

دانشگاه خواجہ ناصرالدین طوسی
دانشکده مهندسی کامپیوتر

پروژه دوره کارشناسی

مهندسی کامپیوتر

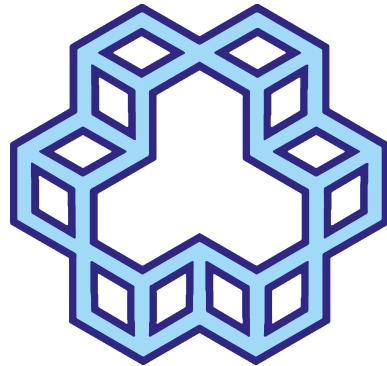
تشخیص اجزای خودرو در یک محیط تعاملی با
استفاده از شبکه های عصبی عمیق

علی دشت بزرگ

استاد راهنما

دکتر بهروز نصیحت کن

تابستان ۱۴۰۴



دانشگاه خواجہ ناصرالدین طوسی
دانشکده مهندسی کامپیوتر

پروژه دوره کارشناسی
مهندسی کامپیوتر

عنوان

تشخیص اجزای خودرو در یک محیط تعاملی با
استفاده از شبکه های عصبی عمیق

نگارش
علی دشت بزرگ

استاد راهنما

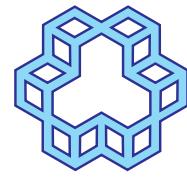
دکتر بهروز نصیحت کن

تابستان ۱۴۰۴

رَبِّ الْجَنَّاتِ وَالْجَمَارِ

تقدیم به:

به پدر ، مادر و خواهرم



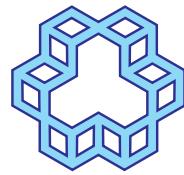
دانشگاه صنعتی خواجه نصیرالدین طوسی

تأییدیه هیئت داوران جلسه‌ی دفاع از پروژه کارشناسی

هیأت داوران پس از مطالعه‌ی پروژه و شرکت در جلسه‌ی دفاع از پایان‌نامه تهیه شده با عنوان «تشخیص اجزای خودرو در یک محیط تعاملی با استفاده از شبکه‌های عصبی عمیق» توسط آقای علی دشت بزرگ صحبت و کفایت تحقیق انجام شده را برای اخذ درجه‌ی کارشناسی در رشته‌ی مهندسی کامپیوتر در تاریخ تابستان ۱۴۰۴ مورد تأیید قرار دادند.

۱. استاد راهنما: دکتر بهروز نصیحت کن امضا

۲. استاد داور: دکتر بابک ناصرشريف امضا



دانشگاه صنعتی خواجه نصیرالدین طوسی

اظهارنامه دانشجو

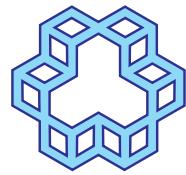
اینجانب علی دشت بزرگ به شماره دانشجویی ۴۰۰۵۰۹۳ کارشناسی رشته‌ی مهندسی کامپیوتر دانشکده دانشگاه خواجه نصیرالدین طوسی گواهی می‌نمایم که تحقیقات ارائه شده در این پایان‌نامه با عنوان:

تشخیص اجزای خودرو در یک محیط تعاملی با استفاده از شبکه‌های عصبی عمیق

توسط اینجانب انجام و بدون هرگونه دخل و تصرف است و موارد نسخه برداری شده از آثار دیگران را با ذکر کامل مشخصات منع ذکر کرده‌ام. در صورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم (قانون حمایت از حقوق مؤلفان و مصنفان و قانون ترجمه و تکثیر کتب و نشریات و آثار صوتی، ضوابط و مقررات آموزشی، پژوهشی و انصباطی وغیره) با اینجانب رفتار خواهد شد. در ضمن، مسئولیت هرگونه پاسخگویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذی صلاح (اعم از اداری و قضایی) به عهده‌ی اینجانب خواهد بود و دانشگاه هیچ گونه مسئولیتی در این خصوص نخواهد داشت.

نام و نام خانوادگی دانشجو: علی دشت بزرگ

تاریخ و امضای دانشجو:



دانشگاه صنعتی خواجه نصیرالدین طوسی

حق طبع، نشر و مالکیت نتایج

حق چاپ و تکثیر این پایان نامه متعلق به نویسنده‌گان آن می‌باشد. بهره برداری از این پایان نامه در چهارچوب مقررات کتابخانه و با توجه به محدودیتی که توسط استاد راهنمای شرح زیر تعیین می‌گردد، بلامانع است:

□ بهره‌برداری از این پروژه برای همگان و با ذکر منبع، بلامانع است.

□ بهره‌برداری از این پروژه با اخذ مجوز از استاد راهنمای و با ذکر منبع، بلامانع است.

□ بهره‌برداری از این پروژه تا تاریخ _____ ممنوع است.

استاد راهنمای دکتر بهروز نصیحت کن امضا

قدردانی

اکنون که به یاری پروردگار و یاری و راهنمایی اساتید بزرگ موفق به پایان این رساله شده‌ام وظیفه خود دانشته که نهایت سپاسگزاری را از تمامی عزیزانی که در این راه به من کمک کرده‌اند را به عمل آورم: در آغاز از استاد بزرگ و دانشمند جناب آقای دکتر بهروز نصیحت کن که راهنمایی این پایان‌نامه را به عهده داشته‌اند کمال تشکر را دارم. از داور گرامی ... که زحمت داوری و تصحیح این پایان‌نامه را به عهده داشتند کمال سپاس را دارم. خالصانه از تمامی اساتید و معلمان و مدرسانی که در مقاطع مختلف تحصیلی به من علم آموخته و مرا از سرچشمۀ دانایی سیراب کرده‌اند متشکرم. از کلیه هم دانشگاهیان و همراهان عزیز، دوست خوبیم آقای دانیال فاضل پور نهایت سپاس را دارم.

و در پایان این پایان‌نامه را تقدیم می‌کنم به خانواده ام که با حضور و همراهی اشان همیشه راه را به من نشان داده اند و مرا در این راه استوار و ثابت قدم نموده اند.

علی دشت بزرگ

تابستان ۱۴۰۴

چکیده

این پایان نامه یک مدل سبک و تعاملی برای تشخیص قطعات خودرو بر اساس نقطه یا خطی که کاربر روی تصویر انتخاب می کند، ارائه می دهد. در این روش، با کلیک کاربر بر روی نقطه ای از تصویر خودرو یا انتخاب یک خط، بخشی از تصویر در اطراف ورودی استخراج شده و به شبکه عصبی داده می شود تا کلاس آن نقطه (مانند چرخ جلو سمت چپ، نقاط مرکزی یا نقاط گوشه ای) پیش بینی شود. انگیزه و کاربرد اصلی این کار، تسريع فرایند بر چسب گذاری در حین حاشیه نویسی بخش های مختلف خودرو بوده است. مدل پیشنهادی از یک شبکه EfficientNet^۱ به عنوان بخش استخراج ویژگی و یک لایه پرسپترون چند لایه ساده (MLP) برای طبقه بندی استفاده می کند. تمرکز مدل بر استفاده از بخش محلی تصویر به جای پردازش کل صحنه، باعث کاهش قابل توجه هزینه محاسباتی در عین حفظ دقت مناسب می شود. نتایج آزمایش ها بر روی مجموعه داده ای شامل نقاط کلیدی قطعات خودرو نشان می دهد که این روش سریع، ساده برای پیاده سازی، و کارآمد برای شناسایی آنی قطعات خودرو در شرایط واقعی است. همچنین، مدل نقطه ای به دقت ریز کلان^۲ ۰.۸۸ و مدل خطی به ۰.۸۲ دست یافت که نشان دهنده توانایی بالای رویکرد پیشنهادی در هر دو سناریوی تعاملی است.

واژگان کلیدی تشخیص اجزای خودرو، محیط تعاملی، شبکه های عصبی عمیق، بینایی ماشین، شناسایی قطعات خودرو

۱EfficientNet: خانواده ای از شبکه های عصبی پیچشی توسعه یافته توسط گوگل که با استفاده از مقیاس بندی ترکیبی عمق، عرض و
وضوح، دقت بالا را با تعداد پارامتر کمتر به دست می آورد.

²Micro Precision

فهرست مطالب

ث

فهرست تصاویر

ج

فهرست جداول

۱	مقدمه	فصل ۱:
۱	عنوان تحقیق	۱.۱
۲	تعریف مسأله	۲.۱
۳	تاریخچه‌ای از موضوع تحقیق	۳.۱
۴	تعریف موضوع تحقیق	۴.۱
۴	نوآوری تحقیق	۵.۱
۵	روش انجام تحقیق	۶.۱
۶	خلاصه فصل‌ها	۷.۱
۷	جمع‌بندی	۸.۱
۹	مفاهیم بنیادی	فصل ۲:
۹	مقدمه	۱.۲
۱۰	تعاریف، اصول و مبانی نظری	۲.۲
۱۰	شبکه‌های عصبی مصنوعی	۱.۲.۲
۱۰	پرسپترون چندلایه (MLP)	۲.۲.۲
۱۱	شبکه‌های عصبی پیچشی (CNN)	۳.۲.۲
۱۱	EfficientNet	۴.۲.۲

۱۱	مروری بر ادبیات موضوع	۳.۲
۱۲	داده	۴.۲
۱۲	ToosiCubix چارچوب	۵.۲
۱۴	صورت‌بندی مسئله	۱.۵.۲
۱۴	انواع حاشیه‌نویسی و نقش آن‌ها	۲.۵.۲
۱۵	راهبرد بهینه‌سازی	۳.۵.۲
۱۵	مدیریت ابهامات	۴.۵.۲
۱۶	تنظیم دقیق در حوزه پیکسل	۵.۵.۲
۱۶	خلاصه	۶.۵.۲
۱۶	دستیار نیمه‌خودکار برچسب‌گذاری	۶.۲
۱۷	نتیجه‌گیری	۷.۲
۱۹	فصل ۳: روش کار	
۱۹	مقدمه	۱.۳
۲۰	SAM	۲.۳
۲۰	نوآوری	۱.۲.۳
۲۱	آموزش	۲.۲.۳
۲۲	معماری پیشنهادی	۳.۳
۲۳	ورودی‌ها و داده‌ها	۴.۳
۲۵	داده‌های نقاط کلیدی	۱.۴.۳
۲۵	افزایش داده‌ها - نقاط (Data Augmentation)	۱.۱.۴.۳
۲۷	داده‌های خطوط	۲.۴.۳
۲۷	نحوه برش تصویر برای خطوط	۱.۲.۴.۳
۲۹	افزایش داده‌ها - خطوط (Data Augmentation)	۲.۲.۴.۳
۳۱	روش آموزش	۵.۳
۳۱	تنظیمات مشترک	۱.۵.۳

۳۱	۲.۵.۳ تفاوت‌های آموزش برای هر مدل
۳۲	۶.۳ تحلیل چند مقیاسه
۳۲	۱.۶.۳ ساده
۳۲	۲.۶.۳ میانگین
۳۲	۳.۶.۳ ماکسیمم
۳۳	۷.۳ روش ارزیابی
۳۳	۱.۷.۳ معیارهای ارزیابی
۳۴	۲.۷.۳ روش آزمایش
۳۴	۸.۳ جمع‌بندی
۳۶	فصل ۴: بحث و نتایج
۳۶	۱.۴ مقدمه
۳۶	۲.۴ معیارهای ارزیابی مدل
۳۶	۱.۲.۴ سرعت
۳۷	۲.۲.۴ دقت (Precision)
۳۷	۳.۲.۴ بازیابی (Recall)
۳۸	۴.۲.۴ امتیاز F1
۳۹	۳.۴ نتایج
۳۹	۱.۳.۴ مدل نقاط
۳۹	۲.۳.۴ مدل خط
۴۰	۳.۳.۴ نمودارهای تابع هزینه و دقت
۴۰	۴.۴ نمونه‌های اجرا
۴۱	۵.۴ جمع‌بندی
۴۶	فصل ۵: نتیجه‌گیری
۴۶	۱.۵ مقدمه
۴۶	۲.۵ جمع‌بندی

ت

فهرست مطالب

۳.۵ تحقیقات آینده ۴۷

کتابنامه ۴۹

فهرست تصاویر

۱۳	نمونه نقاط کلیدی	۱.۲
۱۴	نمونه هایی از اجرای ToosiCubix	۲.۲
۱۷	منوی برنامه قبل و بعد از تغییرات	۳.۲
۲۰	معماری کلی مدل SAM	۱.۳
۲۱	معماری رمزگشایی مدل SAM	۲.۳
۲۴	معماری مدل پیشنهادی	۳.۳
۲۹	نمونه ای از برش عکس با ورودی یک خط	۴.۳
۳۳	نمونه ای از ورودی با درنظر گرفتن چند مقیاس	۵.۳
۴۱	نمودارهای تابع هزینه و دقت مدل نقاط	۱.۴
۴۱	نمودارهای تابع هزینه و دقت مدل خط	۲.۴
۴۲	ادغام مدل با برنامه برچسب گزاری (قسمت اول)	۳.۴
۴۳	ادغام مدل با برنامه برچسب گزاری (قسمت دوم)	۴.۴

فهرست جداول

۱.۳	فراوانی داده های نقاط	۲۵
۲.۳	فراوانی داده های خطوط	۲۷
۱.۴	زمان استنتاج مدل برای سه حالت مختلف بر روی تصاویر تست	۳۷
۲.۴	مقایسه نتایج مدل نقاط روی مجموعه ارزیابی و تست	۳۹
۳.۴	مقایسه نتایج مدل خطوط روی مجموعه ارزیابی و تست	۴۰

فصل ۱

مقدمه

فرآیند برچسب‌گذاری داده‌ها در پروژه‌های بینایی ماشین، به ویژه زمانی که شامل شناسایی اجزای جزئی و پیچیده اشیاء است، یکی از پرهزینه‌ترین و زمان‌برترین مراحل توسعه به شمار می‌رود. در حوزه‌ی خودرو، برچسب‌گذاری دقیق بخش‌های مختلف وسیله نقلیه در تصاویر، نیازمند صرف زمان زیاد و اغلب دانش تخصصی می‌باشد. برای کاهش این بارکاری، ابزارهای برچسب‌گذاری نیمه‌خودکار می‌توانند با ارائه پیشنهادهای خودکار به برچسب‌زن‌ها، به تسريع و تسهیل این فرآیند کمک کنند. در این پژوهش، روشی برای طبقه‌بندی بخش‌های خودرو بر اساس یک نقطه و یا یک خط ورودی ارائه شده است؛ به این صورت که کاربر نقطه و یا خطی را بر روی تصویر انتخاب می‌کند و سیستم با برش ناحیه‌ای پیرامون آن نقطه و استفاده از یک مدل یادگیری عمیق، بخش متناظر خودرو را شناسایی و طبقه‌بندی می‌نماید. این رویکرد موجب افزایش سرعت برچسب‌گذاری، بهبود یکپارچگی نتایج و کاهش خطاهای انسانی خواهد شد.

