



مبانی هوش محاسباتی (پروژه چهارم)

کیارش آستانبوس ۴۰۰۱۲۶۲۵۷۰
محمد رضا نادری ۴۰۰۱۲۶۲۳۸۶
مهر ۱۴۰۳

فهرست مطالب

۲	(۱) فاز ۱
۲	Preprocess ۱.۱(۱)
۲	انتخاب معماری ۲.۱(۱)
۳	نمودار ها ۳.۱(۱)
۴	ارزیابی ۴.۱(۱)
۴	بررسی پارامتر ها ۵.۱(۱)
۶	(۲) فاز ۲
۶	۱. کاهش پارامتر ۱.۱(۲)
۶	۲. نمودار ها ۲.۱(۲)
۷	۳. ارزیابی ۳.۱(۲)
۷	۴. بررسی پارامتر ها ۴.۱(۲)
۹	(۳) فاز ۳
۹	۱. متدهای بررسی شده ۱.۱(۳)
۹	۲. فاز ۱ ۲.۱(۳)
۹	۱.۱. نمودار ها ۱.۱.۱(۳)
۱۰	۲.۱. ارزیابی ۲.۱.۱(۳)
۱۰	۳.۱. بررسی پارامتر ها ۳.۱.۱(۳)
۱۱	۱.۲. فاز ۲ ۱.۲.۱(۳)
۱۱	۱.۱. نمودار ها ۱.۱.۱(۳)
۱۲	۲.۱. ارزیابی ۲.۱.۱(۳)
۱۲	۳.۱. بررسی پارامتر ها ۳.۱.۱(۳)
۱۴	(۴) فاز ۴
۱۴	۱ Dataset ۱.۱(۴)
۱۴	۱ Part ۱.۱.۱(۴)
۱۵	۲ Part ۲.۱.۱(۴)
۱۷	۲ Dataset ۲.۱.۱(۴)
۱۷	۱ Part ۱.۲.۱(۴)
۱۸	۲ and ۱ Part ۲.۲.۱(۴)

(۱) فاز ۱

Preprocess ۱.(۱)

برای فاز ۱ و فاز ۲ پیش پردازش یکسانی انتخاب شده است که بعد از تقسیم داده ها، صرفا ریسایز و کراپ می شوند. همچنین میانگین و انحراف از معیار رنگ کل دیتاست محاسبه شده و داده ها بر اساس آن نرمالایز می شوند که در فاز ۴ خروجی بهتری داشته باشیم. البته به دلیل بالانس نبودن تعداد کلاس ها، برای اینکه مدل در هر بچ تعداد یکسانی از کلاس هارا ببیند و به سمت یک کلاس خاص بایاس نشود، Train loader را با پارامتر مناسب تنظیم کردیم که در هر بچ به یک اندازه از هر کلاس داده دریافت کند. همچنین بعد از آزمایش و خطا، و با توجه به اندازه دیتاست، بچ سایز ۳۲ انتخاب شد. که باعث آموزش خوبی شد هم زمان ترین کم بود.

۱.۲. انتخاب معماری

در ابتدا معماری خیلی ساده دارای ۲ لایه کانولوشن و ۴ لایه کانولوشن انتخاب شده بود که به خوبی اورفیت می شد ولی

دقت ولیدیشن ۵۰ درصد داشت.

این مقادیر دقต با بررسی مقادیر مختلف هایپرپارامتر ها نیز بهبود پیدا نکرد که نشان می داد معماری ضعیفی انتخاب شده است.

با افزایش لایه های کانولوشن تا ۵ الی ۶ لایه توانستیم به دقت ۸۰ درصد ولیدیشن و ۹۹ ترین برسیم ولی مشکلی که وجود داشت این بود حدود ۱ میلیون پارامتر در لایه های کانولوشن وجود داشت و تعداد پارامتر های مدل برای فاز ۲ به زیر ۴۰۰،۰۰۰ نمی رسید. همچنین معماری هایی که بیشتر از ۶ لایه داشتن عملکرد بدتری نسبت به معماری هایی با لایه های کمتر داشتند چون پیچیدگی مدل برای این دیتاست بالا بود.

```
class MinimalCNN(nn.Module):
    def __init__(self):
        super(MinimalCNN, self).__init__()
        self.features = nn.Sequential(
            # First Conv Layer
            nn.Conv2d(3, 40, kernel_size=3, padding=1), # 128x128x3 -> 128x128x32
            nn.BatchNorm2d(40),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2), # 128x128x32 -> 64x64x32

            # Second Conv Layer
            nn.Conv2d(40, 80, kernel_size=3, padding=1), # 64x64x32 -> 64x64x64
            nn.BatchNorm2d(80),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2), # 64x64x64 -> 32x32x64

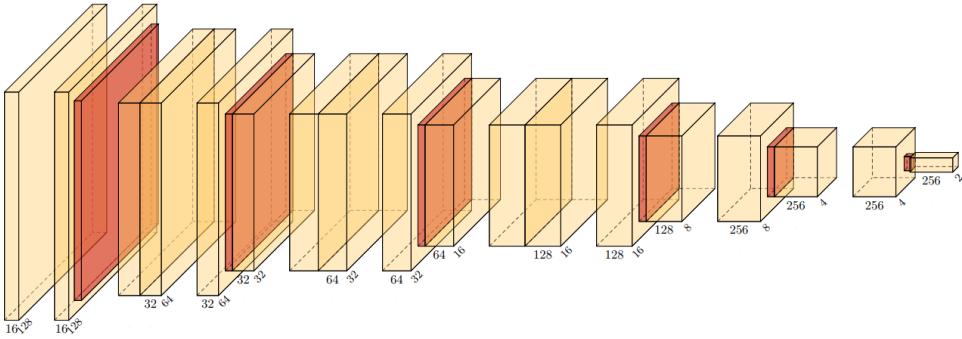
            # Third Conv Layer
            nn.Conv2d(80, 160, kernel_size=3, padding=1), # 32x32x64 -> 32x32x128
            nn.BatchNorm2d(160),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2), # 32x32x128 -> 16x16x128

            # Fourth Conv Layer (Optional to increase parameters further)
            nn.Conv2d(160, 320, kernel_size=3, padding=1), # 16x16x128 -> 16x16x256
            nn.BatchNorm2d(320),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2), # 16x16x256 -> 8x8x256
        )

        # Flatten and pass through fully connected layers
        self.classifier = nn.Sequential(
            nn.Flatten(),
            nn.Linear(32 * 8 * 8, 512), # Increased size for more parameters
            nn.ReLU(),
            nn.Linear(512, 10) # Output layer for 10 classes
        )

    def forward(self, x):
        x = self.features(x)
        x = self.classifier(x)
        return x
```

