP_RATE: number = 10;
  let dropped_frames: number = 0;

  setInterval( handler: () => {...}, timeout: 1000 / 30); // Run object detection at 30
```

شکل ۴-۵: قطعه کد تابع تشخیص اشیا

هر فریم خوانده شده، به مدل ارسال می‌شود. بعد از استنتاج مدل بر روی فریم، نتیجه در قالب یک لیست تودرتو^۶ برگردانده می‌شود. لیست اول، اطلاعات جعبه‌ها، لیست دوم امتیاز اشیا (میزان اطمینان مدل)، لیست سوم اسامی کلاس‌ها و لیست چهارم داده‌های معتبر می‌باشد.

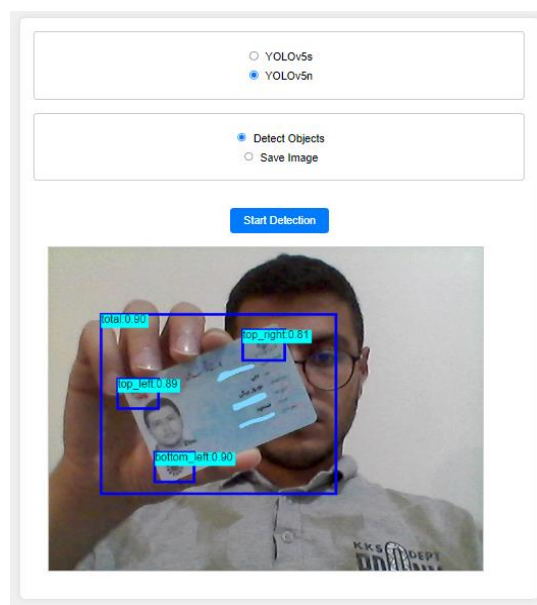
```
const [boxes, scores, classes, valid_detections] = res;
const boxes_data: ... = boxes.dataSync();
const scores_data: ... = scores.dataSync();
const classes_data: ... = classes.dataSync();
const valid_detections_data = valid_detections.dataSync()[0];
```

شکل ۴-۶: لیست تودرتو نتایج

در ادامه لازم است که نتایج تشخیص مدل را در واسط کاربری مشاهده کنیم. برای این کار، باید با استفاده از مختصات جعبه‌های تشخیص داده شده، یک مستطیل رسم کنیم. سپس باید نام آن کلاس و همچنین میزان اطمینان را در قالب یک برچسب، بالای جعبه قرار دهیم. پیمایش بر روی داده‌های معتبر (لیست چهارم) اجرا می‌شود. اگر امتیاز آن داده از یک حد آستانه (در اینجا ۰.۷۵) کمتر باشد، آن شیء نمایش داده نمی‌شود. در غیر این صورت، با استفاده از اطلاعات موجود در دیگر لیست‌ها، مستطیل جعبه دربرگیرنده، نام کلاس، و میزان اطمینان آن رسم می‌شود. همانطور که پیشتر هم عنوان شد، این مرحله،

^۶ Nested List

در هر دو حالت تشخیص و ذخیره اجرا می‌شود. یک نمونه از مکان‌یابی کارت به همراه نمادهای آن در حالت تشخیص در شکل ۷-۴ آورده شده است.



شکل ۷-۴: مکان‌یابی کارت توسط سامانه در حالت تشخیص

برای نمایش فریم دریافت شده از دوربین به کاربر و همچنین رسم جعبه‌ها و نتایج، باید از شیء بوم^۷ در HTML استفاده کنیم. این شیء قابلیت رسم شکل‌های مختلف و نمایش آن‌ها را به ما می‌دهد. هر بوم، متشکل از یک زمینه^۸ دو بعدی می‌باشد که قابلیت رسم تصاویر و اشکال دو بعدی را فراهم می‌کند.

کشیدن فریم بر روی بوم و همچنین رسم جعبه‌ها، به ترتیب با استفاده از متد `drawImage` و `fillRect` شیء زمینه انجام می‌شود. همچنین این شیء دارای صفات^۹ متعددی نظیر فونت، رنگ و... می‌باشد.

^۷ canvas

^۸ Context

^۹ Attributes

پس از رسم جعبه‌ها، اگر سامانه در حالت ذخیره باشد، علاوه بر شروط قبلی (اطمینان بیش از حد آستانه و چرخیدن در محور عمود بر کارت)، نیاز است که شروط زیر نیز جهت اصلاح زاویه و سپس ذخیره آن بررسی شوند:

- نیاز است که ۴ شی تشخیص داده شده باشد.
