



دانشگاه ملایر

دانشکده فنی مهندسی

گروه مهندسی کامپیوتر

موضوع

# تشخیص ترافیک در تصاویر ماهواره ای به کمک پردازش تصویر و یادگیری عمیق

دانشجو

حسن احمدی دوگر

استاد راهنما

جناب آقای دکتر حمیدرضا افتخاری

تابستان ۱۴۰۳

## تقدیر و تشکر

با کمال احترام و ادب، بدین وسیله از تمامی عزیزان و بزرگوارانی که در این مسیر همراه و حامی من بودند، صمیمانه تقدیر و تشکر می‌کنم.

ابتدا از خانواده‌ام، به ویژه پدر و مادرم، که با محبت و فداکاری‌های بی‌پایانشان همیشه پشتوانه‌ای قوی برای من بودند، سپاسگزارم. آنها با حمایت‌های بی‌دریغ و صبر و تحمل بی‌نظیر خود، به من انگیزه و امید دادند تا به اهدافم نزدیک‌تر شوم و در تمامی مراحل زندگی‌ام به من اعتماد کردند.

همچنین از اساتید گرانقدر و بزرگوارم که با علم، دانش، و راهنمایی‌های سازنده‌شان به من کمک کردند تا مسیر علم و دانش را بهتر بشناسم و پیشرفت کنم، عمیقاً قدردانی می‌کنم. تدریس با شور و اشتیاق و انتقال تجربه‌های گرانبهای آنان، نقش بسزایی در موفقیت من داشته و همواره از راهنمایی‌های ایشان بهره‌مند شدم.

بدون شک، این دستاوردها نتیجه تلاش‌های صادقانه و پشتیبانی‌های بی‌دریغ خانواده و اساتید عزیزم است و من همیشه به خاطر این نعمت‌ها و محبت‌ها شکرگزار خواهم بود.

با احترام و سپاس بی‌پایان،

حسن احمدی دوگر

## چکیده

با رشد سریع شهرنشینی و افزایش تعداد وسایل نقلیه، مدیریت ترافیک به یکی از چالش‌های جدی کلان‌شهرها تبدیل شده است. روش‌های سنتی نظارت بر ترافیک دیگر پاسخگوی پیچیدگی و سرعت تغییرات این حوزه نیستند. در این پژوهش، با بهره‌گیری از تصاویر ماهواره‌ای با وضوح بالا و استفاده از پیشرفته‌ترین مدل‌های یادگیری عمیق، به‌ویژه YOLO (You Only Look Once)، سیستمی هوشمند برای شناسایی و طبقه‌بندی وضعیت ترافیکی طراحی و پیاده‌سازی شد.

مدل پیشنهادی با استفاده از شبکه‌های عصبی کانولوشنی (CNN) و یادگیری انتقالی توانست وضعیت ترافیک شهری را به سه دسته: ترافیک سنگین، خیابان شلوغ، و بدون ترافیک طبقه‌بندی کند. نتایج حاصل نشان دادند که این سیستم قادر است با دقت بالا، وضعیت ترافیکی را در مقیاس شهری شناسایی و تحلیل کند. این پژوهش با بهره‌گیری از داده‌های تصویری ماهواره‌ای، پتانسیل بالای فناوری‌های هوشمند در بهبود مدیریت ترافیک را به نمایش گذاشته است.

با این حال، در برخی تصاویر به دلیل وجود سایه‌ها، اشیاء بزرگ یا تغییرات نوری، مدل در شناسایی دقیق ترافیک با چالش‌هایی مواجه شد. با توجه به نتایج این تحقیق، استفاده از تصاویر با تنوع بیشتر و بهره‌گیری از تکنیک‌های پیشرفته‌تر در آینده می‌تواند به بهبود بیشتر دقت و کارایی مدل کمک کند و گامی مؤثر در جهت مدیریت هوشمند و بهینه ترافیک شهری باشد.

کلیدواژه‌ها: پردازش تصویر، یادگیری عمیق، تصاویر ماهواره‌ای، شبکه عصبی کانولوشنی (CNN)، مدیریت هوشمند ترافیک، مدل YOLO

## فهرست مطالب

عنوان	شماره صفحه
تقدیر و تشکر.....	أ
چکیده.....	ب
<b>فصل اول مقدمه.....</b>	<b>٧</b>
مقدمه.....	٩
تصویر چیست؟.....	١٠
تصاویر ماهواره‌ای چیست؟.....	١١
پردازش تصاویر ماهواره‌ای.....	١٣
اهداف پردازش تصویر ماهواره‌ای.....	١٣
یادگیری ماشین (Machine Learning).....	١٤
اصول اساسی یادگیری ماشین.....	١٤
انواع یادگیری.....	١٥
شبکه‌های عصبی مصنوعی (Artificial Neural Networks - ANN).....	١٦
فرآیند یادگیری.....	١٨
ضرورت کنترل ترافیک.....	١٨

۱۹.....	کارهای گذشته.....
<b>۲۲.....</b>	<b>فصل دوم ابزارهای مورد استفاده در پروژه</b>
۲۳.....	مقدمه.....
۲۴.....	پایتون (Python).....
۲۴.....	یادگیری عمیق (Deep Learning).....
۲۵.....	ویژگی‌های کلیدی یادگیری عمیق.....
۲۶.....	تفاوت‌های کلیدی بین یادگیری ماشین و یادگیری عمیق.....
۲۸.....	قطعه‌بندی تصویر (Image Segmentation).....
۲۸.....	انواع قطعه‌بندی تصویر.....
۲۹.....	مراحل قطعه‌بندی تصویر.....
۲۹.....	تکنیک‌های رایج.....
۳۰.....	TensorFlow.....
۳۰.....	Google Colab.....
۳۱.....	YOLO (You Only Look Once).....
۳۲.....	نسخه‌های مختلف YOLO.....
۳۴.....	Visual Studio Code (VSCode).....
۳۵.....	ایجاد محیط مجازی (Virtual Environment).....
۳۶.....	Roboflow.....
۳۸.....	Kaggle.....
۳۹.....	نتیجه‌گیری.....
<b>۴۴.....</b>	<b>فصل سوم جمع‌آوری داده‌ها و مدل مناسب</b>
۴۱.....	مقدمه.....
۴۳.....	استفاده از داده‌های Roboflow.....
۴۴.....	پیش‌پردازش داده‌ها و آماده‌سازی برای آموزش.....
۴۵.....	انتخاب مدل مناسب.....
۴۶.....	دلیل استفاده از مدل‌های از پیش آموزش‌دیده شده.....

۴۶	یادگیری انتقالی یا (Transfer Learning).....
۴۷	مزایای یادگیری انتقالی.....
۴۷	روش های افزایش کارایی مدل ها.....
۴۹	آموزش مدل.....
۴۹	مراحل آموزش مدل.....
۵۱	نتیجه گیری.....
<b>۵۲</b>	<b>فصل چهارم ارزیابی نتایج</b> .....
۵۳	مقدمه.....
۵۴	معیارهای ارزیابی مدل (Evaluation Metrics).....
۶۲	نمونه های پیش بینی شده.....
۶۵	نتیجه گیری.....
<b>۶۶</b>	<b>فصل پنجم جمع بندی و نتیجه گیری</b> .....
۶۷	مقدمه.....
۶۸	مروری بر روش ها و تکنولوژی ها.....
۶۸	نتایج و ارزیابی ها.....
۶۹	تحلیل چالش ها و نقاط ضعف.....
۶۹	جمع بندی.....
۶۹	نتیجه گیری و پیشنهادات برای کارهای آینده.....
۷۱	منابع.....

## فهرست شکل ها

عنوان	شماره صفحه
شکل ۱ تصویر WorldView ماهواره ای.....	۱۲
شکل ۲ ساختار داخلی شبکه عصبی .....	۱۷
شکل ۳ کارکرد شبکه عصبی .....	۲۷
شکل ۴ طبقه بندی کردن .....	۲۸
شکل ۵ آموزش مدل .....	۳۱
شکل ۶ لایه های مدل یولو .....	۳۴
شکل ۷ اجرا گرفتن در محیط محلی .....	۳۵
شکل ۸ محیط و انجام برجسب گذاری .....	۳۷
شکل ۹ محیط کار در Kaggle .....	۳۸
شکل ۱۰ برجسب زدن داده ها .....	۴۲
شکل ۱۱ انتقال یادگیری .....	۴۸
شکل ۱۲ تنظیم پرامتر و آموزش مدل .....	۵۰
شکل ۱۳ معیار های ارزیابی .....	۵۶
شکل ۱۴ نتایج کلی مدل .....	۵۷
شکل ۱۵ نتیجه f1-curve .....	۵۸
شکل ۱۶ نتیجه precision curve .....	۵۹
شکل ۱۷ Confusion metricx .....	۶۰
شکل ۱۸ Confusion metrix Normalized .....	۶۱
شکل ۱۹ مثال شماره ۱ .....	۶۲
شکل ۲۰ مثال شماره ۲ .....	۶۳
شکل ۲۱ مثال شماره ۳ .....	۶۴



## فصل اول مقدمه

## • مقدمه

امروزه با رشد روزافزون جمعیت و افزایش تعداد خودروها در شهرهای بزرگ، مدیریت ترافیک به یکی از بزرگترین چالش‌های جوامع شهری تبدیل شده است. روش‌های سنتی مدیریت ترافیک، مانند چراغ‌های راهنمایی و نقشه‌های دستی، دیگر توان پاسخگویی به نیازهای مدرن را ندارند. در این راستا، فناوری‌های نوین مانند پردازش تصویر و هوش مصنوعی با استفاده از داده‌های عظیم به یکی از راهکارهای کلیدی برای بهینه‌سازی مدیریت ترافیک تبدیل شده‌اند.

یکی از منابع مهم داده در این زمینه، تصاویر ماهواره‌ای با وضوح بالا هستند که به کمک آن‌ها می‌توان شرایط ترافیکی را در مقیاس وسیع تحلیل کرد. این تصاویر قادرند اطلاعاتی مفید در مورد وضعیت ترافیک در سطح جاده‌ها، تقاطع‌ها و شهرها ارائه دهند و با استفاده از الگوریتم‌های یادگیری عمیق می‌توان از این داده‌ها برای شناسایی و طبقه‌بندی ترافیک به‌طور دقیق بهره برد.

هدف از این پژوهش، طراحی و پیاده‌سازی یک سیستم تشخیص وضعیت ترافیک به‌صورت خودکار با استفاده از تصاویر ماهواره‌ای است. در این پروژه از شبکه‌های عصبی کانولوشنی و مدل‌های یادگیری عمیق مانند YOLO استفاده شده تا بتوانیم وضعیت ترافیک را به دسته‌های مختلفی مانند ترافیک سنگین، سبک و بدون ترافیک طبقه‌بندی کنیم. این روش می‌تواند به مدیران شهری در بهبود جریان ترافیک و کاهش مشکلات ناشی از آن کمک شایانی نماید.

## • تصویر چیست؟

نمایش و ارائه ای از یک شخص، فرد، منظره یا حتی یک ایده یا هر چیز دیگری که بتوان آنرا در قالب یک شکل به دیگران نشان داد، یک تصویر محسوب میشود. در اینجا اشاره به این نکته خالی از لطف نخواهد بود که تصاویر می توانند بصورت آنالوگ یا دیجیتال باشد و همانطور که تصویر آنالوگ با تصویر دیجیتال متفاوت است پردازش هر کدام از این انواع تصویر تفاوت هایی در کاربرد و الگوریتم دارد تصاویر دیجیتال در واقع مجموعه ای از پیکسل ها هستند که هر کدام رنگ خاص خود را دارد و تفاوت رنگها و شدت روشنایی هر کدام از این پیکسل ها نهایتاً چیزی را تشکیل می دهد که به آن تصویر یا تصویر دیجیتال می گوییم از این رو تصاویر دیجیتال ماتریسی از پیکسل ها هستند و میتوان آنها را به صورت یک ماتریس تعریف کرد. یک تصویر آنالوگ برخالف تصاویر دیجیتال شامل پیکسل ها و یا مقادیر رنگ در نقاط مختلف صفحه نیستند درواقع تصاویر آنالوگ به روش هایی غیر دیجیتالی ایجاد می شوند مانند ظاهر کردن فیلم دوربین عکاسی بر روی کاغذ عکاسی که طی یک پروسه شیمیایی رخ می دهد و یا تصاویری که برای تلویزیون مخابره می شوند یا تصاویر عکسبرداری پزشکی همه نمونه هایی از تصویر های آنالوگ هستند. از آنجایی که پردازش تصویر توسط کامپیوتر صورت می گیرد در واقع در پردازش تصاویر آنالوگ و دیجیتال از روش های مشابهی استفاده می شود، چرا که قبل از انجام هرگونه پردازشی بر روی تصاویر آنالوگ ما نیازمند آن هستیم این تصاویر را به ت صاویر دیجیتالی و قابل استفاده در رایانه تبدیل کنیم ۱۰.

البته بررسی دقیق تصاویر و مقایسه ی انواع آنها با یکدیگر هدف این کار پژوهشی نیست و تا انتهای متن منظور از تصویر همان تصویر دیجیتال است. در واقع یک تصویر (دیجیتال)، چیزی نیست جز یک ارائه ی دیجیتال از نور که یک موج محسوب میشود. از این رو یک تصویر را یک سیگنال در نظر میگیریم، و پردازش تصویر دیجیتال، یکی از زیر مجموعه های مباحث سیگنال و سیستم در نظر گرفته میشوند. تصاویر صرف نظر از اینکه یک راه برای ذخیره سازی شکلی از یک شی یا منظره یا یک موجود زنده یا هر مفهوم قابل رسم دیگری در دنیای دیجیتال باشند به عنوان یک روش انتقال اطلاعات مورد استفاده قرار میگیرند. هر تصویر میتواند دارای مقدار زیادی اطلاعات مفید باشد. استخراج این اطلاعات توسط کامپیوتر یکی از اصلی ترین کاربردهای پردازش تصاویر دیجیتال است. در واقع در این کاربرد از پردازش تصاویر دیجیتال، ورودی سیستم ما یک تصویر و خروجی آن یک تصویر دیگر، مجموعه

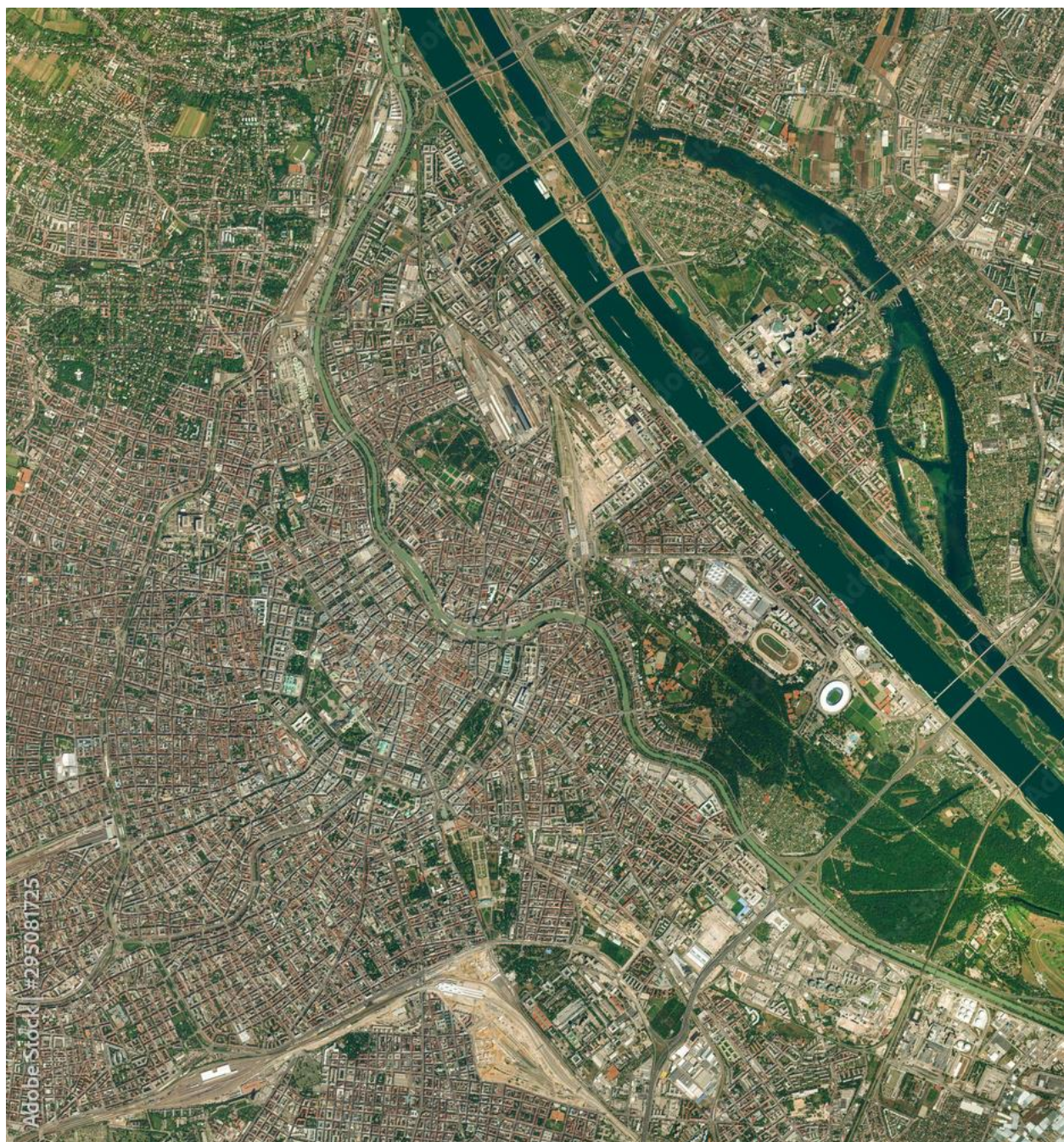
ای از ویژگی های تصویر یا هدف مورد نظر و یا مجموعه ای داده در مورد تصویر باشد. برای رسیدن به این داده ها و درواقع اجرای پردازش بر روی تصاویر ، امروزه روش ها و ابزار های بسیار زیاد و متنوعی وجود دارند.

## • تصاویر ماهواره ای چیست؟

تصاویر ماهواره ای به تصاویری اطلاق می شود که توسط ماهواره های موجود در مدار زمین ثبت و ارسال می شوند. این ماهواره ها از سنسورها و دوربین های پیشرفته ای استفاده می کنند تا اطلاعات و داده های تصویری از سطح زمین، اقیانوس ها و جو زمین جمع آوری کنند. تصاویر ماهواره ای به دلیل پوشش گسترده ای که از سطح زمین ارائه می دهند، به عنوان یکی از مهم ترین منابع داده برای نظارت، تحقیق و تحلیل در زمینه های مختلف شناخته می شوند.

