به نام خدا
دانشگاه تهران
دانشکده مهندسی
برق و کامپیوتر

درس شبکه های عصبی و یادگیری عمیق
تمرین دوم

پرسش ۱	نام و نام خانوادگی	رضا دلیر
	شماره دانشجویی	610300050
پرسش ۲	نام و نام خانوادگی	علی مرجبی داکدره
	شماره دانشجویی	610300104
	مهلت ارسال پاسخ	**1404/2/2**

فهرست

قوانین

قبل از پاسخ دادن به پرسش ها، موارد زیر را با دقت مطالعه نمایید:

- از پاسخ های خود یک گزارش در قالبی که در صفحه ی درس در سامانه ی Elearn با نام ***REPORTS_TEMPLATE.docx*** قرار داده شده تهیه نمایید.
- پیشنهاد می شود تمرین ها را در قالب گروه های دو نفره انجام دهید. (بیش از دو نفر مجاز نیست و تحویل تک نفره نیز نمره ی اضافی ندارد) توجه نمایید الزامی در یکسان ماندن اعضای گروه تا انتهای ترم وجود ندارد. (یعنی، می توانید تمرین اول را با شخص A و تمرین دوم را با شخص B و ... انجام دهید)
- **کیفیت گزارش شما در فرآیند تصحیح از اهمیت ویژهای برخوردار است**؛ بنابراین، لطفا تمامی نکات و فرضهایی را که در پیادهسازیها و محاسبات خود در نظر میگیرید در گزارش ذکر کنید.
- در گزارش خود مطابق با آنچه در قالب نمونه قرار داده شده، برای شکل ها زیرنویس و برای جدول ها بالانویس در نظر بگیرید.
- الزامی به ارائه توضیح جزئیات کد در گزارش نیست، اما باید نتایج بدست آمده از آن را گزارش و تحلیل کنید.
- **تحلیل نتایج الزامی می باشد، حتی اگر در صورت پرسش اشاره ای به آن نشده باشد.**
- **دستیاران آموزشی ملزم به اجرا کردن کدهای شما نیستند**؛ بنابراین، هرگونه نتیجه و یا تحلیلی که در صورت پرسش از شما خواسته شده را به طور واضح و کامل در گزارش بیاورید. در صورت عدم رعایت این مورد، بدیهی است که از نمره تمرین کسر میشود.
- **کدها حتما باید در قالب نوت بوک با پسوند .ipynb تهیه شوند، در پایان کار، تمامی کد اجرا شود و خروجی هر سلول حتما در این فایل ارسالی شما ذخیره شده باشد.** بنابراین برای مثال اگر خروجی سلولی یک نمودار است که در گزارش آورده اید، این نمودار باید هم در گزارش هم در نوت بوک کد ها وجود داشته باشد.
- **در صورت مشاهده ی تقلب امتیاز تمامی افراد شرکتکننده در آن، 100- لحاظ میشود.**
- تنها زبان برنامه نویسی مجاز **Python** است.
- **استفاده از کدهای آماده برای تمرینها به هیچ وجه مجاز نیست. در صورتی که دو گروه از یک منبع مشترک استفاده کنند و کدهای مشابه تحویل دهند، تقلب محسوب می شود.**
- نحوه محاسبه تاخیر به این شکل است: پس از پایان رسیدن مهلت ارسال گزارش، حداکثر تا یک هفته امکان ارسال با تاخیر وجود دارد، پس از این یک هفته نمره آن تکلیف برای شما صفر خواهد شد.
    - سه روز اول: بدون جریمه
    - روز چهارم: ۵ درصد
    - روز پنجم: ۱۰ درصد
    - روز ششم: ۱۵ درصد
    - روز هفتم: ۲۰ درصد

- حداکثر نمره ای که برای هر سوال می توان اخذ کرد ۱۰۰ بوده و اگر مجموع بارم یک **سوال** بیشتر از ۱۰۰ باشد، در صورت اخذ نمره بیشتر از ۱۰۰، اعمال نخواهد شد.
  - برای مثال: اگر نمره اخذ شده از سوال ۱ برابر ۱۰۵ و نمره سوال ۲ برابر ۹۵ باشد، نمره نهایی تمرین ۹۷.۵ خواهد بود و نه ۱۰۰.
- لطفا گزارش، کدها و سایر ضمایم را به در یک پوشه با نام زیر قرار داده و آن را فشرده سازید، سپس در سامانه ی Elearn بارگذاری نمایید:

HW[Number] _[Lastname]_[StudentNumber]_[Lastname]_[StudentNumber].zip

(مثال: HW1_Ahmadi_810199101_Bagheri_810199102.zip)

- برای گروه های دو نفره، بارگذاری تمرین از جانب یکی از اعضا کافی است ولی پیشنهاد می شود هر دو نفر بارگذاری نمایند.

پرسش 1. تشخیص بیماران مبتلا به COVID-19 با استفاده از تصاویر X-Ray

1.1- مقدمه

در این تمرین، به بررسی مسئله ی دسته بندی تصاویر اشعه ی ایکس سینه برای تشخیص سه وضعیت «نرمال»، «التهاب ریوی» و «کووید-۱۹» پرداخته ایم. با توجه به اهمیت تشخیص سریع و دقیق بیماری های ریوی، از روش های یادگیری عمیق و شبکه های عصبی کانولوشنی (CNN) برای آموزش مدل هایی استفاده شده که قادر به تشخیص این وضعیت ها با دقت مناسب باشند. همچنین در بخش پایانی تمرین، از تکنیک یادگیری انتقالی با استفاده از مدل های MobileNetV2 و VGG16 برای بهبود عملکرد مدل بهره گرفته شده است. در طول این تمرین از مدل ارائه شده در این مقاله استفاده کردیم با این تفاوت که از دیتاست دیگری کمک گرفتیم. در ادامه به تفصیل مراحل شرح داده میشود.

1.2- تحلیل دیتاست

در این بخش، ابتدا به بررسی ساختار دیتاست و ویژگی های آن می پردازیم. هدف از این مرحله، آشنایی با نحوه ی توزیع داده ها، تعداد کلاس ها و آماده سازی اولیه برای مراحل بعدی مدل سازی است.

1.2.1- شامل چه کلاس هایی است؟

همانطور که از تصویر زیر مشخص است این دیتاست شامل 3 کلاس است.

تصویر 1.1: کلاس های دیتاست

```
class names:  ['PNEUMONIA', 'NORMAL', 'COVID19']
```

Normal: این کلاس مربوط به افراد سالم بدون آسیب ریوی است.

Pneumonia: این کلاس مربوط به بیمارانی است که دچار التهاب ریوی هستند ولی کرونا ندارند.

Covid19: این کلاس مربوط به بیماران مبتلا به ویروس کوید-19 یا کرونا هستند.

1.2.2- فرمت یا فرمت های داده های موجود در دیتاست چیست؟

با استفاده از داده ساختار set که تمام فرمت های عکس ها را به صورت unique نگهداری میکند در بخشی از کد، تمام فرمت ها را بدست آوردیم و تمامی تصاویر موجود در این دیتاست فرمت JPG است.

تصویر 1.2: فرمت تصاویر دیتاست

```
{'jpg'}
```

1.2.3- توزیع داده ها در هر کلاس به چه صورت است؟

تصویر 1.3: توزیع داده ها در هر کلاس

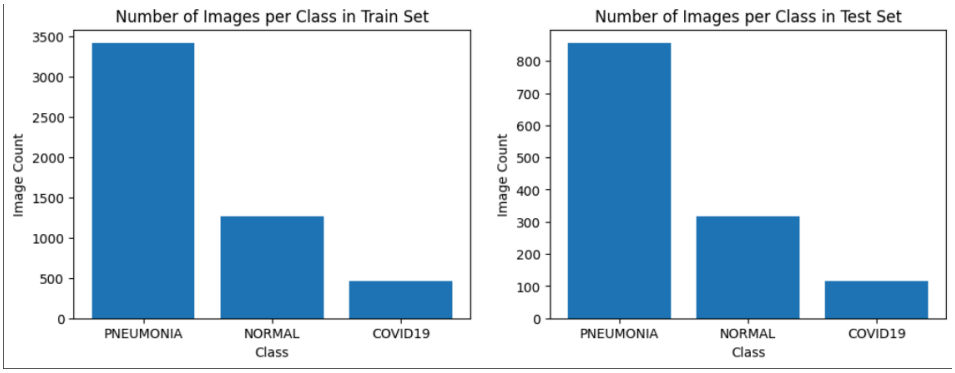
```
number of classes in the PNEUMONIA class in the train set:  3418
number of classes in the NORMAL class in the train set:  1266
number of classes in the COVID19 class in the train set:  460
-----------------------------------------------
number of classes in the PNEUMONIA class in the test set:  855
number of classes in the NORMAL class in the test set:  317
number of classes in the COVID19 class in the test set:  116
```

همانطور که از تصویر بالا مشخص است توزیع داده ها در این مجموعه نامتوازن است. بیشتر تصاویر مربوط به کلاس Pneumonia هستند و کلاس Covid19 کمترین تعداد تصویر را دارد. این عدم توازن هم در مجموعه ی آموزش و هم در مجموعه ی آزمون دیده می شود و می تواند بر عملکرد مدل در تشخیص کلاس های کم نمونه تاثیر

منفی بگذارد که در ادامه به رفع این مشکل خواهیم پرداخت. بنابراین بیشترین توزیع داده ها مربوط به کلاس Pneumonia است و کمترین توزیع را کلاس Covid19 دارد.

1.2.4- هیستوگرام تعداد داده های هر کلاس را برای هر دو مجموعه Train و Test رسم کنید.

نمودار 1: هیستوگرام تعداد داده های هر کلاس



همانطور که قبلتر اشاره شد و در نمودار ها مشخص است توزیع داده ها در بین سه کلاس متوازن نیست.
در هر دو مجموعه ی آموزش و آزمون، کلاس Pneumonia بیشترین تعداد تصویر را دارد، در حالی که کلاس Covid19 کمترین تعداد تصویر را شامل می شود.
این عدم تعادل می تواند باعث شود که مدل در تشخیص نمونه های کلاس Covid19 عملکرد ضعیف تری داشته باشد، چرا که در طول آموزش، کمتر با این نمونه ها مواجه می شود.

1.2.5- توضیح دهید که بالانس بودن کلاس های دیتاست چه مزیتی در شبکه های عصبی دارد؟

بالانس بودن کلاس ها در یک دیتاست باعث می شود که شبکه ی عصبی حین آموزش، همه ی کلاس ها را به یک اندازه ببیند و یاد بگیرد. در صورتی که تعداد نمونه ها در کلاس های مختلف متفاوت باشد، مدل معمولاً به سمت کلاس هایی که تعداد بیشتری دارند متمایل می شود و در نتیجه دقت آن روی کلاس های کم نمونه کاهش پیدا می کند.
این مسئله باعث می شود که مدل در ظاهر دقت خوبی داشته باشد، اما در عمل نتواند دسته بندی درستی انجام دهد، به خصوص در مورد کلاس های اقلیت. بالانس بودن باعث می شود مدل بتواند عملکرد بهتری در تمام کلاس ها داشته باشد، دچار بایاس نشود و توانایی تعمیم بالاتری روی داده های جدید داشته باشد.

1.2.6- در صورت نامتعادل بودن (عدم بالانس) دیتاست، چه راهکاری پیشنهاد میدهید؟(راهکارهای پیشنهادی شامل معرفی ابزارها و کتابخانه های مختلف نیز باشد)

در صورتی که داده های موجود در دیتاست نامتعادل باشند، ممکن است مدل در فرآیند آموزش به سمت کلاس هایی که داده ی بیشتری دارند متمایل شود و در نتیجه دقت کافی برای شناسایی کلاس های اقلیت نداشته باشد. برای رفع این مشکل، راهکارهای مختلفی وجود دارد که می توان آن ها را در دو دسته کلی شامل روش های مبتنی بر داده و روش های مبتنی بر مدل دسته بندی کرد.

یکی از مهم ترین روش ها برای مقابله با عدم توازن داده، استفاده Data Augmentation است. این روش مخصوصا در داده های تصویری بسیار کاربردی بوده و با اعمال تغییراتی مانند چرخش، تغییر مقیاس، قرینه سازی و غیره بر روی تصاویر موجود، نمونه های جدید و متنوعی از کلاس های کم نمونه تولید می کند. این کار باعث می شود که مدل در معرض الگوهای بیشتری قرار گیرد و بهتر بتواند ویژگی های کلاس های اقلیت را یاد بگیرد. برای پیاده سازی این روش می توان از کتابخانه هایی مانند ImageDataGenerator در Keras، یا کتابخانه های پیشرفته تری مثل Albumentations و imgaug استفاده کرد. ما در این تمرین از این تکنیک با استفاده از transform در کتابخانه torchvision بهره می گیریم.

روش دیگر، oversampling یا افزایش مصنوعی نمونه های کلاس اقلیت است که یا با تکرار ساده داده ها انجام می شود و یا با روش هایی مانند SMOTE که داده های جدید را به صورت ترکیبی از داده های موجود تولید می کند. این روش در مسائل طبقه بندی بسیار مؤثر است و با کمک کتابخانه ی imbalanced-learn به راحتی قابل پیاده سازی است. در مقابل، undersampling نیز به عنوان یک روش دیگر مطرح است که با کاهش تعداد نمونه های کلاس های اکثریت سعی در برقراری تعادل دارد، اما ممکن است باعث از دست رفتن اطلاعات مفید شود.

همچنین میتوان به عنوان راهکاری دیگر از وزن دهی به کلاس ها استفاده کرد بدین صورت که به کلاس با تعداد داده کمتر وزن بیشتری و به کلاس با تعداد داده بیشتر وزن کمتری میدهیم تا دیتاست بالانس شود.

در نهایت، برای ارزیابی عملکرد مدل بر روی دیتاست های نامتوازن، نمی توان تنها به معیار accuracy تکیه کرد، چرا که این معیار ممکن است در حضور داده های نامتوازن گمراه کننده باشد. به جای آن، استفاده از معیارهایی مانند precision، recall، f1-score به ما دید دقیق تری از عملکرد مدل، به ویژه در مورد کلاس های اقلیت، خواهد داد.

1.3- پیش پردازش داده ها

Data Augmentation یک تکنیک است که برای افزایش تعداد و تنوع داده ها در یک دیتاست استفاده می شود. این روش به ویژه در مسائل یادگیری ماشین و یادگیری عمیق کاربرد دارد، زیرا مدل های یادگیری عمیق به حجم زیادی از داده برای آموزش نیاز دارند. در مواقعی که داده های کافی در دسترس نباشد، داده افزایی با اعمال تغییراتی مانند چرخش، تغییر اندازه، برش، و تغییر رنگ روی داده های موجود، نمونه های جدید و متنوع تولید می کند.