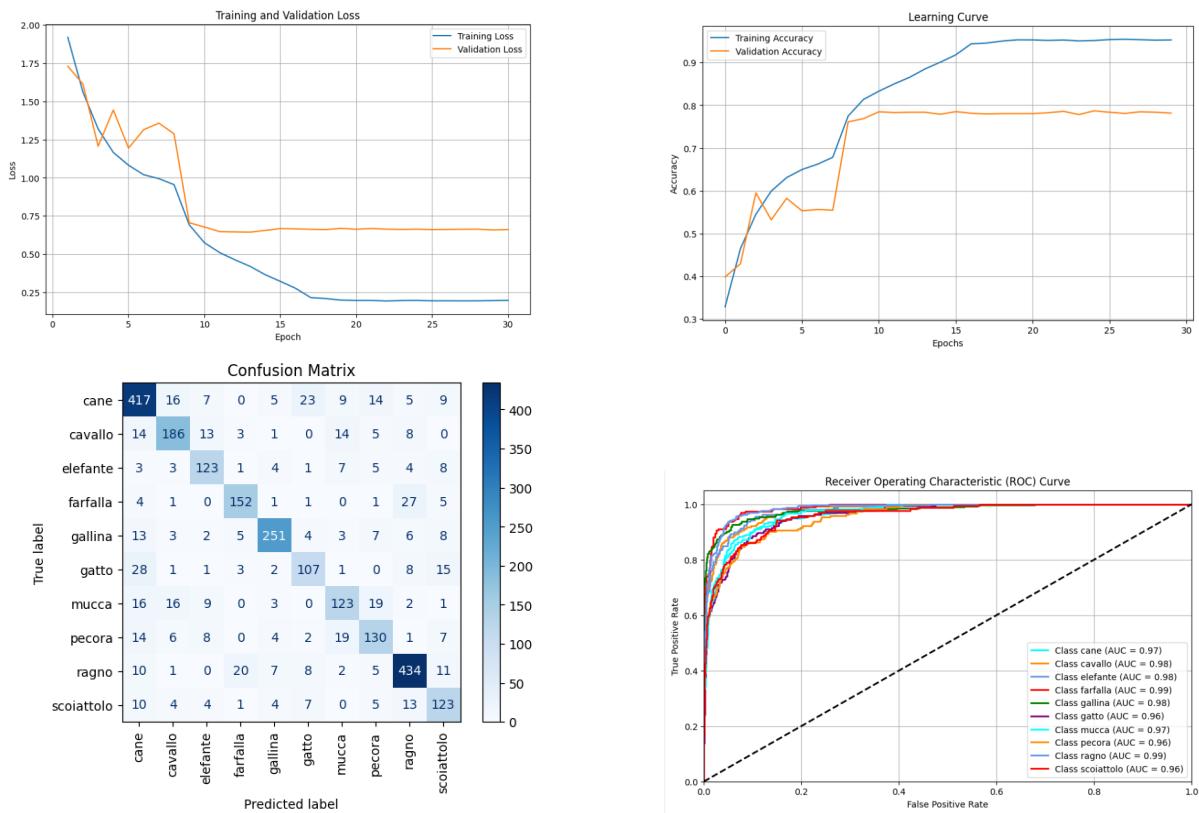
پس در مقالات و منابع موجود جستجو کردیم که به اندازه کافی پیچیده باشد تا در فاز های بعد دقت بالایی بدهد و همچنین تعداد پارامتر های کمی در لایه های کانولوشن داشته باشد.



شکل ۱:

در نهایت بعد از تست‌های فراوان مدل بالا با الهام از Depth-wise PointWise ساخته شد که تشکیل شده است از لایه‌های کاهش می‌یابد. در این مدل ما در لایه آخر ۲۵۶ فیلتر داریم و همچنین با ۶ لایه پولینگ ابعاد از 2×2 به 128×128 کاهش می‌یابد. در این مدل بجای استفاده از stride ۲، مکس پولینگ استفاده شده است. و بعد از هر بار که تعداد فیلترها افزایش می‌یابد، یک لایه با همان تعداد فیلتر داریم تا مدل بتواند به خوبی ویژگی هارا استخراج کن. همچنین در قسمت classifier که فوپی کانکتد هستند از ۱۰۲۴ نورون استفاده شده که در نهایت به ۱۰ نورون برای طبقه‌بندی وصل می‌شوند. این مدل ۱۲۵۵۹۶۲ پارامتر دارد که آن در قسمت کلسیفایر و فوپی کانکتد هستند.

۳.۱ نمودارها



برای نمودار دقیق و لاس، میتوان دید که فاصله ولیدیشن و ترین بسیار زیاد است. برای ولیدیشن بعد از حدود ۱۰ ایپاک و برای تست بعد از ۱۵ ایپاک، مشتق نمودار تقریباً صفر است و تغییری ایجاد نمی‌شود. دلیل این که ما در ابتدا خط شکسته زیاد داریم این است که لرنینگ ریت بزرگی در ابتداء داریم و قدم‌ها بزرگ است، به مرور که لرنینگ ریت کوچک‌تر می‌شود،

نسبت تغییرات خط ها هم کمتر میشود. برای ماتریس گمراهی میتوانیم بینیم قطر اصلی پرزنگ است یعنی کلاس ها نسبتاً درست پیشبینی شده اند اما برای کلاس هایی که اشتباه پیشبینی شده اند، بیشترین اشتباه مربوط به کلاس Cane است که نمونه های کلاس های دیگر اشتباهها با لیبل این کلاس پرداخت میشند. دلیل این مورد هم این است که داده ها بالанс نبوده و تعداد داده های این کلاس نسبت به کلاس های دیگر بیشتر بوده و مدل کلاس های دیگر را به خوبی یاد نگرفته.

برای نمودار ROC ما نسبت TP به FP را میتوانیم بسنجدیم. با توجه به نمودار، این مدل برای هیچ کلاسی مقدار ۱ برای AUC نگرفته است یعنی کلسیفیکیشن بدون ایراد نبوده که با توجه به دقت مدل قابل پیشبینی است. اما بهترین طبقه بندی این مدل برای کلاس farfalla با auc ۹۹ بوده است و بدترین طبقه بندی برای کلاس های Gatto و scoiattolo با auc ۹۶ بوده است.

۴.۱ ارزیابی



شکل ۲:

ارزیابی مدل :

۷۷ : Score F1 .

۷۷ : Recall .

۷۷ : Precision .

۷۸ : Validation Accuracy .

۹۵ Training: Acuracy .

از ۱۰ عکس نمایش داده شده ۳ عکس به اشتباه پرداخت شده اند که با دقت ۷۷ درصدی ما تقریباً تطابق دارد. همچنین با توجه به اختلاف ۱۸ درصدی بین داده های تست و ترین، میتوان فهمید مدل اورفیت شده است.

۵.۱ بررسی پارامتر ها

بیشترین آزمایش و خطای برای کل پروژه در این فاز انجام شد که نتایج به صورت کامل در کد ذکر شده اند و برای خلاصه بودن داکیومنت فقط چند مورد ذکر خواهد شد.

Bath Size : در ابتدا بچ سایز ۳۲ انتخاب شده بود که نسبت به پیچیدگی معماری های اولیه دقت قابل قبولی داشت. برای معماری های بعدی بچ سایز های ۱۲۸ و ۲۵۶ هم امتحان شد که چون تعداد داده ها کم بود، در نتیجه تعداد دفعاتی که در هر ایپاک پارامتر های مدل آپدیت میشد کم بود و مدل به Generalization خوبی نمیرسید و در طی تست مدل

های مختلف همان بچ سایز ۳۲ و ۶۴ استفاده می‌شد که هم Generalization خوبی داشتند و از نظر زمان اجرایی هم مناسب بودند.

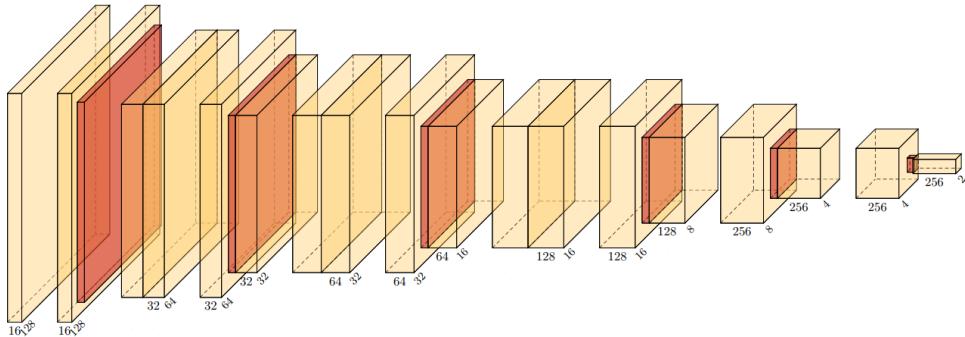
Optimizer : در ابتدا آپتیمایزر Adam استفاده می‌شد که عملکر قابل قبولی داشت. بعد از انتخاب مدلنهایی برای بهبود مدل، طی جستجو در مقالات موجود، آپتیمایز SGD و لرنینگ ریت داینامیک انتخاب شد که در این فاز توانستیم حدود ۳ درصد افزایش دقت داشته باشیم. دلیل این مورد هم است که با توجه به ماهیت Adam ، Adaptive SGD چون تلاش می‌کند از sharp minima دوری کند نتیجه بهتری دارد و همچنین چون لرنینگ ریت داینامیک تنظیم شده است در ابتدا قدم‌های بزرگی به سمت local minima بر می‌داریم و به مرور با کوچک‌تر کردن لرنینگ ریت به local minima نزدیک‌تر می‌شویم.