 - نیاز است که مراکز نمادها درون جعبه شی کارت باشد.
- با صحت این شروط، اصلاح زاویه آغاز می‌شود. برای اصلاح زاویه، نیاز است که مراکز سه نماد کارت ملی یافت شوند. این کار برای بررسی صحت شروط قبلا انجام شده است. سپس، نیاز است که زاویه خط واصل بین مرکز دو نماد سمت چپ و مرکز دو نماد بالا بررسی شود.
- برای بررسی، تانژانت وارون بین مراکز مذکور با استفاده از مختصات X و Y مربوطه محاسبه می‌شوند. ابتدا نمادهای بالای کارت مورد بررسی قرار می‌گیرند. اگر شیب این خط مخالف صفر باشد (تصویر دارای دوران باشد)، باید تصویر را به اندازه همین زاویه (که از تانژانت وارون بدست آمده) بچرخانیم. علامت این دوران توسط خود تابع تانژانت وارون محاسبه شده است.
- در غیر این صورت (زاویه خط بین دو نماد بالا مساوی صفر است)، متوجه می‌شویم که کارت دوران نداشته، اما برای اطمینان، نیاز است که زاویه بین نمادهای سمت چپ را نیز بررسی کنیم، چرا که ممکن است کارت سروته گرفته شده باشد و این در صورتی رخ می‌دهد که شیب خط واصل دو مرکز منفی باشد^{۱۰}.

پس از تشخیص مراکز نمادها و استخراج زاویه دوران، تصویر کارت از بقیه تصویر جدا می‌شود. در این مرحله، با استفاده از مختصات تشخیص داده شده، یک مستطیل با ابعاد جعبه دربرگیرنده ترسیم می‌شود و تصویر کارت از تصویر اصلی برش می‌خورد. تصویر بریده شده باید در یک بوم جدید با زمینه سیاه قرار داده شود.

^{۱۰} همانطور که عنوان شد، اجرای این مرحله تنها به دلیل اطمینان از وضعیت کارت صورت می‌گیرد. سامانه تشخیص کارت ملی، با هدف همکاری کاربر توسعه داده شده است و کاربر مجاز است کارت را تنها تا نود درجه به هر دو طرف بچرخاند.

برای اطمینان از این که تصویر چرخیده شده به طور کامل در بوم جدید جا بگیرد، قطر مستطیل تصویر برش داده شده با استفاده از طول و عرض آن محاسبه می شود. قطر محاسبه شده تعیین کننده ابعاد اولیه بوم جدید است، چرا که اندازه بوم جدید برای دربرگرفتن تصویر چرخانده شده، در بدترین حالت اندازه ای برابر با قطر تصویر بریده شده خواهد گرفت. این بوم جدید به رنگ سیاه پر می شود تا تصویر نهایی حاشیه سیاه داشته باشد.

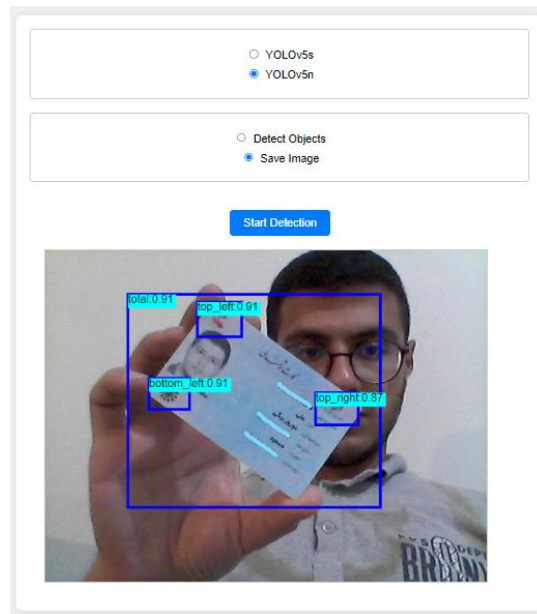
در ادامه، تصویر کارت در مرکز بوم جدید قرار می گیرد و سپس بوم حول مرکز خود به میزان زاویه محاسبه شده چرخانده می شود. بعد از چرخش، تصویر کارت با استفاده از مقادیر انحراف^{۱۱} محاسبه شده، در مرکز بوم جدید قرار می گیرد. این مقادیر برای جاگذاری صحیح تصویر کارت در بوم چرخیده به وسیله اختلاف اندازه بیشینه (قطر جعبه دربرگیرنده) بوم و میزان طول و عرض جعبه دوران یافته محاسبه می شوند، چرا که قطر جعبه دربرگیرنده بیشینه حالت چرخش کارت را در نظر می گیرد و پس از چرخش این مقدار اصلاح می شود. در نهایت، تصویر چرخیده شده با حاشیه های سیاه در مرکز بوم قرار می گیرد که عکس اصلاح شده کارت را به صورت دقیق نمایش می دهد.