ماهواره های تصویربرداری معمولاً در مدارهای مختلفی از جمله مدار پایین زمین (LEO) ، مدار میانی (MEO) و مدار زمین ثابت (GEO) قرار دارند و هر کدام از این مدارها کاربردهای مختلفی دارند. بسته به نوع سنسورهای به کار رفته در این ماهواره ها، تصاویر ممکن است شامل بخش های مختلف طیف الکترومغناطیسی مانند نور مرئی، مادون قرمز یا ماوراء بنفش باشند که این امر امکان استفاده از تصاویر در شرایط مختلف روز و شب یا آب و هوای نامساعد را فراهم می کند.





شکل ۱ تصویر WorldView ماهواره‌ای

## • پردازش تصاویر ماهواره‌ای

تصاویر خامی که توسط ماهواره‌ها ثبت می‌شوند، برای استفاده باید پردازش شوند. پردازش تصویر شامل بهبود کیفیت تصویر، تشخیص اشیاء و تحلیل داده‌هاست. بسته به هدف استفاده، تصاویر ممکن است از طریق الگوریتم‌های مختلفی تحلیل شوند که از جمله مهم‌ترین آن‌ها می‌توان به تشخیص اشیاء و تحلیل تغییرات زمانی

## • اهداف پردازش تصویر ماهواره‌ای:

### ○ بهبود کیفیت تصویر:

حذف نویزها، افزایش وضوح، و بهبود روشنایی تصاویر برای استخراج دقیق‌تر اطلاعات از آن‌ها.

### ○ تشخیص و شناسایی اشیاء:

شناسایی و طبقه‌بندی اشیاء مختلف روی سطح زمین مانند خودروها، ساختمان‌ها، جنگل‌ها، دریاها و جاده‌ها.

### ○ تحلیل تغییرات زمانی:

مقایسه تصاویر ثبت‌شده در بازه‌های زمانی مختلف برای بررسی تغییرات در سطح زمین، پوشش گیاهی، تغییرات اقلیمی و پدیده‌های طبیعی.

### ○ نقشه‌برداری دقیق:

تولید نقشه‌های دقیق از مناطق مختلف برای اهدافی همچون برنامه‌ریزی شهری، مدیریت منابع و زیرساخت‌های عمرانی.

### ○ پیش‌بینی و مدیریت بحران:

تشخیص و تحلیل بلایای طبیعی مانند سیل، آتش‌سوزی، زلزله و طوفان برای کمک به مدیریت بهتر بحران‌ها و اقدامات امدادی.

### ○ نظارت بر محیط زیست و منابع طبیعی:

پیگیری و بررسی تغییرات در اکوسیستم‌ها، مناطق جنگلی، دریاچه‌ها و منابع آبی برای حفظ و مدیریت بهتر محیط زیست.

### ○ تحلیل جغرافیایی و اقلیمی:

بررسی شرایط جغرافیایی و اقلیمی مختلف مانند دمای سطح زمین، الگوهای بارش و تغییرات دمایی برای تحقیقات علمی و محیطی.

## ○ شناسایی و ارزیابی منابع:

پردازش تصویر ماهواره‌ای برای شناسایی منابع زیرزمینی، مانند معادن و منابع نفتی، که امکان بررسی از سطح زمین به صورت دقیق‌تری فراهم می‌کند.

## • یادگیری ماشین (Machine Learning)

یادگیری ماشین (Machine Learning) یک زیرمجموعه از هوش مصنوعی است که به سیستم‌ها این امکان را می‌دهد که بدون برنامه‌ریزی صریح، از داده‌ها یاد بگیرند و پیش‌بینی کنند. به عبارت دیگر، یادگیری ماشین به سیستم‌ها این قابلیت را می‌دهد که با استفاده از الگوریتم‌ها و مدل‌های آماری، الگوها و ارتباطات موجود در داده‌ها را شناسایی کنند و از آن‌ها برای انجام وظایف خاص استفاده نمایند [۱].

## • اصول اساسی یادگیری ماشین:

- مدل‌سازی: مدل‌های یادگیری ماشین برای شبیه‌سازی فرآیندهای تصمیم‌گیری و پیش‌بینی از داده‌های آموزشی استفاده می‌کنند.
- آموزش: مدل‌ها با استفاده از مجموعه‌ای از داده‌های آموزشی (Training Data) آموزش داده می‌شوند تا الگوها و ویژگی‌های مختلف را یاد بگیرند.
- پیش‌بینی و ارزیابی: پس از آموزش، مدل‌ها می‌توانند بر روی داده‌های جدید پیش‌بینی انجام دهند و با استفاده از داده‌های آزمایشی (Test Data) ارزیابی شوند.

## • انواع یادگیری

بر اساس کیفیت داده های موجود برای آموزش یادگیرنده می توانیم فرایند یادگیری را به سه دسته اصلی تقسیم کنیم :

○ **یادگیری با نظارت :** در این نوع به رایانه نمونه هایی از ورودی و همچنین خروجی های مورد انتظار را ارائه می کنیم و هدف این است که یادگیرنده بتواند یک قانون و روش کلی که از ورودی داده شده به خروجی مطلوب برسد را کشف کند [۱] .

به بیان ریاضی، هنگامی که متغیر ورودی ( $X$ ) و متغیر خروجی ( $Y$ ) موجودند و می توان بر اساس آنها از یک الگوریتم برای حصول یک تابع نگاشت ورودی به خروجی استفاده کرد در واقع یادگیری نظارت شده است. تابع نگاشت به صورت  $X(f = Y)$  نشان داده می شود .

○ **یادگیری نظارت نشده :** در این نوع از یادگیری، یادگیرنده سعی می کند تا بتواند خودش ساختار موجود در داده های ارائه شده را کشف کند و هیچ گونه لیبل گذاری یا مشخص کردن خروجی مورد انتظار در این مدل از یادگیری وجود نخواهد داشت [۱] .

به بیان ریاضی، یادگیری نظارت نشده مربوط به زمانی است که در مجموعه داده فقط متغیرهای ورودی ( $X$ ) وجود داشته باشند و هیچ متغیر داده خروجی موجود نباشد. به این نوع یادگیری، نظارت ۱۵ نشده گفته می شود زیرا برخلاف یادگیری نظارت شده، هیچ پاسخ صحیح داده شده ای وجود ندارد و ماشین خود باید به دنبال پاسخ باشد .

یادگیری بدون نظارت می تواند خودش یک هدف باشد از آنجایی که ما به دنبال کشف ساختارهایی در داده های ارائه شده هستیم.

○ **یادگیری تقویتی :** در این روش یادگیری برنامه یادگیرنده در یک محیط پویا و در راستای رسیدن به یک هدف به تعامل با داده ها و محیط می پردازد. فرایند یادگیری در این روش به این صورت است که در حین تعامل یادگیرنده و حرکت آن در محیط، ما یادگیرنده را به کمک پاداش و تنبیه راهنمایی



میکنیم. قدم برداشتن در مسیر درست = پاداش و امتیاز بیشتر، قدم برداشتن در مسیر اشتباه = تنبیه و کمتر شدن امتیاز.

با استفاده از یادگیری تقویتی، ماشین می آموزد که تصمیمات مشخصی را در محیطی که دائم در معرض آزمون و خطا است اتخاذ کند.

البته روش های دیگری به جز این روش ها وجود دارد که در کاربردهای خاص مورد استفاده قرار می گیرند، اما این سه روش را میتوان سه روش اصلی و تعیین کننده در دنیای یادگیری ماشین تلقی کرد.

یادگیری ماشین یکی از زیر شاخه های هوش مصنوعی محسوب میشود. چرا که عموماً خروجی روش های مبتنی بر یادگیری ماشین، بیشتر به خروجی های تهیه شده توسط انسان نزدیکند.

به علاوه در فرایند استفاده از یک الگوریتم مبتنی بر یادگیری، خبری از مشخص کردن همه ی حالت های ممکن برای داده های مورد استفاده در سیستم وجود ندارد و به نوعی میتوان رفتار ماشین در روش های مبتنی بر یادگیری را بیشتر شبیه به رفتار انسان دانست.

## • شبکه های عصبی مصنوعی (Artificial Neural Networks - ANN)

شبکه های عصبی مصنوعی، یکی از مهم ترین مفاهیم در حوزه هوش مصنوعی و یادگیری ماشین هستند که الهام گرفته از ساختار و عملکرد مغز انسان است. این شبکه ها از واحدهای پردازشی به نام نورون (یا واحد) تشکیل شده اند و به صورت لایه ای سازمان دهی شده اند [۲].

## • ساختار شبکه عصبی

شبکه های عصبی مصنوعی از سه بخش اصلی تشکیل می شوند:

### ○ لایه ورودی (Input Layer):

نورون های این لایه، داده های خام ورودی را دریافت می کنند. تعداد نورون ها در این لایه برابر با تعداد ویژگی های داده های ورودی است.

○ لایه‌های مخفی (Hidden Layers):

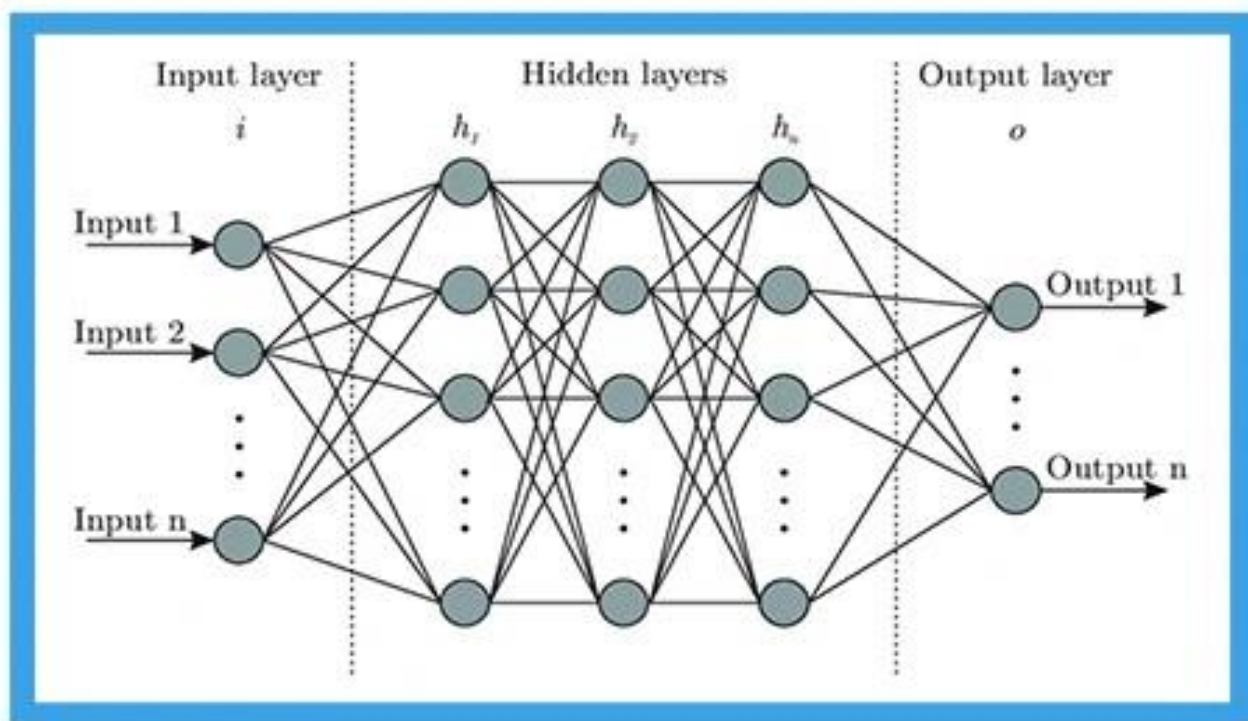
لایه‌های مخفی بین لایه ورودی و لایه خروجی قرار دارند و وظیفه پردازش و یادگیری داده‌ها را بر عهده دارند. تعداد لایه‌های مخفی و نورون‌ها در هر لایه می‌تواند بر اساس نوع مسئله متفاوت باشد.

○ لایه خروجی (Output Layer):

نورون‌های این لایه خروجی نهایی شبکه را تولید می‌کنند که می‌تواند یک کلاس (در مسائل دسته‌بندی) یا یک مقدار عددی (در مسائل پیش‌بینی) باشد.

• نحوه عملکرد

هر نورون در شبکه عصبی وزن‌های متغیری دارد که با داده‌های ورودی ترکیب می‌شوند و سپس به یک تابع فعال‌سازی (Activation Function) اعمال می‌شوند. این فرآیند به شبکه اجازه می‌دهد که روابط پیچیده بین ورودی و خروجی را مدل‌سازی کند. با هر بار اجرای فرآیند یادگیری، وزن‌ها به روزرسانی می‌شوند تا شبکه به تدریج به دقت بالاتری دست یابد [۲]. [۳]



شکل ۲ ساختار داخلی شبکه عصبی

## • فرآیند یادگیری

شبکه‌های عصبی از فرآیند یادگیری با نظارت (Supervised Learning) استفاده می‌کنند. در این روش، شبکه با یک مجموعه داده که شامل ورودی‌ها و خروجی‌های مطلوب است آموزش داده می‌شود. سپس، الگوریتم‌های بهینه‌سازی مانند پس‌انتشار خطا (Backpropagation) برای تنظیم وزن‌ها استفاده می‌شوند.

## • ضرورت کنترل ترافیک

ضرورت کنترل ترافیک از چندین جنبه مهم مورد بررسی قرار می‌گیرد:

○ **افزایش ایمنی** : اولین و اصلی‌ترین دلیل استفاده از سیستم‌های کنترل ترافیک، حفظ ایمنی رانندگان، مسافران و عابران پیاده است. در جاده‌ها و تقاطع‌هایی که هیچ‌گونه کنترلی وجود ندارد، احتمال وقوع تصادفات و آسیب‌ها بسیار بیشتر است. کنترل‌های ترافیکی از طریق چراغ‌های راهنمایی و علائم، به هدایت و هماهنگی حرکت وسایل نقلیه کمک می‌کنند و از بروز برخوردها جلوگیری می‌کنند [۴]

○ **نظم‌دهی به حرکت وسایل نقلیه** : کنترل ترافیک به مدیریت و نظم‌دهی حرکت وسایل نقلیه در تقاطع‌ها و خیابان‌های شلوغ کمک می‌کند. بدون کنترل ترافیکی، تقاطع‌ها به مناطق خطرناک و آشوب‌زده تبدیل می‌شوند که باعث تصادف و تراکم شدید می‌شود. این سیستم‌ها به‌ویژه در مناطقی که تراکم بالایی از وسایل نقلیه وجود دارد، ضروری هستند [۴]

○ **کاهش شدت تصادفات** : وجود سیستم‌های کنترل ترافیک باعث می‌شود تا رانندگان بهتر با شرایط جاده هماهنگ شوند و در نتیجه تصادفات با شدت کمتری رخ دهند. این سیستم‌ها به رانندگان اجازه می‌دهند که با سرعت و دقت بیشتری حرکت کنند و خطر برخورد را کاهش می‌دهند [۴]

○ **جلوگیری از تراکم و انسداد ترافیک** : یکی از نتایج مستقیم عدم وجود سیستم‌های کنترل ترافیک، تراکم سنگین و حتی مسدود شدن خیابان‌ها است. سیستم‌های کنترل ترافیک به جلوگیری از ایجاد

ترافیک سنگین و کاهش زمان توقف وسایل نقلیه کمک می‌کنند، که به‌نوبه خود باعث روان‌تر شدن جریان حرکت وسایل نقلیه می‌شود [۴]

○ **افزایش بهره‌وری سوخت:** با کاهش زمان توقف خودروها در ترافیک، سیستم‌های کنترل ترافیک به کاهش مصرف سوخت و بهبود بهره‌وری انرژی کمک می‌کنند. وسایل نقلیه‌ای که مدت زمان کمتری در ترافیک گیر می‌کنند، سوخت کمتری مصرف می‌کنند و در نتیجه آلودگی هوا نیز کاهش می‌یابد [۴]

در مجموع، کنترل ترافیک برای بهبود ایمنی، کاهش تراکم، و حفظ محیط زیست از اهمیت بالایی برخوردار است. این سیستم‌ها با بهینه‌سازی حرکت وسایل نقلیه و کاهش زمان توقف در تقاطع‌ها، تأثیرات مثبتی بر بهره‌وری و محیط زیست دارند [۴]

## • کارهای گذشته

در سال‌های اخیر، استفاده از یادگیری ماشین در مدیریت ترافیک به یکی از رویکردهای محبوب و مؤثر تبدیل شده است. این تکنیک‌ها به منظور بهبود عملکرد سیستم‌های حمل و نقل و کاهش مشکلاتی مانند ترافیک، تصادفات و آلودگی هوا به کار گرفته می‌شوند. در ادامه به برخی از کاربردهای کلیدی یادگیری ماشین در این زمینه اشاره می‌کنم:

○ **پیش‌بینی ترافیک:** یادگیری ماشین می‌تواند برای پیش‌بینی الگوهای ترافیک استفاده شود. این پیش‌بینی‌ها به مدیران ترافیک کمک می‌کند تا با تغییر زمان‌بندی چراغ‌های راهنمایی و اطلاع‌رسانی به رانندگان در مورد ترافیک، از ایجاد ترافیک سنگین جلوگیری کنند. این کاربرد به ویژه در شرایطی که داده‌های تاریخی کافی موجود نیست، چالش‌هایی دارد [۵]

○ **تشخیص و شناسایی اشیاء:** تکنیک‌های یادگیری عمیق، به‌ویژه شبکه‌های عصبی کانولوشنی (CNN)، برای شناسایی و تشخیص وسایل نقلیه و عابران پیاده در زمان واقعی به کار می‌روند. این تکنیک‌ها می‌توانند به بهینه‌سازی زمان‌بندی چراغ‌های راهنمایی و کاهش تصادفات کمک کنند [۵]

○ **مدیریت سیگنال‌های ترافیکی:** استفاده از یادگیری تقویتی (Reinforcement Learning) برای بهینه‌سازی زمان‌بندی چراغ‌های راهنمایی، یکی دیگر از کاربردهای کلیدی است. این الگوریتم‌ها

می‌توانند به‌طور خودکار زمان‌بندی‌های بهینه را بر اساس جریان واقعی ترافیک یاد بگیرند و پیاده‌سازی کنند [۵]

○ **تحلیل داده‌های ترافیکی:** یادگیری ماشین می‌تواند به تحلیل داده‌های واقعی ترافیک کمک کند، از جمله شناسایی الگوهای غیرعادی و پیش‌بینی مشکلات ترافیکی قبل از وقوع آن‌ها. این اطلاعات به مدیران ترافیک کمک می‌کند تا به‌سرعت واکنش نشان دهند و بهبودهایی در زیرساخت‌های ترافیکی ایجاد کنند [۵]

○ **مدیریت پارکینگ:** یادگیری ماشین می‌تواند به شناسایی فضاهای پارکینگ خالی در زمان واقعی کمک کند، که منجر به کاهش زمان و سوختی می‌شود که رانندگان برای پیدا کردن پارکینگ صرف می‌کنند. این سیستم‌ها با تحلیل داده‌های دوربین‌ها و سنسورها عمل می‌کنند [۵]



## **فصل دوم ابزارهای مورد استفاده در پروژه**

## • مقدمه

در این فصل به بررسی ابزارها و محیط‌های توسعه‌ای که در پیاده‌سازی و بهبود پروژه استفاده شده‌اند، پرداخته می‌شود. پروژه حاضر که به تحلیل ترافیک با استفاده از یادگیری عمیق و مدل‌های پیشرفته پردازش تصویر می‌پردازد، نیازمند بهره‌گیری از ابزارها و فناوری‌های متنوعی است که در اینجا معرفی می‌شوند.