Learning Rate : این مورد در ابتدا به صورت دیفالت همان ۰.۰۰۱ داشت. در نهایت با آدام نتیجه خوبی داشت. ولی وقتی مقدار بزرگی انتخاب شد مثل ۰.۰۱ یا ۰.۱ چون قدم‌های بزرگی به سمت local minima برمی‌دارد اصلاً قادر به پیدا کردن local minima نبود و عملکرد مدل به شدت ضعیف بود. در نهایت لرنینگ ریت داینامیک با شروع از ۰.۱ داشت شد که اگر loss مدل ۲ پشت سر هم بهبود نمیافتد، لرنینگ ریت نصف می‌شد.

Epoch : ابتدا ایپاک ۲۰ انتخاب شد و ایپاک‌های ۳۰ و ۵۰ نیز امتحان شدند. در نهایت با توجه به مدل و هایپرپارامترهای دیگر، مدل در همان ۲۰ ایپاک به بهترین حالت خود می‌رسید و نیازی به ایپاک بیشتر نبود.
L2 Regularization : دقت ترین این مدل ۹۹ بود که با weight decay آن را کمی کنترل کردیم و حدود ۶ درصد کاهش دقت در ترین و ۳ درصد افزایش دقت در تست و ولیدیشن داشتیم. دلیل این مورد هم این بود که L2 Regularization بر وزن‌های بزرگ یک جریمه اعمال می‌کند و باعث می‌شود مدل بهتر تعیین یابد.

۲) فاز ۲

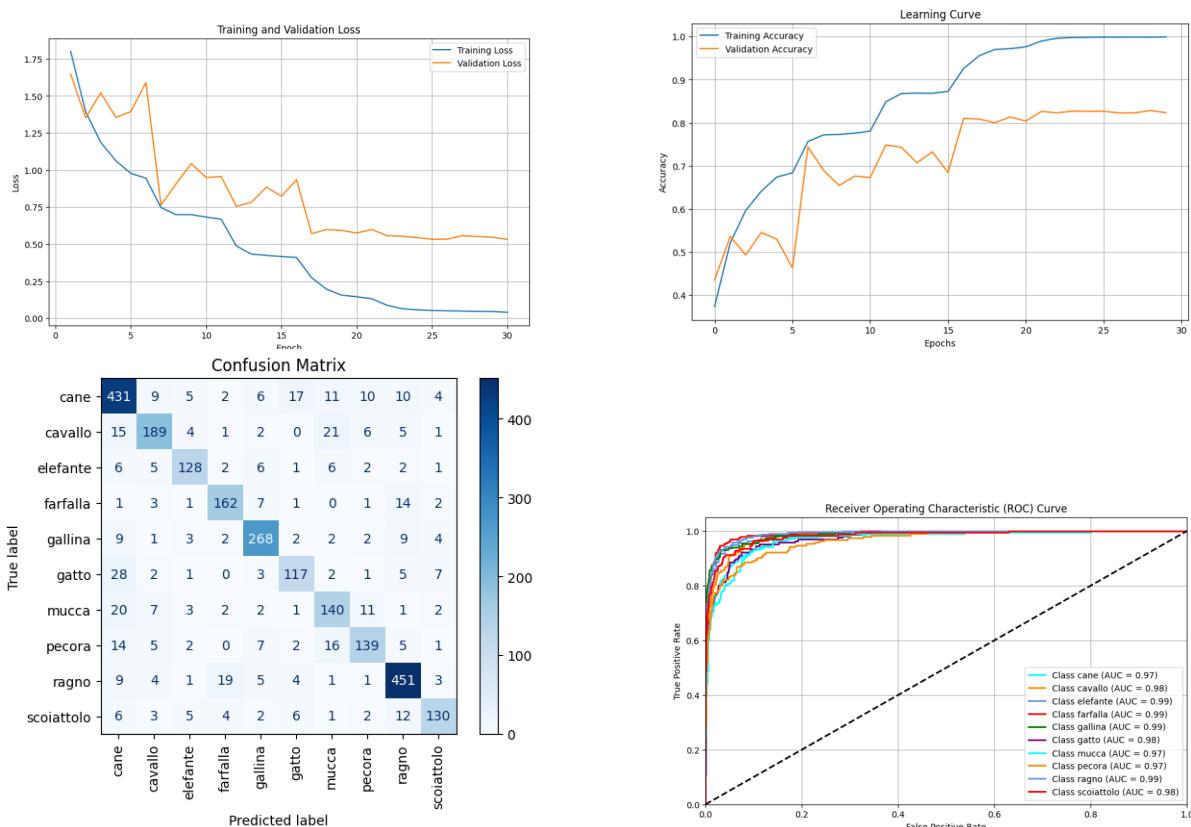
۱.) کاهش پارامتر



شکل ۳:

در این فاز مثل مدل های معروف مانند ResNet، برای کاهش پارامتر ها در لایه فولی کانکتیو، از Global Average Pooling استفاده کردیم که تعداد پارامتر های لایه آخر را به ۲۵۷۰ تا کاهش داد و مدل نهایی ۱۹۸۶۸۲ تا پارامتر دارد.

۲.) نمودار ها



برای نمودار های دقت و لاس، مثل فاز قبل در ابتدا تغییرات بزرگ است و به مرور تغییرات کمتر میشود. و بعد از حدود ۲۰ ایپاک رشد نمودار متوقف شده.

نسبت به فاز قبل چون کمی بهبود داشتیم، نمودار ROC هم بهتر شده. هم طبقه بندی کلاس ها به یکدیگر نزدیک شده و هم کمترین auc برابر ۹۷ است. ولی همچنان بالاترین ۹۹ است. ولی فاز قبل ۲ تا ۹۹ داشتیم ولی در این فاز ۴ تا. تحلیل های ماتریس گمراهی برای فاز قبل در این فاز نیز صدق میکند. ولی میتوان افزود که کلاس های gatto و cane و همچنین کلاس های farfalla و ragno به هم شبیه ترند چون در ماتریس گمراهی بیشترین پیشینی اشتباہ بین این دو جفت کلاس ها بوده است.

۳.۲ ارزیابی



شکل ۴:

ارزیابی مدل :

F1 : Score F1 •

R1 : Recall •

P1 : Precision •

VA : Validation Accuracy •

۹۹ Training: Acuracy •

نسبت به فاز قبل حدود ۳ درصد افزایش دقت داشتیم که دلیل آن این است چون پارامتر های لایه فولی کانکتد را کاهش داده ایم، مدل بهتر تعمیم یافته است و نویز های گرادیان کاهش یافته است. همچنین احتمال رخداد gradient vanishing کاهش یافته است.

۴.۲ بررسی پارامتر ها

در این فاز از هایپر پارامتر های فاز قبل استفاده کردیم که نتایج قابل گرفتیم و نتیجه عملکرد مدل با پارامتر های دیگر به صورت زیر است.

- با نرخ یادگیری ۰.۰۰۱ و بج سایز ۱۲۵ و آپتیمایزر SGD : دقت ۷۶ برای ولیدیشن و ۹۹ برای ترین
- با نرخ یادگیری داینامیک و بج سایز ۶۴ و آپتیمایزر SGD : دقت ۷۸ برای ولیدیشن و ۹۹ برای ترین
- مثل مورد دوم به همراه decay weight : دقت ۸۲ برای ولیدیشن و ۹۹ برای ترین.