هدف اصلی داده افزایی بهبود عملکرد مدل و جلوگیری از overfitting است. با ایجاد تنوع در داده ها، مدل بهتر می تواند ویژگی های عمومی تر را یاد بگیرد و این امر باعث می شود که مدل در مواجهه با داده های جدید دقت بالاتری داشته باشد. این تکنیک به ویژه در پردازش تصاویر بسیار مفید است و در کتابخانه هایی مانند Keras، Albumentations، و Torchvision قابل استفاده است.

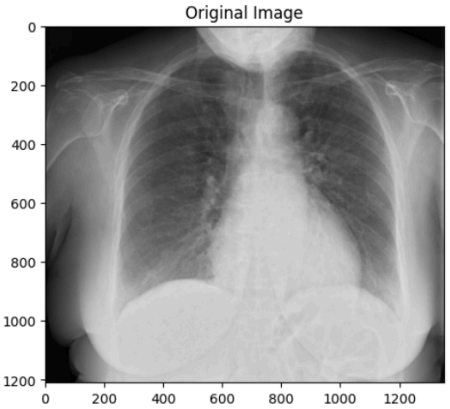
در این تمرین ما از Data Augmentation برای بزرگتر کردن دیتاست استفاده کردیم زیرا به دلیل نامتعادل بودن کلاس ها مجبور شدیم از هر کلاس به اندازه مینیمم تعداد داده های کلاس ها انتخاب کنیم و بعد از این کار تعداد داده ها بسیار کمتر شد. بنابراین با استفاده از Data Augmentation توانستیم تعداد داده ها را بیشتر کنیم(با حفظ تعادل کلاس ها).
در ابتدا طبق تصویر 1.3 از هر کدام از کلاس های داده های آموزش و تست به ترتیب 460 و 116 داده را جدا کردیم و سپس شروع به Data Augmentation کردیم.

ما تعداد داده های دیتاست را به صورت زیر 6 برابر کردیم:
1- چرخش به اندازه 20 درجه
2- قرینه کردن نسبت به محور عمودی
3- قرینه کردن نسبت به محور افقی
4- نصف کردن کنتراست و روشنایی هر داده
5- بریدن و کوچک کردن هر عکس به اندازه 0.8 اندازه اصلی
6- اضافه کردن فیلتر Gaussian Blur به عکس با کرنل 5

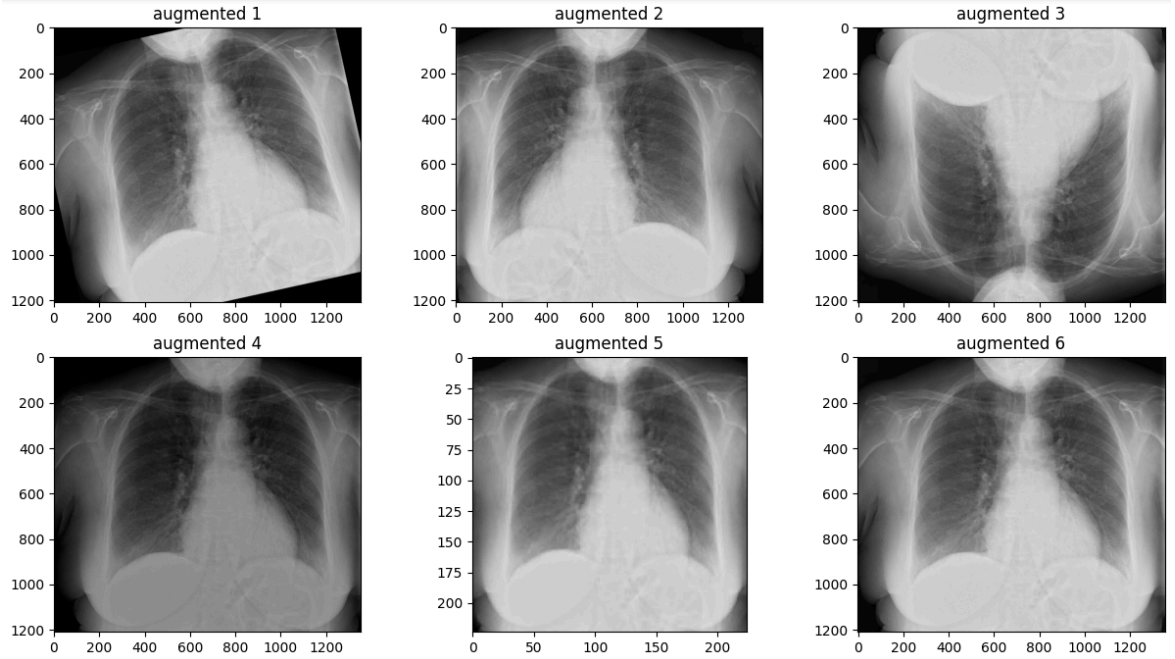
بنابراین بدین ترتیب به ازای هر دیتا 6 عکس جدید به دیتاست اضافه شد و تمامی این عکس ها را در پوشه ای جدید ذخیره کردیم تا در فرایند یادگیری از این عکس ها استفاده کنیم.

برای یکی از عکس های گفته شده نتایج Data Augmentation به صورت زیر قابل مشاهده است:

تصویر 1.4: نمونه ای از عکس اصلی



تصویر 1.5: نتایج Data Augmentation بر روی عکس اصلی بالا



در تصاویر 1 تا 6 طبق موارد 1 تا 6 بالا که ذکر شده است تصاویر دچار چرخش، قرینه و ... شده اند.

دیتاست جدید در پوشه ای جدید به اسم new_dataset در دو پوشه train و تست به صورت جداگانه ذخیره شده اند. همچنین نیاز است که ذکر کنیم در این پروژه، روی داده های تست عملیات data augmentation انجام نشده است. دلیل این کار این است که هدف از مجموعه ی تست، ارزیابی عملکرد واقعی مدل روی داده هایی است که قبلا ندیده و مشابه داده های دنیای واقعی هستند. اگر روی داده های تست هم افزایش داده انجام می دادیم، نتایج ارزیابی ممکن بود غیرواقعی و اغراق آمیز باشد. بنابراین برای داشتن یک ارزیابی منصفانه و دقیق، داده های تست را بدون هیچ تغییری نگه داشتیم.

1.4- آماده سازی مدل

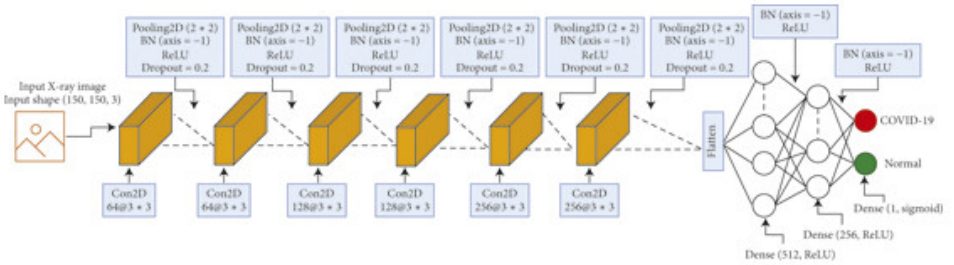
در این مرحله، هدف ساخت و آماده سازی مدل اصلی مبتنی بر CNN برای دسته بندی تصاویر اشعه ی ایکس است. با توجه به توضیحات مقاله، معماری مدل طراحی شده شامل چندین لایه ی کانولوشن، نرمال سازی، فعال سازی و

دراپ اوت است که به صورت مرحله به مرحله پیاده سازی شده اند. در ادامه، جزئیات معماری مدل مطابق با توضیحات مقاله آورده شده است.

تصویر 1.6: جدول اطلاعات مدل در مقاله اصلی

Parameter	Value
Input dimension	(150, 150, 3)
Filter to learn	64, 128, 256
Max pooling	2 × 2
Batch normalization	Axis = −1
Activation functions	ReLU, sigmoid
Dropout rate	20%
Kernel size	3 × 3
Epochs	50
Optimizer	Adam
Loss function	binary_crossentropy

تصویر 1.7: شکل مدل ارائه شده در مقاله اصلی



در سامری مدل میتوانیم مشاهده کنیم همانند چیزی که در مقاله ذکر شده بود دارای 38 لایه است که از لایه های convolution, max pooling, dropout, activation, batch Normalization, flatten and fully connected تشکیل شده است. سایز ورودی تصویر بر روی 224*224 تنظیم شده است که سایز مناسبی برای این تسک است زیرا استفاده از سایز بیشتر موجب کند شدن فرایند آموزش و همچنین کمبود حافظه RAM در میان فرایند آموزش خواهد شد. بنابراین با توجه به ساختار سه کلاسه ی دیتاست، خروجی مدل به صورت یک لایه ی Dense با سه نورون و تابع فعال سازی softmax تعریف شده است.

در این تمرین برای آموزش مدل اصلی، از بهینه ساز Adam با نرخ یادگیری پیش فرض 0.001 استفاده شده است. با وجود اینکه در مقاله اشاره شده از نرخ یادگیری متغیر استفاده شده، در این پیاده سازی ترجیح داده شد از یک نرخ ثابت استفاده شود. دلیل این انتخاب، سادگی روند آموزش و پایداری عملکرد Adam با نرخ پیش فرض در بسیاری از مسائل یادگیری عمیق است. در آزمایش های اولیه نیز مشخص شد که مدل با همین مقدار نرخ یادگیری به خوبی همگرا می شود و دقت قابل قبولی به دست می آید. استفاده از نرخ های کمتر (مثلا 0.0001) باعث کاهش سرعت همگرایی مدل میشود و نرخ های بزرگ تر (مثلاً 0.01) موجب نوسان در مقدار loss و کاهش پایداری آموزش میگردد.

بر اساس نمودارهای دقت و خطا در طول آموزش، میتوان مشاهده کرد که نرخ ثابت 0.001 عملکردی نسبتاً پایدار و قابل قبول داشته و مدل به تدریج به یک نقطه ی تعادل رسیده است. بنابراین در این تمرین، نیازی به استفاده از نرخ یادگیری متغیر احساس نشد.

1.5- آموزش و ارزیابی مدل

مطابق جدول نام برده شده در مقاله اصلی مدل را با 50 ایپاک آموزش دادیم و همچنین نرخ یادگیری را نرخ یادگیری پیش فرض در بهینه ساز Adam یا 0.001 در نظر گرفتیم. برای فرایند آموزش نیاز به Data loader هایی داشتیم که تصاویر را به ترتیب باز کند که به کمک کتابخانه های tf.keras دیتا لودر هایی برای داده های آموزش، validation و تست ساختیم. همچنین مطابق خواسته مسئله از 35 درصد داده ها برای validation استفاده کردیم.

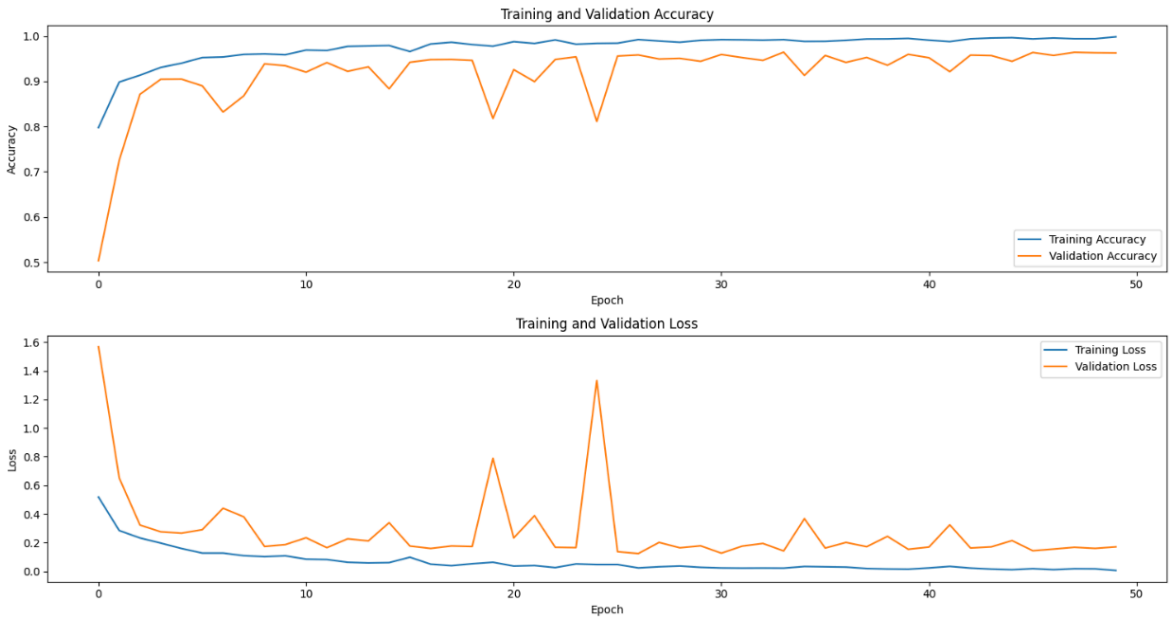
همچنین داده ها به صورت batch با اندازه ی 32 به مدل داده شدند و ترتیب آن ها به صورت تصادفی در هر epoch تغییر کرد (shuffle).

مدل در طول فرایند آموزش روند نسبتاً پایداری را طی کرد. نمودار دقت و خطای آموزش و اعتبارسنجی نشان داد که مدل پس از حدود 30 ایپاک به دقت مناسبی رسیده و تغییرات آن در ایپاک های پایانی خیلی کمتر شده است، به این معنا که مدل به تدریج در حال همگرایی بوده است. همچنین پس از تقریبا 3 ایپاک به دقت بالای 90 درصد رسیدیم که در داده های اعتبار سنجی نیز این مورد صادق بوده است.
در عین حال، عدم وجود نوسان شدید در مقدار loss در مجموعه ی validation نشان می دهد که نرخ یادگیری انتخاب شده مناسب بوده و نیازی به تنظیم دینامیک آن احساس نشد.

پس از اتمام آموزش، عملکرد مدل روی مجموعه ی تست مورد ارزیابی قرار گرفت. برای این منظور، از معیارهایی مانند recall، precision، accuracy, confusion matrix و f1-score استفاده شد تا علاوه بر عملکرد کلی، توانایی مدل در تشخیص صحیح هر کلاس به صورت مجزا نیز بررسی شود. این ارزیابی ها به ما امکان می دهد تا بفهمیم آیا مدل تنها بر کلاس های پرتعداد تمرکز کرده یا اینکه به طور متوازن توانسته همه ی کلاس ها را یاد بگیرد.
هر کدام از این موارد به تفصیل در زیر گزارش شده است.