پس از اصلاح زاویه، تصویر را به فرمت یک رشته base64 تبدیل می کنیم. استفاده از رشته base64 برای انتقال تصاویر در دنیای وب یک امر رایج است. این رشته می تواند در ادامه برای ذخیره سازی و دیگر کاربردها به یک سرور ارسال شود. رمزگذاری^{۱۲} یک تصویر به رشته Base64 شامل تبدیل داده های باینری تصویر به یک قالب متنی است. این فرآیند بایت های خام فایل تصویر را می گیرد و آنها را به صورت دنباله ای از کاراکترهای ASCII نشان می دهد. الگوریتم رمزگذاری داده های باینری را به تکه های ۶ بیتی تقسیم می کند، هر تکه را به یک کاراکتر متناظر از مجموعه کاراکترهای Base64 که شامل حروف، ارقام و برخی نمادها می شود نگاشت می کند و سپس این کاراکترها را در یک رشته ترکیب می کند. این رمزگذاری تضمین می کند که داده های باینری می توانند به صورت ایمن منتقل یا در قالب های مبتنی بر متن مانند JSON یا XML ذخیره شوند.

¹¹ Offset

¹² Encoding

یک نمونه از تشخیص کارت در حالت ذخیره به همراه اصلاح زاویه آن در شکل ۸-۴ و شکل ۹-۴ آورده شده است. در شکل ۸-۴، کارت تشخیص داده شده توسط سامانه و در شکل ۹-۴ تصویر کارت همراه با اصلاح زاویه قابل مشاهده است.

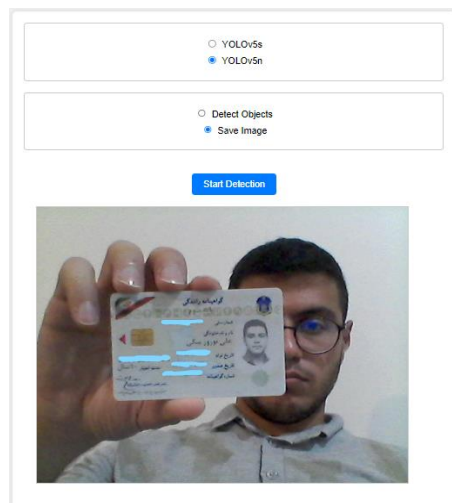


شکل ۸-۴: مکان‌یابی کارت توسط سامانه در حالت ذخیره

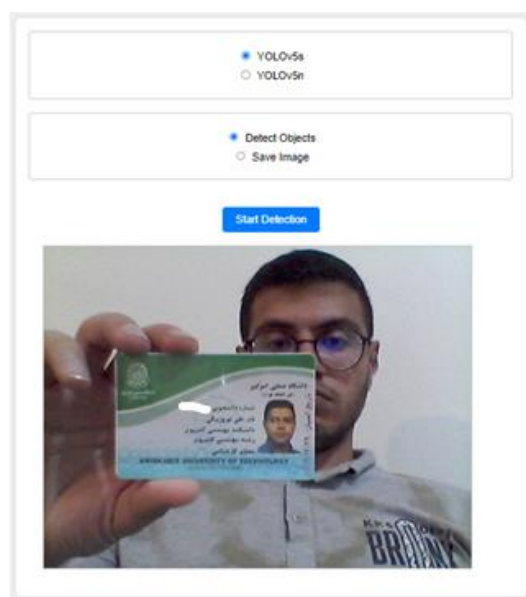


شکل ۹-۴: تصویر کارت ملی به همراه اصلاح زاویه شده

علاوه بر این، یکی دیگر از اهداف سامانه، عدم تشخیص کارت‌های دیگر نظیر گواهینامه و کارت دانشجویی توسط سامانه می‌باشد. این مهم نیز در شکل ۴-۱۰ و شکل ۴-۱۱ نشان داده شده است.



شکل ۴-۱۰: عدم تشخیص گواهینامه رانندگی توسط سامانه



شکل ۴-۱۱: عدم تشخیص کارت دانشجویی توسط سامانه

نکته قابل توجه، افت سرعت سامانه و ایجاد تاخیر در به‌روزرسانی فریم‌ها هنگام استفاده از مدل yolov5s است. همانطور که پیشتر هم اشاره شد، این مدل به دلیل حجم بالایی که دارد، عملکرد سیستم

را در زمان استنتاج تحت تاثیر قرار می‌دهد. در فصل دوم، عنوان شد که دو مدل رقابت نزدیکی بر روی داده‌های آزمون داشتند. اما در زمان اجرا، مشاهده می‌شود که مدل پیچیده‌تر yoloV5s، عملکرد سامانه را تحت تاثیر منفی قرار می‌دهد. این ویژگی، در تصمیم‌گیری ذی‌نفعان در استقرار پروژه موثر می‌باشد.