دلیل اینکه L_2 Regularization در فاز اول روی دقیق ترین تاثیرگذار بود ولی در فاز دوم تاثیرگذار نداشت این است که مدل اول پارامترهای بسیار بیشتری دارد در نتیجه جرمیمه که به آن اعمال میشود بیشتر است و بهتر در کل شبکه پخش میشود.

۳) فاز ۳

۱.۳) متد های بررسی شده

Data Augmentation : در این قسمت که در پیش پردازش داده ها اعمال شده است روی داده های ترین تغییراتی اعمال کردیم که باعث میشود وقتی یک عکس به مدل داده می شود، هر دفعه با تغییرات زندوم به مدل داده شود تا مدل آن را حفظ نکند. این تغییرات شامل تغییرات در روشنایی تصویر، flip ، چرخش ، تغییر کانتراست، تغییر Saturation و تغییر hue است.

Early Stopping : هنگام ترین مدل، اگر ۵ ایپاک پشت سر هم loss ترین بیشتر از ۰.۰۵٪ بебود نیابد، عملیات ترین متوقف می شود.

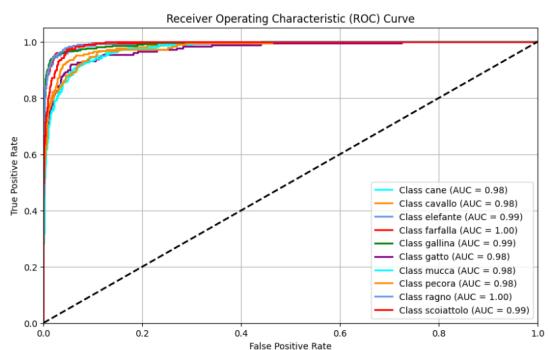
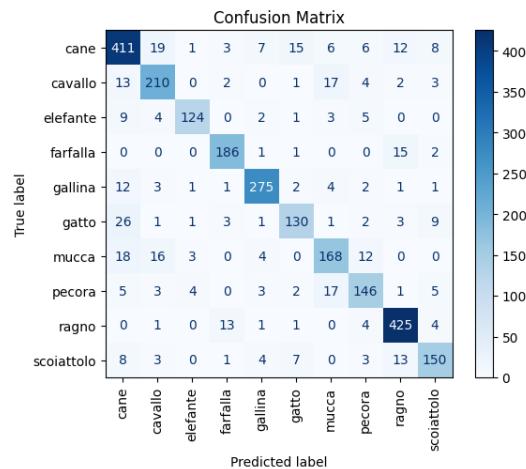
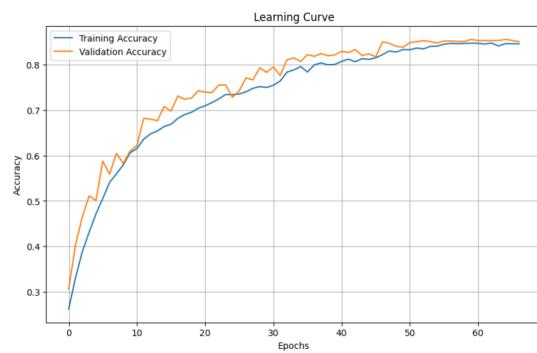
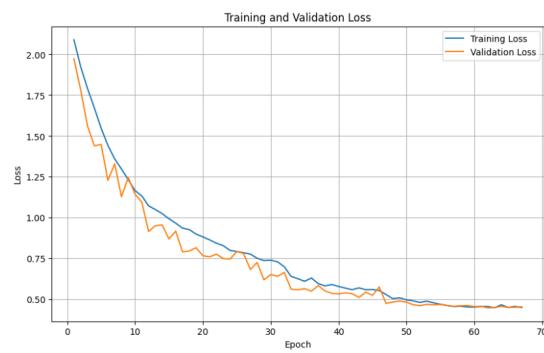
Connection Residual skip : در ابتدا هدف بر پیاده سازی این مورد هم بود و ساختار مدل اجازه میداد تا connection را پیاده ساز یکنیم اما چون اورفیت فقط با موارد دیگر جلوگیری شده بود، این مورد پیاده سازی نشد تا قدرت مدل حفظ شود.

Drop Out : در لایه کلسیفایر برای هر دو مدل از دراپ اوت استفاده شده که برای مدل فاز اول از دو دراپ اوت با شانس ۰.۵ و ۰.۲۵ و همچنین برای مدل فاز دوم از یک دراپ اوت با شانس ۰.۵ استفاده شده است.

L2 Regularization : برای هر دو مدل از Decay Weight با مقدار ۰.۰۰۵٪ استفاده شده است که بهترین خروجی برای جلوگیری از اورفیت و همچنین دقت بالا را داشت.

۲.۱) فاز ۱

۱.۲.۳) نمودار ها



میتوان به وضوح مشاهده کرد نمودار لاس و دقت، نسبت به دو فاز قبل smooth هستند و همچنین نمودار ولیدیشن و ترین، در ایپاک های نهایی به هم بسیار نزدیک میشوند که نشان میدهد مدل به خوبی جنرالایز شده و خبری از اورفیت نیست.

برای ماتریس گمراهی همچنان تحلیل های قبلی صدق میکند ولی میتواند دید که قطر اصلی پرنگ تر شده که دلیل آن افزایش دقت مدل است. ولی همچنان اشتباہ پرداخت کردن بین آن دو جفت کلاس موجود است. نمودار ROC بسیار پیشرفته خوبی داشته است و مساحت زیر قوس برای دو کلاس farfalla و ragno برابر ۱ است که یعنی مدل به خوبی این دو کلاس را طبقه بندی میکند. و کمترین مساحت هم برابر نود و هشت صدم است که ۲ درصد رشد نسبت به فاز اول است.

۲.۲.۳) ارزیابی



: شکل ۵

ارزیابی مدل :

۸۴ : Score F1 .

۸۴ : Recall .

۸۴ : Precision .

۸۵ : Validation Accuracy .

۸۴ Training: Acuracy .

تفاوت بین دقت ها حدود ۱ درصد است که یعنی مدل به شکل عالی ترین و جنرالایز شده و همچنین از ۱۰ عکس موجود، هر ده تا به درستی پیشنبینی شده اند.

