

به نام خدا



دانشگاه تهران



دانشکده مهندسی برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین سوم

نام و نام خانوادگی	آرمان مجیدی	پرسش ۱
شماره دانشجویی	810100205	
نام و نام خانوادگی	آرین فیروزی	پرسش ۲
شماره دانشجویی	810100196	
مهلت ارسال پاسخ	۱۴۰۱.۰۸.۲۹	

فهرست

پرسش ۱. سگمنتیشن تومور مغزی از روی تصاویر MRI 1

1-۱. توصیف مدل ارائه شده 1

۲-۱. آماده سازی مجموعه داده 1

۳-۱. داده افزایی 3

4-1. بهینه ساز، معیارها و تابع هزینه 5

5-1. پیاده سازی مدل 5

6-1. آموزش مدل 6

7-1. ارزیابی مدل 7

پرسش ۲. تشخیص تابلو های رهنمایی و رانندگی 10

۱-۱. توصیف کلی 10

1-2. آماده سازی مجموعه داده ها 10

2-2. تنظیم دقیق مدل دو مرحله ای 12

2-3. تنظیم دقیق مدل تک مرحله ای 15

2-4. مقایسه مدل ها 18

شکل‌ها

پرسش 1: سگمنتیشن تومور مغزی از روی تصاویر MRI

1. شکل 1.1: عکس‌های دیتاست به همراه ماسک‌های مربوطه (تقویت نشده)
2. شکل 1.2: عکس‌های دیتاست به همراه ماسک‌های مربوطه (تقویت شده)
3. شکل 1.3: ساختار مدل پیشنهادی مقاله
4. شکل 1.4: نمودارهای دقت (Accuracy)، ضریب Dice، و IoU و Loss
5. شکل 1.5: نمونه‌ای از تصاویر ارزیابی همراه با ماسک‌های حقیقی و پیش‌بینی شده

پرسش 2: تشخیص تابلوهای راهنمایی و رانندگی

6. شکل 2.1: نمونه‌ای از تابلوهای annotate شده
7. شکل 2.2: هیستوگرام داده‌ها بر حسب کلاس
8. شکل 2.3: هیستوگرام داده‌ها بر حسب اندازه
9. شکل 2.4: هیستوگرام داده‌های آموزشی بر حسب کلاس
10. شکل 2.5: هیستوگرام داده‌های آموزشی بر حسب اندازه
11. شکل 2.6: هیستوگرام داده‌های آزمون بر حسب کلاس
12. شکل 2.7: هیستوگرام داده‌های آزمون بر حسب اندازه
13. شکل 2.8: نمودار AP بر حسب IoU برای مدل دو مرحله‌ای
14. شکل 2.9: نمودار AP بر حسب کلاس برای مدل دو مرحله‌ای
15. شکل 2.10: نمودار AP بر حسب اندازه برای مدل دو مرحله‌ای
16. شکل 2.11: تصاویر پیش‌بینی شده (قرمز) و واقعی (سبز) توسط مدل دو مرحله‌ای
17. شکل 2.12: نمودار AP بر حسب IoU برای مدل تک مرحله‌ای
18. شکل 2.13: نمودار AP بر حسب کلاس برای مدل تک مرحله‌ای
19. شکل 2.14: نمودار AP بر حسب اندازه برای مدل تک مرحله‌ای
20. شکل 2.15: تصاویر پیش‌بینی شده (قرمز) و واقعی (سبز) توسط مدل تک مرحله‌ای

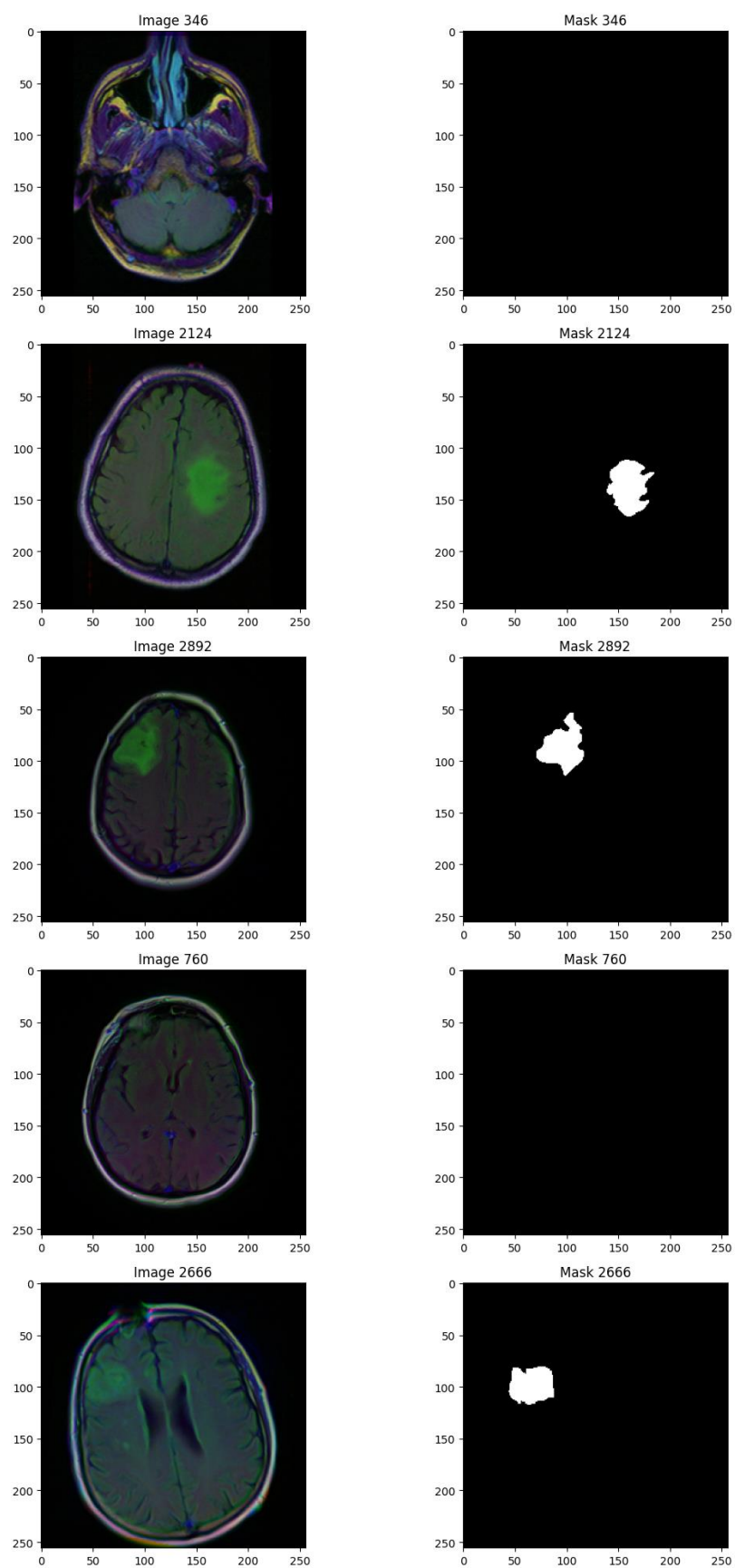
پرسش ۱. سگمنتیشن تومور مغزی از روی تصاویر MRI