۳-۴ - نتیجه‌گیری

در این فصل، به بررسی فرآیند پیاده‌سازی واسط کاربری پرداختیم. مشاهده شد که شبکه عصبی YOLOv5، قابلیت صدور مدل‌های خود را به انواع مختلفی از فرمت‌های دیگر فراهم کرده است. با توجه به اینکه پیاده‌سازی واسط کاربری جهت استقرار بر مرورگر و اجرا در سمت کاربر طراحی شده است، باید مدل به فرمتی صادر شود که قابلیت مذکور را داشته باشد.

زبان مورد نیاز برای پیاده‌سازی در مرورگر، زبان جاوا اسکریپت می‌باشد. به همین جهت، مدل‌های آموزش دیده به فرمت tfjs صادر شدند. این فرمت، به کمک کتابخانه TensorFlow.js در زبان جاوا اسکریپت بارگذاری می‌شود.

با بارگذاری مدل در برنامه، می‌توان با اتصال به دوربین کاربر، جریان فریم‌های آن را دریافت کرد و برای تشخیص به مدل واگذار کرد. مدل با استنتاج بر فریم، اشیا موجود در تصویر را تشخیص می‌دهد در صورت معتبر بودن نتایج، زاویه کارت ملی را اصلاح می‌کند. با انجام این کار، از ارسال حجم داده‌های زیاد و همچنین پردازش‌های غیرضروری سرور برای اصلاح زاویه جلوگیری می‌شود.

نکته مهم در واسط کاربری، افت عملکرد آن هنگام استفاده از مدل yoloV5s می‌باشد. این مدل به دلیل حجم بالا، تشخیص اشیا و به‌روزرسانی فریم‌ها را با اختلال مواجه می‌کند و باعث کاهش تجربه کاربری می‌شود. با توجه به اینکه هر دو مدل عملکرد مشابهی بر داده‌های آزمون داشتند، می‌توان نتیجه‌گیری کرد که استفاده از مدل‌های ساده‌تر و استقرار آن در مرورگر، تصمیمی معقولانه خواهد بود.

فصل پنجم

جمع‌بندی، نتیجه‌گیری، و پژوهش‌های آتی

۵-۱- جمع‌بندی و نتیجه‌گیری

داشتن دقت بالا در تشخیص کارت ملی و توانایی اجرای پردازش‌های اولیه در سمت کاربر از نیازمندی‌های اساسی یک سامانه استخراج اطلاعات از کارت ملی به‌شمار می‌رود. در این پژوهش، سامانه‌ای جهت مکان‌یابی و اصلاح زاویه کارت ملی در مرورگر وب طراحی و پیاده‌سازی شد. استفاده از شبکه عصبی YOLOv5 برای تشخیص اشیاء و کتابخانه TensorFlow.js برای استقرار مدل آموزش دیده در مرورگر پیشنهاد داده شد. همانطور که مشاهده شد، روش پیشنهادی مذکور توانست با دقت بالا و در سمت کاربر، کارت ملی را تشخیص دهد و زاویه آن را در صورت لزوم اصلاح کند.

فرآیند آموزش شبکه تشخیص اشیاء شامل جمع‌آوری داده‌ها، برچسب‌زنی، و تنظیم دقیق مدل بود که با موفقیت انجام شد. نتایج ارزیابی‌ها نشان‌دهنده دقت بالای شبکه در شرایط مختلف نوری و زوایای متفاوت کارت ملی بود. همچنین، فرآیند استقرار نرم‌افزار در مرورگر با زبان قدرت‌مند جاوا اسکریپت نیز بررسی و آزمایش شد.

پیاده‌سازی سامانه شناسایی کارت ملی با استفاده از تکنولوژی‌های یادگیری عمیق و استقرار آن در مرورگر، مزایای متعددی از جمله دسترسی آسان، دقت بالا و کاهش زمان پردازش را به همراه دارد. استفاده از شبکه عصبی YOLOv5 به عنوان یک مدل پیشرفته تشخیص اشیاء، باعث شد که دقت و کارایی سامانه به طور قابل توجهی افزایش یابد.