۳.۲.۳) بررسی پارامتر ها

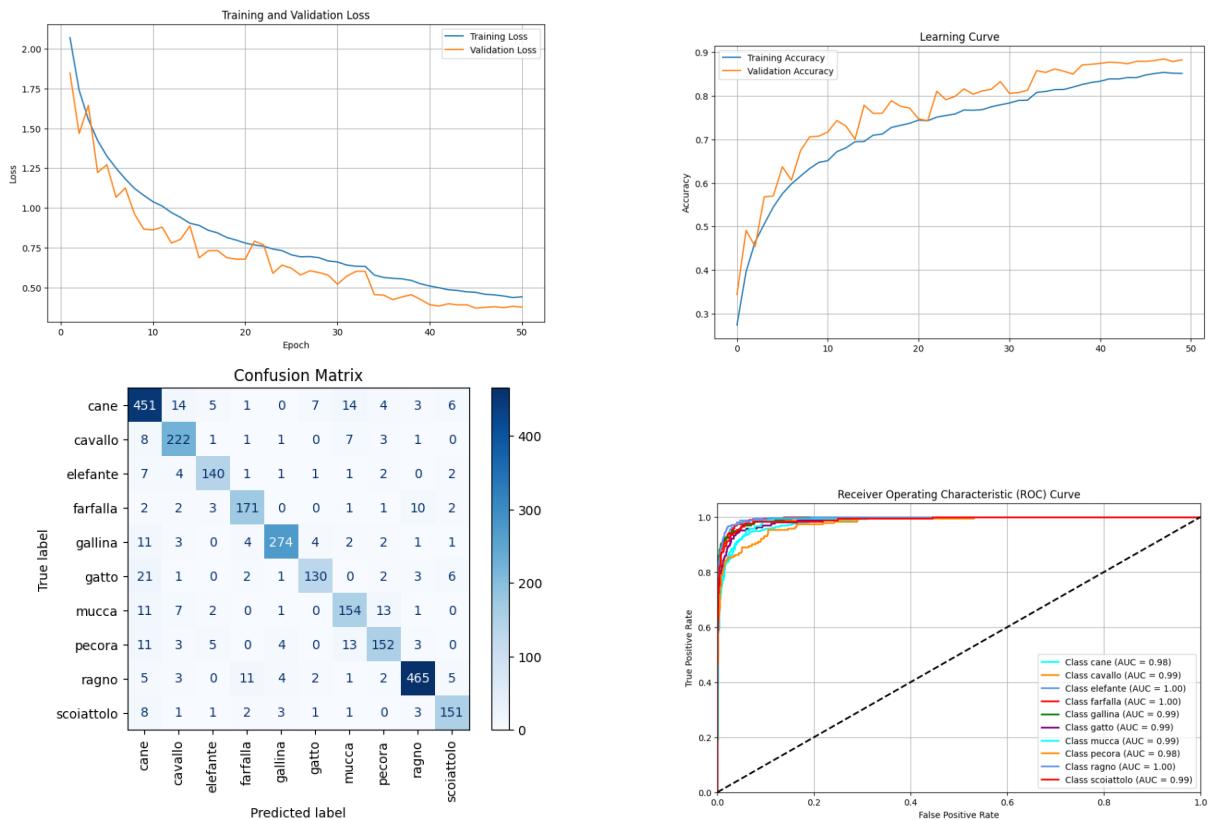
برای اکتیویشن فانکشن، LeakyReLU هم بررسی شد که شاید به خاطر مقدار دادن برای مقادیر زیر ۰ در دقت تاثیر مثبت بگذارد که در نهایت در دقت تاثیری نگذشت و به دلیل ماهیت آن صرفقاً محاسبات و در نتیجه زمان ترین بیشتر شدو. به عبارتی Vanishing Gradient پیش نیامده بود که از آن جلوگیری کنیم.

مراحل ترین مدل به ازای پارامتر های زیر داخل کد موجود است.
کوچک تر : مدل تقریباً همان دقت قبل را داشت فقط چون جرمیه ای که برای وزن ها در نظر گرفته میشد کوچک تر بود، دیرتر همگرا شد.

Adam Optimizer : چون لرنینگ ریت بسیار بالا بود مدل به اصطلاح **Overshoot** میکرد و اصلا پیشرفتی نکرد. با لرنینگ ریت 0.005 مدل دقیق 83% درصد میداد.
Momentum : مدل های معروف از مومنتوم هم استفاده کردند و ماهم در ابتدا این پارامتر را استفاده کردیم. چون لرنینگ ریت اولیه بسیار بزرگ بود . همچنین مومنتوم به روز رسانی گرادیان را افزایش میدهد، ترکیب این دو باعث شد مدل اصلا یاد نگیرد. با کاهش هر کدام از این پارامتر ها در تست های انجام شده، نتایج بهتر میشوند.

۳.۲ فاز ۲

۱.۳ نمودار ها





شکل ۶:

ارزیابی مدل :

۸۷ : Score F1 •

۸۷ : Recall •

۸۷ : Precision •

۸۸ : Validation Accuracy •

۸۵ Training: Acuracy •

نمودار لاس و دقت بسیار مشابه قسمت قبلی است ولی در نهایت ولیدیشن و ترینینگ باهم کمی فاصله دارند و ولیدیشن دقت بالاتری دارد که دلیل آن وجود داپ اوت و همچنین دیتا آگمنتیشن است. ماتریس گمراهی در ظاهر مثل فاز قبل است اما اگر دقت کنیم کمی متفاوت است. مدل برای پیشنبینی بعضی کلاس ها بهبود یافته مثل cane و لی برای بعضی کلاس ها افت دقت داشته مثل gatto. البته که مقادیر بهبود بیشتر از افت بوده و برای دو کلاسی است که داده های بیشتری از آن ها داشتیم و مدل مایل شده به یاد گرفتن دو کلاسی که داده بیشتری از آنها موجود است.

نمودار ROC بهبود یافته است و ۳ کلاس مساحت ۱ دارند و بدترین بازدهی برای دو کلاس آن هم فقط با مساحت ۹۸ صدم بوده است. اگر این نمودار را با نمودار فاز اول مقایسه کنیم میتوان دید که طبقه بندی کلاس ها چقدر بالانس تر و دقیق تر شده است.

در این فاز دقت ۹۰ درصد هم گرفتیم که متسفانه پاک شد ():

۳.۳) بررسی پارامتر ها

کوچک تر: حدود ۳ درصد افت دقت ولیدیشن داشتیم که میتوان یکی از دلایل آن را این در نظر گرفت که چون جریمه در نظر گرفته شده کاهش یافته، پس مدل به خوبی Generalize نشده. و اگر با موارد دیگر از اورفیت جلوگیری نمیکردیم، حتی اورفیت هم میشد.

بزرگ تر: کمی دقت ترین افزایش یافته ولی باز هم مدل اورفیت نشده که از دلایل آن میشود به این اشاره کرد چون ما جریمه بزرگی برای وزن ها در نظر گرفتیم، مدل را مجبور کردیم ساده تر یاد بگیرد.

Adam Optimizer: با لرنینگ ریت ۰.۰۵۱۰ انجام شد که حدود ۶ درصد دقتی از مدل اصلی کمتر بود. به این دلیل که هم برای مدل ما SGD بہتر عمل میکند و به عبارتی در ایپاک های نهایی مسیر بهینه را overshoot میکرد. البته رشد آدام

سریع‌تر از SGD بود که چون SGD با لرنینگ ریت بزرگ شروع میکرد، چند اپیاک طول میکشید تا لرنینگ ریتش کاهش یابد و باز بتواند مدل را بهبود بدهد ولی در نهایت وقتی دقت مدل به ۸۲ درصد رسید، SGD بهتر همگرا شد.