پایتون، به عنوان یکی از زبان‌های برنامه‌نویسی قدرتمند و پرکاربرد، نقش مهمی در پیاده‌سازی مدل‌های یادگیری ماشین و یادگیری عمیق دارد. این زبان به دلیل وجود کتابخانه‌های غنی مانند TensorFlow و Keras، به یکی از انتخاب‌های اصلی برای توسعه‌دهندگان تبدیل شده است. TensorFlow، به عنوان یک چارچوب قدرتمند یادگیری عمیق، امکان ساخت و آموزش مدل‌های پیشرفته را فراهم می‌کند. این چارچوب با استفاده از شبکه‌های عصبی عمیق، فرآیند شناسایی و طبقه‌بندی داده‌های ترافیکی را بهبود می‌بخشد.

همچنین در این پروژه از مدل YOLO (You Only Look Once)، یکی از مدل‌های برجسته تشخیص اشیاء، استفاده شده است. این مدل برای شناسایی و تحلیل خودروها و وضعیت ترافیک در تصاویر ماهواره‌ای به کار گرفته شد. در این راستا، Roboflow به عنوان ابزاری برای برچسب‌گذاری داده‌ها و آماده‌سازی آن‌ها برای مدل‌های تشخیص اشیاء مورد استفاده قرار گرفت.

علاوه بر این، در پروژه از تکنیک قطعه‌بندی تصویر (Image Segmentation) نیز بهره گرفته شده است. این تکنیک به مدل YOLOv11 کمک می‌کند تا تصاویر ترافیکی را به بخش‌های مختلف تقسیم کرده و نواحی ترافیکی را به دقت شناسایی کند. با این روش، مدل توانست وضعیت ترافیک را به درستی تشخیص داده و آن را به دسته‌هایی مانند ترافیک سنگین، سبک و بدون ترافیک طبقه‌بندی کند.

برای مدیریت پروژه و اجرای مدل‌ها از محیط‌هایی مانند Kaggle و VS Code استفاده شد. Kaggle به عنوان یک پلتفرم رایگان برای اجرای مدل‌های یادگیری عمیق و ذخیره‌سازی نتایج مورد استفاده قرار گرفت، در حالی که VS Code به عنوان یک محیط توسعه یکپارچه (IDE) برای کدنویسی و دیباگ پروژه استفاده شد.



همچنین به منظور مدیریت پکیج‌ها و وابستگی‌های مختلف پروژه، از محیط‌های مجازی (Virtual Environments) استفاده گردید. این محیط‌ها به ما امکان می‌دهند که پکیج‌های مورد نیاز هر پروژه را به صورت مجزا مدیریت کرده و از تداخل بین آن‌ها جلوگیری کنیم.

این ابزارها و چارچوب‌ها نقش کلیدی در موفقیت پروژه دارند و در این فصل به طور جامع به تحلیل و بررسی آن‌ها پرداخته خواهد شد.

## • پایتون (Python)

پایتون یکی از محبوب‌ترین و پرکاربردترین زبان‌های برنامه‌نویسی در زمینه علم داده، یادگیری ماشین و یادگیری عمیق است. این زبان به دلیل سادگی در خوانایی، کتابخانه‌های غنی، و پشتیبانی گسترده از پروژه‌های یادگیری ماشین، در پروژه من نقش مهمی ایفا کرد. پایتون ابزارهایی مانند Numpy، Pandas، Matplotlib، و به ویژه کتابخانه‌های یادگیری ماشین مثل Scikit-learn و Keras را در اختیارم قرار داد که پایه گذار بسیاری از فرآیندهای پردازش داده و آموزش مدل‌ها بودند.

در ابتدای پروژه، یادگیری پایتون را به عنوان اولین قدم شروع کردم و با مفاهیم ابتدایی این زبان آشنا شدم. سپس به مرور زمان و با آموزش دیدن بیشتر، از آن برای اجرای الگوریتم‌های یادگیری ماشین استفاده کردم. پایتون به دلیل انعطاف‌پذیری بالا و کتابخانه‌های گسترده، ابزاری حیاتی برای این پروژه بود.

## • یادگیری عمیق (Deep Learning)

شاخه‌ای از یادگیری ماشین است که از شبکه‌های عصبی مصنوعی با لایه‌های متعدد برای یادگیری ویژگی‌های پیچیده از داده‌ها استفاده می‌کند. برخلاف مدل‌های ساده‌تر، یادگیری عمیق به دلیل وجود لایه‌های متوالی می‌تواند روابط غیرخطی و پیچیده بین ورودی و خروجی‌ها را یاد بگیرد. هر لایه در شبکه عصبی کانولوشنی (CNN) یا بازگشتی (RNN) ویژگی‌های جدیدی از داده‌ها استخراج می‌کند و این ویژگی‌ها برای تصمیم‌گیری و پیش‌بینی‌های دقیق‌تر در مراحل بعدی استفاده می‌شوند. [۳]

شبکه‌های عصبی عمیق شامل لایه‌های ورودی، مخفی و خروجی هستند. لایه‌های مخفی به شبکه کمک می‌کنند که ویژگی‌های پیچیده‌تر و انتزاعی‌تر را بیاموزد. در این فرآیند، داده‌های خام از لایه ورودی به لایه‌های مخفی عبور می‌کنند و هر لایه به صورت خودکار ویژگی‌های مشخصی از داده‌ها را یاد می‌گیرد. از این مدل‌ها در کاربردهایی مانند تشخیص تصویر، پردازش زبان طبیعی (NLP)، خودروهای خودران، و حتی پزشکی استفاده می‌شود. [۳]

یکی از بزرگترین مزایای یادگیری عمیق این است که خودآموز است، به این معنا که می‌تواند بدون نیاز به مهندسی ویژگی‌های دستی، از داده‌ها الگوهای معنی‌داری بیاموزد. با این حال، مدل‌های یادگیری عمیق به حجم عظیمی از داده‌ها و قدرت محاسباتی بالا نیاز دارند، که این امر باعث می‌شود در حوزه‌هایی که داده‌های زیاد و منابع پردازشی قوی در دسترس هستند، بسیار کارآمد باشند. [۳]

با توجه به رشد سریع این تکنولوژی، یادگیری عمیق به یکی از مهم‌ترین روش‌های محاسباتی در هوش مصنوعی تبدیل شده و همچنان در حال پیشرفت است. یکی از چالش‌های اصلی این روش، جلوگیری از بیش‌برازش (overfitting) است که با استفاده از تکنیک‌هایی مانند Dropout و Regularization مدیریت می‌شود. [۳]

## • ویژگی‌های کلیدی یادگیری عمیق:

- **شبکه‌های عصبی عمیق:** یادگیری عمیق از شبکه‌های عصبی با چندین لایه استفاده می‌کند که قادرند ویژگی‌های مختلف داده‌ها را در سطوح مختلف تحلیل کنند. این لایه‌ها شامل لایه‌های ورودی، لایه‌های مخفی و لایه‌های خروجی هستند.
- **قابلیت یادگیری ویژگی‌های پیچیده:** با افزایش تعداد لایه‌ها، شبکه‌های عصبی عمیق قادرند ویژگی‌های پیچیده‌تری از داده‌ها را شناسایی کنند که برای یادگیری ماشین سنتی دشوار است.
- **خودآموزی:** این شبکه‌ها قادرند به طور خودکار ویژگی‌های مهم را از داده‌ها استخراج کنند و به پیش‌بینی‌های دقیق‌تر و عملکرد بهتری دست یابند.

- **تفاوت‌های کلیدی بین یادگیری ماشین و یادگیری عمیق**

- **سطح پیچیدگی :** یادگیری عمیق معمولاً برای داده‌های پیچیده و حجیم که ویژگی‌های زیادی دارند، مناسب‌تر است، در حالی که یادگیری ماشین سستی برای مسائل ساده‌تر و داده‌های کمتر پیچیده به کار می‌رود.
- **پیش‌پردازش داده‌ها :** در یادگیری ماشین، اغلب نیاز به پیش‌پردازش و استخراج ویژگی‌های دستی است، در حالی که یادگیری عمیق می‌تواند ویژگی‌های مهم را به طور خودکار از داده‌ها استخراج کند.

- **کاربردهای یادگیری عمیق**

- **تشخیص تصویر (Image Recognition):**  
در تشخیص اشیاء، چهره‌ها، و طبقه‌بندی تصاویر در دسته‌بندی‌های مختلف به کار می‌رود.
- **بینایی کامپیوتر (Computer Vision):**  
استفاده در خودروهای خودران برای تشخیص موانع، مسیرها و ترافیک.
- **تشخیص گفتار (Speech Recognition):**  
شناسایی و تبدیل گفتار به متن در دستیارهای صوتی مانند Siri و Google Assistant.
- **پردازش زبان طبیعی (Natural Language Processing - NLP):**  
ترجمه خودکار زبان‌ها، تشخیص احساسات و پاسخگویی خودکار در چت‌بات‌ها.
- **خودروهای خودران (Autonomous Vehicles):**  
یادگیری عمیق برای تشخیص محیط، اجتناب از موانع و تصمیم‌گیری آنی در وسایل نقلیه بدون راننده.
- **بازی‌های کامپیوتری و شبیه‌سازی‌ها (Game AI):**  
ایجاد هوش مصنوعی برای بازی‌های ویدیویی و شبیه‌سازی‌های پیچیده.
- **تشخیص تقلب (Fraud Detection):**

بررسی تراکنش‌های مالی و شناسایی رفتارهای غیرعادی و تقلب.

#### ○ پزشکی (Medical Diagnosis):

تشخیص بیماری‌ها از روی تصاویر پزشکی مانند MRI و سی‌تی اسکن‌ها.

#### ○ تولید تصاویر و ویدیوها (Generative Adversarial Networks - GANs):

تولید تصاویر واقعی، ویدیوها و حتی صداها مصنوعی.

#### ○ مدیریت ترافیک:

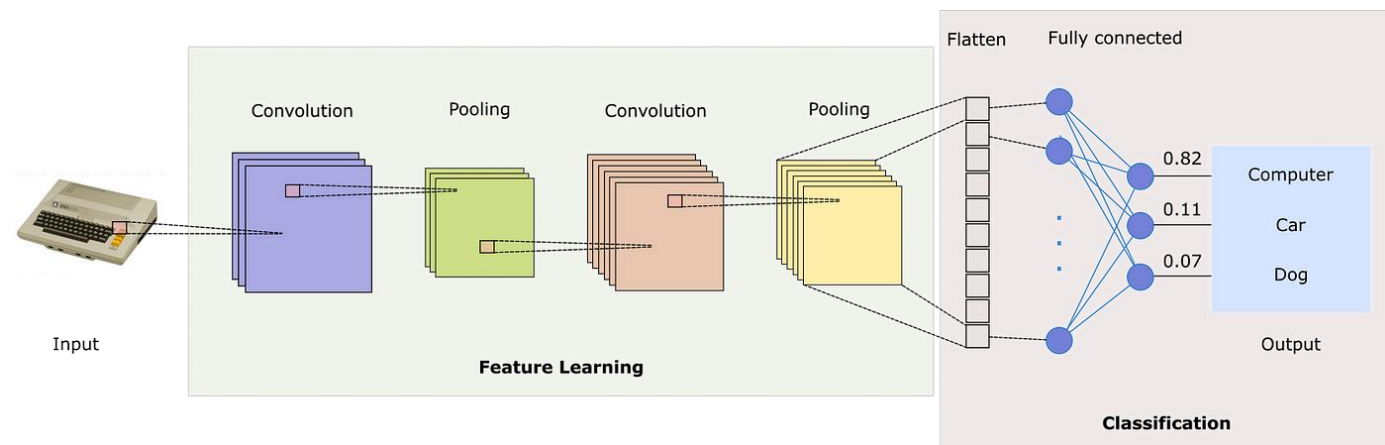
تحلیل تصاویر ماهواره‌ای و دوربین‌های ترافیکی برای کنترل و مدیریت بهینه جریان ترافیک.

#### ○ تجارت الکترونیک و پیشنهاددهی (Recommendation Systems):

ارائه پیشنهادهای شخصی‌سازی شده در سایت‌هایی مانند Amazon و Netflix بر اساس رفتار کاربر.

#### ○ پیش‌بینی بازارهای مالی (Financial Market Prediction):

پیش‌بینی تغییرات قیمت سهام و تجزیه و تحلیل روندهای بازار.



شکل ۳ کارکرد شبکه عصبی

- **قطعه بندی تصویر (Image Segmentation) :**

قطعه بندی تصویر، یکی از تکنیک های پیشرفته در بینایی ماشین است که به تقسیم بندی یک تصویر به بخش های مختلف (سگمنت ها) بر اساس ویژگی های پیکسلی آن می پردازد. هدف از قطعه بندی تصویر، جداسازی نواحی خاص یا اشیاء در تصویر برای تحلیل دقیق تر است. هر پیکسل در تصویر به یکی از دسته های مشخص اختصاص داده می شود تا مرزهای دقیق بین اشیاء مختلف یا نواحی مختلف تصویر مشخص شوند.



شکل ۴ طبقه بندی کردن

- **انواع قطعه بندی تصویر:**

- **قطعه بندی معنایی (Semantic Segmentation) :** هر پیکسل تصویر به یکی از دسته های کلی مانند آسمان، زمین، انسان یا خودرو تعلق می گیرد، بدون توجه به تعداد یا مکان دقیق اشیاء.

○ **قطعه‌بندی نمونه‌ای (Instance Segmentation) :** علاوه بر شناسایی دسته هر پیکسل، هر نمونه از یک شیء نیز به صورت جداگانه تشخیص داده می‌شود. به عنوان مثال، هر خودرو به‌طور جداگانه از بقیه خودروها شناسایی می‌شود.

○ **قطعه‌بندی پاناپتیک (Panoptic Segmentation) :** ترکیبی از قطعه‌بندی معنایی و نمونه‌ای، که هم دسته کلی پیکسل‌ها و هم مرزهای دقیق هر نمونه خاص را تعیین می‌کند.

### • مراحل قطعه‌بندی تصویر:

- پیش‌پردازش تصویر: بهبود کیفیت تصویر و آماده‌سازی آن برای تحلیل پیکسلی.
- استخراج ویژگی‌ها: شناسایی ویژگی‌های مهم مانند لبه‌ها و تغییرات رنگ در تصویر.
- طبقه‌بندی پیکسل‌ها: هر پیکسل به یکی از دسته‌های موردنظر اختصاص داده می‌شود.
- تولید نقشه قطعه‌بندی: خروجی نهایی، یک نقشه از تصویر است که هر بخش آن با یک رنگ یا دسته خاص مشخص شده است.

### • تکنیک‌های رایج:

- **FCN (Fully Convolutional Network) :** شبکه عصبی کانولوشنی که از لایه‌های کانولوشنی کامل برای قطعه‌بندی تصویر استفاده می‌کند.
- **U-Net :** مدلی که به طور خاص برای تصاویر پزشکی طراحی شده و به دلیل ساختار U شکل آن، برای قطعه‌بندی دقیق تصاویر استفاده می‌شود.
- **Mask R-CNN :** ترکیبی از R-CNN برای شناسایی نواحی پیشنهادی و ماسک‌گذاری برای قطعه‌بندی دقیق هر نمونه.

### • کاربردها :

- تصاویر پزشکی: شناسایی دقیق نواحی مهم مانند تومورها یا بافت‌های خاص در تصاویر پزشکی.
- خودروهای خودران: قطعه‌بندی جاده‌ها، علائم ترافیکی، عابران پیاده و دیگر اشیاء برای کمک به خودروهای خودران در تحلیل محیط.
- پردازش تصاویر ماهواره‌ای: شناسایی نواحی مختلف مانند جنگل‌ها، شهرها و آب‌ها در تصاویر ماهواره‌ای.

○ بینایی رباتیک : کمک به ربات‌ها در درک دقیق‌تر از محیط اطراف برای انجام وظایف پیچیده‌تر.

## • چالش‌ها :

○ تغییرات پیچیده در تصویر : تغییرات شدید در نور، زاویه و جزئیات تصویر می‌تواند قطعه‌بندی را مشکل کند.

○ دقت بالا : نیاز به دقت بالای مدل‌ها در تشخیص مرزهای اشیاء و تفکیک نمونه‌ها.

○ مقیاس بزرگ تصاویر : تحلیل تصاویر با وضوح بالا یا داده‌های حجیم می‌تواند به قدرت محاسباتی بالایی نیاز داشته باشد.

## • TensorFlow

پس از آشنایی با مفاهیم پایه یادگیری ماشین، تصمیم گرفتم از کتابخانه قدرتمند TensorFlow برای پیاده‌سازی مدل‌های یادگیری عمیق استفاده کنم. TensorFlow یکی از محبوب‌ترین کتابخانه‌ها برای ساخت و آموزش مدل‌های یادگیری عمیق است که توسط Google توسعه داده شده است. با استفاده از این کتابخانه، توانستم شبکه‌های عصبی پیچیده‌تری بسازم و مدل‌هایی با دقت بالاتر برای تشخیص اشیاء در تصاویر ماهواره‌ای پیاده‌سازی کنم.

TensorFlow به من این امکان را داد که به راحتی از GPU (واحد پردازش گرافیکی) برای تسریع فرآیند آموزش مدل‌ها استفاده کنم و محاسبات سنگین را با سرعت بیشتری انجام دهم. در پروژه خود، از این ابزار برای پیاده‌سازی شبکه‌های عصبی کانولوشنی (Convolutional Neural Networks) و آموزش آن‌ها بر روی داده‌های تصویری استفاده کردم.

## • Google Colab

یکی از ابزارهای مهمی که در این پروژه استفاده کردم، Google Colab بود. Google Colab یک محیط اجرایی رایگان بر پایه Jupyter Notebook است که توسط Google ارائه شده و امکان استفاده از GPU را به صورت رایگان فراهم می‌کند. به دلیل محدودیت‌های سخت‌افزاری سیستم خودم و نیاز به قدرت پردازشی بیشتر برای آموزش مدل‌های یادگیری عمیق، از Google Colab برای اجرای مدل‌ها و آموزش آن‌ها با استفاده از GPU استفاده کردم.

Google Colab محیطی بسیار مناسب برای کار با پروژه‌های یادگیری عمیق است، چرا که نه تنها از کتابخانه‌هایی مانند TensorFlow پشتیبانی می‌کند، بلکه به راحتی می‌توان کدها را به اشتراک گذاشت و در محیطی تحت وب اجرا کرد. با استفاده از این ابزار، توانستم مدل‌های سنگین خود را بدون نیاز به سخت‌افزار پیشرفته بر روی GPU اجرا کنم و به نتایج سریعتری دست پیدا کنم.