۱-۱. توصیف مدل ارائه شده

در مقاله آپلود شده هدف ما سگمنتیشن تومور مغزی با استفاده از تصاویر MRI و ماسک‌های تومور این تصاویر می‌باشد. شبکه استفاده از این مقاله، ترکیب U-Net و VGG16 می‌باشد. برای شبکه پیشنهادی می‌توان فرض کرد که یک قسمت contracting و یک قسمت expanding داریم. قسمت contracting شبکه همان VGG16 است که وزن‌های آن به وسیله آموزش توسط دیتاست imagenet مقداردهی شده‌اند. ما به دنبال پیاده‌سازی Transfer Learning هستیم، در نتیجه مشابه مقاله، وزن‌های قسمت VGG16 را فریز می‌کنیم. همچنین، چون مدل ما باید به سبک U-Net باشد، نیاز به یکسری Residual از لایه‌های contracting(VGG16) داریم. استفاده از Transfer Learning از این جهت کاربرد دارد که قسمت contracting مدل ما با یک دیتاست عظیم آموزش دیده‌است و توانایی جداسازی feature های مناسب را دارد.

۱-۲. آماده‌سازی مجموعه داده

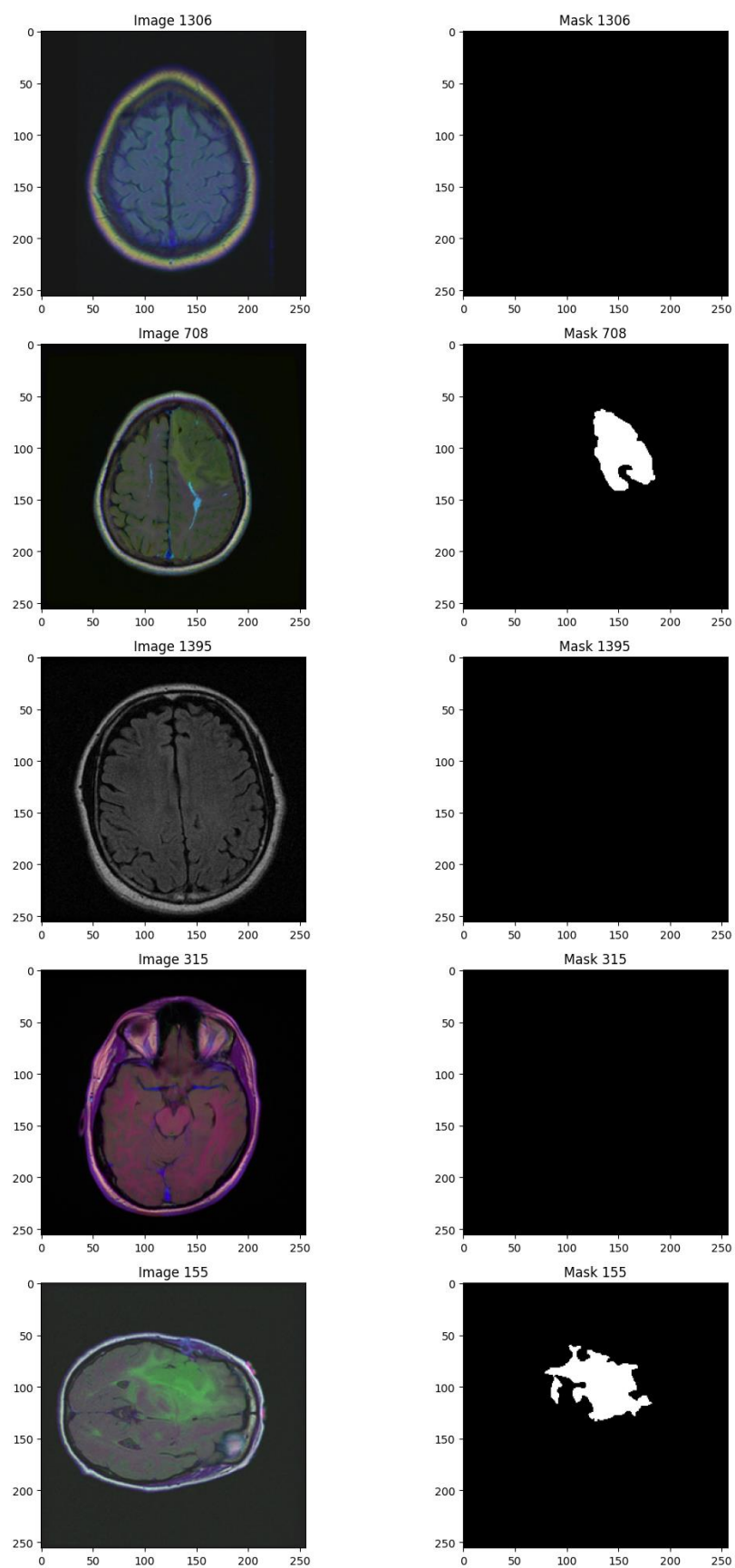
برای این تمرین، به علت کندی فراوان cpu برای آموزش داده، از Kaggle استفاده کردم. به دلیل این که دیتاست داده شده خود در Kaggle می‌باشد، دیتاست را به عنوان input به notebook می‌دهیم. در نتیجه دیتاست ما در مسیر "kaggle/input/lgg-mri-segmentation/kaggle_3m/" خواهد بود. حال، با استفاده از تابع تعریف شده dataset_loader() عکس‌های خود را به سه دسته validation, training و test تقسیم می‌کنیم. این تابع، با استفاده از کتابخانه tifffile تمام عکس‌های با فرمت tif و tiff را لود می‌کند. سپس این عکس‌ها را به آرایه numpy تبدیل می‌کنیم. سپس برای صحت‌سنجی، سائز داده‌های train, validation و test را بررسی می‌کنیم. برای train تعداد 3143 عکس و برای validation و test تعداد 393 عکس داریم. سپس 5 عکس از دیتاست خود به همراه ماسک آن‌ها را نشان می‌دهیم. این تصویر، در شکل 1.1 نشان داده شده‌است.



شکل 5.1.1 عکس از دیتاست به همراه ماسک‌هایشان (تقویت نشده)

۱-۳. داده افزایی

برای تقویت داده، مطابق فایل تمرین از کتابخانه Albumentations استفاده می کنیم. سپس، با استفاده از تکنیک های تقویت داده مانند چرخش تصادفی 90 درجه ای عکس، آینه ای کردن عکس، تغییر تصادفی روشنایی و کنتراست عکس و تغییر رنگ و اشباع عکس، اقدام به augment کردن عکس ها می کنیم. شکل 1.2، مثالی از تصاویر تقویت شده را نشان می دهد.