1.5.1- نمودار خطا و دقت در طول روند آموزش را رسم کنید.

نمودار 1.2: خطا و دقت در طول روند آموزش



از ابتدای آموزش، دقت مدل در داده های آموزش به سرعت افزایش یافته و از حدود 0.8 شروع شده و در ایپاک های اولیه به بالای 0.9 رسیده است. این روند صعودی ادامه یافته تا دقت مدل در آموزش در ایپاک های پایانی تقریباً به 100 درصد نزدیک می شود، که نشان دهنده ی یادگیری کامل مدل بر روی داده های آموزشی است. در مورد داده

های اعتبارسنجی، دقت اولیه نسبتاً پایین و حدود 0.5 است، ولی به سرعت به بالای 0.9 می رسد. با این حال، برخلاف آموزش، نمودار دقت validation دارای نوساناتی است. این نوسان ها به ویژه در اپیاک های میانی به چشم می خورند. از آنجا که بعد از این اپیاک ها نوسانات بسیار کمتر شده است میتوان نتیجه گرفت که مدل به سمت overfitting پیش نرفته است و در حال یادگیری بوده است. به نظر من این مدل برای اینکه عملکردی بی نقص داشته باشد نیاز به داده های بیشتر و جدید تری است تا بتواند به خوبی تشخیص درستی داشته باشد.

در نمودار loss در طی فرایند آموزش همان اطلاعات بالا را مجددا میتوان دریافت کرد و گواهی بر اطلاعات بالا است. هر کجا که دقت به طور ناگهانی کمتر شده است loss افزایش داشته است و بالعکس. همچنین loss در ابتدا از مقادیر نزدیک به 1.6 شروع شده و به سرعت کاهش یافته از تا در نهایت به مقادیر معقولی رسیده است.

1.5.2- مدل را بر روی داده های test تست کرده و عملکرد را گزارش کنید. (راهکاری برای بهتر کردن آن دارید؟ حتی اگر لازم است معماری مدل را تغییر دهید)

پس از آموزش کامل مدل، عملکرد آن بر روی داده های تست مورد ارزیابی قرار گرفت. داده های تست نیز به صورت جداگانه و بدون اعمال هیچ گونه data augmentation بارگذاری شدند تا ارزیابی نهایی به صورت واقعی و بدون تغییر در داده ها انجام گیرد.
در نهایت دقت نهایی مدل در حدود 97.41% بر روی داده های تست به دست آمد. این در حالی است که بر روی داده های ترین در نهایت به دقت 99.8 و بر روی داده های validation به دقت 96.27 درصد رسیده ایم.

علاوه بر دقت کلی، با استفاده از ابزارهایی مانند confusion matrix و عملکرد مدل روی هر کلاس به طور جداگانه بررسی شد. نتایج نشان داد که مدل در تمیز دادن کلاس های Covid19 و Normal بدون هیچگونه خطایی عمل کرده است زیرا این دو کلاس تفاوت زیادی با یکدیگر دارند ولی کلاس Pneumonia به دلیل اینکه شباهت زیادی به هر دو کلاس دارد و به گونه ای میان دو کلاس دیگر است، در داده هایی دچار اشتباه شده است. این در حالی است که همچنان در درصد بسیار بالایی از داده ها به خوبی عمل کرده است.

با توجه به این ارزیابی، چند راهکار برای بهبود عملکرد مدل پیشنهاد می شود:

یکی از راهکارها استفاده از Transfer Learning با مدل هایی مانند MobileNetV2 یا VGG16 است که در ادامه ی تمرین پیاده سازی شده اند. این مدل ها به دلیل آموزش اولیه روی دیتاست های بزرگ و عمومی مانند ImageNet، توانایی بالاتری در استخراج ویژگی های عمومی تصویر دارند و می توانند در دیتاست های کوچک تر مانند این پروژه عملکرد بهتری ارائه دهند.
در طول فرایند یادگیری به بیشترین مشکلی که برخورد داشتم این بود که با کمبود حافظه رم مواجه میشدم و google colab دچار اشکال میشد و فرایند آموزش تکمیل نمیشد. برای رفع کردن این مشکل سایز تصاویر را در فرایند آموزش کاهش دادم و همچنین از دستور tf.data.AUTOTUNE که برای دسترسی دادن برای استفاده بهینه و صحیح از منابع است استفاده کردم. همچنین در فرایند shuffle کردن داده ها تعداد shuffle کردن ها در هر لحظه را به 100 نمونه کاهش دادم تا دچار کمبود منابع RAM نشویم.

در نهایت، در صورت وجود منابع محاسباتی بیشتر، امکان استفاده از معماری های پیشرفته تر مانند EfficientNet یا تنظیم دقیق تر هایپر پارامترها نیز می تواند عملکرد مدل را بهبود ببخشد.

1.5.3- مدل را با معیارهای f1-score ، Precision ، Accuracy و recall ارزیابی و هر کدام از این معیارها را به صورت مفهومی توضیح دهید که هر کدام می تواند چه دیدی از مدل آموزش دیده به طراح آن دهد.

برای ارزیابی نهایی مدل، علاوه بر معیار دقت از معیارهای Recall ،Precision و F1-score نیز استفاده شد تا تحلیل کامل تری از عملکرد مدل به دست آید. هر یک از این معیارها اطلاعات متفاوتی درباره ی نحوه ی رفتار مدل در پیش بینی کلاس ها ارائه می دهند که در ادامه توضیح داده شده اند:

Accuracy:
نشان می دهد که چند درصد از کل نمونه های تست به درستی طبقه بندی شده اند. این معیار ساده ترین شاخص ارزیابی است اما در دیتاست های نامتوازن ممکن است گمراه کننده باشد، چون ممکن است مدل فقط کلاس های پرتعداد را به خوبی پیش بینی کند. در این مدل به accuracy معادل 97.41 درصد رسیدیم.

```
Test Accuracy: 0.9741
```

Precision:
برای هر کلاس، درصد نمونه هایی که مدل به عنوان آن کلاس پیش بینی کرده و واقعاً متعلق به همان کلاس بوده اند را نشان می دهد. این معیار اهمیت زیادی دارد زمانی که بخواهیم از False Positives جلوگیری کنیم. مثلاً در تشخیص بیماری، precision بالا به این معناست که مدل کمتر افراد سالم را به اشتباه بیمار تشخیص می دهد. در این مدل به precision های زیر برای هر کلاس رسیدیم:

تصویر 1.8: بررسی Precision در کلاس های مختلف

	precision
COVID19	1.00
NORMAL	0.96
PNEUMONIA	0.97

Recall:
نشان می دهد مدل از بین کل نمونه های واقعیِ یک کلاس، چه درصدی را به درستی تشخیص داده است. این معیار مهم است زمانی که هدف از دست ندادن نمونه های حساس (False Negatives) باشد. مثلاً در مورد تشخیص COVID-19، recall بالا به این معناست که مدل اکثر بیماران را به درستی شناسایی کرده است. در این مدل به recall های زیر برای هر مدل دست یافتیم:

تصویر 1.9: بررسی Recall در کلاس های مختلف

	recall
COVID19	0.98
NORMAL	0.97
PNEUMONIA	0.97

F1-score:
میانگین هارمونیک بین Precision و Recall است و زمانی استفاده می شود که بخواهیم بین این دو معیار تعادل ایجاد کنیم. این شاخص به خصوص زمانی کاربرد دارد که داده ها نامتوازن باشند یا اشتباه در هر دو حالت (False Positive و False Negative) اهمیت داشته باشد. در این مدل به f1-score های زیر برای هر کلاس دست یافتیم:

تصویر 1.10: بررسی F1-Score در کلاس های مختلف

	f1-score
COVID19	0.99
NORMAL	0.97
PNEUMONIA	0.97

در ارزیابی مدل روی داده های تست، از این چهار معیار استفاده شد تا عملکرد مدل نه فقط از نظر دقت کلی، بلکه در تشخیص صحیح و دقیق هر کلاس نیز بررسی شود. به کمک این معیارها می توان بهتر فهمید که آیا مدل در تشخیص کلاس های با تعداد نمونه های کمتر مانند COVID-19 هم عملکرد خوبی دارد یا نه، و در صورت نیاز، جهت بهبود معماری یا تکنیک های آموزشی تصمیم گیری کرد.

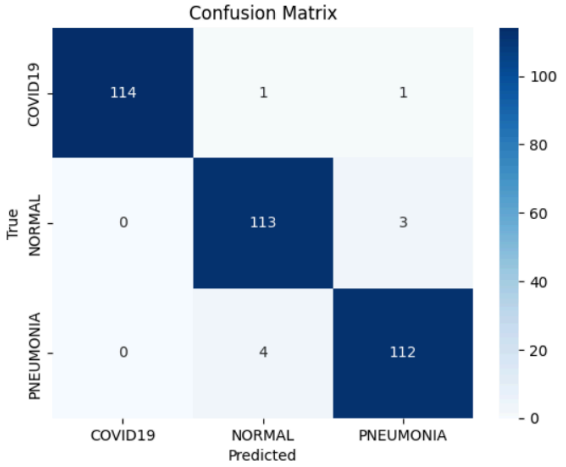
در نهایت کل معیار ها و اطلاعات ارزیابی این مدل در جدول زیر قابل مشاهده است:

نمودار 1.3: ارزیابی مدل بر روی داده های تست

	precision	recall	f1-score	support
COVID19	1.00	0.98	0.99	116
NORMAL	0.96	0.97	0.97	116
PNEUMONIA	0.97	0.97	0.97	116
accuracy			0.97	348
macro avg	0.97	0.97	0.97	348
weighted avg	0.97	0.97	0.97	348

1.5.4- Confusion Matrix را تشکیل داده و بررسی کنید کدام کلاس راحت تر و کدام سخت تر تشخیص داده شده اند؟

تصویر 1.11: ماتریس آشفتگی مدل بر روی داده های تست



طبق این ماتریس، مدل توانسته است اکثر نمونه ها را به درستی تشخیص دهد که نشان از عملکرد قوی آن دارد.

کلاس COVID-19:

از مجموع 116 تصویر متعلق به این کلاس، 114 تصویر به درستی طبقه بندی شده اند، در حالی که تنها 2 مورد اشتباه شناسایی شده اند (1 مورد به عنوان Normal و 1 مورد به عنوان Pneumonia). این عملکرد نشان دهنده ی دقت بسیار بالا در شناسایی COVID-19 است که اهمیت زیادی از نظر کاربردی دارد، زیرا تشخیص صحیح این کلاس حیاتی است.

کلاس Normal:
مدل 113 نمونه را به درستی به عنوان Normal تشخیص داده و تنها 3 مورد را به اشتباه به عنوان Pneumonia پیش بینی کرده است. هیچ نمونه ای از این کلاس به اشتباه به عنوان COVID-19 شناخته نشده است، که نشان از دقت بالای مدل در تفکیک تصاویر Normal از موارد بیماری دار دارد.

کلاس Pneumonia:
عملکرد مدل در این کلاس نیز قابل قبول است؛ از 116 تصویر، 112 مورد به درستی تشخیص داده شده اند و فقط 4 مورد به اشتباه Normal پیش بینی شده اند. مورد اشتباهی به عنوان COVID-19 پیش بینی نشده است، که نشانه ی خوبی از توانایی مدل در تفکیک pneumonia از COVID-19 است.

در نهایت با توجه به نتایج بالا به خوبی نشان داده شد که این مدل در تشخیص کلاس های Normal و Covid19 عملکرد بسیار بسیار عالی ای دارد. در مورد کلاس Pneumonia نیز این مدل به خوبی عمل کرده است اما چون این کلاس شباهت هایی به هر دو کلاس دیگر دارد و به گونه ای میان دو کلاس دیگر قرار میگیرد مقدار خطا اندکی بیشتر از دو کلاس دیگر است. همچنین بیشتر اشتباهات مربوط به تشخیص نادرست بین Pneumonia و Normal بوده که با توجه به شباهت های تصویری بین این دو کلاس، قابل انتظار است. به طور کلی، عملکرد مدل بسیار مناسب است و نیازی به تغییرات اساسی در معماری دیده نمی شود.

1.6- یادگیری انتقالی

یادگیری انتقالی یکی از روش های قدرتمند در یادگیری ماشین و به ویژه یادگیری عمیق است که امکان استفاده از مدل های pre-trained را در مسائل جدید فراهم می سازد. ایده ی اصلی در یادگیری انتقالی این است که دانش کسب شده از یک مسئله، می تواند در حل مسئله ای دیگر مورد استفاده قرار گیرد، به ویژه زمانی که داده های آموزش برای مسئله ی جدید محدود باشند.

مدل هایی مانند VGG16 و MobileNetV2 که بر روی دیتاست های بزرگ و عمومی مانند ImageNet آموزش دیده اند، توانسته اند ویژگی های پیچیده ای از تصاویر را در سطوح مختلف استخراج کنند. به همین دلیل، می توان از آن ها به عنوان feature extractor در مسائل مشابه، نظیر طبقه بندی تصاویر پزشکی، استفاده کرد.

در فرآیند یادگیری انتقالی، معمولاً لایه های اولیه مدل که مسئول استخراج ویژگی های عمومی از تصویر هستند، حفظ می شوند؛ زیرا این ویژگی ها مانند تشخیص لبه، بافت و شکل، در بسیاری از مسائل تصویری مشترک اند. در مقابل، بخش انتهایی مدل (لایه های fully connected یا dense) که برای دسته بندی خاصی آموزش دیده اند، حذف شده و با لایه های جدیدی جایگزین می شوند که متناسب با مسئله ی جدید طراحی شده اند. در این تمرین ما نیز به همین صورت عمل کرده ایم و لایه های ابتدایی را نگه داشتیم و در انتها به آن لایه هایی برای تشخیص سه کلاس اضافه کردیم و تغییراتی متناسب آنها ایجاد کردیم.

استفاده از این تکنیک موجب کاهش زمان آموزش، نیاز کمتر به داده و در عین حال دستیابی به عملکرد قابل قبول می شود. در ادامه، دو مدل VGG16 و MobileNetV2 به صورت جداگانه روی دیتاست تشکیل شده در بخش قبلی آموزش داده می شوند و عملکرد آن ها مطابق معیارهای ارزیابی قبلی بررسی و ثبت خواهد شد.

1.6.1- یادگیری انتقالی در مدل MobileNetV2:

MobileNetV2 یکی از مدل های سبک و کارآمد یادگیری عمیق است که به طور ویژه برای کاربردهای موبایل و سیستم های با منابع محدود طراحی شده است. در این آزمایش، از این مدل به عنوان یک مدل از پیش آموزش دیده در قالب یادگیری انتقالی استفاده شده و لایه های پایانی آن متناسب با مسئله ی ما بازآموزی شده اند.