The screenshot shows a Google Colab interface with a Jupyter Notebook. The code cell contains a command to train a YOLO model. The output displays training progress for epochs 13 through 17, including metrics like GPU memory, box loss, class loss, detection loss, instances, and size.

```
!yolo task=detect mode=train model=yolov8l.pt data={dataset.location}/data.yaml epochs=20 imgsz=640
```

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
13/20	10.5G	1.344	0.6584	1.026	302	640: 100% 94/94 [01:22<00:00, 1.14it/s]
Class		Images	Instances	Box(P	R	mAP50 mAP50-95): 100% 5/5 [00:03<00:00, 1.61it/s]
all		136	1923	0.953	0.959	0.98 0.568
14/20	10.4G	1.296	0.6256	1.006	284	640: 100% 94/94 [01:22<00:00, 1.14it/s]
Class		Images	Instances	Box(P	R	mAP50 mAP50-95): 100% 5/5 [00:03<00:00, 1.45it/s]
all		136	1923	0.971	0.959	0.98 0.603
15/20	10.5G	1.234	0.5862	0.9883	307	640: 100% 94/94 [01:21<00:00, 1.15it/s]
Class		Images	Instances	Box(P	R	mAP50 mAP50-95): 100% 5/5 [00:03<00:00, 1.39it/s]
all		136	1923	0.97	0.964	0.985 0.607
16/20	10.3G	1.205	0.5652	0.9877	283	640: 100% 94/94 [01:22<00:00, 1.14it/s]
Class		Images	Instances	Box(P	R	mAP50 mAP50-95): 100% 5/5 [00:03<00:00, 1.51it/s]
all		136	1923	0.968	0.974	0.984 0.627
17/20	10.5G	1.156	0.5369	0.9638	301	640: 100% 94/94 [01:21<00:00, 1.15it/s]
Class		Images	Instances	Box(P	R	mAP50 mAP50-95): 100% 5/5 [00:03<00:00, 1.60it/s]
all		136	1923	0.979	0.974	0.988 0.636

✓ 27s completed at 5:39 AM

شکل ۵ آموزش مدل

## • YOLO (You Only Look Once):

یکی از الگوریتم‌های محبوب و پرکاربرد در زمینه تشخیص اشیاء (Object Detection) است. YOLO به دلیل سرعت بالا و دقت خوبش، به ویژه در کاربردهایی که نیاز به تشخیص بلادرنگ (Real-Time Detection) دارند، مورد استفاده قرار می‌گیرد. این الگوریتم در تشخیص همزمان چندین شیء در یک تصویر بسیار مؤثر است.

[۶]



- **مفهوم کلی YOLO:**

YOLO یک رویکرد کاملاً جدید به تشخیص اشیاء دارد که برخلاف روش‌های قدیمی که تصویر را به بخش‌های مختلف تقسیم می‌کردند، کل تصویر را در یک نگاه (یا پاس) پردازش می‌کند. در واقع، YOLO تصویر را به چندین سلول شبکه‌ای تقسیم می‌کند و هر سلول وظیفه پیش‌بینی مرزهای اشیاء و دسته‌بندی آن‌ها را دارد. به این ترتیب، این الگوریتم تنها یک بار به تصویر نگاه می‌کند و تمام اشیاء موجود در آن را شناسایی می‌کند. [۶]

- **مراحل اصلی کار YOLO:**

- **تقسیم تصویر به سلول‌های شبکه‌ای (Grid Cells):**

در ابتدا، تصویر به یک شبکه  $S \times S$  تقسیم می‌شود. هر سلول از این شبکه وظیفه دارد که پیش‌بینی کند آیا در مرکز خود شیء‌ای وجود دارد یا خیر.

- **پیش‌بینی مرزهای اشیاء (Bounding Boxes):**

هر سلول از شبکه تعدادی مرز (Bounding Box) را به همراه یک امتیاز اعتماد (Confidence Score) پیش‌بینی می‌کند. امتیاز اعتماد نشان می‌دهد که چقدر مطمئن هستیم که یک شیء در مرز پیش‌بینی شده وجود دارد.

- **طبقه‌بندی اشیاء:**

برای هر مرز پیش‌بینی شده، YOLO همچنین دسته‌بندی شیء موجود را پیش‌بینی می‌کند. این دسته‌بندی می‌تواند هر چیزی مانند خودرو، انسان، سگ و ... باشد.

- **ادغام پیش‌بینی‌ها با Non-Max Suppression:**

الگوریتم YOLO ممکن است چندین مرز را برای یک شیء پیش‌بینی کند. برای جلوگیری از چندین پیش‌بینی اشتباه برای یک شیء، از تکنیکی به نام Non-Max Suppression استفاده می‌شود که تنها پیش‌بینی با بالاترین امتیاز اعتماد حفظ می‌شود و بقیه حذف می‌شوند.

- **نسخه‌های مختلف YOLO:**

- **Yolov<sup>۱</sup> :** اولین نسخه از YOLO که به عنوان یک مدل سریع اما کمتر دقیق نسبت به مدل‌های پیچیده‌تر مثل R-CNN معرفی شد.
- **YOLOv<sup>۲</sup> و YOLOv<sup>۳</sup> :** در این نسخه‌ها بهبودهای بسیاری از نظر دقت و سرعت اعمال شد. YOLOv<sup>۲</sup> توانایی تشخیص بهتر و دقیق‌تر را ارائه داد و YOLOv<sup>۳</sup> از شبکه‌های عمیق‌تر و ویژگی‌های چند مقیاسی برای بهبود عملکرد استفاده کرد.
- **YOLOv<sup>۴</sup> و YOLOv<sup>۵</sup> :** این نسخه‌ها از تکنیک‌های پیشرفته‌تری مانند ویژگی‌های کانال-محور و بهینه‌سازی‌های مربوط به یادگیری برای افزایش کارایی و دقت استفاده می‌کنند.
- و.... که هم اکنون YOLOv<sup>۱۱</sup> نیز منتشر شده در هر سری از این نسخه‌ها معماری این نسخه‌ها نیز متفاوت است

### • مزایای YOLO:

- **سرعت بالا :** YOLO به دلیل معماری خاصش قادر است در زمان بسیار کوتاهی تصاویر را پردازش کرده و اشیاء را تشخیص دهد، به طوری که برای کاربردهای بلادرنگ بسیار مناسب است.
- **دقت مناسب :** با وجود سرعت بالا، YOLO دقت مناسبی را در تشخیص اشیاء ارائه می‌دهد و می‌تواند اشیاء مختلف را به خوبی شناسایی و طبقه‌بندی کند.

### • معایب YOLO:

- **مشکلات در تشخیص اشیاء کوچک :** به دلیل تقسیم تصویر به شبکه‌های سلولی، YOLO ممکن است در تشخیص اشیاء بسیار کوچک با دقت بالا دچار مشکل شود.
- **محدودیت در دقت بالا :** گرچه YOLO دقت خوبی دارد، اما نسبت به روش‌های دقیق‌تری مثل Faster R-CNN در موارد خاص ممکن است عملکرد ضعیف‌تری داشته باشد.

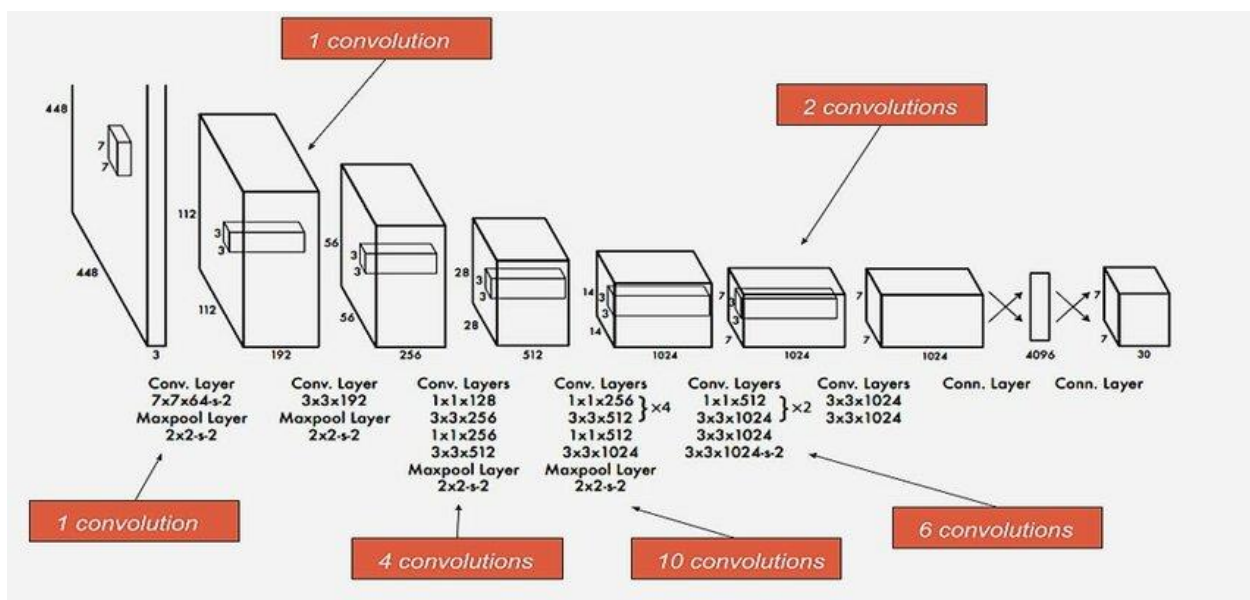
### • کاربردهای YOLO :

- **خودروهای خودران :** تشخیص علائم ترافیکی، خودروها، عابران پیاده و سایر اشیاء در زمان واقعی برای خودروهای خودران.
- **نظارت و امنیت :** شناسایی و ردیابی افراد و اشیاء در محیط‌های مختلف برای نظارت و افزایش امنیت.
- **بینایی رباتیک :** کمک به ربات‌ها برای شناسایی و تعامل با اشیاء در محیط‌های پیچیده.

- تجزیه و تحلیل تصاویر و ویدئوها : استفاده در زمینه‌هایی مانند تحلیل تصاویر پزشکی، سیستم‌های هوشمند، و تشخیص فعالیت‌ها در ویدئوها.

## • چالش‌های YOLO:

- تغییرات نور و زاویه : عملکرد ممکن است تحت تأثیر تغییرات نوری و زاویه‌های مختلف تصویر قرار بگیرد.
- تعادل بین سرعت و دقت : در حالی که YOLO بسیار سریع است، اما تعادل بین سرعت و دقت همواره یکی از چالش‌های این الگوریتم بوده است.



شکل ۶ لایه‌های مدل یولو

## • Visual Studio Code (VSCode)

پس از آشنایی با Google Colab، تصمیم گرفتم از Visual Studio Code (VSCode) برای توسعه و ویرایش کدها استفاده کنم. VSCode یکی از محبوب‌ترین ویرایشگرهای کد است که به دلیل امکانات گسترده، اکستنشن‌های کاربردی، و محیط کاربری کاربرپسند، ابزاری عالی برای توسعه‌دهندگان محسوب می‌شود.

یکی از ویژگی‌های کلیدی VSCode، امکان استفاده از Jupyter Notebook در داخل این ویرایشگر است. با نصب اکستنشن‌های مربوط به Jupyter، توانستم همان تجربه اجرای کدها به صورت "باکس باکس" را در VSCode داشته باشم. این ویژگی به من اجازه داد که به راحتی کدهایم را در محیطی آفلاین اجرا کنم و از امکانات VSCode برای مدیریت پروژه و اجرای کدها بهره ببرم.

```

from ultralytics import YOLO
import numpy as np
from PIL import Image
import cv2

model = YOLO("yolov8n.pt")
image = Image.open(r"C:\Users\Hirad\Desktop\top-view-of-kyiv-traffic-cars-and-snow-video.jpg")
image = np.asarray(image)
results = model.predict(image)
print(results)

0: 384x640 5 cell phones, 359.7ms
Speed: 15.6ms preprocess, 359.7ms inference, 0.0ms postprocess per image at shape (1, 3, 384, 640)
[ultralytics.engine.results.Results object with attributes:
  boxes: ultralytics.engine.results.Boxes object
  keypoints: None
  masks: None
  names: {0: 'person', 1: 'bicycle', 2: 'car', 3: 'motorcycle', 4: 'airplane', 5: 'bus', 6: 'train', 7: 'truck', 8: 'boat', 9: 'traffic light', 10: 'fire hydrant', :
  obb: None
  orig_img: array([[[[ 77, 79, 74],

```

شکل ۷ اجرا گرفتن در محیط محلی

## • ایجاد محیط مجازی (Virtual Environment)

به منظور مدیریت کتابخانه‌ها و وابستگی‌های مختلف پروژه، از محیط مجازی (Virtual Environment) استفاده کردم. با ایجاد محیط‌های مجازی جداگانه، توانستم کتابخانه‌های مختلف را نصب کرده و بدون نگرانی از تداخل نسخه‌ها یا پیکربندی‌ها، پروژه‌های مختلف را مدیریت کنم. محیط مجازی به من اجازه داد تا به راحتی نسخه‌های خاصی از کتابخانه‌های مورد نیازم را نصب کنم و پروژه‌هایم را به صورت ایزوله و مجزا از هم نگه دارم. در پروژه خود، از این ویژگی برای نصب YOLOv۱۱ و سایر کتابخانه‌های مورد نیاز استفاده کردم. این کار باعث شد که تمام وابستگی‌های پروژه به صورت مجزا مدیریت شود و از بروز مشکلات احتمالی در پیکربندی سیستم جلوگیری شود.

## • Roboflow

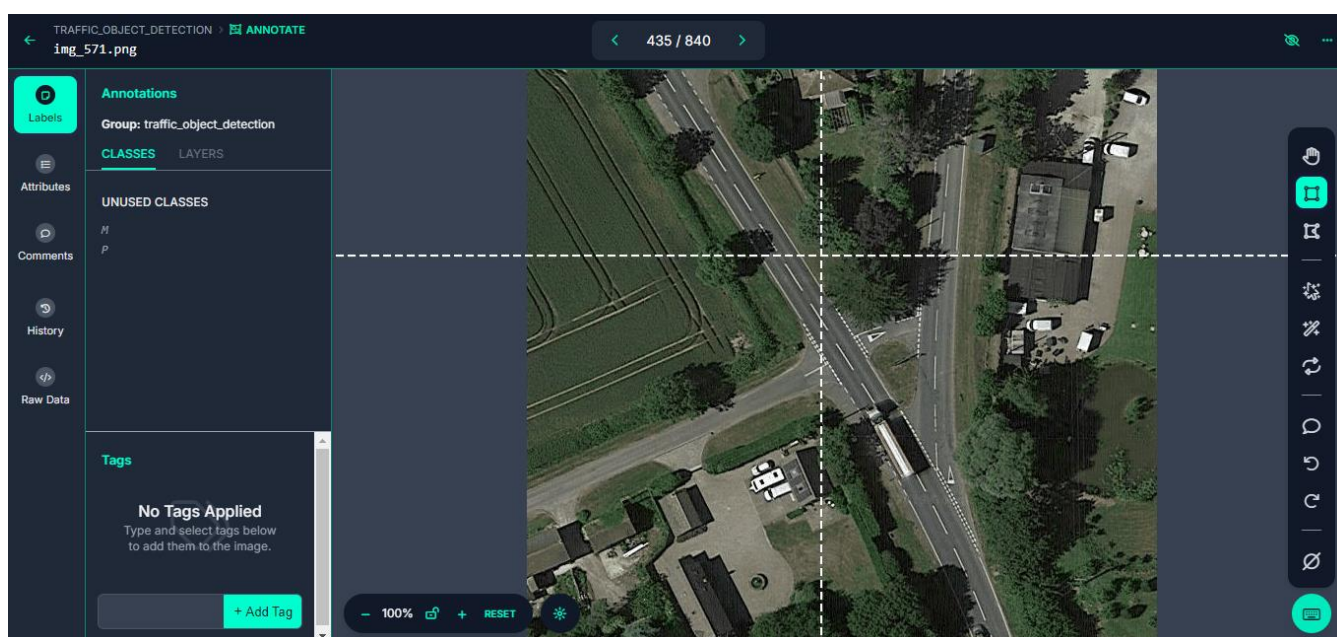
**Roboflow** یک پلتفرم کاربردی برای جمع‌آوری، برچسب‌گذاری، و مدیریت مجموعه داده‌های تصویری است که به ویژه در زمینه‌های مرتبط با بینایی کامپیوتر مانند تشخیص اشیاء، تقسیم‌بندی تصویر، و تشخیص متون به کار می‌رود. این پلتفرم به کاربران اجازه می‌دهد تا به سرعت مجموعه داده‌های خود را آپلود کنند، آن‌ها را برچسب‌گذاری کنند و برای آموزش مدل‌های یادگیری عمیق آماده کنند.

از قابلیت‌های کلیدی **Roboflow** می‌توان به موارد زیر اشاره کرد:

- **پیش‌پردازش داده‌ها Roboflow**: به کاربران این امکان را می‌دهد که داده‌های تصویری خود را به راحتی پیش‌پردازش کنند. این شامل تغییر اندازه تصاویر، انجام تغییرات مختلف بر روی تصاویر مانند چرخش، برش، و افزایش داده‌ها (Data Augmentation) برای بهبود عملکرد مدل است.
- **فرمت‌های متنوع**: این پلتفرم از فرمت‌های مختلف داده برای آموزش مدل‌های یادگیری عمیق پشتیبانی می‌کند. شما می‌توانید داده‌هایتان را به فرمت‌های سازگار با مدل‌های معروف مانند YOLO، TensorFlow، PyTorch، و Detectron صادر کنید.
- **برچسب‌گذاری و مدیریت داده‌ها Roboflow**: ابزارهای قدرتمندی برای برچسب‌گذاری دستی و خودکار تصاویر فراهم می‌کند. این ابزارها برای تشخیص اشیاء، تقسیم‌بندی، و دیگر کاربردهای بینایی کامپیوتر به کار گرفته می‌شوند.
- **افزایش داده‌ها (Data Augmentation)**: یکی از ویژگی‌های بسیار کاربردی Roboflow، ارائه تکنیک‌های متنوع افزایش داده‌ها است که به بهبود عملکرد مدل‌های یادگیری عمیق کمک می‌کند. این ویژگی‌ها شامل تغییرات رنگ، چرخش، کشش و دیگر تغییرات بر روی تصاویر است که به تقویت مجموعه داده‌ها کمک می‌کند.
- **اشتراک‌گذاری و همکاری Roboflow**: به کاربران اجازه می‌دهد تا پروژه‌های خود را به صورت عمومی یا خصوصی به اشتراک بگذارند و به دیگران دسترسی دهند تا بر روی آن‌ها کار کنند. این امکان، همکاری تیمی را تسهیل می‌کند.