شکل 5.1.2 عکس از دیتاست به همراه ماسک‌هایشان (تقویت شده)

1-4. بهینه‌ساز، معیارها و تابع هزینه

دو معیار IoU (Intersection over Union) و Dice Coefficient از پرکاربردترین معیارهای ارزیابی در مسئله‌های Image Segmentation می‌باشند. این دو معیار شباهت بین ناحیه پیش‌بینی شده و ناحیه واقعی را بررسی می‌کنند.

معیار IoU، نسبت اشتراک دو مجموعه نسبت به اتحاد آن‌ها را نشان می‌دهد.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \rightarrow \begin{cases} A: \text{Predicted Region} \\ B: \text{True Region} \end{cases}$$

مقدار IoU همیشه میان 0 و 1 می‌باشد. تمایل داریم تا مقدار IoU به یک میل کند. این معیار، معمولاً برای مسائل Image Segmentation استفاده می‌شود.

معیار Dice Coefficient، شباهت دو مجموعه را با دو برابر اشتراک آن‌ها نسبت به جمع اندازه‌هایشان ارزیابی می‌کند.

$$Dice\ Coeff = \frac{|A \cap B|^2}{|A| + |B|} \rightarrow \begin{cases} A: \text{Predicted Region} \\ B: \text{True Region} \end{cases}$$

مقدار Dice Coefficient همیشه میان 0 و 1 می‌باشد. تمایل داریم تا این معیار نیز به یک میل کند. این معیار، به علت وجود توان 2 در صورت، حساسیت بیشتری به ناحیه‌های کوچک دارد؛ در نتیجه در مسائل Medical Image Segmentation استفاده بیشتری دارد.

رابطه میان IoU و Dice Coefficient به صورت زیر می‌باشد:

$$Dice\ Coeff = \frac{2 \times IoU}{IoU + 1}$$

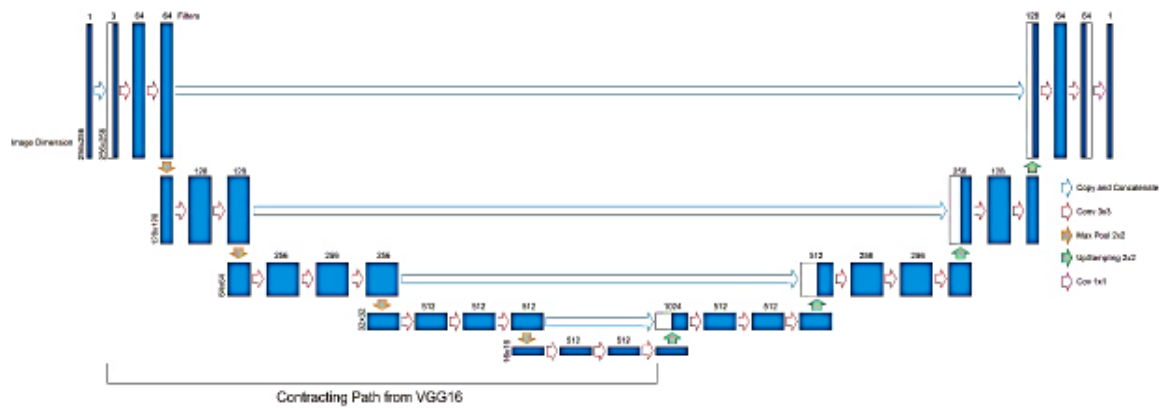
تابع هزینه Dice Loss نیز رابطه زیر را با Dice Coefficient دارد:

$$Dice\ Loss = 1 - Dice\ Coeff$$

برای این سوال، من تعداد اپاک‌ها را برابر با 20، batch size را برابر 32، بهینه‌ساز را Adam با نرخ یادگیری 0.0001، تابع هزینه را برابر با dice_loss و معیارها را برابر با iou_metric، dice_coefficient و accuracy قرار دادم.

1-5. پیاده‌سازی مدل

مدل پیشنهادی توسط مقاله در شکل 1.3 آمده‌است.



شکل 1.3. مدل پیشنهادی مقاله

این مدل در تابع `unet_vgg16.py` پیاده‌سازی شده‌است. نحوه پیاده‌سازی آن نیز در بخش 1-1 توضیح داده‌شد.

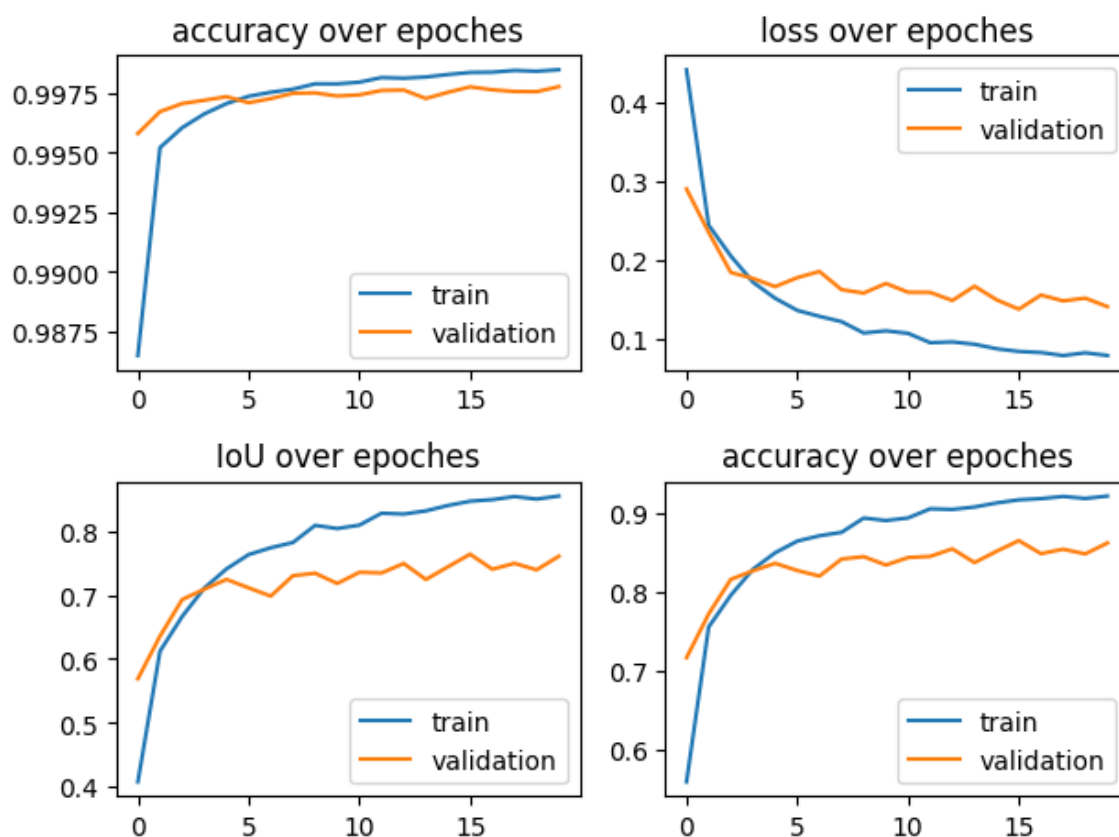
6-1. آموزش مدل

تعداد epoch و batch size در قسمت 1-4 بیان شد. پس از آموزش مدل به میزان 20 ایپاک، به پارامترهای زیر رسیدیم:

$$\begin{cases} accuracy: 0.9985 & val\ accuracy: 0.9977 \\ dice\ coeff: 0.9218 & val\ dice\ coeff: 0.8620 \\ iou\ metric: 0.8558 & val\ iou\ metric: 0.7609 \\ loss: 0.0782 & val\ loss: 0.1403 \end{cases}$$

در نتیجه، یادگیری خوب بوده‌است.

نمودارهای `accuracy`، `loss`، `IoU` و `Dice Coefficient` در شکل 1.4 آمده‌اند. به ترتیب، در بالا سمت چپ نمودار `accuracy`، در بالا سمت راست نمودار `loss`، در پایین سمت چپ نمودار `IoU` و در پایین سمت راست نمودار `Dice Coefficient` حضور دارند.

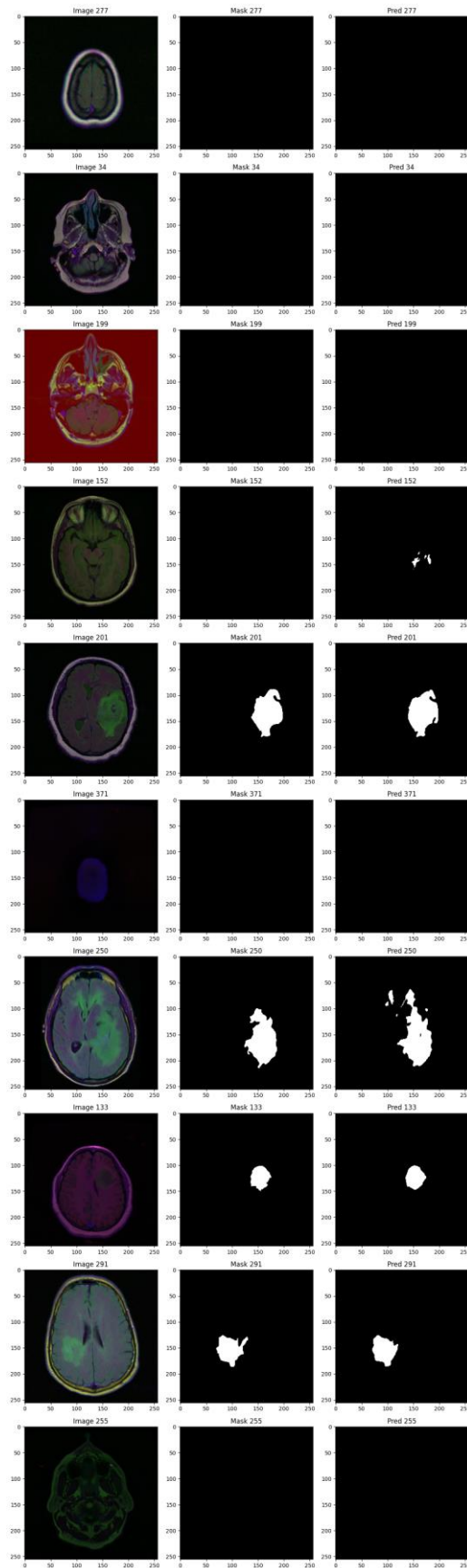


شکل 1.4. نمودارهای **Dice Coefficient** و **IoU loss accuracy**

همان‌طور که مشاهده می‌شود، **IoU** و **Dice Coefficient** ما به مقدار قابل توجهی پیشرفت کرده‌است و میزان آن به صورت تقریبی برابر با 0.7 می‌باشد. این نشان دهنده این است که مدل به خوبی قسمت **Segment** را تشخیص می‌دهد و همچنین اشتراک پیش‌بینی ما با ماسک واقعی، تقریباً 70% اجتماع آن هاست.

1-7. ارزیابی مدل

این 10 نمونه در شکل 1.5 نشان داده شده‌اند.



شکل 10.1.5. نمونه از تصاویر ارزیابی همراه با ماسک حقیقی و ماسک پیش‌بینی

باتوجه به شکل 1.5، می‌توان عملکرد مورد قبول مدل را تایید کرد. همچنین، باتوجه به عکس‌هایی که ماسک واقعی آن‌ها تومور ندارد می‌توان گفت که عملکرد مدل در مورد این دسته از عکس‌ها مورد قبول است و از 6 عکس با این ویژگی، تنها یکی از آن‌ها به اشتباه ماسک با تومور بدست آمده‌است. در نتیجه اگر این 10 عکس را به عنوان جامعه هدف فرض کنیم، از 6 نفری که تومور ندارند تنها 1 نفر به اشتباه دارای تومور شناسایی می‌شود.

از 4 عکس باقی‌مانده که تمام آن‌ها تومور دارند، تمام آن‌ها شناسایی می‌شوند. البته ماسک پیش‌بینی شده تفاوت‌هایی با ماسک اصلی دارد، اما این تفاوت زیاد نیست. همچنین، با بررسی عکس هفتم و ماسک واقعی و ماسک پیش‌بینی آن، می‌توان ادعا کرد که مدل ما به ناحیه رنگی تصاویر MRI حساسیت خوبی دارد، در نتیجه قسمت زیادی از ناحیه بالایی مغز (که در MRI سبز شده‌است) را نیز به عنوان تومور تشخیص می‌دهد. در نتیجه اگر این 10 عکس را به عنوان جامعه هدف فرض کنیم، از 4 نفری که تومور ندارند هیچ فردی به اشتباه بدون تومور شناسایی نمی‌شود و تنها در محل و مقدار تومور تفاوت‌هایی وجود دارد. پیش‌بینی می‌شود با آموزش مدل به تعداد ایپاک‌های بیشتر، این تفاوت‌ها به مرور کمتر و کمتر شود.

پرسش ۲. تشخیص تابلو های راهنمایی و رانندگی

۱-۱. توصیف کلی

در این سوال، هدف پیاده سازی و finetune کردن دو مدل برای تشخیص تابلو های راهنمایی و رانندگی بود. به این منظور ابتدا از کتابخانه ی tensorflow برای خواندن دیتاست استفاده شد، ولی به علت پیچیدگی این کتابخانه در پیاده سازی مدل های این پرسش، تصمیم گرفتیم از pytorch استفاده کنیم. به همین علت، بخش 1-2 در این داکيومنت از tf و سایر بخش ها از torch استفاده میکنند. کدهای این تمرین در مرحله ترین با استفاده از colab و در سایر مراحل به صورت local ران شده اند و inconsistency ها مربوط به نحوه اجرا از این مورد نشأت میگیرد.