۴) فاز ۴

۱ Dataset ۱.۴

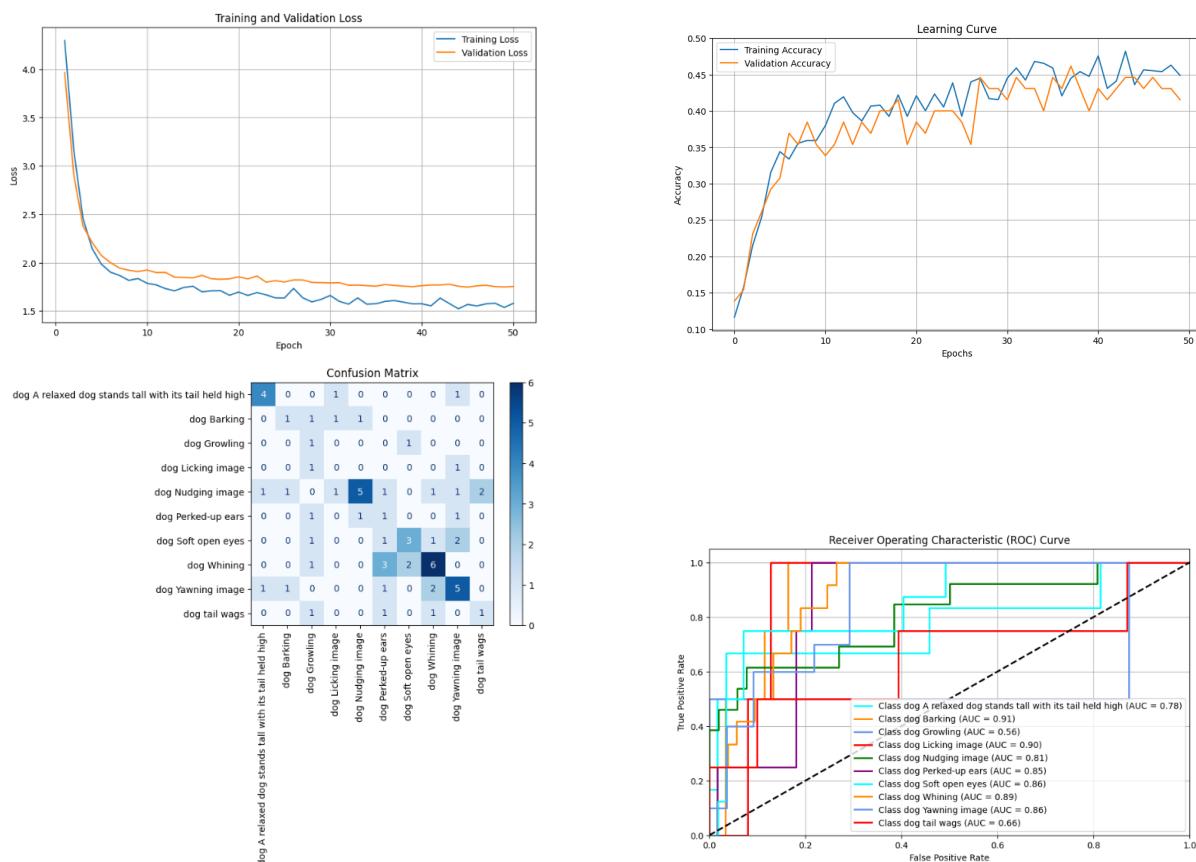
۱ Part ۱.۱.۴

در این قسمت با توجه به مدلی که در فاز قبل آموزش داده ایم به Learning Transfer میپردازیم. در بهترین مدل انتخاب شده در فاز قبل مدل با مقدار دیتای نسبتاً خوبی train شده است. حال مدل دارای نورون هایی است که به وزن های بهینه‌ی نسبتاً خوبی رسیده. حال این وزن ها در لایه feature میتوانند به عنوان خوبی استفاده شوند.

اما نکته قابل توجه این است که این وزن ها اگر خوب ترین شده باشند به تسك ها یا دیتاست وابسته نیستند و صرفاً feature ها را از عکس extract میکنند. اما دیتاست محدود در فاز یک که مدل بر روی آن ترین شده است نمیتواند با تسك یا دیتاست داده شده به نتایج خیلی خوبی دست پیدا کند و انتظار generalize شدن مدل را نمیتوانیم داشته باشیم همچنین دیتاست موجود در فاز ۴ حجم دیتای زیادی ندارد که باز هم با استفاده از offline data agumentation میزان این دیتا را زیاد کردیم تا مدل به نتیجه بهتری در هنگام ترین برسد.

همچنین پایین آوردن size batch به عدد ۸ به دلیل دیتا کم به نتیجه بهتر کمک کرده است. در قسمت اول لایه conv را فریز کردیم و فقط لایه کلسیفایر ترین میشود و تعداد پارامتر های قابل ترین به ۲۵۷۰ رسیده است.

نمودار ها

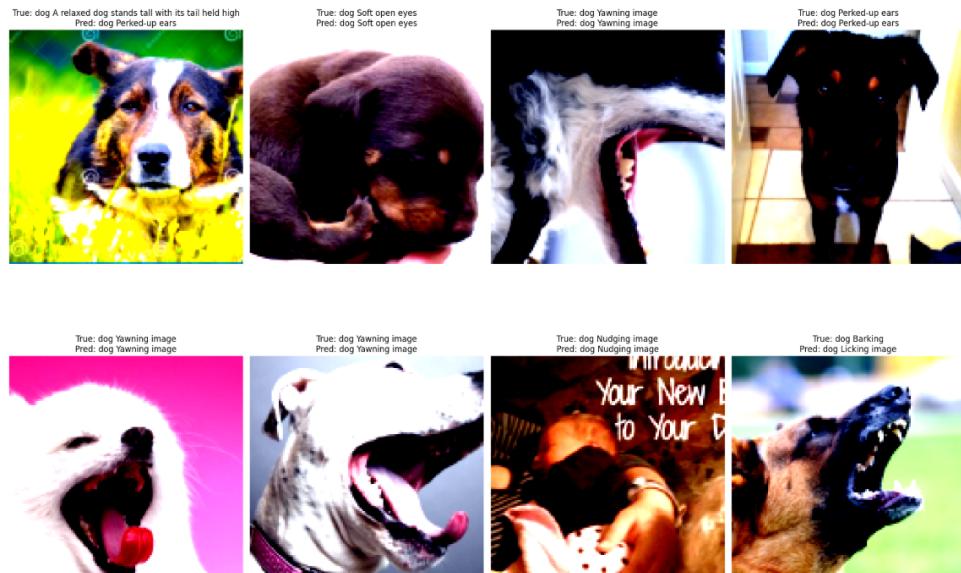


در نمودار لاس و دقت روند به ترتیب کاهشی و افزایشی است که در مجموع یعنی مدل دارد روند مثبتی در یادگیری را طی میکند علت بعضی نوسانات هم به تغییر لرنینگ ریت به صورت داینامیک برمیگردد ماتریس گمراهی نشان میدهد تعداد اندکی از کلاس ها خوبی ترین شده اند اما اکثراً کلاس بندی خوبی را ندارند و

ماتریسی که طبق انتظار بخواهیم بیشترین مقادیر بر روی قطر اصلی باشد در اینجا نداریم و کلاس های اشتباه پیشبینی شده بسیار زیاد اند. در نمودار ROC هم با توجه نسبت TP به FP فقط کلاس barking Dog بالاترین درصد را دارد و کلاس های دیگر نتوانسته اند به این خوبی باشند.

که این میتواند به دلایلی باشد که در ابتدای فاز ۴ ذکر شده است همچنین اشتباهات زیاد بین کلاس های مشابه (مانند سگ هایی که وضعیت مشابه دارند) نشان دهنده چالش های مرتبط با تمایز ویژگی های ظرفیت بین کلاس ها است.

ارزیابی



شکل ۷:

ارزیابی مدل :

۴۴ : Score F1 .

۴۴ : Recall .

۴۷ : Precision .

۴۱ : Validation Accuracy .

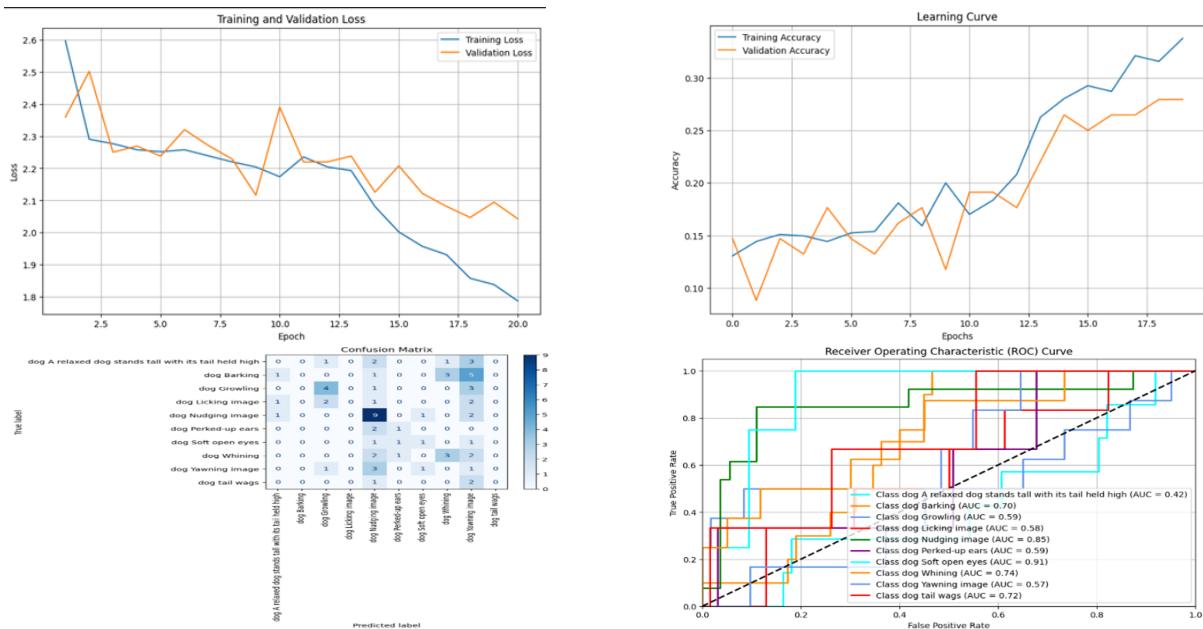
۴۴ Training: Accuracy .