○ ادغام با مدل‌های یادگیری عمیق: پس از آماده‌سازی مجموعه داده، شما می‌توانید آن را برای آموزش مدل‌های مختلف یادگیری عمیق استفاده کنید Roboflow. از انواع چارچوب‌های یادگیری عمیق پشتیبانی می‌کند و به شما امکان می‌دهد داده‌هایتان را به راحتی در پروژه‌های مختلف استفاده کنید.

به طور کلی، Roboflow یک پلتفرم کارآمد برای آماده‌سازی داده‌های تصویری و تسريع فرآیند آموزش مدل‌های یادگیری عمیق است و در پروژه‌هایی که به تشخیص و تقسیم‌بندی اشیاء نیاز دارند، به خوبی مورد استفاده قرار می‌گیرد.



شکل ۸ محیط و انجام برچسب گذاری

## Kaggle •

Kaggle یک پلتفرم محبوب برای اجرای پروژه‌های یادگیری ماشین و یادگیری عمیق است که به کاربران امکان استفاده از GPU و TPU رایگان را می‌دهد. این قابلیت باعث می‌شود تا فرآیند آموزش مدل‌ها، به‌ویژه در پروژه‌های بزرگ و پیچیده مانند تشخیص اشیاء و تقسیم‌بندی تصاویر، بسیار سریع‌تر انجام شود. کاربران می‌توانند از نوت‌بوک‌های مبتنی بر کلاود برای نوشتن، اجرای کد و به اشتراک گذاری نتایج استفاده کنند. این پلتفرم همچنین محیطی مناسب برای همکاری تیمی و استفاده از مجموعه داده‌های آماده است، که می‌تواند به تسریع فرآیند توسعه پروژه‌ها کمک کند.

The screenshot shows a Kaggle notebook titled 'notebook5442ad5b21'. The code cell contains the following output:

```

100/100 14.4G 0.338 0.682 0.2866 0.9167 29
Class Images Instances Box(P R mAP50 m
all 352 931 0.966 0.966 0.98 0.939 0.968

0.968 0.981 0.887

100 epochs completed in 9.005 hours.
Optimizer stripped from runs/segment/train/weights/last.pt, 124.8MB
Optimizer stripped from runs/segment/train/weights/best.pt, 124.8MB

Validating runs/segment/train/weights/best.pt...
Ultralytics 8.3.7 Python-3.10.14 torch-2.4.0 CUDA:0 (Tesla P100-PCIE-16GB, 16269MiB)
YOLO11x-seg summary (fused): 491 layers, 62,005,593 parameters, 0 gradients, 318.5 GFLOPs
Class Images Instances Box(P R mAP50 m
all 352 931 0.967 0.969 0.982 0.94 0.968
0.97 0.98 0.89 72 86 0.964 1 0.991 0.969 0.964
Crowded Street
1 0.991 0.94 339 670 0.955 0.919 0.962 0.886 0.957
No Traffic
0.921 0.957 0.811 144 175 0.982 0.989 0.992 0.966 0.982
Traffic
0.989 0.992 0.918
Speed: 0.2ms preprocess, 44.2ms inference, 0.0ms loss, 0.8ms postprocess per image
Results saved to runs/segment/train
Learn more at https://docs.ultralytics.com/modes/train

```

The right sidebar shows the 'Notebook' panel with sections for 'Input', 'Output', and 'Table of contents'. The 'Table of contents' section indicates 'No sections detected' and suggests 'Add markdown headers to add a section'.

شکل ۹ محیط کار در Kaggle

## • نتیجه گیری

پروژه حاضر با هدف تحلیل و شناسایی وضعیت ترافیک با استفاده از ابزارها و تکنیک‌های پیشرفته در حوزه یادگیری عمیق و بینایی کامپیوتر انجام شد. با استفاده از زبان برنامه‌نویسی پایتون و کتابخانه‌هایی مانند TensorFlow و مدل YOLO، توانستیم به نتایجی دقیق و مؤثر در شناسایی و طبقه‌بندی اشیاء در تصاویر ترافیکی دست پیدا کنیم. همچنین استفاده از پلتفرم‌های Roboflow و Kaggle برای برچسب‌گذاری داده‌ها و آموزش مدل‌ها، به ما امکان بهبود عملکرد مدل‌ها و تسریع فرآیند توسعه را داد.

مهم‌ترین دستاوردهای این پروژه شامل بهبود دقت در شناسایی اشیاء ترافیکی، کاهش خطاهای پیش‌بینی و استفاده بهینه از منابع محاسباتی بود. از طریق این سیستم، امکان بهبود مدیریت ترافیک و افزایش کارایی در کنترل و هدایت جریان ترافیک در شرایط مختلف فراهم شد.

در نهایت، این پروژه نشان داد که استفاده از تکنیک‌های یادگیری عمیق، به‌ویژه مدل‌های پیشرفته مانند YOLO، می‌تواند در کاربردهای بلادرنگ (Real-Time) مؤثر باشد و به‌طور قابل توجهی به کاهش ترافیک و افزایش ایمنی جاده‌ها کمک کند. با توجه به پیشرفت‌های آینده در این زمینه، انتظار می‌رود که سیستم‌های هوشمند ترافیکی بهبود یابند و قابلیت‌های بیشتری به آن‌ها افزوده شود.



## **فصل سوم جمع اوری داده ها و مدل مناسب**

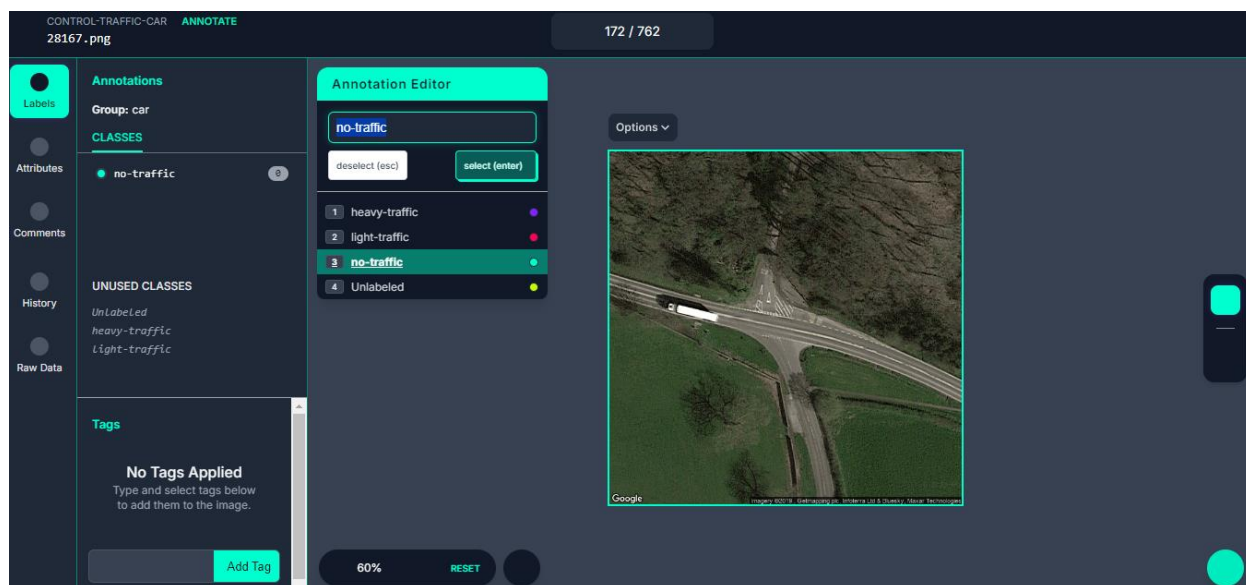
## • مقدمه

در فصل "جمع‌آوری داده‌ها و آموزش ماشین"، به بررسی فرآیندهای اصلی آماده‌سازی داده‌ها و آموزش مدل‌های یادگیری ماشین می‌پردازیم. داده‌ها به عنوان یکی از مهم‌ترین ارکان یادگیری ماشین، نقشی کلیدی در کیفیت و دقت مدل‌های یادگیری ایفا می‌کنند. از این رو، ابتدا مراحل جمع‌آوری، پاک‌سازی و برچسب‌گذاری داده‌ها را شرح می‌دهیم. سپس، به انتخاب مدل مناسب و روش‌های بهینه‌سازی آن برای دستیابی به نتایج بهتر می‌پردازیم. در نهایت، چگونگی استفاده از یادگیری انتقالی و روش‌های افزایش کارایی مدل‌ها را توضیح خواهیم داد.

## • جمع‌آوری داده، پاک‌سازی و برچسب‌گذاری

برای این پروژه، فرآیند جمع‌آوری داده‌ها یکی از چالش‌های پیچیده و زمان‌بر بود. در ابتدا، تلاش زیادی کردم تا از منابع معتبر و شناخته‌شده‌ای مانند Google Earth، Google Maps، Kaggle و چندین سایت دیگر، داده‌های آماده و مناسب برای نیازهای خاص این پروژه پیدا کنم. متأسفانه، باوجود جستجوهای متعدد و بررسی منابع مختلف، هیچ دیتاستی که به طور کامل با نیازهای من سازگار باشد، در دسترس نبود. این موضوع من را وادار کرد تا رویکرد متفاوتی اتخاذ کنم و به صورت دستی داده‌ها را جمع‌آوری کنم.

برای این منظور، از Google Earth و سایر منابع با تصاویر جهانی با کیفیت بالا استفاده کردم. ابتدا با دقت بالا مناطق مورد نظر را پیدا کردم و با زوم کردن روی جزئیات، از تصاویر اسکرین‌شات گرفتم. این مرحله نه تنها زمان زیادی از من گرفت، بلکه به دلیل دقتی که در انتخاب هر تصویر نیاز بود، به صورت مداوم بین منابع مختلف حرکت می‌کردم و برای به دست آوردن تصاویر با وضوح بالا، بارها و بارها مناطق مختلف را بررسی می‌کردم. به طور خلاصه، هر تصویری به صورت دستی و با جزئیات فراوان انتخاب شد تا بتوانم از آن برای مرحله‌ی بعدی استفاده کنم.



شکل ۱۰ برچسب زدن داده ها

یکی از بزرگترین چالش‌ها در این فرآیند، کیفیت تصاویر بود. پس از جمع‌آوری اسکرین‌شات‌ها، لازم بود کیفیت آنها را بهبود ببخشم تا مدل یادگیری ماشینی بتواند بهترین نتیجه را از این داده‌ها بگیرد. برای این منظور، از ابزار قدرتمندی مثل Gigapixel AI استفاده کردم که به من امکان داد وضوح و جزئیات تصاویر را به طور چشمگیری افزایش دهد. این ابزار بهینه‌سازی تصاویر را به صورت هوشمند انجام داد و به من کمک کرد تا تصاویری با کیفیت بهتر و جزئیات بیشتر در اختیار داشته باشم.

در مرحله بعد، نوبت به برچسب‌گذاری دقیق تصاویر رسید. برای این کار از پلتفرم Roboflow استفاده کردم. هر تصویر را با دقت بالا و توجه به جزئیات برچسب‌گذاری کردم تا مدل یادگیری ماشینی بتواند با داده‌های دقیق و کاملی آموزش ببیند. این مرحله، با وجود حجم بالای تصاویر و دقت مورد نیاز، یکی از مهم‌ترین بخش‌های پروژه بود، چرا که برچسب‌گذاری دقیق تأثیر مستقیمی بر دقت نهایی مدل داشت.

## • استفاده از داده‌های Roboflow

برای برچسب‌گذاری داده‌ها، از Roboflow استفاده کردم. این سایت به ابزار فوق‌العاده برای آماده‌سازی و برچسب‌گذاری تصاویر که کمک کرد داده‌ها را به راحتی مدیریت کنم و کیفیت برچسب‌گذاری‌ها را بالا ببرم. این ابزار به من کمک کرد تا سریع‌تر به داده‌های باکیفیت و آماده برای آموزش برسیم.

در مجموع، فرآیند جمع‌آوری و آماده‌سازی داده‌ها، اگرچه به ظاهر ساده به نظر می‌رسید، اما به دلیل نبودن دیتاست‌های آماده و نیاز به انجام دستی بسیاری از مراحل، بسیار زمان‌بر و پیچیده بود. با این حال، نتیجه نهایی یک مجموعه داده با کیفیت بالا بود که می‌توانست پایه‌ای مطمئن برای آموزش مدل باشد.

بعد از جمع‌آوری تصاویر، داده‌ها به سه دسته سگمنت تقسیم‌بندی می‌کنیم:

○ **ترافیک:** زمانی که تعداد ماشین‌هایی که پشت چراغ قرمز توقف کرده‌اند، یا در خیابان توقف کردند پشت سرهم.

○ **خیابان شلوغ:** زمانی که تعداد ماشین‌ها درون خیابان هستند به قدری نیست که باعث توقف خودروها شود اما با سرعت کمی در حرکت هستند.

○ **بدون ترافیک:** زمانی که تعداد ماشین‌ها خیلی کم یا هیچ خودرویی در تصویر وجود نداشته باشد.

برای برچسب‌گذاری دقیق داده‌ها، از پلتفرم Roboflow استفاده شد. این پلتفرم نه تنها امکان برچسب‌گذاری داده‌ها را فراهم کرد، بلکه ابزارهای قدرتمندی برای پیش‌پردازش و افزایش داده‌ها (Data Augmentation) ارائه داد. از آنجایی که تعداد داده‌های اصلی کم بود، از تکنیک‌های داده‌افزایی استفاده شد تا حجم داده‌ها افزایش یابد.

یکی از چالش‌های اصلی در این پروژه عدم توازن کلاس‌ها بود؛ به این معنا که تعداد داده‌های مربوط به برخی کلاس‌ها کمتر از بقیه بود. برای مدیریت این مسئله، در کدگذاری پروژه، ما داده‌افزایی را، روی داده‌های که تعداد کمتری دارند انجام می‌دهیم تا داده‌های ما به تعادل نسبی برسند.

## • پیش‌پردازش داده‌ها و آماده‌سازی برای آموزش

برای بهبود دقت مدل و آماده‌سازی داده‌ها جهت آموزش، مراحل مختلفی از پیش‌پردازش روی تصاویر انجام گرفت. یکی از اولین گام‌ها تنظیم اندازه تصاویر بود. تمام تصاویر به یک اندازه استاندارد تغییر سایز داده شدند تا مدل بتواند ورودی‌های یکنواخت و همگنی داشته باشد. این کار باعث کاهش پیچیدگی مدل و افزایش سرعت پردازش شد.

علاوه بر تغییر سایز، تنظیم روشنایی تصاویر نیز یکی از اقدامات مهم در پیش‌پردازش بود. تصاویر ماهواره‌ای ممکن است به دلیل شرایط نوری مختلف دارای روشنایی‌های متفاوتی باشند که می‌تواند بر کیفیت تشخیص مدل تأثیر منفی بگذارد. بنابراین، با تنظیم روشنایی و یکنواخت‌سازی نور تصاویر، این مشکل تا حد زیادی برطرف شد.

همچنین از تکنیک‌های داده‌افزایی (Data Augmentation) برای افزایش تنوع داده‌ها استفاده شد. به‌طور مثال، تصاویر چرخانده شدند یا تغییرات کوچکی در آن‌ها اعمال شد تا مدل بتواند به خوبی با شرایط مختلف روبرو شود. این تکنیک به ما کمک کرد تا تعداد تصاویر موثر در هر کلاس افزایش یابد و مدل در برابر داده‌های جدید عملکرد بهتری داشته باشد.

این پیش‌پردازش‌ها به همراه برچسب‌گذاری دقیق داده‌ها، مجموعه‌ای مناسب را برای آموزش مدل فراهم کردند. داده‌ها پس از پیش‌پردازش به دو دسته تقسیم شدند: مجموعه داده آموزش و مجموعه داده اعتبارسنجی، که به مدل کمک کرد به‌صورت بهینه‌تری آموزش ببیند.

## • انتخاب مدل مناسب

در این پروژه از مدل YOLOv11 برای آموزش و تشخیص اشیاء استفاده شد. این مدل به دلیل سرعت بالا و دقت مناسب، یکی از بهترین انتخاب‌ها برای مسائل مرتبط با تشخیص و طبقه‌بندی اشیاء در تصاویر محسوب می‌شود. YOLOv11 به‌ویژه برای پردازش تصاویر در زمان واقعی (Real-Time) طراحی شده است و توانایی شناسایی دقیق و سریع اشیاء مختلف در تصاویر را داراست.

YOLOv11 با استفاده از معماری بهینه شده‌اش، قابلیت‌های بالایی در تشخیص اشیاء با اندازه‌ها و موقعیت‌های مختلف دارد. این مدل از روش‌هایی مانند Feature Pyramid Network (FPN) و Path Aggregation Network (PAN) برای استخراج ویژگی‌های چندسطحی و افزایش دقت در شناسایی اشیاء استفاده می‌کند. همچنین، YOLOv11 از تکنیک‌های مدرن مانند Anchor-Free Detection برای کاهش پیچیدگی پردازش و بهبود سرعت مدل بهره می‌برد.

در این پروژه، ابتدا از YOLOv11 به عنوان مدل اصلی برای تشخیص ترافیک و خودروها استفاده شد. مدل با استفاده از داده‌های برجسته‌گذاری شده، توانست به‌طور دقیق وضعیت ترافیک را طبقه‌بندی کند. آموزش مدل با استفاده از تکنیک‌های افزایش داده‌ها (Data Augmentation) و بهینه‌سازی‌های پیشرفته انجام شد تا عملکرد مدل در شرایط نوری و زوایای مختلف تصاویر بهبود یابد.

این رویکرد به ما این امکان را داد که از قابلیت‌های YOLOv11 برای تشخیص ترافیک در زمان واقعی بهره ببریم و دقت بالاتری در تحلیل وضعیت ترافیک نسبت به مدل‌های سنتی داشته باشیم. برخلاف مدل‌های پیچیده‌تر مانند EfficientNet، که بیشتر برای استخراج ویژگی‌های عمومی تصاویر استفاده می‌شوند، YOLOv11 به‌طور خاص برای تشخیص بلادرنگ در کاربردهایی مانند ترافیک و امنیت طراحی شده است.

در نتیجه، استفاده از YOLOv11 به ما امکان داد که ترافیک را با دقت بالا و سرعت سریع تشخیص دهیم و به این وسیله کارایی سیستم‌های مدیریت ترافیک را بهبود بخشیم.

## • دلیل استفاده از مدل‌های از پیش آموزش دیده شده

در این پروژه، از YOLOv11 به عنوان مدل اصلی برای یادگیری انتقالی (Transfer Learning) استفاده شد. این روش به ما امکان می‌دهد که مدل از دانش و ویژگی‌هایی که در مراحل پیشین بر روی داده‌های گسترده‌ای آموخته است، بهره‌برد و دیگر نیازی نباشد همه چیز از ابتدا آموزش داده شود. استفاده از یادگیری انتقالی باعث صرفه‌جویی در زمان آموزش و همچنین افزایش دقت مدل می‌شود، چرا که مدل قبلاً بر روی مجموعه‌های بزرگی از داده‌ها مانند COCO یا ImageNet آموزش دیده است.