۱.۱ عنوان تحقیق

در بسیاری از پروژه‌های بینایی ماشین، تهیه مجموعه داده‌های بزرگ و باکیفیت همراه با برچسب‌گذاری دقیق، یکی از پرهزینه‌ترین و زمان‌برترین مراحل توسعه محسوب می‌شود. این موضوع در حوزه‌ی خودرو و بهویژه در برچسب‌گذاری بخش‌های جزئی یک وسیله نقلیه اهمیت بیشتری می‌یابد، زیرا فرآیند شناسایی و برچسب‌گذاری دقیق اجزای خودرو اغلب نیازمند صرف زمان زیاد و دانش تخصصی است. استفاده از ابزارهای نیمه‌خودکار با

حضور کاربر در حلقه (human-in-the-loop) می‌تواند با ارائه‌ی پیشنهادهای اولیه و خودکار به برچسبزن‌ها، سرعت و یکپارچگی فرآیند برچسب‌گذاری را به طور قابل توجهی افزایش دهد [۱، ۲، ۳]. نمونه‌هایی از این رویکرد را می‌توان در روش‌های برچسب‌گذاری مبتنی بر کلیک مانند Extreme Clicking و مدل‌های پیشرفته‌تری همچون Segment Anything Model (SAM) مشاهده کرد که با دریافت نقاط یا نواحی اولیه از کاربر، بخش متناظر را به صورت خودکار استخراج می‌کنند. در این پژوهش، روشی برای طبقه‌بندی بخش‌های خودرو بر اساس نقطه‌ی ورودی کاربر ارائه شده است؛ به این صورت که کاربر با انتخاب یک نقطه بر روی تصویر، ناحیه‌ای کوچک پیرامون آن نقطه برش داده شده و سپس با استفاده از یک مدل یادگیری عمیق، بخش متناظر خودرو شناسایی و طبقه‌بندی می‌شود. این رویکرد، ضمن کاهش خطاهای انسانی، می‌تواند به طور چشمگیری فرآیند برچسب‌گذاری مجموعه‌داده‌های تصویری را تسریع کند و کیفیت داده‌های خروجی را بهبود بخشد [۴، ۵].

۲.۱ تعریف مسئله

در بسیاری از پروژه‌های بینایی ماشین، به ویژه در زمینه‌ی تحلیل تصاویر خودرو، برچسب‌گذاری دقیق بخش‌های مختلف وسیله نقلیه یکی از چالش‌های اصلی به شمار می‌رود. این فرآیند نیازمند دقت بالا، دانش تخصصی و صرف زمان قابل توجه است که می‌تواند منجر به افزایش هزینه‌ها و تأخیر در توسعه سیستم‌های هوشمند شود. به علاوه، خطاهای انسانی در برچسب‌گذاری دستی باعث کاهش کیفیت داده‌ها و در نتیجه کاهش دقت مدل‌های یادگیری عمیق می‌شود.

در این تحقیق، مسئله اصلی برچسب‌گذاری نیمه خودکار بخش‌های خودرو بر اساس ورودی‌های کاربر مانند نقطه یا خط روی تصویر است. هدف، طراحی سیستمی است که با دریافت یک نقطه یا خط ورودی، ناحیه‌ی مربوط به آن بخش خودرو را شناسایی و طبقه‌بندی کند. این کار با استفاده از مدل‌های یادگیری عمیق و برش نواحی کوچک پیرامون نقطه ورودی انجام می‌شود تا فرآیند برچسب‌گذاری سریع‌تر، دقیق‌تر و کم‌هزینه‌تر گردد. پرسش‌های اساسی این تحقیق عبارت‌اند از:

- چگونه می‌توان با استفاده از ورودی‌های نقطه‌ای یا خطی کاربر، بخش‌های مختلف خودرو را به صورت خودکار شناسایی و طبقه‌بندی کرد؟
- چه مدل یادگیری عمیقی برای تشخیص دقیق بخش‌های کوچک و پیچیده خودرو مناسب‌تر است؟

- چگونه می‌توان سرعت و دقت فرآیند برچسب‌گذاری را با استفاده از این روش افزایش داد و خطاهای انسانی را کاهش داد؟

با پاسخ به این پرسش‌ها، این تحقیق قصد دارد تا گامی مؤثر در جهت تسهیل فرآیند برچسب‌گذاری داده‌های بینایی ماشین، به ویژه در حوزه خودرو، بردارد.

۳.۱ تاریخچه‌ای از موضوع تحقیق

مطالعه و بررسی کارهای پیشین، پایه‌ای اساسی برای هر پژوهش علمی است که به مشخص شدن جایگاه تحقیق و نوآوری‌های آن کمک می‌کند. در زمینه برچسب‌گذاری نیمه‌خودکار تصاویر خودرو، پژوهش‌های متعددی صورت گرفته است که از روش‌های مختلف یادگیری ماشین و یادگیری عمیق برای تسریع و بهبود کیفیت برچسب‌گذاری استفاده کرده‌اند.

برای نمونه، روش‌هایی مبتنی بر برچسب‌گذاری با کلیک مانند [۲] Extreme Clicking پیشنهاد شده‌اند که با دریافت نقاط کلیدی از کاربر، ناحیه‌های مورد نظر را به صورت خودکار استخراج می‌کنند. همچنین مدل‌های پیشرفته‌تر مانند Segment Anything Model (SAM) [۳] با بهره‌گیری از معماری‌های قدرتمند یادگیری عمیق، امکان استخراج دقیق بخش‌های مختلف تصویر را با دریافت ورودی‌های اولیه از کاربر فراهم کرده‌اند. در حوزه‌ی تحلیل تصاویر خودرو، پژوهش‌هایی نیز برای شناسایی و طبقه‌بندی بخش‌های مختلف وسیله نقلیه انجام شده است. این پژوهش‌ها عمدتاً بر روی استخراج ویژگی‌های بصری بخش‌های مختلف خودرو و استفاده از مدل‌های طبقه‌بندی متمرکز بوده‌اند [۴، ۵]. با این حال، نیاز به روشی کارآمد و دقیق که بتواند کمترین دخالت انسانی و بر اساس ورودی‌های ساده‌ای مانند نقطه یا خط، برچسب‌گذاری را انجام دهد، همچنان پابرجاست.

در این تحقیق سعی شده است با ترکیب مزایای روش‌های نیمه‌خودکار و مدل‌های یادگیری عمیق پیشرفته، چارچوبی برای طبقه‌بندی بخش‌های خودرو ارائه شود که ضمن حفظ دقت، سرعت و کارایی فرآیند برچسب‌گذاری را بهبود بخشد.

۴.۱ تعریف موضوع تحقیق

موضوع این تحقیق، توسعه روشی کارآمد و سریع برای طبقه‌بندی بخش‌های مختلف خودرو در تصاویر بر اساس نقطه یا خط ورودی کاربر است. در این حوزه، نیاز به ارائه روشی وجود دارد که علاوه بر دقت بالا، بتواند در زمان کوتاه و به صورت بلادرنگ روی سیستم‌های کاربران نهایی^۱ اجرا شود. این امر به دلیل کاربردهای عملی مانند ابزارهای برچسب‌گذاری نیمه خودکار در محیط‌های صنعتی و تحقیقاتی اهمیت فراوانی دارد. فرضیات اصلی مسئله عبارتند از:

- مدل یادگیری عمیق باید قادر باشد با ورودی‌های کوچک و محدود (مانند یک نقطه یا خط مشخص شده توسط کاربر) بخش متاظر خودرو را به دقت طبقه‌بندی کند.
- اجرای مدل باید به حدی سریع باشد که تجربه کاربری روان و بدون تأخیر قابل توجه فراهم کند، به خصوص در سخت‌افزارهای معمول کاربران نهایی.
- ناحیه برش داده شده از تصویر باید به اندازه‌ای کوچک باشد که پردازش مدل سریع انجام شود، اما در عین حال شامل اطلاعات کافی برای شناسایی دقیق بخش خودرو باشد.
- سیستم باید بتواند خطاها انسانی را کاهش دهد و کیفیت برچسب‌گذاری را بهبود بخشد.

با توجه به این فرضیات، هدف تحقیق طراحی و پیاده‌سازی مدلی است که علاوه بر دقت قابل قبول، از نظر سرعت اجرای بهینه باشد تا در کاربردهای عملی و در شرایط واقعی قابل استفاده باشد.

۵.۱ نوآوری تحقیق

هدف اصلی این تحقیق، ارائه روشی نوین برای طبقه‌بندی بخش‌های خودرو بر اساس ورودی کاربر (نقطه یا خط) است که بتواند با دقت بالا و در عین حال با سرعت بسیار زیاد، روی سیستم‌های معمول کاربران نهایی اجرا شود. این ویژگی امکان استفاده عملی از مدل را در ابزارهای برچسب‌گذاری نیمه خودکار و کاربردهای صنعتی و پژوهشی فراهم می‌کند.

¹End Users

نوآوری‌های اصلی این تحقیق عبارتند از:

- ترکیب انواع روش‌های برش ناحیه‌ای کوچک پیرامون ورودی کاربر با یک مدل یادگیری عمیق بهینه‌شده برای اجرا با سرعت بالا.
- طراحی و پیاده‌سازی معماری‌ای که بتواند روی سخت‌افزارهای رایج کاربران نهایی (مانند لپ‌تاپ‌ها یا رایانه‌های رومیزی بدون GPU پیشرفته) عملکرد بلاذرنگ ارائه دهد.
- دستیابی به مدلی با دقت ۸۷ درصد بر روی مجموعه داده‌ی آزمون افزایش یکپارچگی و کیفیت داده‌های برچسب‌گذاری شده از طریق کاهش خطاهای انسانی.

اهمیت این تحقیق در آن است که با ارائه روشی سریع و دقیق، می‌توان فرآیند برچسب‌گذاری مجموعه داده‌های تصویری را به طور چشمگیری تسهیل و تسریع کرد. این امر نه تنها هزینه‌ها و زمان توسعه پژوهه‌های بینایی ماشین را کاهش می‌دهد، بلکه امکان تولید داده‌های باکیفیت تر برای آموزش مدل‌های یادگیری عمیق را فراهم می‌سازد. همچنین نتایج این پژوهش می‌تواند به عنوان پایه‌ای برای تحقیقات آینده در حوزه تعامل انسان و ماشین در برچسب‌گذاری داده‌ها و توسعه ابزارهای هوشمند مورد استفاده قرار گیرد.

۶.۱ روش انجام تحقیق

در این پژوهش، از مجموعه داده‌هایی استفاده شده است که توسط نرم‌افزار معرفی شده در [۶] با نام ToosiCubix برچسب‌گذاری شده‌اند. این مجموعه داده شامل تصاویر خودروهایی است که بخش‌های مختلف آن‌ها با دقت بالا توسط ابزار مذکور در قالب نقاط موردنظر نشانه‌گذاری شده‌اند.

برای طراحی و بهینه‌سازی مدل یادگیری عمیق، روش‌های مختلفی برای تبدیل ورودی نقطه‌ای کاربر به ورودی قابل استفاده توسط مدل بررسی شده است. این ورودی‌ها معمولاً شامل نواحی برش داده شده کوچک اطراف نقطه انتخاب شده می‌باشد که ساختار و اطلاعات لازم برای تشخیص بخش‌های خودرو را حفظ کند. جزئیات و مقایسه این روش‌ها در فصل‌های بعدی به طور مفصل شرح داده خواهد شد.

روش‌های گرددآوری داده، طراحی مدل، آموزش و ارزیابی آن در قالب یک چرخه تکرارشونده انجام شده است تا به بهترین عملکرد ممکن از نظر دقیق و سرعت دست یابیم. آزمایش‌های متعددی به منظور انتخاب ساختار بهینه مدل و تعیین پارامترهای مناسب صورت گرفته است. این روند کلی پژوهش، ضمن تأکید بر اهمیت کیفیت داده‌ها و سرعت اجرا، پایه‌ای برای توسعه مدل‌های کاربردی و کارآمد در حوزه برچسب‌گذاری نیمه خودکار تصاویر خودرو فراهم کرده است.

۷.۱ خلاصه فصل‌ها

این پایان‌نامه شامل پنج فصل می‌باشد که هر فصل به بررسی جنبه‌ای از پژوهش می‌پردازد:

- فصل اول به مقدمه پژوهش اختصاص دارد و ضمن معرفی مسئله، اهداف، نوآوری‌ها و روش انجام تحقیق را تشریح می‌کند.
- فصل دوم به مفاهیم بنیادی مرتبط با موضوع، از جمله ساختار شبکه‌های عصبی پیچشی^۱ و معماری‌های پیشرفته‌ای مانند EfficientNet می‌پردازد تا زمینه لازم برای فهم روش ارائه شده فراهم گردد.
- فصل سوم به معرفی روش اصلی پژوهش اختصاص یافته است و جزئیات طراحی مدل، پردازش داده‌ها و نحوه استفاده از ورودی‌های کاربر شرح داده می‌شود.
- فصل چهارم به ارائه نتایج آزمایش‌ها، ارزیابی عملکرد مدل و تحلیل داده‌های به دست آمده می‌پردازد.
- در نهایت، فصل پنجم به جمع‌بندی کلی پژوهش، بحث درباره نتایج، بررسی محدودیت‌ها و پیشنهاد مسیرهای آینده تحقیقات اختصاص یافته است.