نتایج به‌دست‌آمده نشان‌دهنده قابلیت بالای این سامانه برای استفاده در کاربردهای واقعی و عملیاتی است. به دلیل رسیدن به این اطمینان و دقت، از آموزش دیگر شبکه‌های عصبی تشخیص اشیاء صرف نظر شد.

همچنین مقایسه‌ای بین دو مدل اصلی و سبک‌تر YOLOv5 صورت گرفت. دیدیم که عملکرد هر دو مدل بر روی داده‌های آزمون بسیار نزدیک به هم و با دقت بسیار بالا بود، چرا که مجموعه داده آموزشی بسیار زیاد و با تنوع فراوان تدوین شده بود. با توجه به این نزدیکی، عملکرد مدل‌ها در زمان اجرا عامل تاثیرگذار در مقایسه این دو مدل است. همانطور که عنوان شد، مدل سبک‌تر YOLOv5n به دلیل داشتن حجم کمتر، در زمان اجرا عملکرد بهتری از خود در واسط کاربری نمایش می‌دهد.

با استفاده از این سامانه، کارت ملی کاربران در سیستم خودشان به آسانی تصویربرداری می‌شود و سپس برای استخراج اطلاعات و خوانش آنها به سمت سرور اصلی فرستاده می‌شود. با اینکار، بار پردازش اضافی از سرور برداشته می‌شود و باعث افزایش کارایی و صرفه‌جویی در هزینه و انرژی خواهد شد.

۵-۲- پیشنهادهای آتی

برای بهبود و توسعه بیشتر این سامانه، پیشنهاد می‌شود که در پژوهش‌های آتی به جنبه‌های زیر پرداخته شود:

- برچسب‌زنی اشیا چرخیده: توسعه روش‌های جدید برای برچسب‌زنی اشیا چرخیده و تشخیص زاویه دوران کارت ملی به منظور افزایش دقت تشخیص.

- ادغام با تکنولوژی‌های دیگر: بررسی امکان ادغام سامانه با تکنولوژی‌های دیگر نظیر تشخیص چهره و تطبیق تصویر به منظور افزایش کارایی و امنیت سامانه.

توجه به این جنبه‌ها می‌تواند به بهبود کیفیت و کارایی سامانه پیشنهادی کمک کرده و زمینه‌ساز توسعه بیشتر در کاربردهای مختلف باشد.

منابع و مراجع

- [١] S. D. J. Redmon, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779-788, 2016.
- [٢] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, and Scott Reed, "SSD: Single Shot MultiBox Detector," *Lecture Notes in Computer Science, Springer International Publishing*, pp. 21-37, 2016.
- [٣] J. Redmon and A. Farhad, "YOLOv3: An incremental improvement," *arXiv preprint arXiv*, 2018.
- [٤] A. S. G. Jocher, J. Borovec, A. Chaurasia, L. Changyu, V. Vasiljevic, J. Kwon, Y. Fang, F. Yu, L. Wang, N. Hogan, and J. Laughlin, "YOLOv5," 2020.
- [٥] A. Paszke, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019.
- [٦] J. Wu, Lumei Su, Zhiwei Lin, Yuhan Chen, Jiaming Ji, and Tianyou Li, "Object Detection of Flexible Objects with Arbitrary Orientation Based on Rotation-Adaptive YOLOv5," *Sensors* 23, vol. 10, p. 4925, 2023.
- [٧] Q. H. D. Hoan Tran Viet, "A Robust End-To-End Information Extraction System for Vietnamese Identity Cards," *2019 6th NAFOSTED Conference on Information and Computer Science (NICS)*, 2019.
- [٨] J. V. Mabel Rose, N. Bhavana, "YOLO-VEHICLE: REALTIME VEHICLE LICENCE PLATE DETECTION AND CHARACTER RECOGNITION USING YOLOV7 NETWORK," *International Journal of Data Science and Artificial Intelligence (IJDSAI)*, 2024.
- [٩] R. Khanam and M. Hussain, "WHAT IS YOLOV5: A DEEP LOOK INTO THE INTERNAL FEATURES OF THE POPULAR OBJECT DETECTOR," 2024.

- [١٠] N. T. a. Daniel Smilkov, Y. A. and, A. Y. and, N. K. and, P. Y. and, K. Z. and, *et al.*, "TensorFlow.js: Machine Learning for the Web and Beyond," *CoRR*, vol. abs/1901.05350, Fri, 01 Feb 2019 13:39:59 +0100 2019.
- [١١] M. Abadi and G. Savannah, "TensorFlow: A System for Large-Scale Machine Learning," *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 26.٢٠١٦, ٢٨٣-٥