در پروژه حاضر، داده‌های برچسب گذاری شده به مدل YOLOv11 داده شد تا مدل از طریق یادگیری انتقالی بتواند ویژگی‌های عمومی و مرتبط با تشخیص اشیاء و ترافیک را بیاموزد. مدل YOLOv11، که قبلاً روی داده‌های بزرگ آموزش دیده بود، توانست به سرعت ویژگی‌های تصاویر ترافیکی را استخراج کند و با دقت بالاتری طبقه‌بندی‌ها را انجام دهد. این رویکرد هم از نظر کارایی و هم از نظر دقت، بهبود قابل توجهی در تحلیل ترافیک به همراه داشت. در نهایت، انتخاب YOLOv11 به عنوان مدل یادگیری انتقالی باعث شد که دقت بالایی در طبقه‌بندی تصاویر و تشخیص ترافیک به دست آید، که یکی از اهداف اصلی پروژه بود.

## • یادگیری انتقالی یا (Transfer Learning):

یادگیری انتقالی (Transfer Learning) یکی از تکنیک‌های مؤثر در حوزه یادگیری عمیق است که به ما امکان می‌دهد از دانش و تجربه یک مدل از پیش آموزش دیده شده روی داده‌های عمومی، برای حل مسائل جدید استفاده کنیم. در این روش، به جای آموزش مدل از ابتدا، از یک مدل که قبلاً روی یک مجموعه داده بزرگ و عمومی (مثل ImageNet) آموزش دیده، استفاده می‌شود. این مدل ویژگی‌های اصلی و عمومی داده‌ها را از قبل یاد گرفته است و ما می‌توانیم آن را با تنظیم دقیق‌تر برای نیازهای خاص پروژه خود دوباره آموزش دهیم. [۷]

## • مزایای یادگیری انتقالی:

- کاهش زمان و هزینه‌های محاسباتی: با استفاده از مدل‌های از پیش آموزش دیده شده، نیازی به آموزش مدل از صفر نیست و این باعث می‌شود زمان و منابع محاسباتی کمتری نیاز باشد. [۸]
- بهبود دقت با داده‌های کمتر: مدل‌هایی که قبلاً روی داده‌های بزرگ آموزش دیده‌اند، به خوبی می‌توانند الگوهای عمومی را شناسایی کنند. وقتی این مدل‌ها را روی مجموعه داده‌های کوچک‌تری دوباره آموزش می‌دهیم، می‌توانند دقت بالاتری را حتی با داده‌های محدود به دست آورند. [۸]
- یادگیری بهتر ویژگی‌های پیچیده: مدل‌های از پیش آموزش دیده شده، از تجربیات گسترده خود روی داده‌های مختلف استفاده می‌کنند تا ویژگی‌های پیچیده را بهتر درک کنند، که این امر به مدل کمک می‌کند در شناسایی دقیق‌تر اشیاء و الگوها بهتر عمل کند. [۸]

## • روش‌های افزایش کارایی مدل‌ها:

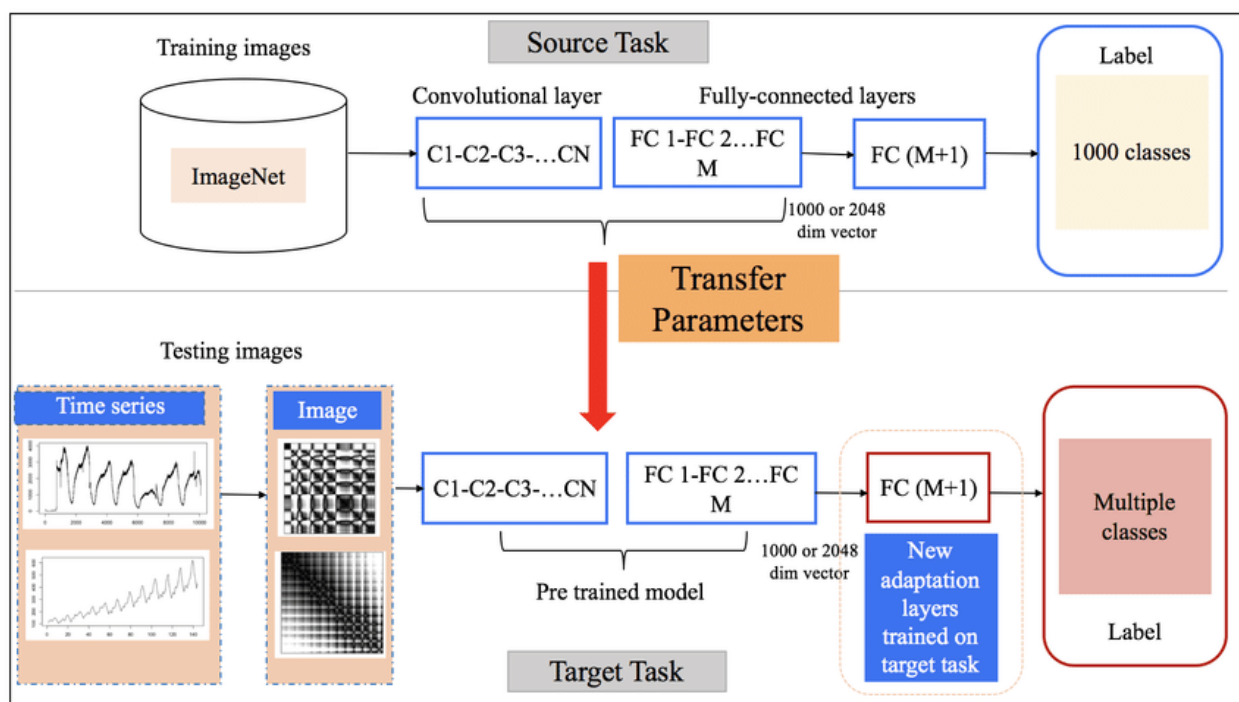
- برای بهبود عملکرد مدل‌ها، علاوه بر استفاده از یادگیری انتقالی، می‌توان از چندین تکنیک پیشرفته بهره برد:
- تنظیم دقیق (Fine-Tuning): در این مرحله، بخش‌هایی از مدل که بیشتر به داده‌های عمومی وابسته‌اند ثابت نگه داشته می‌شوند و فقط لایه‌های بالاتر یا تخصصی‌تر مدل با داده‌های جدید دوباره آموزش داده می‌شوند. این کار باعث می‌شود مدل روی داده‌های خاص پروژه بهتر عمل کند. [۹]
  - استفاده از داده‌های افزوده (Data Augmentation): یکی از روش‌های مهم برای بهبود عملکرد مدل‌ها، افزایش تنوع داده‌هاست. با اعمال تغییرات مختلفی مثل چرخاندن، برش، تغییر روشنایی و رنگ در تصاویر، می‌توان مجموعه داده‌های جدیدی ایجاد کرد که به مدل کمک می‌کند در شرایط واقعی بهتر عمل کند و بهبود کارایی را به همراه داشته باشد. [۹]
  - تنظیم نرخ یادگیری (Learning Rate): با استفاده از تکنیک‌های تنظیم نرخ یادگیری، می‌توان فرآیند یادگیری مدل را بهبود داد. در ابتدا از نرخ یادگیری بالا استفاده می‌شود تا مدل سریع‌تر یاد بگیرد، و در مراحل پایانی، نرخ یادگیری کاهش می‌یابد تا مدل دقیق‌تر شود. [۹]
  - جلوگیری از بیش‌برازش (Overfitting): یکی از مشکلاتی که ممکن است در آموزش مدل‌ها به وجود بیاید، بیش‌برازش است، یعنی مدل روی داده‌های آموزشی بیش از حد تمرکز کند و در تعمیم‌دهی



به داده‌های جدید ضعیف عمل کند. برای جلوگیری از این مشکل، تکنیک‌هایی مثل Dropout یا Regularization استفاده می‌شود که باعث افزایش توانایی مدل در تعمیم‌دهی به داده‌های جدید می‌شود. [۹]

این روش‌ها در کنار یادگیری انتقالی، به مدل‌ها کمک می‌کنند که هم در زمان کمتر و هم با دقت بالاتر به نتایج مطلوب برسند و در پروژه‌های مختلف عملکرد بهتری داشته باشند.

به صورت کلی انتقال یادگیری یعنی اینکه ما از لایه‌های CNN داخل لایه‌های آموزشی یک ماشین دیگه استفاده کنیم اینکار باعث میشه که چیزهای که از قبل یاد گرفته رو به خاطر داشته که همین کار باعث میشه وظیفه جدید رو با کیفیت و دقت بالاتری یاد بگیره قدرت تشخیص و تصمیم بهتری پیدا می‌کند.



شکل ۱۱ انتقال یادگیری

## • آموزش مدل

پس از جمع آوری و برچسب گذاری داده‌ها، مرحله مهم آموزش مدل آغاز شد. هدف از این مرحله، بهینه‌سازی مدل انتخاب شده برای انجام وظایف تعیین شده در پروژه است. در این پروژه از مدل YOLOv11 برای تشخیص و طبقه‌بندی ترافیک استفاده شد. این مدل به دلیل سرعت بالا و دقت قابل توجه، انتخابی مناسب برای تحلیل تصاویر ترافیکی در زمان واقعی بود.

## • مراحل آموزش مدل:

- **تنظیم پارامترهای آموزش:** در این مرحله، پارامترهایی مانند نرخ یادگیری، تعداد اپوک‌ها، و اندازه دسته‌ها تعیین شدند. نرخ یادگیری در ابتدا به‌طور آزمایشی با مقادیر مختلف تنظیم شد تا بهترین عملکرد به دست آید. تعداد اپوک‌ها نیز بر اساس اندازه داده‌ها و پیچیدگی مدل تعیین شد.
- **یادگیری انتقالی (Transfer Learning):** برای بهبود سرعت و دقت آموزش، از یادگیری انتقالی استفاده شد. مدل YOLOv11 که قبلاً روی مجموعه داده‌هایی مانند COCO آموزش دیده بود، به عنوان پایه مورد استفاده قرار گرفت. سپس مدل با استفاده از داده‌های پروژه برای وظایف خاص، مانند طبقه‌بندی وضعیت ترافیک، تنظیم شد.
- **افزایش داده‌ها (Data Augmentation):** برای بهبود عملکرد مدل در شرایط نوری و زوایای مختلف تصاویر، از تکنیک‌های افزایش داده‌ها مانند چرخش، تغییر مقیاس و تغییرات در رنگ تصاویر استفاده شد. این کار کمک کرد تا مدل تنوع بیشتری در داده‌ها تجربه کند و در شرایط مختلف بهتر عمل کند.



## • نتیجه گیری

در این فصل، به تفصیل فرآیند جمع‌آوری داده‌ها، برچسب‌گذاری دقیق و استفاده از مدل‌های یادگیری عمیق برای تشخیص اشیاء و وضعیت ترافیک بررسی شد. استفاده از ابزارهایی مانند Google Earth و Roboflow برای جمع‌آوری و آماده‌سازی داده‌ها و همچنین به کارگیری مدل YOLOv11 به دلیل سرعت بالا و دقت مناسب، امکان تشخیص بلادرنگ و دقیق ترافیک را فراهم کرد.

استفاده از یادگیری انتقالی به ما این امکان را داد که مدل از دانش کسب‌شده در مراحل قبلی بهره‌مند شود و نیاز به آموزش از صفر نداشته باشد. به این ترتیب، دقت مدل در طبقه‌بندی تصاویر و تحلیل وضعیت ترافیک به طور چشمگیری افزایش یافت. همچنین، تکنیک‌هایی مانند افزایش داده‌ها (Data Augmentation) و تنظیم نرخ یادگیری نقش کلیدی در بهبود کارایی مدل داشتند و به ما کمک کردند تا مدل با داده‌های مختلف و شرایط نوری متفاوت بهتر سازگار شود.

در مجموع، این فصل نشان داد که ترکیب مناسب ابزارهای پردازش تصویر، تکنیک‌های یادگیری عمیق، و استفاده از مدل‌های از پیش آموزش دیده‌شده، تأثیر مثبتی بر روی دقت و کارایی سیستم مدیریت ترافیک داشت و بهبود قابل توجهی در تحلیل وضعیت ترافیک فراهم آورد.

## فصل چهارم ارزیابی نتایج

## • مقدمه

در این فصل، به تحلیل و ارزیابی نتایج به دست آمده از آموزش و اجرای مدل‌های یادگیری عمیق می‌پردازیم. هدف از این ارزیابی، بررسی میزان موفقیت مدل در دستیابی به اهداف پروژه و میزان دقت آن در طبقه‌بندی داده‌ها است. پس از اجرای مراحل جمع‌آوری داده‌ها، برچسب‌گذاری و آموزش مدل، ارزیابی نتایج به ما این امکان را می‌دهد تا به دقت عملکرد مدل پی ببریم و نقاط قوت و ضعف آن را شناسایی کنیم.

در این فصل، ابتدا معیارهای اصلی ارزیابی مدل مانند دقت، Precision، Recall و Accuracy شرح داده خواهد شد. سپس، عملکرد مدل در شرایط مختلف و با استفاده از داده‌های آزمون مورد تحلیل قرار می‌گیرد. علاوه بر این، به بررسی تأثیر تکنیک‌های بهبود عملکرد مانند یادگیری انتقالی و افزایش داده‌ها بر نتایج نهایی خواهیم پرداخت. در نهایت، نقاط ضعف مدل بررسی شده و راهکارهای بهبود برای پروژه‌های آینده ارائه خواهند شد.

## • معیارهای ارزیابی مدل (Evaluation Metrics)

### ○ Box Loss

Box loss به میزان خطایی اشاره دارد که مدل در پیش‌بینی دقیق مکان جعبه‌های اطراف (bounding boxes) در یک تصویر مرتکب می‌شود. در یک مدل تشخیص شیء مثل YOLO، مدل باید به درستی مکان دقیق شیء را در تصویر تعیین کند. برای این کار، از جعبه‌های اطراف استفاده می‌شود Box loss. مقایسه این جعبه‌های پیش‌بینی شده با جعبه‌های واقعی (ground truth) می‌پردازد. هرچه جعبه پیش‌بینی شده با جعبه واقعی تطابق بیشتری داشته باشد، این خطا کمتر خواهد بود. [۱۰]

در YOLO، از توابع مختلفی مثل IoU (Intersection over Union) برای محاسبه box loss استفاده می‌شود. هرچه IoU بیشتر باشد، به این معنی است که پیش‌بینی مدل دقیق‌تر است و box loss کمتر خواهد بود. [۱۰]

### ○ Segmentation Loss (Seg Loss)

Segmentation loss مرتبط با وظایف تقسیم‌بندی تصویر (image segmentation) است. در تقسیم‌بندی تصویر، مدل باید هر پیکسل از تصویر را به یکی از کلاس‌های شیء موجود در تصویر اختصاص دهد. این فرآیند پیچیده‌تر از پیش‌بینی bounding boxes است، زیرا نیاز به دقت بالاتری دارد. [۱۰]

در YOLOv<sup>۸</sup> که شامل segmentation نیز می‌شود، segmentation loss نشان‌دهنده میزان خطایی است که مدل در اختصاص دادن پیکسل‌ها به کلاس‌های مختلف مرتکب می‌شود. معمولاً از معیارهایی مانند binary cross-entropy یا IoU برای محاسبه این خطا استفاده می‌شود. [۱۰]

### ○ Classification Loss (Cls Loss)

Cls loss یا classification loss به خطایی اشاره دارد که مدل در تشخیص صحیح کلاس شیء مرتکب می‌شود. پس از شناسایی مکان جعبه‌ها (bounding boxes)، مدل باید کلاس صحیح هر شیء را پیش‌بینی کند (مثلاً ماشین، عابر پیاده، دوچرخه و غیره). Cls loss اختلاف بین برجسب پیش‌بینی شده مدل و برجسب واقعی کلاس شیء را اندازه‌گیری می‌کند. [۱۰]

این loss معمولاً با استفاده از categorical cross-entropy محاسبه می‌شود. هرچه مدل در تشخیص کلاس‌های شیء دقیق‌تر باشد، این loss کمتر خواهد بود.

#### ○ DFL Loss (Distribution Focal Loss)

DFL loss یا Distribution Focal Loss یکی از انواع loss functions است که برای بهبود دقت مدل در مکان‌یابی اشیاء و افزایش تمرکز بر نقاط بحرانی (مانند لبه‌ها) استفاده می‌شود. این نوع loss به مدل کمک می‌کند تا در پیش‌بینی‌های خود بیشتر به نقاطی تمرکز کند که دقت بیشتری نیاز دارند، به خصوص در مرزها و لبه‌های شیء. [۱۰]

این loss معمولاً در مدلی مثل YOLO استفاده می‌شود تا عملکرد مدل را در وظایف دقیق‌تر مانند detection و localization بهبود دهد. DFL loss همچنین نقش مهمی در کاهش box loss دارد و کمک می‌کند که مدل به جای توجه یکسان به همه پیکسل‌ها، روی پیکسل‌های بحرانی تمرکز کند. [۱۰]

#### ○ Precision

Precision یکی از معیارهای مهم برای ارزیابی عملکرد یک مدل طبقه‌بندی است. Precision نشان می‌دهد که از بین تمام پیش‌بینی‌هایی که مدل به عنوان کلاس "مثبت" (مثلاً وجود یک شیء خاص) کرده است، چه درصدی واقعاً درست بوده‌اند.

precision به مدل کمک می‌کند تا بسنجد که چقدر از پیش‌بینی‌های مثبتش دقیق بوده‌اند. اگر مدل درصد بالایی از مثبت‌های کاذب (False Positives) داشته باشد، precision کاهش پیدا می‌کند.