¹CNN: Convolutional Neural Networks

۸.۱ جمع‌بندی

در این فصل، نکات مهم و چارچوب کلی پایان‌نامه بیان گردید تا خواننده بتواند دیدی جامع نسبت به موضوع، اهداف، نوآوری‌ها و روش تحقیق به دست آورد. در این فصل، اهمیت موضوع برچسب‌گذاری داده‌ها در پروژه‌های بینایی ماشین به ویژه در حوزه خودرو، و ضرورت استفاده از روش‌های نیمه خودکار با حضور کاربر برای افزایش سرعت و دقت برچسب‌گذاری تشریح شد. همچنین ساختار کلی پایان‌نامه و فصول آتی به صورت خلاصه معرفی گردید تا خواننده با مسیر پژوهش و سازماندهی مطالب آشنا شود. این چارچوب، زمینه مناسبی برای مطالعه فصل‌های بعدی فراهم می‌آورد و کمک می‌کند تا مطالب به صورت منسجم و هدفمند دنبال شوند.

فصل ۲

مفاهیم بنیادی

۱.۲ مقدمه

در سال‌های اخیر، پیشرفت‌های چشمگیر در حوزه یادگیری عمیق^۱ و بهویژه شبکه‌های عصبی مصنوعی^۲ تحول بزرگی در حل مسائل پیچیده بینایی ماشین ایجاد کرده است. این شبکه‌ها با الهام از ساختار و عملکرد مغز انسان طراحی شده‌اند و قادرند ویژگی‌ها و الگوهای پنهان در داده‌ها را به صورت خودکار یاد بگیرند.

درک مفاهیم بنیادی شبکه‌های عصبی و معماری‌های مختلف آن‌ها، پیش‌نیاز مهمی برای فهم روش‌های مورد استفاده در این تحقیق است. در این فصل، ابتدا به معرفی مفاهیم پایه‌ای شبکه‌های عصبی پرداخته می‌شود و سپس ساختارهای مهمی همچون پرسپیترون چندلایه^۳ و شبکه‌های عصبی پیچشی بررسی خواهند شد. در ادامه نیز معماری پیشرفته EfficientNet که در این پژوهش به عنوان شبکه پایه مورد استفاده قرار گرفته است، معرفی و تحلیل می‌شود.

هدف از این فصل، ایجاد بستری نظری و مفهومی برای درک بهتر جزئیات فنی روش پیشنهادی است تا خواننده بتواند با دیدی روشن، مراحل طراحی، آموزش و ارزیابی مدل را در فصول بعدی دنبال کند.

¹Deep Learning ²Artificial Neural Networks ³Multi-Layer Perceptron, MLP

۲.۲ تعاریف، اصول و مبانی نظری

در این بخش، به معرفی و توضیح مفاهیم پایه‌ای و اصول نظری مرتبط با شبکه‌های عصبی مصنوعی و معماری‌های مورد استفاده در این تحقیق پرداخته می‌شود. این مفاهیم شامل شبکه‌های عصبی مصنوعی، شبکه‌های پرسپترون چندلایه، شبکه‌های عصبی پیچشی و معماری EfficientNet می‌باشند.

۱.۲.۲ شبکه‌های عصبی مصنوعی

شبکه‌های عصبی مصنوعی مدل‌هایی الهام گرفته از ساختار و عملکرد مغز انسان هستند که برای یادگیری الگوها و روابط پیچیده میان داده‌ها به کار می‌روند. این شبکه‌ها از لایه‌هایی تشکل از واحدهای محاسباتی به نام نورون تشکیل شده‌اند که هر یک ورودی‌ها را دریافت کرده، با وزن‌های قابل یادگیری ترکیب می‌کنند و سپس نتیجه را از طریق تابع فعال‌سازی عبور می‌دهند. آموزش شبکه‌های عصبی با استفاده از الگوریتم‌هایی نظیر پس انتشار خطأ^۱ و روش‌های بهینه‌سازی مانند SGD (Stochastic Gradient Descent) یا Adam انجام می‌شود. این ساختار امکان تقریب توابع بسیار پیچیده و استخراج ویژگی‌های غیرخطی را فراهم می‌کند.

۲.۲.۲ پرسپترون چندلایه (MLP)

پرسپترون چندلایه یک نوع شبکه عصبی پیش خور^۲ است که شامل یک لایه ورودی، یک یا چند لایه میانی (پنهان) و یک لایه خروجی می‌باشد. هر لایه پنهان از تعدادی نورون با تابع فعال‌سازی غیرخطی تشکیل شده است که به مدل اجازه می‌دهد روابط پیچیده میان داده‌ها را یاد بگیرد. MLP‌ها برای مسائل طبقه‌بندی و رگرسیون عمومی به کار می‌روند و به دلیل ساختار ساده‌شان، پایه‌ای برای درک مفاهیم پیشرفته‌تر شبکه‌های عصبی محسوب می‌شوند. محدودیت اصلی آن‌ها در کاربردهای بینایی ماشین این است که ویژگی‌های مکانی و ساختار فضایی تصاویر را به خوبی مدل‌سازی نمی‌کنند.

¹Backpropagation

²Feedforward Neural Network

۳.۰.۰.۲ شبکه‌های عصبی پیچشی (CNN)

شبکه‌های عصبی پیچشی، نوعی شبکه عصبی تخصصی برای پردازش داده‌های ساختارمند به ویژه تصاویر هستند. هسته اصلی این شبکه‌ها لایه پیچشی است که با اعمال فیلترهای قابل یادگیری بر روی ورودی، ویژگی‌های مکانی و محلی داده را استخراج می‌کند. این فیلترها در کل تصویر به اشتراک گذاشته می‌شوند و باعث کاهش تعداد پارامترها و افزایش کارایی می‌گردند. پس از لایه‌های پیچشی، معمولاً از لایه‌های تجمعی^۱ برای کاهش ابعاد و از لایه‌های کاملاً متصل^۲ برای تصمیم‌گیری نهایی استفاده می‌شود. CNN‌هاستون فقرات بسیاری از مدل‌های پیشرفته در بینایی ماشین، تشخیص اشیاء و طبقه‌بندی تصاویر هستند.

EfficientNet ۴.۰.۰.۲

یک خانواده از معماری‌های شبکه‌های عصبی پیچشی است که با هدف دستیابی به بیشترین دقت ممکن در طبقه‌بندی تصاویر با کمترین منابع محاسباتی طراحی شده‌اند [۷]. این معماری بر پایه ایده «مقیاس‌بندی مرکب»^۳ شکل گرفته است که در آن ابعاد شبکه (عمق، عرض و رزولوشن ورودی) به طور همزمان و متناسب افزایش می‌یابند. نسخه پایه این مدل، EfficientNet-B0، با استفاده از جستجوی معماری عصبی^۴ طراحی شده و سپس نسخه‌های بزرگ‌تر آن با افزایش هماهنگ پارامترها ایجاد شده‌اند. EfficientNet با نسبت بسیار بالای دقت به هزینه محاسباتی، انتخابی مناسب برای کاربردهایی است که محدودیت منابع سخت‌افزاری دارند، از جمله سیستم‌های بلاذرنگ و دستگاه‌های لبه‌ای. به همین دلیل، این معماری پایه برای طراحی مدلی که در محیط کاربران نهایی و با منابع محدود اجرا می‌شود، انتخاب شده است.

۳.۰.۲ مروری بر ادبیات موضوع

در حوزه بخش‌بندی تصاویر، مدل Segment Anything Model (SAM)^[۳] به عنوان یکی از پیشرفتهای مهم مطرح شده است. این مدل که توسط AI توسعه یافته، قادر است با دریافت ورودی‌هایی مانند

^۱Pooling Layers ^۲Fully Connected Layers ^۳Compound Scaling ^۴Neural Architecture Search, NAS

نقشه یا جعبهٔ مرزی از کاربر، ماسک بخش‌بندی مربوط به شئ مورد نظر را در تصویر تولید کند. SAM از یک استخراج‌کنندهٔ ویژگی قدرتمند و یک ماثول پیش‌بینی ماسک بهره می‌برد و به دلیل آموزش بر روی مجموعه داده بسیار بزرگ، می‌تواند بدون نیاز به آموزش مجدد، در دامنه‌های مختلف عملکرد مطلوبی ارائه دهد.

تفاوت اصلی تحقیق حاضر با SAM در هدف نهایی پردازش است. در حالی که SAM ورودی کاربر را برای تولید ماسک بخش‌بندی استفاده می‌کند، رویکرد ما به جای بخش‌بندی، از این ورودی برای دسته‌بندی بخشی از تصویر که ورودی به آن اشاره دارد استفاده می‌کند. این تغییر هدف نیازمند طراحی معماری و لایه‌های خروجی متفاوت نسبت به SAM بوده و ما قادر می‌سازد تا مستقیماً نوع ورودی مورد نظر را پیش‌بینی کنیم. ضمن همین موضوع، ما می‌توانیم برای طراحی مدل به سمت روش‌های سبک‌تر و بهینه‌تر برای اجرا در محیط‌های کاربران نهایی با منابع محدود حرکت کنیم.

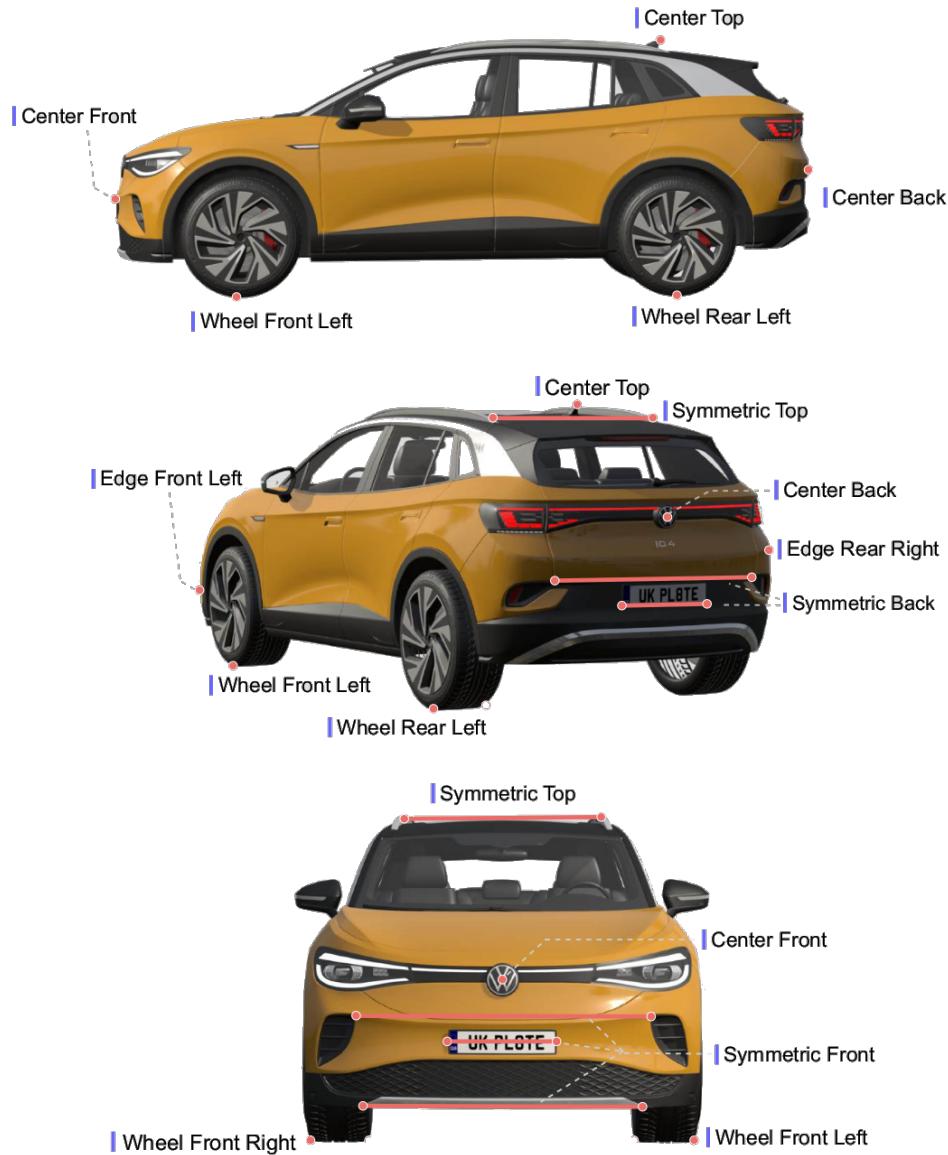
۴.۲ داده

در این پایان‌نامه، هدف ما گروه‌بندی مجموعه‌ای از نقاط کلیدی^۱ و خطوط است. برای اختصار، در ادامه‌ی متن، به هر دو مورد به عنوان «نقاط کلیدی» اشاره خواهیم کرد. برای آشنایی بهتر با این نقاط، به شکل ۱.۲ توجه کنید. در این مسئله، نقاط در ۱۵ کلاس و خطوط در ۳ کلاس مختلف طبقه‌بندی شده‌اند.