ابتدا مدل MobileNetV2 با ورودی تصویر 3*224*224 بارگذاری شده است، include_top=False نشان دهنده این است که لایه های Fully Connected نهایی حذف شده اند.

وزن های آموزش داده شده ی ImageNet حفظ شده اند (weights='imagenet') و به دلیل اینکه base_model.trainable = False تعریف شده، در طی آموزش بدون تغییر باقی می مانند. این کار باعث می شود فقط لایه های بالایی آموزش ببینند و زمان آموزش کاهش یابد.

پس از بخش feature extractor، لایه های زیر افزوده شده اند تا وظیفه طبقه بندی سه کلاسه را انجام دهند:

GlobalAveragePooling2D: این لایه به جای استفاده از لایه ی Flatten، به طور میانگین روی ویژگی های استخراج شده عمل کرده و تعداد پارامترها را کاهش می دهد، در نتیجه خطر overfitting کمتر می شود.

Dense(128, activation='relu'): یک لایه ی کاملاً متصل با 128 نورون و تابع فعال سازی ReLU برای یادگیری الگوهای پیچیده تر.

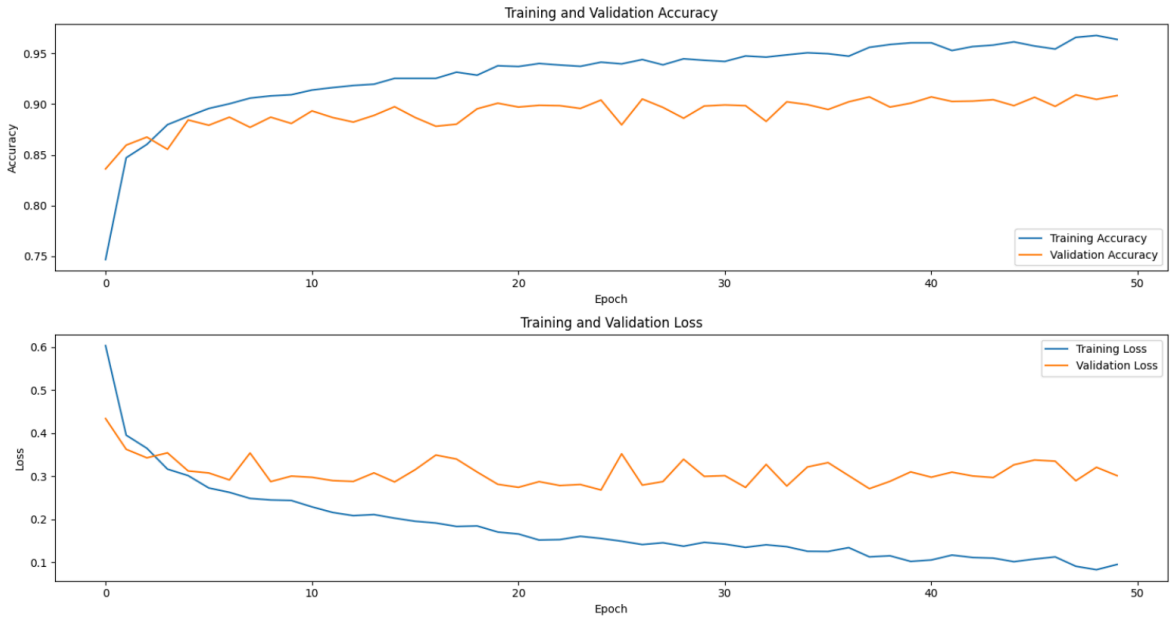
Dropout(0.3): برای جلوگیری از بیش برازش، 30 درصد از نورون ها به صورت تصادفی در هر مرحله آموزش غیرفعال می شوند.

لایه آخر که یک لایه با 3 نورون خروجی است که برای مشخص کردن کلاس نهایی استفاده میشود.

همچنین بقیه تنظیمات مثل بهینه ساز و تابع هزینه نیز مانند مدل اصلی استفاده شده است.

نتایج این مدل در زیر قابل مشاهده است:

تصویر 1.12: نمودار دقت و loss در طی فرایند آموزش در مدل MobileNetV2



همانطور که از نمودار مشخص است برخلاف مدل اصلی به نظر میرسد که تا آخرین اپیاک دقت در حال افزایش تدریجی بوده است و همواره دقت در داده های آموزش بهتر شده اند، اما این مورد درباره داده های Validation صادق نیست و به نظر میرسد که مدل از نقطه ای به بعد در داده های Validation افزایش قابل توجهی نداشته است. البته فاصله میان داده های آموزش و validation به میزانی نیست که نشان دهنده overfitting باشد. در نهایت احتمالا چون مدل در لایه های ابتدایی فریز شده است توانایی دستیابی به عملکرد خارق العاده مانند مدل اصلی این تمرین نداشته است و دقت آن چنان دچار بهبود نشده است. همچنین نمودار خطای آموزش نشان می دهد که مدل به خوبی به همگرایی رسیده و مقدار خطا به زیر 0.1 کاهش یافته است. هرچند که خطای اعتبارسنجی پس از چند اپیاک نخست به حد مشخصی رسیده و تغییر چشمگیری نداشته، اما ثبات آن نشان می دهد که مدل از لحاظ تعمیم پذیری

عملکرد قابل قبولی داشته است. از آنجا که لایه های پایه ی MobileNetV2 در این مدل قفل شده اند، انتظار می رود که fine-tuning این لایه ها بتواند موجب کاهش بیشتر خطای اعتبارسنجی و افزایش دقت کلی مدل شود.

دقت این مدل بر روی داده های تست به صورت زیر است:

تصویر 1.13: دقت بر روی داده های تست در مدل MobileNetV2

```
Test Accuracy: 0.9167
```

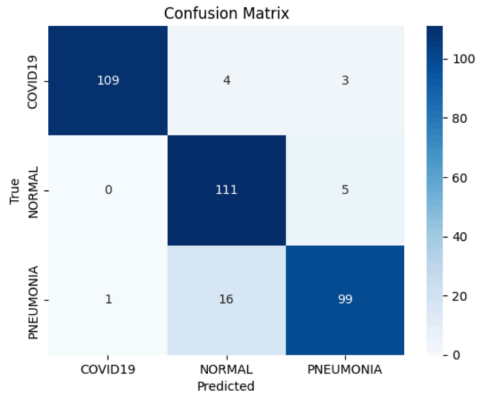
همچنین دیگر معیار های ارزیابی به صورت زیر هستند:

تصویر 1.14: ارزیابی مدل MobileNetV2

	precision	recall	f1-score	support
COVID19	0.99	0.94	0.96	116
NORMAL	0.85	0.96	0.90	116
PNEUMONIA	0.93	0.85	0.89	116
accuracy			0.92	348
macro avg	0.92	0.92	0.92	348
weighted avg	0.92	0.92	0.92	348

نتایج حاصل از معیارهای ارزیابی این مدل نیز نشان دهنده ی عملکرد متعادل و نسبتاً قدرتمند آن در دسته بندی تصاویر به سه کلاس ذکر شده است. میانگین دقت کلی مدل برابر با 92 درصد است که بیانگر قدرت قابل قبول مدل در تشخیص صحیح نمونه هاست. معیار F1-score که توازنی میان precision و recall ارائه می دهد، برای کلاس های مختلف بین 0.89 تا 0.96 متغیر است. کلاس Covid19 بالاترین مقدار F1-score یعنی 0.96 را به دست آورده که نشان دهنده ی تشخیص بسیار دقیق این کلاس توسط مدل است، در حالی که کلاس Pneumonia با F1-score معادل 0.89 اندکی عملکرد ضعیف تری نسبت به سایر کلاس ها دارد.

تصویر 1.15: ماتریس درهم ریختگی مدل MobileNetV2



ماتریس درهم ریختگی نیز حاکی از آن است که بیشترین اشتباهات مدل مربوط به تداخل بین کلاس های Normal و Pneumonia بوده، که می تواند به دلیل شباهت های ظاهری تصاویر در این دو کلاس باشد. با این حال، عدم وجود خطای طبقه بندی کلاس Normal به Covid19 و بالعکس، نشان می دهد که مدل توانسته مرزهای تصمیم گیری نسبتا مشخصی میان این کلاس ها یاد بگیرد. در مجموع، معیارهای ارزیابی نشان می دهند که این مدل علیرغم قفل بودن لایه های ابتدایی، عملکردی بسیار مناسب در تشخیص سه کلاس مذکور ارائه داده و قابلیت تعمیم پذیری بالایی از خود نشان داده است.

در مجموع، مدل MobileNetV2 با استفاده از تکنیک یادگیری انتقالی توانسته عملکرد قابل قبولی در طبقه بندی تصاویر اشعه ی ایکس سینه به سه کلاس اصلی ارائه دهد. این مدل با وجود سبک بودن و داشتن تعداد پارامترهای کمتر نسبت به مدل های سنگین تر، دقت بالا و تعمیم پذیری خوبی از خود نشان داده است. نتایج ارزیابی بیانگر آن

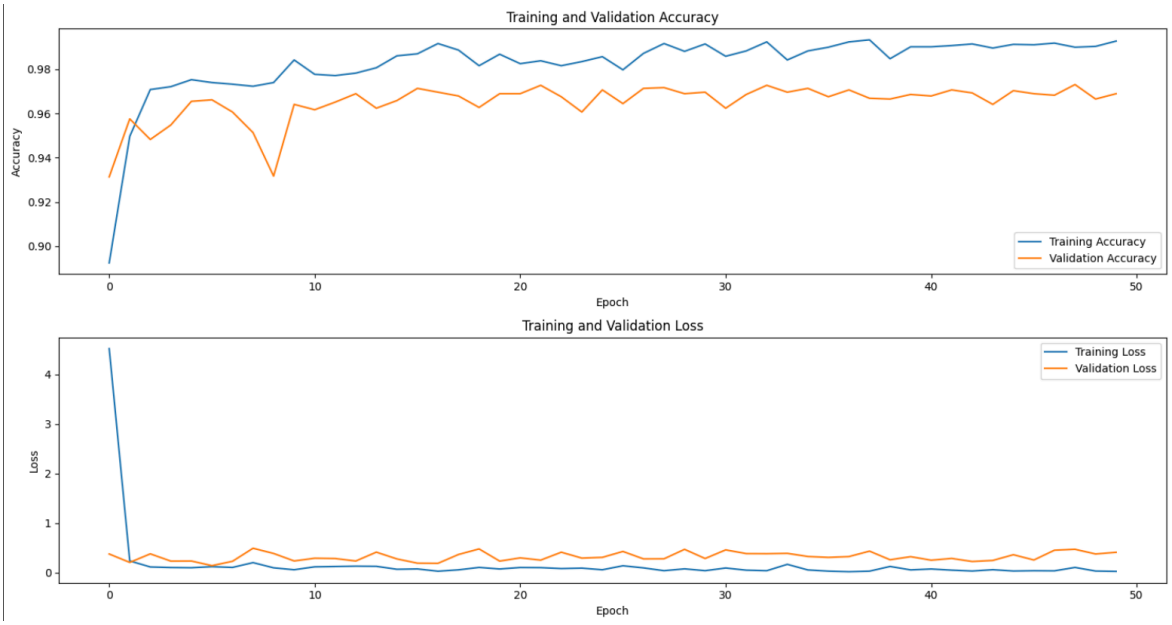
است که مدل در شناسایی صحیح کلاس Covid19 بسیار موفق عمل کرده و در سایر کلاس ها نیز عملکرد رضایت بخشی داشته است. با توجه به اینکه لایه های پایه ی مدل در طول آموزش قفل بوده اند، می توان نتیجه گرفت که حتی بدون fine-tuning نیز این مدل توانایی خوبی در استخراج ویژگی های موثر داشته است. با این حال، امکان بهبود بیشتر عملکرد با fine-tuning بخش هایی از مدل یا استفاده از داده های بیشتر همچنان وجود دارد.

1.6.2- یادگیری انتقالی در مدل VGG16:

این مدل نیز با استفاده از تکنیک یادگیری انتقالی و با بهره گیری از وزن های آموزش دیده روی دیتاست ImageNet به کار گرفته شده است. در ساختار فعلی، لایه های انتهایی مدل برای تطبیق با مسئله ی طبقه بندی سه کلاسه ی تصاویر اشعه ی ایکس سینه بازطراحی شده اند. هدف این بخش، ارزیابی توانایی VGG16 در یادگیری و تعمیم به داده های پزشکی و مقایسه ی آن با سایر مدل هاست.

تمامی تنظیمات این مدل نیز مشابه مدل MobileNetV2 است و هیچ تفاوتی با آن ندارد، بنابراین لایه های ابتدایی از این مدل بدون تغییر نگه داشته شده اند و در نهایت به آن لایه هایی برای طبقه بندی 3 کلاس در این مسئله اضافه شده است. نمودار دقت و خطا در طول فرایند یادگیری نیز به صورت زیر است:

تصویر 1.16: نمودار دقت و خطا در طول فرایند یادگیری در مدل VGG16



همانطور که از نمودار مشخص است مدل به خوبی و با دقت بسیار بالایی آموزش دیده شده است و دقت در داده های آموزش به بالای 98 درصد رسیده است. در داده های validation نیز دقت به بالای 97 درصد رسیده است که بسیار خوب است و نشان دهنده یادگیری و تعمیم پذیری بسیار خوب مدل است. در نمودار خطا نیز این مورد تایید می شود زیرا خطا دائما در کمتر از 1 باقی مانده است. همچنین اندکی اعوجاج در نمودار و بالا و پایین شدن دقت و خطا طبیعی است و اثری از overfitting به چشم نمی خورد.