#### ○ Recall

Recall (به یادآوری نیز معروف است) نیز یکی دیگر از معیارهای ارزیابی است و نشان می‌دهد که از بین تمام موارد "مثبت" واقعی (مثلاً تمام مواردی که واقعاً شیء مورد نظر در تصویر وجود دارد)، مدل چقدر توانسته است آن‌ها را به درستی پیش‌بینی کند. [۱۰]

Recall به مدل کمک می‌کند تا بسنجد که چقدر توانسته است تمام موارد مثبت را پیدا کند. اگر مدل درصد بالایی از مثبت‌های واقعی را نادیده بگیرد (False Negatives) زیاد داشته باشد، recall کاهش می‌یابد. [۱۰]



فرمول محاسبه recall به صورت زیر است:

		Predicted	
		0	1
Actual	0	TN	FP
	1	FN	TP

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

شکل ۱۳ معیار های ارزیابی

○ metrics/mAP<sup>۵۰</sup> (M) و metrics/mAP<sup>۵۰</sup> (B) :

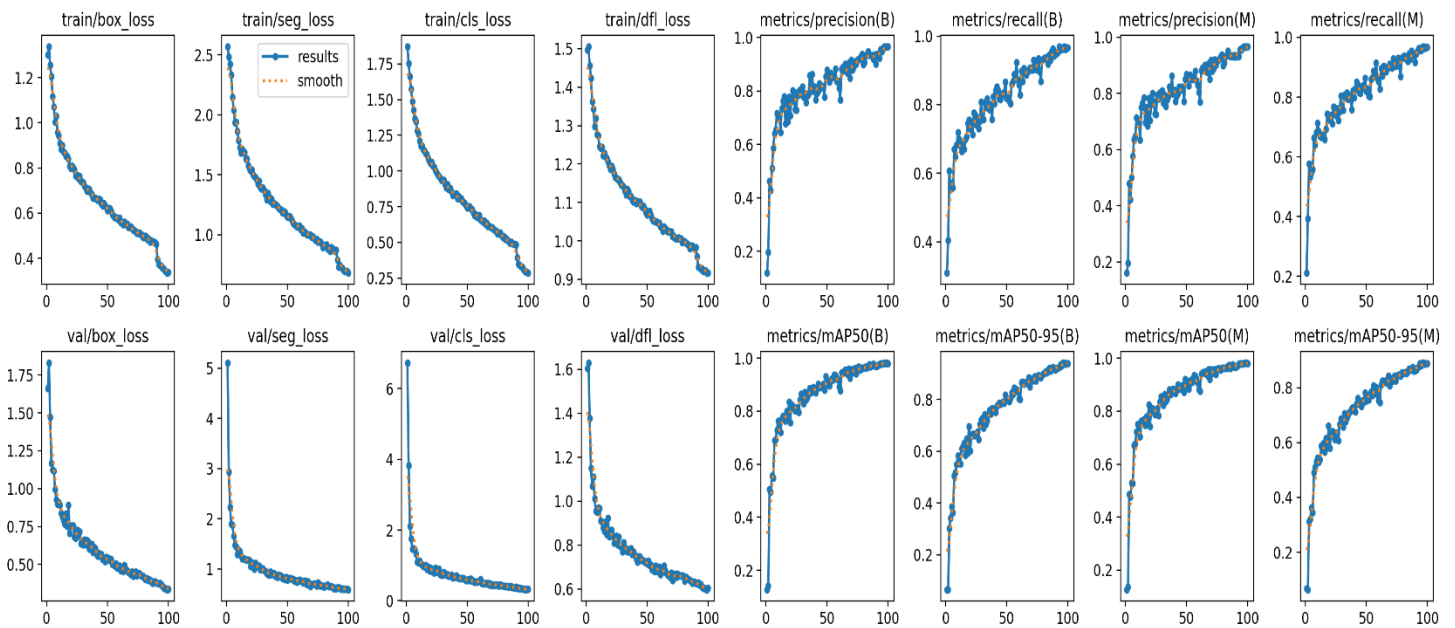
میانگین دقت متوسط (Mean Average Precision) که معیار مهمی در ارزیابی مدل های شناسایی

اشیاست. در اینجا معیار mAP با حداقل دقت ۵۰٪ در نظر گرفته شده است. [۱۰]

○ **metrics/mAP<sub>50-95</sub>(M)** و **metrics/mAP<sub>50-95</sub>(B)** :

این متریک دقت میانگین (mAP) در بازه‌های مختلف (از ۵۰٪ تا ۹۵٪) را نشان می‌دهد و معیاری کلی‌تر از عملکرد مدل به شمار می‌رود. [۱۰]

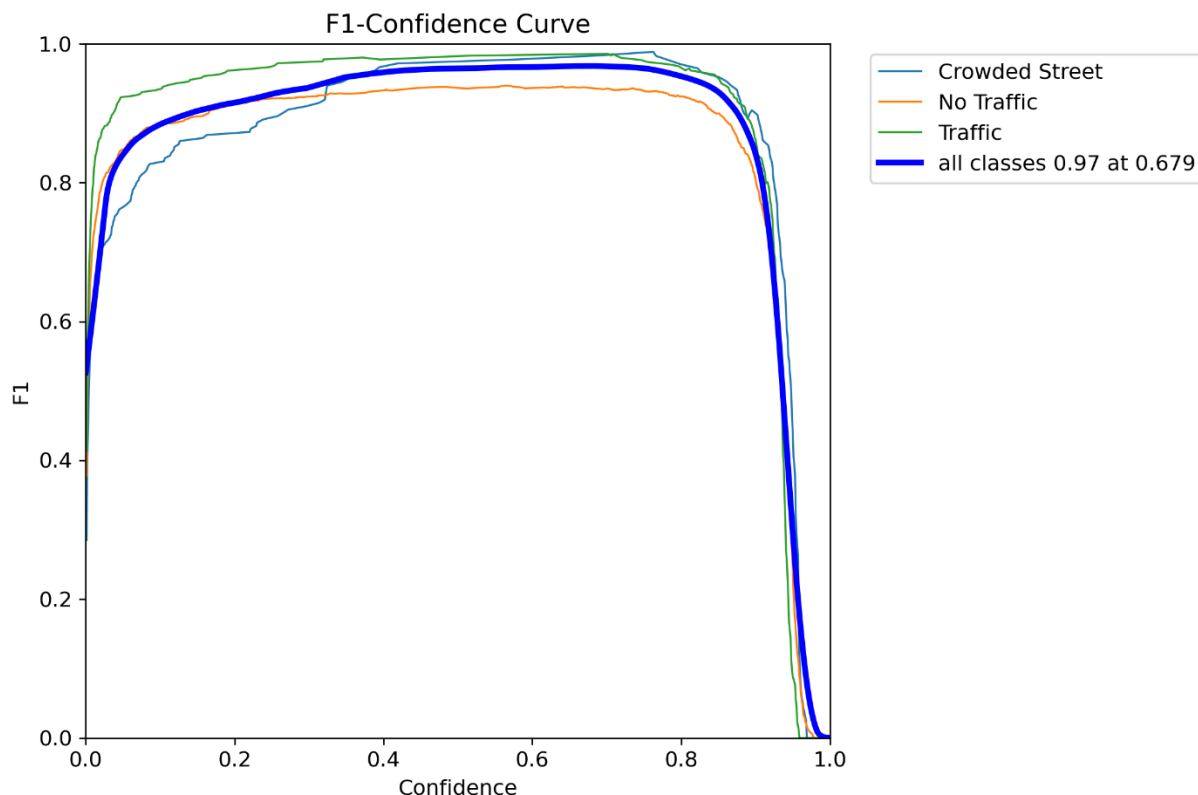
تمامی متریک‌های که تعریف کردیم همه برای ارزیابی مدل ما استفاده می‌شوند مقادیر این المان‌ها در شکل زیر برای مدل ما بعد از آموزش رسم شده است



شکل ۱۴ نتایج کلی مدل

○ **F<sub>1</sub>-confidence curve**

تغییرات F<sub>1</sub> Score را با توجه به آستانه اعتماد (confidence threshold) مدل نشان می‌دهد. این نمودار کمک می‌کند تا بفهمید که با تغییر سطح اعتماد مدل به پیش‌بینی‌هایش، تعادل بین دقت و بازخوانی چگونه تغییر می‌کند. به عبارت دیگر، این نمودار بررسی می‌کند که مدل در کدام سطح اعتماد بهترین عملکرد را از نظر دقت و بازخوانی دارد. [۱۱]



شکل ۱۵ نتیجه f1-curve

#### ○ Precision-confidence curve

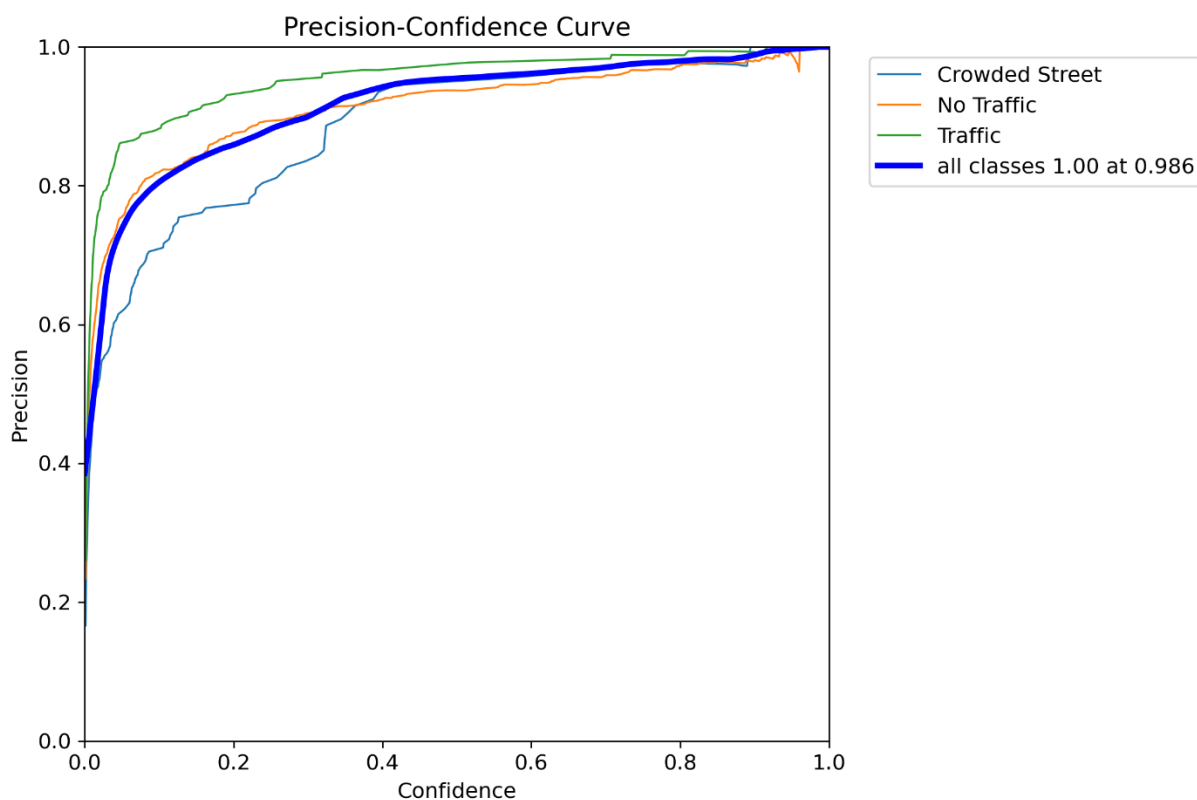
نشان‌دهنده رابطه بین دقت (Precision) و آستانه اعتماد (Confidence Threshold) است. این نمودار نشان می‌دهد که با تغییر آستانه اعتماد مدل، دقت پیش‌بینی‌های آن چگونه تغییر می‌کند. [۱۱]

در مدل‌های تشخیص اشیاء، آستانه اعتماد تعیین می‌کند که مدل چقدر باید به پیش‌بینی خود اطمینان داشته باشد تا آن را به عنوان یک پیش‌بینی مثبت در نظر بگیرد. Precision-confidence curve به تحلیل این موضوع می‌پردازد که با افزایش یا کاهش این آستانه، دقت پیش‌بینی‌های مدل چگونه تحت تأثیر قرار می‌گیرد. به طور کلی، با افزایش آستانه اعتماد: [۱۱]

دقت (Precision) معمولاً افزایش می‌یابد، زیرا مدل تنها پیش‌بینی‌هایی را در نظر می‌گیرد که به آن‌ها اعتماد بالایی دارد. [۱۱]

در مقابل، ممکن است فراخوانی (Recall) کاهش پیدا کند، زیرا مدل تعداد کمتری از پیش‌بینی‌ها را به عنوان مثبت در نظر می‌گیرد. [۱۱]

این نمودار برای پیدا کردن تعادل بین دقت و حساسیت (precision) و (recall) و تنظیم بهترین آستانه اعتماد برای مدل بسیار مفید است.



شکل ۱۶ نتیجه precision curve

#### ○ Confusion Matrix (ماتریس سردرگمی)

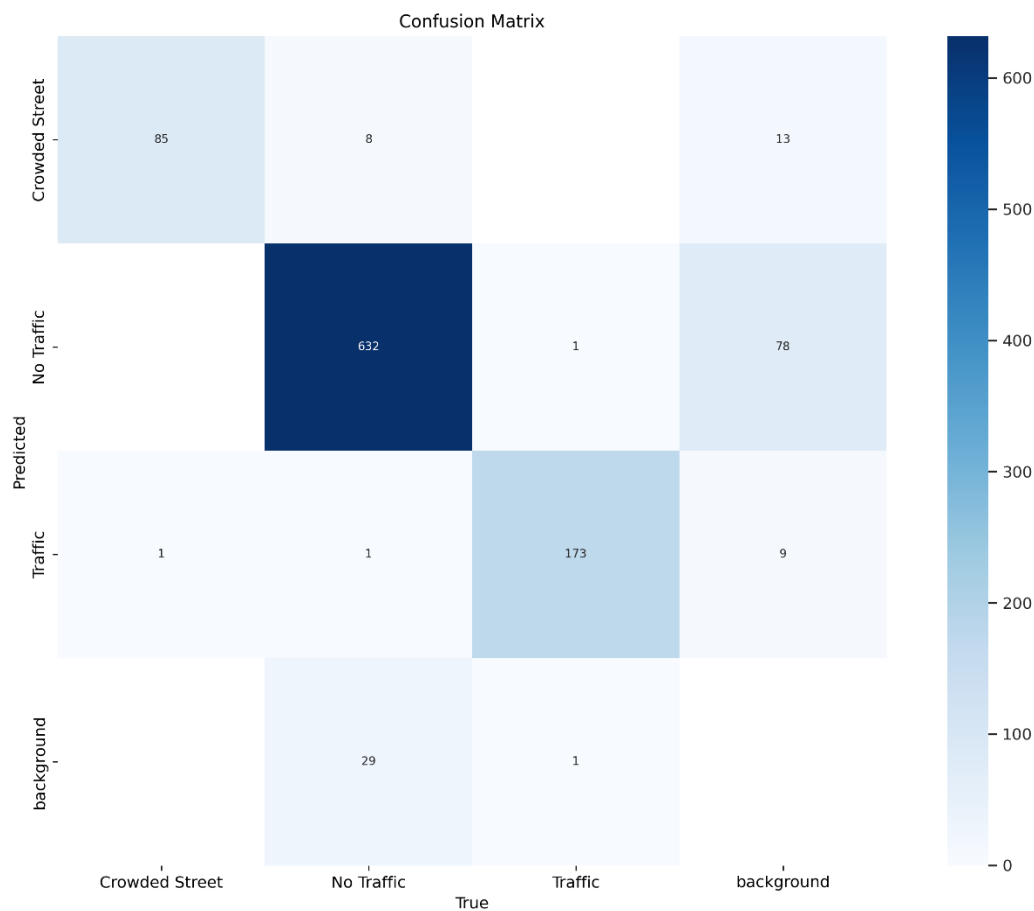
ماتریس سردرگمی یکی از ابزارهای اصلی برای ارزیابی عملکرد مدل‌های طبقه‌بندی است. این ماتریس شامل چهار بخش اصلی است: [۱۱]

❖ **True Positives (TP):** پیش‌بینی‌های صحیح برای کلاس مثبت.

❖ **True Negatives (TN):** پیش‌بینی‌های صحیح برای کلاس منفی.

❖ **False Positives (FP):** پیش‌بینی‌های نادرست برای کلاس مثبت (اشتباهات مثبت).

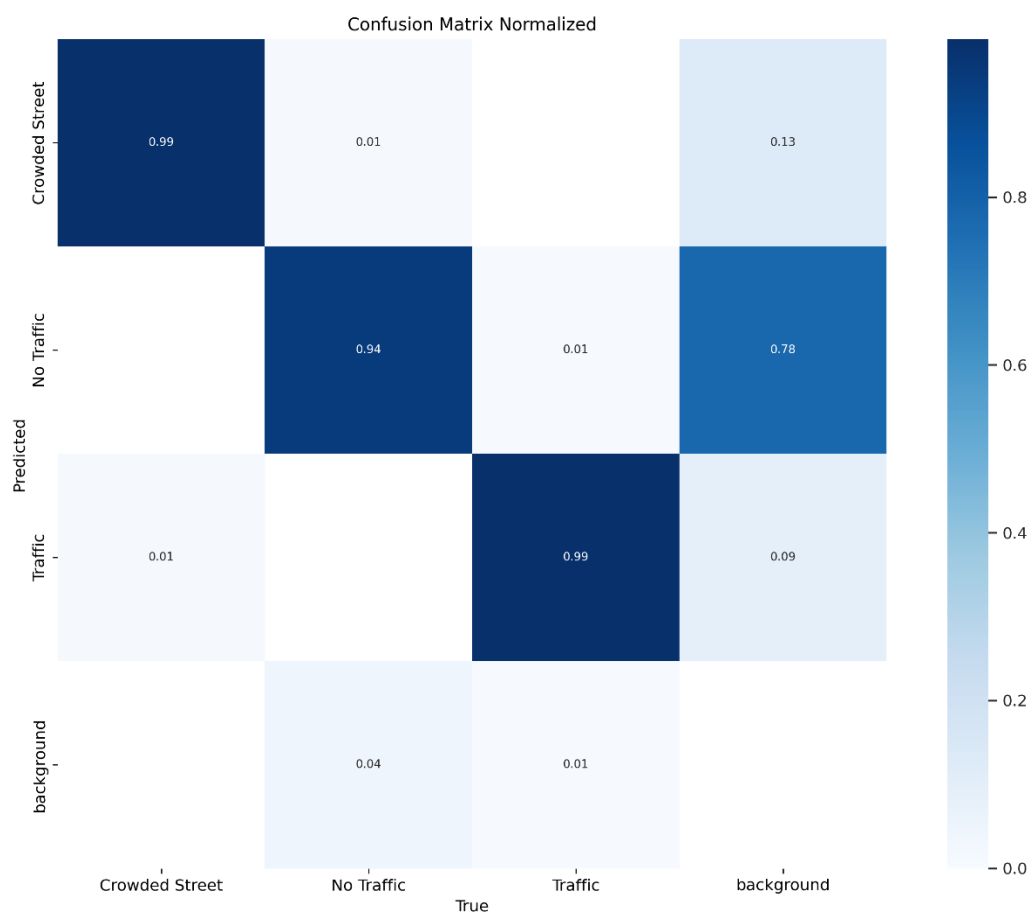
❖ **False Negatives (FN):** پیش‌بینی‌های نادرست برای کلاس منفی (اشتباهات منفی).



شکل ۱۷ Confusion matrix

### ○ Normalized Confusion Matrix (ماتریس سردرگمی نرمال شده)

ماتریس سردرگمی نرمال شده نسخه‌ای است که نتایج را به صورت نسبی و درصدی نمایش می‌دهد. در اینجا مقادیر هر خانه به عنوان درصدی از کل نمونه‌های موجود در آن کلاس بیان می‌شود. [۱۱]



شکل ۱۸ Confusion matrix Normalized

○ تفاوت‌ها:

- ❖ **Confusion Matrix**: تعداد نمونه‌های صحیح و نادرست پیش‌بینی شده را نشان می‌دهد.
- ❖ **Normalized Confusion Matrix**: درصد پیش‌بینی‌های صحیح و نادرست نسبت به کل نمونه‌های هر کلاس را نمایش می‌دهد.