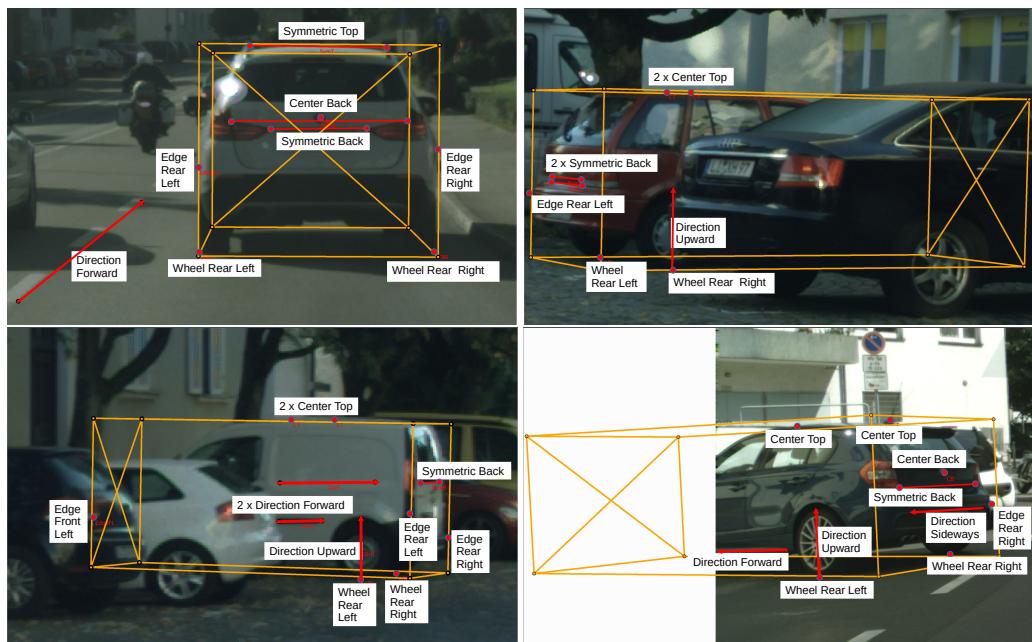
۵.۲ چارچوب ToosiCubix

در این بخش، مروری بر چارچوب ToosiCubix به عنوان پایهٔ مدل حاشیه‌نویسی خود ارائه می‌کنیم. این چارچوب مسئلهٔ برآورده مکعب‌های سه‌بعدی برای وسایل نقلیه را تنها با استفاده از تصاویر تک‌چشمی و پارامترهای درونی دوربین حل می‌کند. هدف ما در اینجا بازنویسی کامل روابط ریاضی نیست، بلکه توضیح نقش هر بخش است. برای فرمول‌بندی‌های دقیق ریاضی، خواننده را به مقالهٔ اصلی ToosiCubix [۶] ارجاع می‌دهیم.

¹Keypoints



شکل ۱.۲: نقاط کلیدی مورد نظر در این پایان‌نامه که از [۶] ToosiCubix برگرفته شده‌اند.



شکل ۲.۲: نمونه هایی از مکعب های تولید شده توسط ToosiCubix

۱.۵.۲ صورت‌بندی مسئله

چارچوب ToosiCubix هر وسیله نقلیه را با یک مکعب سه‌بعدی مدل می‌کند که با چرخش، انتقال و ابعاد توصیف می‌شود. با داشتن حاشیه‌نویسی دو بعدی بخش‌های خاص خودرو، هدف بازسازی مکعب کامل سه‌بعدی است. این حاشیه‌نویسی‌ها به عنوان قیود هندسی عمل می‌کنند و با ترکیب آن‌ها با پارامترهای دوربین می‌توان مکعب را تا مقیاس بازسازی کرد. برای روابط دقیق تصویرسازی، به مقاله اصلی رجوع شود.

۲.۵.۲ انواع حاشیه‌نویسی و نقش آن‌ها

این چارچوب بر مجموعه متنوعی از حاشیه‌نویسی‌های دو بعدی تکیه دارد که در شکل ۱.۲ نمایش داده شده است و هر کدام هدف خاصی در محدود کردن مکعب دارند:

- **نقاط تماس چرخ با زمین:** این نقاط محل تماس چرخ‌ها با زمین را مشخص می‌کنند و جای‌گذاری جانبی مکعب و طول خودرو را محدود می‌سازند.
- **نقاط روی خط مرکزی:** نشان برند، آنتن یا دیگر ویژگی‌های روی خط مرکزی، قیودی برای تعیین موقعیت

خودرو روی صفحه تقارن ایجاد می‌کنند.

- **لبه‌های عمودی:** مشخص کردن نقاط روی لبه‌های عمودی مکعب، ارتفاع خودرو را به مکعب پیوند می‌زند.

- **گوشه‌ها:** در وسایل نقلیه‌ای مانند کامیون یا اتوبوس، گوشه‌ها قیود قوی فراهم می‌کنند زیرا مستقیماً به رأس‌های مکعب متناظر می‌شوند.

- **خطوط متقارن:** تقارن‌هایی در جلو، عقب و یا بالای خودرو به صورت خطوط حاشیه‌نویسی می‌شوند. تقارن آن‌ها قیود بیشتری نسبت به نقاط منفرد ایجاد می‌کند.

- **بردارهای جهتی:** در مواقعي که سایر ویژگی‌ها پوشانده یا مبهم باشند، بردارهای رو به جلو، بالا یا کنار رسم می‌شوند. این بردارها ابهام جهت‌گیری خودرو را از بین برده و برآورد مکعب را پایدارتر می‌سازند.

(شکل ۲.۲ نمونه‌ای از نتایج این حاشیه‌نویسی‌ها را نشان می‌دهد.)

۳.۵.۲ راهبرد بهینه‌سازی

چارچوب ToosiCubix قیود حاصل از حاشیه‌نویسی‌ها را به معادلاتی تبدیل می‌کند که نقاط دو بعدی را به متناظر سه بعدی شان در مکعب پیوند می‌دهد. حل این دستگاه شامل برآورد تدریجی وضعیت و ابعاد خودرو است. رویکرد coordinate descent به صورت متناوب وضعیت (با استفاده از الگوریتم Perspective-n-Point) و ابعاد/متغیرهای کمکی را به روزرسانی می‌کند. برای جزئیات ریاضی این فرآیند، به مقاله اصلی مراجعه شود.

۴.۵.۲ مدیریت ابهامات

از آنجا که دید تک‌چشمی ذاتاً دچار ابهام مقیاس است، ToosiCubix از توزیع‌های احتمالی ابعاد خودرو استفاده می‌کند. این توزیع‌ها از داده‌های آماری استخراج شده و به صورت گاوی مدل‌سازی می‌شوند. اضافه کردن این توزیع‌ها به فرآیند بهینه‌سازی، اطلاعات ابعاد گمشده را جبران کرده و ابهام مقیاس را کاهش می‌دهد، هرچند در موارد دشوار هنوز به نشانه‌های اضافی نیاز است.

۵.۵.۲ تنظیم دقیق در حوزه پیکسل

پس از دستیابی به یک راه حل اولیه، ToosiCubix یک مرحله اصلاح در فضای تصویر اعمال می‌کند. در این مرحله، خطای تصویرسازی دوباره در حوزه پیکسل کمینه می‌شود تا مکعب دقیق‌تر با حاشیه‌نویسی‌ها منطبق گردد. جزئیات ریاضی این ریزبینی در مقاله اصلی آمده است.

۶.۵.۲ خلاصه

به طور خلاصه، چارچوب ToosiCubix حاشیه‌نویسی‌های پراکنده دو بعدی را به تخمین‌های قابل اعتماد از مکعب‌های سه بعدی تبدیل می‌کند. هر نوع حاشیه‌نویسی قیود خاصی وارد می‌کند و پیکان‌های جهتی به عنوان محافظتی در برابر خطاهای جهت‌گیری عمل می‌کنند. بهینه‌سازی با ترکیب هندسه سه بعدی و دانش پیشین انجام می‌شود و در نهایت یک مرحله ریزبینی در فضای پیکسل نتیجه را دقیق‌تر می‌سازد. برای بررسی کامل روابط ریاضی، به مقاله اصلی رجوع شود.

۶.۲ دستیار نیمه‌خودکار برچسب‌گذاری

روش ToosiCubix [۶] تلاش دستی موردنیاز برای برچسب‌گذاری مکعب‌های سه بعدی را کاهش می‌دهد، اما این فرآیند همچنان زمان بر است. به همین منظور، در این پژوهش یک رویکرد نیمه‌خودکار ارائه می‌شود که در آن مدل‌های یادگیری عمیق نقاط و خطوط کلیدی خودرو را به صورت اولیه پیشنهاد می‌دهند (شکل ۱۰.۲). این مقادیر صرفاً به عنوان تخمین اولیه عمل کرده و کاربر همچنان مسئول تأیید، اصلاح یا تکمیل آن‌هاست. بدین ترتیب، ترکیب سرعت سیستم با دقت کاربر، کیفیت برچسب‌گذاری را حفظ کرده و زمان موردنیاز را به طور چشمگیری کاهش می‌دهد.

برای پشتیبانی از این فرآیند، رابط کاربری ابزار برچسب‌گذاری نیز تغییر یافت تا قابلیت پیشنهاد خودکار و اصلاح تعاملی نقاط و خطوط کلیدی در آن گنجانده شود. شکل ۳.۲ منوی برنامه را قبل و بعد از این تغییرات نشان می‌دهد. این تغییرات باعث شد دستیار نیمه‌خودکار به طور طبیعی در چرخه برچسب‌گذاری ادغام شده و کاربر بتواند با حداقل تغییر در روند کاری خود از آن استفاده کند.

Center: Back	Wheel: Rear Right 69% (Heads: [19, 88, 99])
Center: Front	Wheel: Front Right 29% (Heads: [77, 11, 0])
Center: Top	Wheel: Front Left 1% (Heads: [3, 0, 0])
Wheel: Rear Left	Wheel: Rear Left 0% (Heads: [0, 0, 0])
Wheel: Rear Right	Edge: Front Right 0% (Heads: [0, 0, 0])
Wheel: Front Left	Background 0% (Heads: [0, 0, 0])
Wheel: Front Right	Edge: Rear Right 0% (Heads: [0, 0, 0])
Corner: Rear Left Top	Edge: Front Left 0% (Heads: [0, 0, 0])
Corner: Rear Right Top	Center: Back 0% (Heads: [0, 0, 0])
Corner: Front Left Top	Center: Front 0% (Heads: [0, 0, 0])
Corner: Front Right Top	Edge: Rear Left 0% (Heads: [0, 0, 0])
Edge: Rear Left	Center: Top 0% (Heads: [0, 0, 0])
Edge: Rear Right	Corner: Rear Right Top 0% (Heads: [0, 0, 0])
Edge: Front Left	Corner: Front Right Top 0% (Heads: [0, 0, 0])
Edge: Front Right	Corner: Front Left Top 0% (Heads: [0, 0, 0])
	Corner: Rear Left Top 0% (Heads: [0, 0, 0])

شکل ۳.۲: منوی ابزار برچسب‌گذاری قبل (چپ) و بعد (راست) از اضافه شدن قابلیت‌های نیمه‌خودکار برای پیشنهاد و اصلاح نقاط و خطوط کلیدی.

۷.۲ نتیجه‌گیری

با توجه به بررسی مدل‌های موجود در حوزه بخش‌بندی تصاویر، مدل Segment Anything Model (SAM) به عنوان یک پیشرفت مهم در زمینه استخراج ماسک‌های بخش‌بندی بدون نیاز به آموزش مجدد شناخته شده است. با این حال، این مدل بیشتر بر تولید دقیق ماسک‌ها متمرکز است و هدف آن دسته‌بندی مستقیم بخش‌های تصویر نیست.

در تحقیق حاضر، با الهام از قابلیت‌های SAM در استفاده از ورودی‌های نقطه‌ای، رویکردی توسعه داده شده که به جای بخش‌بندی، به دسته‌بندی قطعات مشخص شده توسط کاربر می‌پردازد. این تقاضا اساسی در هدف، فرصت‌های جدیدی برای کاربردهای دقیق‌تر و هدفمندتر در حوزه تشخیص و طبقه‌بندی فراهم می‌کند. بنابراین، کار حاضر علاوه بر تکمیل و توسعه تحقیقات موجود، چارچوبی برای دسته‌بندی به کمک ورودی‌های هدفمند ارائه می‌دهد که می‌تواند در پروژه‌ها و حوزه‌های مختلف به کار گرفته شود.

فصل ۳

روش کار

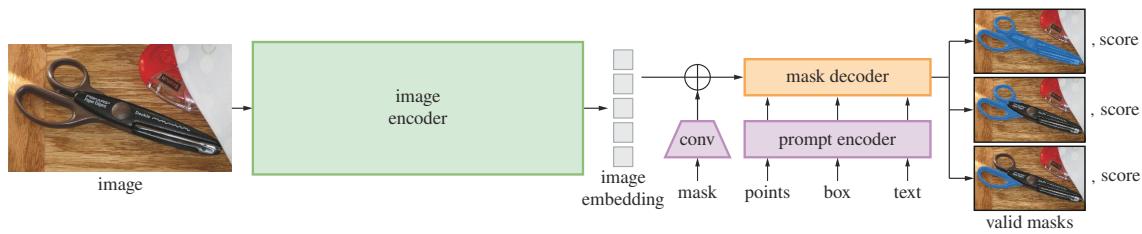
۱.۳ مقدمه

در این فصل، روش تحقیق و معماری مدل‌های پیشنهادی برای دسته‌بندی نقاط کلیدی و خطوط در تصاویر خودرو به طور کامل شرح داده می‌شوند. هدف از این فصل ارائه توضیح دقیق و گام‌به‌گام درباره داده‌ها، ورودی‌ها، معماری مدل، روش آموزش و ارزیابی عملکرد است.

مدل پیشنهادی شامل دو شبکه مجزا است که هر دو از یک معماری پایه مشترک استفاده می‌کنند، اما نوع ورودی و داده‌های آموزشی آن‌ها متفاوت است. مدل اول برای دسته‌بندی خطوط موجود در تصویر و مدل دوم برای دسته‌بندی نقاط کلیدی طراحی شده است.

در ادامه، ابتدا معماری کلی شبکه تشریح شده، سپس داده‌های مورد استفاده و نحوه آماده‌سازی ورودی‌ها توضیح داده می‌شوند. پس از آن، جزئیات مربوط به آموزش مدل‌ها و تنظیمات ابرپارامترها^۱ و در نهایت معیارها و روش‌های ارزیابی مدل‌ها بیان می‌گردد.

¹Hyperparameters



شکل ۱.۳: معماری مدل SAM. منبع: [۳]

SAM ۲.۳

قبل از رسیدن به معماری نهایی خود، ابتدا مدل SAM را بررسی کردیم. ساختار این مدل در شکل ۱.۳ نشان داده شده است. تصویر ورودی ابتدا وارد یک شبکه پایه از نوع ViT می‌شود که می‌تواند یکی از مدل‌های ViT-Small, ViT-Large, ViT-Huge باشد. پس از عبور از این شبکه، ویژگی‌های کلی تصویر به دست می‌آید.