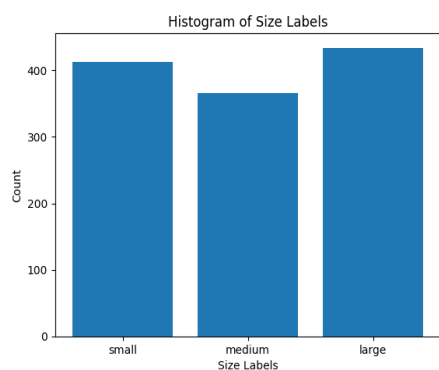
1-2. آماده سازی مجموعه داده ها

برای این بخش، دادگان GTSDb دانلود و در فولدر data قرار داده شدند. این مجموعه داده شامل حدود 900 عکس با بیش از 1200 تابلو راهنمایی رانندگی و annotation های آن است. تابلو ها در این مجموعه به 42 کلاس تقسیم شده اند ولی با توجه به صورت سوال، ما آن را به 4 دسته تقسیم کرده و کار را همان کلاس ها پیش میبریم. این مجموعه 2 بار توسط کتابخانه هایی که در بالا ذکر شد خوانده شده که در ادامه به نحوه ی اجرای آن خواهیم پرداخت. نمونه ای از تابلو های annotate شده در مجموعه دادگان در تصویر 2.1 قابل مشاهده است.

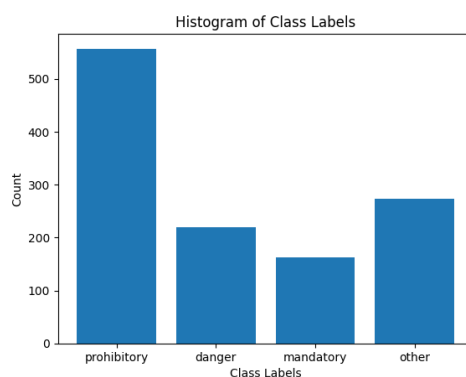


شکل 2.1 نمونه تابلو annotate شده

در گام بعدی، خواسته شده که مجموعه دادگان با توجه با اندازه اشیا به سه دسته تقسیم شود. با پیروی از الگوی معرفی شده در مقاله، اشیا با عرض کمتر از 32 به کلاس کوچک، بین 32 و 45 به کلاس متوسط و سایر به کلاس بزرگ تعلق دارند. هیستوگرام مربوط به این تقسیم بندی و تقسیم بندی کلاس 4 تایی که قبلا معرفی شد، به ترتیب در تصاویر 2.2 و 2.3 قابل مشاهده میباشد.

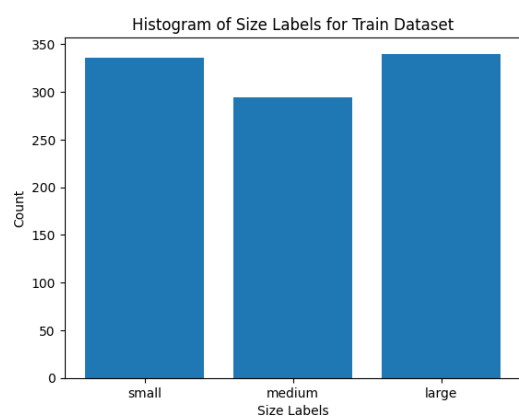


شکل 2.3 هیستوگرام داده ها بر حسب اندازه

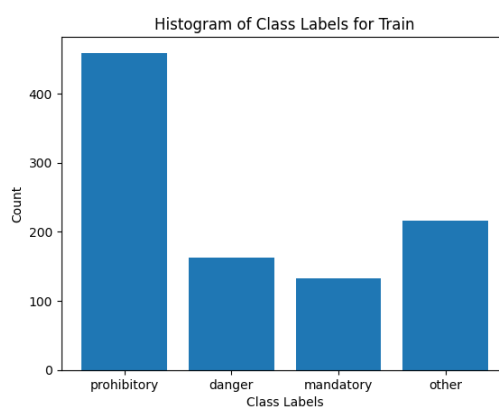


شکل 2.2 هیستوگرام داده ها بر حسب کلاس ها

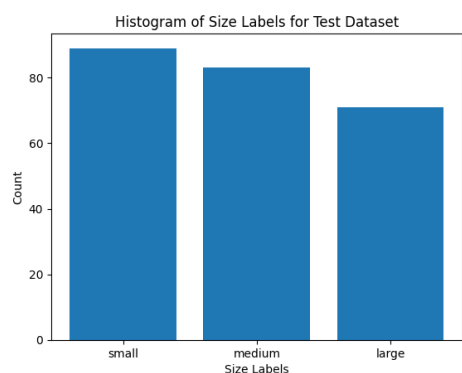
تقسیم بندی این داده ها، از جایی که مدل را از تنسورفلو به تورچ تغییر دادیم، با استفاده از `pytorch` انجام شده است. به این منظور، ابتدا کلاسی را برای دیتاست تعریف کردیم و سپس داده ها را با نسبت 0.2 به دادگان تست و ترین تقسیم بندی کرده و هیستوگرام های سایز و کلاس را برای آن نیز رسم کردیم. هیستوگرام های ذکر شده در تصویر 2.4 تا 2.7 قرار دارند.



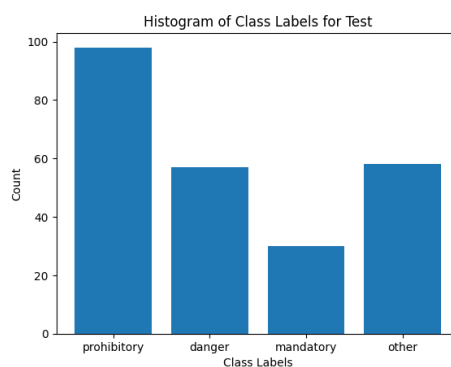
شکل 2.5 هیستوگرام داده های آموزش بر حسب اندازه