نتایج ارزیابی این مدل بر روی داده های تست به صورت زیر است:

تصویر 1.17: دقت مدل VGG16 بر روی داده های تست

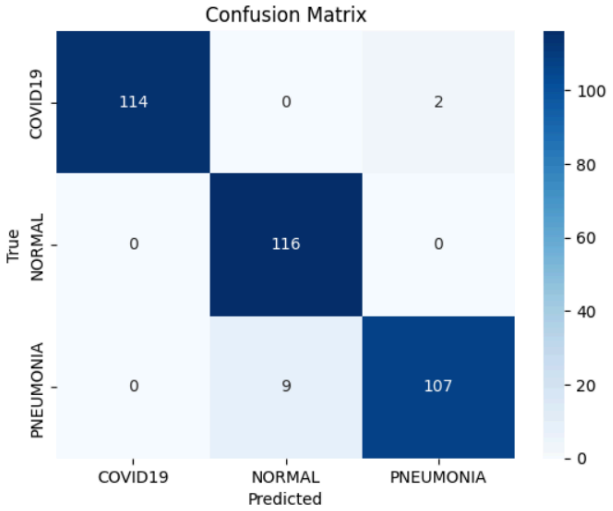
```
Test Accuracy: 0.9684
```

تصویر 1.18: ارزیابی مدل VGG16 بر روی داده های تست

	precision	recall	f1-score	support
COVID19	1.00	0.98	0.99	116
NORMAL	0.93	1.00	0.96	116
PNEUMONIA	0.98	0.92	0.95	116
accuracy			0.97	348
macro avg	0.97	0.97	0.97	348
weighted avg	0.97	0.97	0.97	348

جدول معیارهای ارزیابی نشان دهنده ی عملکرد درخشان این مدل است. دقت مدل برابر با 96.84 درصد است که نسبت به مدل MobileNetV2 بهبود یافته است اما هنوز مقدار خیلی کمی از مدل اصلی در مقاله کمتر است.
F1-score برای کلاس Covid19 برابر با 0.99 و برای کلاس های Normal و Pneumonia به ترتیب 0.96 و 0.95 گزارش شده است، که هر سه مقدار بسیار بالا و متعادل هستند.

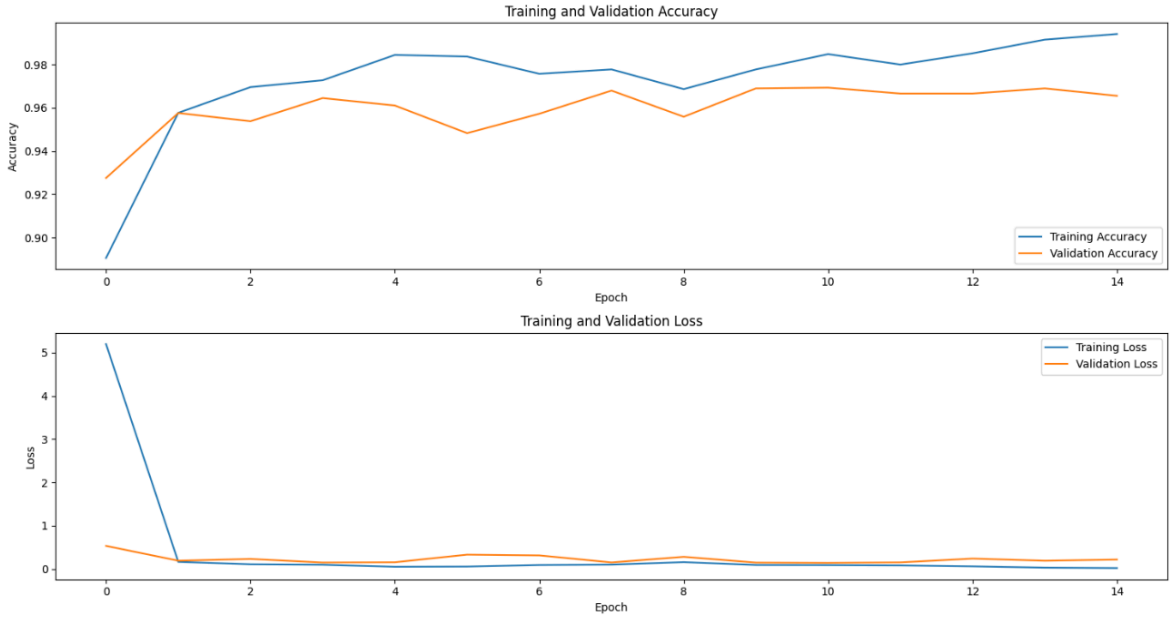
تصویر 1.19: ماتریس آشفتگی مدل VGG16



ماتریس آشفتگی نیز این موضوع را تأیید می کند؛ مدل موفق به پیش بینی صحیح اکثر نمونه ها شده است و تنها در کلاس Pneumonia کمی با تداخل با کلاس Normal مواجه بوده است (۹ مورد اشتباه). عدم وجود هیچ اشتباهی بین Covid19 و دیگر کلاس ها نیز نشان می دهد که VGG16 در تشخیص این بیماری حیاتی عملکردی بسیار دقیق دارد.

با وجود اینکه عملکرد این مدل بسیار خوب بوده است به دلیل تفاوتی حدود دو تا سه درصدی میان دقت داده های تست و به این مورد که آیا مدل دچار overfitting شده است یا خیر شک کردم و تصمیم گرفتم که مدل را مجددا با 15 اپیاک ترین کنم تا از overfitting احتمالی جلوگیری کنم. پس از آموزش مجدد این مدل نتایج به صورت زیر بود:

تصویر 1.20: نمودار دقت و خطا در طول فرایند یادگیری در مدل VGG16 با 15 اپیاک



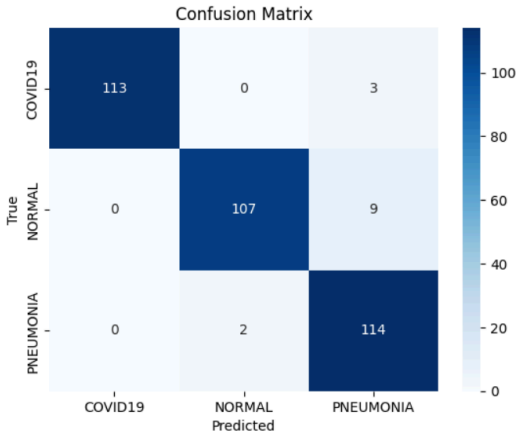
تصویر 1.21: دقت مدل VGG16 در داده های تست با 15 ایپاک

```
Test Accuracy: 0.9598
```

تصویر 1.22: ارزیابی مدل VGG16 بر روی داده های تست با 15 ایپاک

	precision	recall	f1-score	support
COVID19	1.00	0.97	0.99	116
NORMAL	0.98	0.92	0.95	116
PNEUMONIA	0.90	0.98	0.94	116
accuracy			0.96	348
macro avg	0.96	0.96	0.96	348
weighted avg	0.96	0.96	0.96	348

تصویر 1.23: ماتریس آشفتگی مدل VGG16 با 15 ایپاک



همانطور که از نمودار ها و تصاویر بالا مشخص است دقت و دیگر معیار های ارزیابی برای همان مدل که با 15 ایپاک آموزش دیده است کمتر از زمانی است که مدل با 50 ایپاک آموزش دیده شده است. بنابراین این نشان دهنده این است که مدل با 50 ایپاک دچار overfitting نشده است و از generalization بهتری برخوردار است.

در نهایت به مقایسه سه مدل در یک جدول می پردازیم:

Model	Accuracy (%)	Precision (avg)	Recall (avg)	F1-Score (avg)	Number of Parameters (trainable)	Analysis
Model in the paper	**97.41**	**0.97**	**0.97**	**0.97**	**Normal (million 2)**	**high accuracy and trained completely, strong ability to predict, stable**
MobileNetV2	**91.67**	**0.92**	**0.92**	**0.92**	**low and light (thousand 165)**	**balanced and good performance with lightweight architecture**

VGG16	**96.84**	**0.97**	**0.97**	**0.97**	**high (million 13)**	**stable performance and very accurate**

1.7- نتیجه گیری

در این پروژه، هدف طراحی و ارزیابی یک مدل یادگیری عمیق برای طبقه بندی تصاویر اشعه ی ایکس سینه به سه کلاس Normal ،Covid19 و Pneumonia بوده است. ابتدا یک مدل پایه با ساختار CNN از ابتدا طراحی و آموزش داده شد. این مدل شامل ترکیبی از لایه های کانولوشن، نرمال سازی، فعال سازی ReLU، dropout، لایه ی flatten و لایه های Fully Connected بود. مدل پایه توانست دقت بسیار بالایی به دست آورد و در آزمون نهایی، با دقت کلی 97 درصد عملکردی در سطح مدل های پیچیده تر از پیش آموزش دیده نشان داد. سپس به سراغ یادگیری انتقالی رفتیم و دو مدل معروف MobileNetV2 و VGG16 را مورد استفاده قرار دادیم. در هر دو مدل، لایه های ابتدایی به صورت ثابت نگه داشته شدند و تنها لایه های انتهایی با داده های جدید بازآموزی شدند تا تطبیق مناسبی با مسئله ی مورد نظر حاصل شود. مدل MobileNetV2 با وجود حجم کم و معماری سبک، توانست دقت کلی 92 درصد را به دست آورد و در تشخیص کلاس Covid19 بسیار موفق عمل کند. در مقابل، مدل VGG16 نیز با ساختار عمیق تر خود، عملکردی در سطح مدل پایه از خود نشان داد و با دقت نهایی 97 درصد، تعادل بسیار خوبی در میان دقت، Recall ،Precision و F1-score در تمام کلاس ها حفظ کرد.

با مقایسه ی دقیق سه مدل، می توان دریافت که مدل پایه در عین حال که از ابتدا آموزش داده شده، به خوبی توانسته با مدل های پیشرفته ی انتقال یادگیری رقابت کند. این موضوع اهمیت طراحی دقیق معماری و تنظیم مناسب پارامترهای مدل را نشان می دهد. مدل MobileNetV2 مناسب شرایطی است که محدودیت منابع وجود دارد یا مدل باید در سیستم هایی مانند موبایل یا دستگاه های کم توان اجرا شود. در طرف مقابل، مدل VGG16 و مدل پایه در شرایطی که دقت بالا و تعمیم پذیری قوی مورد نیاز است، انتخاب های بسیار مناسبی هستند.

در نهایت، با وجود قدرت مدل های از پیش آموزش دیده، همچنان نیاز به طراحی مدل های خاص و سفارشی مانند آنچه در مقاله اصلی ارائه شده وجود دارد. چرا که این مدل ها می توانند با توجه به ماهیت داده های خاص، معماری متناسب تری داشته باشند و در بعضی موارد، عملکرد بهتری نسبت به مدل های عمومی نشان دهند. علاوه بر این، توسعه ی مدل های سبک تر و دقیق تر، گامی مهم در جهت فراهم آوردن امکان استفاده از هوش مصنوعی در کاربردهای واقعی و بالینی به شمار می رود. بنابراین، پژوهش در زمینه طراحی معماری های اختصاصی و بهینه، همچنان ارزشمند و ضروری است. بنابراین هنوز نیاز است که مدل های جدید تر و اختصاص یافته به یک مسئله خاص طراحی و پیاده سازی شود، زیرا یک مدل از پیش آموزش دیده یا بسیار سنگین است و میتواند دقتی تقریبا مشابه مدل خاص طراحی شده نشان دهد و یا سبک است ولی دقتی کمتر دارد، پس با طراحی مدلی خاص میتوان به بهینه ترین طراحی برای یک مسئله خاص رسید.

همچنین در نهایت بهترین مدل برای این مسئله مدلی است که مقاله ارائه داده و میتواند با دقت بسیار بالایی کلاس ها را از یکدیگر جدا کند و همچنین این مدل سبک تر است.

پرسش ۲ - طبقه بندی خودرو با استفاده از VGG16 و SVM

2.1- مقدمه:

در این پروژه، تصمیم به طبقه بندی (Classification) داده های تصویری مدل های مختلفی از اتومبیل های برند TOYOTA داریم، که از بین حدودا 30 مدل، 10 مدل منتخب شده و با بررسی کوتاهی بر روی دیتاست آغاز و با Data Augmentation پیش رفته و چندین مدل و روش را برای طبقه بندی آنها نیز پیشی میگیریم و نتایج را نیز مختصرا بررسی خواهیم کرد.

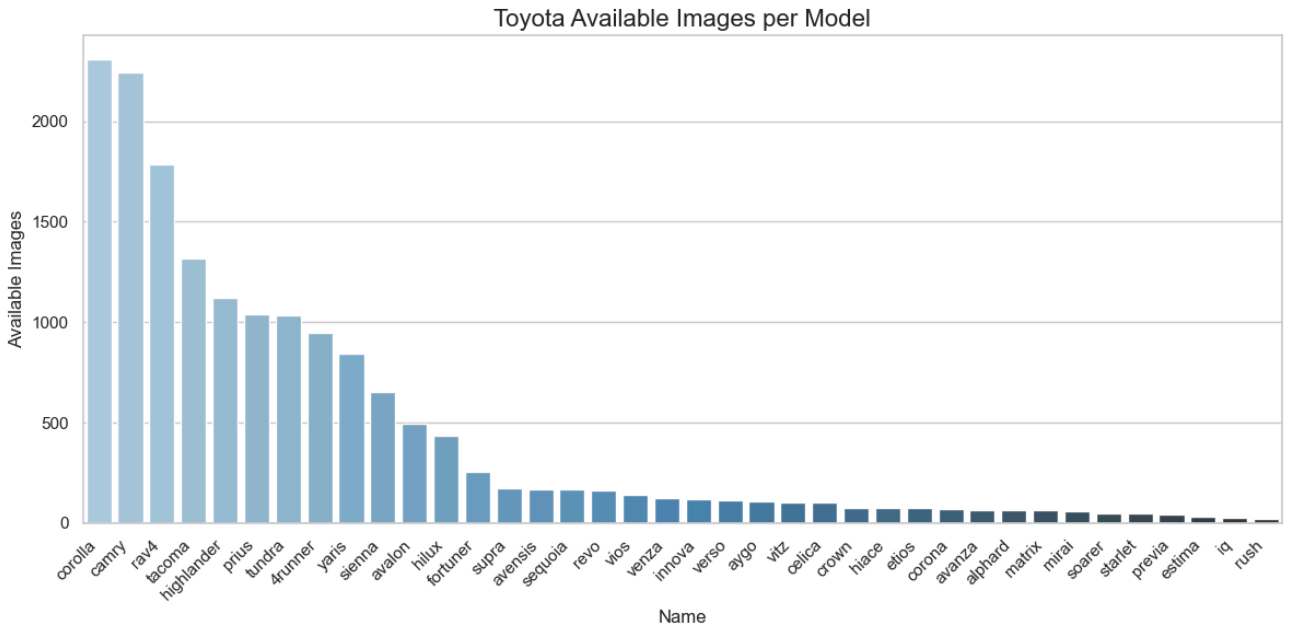
2.2.1- توصیف اولیه داده ها:

مجموعه داده شامل تصاویری از مدل‌های متعدد خودروی تویوتا است که هر مدل یک کلاس مجزا برای طبقه‌بندی را نشان می‌دهد. بررسی اولیه نشان داد که توزیع تعداد تصاویر در هر کلاس بسیار نامتوازن است و از کمترین تعداد ۲۳ تصویر تا بیش از ۲۳۰۰ تصویر برای هر مدل متغیر است. مدل ها شامل موارد زیر هستند:

```
4runner, alphard, avalon, avanza, avensis, aygo, camry, celica
corolla, corona, crown, estima, etios, fortuner, hiace, highlander
hilux, innova, iq, matrix, mirai, previa, prius, rav4
revo, rush, sequoia, sienna, soarer, starlet, supra, tacoma
,tundra, venza, verso, vios, vitz, yaris
```

و میزان توزیع آنها به صورت زیر است:

تصویر 1.1: نمودار توزیع داده ها در مدل های مختلف



2.2.2- انتخاب کلاس ها

با توجه به نیاز شبکه‌های عصبی به حجم قابل توجهی داده برای هر کلاس جهت یادگیری موثر، تصمیم گرفته شد زیرمجموعه‌ای از 10 مدل خودرو برای وظیفه طبقه‌بندی انتخاب شود. برای کاهش چالش‌های ناشی از تعداد بسیار کم تصاویر و احتمال عدم توازن شدید کلاس‌ها که می‌تواند بر آموزش مدل تأثیر بگذارد، 10 کلاسی که بیشترین تعداد تصویر را داشتند انتخاب شدند. این انتخاب نقطه شروع قوی‌تری را با تعداد تصویر در محدوده 652 تا 2311 برای کلاس‌های منتخب فراهم کرد. کلاس‌های منتخب عبارت بودند از:

- کرولا (Corolla)
- کمری (Camry)
- راو4 (RAV4)
- تاکوما (Tacoma)
- هایلندر (Highlander)
- پریوس (Prius)
- توندرا (Tundra)
- 4رانر (4Runner)
- یاریس (Yaris)
- سینا (Sienna)

2.2.3- آماده‌سازی داده‌ها:

داده‌های تصویری خام نیاز به چندین مرحله پیش‌پردازش داشتند تا برای آموزش مدل مناسب شوند:

بارگیری و تغییر اندازه تصاویر: تصاویر برای 10 کلاس منتخب از دایرکتوری‌های مربوطه لود شدند. اندازه آن‌ها به ابعاد ثابت (224x224 پیکسل) تغییر داده شد و به فرمت استاندارد (آرایه‌های NumPy با 3 کانال رنگی) تبدیل شدند.

نرمال‌سازی: مقادیر پیکسل با تقسیم بر 255 به محدوده‌ای مناسب برای شبکه‌های عصبی (مانند [0, 1]) نرمال شدند.

افزایش داده (Data Augmentation): برای رفع عدم توازن باقیمانده در 10 کلاس منتخب و افزایش تنوع داده‌های آموزشی، تکنیک‌های افزایش داده برای کلاس‌هایی با تعداد تصویر نسبتاً کمتر (مانند آن‌هایی که کمتر از 1500 تصویر داشتند) اعمال شد. تکنیک‌هایی مانند چرخش، زوم، تغییر موقعیت، برش و ورق زدن افقی با استفاده از ImageDataGenerator اعمال شدند. تصاویر افزایش یافته تولید و در کنار تصاویر اصلی ذخیره شدند.

تقسیم داده‌ها (Data Splitting): مجموعه داده ترکیبی (تصاویر اصلی از هر 10 کلاس به علاوه تصاویر افزایش یافته تولید شده برای کلاس‌های هدف) به مجموعه‌های آموزشی و آزمایشی تقسیم شد. نسبت تقسیم رایج (مثلاً 80% آموزش، 20% تست) اعمال شد. برای ارزیابی دقیق، تقسیم سه‌بخشی (آموزش، اعتبار سنجی، تست) مورد بحث قرار گرفته و پیاده‌سازی شد، که مجموعه‌های جداگانه‌ای را برای آموزش، نظارت بر عملکرد در طول آموزش (اعتبارسنجی)، و ارائه یک ارزیابی نهایی ایجاد کرد.

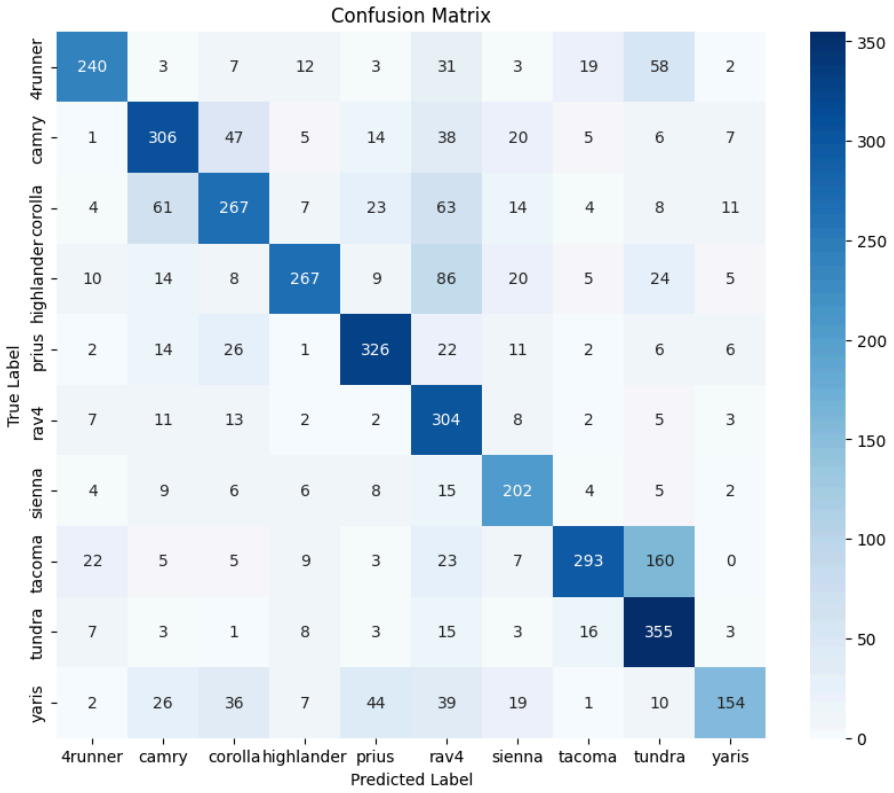
2.3- مدل VGG16 (با Transfer Learning)

این مدل از پیش آموزش دیده شده، مدلی معروف و اموزش دیده شده روی مجموعه داده ی بزرگی است که 1000 کلاس برای طبقه بندی داشته است؛ این بدین معناست که لایه آخر آن نیز متناسب با همان 1000 کلاس است، و حال نیاز است که تغییری در آن ایجاد کرده تا متناسب با هدف مد نظر شود.

از این جهت به بخش آخر مدل VGG16 (در بخش لایه های fully-connected)، لایه ای شامل 10 نورون اضافه می‌شود؛ علاوه بر آن لایه های قبلی یا در واقع لایه های قبل از لایه های تماما متصل، که نقش استخراج ویژگی عکس ها را دارند را فریز یا غیر قابل تغییر میکنیم و سپس با داده های موجود، فرآیند fine-tuning را انجام میدهیم.

بعد از انجام این فرآیند، نتایج به صورت زیر بوده است: (**دقت Accuracy در مجموعه تست:** 67%)

تصویر 1.2: ماتریس آشفتگی مدل VGG16



تحلیل ماتریس درهم‌ریختگی (Confusion Matrix):

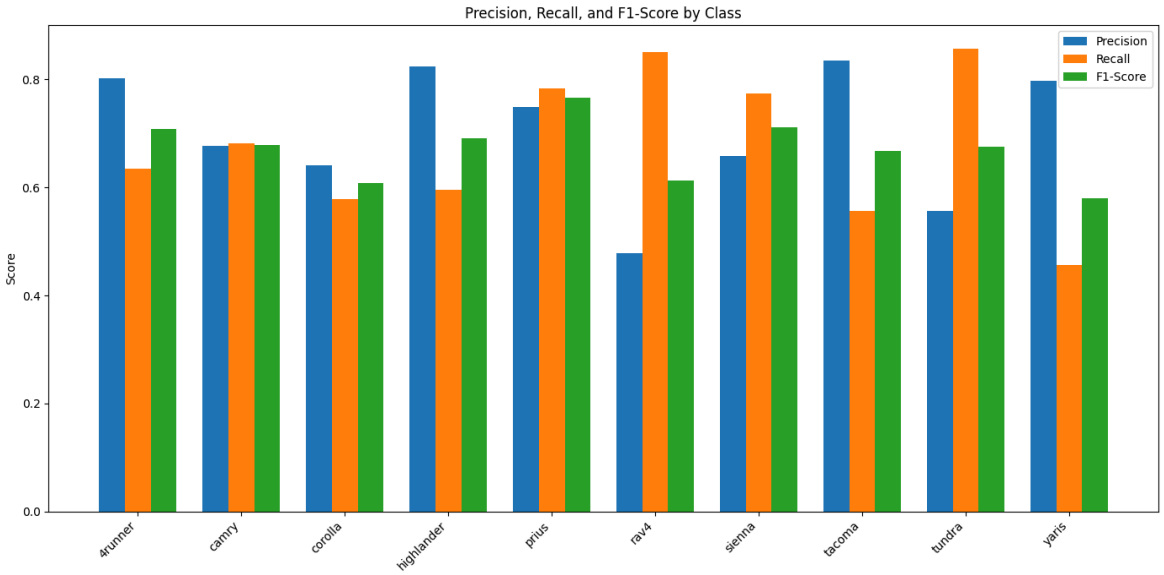
کلاس‌هایی که به خوبی طبقه‌بندی شده‌اند: برخی کلاس‌ها عملکرد نسبتاً قوی‌ای با تعداد بالا روی قطر اصلی ماتریس (پیش‌بینی‌های صحیح) در مقایسه با خانه‌های خارج از قطر در همان سطر/ستون نشان می‌دهند. به عنوان مثال:

- کلاس Prius با 326 پیش بینی صحیح
- کلاس RAV4 با 304 پیش بینی صحیح
- کلاس Camry با 306 پیش بینی صحیح
- کلاس Tacoma با 293 پیش بینی صحیح

کلاس‌هایی با اشتباه طبقه‌بندی بیشتر: سایر کلاس‌ها فعالیت بیشتری در خانه‌های خارج از قطر نشان می‌دهند که بیانگر اشتباهات طبقه‌بندی مکرر است.

- به نظر می‌رسد Yaris به طور خاص چالش‌برانگیز است، به طوری که بسیاری از خودروهای پاریس واقعی به اشتباه به عنوان مدل‌های دیگر طبقه‌بندی شده‌اند (مثلاً 44 مورد به عنوان Prius، 36 مورد به عنوان Corolla). مقدار روی قطر اصلی آن (154) نسبت به ساپورت (تعداد نمونه‌های واقعی این کلاس) نسبتاً پایین است.
- Tundra تعداد بالایی پیش‌بینی صحیح دارد (355 مورد)، اما همچنین مقداری درهم‌ریختگی نشان می‌دهد، به ویژه اینکه به اشتباه به عنوان Tacoma طبقه‌بندی شده است (160 مورد).
- Corrola نیز تعداد قابل توجهی اشتباه طبقه‌بندی دارد (مثلاً 61 مورد به عنوان Camry و 63 مورد به عنوان RAV4).
- RAV4 در حالی که مقدار بالایی روی قطر اصلی دارد، به نظر می‌رسد هدف رایج برای اشتباه طبقه‌بندی شدن از سوی کلاس‌های دیگر است (مثلاً 64 مورد Corolla واقعی به اشتباه به عنوان RAV4 طبقه‌بندی شده‌اند).

تصویر 1.3: نمودار دقت، بازخوانی و F1 مدل VGG16 در هر کلاس



گزارش طبقه‌بندی جزئیات عددی که مشاهدات ماتریس درهم‌ریختگی را تأیید می‌کنند، ارائه می‌دهد:

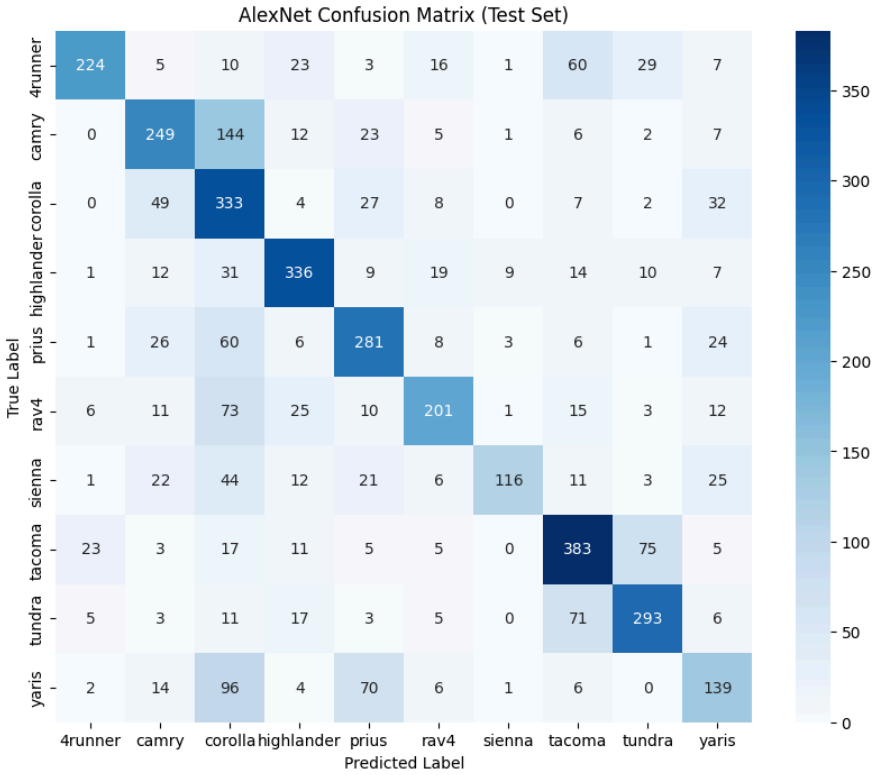
- **دقت بالا (High Precision):** کلاس‌هایی مانند Tacoma و Highlander و 4Runner و Yaris با دقت های به ترتیب 0.83، 0.82، 0.80، 0.80 دقت بالایی دارند. این به این معنی است که وقتی مدل یکی از این کلاس‌ها را پیش‌بینی می‌کند، غالبا درست است. (هرچند برای Yaris، به مقدار Recall دقت کنید).
- **بازیابی بالا (High Recall):** کلاس‌هایی مانند RAV4 و Tundra و Sienna با بازیابی به ترتیب 0.85، 0.86، 0.77 بازیابی بالایی دارند. این به این معنی است که مدل در پیدا کردن بیشتر نمونه‌های واقعی از این کلاس‌ها خوب عمل می‌کند.