یکی از مزایای اصلی SAM این است که ویژگی‌ها تنها یکبار استخراج می‌شوند و تا ورود تصویر بعدی نیازی به پردازش دوباره شبکه پایه نیست. پس از شبکه پایه، ورودی کاربر (که می‌تواند یک باکس یا یک نقطه باشد) رمزگذاری می‌شود^۱ و سپس وارد رمزگشا^۲ می‌شود. این رمزگشا شامل مجموعه‌ای از تبدیل‌کننده‌ها^۳ است و دو خروجی تولید می‌کند: یکی برای ماسک‌های تولیدشده و دیگری برای IoU ماسک‌ها. با بررسی این ساختار و محدودیت‌های آن، دو محور اصلی برای بهبود مدل خود یافتیم: نوآوری در معماری و استراتژی آموزش. در ادامه، هر یک از این رویکردها در زیر بخش‌های مربوطه شرح داده می‌شوند.

۱.۲.۳ نوآوری

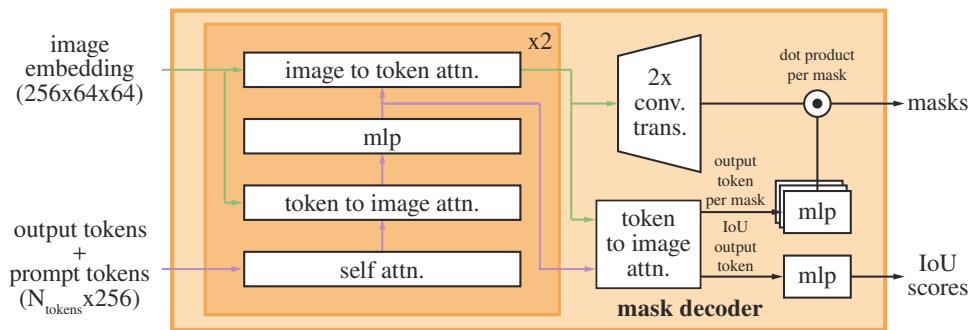
برای دریافت خروجی کلاس‌بندی از SAM، ابتدا بخش‌های مختلف شبکه را بررسی کردیم:

۱. گزینه نخست، اعمال تغییرات بر بخش استخراج ویژگی بود. اما به دلیل پیچیدگی بالای این بخش، امکان دستکاری آن وجود نداشت.

¹Encoder

²Decoder

³Transformer



شکل ۲.۳: معماری رمزگشای SAM. منبع: [۳]

۲. روش دوم می‌توانست تغییر در بخش رمزگذاری کوئری باشد، ولی پس از بررسی دریافتیم که فضای مانور و امکان ایجاد تغییرات چشمگیر در این بخش محدود است.

۳. روش سوم، که در نهایت انتخاب شد، بررسی بخش رمزگشا بود (شکل ۲.۳). کاری که انجام دادیم این بود که یک توکن جدید برای کلاسیندی اضافه کردیم. این توکن به همراه سایر توکن‌ها و تصویر وارد مکانیزم توجه^۱ می‌شود. خروجی رمزگشا سپس از یک پرسپترون چندلایه عبور می‌کند و نتیجه کلاسیندی به دست می‌آید.

مزیت اصلی این ساختار آن است که در عین دستیابی به خروجی کلاسیندی، قابلیت‌های اصلی SAM در زمینه ماسک‌سازی نیز حفظ می‌شوند و می‌توان از آن‌ها در آینده بهره گرفت.

۲.۲.۳ آموزش

برای آموزش SAM چند رویکرد مختلف را دنبال کردیم که در ادامه توضیح داده می‌شوند:

۱. در روش اول، هر تصویر دیتابست به صورت جداگانه پردازش می‌شد و پس از استخراج ویژگی‌ها، نقاط همان تصویر به مدل وارد می‌شد. کل شبکه آموزش داده می‌شد و به دلیل محدودیت منابع، از مدل به عنوان شبکه پایه استفاده کردیم. این روش به سه دلیل شکست خورد:

(آ) مدل در هر بچ تنها به یک تصویر عادت می‌کرد و تنوع کافی در داده‌ها وجود نداشت.

¹Attention

(ب) استفاده از مدل پایه ضعیفتر، یادگیری را کند و باعث کم برآش^۱ شد.

(ج) به دلیل آموزش کل شبکه، مجبور بودیم نقاط را تک تک وارد کنیم تا از توقف برنامه به علت کمبود حافظه گرافیکی جلوگیری شود.

۲. در روش دوم، بخش استخراج ویژگی منجمد شد و در آموزش شرکت نمی‌کرد، ولی محاسبه ویژگی‌ها در زمان اجرا هنوز مشکلات قبلی را ایجاد می‌کرد.

۳. در روش سوم، ابتدا با مدل ViT-Huge تمام ویژگی‌های تصاویر پیش از شروع آموزش استخراج شدند و سپس در زمان اجرا بارگذاری شدند. با توجه به اینکه شبکه پایه آموزش داده نمی‌شد و پردازش اولیه تصویر انجام شده بود، توانستیم واحد پردازش را از «تصویر» به «نقطه» تغییر دهیم. مزایای این تغییر عبارتند از:

(آ) در هر بچ، نقاطی از تصاویر مختلف وجود داشتند و مدل به یک تصویر خاص عادت نمی‌کرد.

(ب) ویژگی‌های پیش‌پردازش شده کیفیت بالاتری داشتند و ورودی مناسبی برای شبکه فراهم می‌کردند.

(ج) آموزش تنها بخشی از شبکه باعث افزایش چشمگیر سرعت آموزش شد.

این روش در نهایت ما را قادر ساخت به دقت ۸۴.۶ دست یابیم.

با وجود بهبودهایی که در بخش نوآوری و آموزش ایجاد شد، همچنان محدودیت‌های جدی وجود داشت: اجرای مدل پایه ViT-Huge برای پیش‌پردازش ویژگی‌ها نیازمند پردازنده‌های گرافیکی قوی بود و برای کاربردهای سبک یا در لبه مناسب نبود. علاوه بر این، حجم داده‌ها و زمان مورد نیاز برای پیش‌پردازش و ذخیره ویژگی‌ها، پیچیدگی عملیاتی بالایی ایجاد می‌کرد. به همین دلایل، تصمیم گرفتیم از این رویکرد صرف نظر کنیم و به سمت معماری سبک‌تر و بهینه‌تر حرکت کنیم که بتواند ویژگی‌های مشابه را بدون محدودیت‌های سخت‌افزاری فراهم کند.

۳.۳ معماری پیشنهادی

هر دو مدل پیشنهادی (مدل خط و مدل نقطه) از یک معماری پایه‌ی مشترک استفاده می‌کنند. این معماری شامل سه بخش اصلی است:

¹Underfitting

۱. استخراج ویژگی‌ها: در این بخش از EfficientNet-B0 به عنوان شبکه‌ی پایه استفاده شده است. این

شبکه که به صورت پیش‌آموزش یافته بر روی مجموعه‌داده‌ی ImageNet [۸] قرار دارد، وظیفه‌ی استخراج ویژگی‌های مکانی و محتوایی تصویر را بر عهده دارد. خروجی این بخش یک نگاشت ویژگی با ابعاد $[B, 1280, H, W]$ است.

۲. تجمعی ویژگی‌ها: برای کاهش ابعاد و تبدیل نگاشت ویژگی به یک نمایش برداری فشرده، از لایه تجمعی Adaptive Average Pooling استفاده می‌شود. این لایه ابعاد مکانی W, H را به ۱,۱ کاهش داده و یک بردار ویژگی با اندازه‌ی ۱۲۸۰ تولید می‌کند.

۳. شبکه پرسپترون چندلایه (MLP): بردار ویژگی به یک شبکه‌ی MLP سه‌لایه داده می‌شود. این شبکه شامل لایه‌های خطی همراه با ReLU و یک لایه Dropout برای کاهش بیش برازش^۲ است. خروجی نهایی این بخش تعداد کلاس‌های مسئله را تعیین می‌کند. در مدل خط، خروجی شامل ۶ کلاس و در مدل نقطه شامل ۱۶ کلاس است.

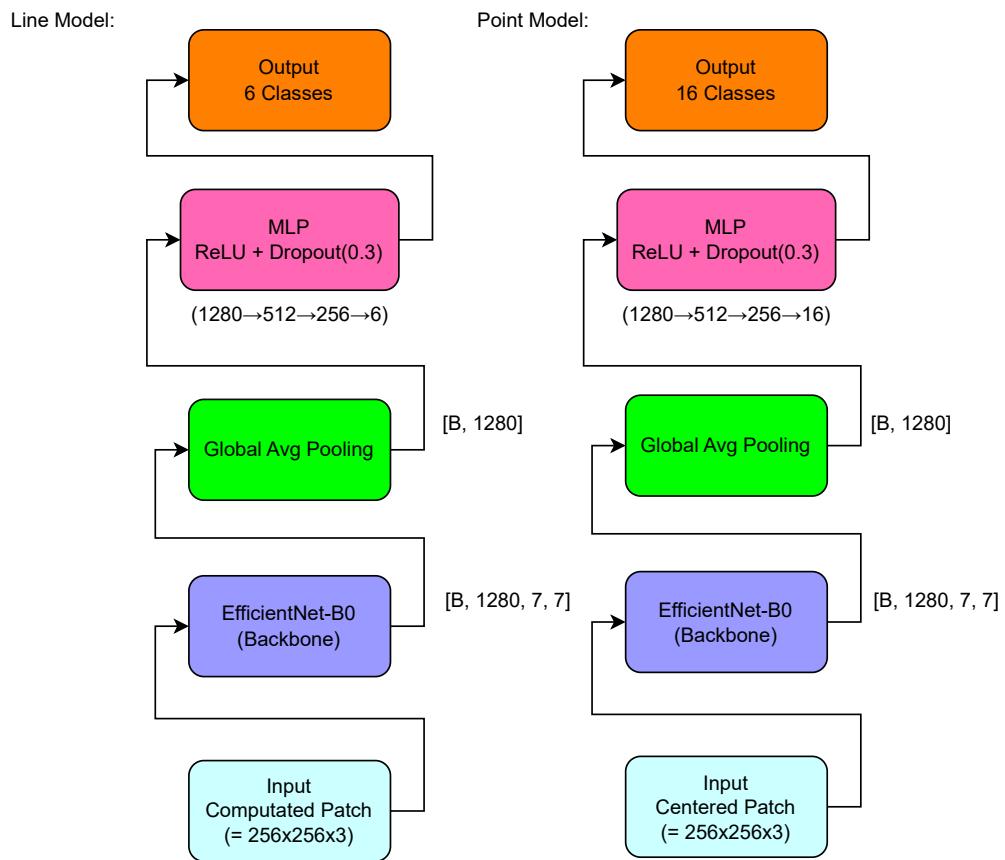
ساختار کلی هر دو مدل در شکل ۳.۳ نمایش داده شده است. همان‌طور که مشاهده می‌شود، تفاوت اصلی میان دو مدل در داده‌های ورودی و تعداد کلاس‌های خروجی است؛ در حالی که ساختار پایه و روند پردازش ویژگی‌ها در هر دو مدل یکسان باقی می‌ماند.

۴.۳ ورودی‌ها و داده‌ها

برای آموزش هر یک از مدل‌های خود، یک مجموعه‌داده فراهم کرده‌ایم که شامل تصاویر و برچسب‌های مربوطه است. این مجموعه‌داده به سه بخش آموزش، تست و ارزیابی تقسیم شده است، به‌گونه‌ای که ۸۰ درصد از داده‌ها برای آموزش، ۱۰ درصد برای تست و ۱۰ درصد برای ارزیابی مدل اختصاص یافته‌اند. جزئیات هر کدام از این مجموعه‌ها در ادامه آمده است.

^۱ لایه‌ی Adaptive Average Pooling میانگین هر نگاشت ویژگی مکانی را محاسبه می‌کند و ابعاد ورودی $W \times H$ را به ابعاد خروجی دلخواه (در اینجا 1×1) کاهش می‌دهد.

²Overfitting



شکل ۳.۳: معماری کلی مدل‌های پیشنهادی مبتنی بر EfficientNet-B0

۱.۴.۳ داده‌های نقاط کلیدی

مجموعه داده نقاط شامل ۱+۱۵ کلاس است که فراوانی آن‌ها در جدول ۱.۳ آمده است. هر برچسب شامل ۳ عدد است که به صورت $(X, Y, \text{کلاس})$ ذخیره شده‌اند. برای آموزش مدل، با استفاده از مختصات نقاط، یک بخش مربعی از تصویر به اندازه $[H, W]$ (معمولاً $[256, 256]$) حول مرکز نقطه ورودی برش داده شده و به مدل داده می‌شود.

جدول ۱.۳: فراوانی نمونه‌ها در هر کلاس نقاط کلیدی.

کلاس	تعداد نمونه‌ها
پس زمینه ^۱	-
چرخ جلو چپ	۲۲۵۴
چرخ جلو راست	۱۱۶۳
چرخ عقب چپ	۲۶۳۵
چرخ عقب راست	۱۴۱۸
مرکز جلو	۶۳۸
مرکز عقب	۸۵۹
مرکز بالا	۱۷۸
گوشه جلو چپ بالا	۰
گوشه جلو راست بالا	۲
گوشه عقب چپ بالا	۹
گوشه عقب راست بالا	۵
لبه عقب چپ	۱۶۳
لبه عقب راست	۸۴
لبه جلو چپ	۱۰۷
لبه جلو راست	۱۳۳

۱.۱.۴.۳ افزایش داده‌ها - نقاط (Data Augmentation)

برای هر تصویر در این مجموعه داده، عملیات افزایش داده‌های زیر اعمال می‌شود:

۱. تغییر رنگ و نور (Color and Brightness Adjustment): روشنایی، کنترast، اشباع و رنگ

^۱ این کلاس در حین آموزش به صورت پویا تولید می‌شود.

تصویر به صورت تصادفی تغییر داده می‌شوند:

`hue=0.02 saturation=0.2, contrast=0.2, brightness=0.2,` (۱.۳)

۲. **تغییر تاری (Gaussian Blur):** تصویر به صورت تصادفی بلور داده می‌شود:

`kernel_size=3, sigma=(0.1, 2.0)` (۲.۳)

۳. **افزایش تیزی (Sharpness Adjustment):** با احتمال $3/10$ ، تیزی تصویر افزایش داده می‌شود:

`sharpness_factor=2` (۳.۳)

۴. **بلور جایگزین (Optional Gaussian Blur):** با احتمال $2/10$ ، یک فیلتر بلور گاووسی دیگر با شعاع ۱ اعمال می‌شود:

`radius=1` (۴.۳)

۵. **تبدیل به تنسور (Tensor Conversion):** تصویر به Tensor تبدیل می‌شود.