شکل 2.4 هیستوگرام داده های آموزش بر حسب کلاس ها



شکل 2.7 هیستوگرام داده های آزمون بر حسب اندازه



شکل 2.6 هیستوگرام داده های آزمون بر حسب کلاس ها

2-2. تنظیم دقیق مدل دو مرحله ای

مدلی که به عنوان بیس استفاده شده است، Faster R-CNN با شبکه پشتیبان ResNet50-FPN است که در کتابخانه پایتورچ وجود دارد. این مدل یک مدل یادگیری عمیق convolutional است که از RPN برای تشخیص اشیا استفاده میکند. RPN بخشی از شبکه است که region های مختلفی از عکس را تشخیص میدهد و آن را به عنوان شی معرفی میکند. همچنین ساختار FPN که یک شبکه pyramid است در این مدل استفاده میشود که باعث میشود بتوانیم اشیا با انواع مختلف اندازه را تشخیص بدهیم. این دو مدل باهمدیگر یک مدل دو مرحله ای میسازند که تشخیص اشیا را به صورت دقیق تر نسبت به مدل های تک مرحله ای انجام میدهد. اشیا تشخیص داده شده به یک SVM داده میشوند تا عمل طبقه بندی انجام شود. این شبکه همان شبکه پشتیبان است که در بالا معرفی شد. این شبکه از ساختار ResNet پیروی میکند که یک شبکه residual است و از 50 لایه تشکیل شده و مدل بسیار عمیقی به شمار میرود. ترکیب این دو شبکه یک شبکه ی تشخیص اشیا و طبقه بندی آن را ارائه میدهد که بسیار عمیق و دقیق میباشد ولی در عوض سرعت کمتری نسبت به سایر مدل ها دارد.

این مدل به صورت پیش آموزش شده در pytorch قرار دارد و برای استفاده از آن کافیست این مدل را ایمپورت کرده و آپشن pretrained را مقدار صحیح بدهیم.

دادگان در کلاس تعریف شده پیش پردازش میشوند و هر عکس را نرمال سازی میکنیم. دیتاست های ذکر شده به دیتالودر تورچ داده میشوند و به علت کمبود میزان حافظه GPU استفاده شده در این تمرین، مقدار هر بچ را 8 قرار داده ایم.

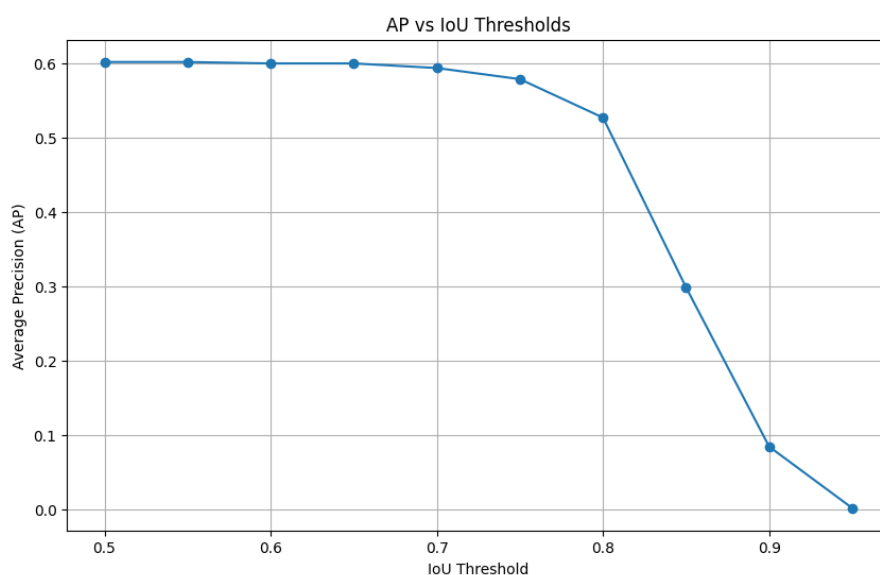
برای ارزیابی مدل ها، از معیار mAP استفاده میکنیم. معیار IoU، نسبت اشتراک دو مجموعه نسبت به اتحاد آن ها را نشان می دهد.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \rightarrow \begin{cases} A: \text{Predicted Region} \\ B: \text{True Region} \end{cases}$$

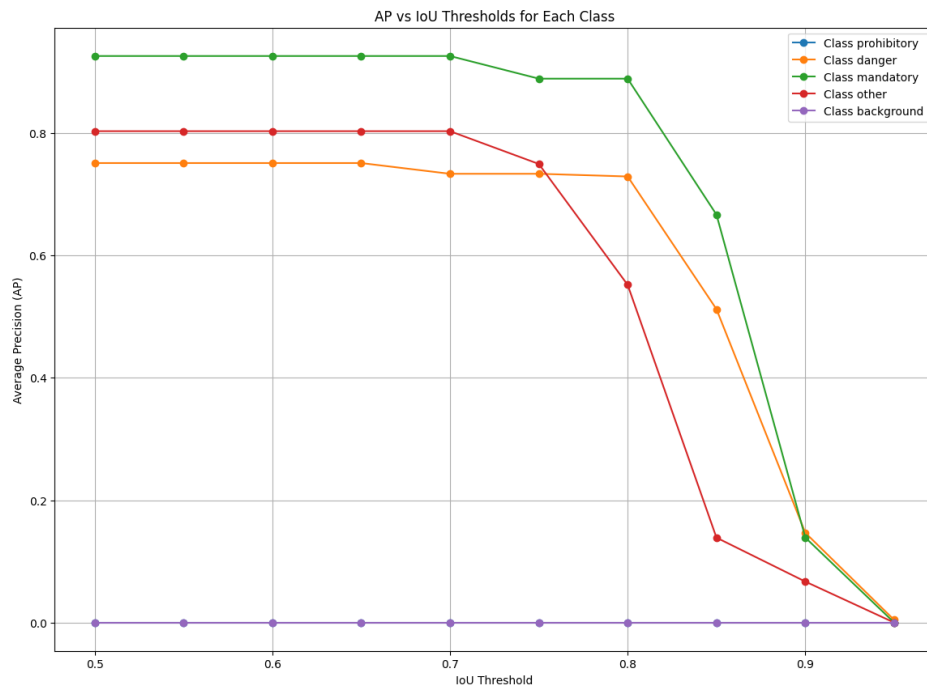
مقدار IoU همیشه میان 0 و 1 می باشد. تمایل داریم تا مقدار IoU به یک میل کند. این معیار، معمولاً برای مسائل Image Segmentation استفاده می شود. معیار mAP دقت مدل تشخیص اشیا را برای threshold های متفاوت IoU اندازه گیری میکند. معیار AP در واقع مساحت زیر نمودار precision-recall است که در تمرین های قبل شرح داده شده بود، و mAP میانگین این معیار برای کلاسه های مختلف را حساب میکند. نمودار های mAP به ازای threshold های مختلف IoU در ادامه برای ارزیابی استفاده شده اند.

برای $loss$ ، مدل تورچ ترکیبی از $loss$ هاس مختلف را ارائه داده که مربوط به خطای طبقه بندی، تشخیص جعبه ای مربوط به اشیا و اندازه ی آن میباشد، در این تمرین خطای محاسبه شده توسط مدل تورچ به عنوان خطا استفاده شده است. برای بهینه سازی از ADAM استفاده کردیم که از روش کاهش گرادیان برای بهینه سازی مدل استفاده میکند. این اپتیمایز سرعت بالایی دارد و به علت $robust$ بودن، گزینه مناسبی برای این کار است. همچنین با توجه به سنگین بودن خود مدل، استفاده از اپتیمایزر های سنگینتر عمل آموزش را بسیار کند میکرد.