○ کاربرد:

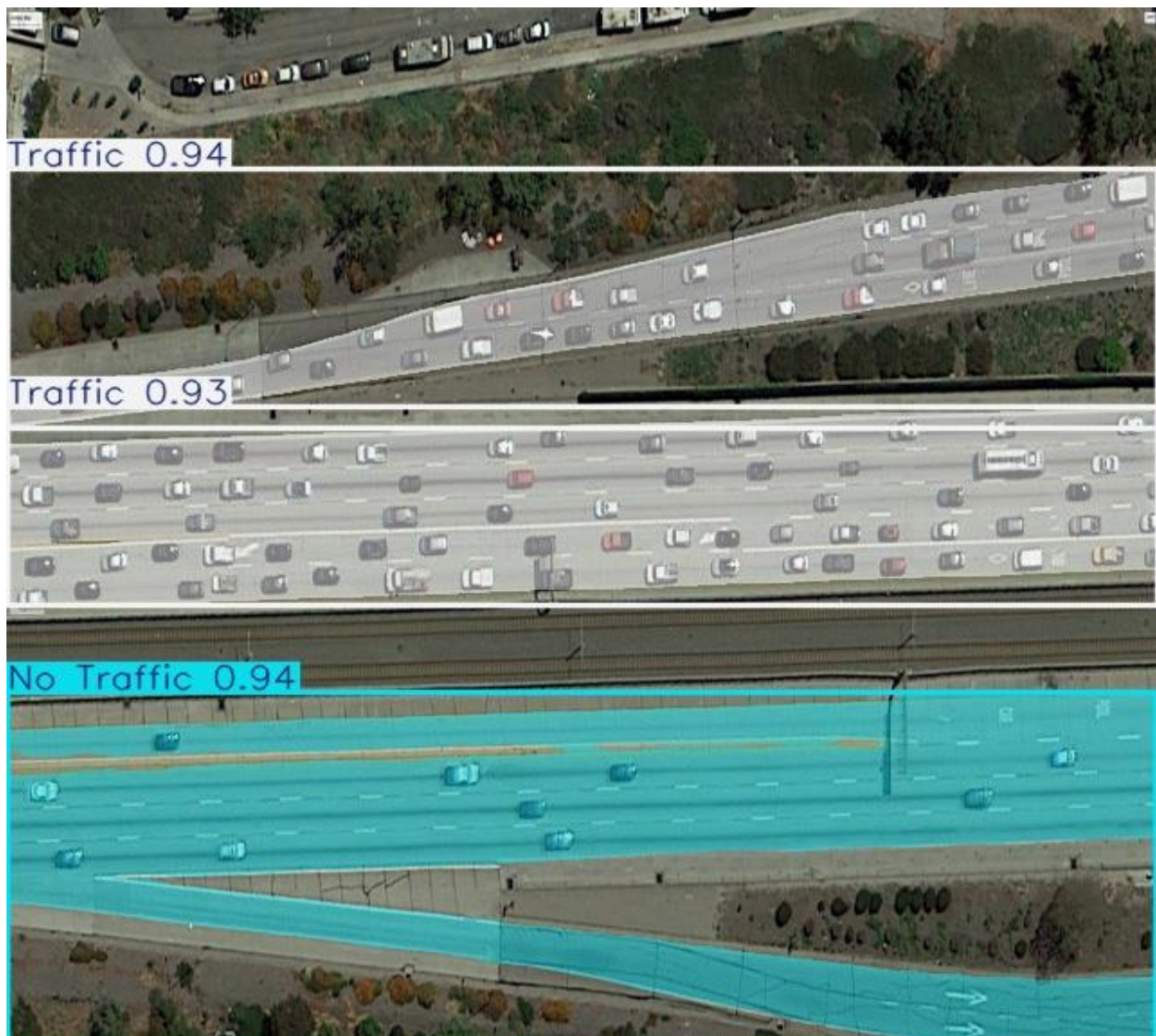
❖ **Confusion Matrix**: برای ارزیابی کلی عملکرد مدل مفید است.

❖ **Normalized Confusion Matrix**: برای بررسی عملکرد مدل روی داده‌های نامتوازن کاربرد

دارد.

• نمونه های پیش بینی شده

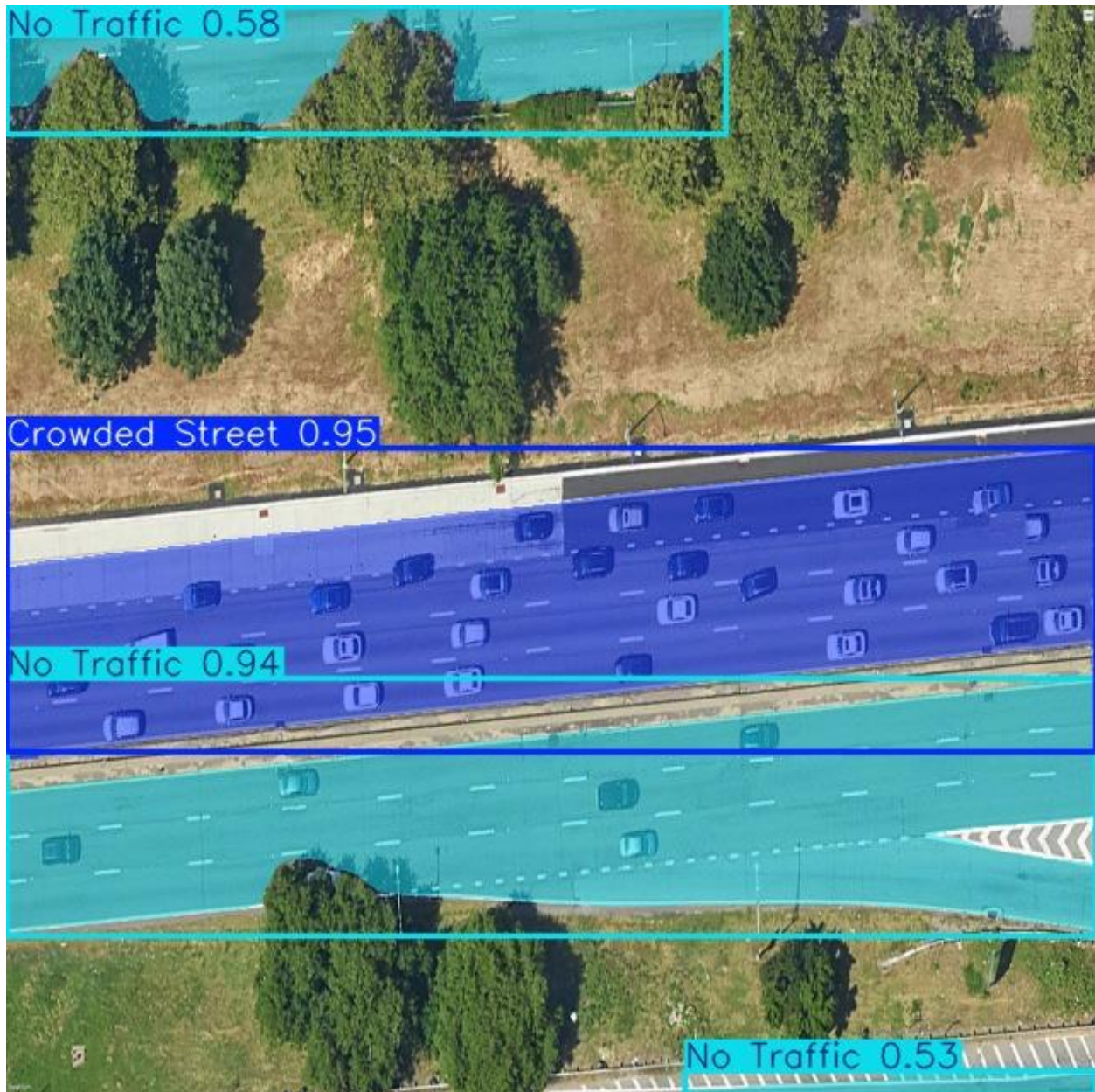
○ تصویر شماره ۱



شکل ۱۹ مثال شماره ۱



○ تصویر شماره ۲



شکل ۲۰ مثال شماره ۲



○ تصویر شماره ۳



شکل ۲۱ مثال شماره ۳

## • نتیجه گیری

در این پروژه، با استفاده از مدل‌های یادگیری عمیق، به بررسی و ارزیابی میزان دقت و کارایی مدل در تشخیص و طبقه‌بندی داده‌ها پرداخته شد. نتایج به دست آمده نشان‌دهنده دقت و عملکرد مطلوب مدل در بسیاری از شرایط بودند. استفاده از معیارهای ارزیابی مانند دقت (Precision)، یادآوری (Recall)، و میانگین دقت متوسط (mAP) به ما کمک کرد تا نقاط قوت مدل در تشخیص اشیاء را به خوبی ارزیابی کنیم. همچنین، به کارگیری تکنیک‌های بهبود عملکرد مانند یادگیری انتقالی و افزایش داده‌ها، تأثیر مثبتی بر دقت نهایی مدل داشت.

با این حال، نقاط ضعفی نیز شناسایی شد، از جمله کاهش عملکرد مدل در داده‌های با توزیع نامتعادل. بررسی ماتریس سردرگمی نشان داد که مدل در برخی موارد دچار خطاهای پیش‌بینی است که می‌تواند با بهبود روش‌های آموزش و بهینه‌سازی مدل کاهش یابد. استفاده از تکنیک‌های پیشرفته‌تر مانند DFL Loss نیز می‌تواند به بهبود دقت در مکان‌یابی اشیاء کمک کند.

در نهایت، نتایج این تحقیق نشان می‌دهد که مدل مورد استفاده پتانسیل بالایی برای کاربردهای عملی دارد، اما برای دستیابی به عملکرد بهتر در پروژه‌های آینده، بهبودهایی در بخش‌های مختلف آن مورد نیاز است.

## **فصل پنجم جمع بندی و نتیجه گیری**

## • مقدمه

پژوهش حاضر با هدف طراحی و پیاده‌سازی سیستمی برای تشخیص وضعیت ترافیک شهری از طریق پردازش تصاویر ماهواره‌ای و استفاده از الگوریتم‌های یادگیری عمیق انجام شد. با توجه به رشد سریع شهرنشینی و افزایش تعداد خودروها در سطح شهرها، مدیریت کارآمد ترافیک یکی از چالش‌های اساسی در جوامع شهری مدرن است. استفاده از تصاویر ماهواره‌ای با وضوح بالا در کنار شبکه‌های عصبی عمیق، پتانسیل بالایی برای بهبود این مدیریت فراهم کرده است. این فصل به بررسی کلی نتایج، تحلیل دستاوردهای اصلی، و ارائه پیشنهادات برای بهبود سیستم‌ها و تحقیقات آتی خواهد پرداخت.

## • مروری بر روش‌ها و تکنولوژی‌ها

در این پروژه از مدل YOLO (You Only Look Once) به عنوان مدل اصلی تشخیص اشیاء استفاده شد که به دلیل معماری ویژه‌اش، امکان پردازش سریع تصاویر و تشخیص بلادرنگ اشیاء را فراهم می‌کند. YOLO یکی از پیشرفته‌ترین مدل‌های یادگیری عمیق در زمینه تشخیص اشیاء است که توانایی شناسایی خودروها و وضعیت ترافیک را با دقت بالا دارد.

شبکه‌های عصبی کانولوشنی (CNN) در این پروژه برای شناسایی ویژگی‌های کلیدی تصاویر به کار گرفته شدند. یکی از مزیت‌های بزرگ این شبکه‌ها توانایی‌شان در تشخیص الگوهای پیچیده بصری است. در کنار آن، استفاده از یادگیری انتقالی (Transfer Learning) باعث شد تا مدل بتواند از دانش مدل‌های از پیش آموزش دیده‌شده استفاده کند و نیاز به آموزش از صفر کاهش یابد. این امر نه تنها سرعت آموزش مدل را افزایش داد، بلکه دقت نهایی سیستم را نیز بهبود بخشید.

## • نتایج و ارزیابی‌ها

مدل YOLO در طول این پژوهش توانست وضعیت ترافیک را به سه دسته اصلی یعنی ترافیک سنگین، متوسط و بدون ترافیک طبقه‌بندی کند. نتایج دقت (Accuracy) و بازخوانی (Recall) نشان داد که مدل با عملکرد بالایی در تشخیص ترافیک عمل می‌کند. در شرایط نوری مناسب و تصاویر با وضوح بالا، مدل توانست به خوبی اشیاء و وضعیت ترافیکی را شناسایی کند.

در ارزیابی‌های انجام شده، دقت پیش‌بینی (Precision) مدل نیز به طور قابل توجهی بالا بود که نشان‌دهنده قابلیت خوب آن در پیش‌بینی وضعیت‌های مختلف ترافیکی با خطای کم است. این نتایج نشان داد که این روش می‌تواند به عنوان یک ابزار کارآمد برای بهبود مدیریت ترافیک در شهرهای بزرگ و پرتراکم استفاده شود. از سوی دیگر، چالش‌هایی همچون وجود سایه‌ها یا اشیاء بزرگ‌تر در تصاویر ماهواره‌ای، بر عملکرد مدل تأثیر گذاشت و در برخی موارد دقت تشخیص کاهش یافت.

## • تحلیل چالش‌ها و نقاط ضعف

اگرچه مدل به کار رفته توانایی بالایی در تشخیص وضعیت ترافیک داشت، اما برخی چالش‌ها مشاهده شد که عملکرد مدل را محدود کرد. یکی از اصلی‌ترین چالش‌ها تفاوت‌های محیطی و بصری تصاویر ماهواره‌ای بود. تغییرات نورپردازی، وجود سایه‌ها، یا اشیاء بزرگ‌تر در تصاویر گاهی اوقات باعث شد مدل نتواند به درستی ترافیک را تشخیص دهد. برای مثال، در تصاویر ماهواره‌ای که شامل سایه‌های بزرگ یا موانع بصری بودند، عملکرد مدل در شناسایی دقیق وضعیت ترافیکی افت کرد.

همچنین، در برخی موارد محدودیت‌هایی در کیفیت تصاویر یا تعداد داده‌های ورودی وجود داشت. اگرچه تصاویر ماهواره‌ای با وضوح بالا انتخاب شده بودند، اما به دلیل تنوع بالای شرایط محیطی در تصاویر مختلف، مدل در برخی موارد با خطاهایی مواجه شد.

## • جمع‌بندی

به طور کلی، این پژوهش نشان داد که استفاده از یادگیری عمیق و تکنیک‌های پردازش تصویر می‌تواند به طور مؤثری در مدیریت ترافیک شهری به کار گرفته شود. مدل YOLO با بهره‌گیری از تصاویر ماهواره‌ای و تکنیک‌های یادگیری انتقالی، توانست وضعیت ترافیکی را با دقت و سرعت بالایی شناسایی کند. با این حال، چالش‌هایی همچون کیفیت تصاویر و شرایط محیطی متنوع نیاز به بهبود در سیستم را نشان داد.

## • نتیجه‌گیری و پیشنهادات برای کارهای آینده

استفاده از تصاویر ماهواره‌ای و یادگیری عمیق در تحلیل ترافیک شهری، گام بزرگی در جهت بهبود مدیریت شهری است. مدل پیشنهادی این پژوهش توانست با دقت قابل قبولی وضعیت ترافیک را در سطح شهر شناسایی کند. با این حال، برای بهبود سیستم در آینده، پیشنهاد می‌شود از تصاویر متنوع‌تری در شرایط مختلف نوری و محیطی استفاده شود تا مدل بتواند در تمامی شرایط عملکرد بهینه داشته باشد.

همچنین، توسعه سیستم با استفاده از مدل‌های پیشرفته‌تر و بهره‌گیری از داده‌های بیشتر می‌تواند به بهبود نتایج کمک کند. ترکیب تصاویر ماهواره‌ای با دیگر منابع داده مانند حسگرهای ترافیکی یا دوربین‌های شهری می‌تواند دقت و کارایی سیستم را به‌طور چشمگیری افزایش دهد و زمینه را برای مدیریت هوشمندتر ترافیک شهری فراهم آورد.

- [١] Y. B. a. M. Özuysal , "Introduction to Machine Learning " , p. ٢٠١٤ ، ٢٤
- [٢] V. Nilsen , "Artificial-Neural-Networks-for-Neuroscientists-A-Primer " , p. ٢٠١٨ ، ١٣
- [٣] M. M. Taye , "Understanding of Machine Learning with Deep Learning: Architectures, Workflow Applications and Future Directions . ٢٠٢٣ آوریل ٢٥ " ،
- [٤] D. Controls , " ٦ Reasons Why Traffic Control Systems Matter . "
- [٥] P. Patil , "Applications of Deep Learning in Traffic Management: A Review. International Journal of Business Intelligence and Big Data Analytics . ٢٠٢٢ ژانویه ٢٠ " ،
- [٦] Y. T. I. G. A. D. A. COMPREHENSIVE , "Ranjan Sapkota ١\* , Rizwan Qureshi ٢ , Marco Flores-Calero ٣ , Chetan Badgujar ٤ , Upesh Nepal ٥ , Alwin Poulouse ٦ , Peter Zeno ٧ , Uday Bhanu Prakash Vaddevolu ٨ , Sheheryar Khan ٩ , Maged Shoman ١٠ , Hong Yan ١١ , and Manoj Karkee " ، ١ p. ٢٦ ، ٤٩ July . ٢٠٢٤
- [٧] H. R. A. K. R. Mohammadreza Iman , "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv ١ to YOLOv ٨ and YOLO-NAS . ٢٠٢٣ " ،
- [٨] H. R. A. K. R. Mohammadreza Iman , "A Review of Deep Transfer Learning and Recent Advancements . ٢٠٢٣ " ،
- [٩] F. K. S. R. Chen K , "Performance improvement of Deep Learning Models using image augmentation techniques . ٢٠٢١ " ،
- [١٠] J. Smith , "YOLO Performance Metrics and Their Importance in Object Detection " ، September . ٢٠٢٣
- [١١] Ultralytics , "YOLO Performance Metrics . ٢٠٢٣ " ،
- [١٢] A. S. A. B. D. C. G. a. S. Beevers , "City Scale Traffic Monitoring Using WorldView Satellite " ، p. ٢٠٢٣ ، ١٨
- [١٣] C. F. Higham , "Deep Learning: An Introduction for Applied Mathematicians " ، p. ٣٢ ، ٢٠١٩
- [١٤] M. B. ١ . M. M. ١ . D. Borth , "Commercial Vehicle Traffic Detection from Satellite Imagery with Deep " ، p. ٢٠٢١ ، ٦



[١٥] J. Brownlee , "A Gentle Introduction to Transfer Learning for Deep Learning ."

[١٦] V. researchers , "CurveFormer++: ٣D Lane Detection by Curve Propagation with Temporal Curve Queries and Attention " , septamer . ٢٠٢٣