۶. **افزودن نویز گوسی (Gaussian Noise):** نویز گوسی به تصویر افزوده می‌شود و مقادیر پیکسل‌ها در بازه $[1, 10]$ محدود می‌شوند:

`std=0.03` (۵.۳)

۷. **نرمال‌سازی (Normalization):** تصاویر نهایی نرمال‌سازی می‌شوند تا میانگین و انحراف معیار کانال‌ها

مطابق مقادیر استاندارد ImageNet [۸] باشند:

$$\mu = [0, 485, 0, 456, 0, 406], \quad \sigma = [0, 229, 0, 224, 0, 225] \quad (6.3)$$

علاوه بر این، قبل از تشکیل هر تصویر، نقاط نیز با نویز گوسی به انحراف معیار ۲ جابجا می‌شوند و برای هر خط، ۵ خط جدید و برای هر نقطه، ۵ نقطه جدید تولید می‌گردد. این کار موجب بهبود توانایی مدل در یادگیری localization و تشخیص دقیق‌تر محل اجزاء می‌شود.

۲.۴.۳ داده‌های خطوط

مجموعه داده خطوط شامل ۶ کلاس است که فراوانی آن‌ها در جدول ۲.۳ آمده است. هر برچسب شامل ۵ عدد است که به صورت (مختصات نقطه اول، مختصات نقطه دوم، label) ذخیره شده‌اند. برای آموزش مدل، با استفاده از مختصات نقاط، بخش مربوطه‌ای از تصویر را برش داده و به مدل می‌دهیم که جزئیات آن در ادامه توضیح داده شده است.

جدول ۲.۳: فراوانی کلاس‌های خطوط در مجموعه داده

کلاس	تعداد نمونه‌ها
پشت متقارن	۱۷۷
بالای متقارن	۱۶۸
جلو متقارن	۸۶
جهت رو به جلو	۷۹
جهت رو به بالا	۲۲
جهت جانبی	۲۱

۱.۲.۴.۳ نحوه برش تصویر برای خطوط

این تابع برای یک خط تعریف شده توسط دو نقطه p_1 و p_2 و ضریب Scale، که طول و عرض نهایی را چند برابر می‌کند، یک مستطیل چرخیده (rotated rectangle) در اطراف خط ایجاد می‌کند و نقاط چهارگوش و نقاط مرکزی جابجا شده (shifted points) را محاسبه می‌نماید. مراحل فرآیند به شرح زیر است:

۱. محاسبه بردار خط و طول آن: ابتدا بردار خط و طول آن بین دو نقطه انتهایی P_{start} و P_{end} محاسبه می‌شود:

$$\vec{L} = P_{\text{end}} - P_{\text{start}}, \quad \ell = \|\vec{L}\| = \sqrt{(x_{\text{end}} - x_{\text{start}})^2 + (y_{\text{end}} - y_{\text{start}})^2} \quad (7.3)$$

۲. بردار واحد خط و بردار عمود بر آن: بردار واحد جهت خط و بردار عمود بر آن برای رسم اضلاع عمودی مستطیل محاسبه می‌شوند:

$$\hat{L} = \frac{\vec{L}}{\ell}, \quad \hat{N} = \frac{(-\Delta y, \Delta x)}{\ell}, \quad \Delta x = x_{\text{end}} - x_{\text{start}}, \quad \Delta y = y_{\text{end}} - y_{\text{start}} \quad (8.3)$$

۳. جابجایی نقاط انتهایی برای مرکز مستطیل: هر نقطه انتهایی برای ایجاد فضای اضافی در طول خط به اندازه $\frac{\ell}{4}$ جابجا می‌شود:

$$P_{\text{start}}^{\text{shift}} = P_{\text{start}} - \hat{L} \cdot \frac{\ell}{4} \cdot \text{scale}, \quad P_{\text{end}}^{\text{shift}} = P_{\text{end}} + \hat{L} \cdot \frac{\ell}{4} \cdot \text{scale} \quad (9.3)$$

۴. محاسبه نقاط گوشه عمود بر خط: طول اضلاع عمود بر خط:

$$h = \frac{\ell}{4} \cdot \text{scale} \quad (10.3)$$

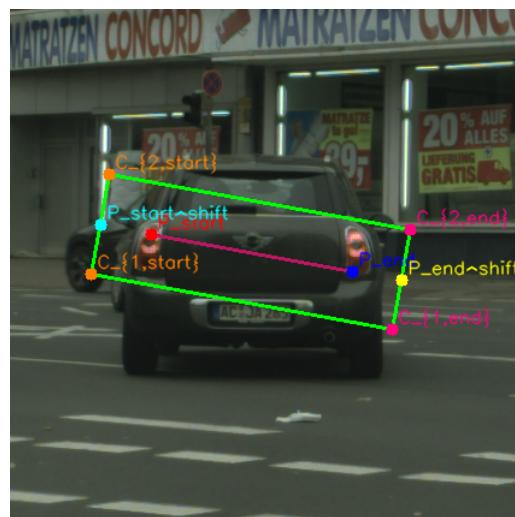
سپس برای هر نقطه جابجا شده، دو نقطه در سمت‌های مختلف عمود رسم می‌کنیم:

$$C_{\backslash} = P_{\text{start}}^{\text{shift}} + \frac{h}{2} \hat{N}, \quad C_{\checkmark} = P_{\text{end}}^{\text{shift}} - \frac{h}{2} \hat{N} \quad (11.3)$$

۵. خروجی تابع: خروجی شامل چهار نقطه گوشه مستطیل و دو نقطه مرکزی جابجا شده است:

$$(C_{\backslash,\text{start}}, C_{\checkmark,\text{start}}, C_{\backslash,\text{end}}, C_{\checkmark,\text{end}}, P_{\text{start}}^{\text{shift}}, P_{\text{end}}^{\text{shift}}) \quad (12.3)$$

با استفاده از نقاط بدست آمده، یک باکس از تصویر برش داده می‌شود و به عنوان ورودی به مدل داده می‌شود. نمونه‌ای از این عملیات را میتوانید در شکل ۴.۳ مشاهده کنید.



شکل ۴.۳: نمونه‌ای از برش عکس با ورودی یک خط

۲.۲.۴.۳ افزایش داده‌ها - خطوط (Data Augmentation)

برای بهبود تعمیم‌پذیری مدل و جلوگیری از بیش‌برازش، در این پژوهش مجموعه‌ای از روش‌های افزایش داده‌ها به کار گرفته شده است. این روش‌ها با استفاده از کتابخانه‌ی *Albumentations* [۹] پیاده‌سازی شده‌اند و بر روی پچ‌های استخراج شده از تصاویر اعمال می‌گردند. مراحل و نوع تغییرات به شرح زیر است:

۱. **وارون‌سازی افقی (Horizontal Flip):** با احتمال ۵٪ تصویر به صورت افقی وارونه می‌شود. این

عمل موجب افزایش مقاومت مدل نسبت به جهت‌گیری اشیاء می‌شود.

۲. **تبديل آفین (Affine Transform):** با احتمال ۷٪ تصویر تحت تغییرات هندسی شامل جابجایی

جزئی (تا ۵٪ ابعاد)، تغییر مقیاس در بازه [۰.۹۵, ۱.۰۵] و چرخش در بازه [-۵°, ۵°] قرار می‌گیرد. این

تغییرات به شبیه‌سازی تغییر زاویه‌ی دید و مقیاس کمک می‌کنند.

۳. **تغییرات رنگ (Color Jitter):** با احتمال ۵٪ روشنایی، کنتراست، اشباع و هیو (Hue) تصویر

به صورت جزئی تغییر داده می‌شود:

$$\text{brightness} = 0,1, \quad \text{contrast} = 0,1, \quad \text{saturation} = 0,1, \quad \text{hue} = 0,0,1 \quad (13.3)$$

۴. تاری گوسی (**Gaussian Blur**): با احتمال $2/0$ فیلتر تاری گوسی با کرنل 3×3 و سیگما در بازه $[0,1,1/0]$ اعمال می‌شود.

۵. شارپ‌سازی یا بلور ساده (**Sharpen/Blur**): با احتمال کلی $2/0$ یکی از دو عمل زیر انتخاب و اعمال می‌شود:

- شارپ‌سازی ملایم با $\alpha \in [0/1, 0/9]$ و روشنایی در بازه $[0/9, 1/0]$
- بلور ساده با کرنل 3×3

۶. نویز گوسی (**Gaussian Noise**): با احتمال $2/0$ نویز گوسی با میانگین در بازه $[1/0, 0/0]$ و انحراف معیار در بازه $[0/0, 0/15]$ به تصویر افزوده می‌شود.

۷. نرمال‌سازی (**Normalization**): تصاویر نهایی نرمال‌سازی می‌شوند تا میانگین و انحراف معیار کانال‌ها مطابق مقادیر استاندارد ImageNet [۸] باشند:

$$\mu = [0, 485, 0, 456, 0, 406], \quad \sigma = [0, 229, 0, 224, 0, 225] \quad (14.3)$$

۸. تبدیل به تنسور (**ToTensorV2**): در پایان، تصویر به فرمت تنسور PyTorch تبدیل می‌شود تا به عنوان ورودی مدل مورد استفاده قرار گیرد.

به کارگیری این مجموعه از روش‌های افزایش داده، تنوع تصاویر ورودی را افزایش داده و موجب می‌گدد مدل نسبت به تغییرات نوری، نویز، مقیاس و جهت‌گیری مقاوم‌تر باشد.

۵.۳ روش آموزش

۱.۵.۳ تنظیمات مشترک

برای تمامی مدل‌ها از بهینه‌ساز Adam استفاده شد. نرخ یادگیری اولیه برابر با 4×10^{-4} و وزن weight decay برابر با $10^{-4} \times 1$ تنظیم شد. برای کاهش تدریجی نرخ یادگیری، از برنامه زمان‌بندی Cosine Annealing LR استفاده گردید. تابع خطا (loss function) از نوع CrossEntropyLoss با label smoothing برابر با 1.0 است. انتخاب شد.

۲.۵.۳ تفاوت‌های آموزش برای هر مدل

مدل تشخیص نقاط:

- تعداد epoch برابر با 10 است.
- scheduler برای 10 مرحله تنظیم شده است.
- تابع خطا با $\text{label smoothing} = 0.1$ و وزن‌های خاص که برای هر کلاس تنظیم شده است بر اساس مقدار فراوانی که در جدول ۱.۳ ذکر شده نسبت به فراوانی کل که این وزن‌ها برای متعادل کردن داده‌های نابرابر کلاس‌ها استفاده شده‌اند.

مدل تشخیص خطوط:

- تعداد epoch برابر با 5 است.
- scheduler برای 10 مرحله تنظیم شده است.
- تابع خطا بدون وزن‌دهی خاص و صرفاً با $\text{label smoothing} = 0.1$ استفاده می‌شود.

۶.۳ تحلیل چند مقیاسه

در زمان آموزش، مدل با استفاده از مقیاس‌های مختلف و انواع داده‌افزونی‌ها آموزش داده شد. نمونه‌ای از مقیاس‌های موردنظر ما را می‌توانید در شکل ۵.۳ مشاهده کنید. هدف از این کار این است که مدل به تغییرات اندازه و نوع داده‌ها عادت کند. نکته‌ی مهم این است که در تمامی حالات استنتاج، تنها یک مدل آموزش دیده در اختیار داریم، اما نحوه‌ی ارائه‌ی ورودی به مدل متفاوت است. این تفاوت در ورودی‌ها می‌تواند بر کیفیت و ثبات پیش‌بینی تأثیرگذار باشد، که در فصل بعد به آن خواهیم پرداخت.

۱.۶.۳ ساده

در حالت ساده، کاربر با انتخاب یک نقطه بر روی تصویر، یک بخش از عکس برش داده می‌شود و به مدل داده می‌شود. خروجی مدل پس از اعمال Softmax^۱ به کاربر نمایش داده می‌شود. این روش ساده‌ترین حالت استنتاج است و سرعت بالایی دارد، اما ممکن است نسبت به نویز یا مقیاس‌های متفاوت حساس باشد.

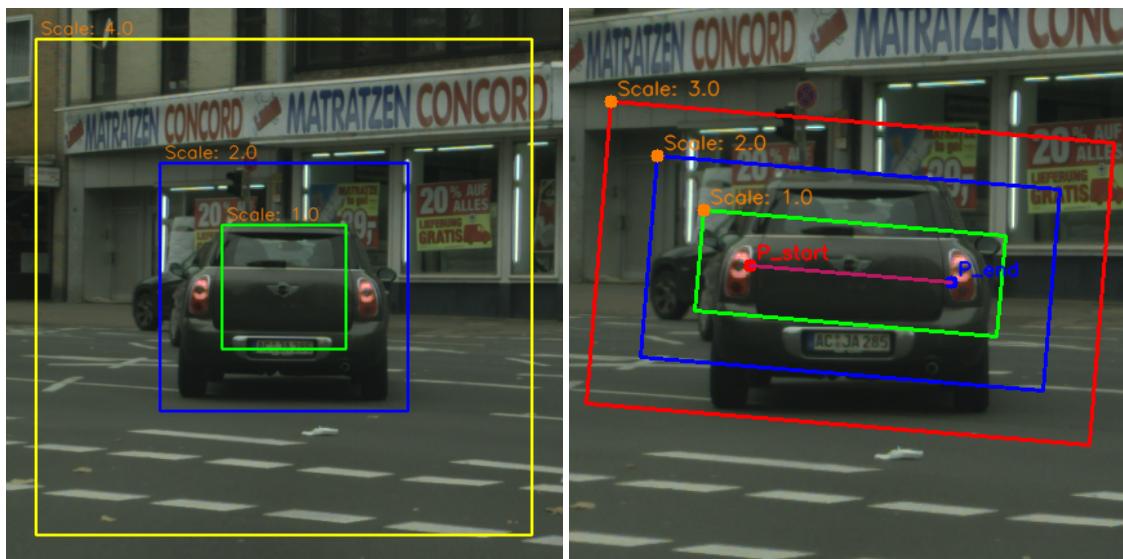
۲.۶.۳ میانگین

در روش میانگین، پس از انتخاب نقطه توسط کاربر، سه بخش با مقیاس‌های متفاوت از تصویر استخراج می‌شود و به مدل داده می‌شود. نتایج سه بخش با هم جمع شده و میانگین گرفته می‌شود. سپس نتیجه نهایی به کاربر نمایش داده می‌شود. این روش باعث افزایش پایداری پیش‌بینی در مقابل تغییرات مقیاس تصویر می‌شود و می‌تواند نتایج دقیق‌تری نسبت به حالت ساده ارائه دهد.