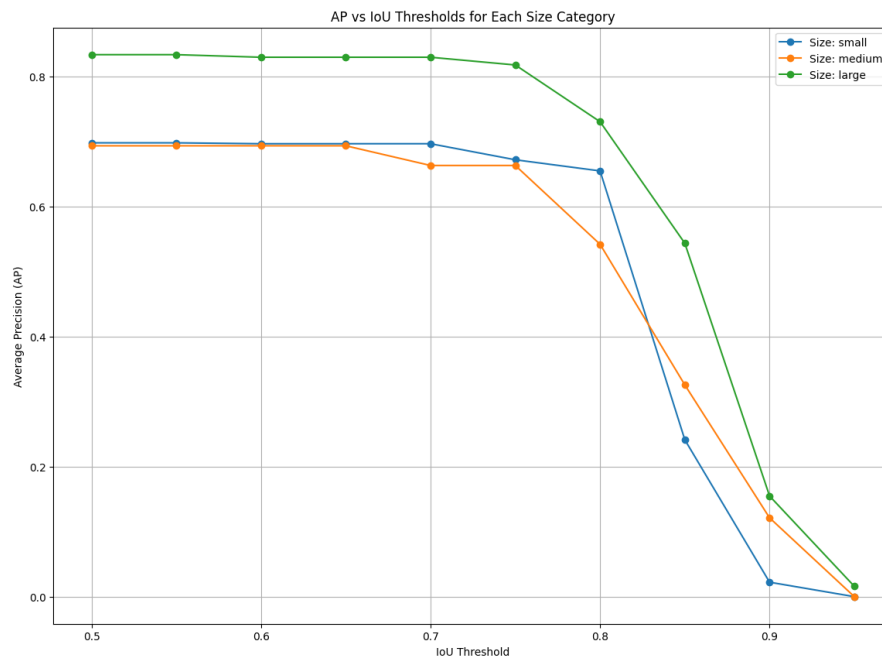
پس از آموزش مدلهای، با استفاده از معیار هایی که از پیش توضیح دادیم، اقدام به تحلیل مدل کردیم. برای این کار از دیتاست تست استفاده شد و نتایج حاصل در تصاویر 2.8 تا 2.10 قابل مشاهده است. مدل پس از 10 اپاک ترین شدن به $mAP = 0.58$ به ازای IoU برابر 0.5 رسید که عدد صحیحی نیست، چون هنگام محاسبه دقت، به این امر دقت نشده بود که کلاس background همواره 0 ap دارد، در نتیجه عدد واقعی احتمالا چیزی حدود 0.72 میباشد، متأسفانه به علت کمبود وقت و نبود منابع کافی برای اجرای دوباره کد، عدد واقعی را به دست نیاوردیم. یک نمونه از تصاویر پیشبینی شده توسط مدل در تصویر 2.11 آمده است.



شکل 2.8 نمودار AP بر حسب IoU مدل دو مرحله ای - مقدار AP بنابر دلایل توضیح داده شده کمتر از مقدار واقعی آن است



شکل 2.9 نمودار AP بر حسب IoU مدل دو مرحله ای بر حسب کلاس



شکل 2.10 نمودار AP بر حسب IoU مدل دو مرحله ای بر حسب اندازه - مقدار AP بنابر دلایل توضیح داده شده کمتر از مقدار واقعی آن است



شکل 2.11 تصویر پیشبینی شده (قرمز) و واقعی (سبز) توسط مدل دو مرحله ای

3-2. تنظیم دقیق مدل تک مرحله ای

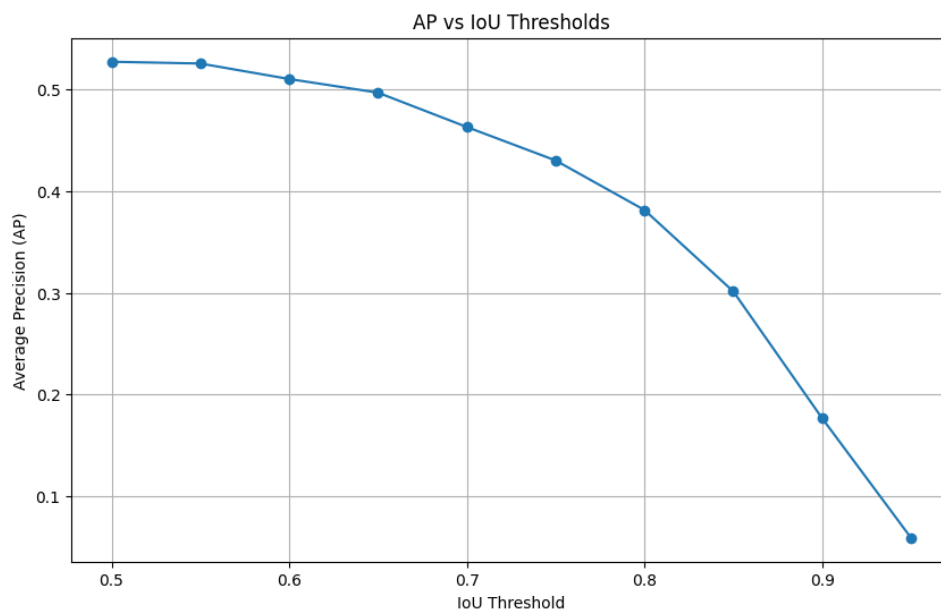
در این مرحله نیز از مدل های آماده ی تورچ برای پیاده سازی مدل خواسته شده استفاده کردیم. مدل معرفی شده از شبکه SDD با شبکه پشتیبان VGG16 استفاده شده است. شبکه SSD یک مدل تشخیص اشیا است که توسط یک شبکه ی عصبی عمیق اشیا را تشخیص می دهد. اشیا تشخیص داده شده توسط این مدل توسط VGG16 که یک مدل طبقه بندی 16 لایه ای است، به کلاس های مختلف طبقه بندی میشوند. ساختار این مدل برای تشخیص real-time و سریع اشیا ایجاد شده است و با اینکه دقت کمتری دارد، سرعت بیشتری نسبت به مدل دو مرحله ای دارد.

تقسیم بندی و آماده سازی دادگان در بخش قبلی توضیح داده شد.