2.4- مدل AlexNet (با Transfer Learning)

برای این مدل نیز، روندی مشابه با VGG16 انجام میشود؛ ابتدا لایه ای مناسب به انتهای مدل اضافه و لایه های زیرین، فریز شده و فرایند fine-tuning آغاز میشود.

سپس، بعد از گرفتن تست از مدل، داریم: (**Accuracy در مجموعه تست:** 63%)

تصویر 1.4: ماتریس آشفتگی مدل AlexNet

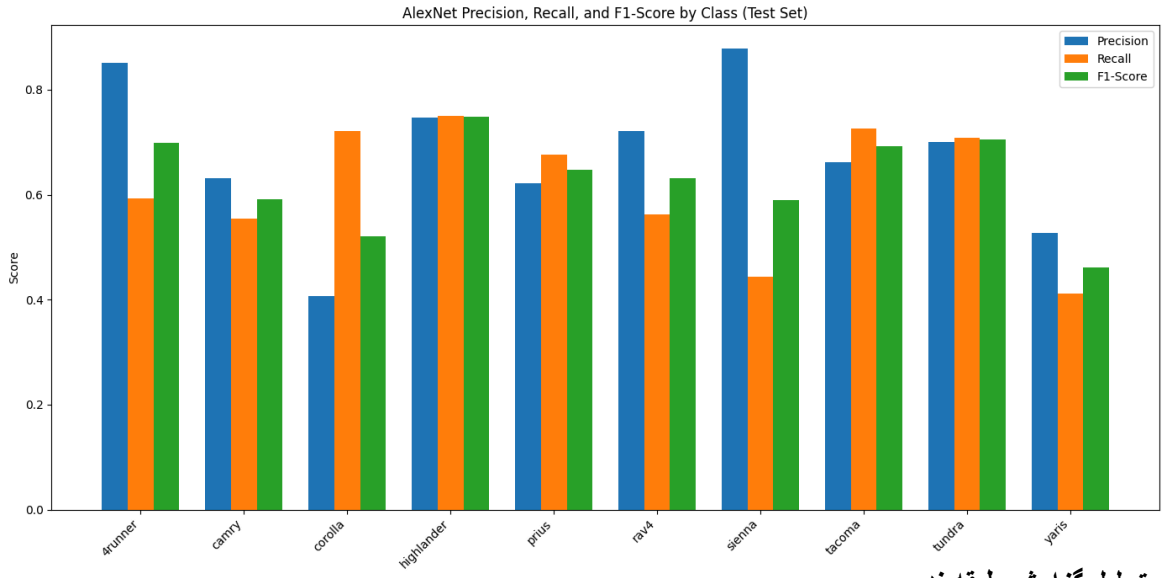


تحلیل ماتریس درهم‌ریختگی (Confusion Matrix):

- **کلاس‌هایی که به خوبی طبقه‌بندی شده‌اند:** مشابه VGG16، برخی کلاس‌ها عملکرد نسبتاً خوبی با مقادیر بالاتر در امتداد قطر اصلی نشان می‌دهند:
  - کلاس Corolla با 333 پیش‌بینی صحیح - جالب است که این مقدار برای Corolla بالاتر از VGG16 است.
  - کلاس Highlander با 336 پیش‌بینی صحیح
  - کلاس Tacoma با 383 پیش‌بینی صحیح - این مقدار کاملاً بالاست.
  - کلاس Tundra با 293 پیش‌بینی صحیح
  - کلاس Prius با 281 پیش‌بینی صحیح
- **کلاس‌هایی با اشتباه طبقه‌بندی بیشتر:**
  - کلاس Yaris همچنان چالش‌برانگیز است (139 پیش‌بینی صحیح) و درهم‌ریختگی با کلاس Corolla (96 مورد) و Prius (70 مورد) نشان می‌دهد. مقدار روی قطر اصلی آن نسبت به سایرین پایین است.

- کلاس Camry در هم‌ریختگی قابل توجهی نشان می‌دهد (مثلاً 144 مورد به اشتباه به عنوان Corolla طبقه‌بندی شده‌اند).
- RAV4 نیز اشتباهات طبقه‌بندی زیادی دارد (مثلاً 73 مورد به عنوان Corolla، 25 مورد به عنوان Highlander).
- Sienna تعداد پیش‌بینی‌های صحیح کمی (116 مورد) در مقایسه با ساپورت خود (261 مورد) دارد که نشان می‌دهد مکرراً به اشتباه طبقه‌بندی می‌شود.

تصویر 1.5: نمودار دقت، بازخوانی و F1 مدل AlexNet در هر کلاس



تحلیل گزارش طبقه‌بندی:

دقت بالا (High Precision): کلاس‌های Sienna و 4Runner و Highlander و RAV4 با دقت بالای به ترتیب 0.88، 0.85، 0.75، 0.72 هستند. وقتی AlexNet این کلاس‌ها را پیش‌بینی می‌کند، اغلب درست است.

بازیابی بالا (High Recall): کلاس‌های Tacoma و Corolla و Tundra با بازیابی بالا به ترتیب 0.73، 0.72، 0.71 هستند. مدل در پیدا کردن نمونه‌های واقعی این کلاس‌ها خوب عمل می‌کند.

کلاس‌هایی با F1-Score بالا:

- کلاس Highlander با 0.75
- کلاس 4Runner با 0.70
- کلاس Tacoma با 0.69
- کلاس Tundra با 0.70

مدل AlexNet که از ابتدا آموزش داده شده است به دقت کلی 63% دست می‌یابد که کمی پایین‌تر از مدل VGG16 با یادگیری انتقالی (67%) است. این مدل در طبقه‌بندی کلاس‌هایی مانند Tacoma، Highlander و Tundra نقاط قوتی از خود نشان می‌دهد. با این حال، با Yaris و Sienna (بازیابی پایین) و همچنین برای Corolla (دقت پایین) و Camry که درهم‌ریختگی قابل توجهی نشان می‌دهند، به طور قابل توجهی دچار چالش است. به نظر می‌رسد آموزش از ابتدا برای AlexNet با این حجم مجموعه داده، کمتر از استفاده از ویژگی‌های از پیش آموزش‌دیده VGG16 مؤثر بوده است.

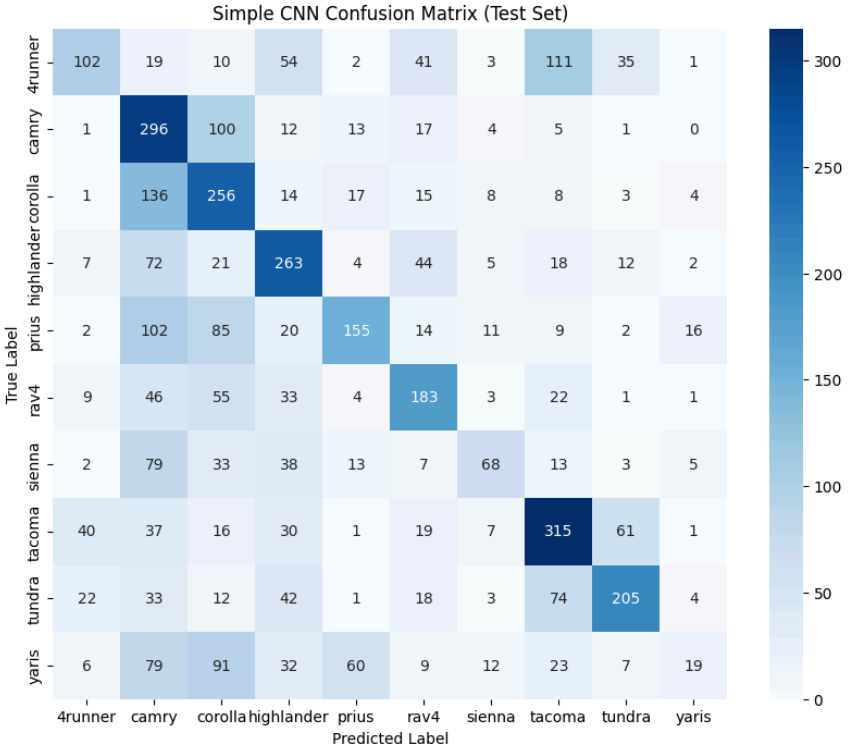
2.5- مدل CNN

علاوه بر معماری‌های شناخته شده، یک شبکه عصبی پیچشی (CNN) ساده‌تر و سفارشی نیز برای این پروژه طراحی و پیاده‌سازی شد. این مدل با داشتن تعداد لایه‌ها و پارامترهای کمتر در مقایسه با VGG16 یا AlexNet، به منظور بررسی عملکرد یک مدل پایه‌ای‌تر و کم‌حجم‌تر در این وظیفه طبقه‌بندی ساخته شد. این مدل نیز به طور کامل و از ابتدا با استفاده از داده‌های آماده شده آموزش داده شد تا توانایی آن در یادگیری مستقیم از تصاویر بررسی شود.

1. لایه Conv با 32 فیلتر
2. لایه Max pooling
3. لایه Conv با 64 فیلتر
4. لایه Max pooling
5. لایه Conv با 128 فیلتر
6. لایه Max pooling
7. لایه Fully-connected با 128 نورون
8. لایه Fully-connected با 10 نورون
9. لایه خروجی با تابع فعالسازی ReLU

بعد از آموزش، نتایج به دست آمده به شکل زیر بوده اند: (**Accuracy در مجموعه آزمایشی:** 45.98%)

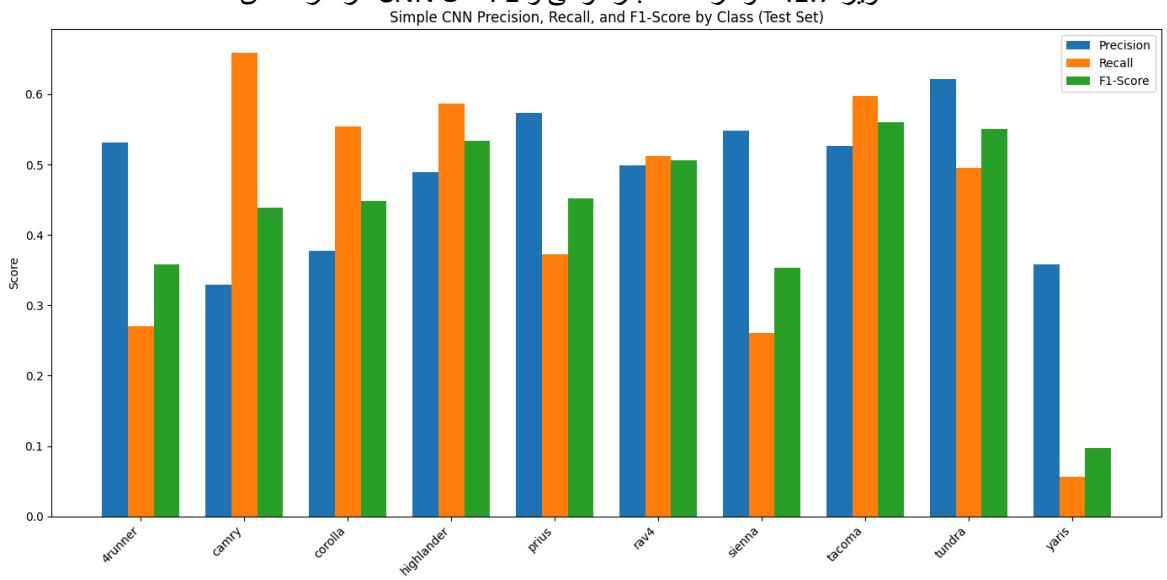
تصویر 1.6: ماتریس آشفتگی مدل CNN



تحلیل ماتریس درهم‌ریختگی (Confusion Matrix):

- **کلاس‌هایی که به نسبت بهتر طبقه‌بندی شده‌اند:** حتی کلاس‌هایی که در این مدل ساده‌تر عملکرد بهتری دارند، مقادیر کمتری روی قطر اصلی نسبت به شبکه‌های عمیق‌تر دارند.
  - کلاس Camry با 296 پیش بینی صحیح
  - کلاس Tacoma با 315 پیش بینی صحیح
  - کلاس Tundra با 205 پیش بینی صحیح
  - کلاس highlander با 263 پیش بینی صحیح
  - کلاس Corolla با 256 پیش بینی صحیح
- **کلاس‌هایی با اشتباه طبقه‌بندی قابل توجه:** بیشتر کلاس‌ها درجه بالایی از درهم‌ریختگی را نشان می‌دهند.
  - کلاس Yaris به طور بسیار ضعیفی طبقه‌بندی شده است (فقط 6 پیش‌بینی صحیح)، به شدت با کلاس Corolla ا(91 مورد) و Prius ا(60 مورد) و سایر کلاس‌ها درهم‌ریخته است. مقدار روی قطر اصلی آن نسبت به ساپورت (338) بسیار پایین است.
  - 4Runner نیز مقدار بسیار پایینی روی قطر اصلی دارد (102)، با Tacoma ا(111 مورد) و Camry ا(19 مورد) درهم‌ریخته است.
  - Sienna نیز مشکل دارد (68 پیش‌بینی صحیح در مقابل ساپورت 261).
  - بسیاری از کلاس‌ها مکرراً به اشتباه در دسته‌های Highlander و Tacoma و Camry و Corolla و Tundra طبقه‌بندی می‌شوند که به نظر می‌رسد متداول‌ترین خروجی‌های پیش‌بینی هستند، صرف نظر از کلاس واقعی.

تصویر 1.7: نمودار دقت، بازخوانی و F1 مدل CNN در هر کلاس



تحلیل گزارش طبقه‌بندی:

به طور کلی مقادیر پایین‌تر دقت، بازیابی و F1-Score: تمام معیارها در کل به طور کلی نسبت به مدل‌های عمیق‌تر پایین‌تر هستند.

کلاس‌هایی با F1-Score بالاتر (به نسبت):

- کلاس Tacoma با (F1: 0.56)
- کلاس Highlander با (F1: 0.53)
- کلاس RAV4 با (F1: 0.51)

- کلاس Tundra با (F1: 0.55)
- کلاس Corolla با (F1: 0.45)
- کلاس Camry با (F1: 0.44)

کلاس‌هایی با F1-Score بسیار پایین:

- Yaris دقت (0.36)، بازیابی (0.06) و (F1-Score (0.10 بسیار پایینی دارد. مدل به سختی هیچ خودروی Yaris را شناسایی نمی‌کند.
- Sienna نیز بازیابی پایین (0.26) و F1-Score پایین (0.35) دارد.
- 4Runner بازیابی پایین (0.27) و F1-Score پایین (0.36) دارد.
- Prius بازیابی پایین (0.37) و F1-Score پایین‌تر (0.45) دارد.