۳.۶.۳ ماکسیمم

در روش ماکسیمم، مشابه روش میانگین، سه بخش با مقیاس‌های متفاوت استخراج و به مدل داده می‌شوند. سپس نتایج آن‌ها با هم جمع شده و از میان آن‌ها بیشینه انتخاب می‌شود تا نتیجه نهایی مشخص گردد. این روش تابع Softmax خروجی خام مدل را به مقادیر بین ۰ و ۱ تبدیل می‌کند به طوری که مجموع آن‌ها برابر ۱ باشد و بتوان آن را به صورت احتمال تفسیر کرد.

به ویژه زمانی مفید است که مدل در مقیاس‌های بزرگ‌تر یا کوچک‌تر عملکرد بهتری نشان دهد، زیرا نتیجه نهایی تحت تأثیر بیشترین پاسخ مدل قرار می‌گیرد.



شکل ۵.۳: نمونه‌ای از ورودی با درنظر گرفتن چند مقیاس.

۷.۳ روش ارزیابی

۱.۷.۳ معیارهای ارزیابی

با توجه به محدودیت داده‌ها در برخی کلاس‌ها، مادقت^۱ و بازیابی^۲ را به صورت جداگانه برای هر کلاس بررسی می‌کنیم. همچنین، دقت کلی مدل روی تمام داده‌ها نیز گزارش می‌شود. سایر معیارهای مرسوم مانند scoreF-نیز برای هر مدل محاسبه شده‌اند تا تصویر کاملی از عملکرد ارائه شود.

¹Precision ²Recall

۲.۷.۳ روش آزمایش

داده‌ها به مجموعه‌های آموزش، ارزیابی و تست تقسیم شده‌اند. ارزیابی روی مجموعه ارزیابی در پایان هر اپک انجام می‌شود تا روند یادگیری مدل را پایش کنیم، در حالی که مجموعه تست تنها در پایان آموزش برای گزارش نهایی عملکرد مدل مورد استفاده قرار می‌گیرد. این روش اطمینان می‌دهد که ارزیابی‌های دوره‌ای روی داده‌های آموزشی انجام نمی‌شوند و تست نهایی بازتاب دقیقی از عملکرد مدل روی داده‌های دیده‌نشده ارائه می‌دهد.

۸.۳ جمع‌بندی

در این فصل، روش کار مدل‌های پیشنهادی برای دسته‌بندی خطوط و نقاط کلیدی خودروها به طور کامل تشریح شد. ابتدا معماری پایه‌ی مشترک مدل‌ها شامل استخراج ویژگی با EfficientNet-B0، تجمعی ویژگی‌ها با Adaptive Average Pooling و طبقه‌بندی نهایی با شبکه‌ی MLP توضیح داده شد. سپس نحوه‌ی آماده‌سازی داده‌ها، برش تصاویر برای خطوط و نقاط کلیدی، و همچنین روش‌های افزایش داده‌ها برای بهبود تعمیم‌پذیری مدل‌ها ارائه گردید.

مدل‌ها با استفاده از مجموعه‌داده‌های تفکیک شده برای آموزش، تست و ارزیابی آموزش داده شدند و تفاوت اصلی آن‌ها در نوع ورودی و تعداد کلاس‌های خروجی بود، در حالی که ساختار پایه و روند پردازش ویژگی‌ها یکسان باقی ماند. به‌طور کلی، این فصل چارچوب جامعی از معماری، داده‌ها و روش‌های آموزشی ارائه کرد که زمینه را برای تحلیل و ارزیابی عملکرد مدل‌ها در فصل‌های بعدی فراهم می‌کند.

فصل ۴

بحث و نتایج

۱.۴ مقدمه

در این فصل، نتایج چند نوع استنتاج^۱ مختلف از مدل به دست آمده که در فصل قبل تشریح شد را بعد از آموزش را بررسی می‌کنیم. به انواع معیارهایی که در این حوزه برای ارزیابی مدل‌ها نیاز است می‌پردازیم سپس به تفاوت نتایج آن‌ها بر روی دو مجموعه ارزیابی و تست که به صورت رندوم از مجموعه داده‌های اصلی جدا شده‌اند، می‌پردازیم. هدف از این بررسی، سنجش پایداری مدل نسبت به ورودی‌های مختلف و بهبود دقت پیش‌بینی است.

۲.۴ معیارهای ارزیابی مدل

۱.۲.۴ سرعت

یکی از معیارهای مهم در ارزیابی مدل، سرعت پردازش آن است. برای سنجش این معیار، زمان استنتاج مدل بر روی نمونه‌های تست اندازه‌گیری شده است. تمامی آزمایش‌ها بر روی سیستمی با مشخصات زیر انجام شده است:

¹Inference

- پردازنده: 2.80 GHz 11th Gen Intel(R) Core(TM) i7-1165G7
- حافظه رم: 16 گیگابایت (قابل استفاده: 15.8 گیگابایت)
- حافظه نهان: ما در بنچمارک تغییرات اولیه در تصاویر را در حافظه نهان ذخیره می‌کنیم که نیاز نباشد هر دفعه دوباره محاسبه شود.

نتایج بنچمارک

جدول ۱.۴: زمان استنتاج مدل برای سه حالت مختلف بر روی تصاویر تست

حالات	زمان به ازای هر ورودی (میلی ثانیه)
961	ساده
982	میانگین
979	ماکسیمم

۲.۲.۴ دقت (Precision)

Precision نشان می‌دهد چه درصدی از نمونه‌هایی که مدل به عنوان مثبت تشخیص داده، واقعاً مثبت بوده‌اند:

$$Precision = \frac{TP}{TP + FP} \quad (1.4)$$

۳.۲.۴ بازیابی (Recall)

Recall نشان می‌دهد چه درصدی از نمونه‌های مثبت واقعی توسط مدل به درستی شناسایی شده‌اند:

$$Recall = \frac{TP}{TP + FN} \quad (2.4)$$

برای داده‌های چندکلاسه، دو روش رایج برای محاسبه میانگین این معیارها وجود دارد:

تمامی کلاس‌ها با هم در نظر گرفته می‌شوند و مقادیر TP, FP, FN برای همه

کلاس‌ها جمع می‌شوند:

$$Precision_{\text{micro}} = \frac{\sum_i TP_i}{\sum_i (TP_i + FP_i)}, \quad Recall_{\text{micro}} = \frac{\sum_i TP_i}{\sum_i (TP_i + FN_i)} \quad (3.4)$$

ابتدا معیارها برای هر کلاس به صورت جداگانه محاسبه می‌شوند و سپس میانگین

آن‌ها گرفته می‌شود:

$$Precision_{\text{macro}} = \frac{1}{N} \sum_{i=1}^N Precision_i, \quad Recall_{\text{macro}} = \frac{1}{N} \sum_{i=1}^N Recall_i \quad (4.4)$$

F1 امتیاز ۴.۲.۴

F1-Score میانگین هماهنگ^۱ بین دقت و بازیابی است و تعادلی میان این دو معیار ایجاد می‌کند:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5.4)$$

در حالت چندکلاسه نیز نسخه‌های macro و micro تعریف می‌شوند:

با استفاده از دقت و بازیابی در حالت micro محاسبه می‌شود:

$$F1_{\text{micro}} = 2 \times \frac{Precision_{\text{micro}} \times Recall_{\text{micro}}}{Precision_{\text{micro}} + Recall_{\text{micro}}} \quad (6.4)$$

ابتدا F1 هر کلاس به صورت جداگانه محاسبه می‌شود و سپس میانگین گرفته می‌شود:

$$F1_{\text{macro}} = \frac{1}{N} \sum_{i=1}^N \left(2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \right) \quad (7.4)$$

¹Harmonic Mean

۳.۴ نتایج

در این بخش نتایج حاصل از ارزیابی دو مدل نقاط و خط ارائه شده است. معیارهای ارزیابی شامل دقت، بازیابی و امتیاز F1 به صورت میانگین های macro و micro محاسبه شده اند. سه حالت مختلف ساده، میانگین و ماکسیمم مورد بررسی قرار گرفته اند تا تاثیر روش های تجمعی روی عملکرد مدل مشخص شود.

۱.۳.۴ مدل نقاط

جدول ۲.۴ نتایج مدل نقاط را نشان می دهد. همان طور که دیده می شود، در حالت ماکسیمم بالاترین مقدار در تمام معیارها به دست آمده است. به طور خاص، F1 از Micro ۰.۸۵ در حالت ساده به **۰.۸۸** در حالت ماکسیمم افزایش یافته است. از طرف دیگر، در معیارهای Macro نیز روند مشابهی دیده می شود؛ امتیاز Macro از **۰.۷۱** به **۰.۶۴** رسیده است که نشان می دهد عملکرد مدل در شناسایی کلاس های کمتر متوازن نیز بهبود یافته است.

جدول ۲.۴: مقایسه نتایج مدل نقاط روی مجموعه ارزیابی و تست

ماکسیمم	میانگین	ساده	معیار
۰.۸۸	۰.۸۷	۰.۸۵	Micro F1
۰.۸۸	۰.۸۷	۰.۸۵	Micro Precision
۰.۸۸	۰.۸۷	۰.۸۵	Micro Recall
۰.۷۱	۰.۷۰	۰.۶۴	Macro F1
۰.۶۸	۰.۶۶	۰.۶۲	Macro Precision
۰.۷۴	۰.۷۳	۰.۶۶	Macro Recall

۲.۳.۴ مدل خط

نتایج مدل خط در جدول ۳.۴ آمده است. در اینجا نیز مشاهده می شود که حالت های میانگین و ماکسیمم نسبت به حالت ساده عملکرد بهتری دارند. به طور خاص، F1 از Micro ۰.۷۳ در حالت ساده به **۰.۸۲** در حالت میانگین رسیده است. همچنین در معیارهای Macro نیز حالت میانگین بالاترین مقادیر را ثبت کرده است. این

موضوع نشان می‌دهد که برای مدل خط، استفاده از روش میانگین‌گیری نسبت به سایر حالت‌ها تاثیر مثبت بیشتری بر دقت و بازیابی داشته است.

جدول ۳.۴: مقایسه نتایج مدل خطوط روی مجموعه ارزیابی و تست

ماکسیمم	میانگین	ساده	معیار
0.79	0.82	0.73	Micro F1
0.79	0.82	0.73	Micro Precision
0.79	0.82	0.73	Micro Recall
0.69	0.70	0.64	Macro F1
0.68	0.70	0.63	Macro Precision
0.69	0.70	0.64	Macro Recall

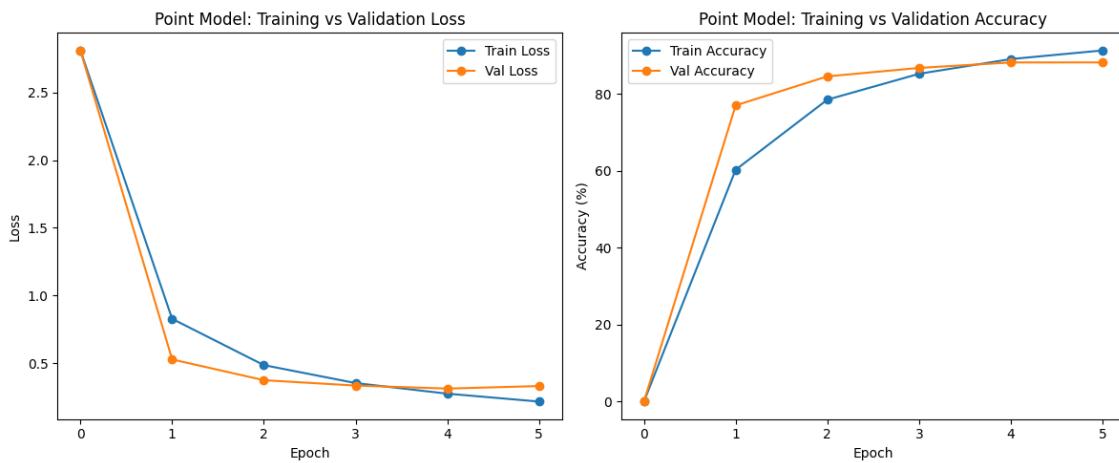
به طور کلی می‌توان نتیجه گرفت که در مدل نقاط، حالت ماکسیمم بهترین عملکرد را ارائه می‌دهد، در حالی که در مدل خط، حالت میانگین موثرتر است. این تفاوت نشان می‌دهد که انتخاب روش تجمعیع باید متناسب با نوع مدل و ساختار ویژگی‌ها انجام شود.