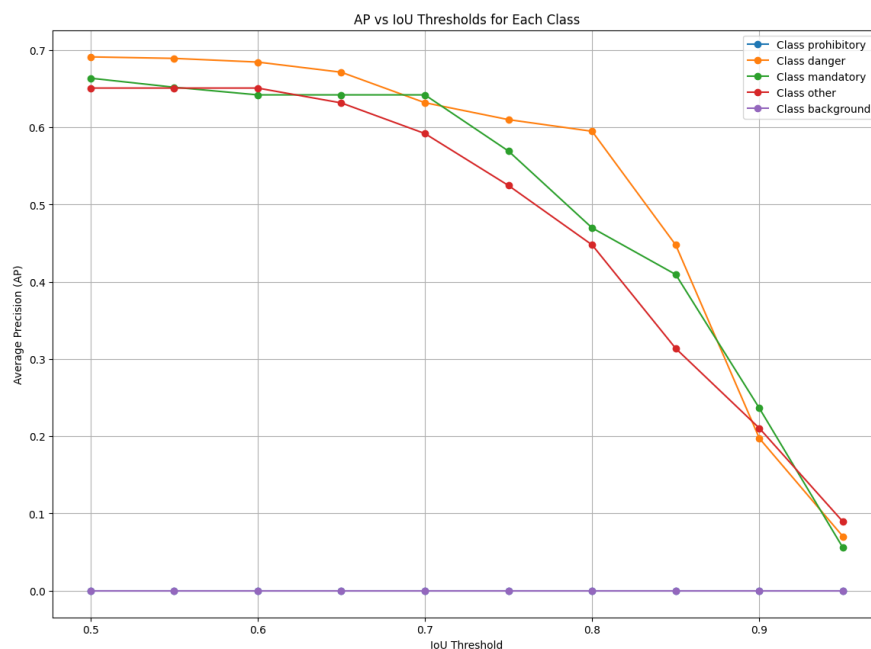
ارزیابی مدل با همان معیار های قبلی انجام میگیرد.

برای تنظیم دقیق، مراحل مشابه با مدل دو مرحله انجام دادیم و سپس اقدام به تحلیل عملکرد مدل کردیم. این مدل نسبت به مدل قبلی به آموزش بیشتری نیاز داشت و 40 اپاک ترین شد، با وجود سرعت بیشتر، همچنان دقت این مدل از مدل قبلی پایینتر است و $mAP=0.52$ بالاترین میزان دقت مدل بود که عدد صحیحی نیست، چون هنگام محاسبه دقت، به این امر دقت نشده بود که کلاس background همواره 0 ap دارد، در نتیجه عدد واقعی احتمالاً چیزی حدود 0.65 میباشد، متأسفانه به علت کمبود وقت و نبود

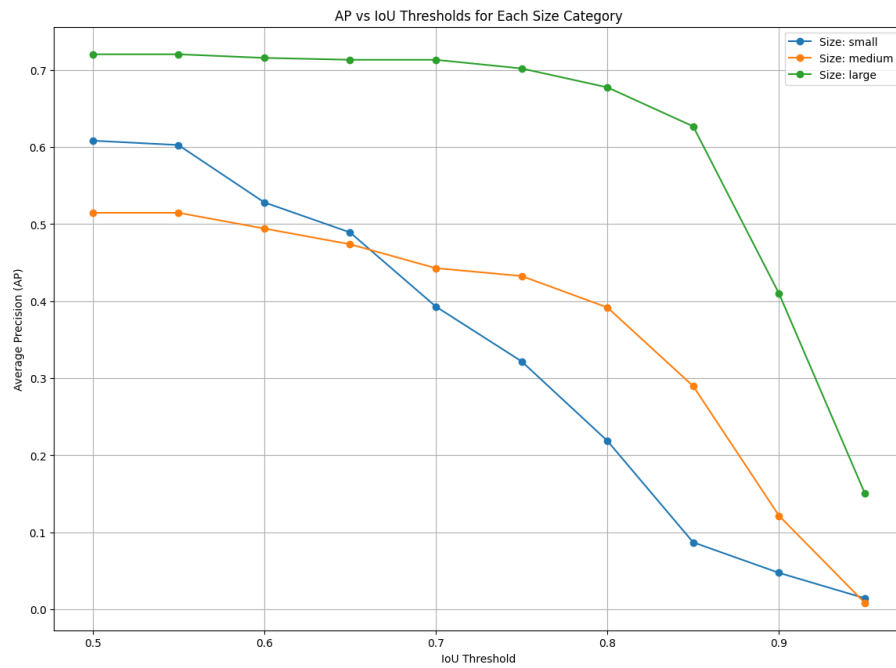
منابع کافی برای اجرای دوباره کد، عدد واقعی را به دست نیاوردیم. نتایج حاصل از آموزش در تصاویر 2.12 تا 2.14 قابل مشاهده است، همچنین تصویر نمونه از پیشبینی این مدل در شکل 2.15 قابل مشاهده است.



شکل 2.12 نمودار AP بر حسب IoU مدل تک مرحله ای - مقدار AP بنابر دلایل توضیح داده شده کمتر از مقدار واقعی آن است



شکل 2.13 نمودار AP بر حسب IoU مدل تک مرحله ای بر حسب کلاس



شکل 2.14 نمودار AP بر حسب IoU مدل تک مرحله ای بر حسب اندازه - مقدار AP بنابر دلایل توضیح داده شده کمتر از مقدار واقعی آن است



شکل 2.15 تصویر پیشبینی شده (قرمز) و واقعی (سبز) توسط مدل تک مرحله ای

2-4. مقایسه مدل ها

با مشاهده نتایج در می یابیم که مدل دو مرحله ای نتایج بهتری نسبت به مدل تک مرحله ای دارد. این نتیجه قابل پیشبینی بود، چرا که مدل دو مرحله ای عمیق تر است و طراحی آن برای افزایش دقت صورت گرفته، در حالی که مدل تک مرحله ای، سرعت را اولویت داده است و لایه های کمتری نسبت به اولی دارد.

برای سائزهای مختلف نیز مدل دو مرحله ای نتایج بهتری دارد. این نتیجه حاصل از بخش FPN است که ساختار pyramid را پیاده کرده است. این ساختار باعث میشود که مدل بتواند در مراحل مختلف، در سائزهای مختلف عکس را تحلیل کند و در نتیجه جوابش تقریباً مستقل از سائز شی باشد. اگر به نتیجه بر حسب اندازه مدل تک مرحله ای دقت کنیم، متوجه میشویم که تابلوهای بزرگ و کوچک کمتر از تابلوهای متوسط تشخیص داده شده اند، در حالی که برای مدل دو مرحله ای این تفاوت کمتر است و سائزهای مختلف تقریباً یک دقت دارند.

برای بهبود مدل ها، میتوان از تکنیک های preprocess برای متعادل کردن کلاسها استفاده کرد، چرا که ما 42 کلاس اصلی را که تقریباً متعادل بودند، تبدیل به 4 کلاس کردیم که دیگر متعادل نیستند و این باعث میشود که برخی از کلاسها بیشتر از دیگری مشاهده شوند. همچنین اگر نخواهیم دیتا را تغییر دهیم، میتوان از تکنیک های آموزش استفاده کرد و مدل را به ازای هر کلاس به صورت جداگانه آموزش داد.