در اینجا میبینیم که مدل نسبت به دیتای موجود در کلاس ها توانسته هرچند که درصد پایینی در نمودار های بالا برای یادگیری گرفته اما کلاس بندی خوبی را در خروجی نشان بدهد

۲ Part ۲.۱.(۴)

در قسمت دوم به جای فریز کردن کل لایه feature فقط بخشی از اون رو فریز کردیم. در این بخش با توجه به مدل طراحی شده نتوانسته است به خوبی ترین شود یکی از دلایل میتواند میزان کمبود دیتا باشد. با بررسی تعداد ایپاک بیشتر به این نتیجه رسیدیم مدل تنها توان همین درصد یادگیری را دارد و بیشتر از آن علی رغم جلوگیری از اورفیت با روش های متفاوت باز هم اورفیت میشود. بعد فریز کردن یک بلاک از لایه های feature تعداد پارامتر ها به ۱۸۸۵۷۰ کاهش پیدا کرده است.

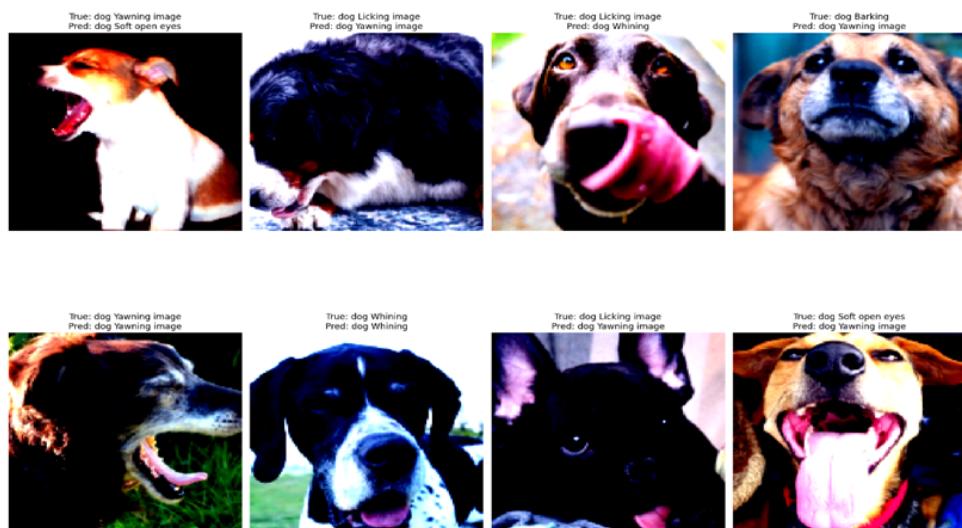
نمودار ها



برای نمودار دقت و لاس و همچنین ماتریس گمراهی تحلیل های پارت قبل برای این پارت هم صدق میکند و صرفا کلاس هایی که به اشتباه پر迪کت شده اند فرق کرده اند و نشان میدهد مدل به مست یادگیری یک سری کلاس دیگر رفته. در پارت قبل whining dog بهتر طبقه بندی شده بود و در این پارت . nudging dog

در نمودار ROC هم با توجه نسبت TP به FP فقط Dog eyes open soft Dog بالترين درصد را دارد و کلاس های دیگر نتوانسته اند به این خوبی باشند. این تغییر درصد ها در کلاس ها میتواند به دلیل بلاکی باشد که در مدل فریز شده است فریز شدن بلاک های دیگر میتواند درصد ها و accuracy ها متفاوتی را به ما بدهد. همچنین در کل مانند پارت قبل دلیل کم شدن accuracy ها میتواند به دلایلی باشد که در ابتدای فاز ۴ ذکر شده است.

ارزیابی



:شکل ۸

ارزیابی مدل :

۲۱ : Score F1 •

۲۴ : Recall •

۲۱ : Precision •

۲۷ : Validation Accuracy •

۳۳ Training: Acuracy •

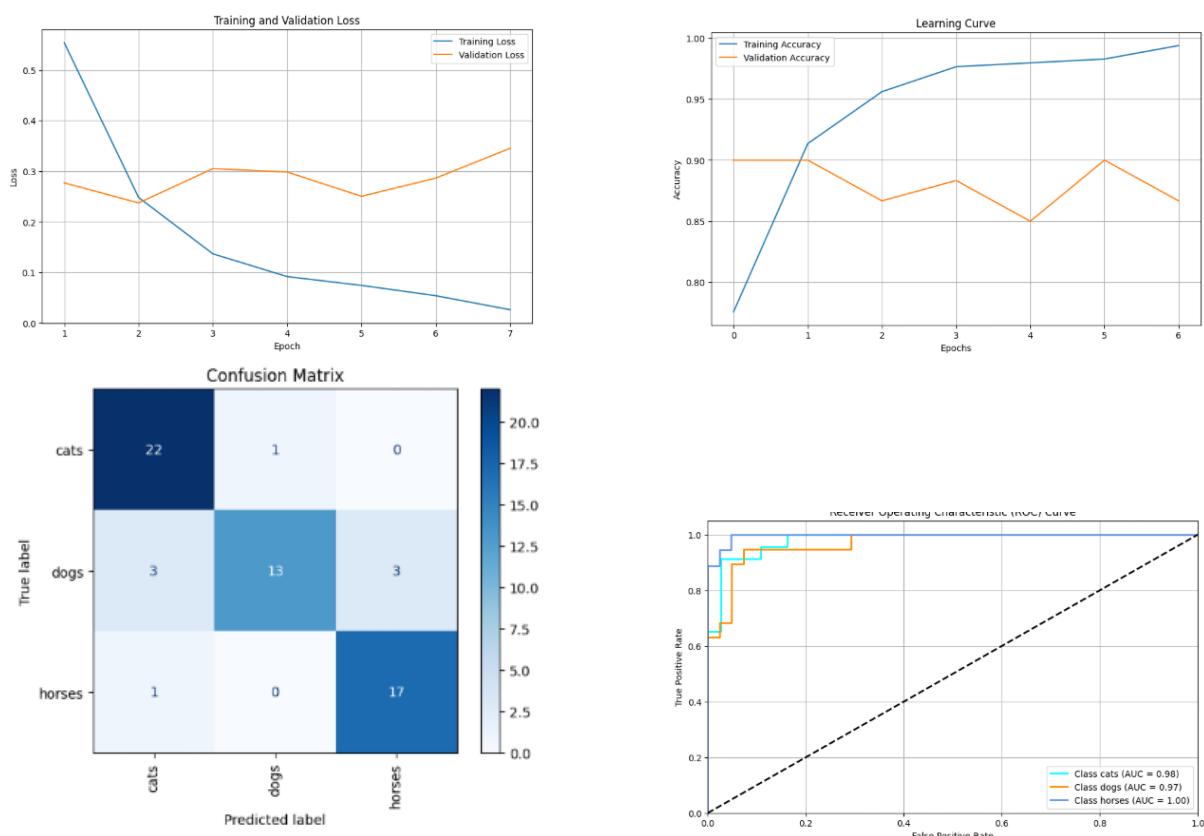
باز هم مانند قسمت قبل تغییری در بهبود کلاس بندی ها انجام نشده و چه بسا بدتر هم شده است و این میتواند وابسته به لایه ای که فریز کردیم در مدل باشد