۳.۳.۴ نمودارهای تابع هزینه و دقت

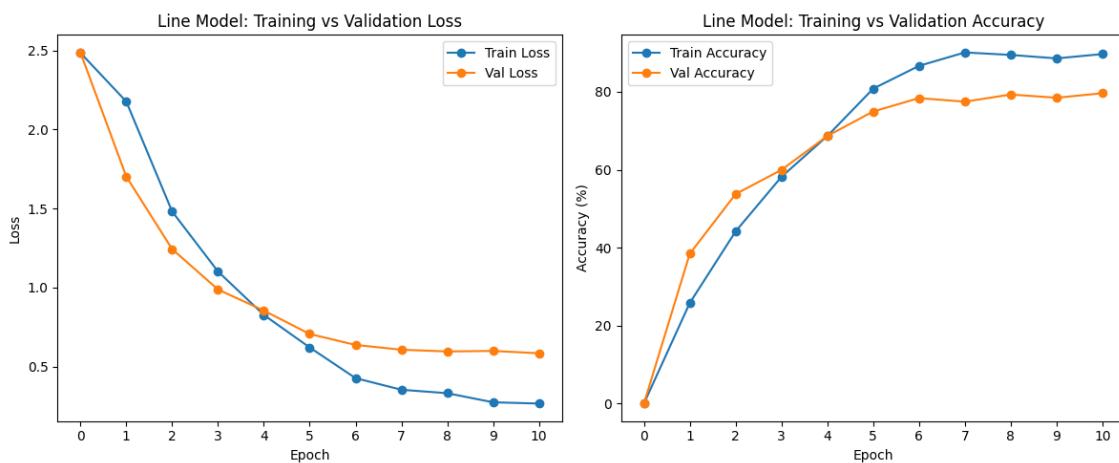
برای درک بهتر روند یادگیری مدل‌ها، در این بخش نمودارهای تغییرات مقدار تابع هزینه و دقت مدل نقاط و مدل خط در طول آموزش ارائه شده است. همان‌طور که در شکل ۲.۴ و شکل ۱.۴ مشاهده می‌شود، این نمودارها شامل مقدار تابع هزینه و دقت در مجموعه آموزش و مجموعه ارزشیابی (Training vs. Validation) هستند. این نمودارها علاوه بر نشان دادن همگرایی مدل‌ها، دیدی از سرعت یادگیری، پایداری و توانایی تعمیم مدل‌ها به داده‌های دیده‌نشده فراهم می‌کنند.

۴.۴ نمونه‌های اجرا

ما مدل‌های خود را با برنامه برچسب‌گذاری *ToosiCubix* [۶] ادغام کردیم که یکی از اهداف پروژه نیز بود نمونه‌هایی از کمک رسانی مدل ما را می‌توانید در شکل ۳.۴ و شکل ۴.۴ مشاهده کنید.



شکل ۲۰.۴: نمودارهای تابع هزینه (چپ) و دقت (راست) مدل نقاط در طول آموزش. خطوط نشان‌دهنده مقادیر مربوط به مجموعه آموزش و مجموعه ارزشیابی هستند و روند همگرایی مدل و بهبود دقت آن را نمایش می‌دهند.

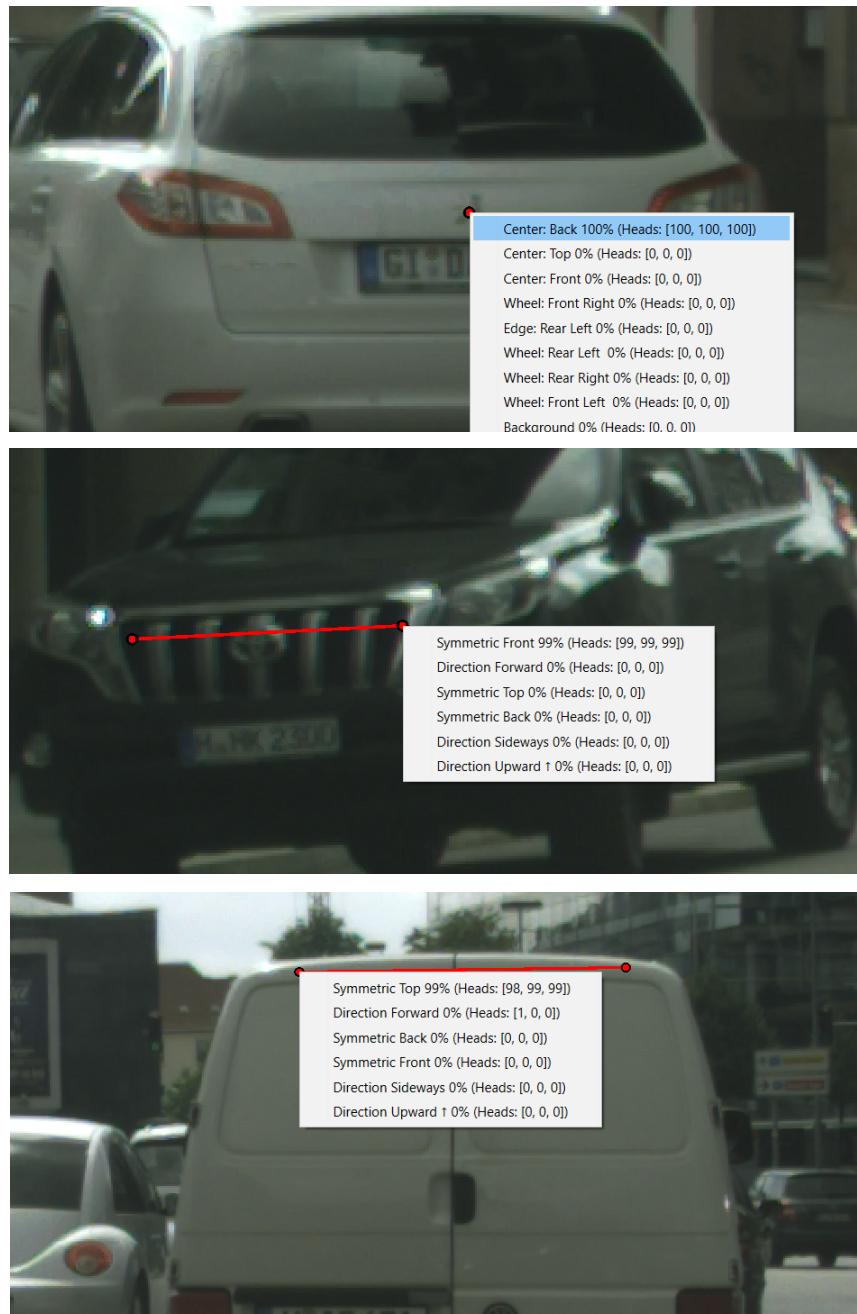


شکل ۲۰.۴: نمودارهای تابع هزینه (چپ) و دقت (راست) مدل خط در طول آموزش. نمودارها شامل مقادیر مربوط به مجموعه آموزش و مجموعه ارزشیابی هستند و روند همگرایی و افزایش دقت مدل خط را نشان می‌دهند.

۵.۴ جمع‌بندی

بر اساس نتایج به دست آمده از آزمایش‌ها، می‌توان جمع‌بندی زیر را ارائه داد:

- در مدل نقاط، استفاده از روش **ماکسیمم** بهترین عملکرد را از نظر تمامی معیارهای ارزیابی (دقت، بازیابی F1) ارائه داده است. این موضوع نشان می‌دهد که مدل نقاط در مواجهه با مقیاس‌های مختلف، زمانی بهترین نتیجه را می‌دهد که بیشینه پاسخ مدل انتخاب شود.



شکل ۳.۴: نمونه‌هایی از ادغام مدل در داخل برنامه برچسب‌گذاری (قسمت اول)



شکل ۴.۴: نمونه‌هایی از ادغام مدل در داخل برنامه برچسب‌گذاری (قسمت دوم)

- در مدل خط، روش میانگین بالاترین دقت و پایداری را داشته است. این نتیجه بیانگر آن است که در این نوع مدل، تجمعیع پاسخ‌ها از مقیاس‌های مختلف و در نظر گرفتن میانگین آن‌ها باعث بهبود عملکرد کلی می‌شود.
- از نظر سرعت، هر سه حالت ساده، میانگین و ماکسیمم تفاوت زیادی با یکدیگر ندارند و اختلاف آن‌ها در حد چند میلی‌ثانیه است. بنابراین، انتخاب روش تجمعیع بیشتر باید بر اساس معیارهای دقت و بازیابی انجام شود تا سرعت.

به طور کلی می‌توان نتیجه گرفت که روش تجمعیع بهینه به نوع مدل بستگی دارد: برای مدل نقاط، روش ماکسیمم مناسب‌تر است، در حالی که برای مدل خط، روش میانگین بهترین نتیجه را ارائه می‌دهد. این موضوع نشان می‌دهد که در طراحی و پیاده‌سازی سیستم‌های مبتنی بر یادگیری عمیق، انتخاب استراتژی استنتاج باید متناسب با ویژگی‌های مدل و داده‌ها انجام گیرد.

فصل ۵

نتیجه‌گیری

۱.۵ مقدمه

این فصل به جمع‌بندی دستاوردهای پژوهش حاضر اختصاص دارد. ابتدا مهم‌ترین یافته‌ها مرور شده، سپس پیامدهای عملی و محدودیت‌های پژوهش بررسی می‌شوند و در پایان مسیرهای توسعه و تحقیقات آینده پیشنهاد خواهند شد.

۲.۵ جمع‌بندی

در این پایان‌نامه، دو مدل مبتنی بر نقاط کلیدی و خطوط برای دسته‌بندی بخش‌های مختلف خودرو بررسی شدند. به منظور بهبود پایداری و دقت، سه روش تجمعی ویژگی‌ها شامل ساده، میانگین و ماکسیمم ارزیابی گردیدند. یافته‌های اصلی عبارت‌اند از:

- در مدل نقاط کلیدی، روش ماکسیمم بهترین عملکرد را در معیارهای دقت، بازیابی و F1-score ارائه داد.
- در مدل خطوط، روش میانگین نتایج پایدارتری نشان داد و از نظر دقت کلی بر سایر روش‌ها برتری داشت.

- زمان پردازش در هر سه حالت تقریباً یکسان بود (حدود ۲۹ ثانیه برای هر تصویر با ۳۰ نقطه کلیدی)، بنابراین انتخاب روش تجمعی باید بر اساس کیفیت نتایج و نه هزینه محاسباتی انجام گیرد.

نتایج نشان می‌دهد که استفاده از ورودی‌های همراه با معماری‌های شبکه‌های عصبی می‌تواند مدل‌هایی کارآمد و مناسب برای اجرا در محیط‌های با محدودیت سخت‌افزاری ایجاد کند.

۳.۵ تحقیقات آینده

چشم‌اندازهای پژوهش آینده شامل موارد زیر است:

- مدل ما در تشخیص جهات خودرو ضعیف عمل می‌کند در حالی که خطوط مرتبط به خودرو را به درستی پیش‌بینی می‌کند خوب است در تحقیقات بعدی به سمت بدست آوردن مجموعه داده‌ای مناسب روی این جهات و همچنین بهینه کردن مدل برای تشخیص این جهات برویم
- جمع آوری داده‌های بیشتر برای دست آوردن دقت بیشتر روی مدل‌ها
- بهره‌گیری از معماری‌های پیشرفته‌تر نظری Transformer برای افزایش دقت و تعمیم‌پذیری.
- ارتقای توانایی مدل در شناسایی نقاط کلیدی و خطوط خودرو در تصاویر با مقیاس‌ها و زاویه‌های دید مختلف.

با پیگیری این مسیرها، انتظار می‌رود رویکرد پیشنهادی به سامانه‌هایی منجر شود که علاوه بر دقت بالا، از نظر کارایی و قابلیت کاربرد در شرایط مختلف عملکرد مطلوبی داشته باشند.

كتاب نامه

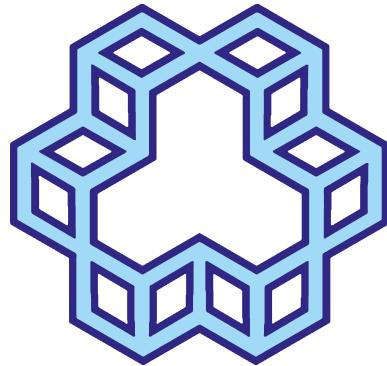
- [1] Bearman, Amy, Russakovsky, Olga, Ferrari, Vittorio, and Fei-Fei, Li. What's the point: Semantic segmentation with point supervision. in *European Conference on Computer Vision*, pp. 549–565. Springer, 2016.
- [2] Papadopoulos, Dim P, Uijlings, Jasper RR, Keller, Frank, and Ferrari, Vittorio. Extreme clicking for efficient object annotation. in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4930–4939, 2017.
- [3] Kirillov, Alexander, Mintun, Eric, Ravi, Nikhila, Mao, Hanzi, Rolland, Chloe, Gustafson, Laura, Xiao, Tete, Whitehead, Spencer, Berg, Alexander C, Lo, Wan-Yen, Dollár, Piotr, and Girshick, Ross. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.
- [4] Russell, Bryan C, Torralba, Antonio, Murphy, Kevin P, and Freeman, William T. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77:157–173, 2008.
- [5] Dutta, Abhishek and Zisserman, Andrew. The via annotation software for images, audio and video. in *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 2276–2279, 2019.
- [6] Nasihatkon, Behrooz, Resani, Hossein, and Mehrzadian, Amirreza. Toosicubix: Monocular 3d cuboid labeling via vehicle part annotations, 2025.
- [7] Tan, Mingxing and Le, Quoc. Efficientnet: Rethinking model scaling for convolutional neural networks. in *Proceedings of the 36th International Conference on Machine Learning*, pp. 6105–6114. PMLR, 2019.
- [8] Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009.

- [9] Buslaev, Alexander, Iglovikov, Vladimir I, Khvedchenya, Eugene, Parinov, Alex, Druzhinin, Mikhail, and Kalinin, Alexandr A. Albumentations: Fast and flexible image augmentations. *Information*, 11(2):125, 2020.

Abstract

This thesis presents a lightweight, car part classification model designed for interactive image-based annotation tasks. Given a user-selected point or line on a car image, a local image patch centered on the input location is extracted and passed to a neural network for classification. The main motivation and usage of this work is to speed up the labeling process during the annotation of different vehicle parts. The model uses an EfficientNet backbone for feature extraction, followed by a simple multi-layer perceptron (MLP) head to predict the class of the target point (e.g., front left tire, center points, edge points). This approach avoids full-scene processing, focusing instead on relevant local context, which significantly reduces computational cost while maintaining competitive accuracy. Experiments on a custom dataset of annotated car part keypoints demonstrate that this method is fast, simple to deploy, and effective for on-demand vehicle part identification in real-world scenarios. In particular, the point-based model achieved a micro precision of **0.88**, while the line-based model achieved **0.82**, highlighting the effectiveness of the proposed approach across both interactive settings.

Keywords Car Part Recognition, Interactive Environment, Deep Neural Networks, Computer Vision, Vehicle Component Detection



K. N. Toosi University of Technology
Faculty of Computer Engineering

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of Bachelor of Science (B.Sc.)
in Computer Engineering

Car Part Recognition in an Interactive Environment Using Deep Neural Networks

By:
Ali Dashtbozorg

Supervisor:
Dr. Behrooz Nasihatkon

Summer 2025