عدم توازن:

- کلاس Camry دقت بسیار پایین (0.33) اما بازیابی نسبتاً بالاتر (0.66) دارد، به این معنی که مدل اغلب Camry را پیش‌بینی می‌کند، اما بسیاری از آن پیش‌بینی‌ها اشتباه هستند.
- Prius و 4Runner و Sienna و Yaris بازیابی بسیار پایینی دارند، که نشان می‌دهد مدل بخش بزرگی از نمونه‌های واقعی این کلاس‌ها را از دست می‌دهد.

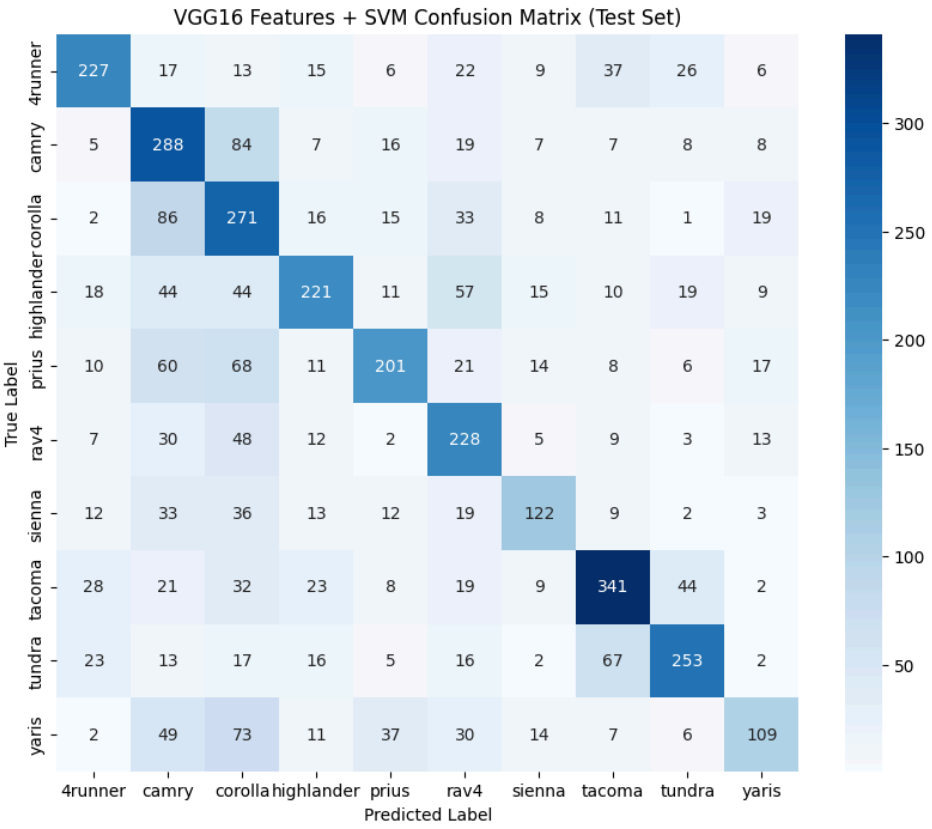
مدل CNN ساده که از ابتدا آموزش دیده است، به دقت کلی به طور قابل توجهی پایین‌تری (46%) نسبت به مدل‌های عمیق‌تر VGG16 و AlexNet دست یافته است. این مدل به طور قابل ملاحظه‌ای با بسیاری از کلاس‌ها، به ویژه چهار مدل ماشینی که بازیابی و F1-Score بسیار پایینی دارند، مشکل دارد. اگرچه به دلیل پارامترهای کمتر سریع‌تر آموزش می‌بیند، اما پیچیدگی محدود آن منجر به عملکرد بسیار ضعیف‌تری در این وظیفه طبقه‌بندی تصویر در مقایسه با استفاده از معماری‌های عمیق‌تر یا یادگیری انتقالی می‌شود.

2.6- مدل VGG16 + Linear SVM

در این رویکرد ترکیبی، از مدل VGG16 از پیش آموزش‌دیده نه به عنوان یک طبقه‌بندی‌کننده نهایی، بلکه به عنوان یک استخراج‌کننده ویژگی استفاده شد. لایه‌های پیچشی VGG16 برای پردازش تصاویر استفاده شدند و بردارهای ویژگی سطح بالا از خروجی آن‌ها استخراج شدند. این بردارهای ویژگی سپس به عنوان ورودی برای آموزش یک طبقه‌بندی‌کننده سنتی‌تر، یعنی ماشین بردار پشتیبان (SVM)، استفاده شدند. این روش مرحله یادگیری ویژگی‌های پیچیده را از طبقه‌بندی نهایی جدا می‌کند و امکان استفاده از یک الگوریتم طبقه‌بندی متفاوت بر روی نمایش‌های غنی بصری استخراج شده توسط یک شبکه عمیق را فراهم می‌آورد.

نتایج بدست آمده در طبقه بندی از این فرآیند به شرح زیر است:

تصویر 1.8: ماتریس آشفتگی مدل VGG + SVM



تحلیل ماتریس درهم‌ریختگی (Confusion Matrix):

با نگاه کردن به نقش حرارتی (heatmap):

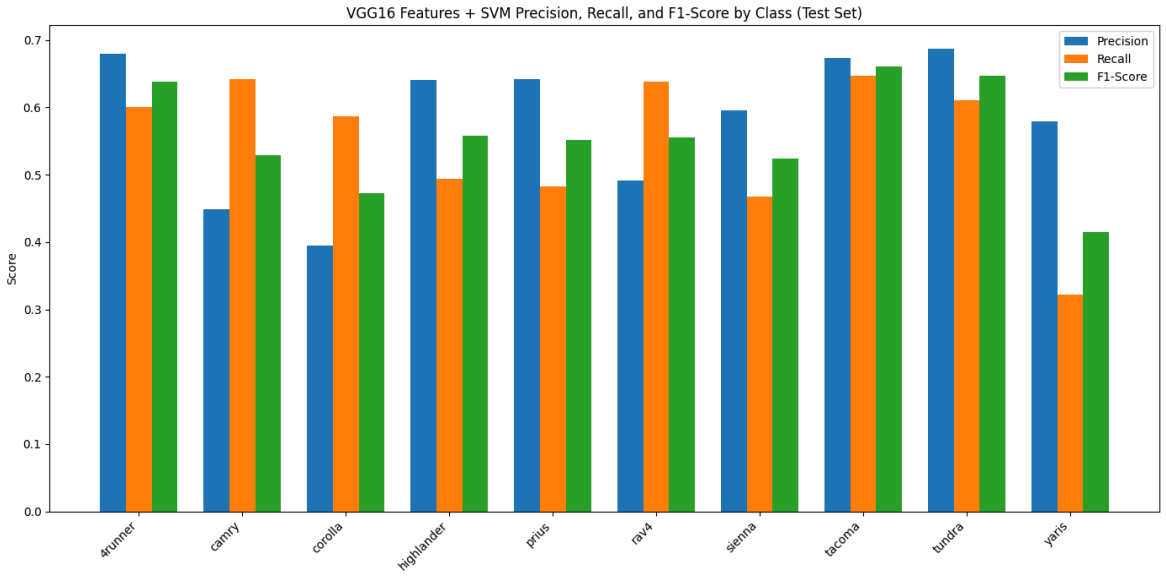
کلاس‌هایی که به نسبت بهتر طبقه‌بندی شده‌اند:

- کلاس Camry با 288 پیش‌بینی صحیح
- کلاس Tacoma با 341 پیش‌بینی صحیح - این یک نقطه قوت برای این مدل است.
- کلاس Corolla با 271 پیش‌بینی صحیح
- کلاس RAV4 با 228 پیش‌بینی صحیح
- کلاس 4Runner با 227 پیش‌بینی صحیح

کلاس‌هایی با اشتباه طبقه‌بندی قابل توجه:

- Yaris همچنان بسیار ضعیف طبقه‌بندی می‌شود (فقط 109 پیش‌بینی صحیح)، درهم‌ریختگی قابل توجهی با Corolla با 73 مورد و Camry با 49 مورد نشان می‌دهد.
- Highlander تعداد قابل توجهی اشتباه طبقه‌بندی نشان می‌دهد، با Corolla با 44 مورد و RAV4 با 57 مورد درهم‌ریخته است.
- Prius نیز مکرراً به اشتباه طبقه‌بندی می‌شود (مثلاً 60 مورد به عنوان Camry و 68 مورد به عنوان Corolla).
- Sienna تعداد پیش‌بینی‌های صحیح کمی (122 مورد) در مقایسه با ساپورت خود (261 مورد) دارد.

تصویر 1.9: نمودار دقت، بازخوانی و F1 مدل VGG + SVM برای هر کلاس



تحلیل گزارش طبقه‌بندی:

به طور کلی مقادیر متوسط دقت و بازیابی: معیارها به طور کلی در محدوده متوسطی نسبت به مدل‌های قبلی قرار دارند.

کلاس‌هایی با F1-Score بالاتر (به نسبت):

- Tacoma با (F1: 0.66) - این نقطه قوت آن را تایید می‌کند.
- 4Runner با (F1: 0.64)
- Tundra با (F1: 0.65)
- Highlander با (F1: 0.56)
- RAV4 با (F1: 0.56)
- Camry با (F1: 0.53)

کلاس‌هایی با F1-Score پایین‌تر:

- Yaris دقت (0.58)، بازیابی (0.32) و F1-Score ا(0.41) پایینی دارد. در شناسایی خودروهای Yaris به طور قابل توجهی مشکل دارد و بخش زیادی از نمونه‌های واقعی را از دست می‌دهد.
- Sienna بازیابی (0.47) و F1-Score ا(0.52) پایین‌تری دارد.
- Corolla دقت (0.40) و F1-Score ا(0.47) پایین‌تری دارد.
- Prius بازیابی (0.48) و F1-Score ا(0.55) پایین‌تری دارد.

2.7- تحلیل مقایسه‌ای:

Avg F1-Score	Avg Recall	Avg Precision	Accuracy	
0.67	0.67	0.71	0.6701	VGG16
0.63	0.63	0.66	0.63	AlexNet
0.44	0.46	0.48	0.4598	CNN
0.56	0.56	0.58	0.56	VGG+SVM

با توجه به جدول و تحلیل‌های قبلی، می‌توان نکات زیر را ذکر کرد:

1. **بهترین عملکرد کلی:** مدل VGG16 با یادگیری انتقالی به وضوح بهترین عملکرد کلی را در میان مدل‌های آزمایش شده دارد. این مدل بالاترین دقت کلی (67.01%) و بالاترین میانگین‌های دقت، بازیابی و F1-Score (هم Macro و هم Weighted Avg) را کسب کرده است. این نشان‌دهنده اثربخشی بسیار بالای استفاده از ویژگی‌های غنی بصری که توسط VGG16 از پیش بر روی داده‌های گسترده ImageNet یاد گرفته شده‌اند، به همراه آموزش لایه‌های نهایی برای وظیفه خاص ما است.

2. **عملکرد آموزش از ابتدا:** مدل AlexNet آموزش دیده از ابتدا عملکردی بهتر از CNN ساده و Linear SVM دارد، اما پایین‌تر از VGG16 با یادگیری انتقالی قرار می‌گیرد. این مدل به دقت کلی 63% دست یافته است. اگرچه AlexNet یک معماری قدرتمند است، آموزش آن از ابتدا با حجم مجموعه داده موجود در مقایسه با بهره‌گیری از یادگیری انتقالی VGG16 چالش‌برانگیزتر بوده و منجر به عملکرد پایین‌تری شده است. AlexNet در برخی کلاس‌ها مانند Tacoma و Highlander نسبتاً خوب عمل کرده است.

3. **عملکرد CNN ساده:** همانطور که انتظار می‌رفت، مدل CNN ساده آموزش دیده از ابتدا ضعیف‌ترین عملکرد را با دقت کلی تنها 45.98% نشان داده است. سادگی معماری آن و تعداد کمتر پارامترها باعث شده که نتواند ویژگی‌های پیچیده لازم برای تمایز قائل شدن بین مدل‌های مختلف خودرو را به خوبی یاد بگیرد. این مدل در طبقه‌بندی بیشتر کلاس‌ها دچار مشکل بوده و درهم‌ریختگی بالایی را در ماتریس درهم‌ریختگی نشان داده است.

4. **عملکرد استخراج ویژگی + SVM:** رویکرد VGG16 Features + Linear SVM عملکردی متوسط دارد و با دقت کلی 56%، بهتر از CNN ساده اما پایین‌تر از هر دو مدل عمیق‌تر (VGG16 TL و AlexNet FS) عمل کرده است. این نشان می‌دهد که ویژگی‌های استخراج شده توسط VGG16 قدرتمند هستند، اما یک مرز تصمیم‌گیری خطی توسط Linear SVM در فضای ویژگی با ابعاد بالا، برای جداسازی بهینه همه کلاس‌ها کافی نیست. این رویکرد در طبقه‌بندی کلاس‌هایی مانند Tacoma نسبتاً موفق بوده است. مزیت اصلی این روش، سرعت بالای آموزش SVM پس از استخراج ویژگی است، که امکان آزمایش سریع با پارامترهای مختلف SVM را فراهم می‌کند.

2.8- جمع‌بندی:

یادگیری انتقالی با استفاده از مدل‌های عمیق از پیش آموزش‌دیده مانند VGG16، کارآمدترین رویکرد برای این وظیفه طبقه‌بندی با توجه به حجم مجموعه داده ما بوده و بهترین عملکرد را ارائه داده است. آموزش شبکه‌های عمیق‌تر مانند AlexNet از ابتدا نیازمند داده بسیار بیشتری است و در این مورد عملکرد پایین‌تری نسبت به یادگیری انتقالی داشته است. استفاده از یک CNN ساده به دلیل پیچیدگی کم، برای این وظیفه طبقه‌بندی دقیق کافی نبوده است. در نهایت، استفاده از ویژگی‌های استخراج شده توسط VGG16 همراه با یک **SVM خطی**، عملکردی بهتر از CNN ساده داشته، اما نشان داده است که **یک طبقه‌بندی‌کننده غیرخطی (مانند SVC با کرنل RBF، در صورت پیاده‌سازی و آزمایش آن)** ممکن است برای بهره‌برداری کامل از این ویژگی‌های پیچیده مورد نیاز باشد و بتواند عملکرد را بهبود بخشد (اگرچه آموزش آن می‌تواند بسیار پرهزینه‌تر باشد).