۲ Dataset ۲.(۴)

۱ Part ۱.۲.(۴)

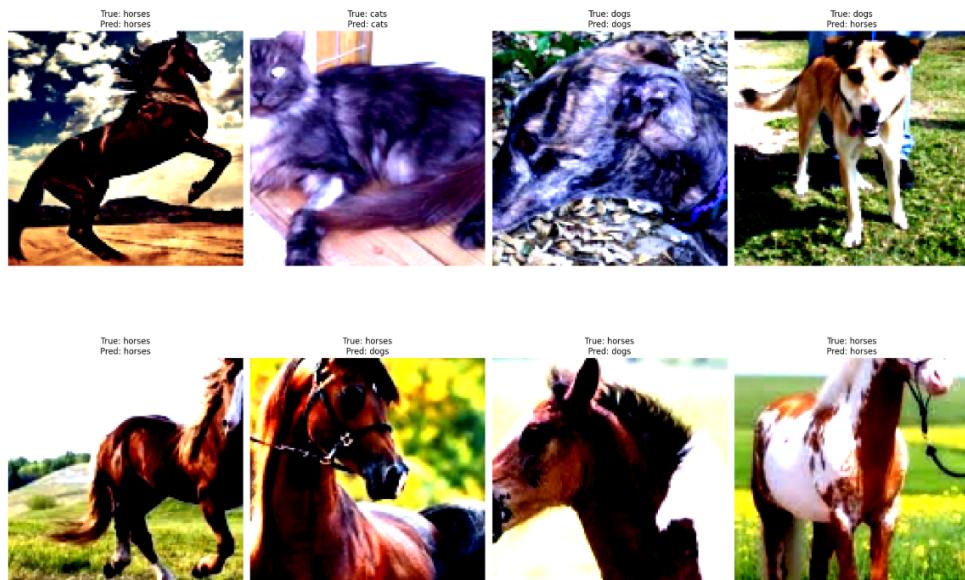
در این قسمت مجدد به بررسی بهترین مدل در فاز ۳ با دیتاست جدید میپردازیم. با توجه به اینکه تعداد کلاسه های دیتاست دوم ۳ تا هست فقط لایه آخر یعنی لایه کلسیفایر را تغییر میدهیم یعنی لایه آخر مدل ۳ کلاس دارد. و در نهایت تعداد پارمتر ها در این حالت به 19688^3 کاهش میابد.
پس از آموزش مجدد مدل، به دقت خوبی میرسیم که نشان میدهد لایه feature در مدل، وزن های خوبی را دارد و توانسته است با این که دیتای کمی داریم اما کلاس بندی خوبی را انجام دهد

نمودار ها



در نمودار لاس و دقت روند آنقدر قابل تحلیل نیست چون در اینجا کمی به نتیجه خوبی رسیده است اما نشان از یادگیری خوبی را در کل میدهد
ماتریس گمراهی نشان میدهد کلاس ها به خوبی ترین شده اند و بهترین کلاس هم کلاس horse است در نمودار ROC هم با توجه نسبت TP به FP کلاس horse بالاترین درصد را دارد و کلاس های دیگر هم به کمی خطأ به خوبی کلاس بندی شده اند.

ارزیابی



شکل ۹:

ارزیابی مدل :

۹۲ : Score F1 .

۹۲ : Recall .

۹۲ : Precision .

۸۶ : Validation Accuracy .

۹۹ Training: Accuracy .

از ۱۰ عکس نمایش داده شده ۲ عکس اشتباه است که نشان میدهد مدل به خوبی نتوانسته اسب یا سگ را در این پوزیشنی که قرار دارند به خوبی یاد بگیرد و به خوبی تشخیص دهد.
همچنین دقت مدل برای داده های تست بسیار خوب است و اگر ترشولد اورفیت شدن را ۷ درصد بگیریم، مدل در مرز اورفیت شدن است ولی کامل اورفیت نشده.

۴ and ۲ Part ۲.۲.۴

در این قسمت به بررسی مدل های معروف pretrained میپردازیم
نکته ای که این مدل ها دارند این هست که روی دیتاست های بسیار بزرگ ترین شده اند که در نتایجی که گرفته شده است مشهود است زیرا accuracy خوبی گرفتیم. . این مدل ها به اندازه ای روی دیتاست های بزرگ ترین شده اند که وزن هایی که در نورون های لایه feature دارند بهینه هستند و به اصطلاح generalize شده اند که میتوان از آن ها برای تسک ها و دیتاست های متفاوت استفاده کرد .

: Resnet⁶⁰

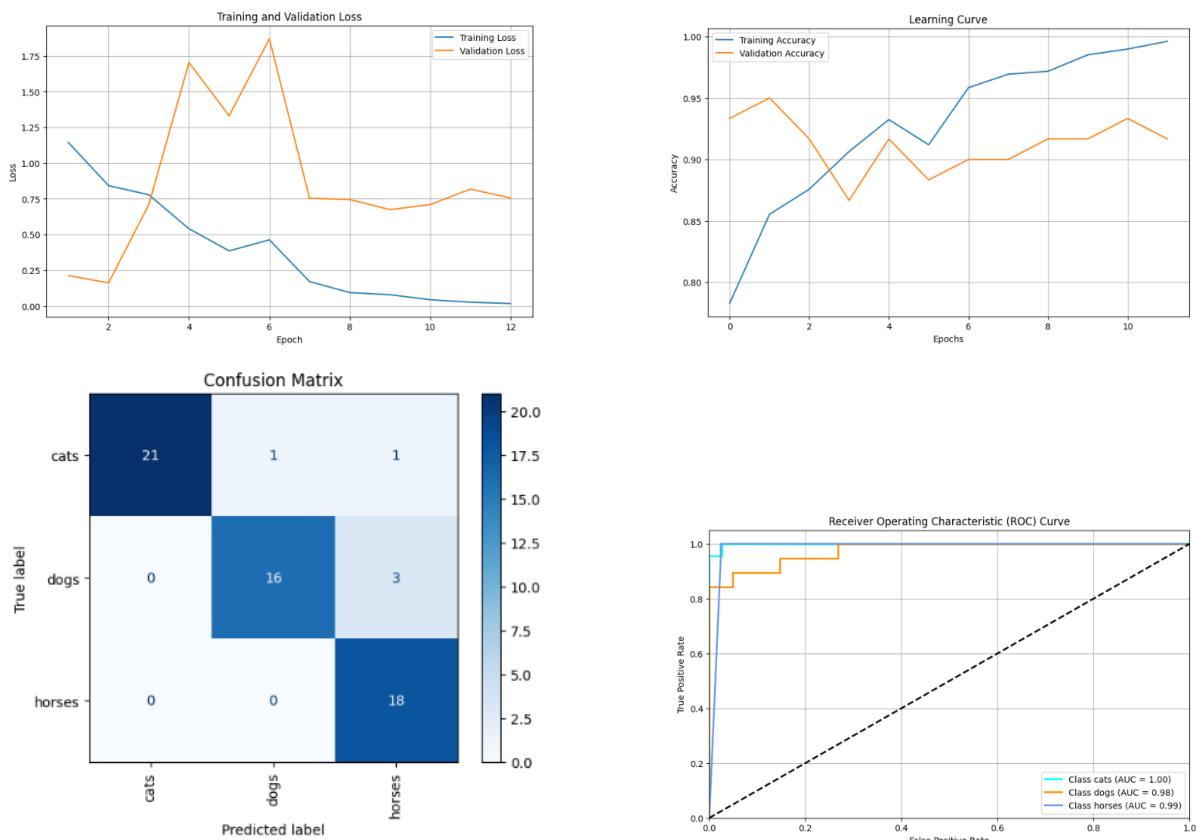
این مدل شامل ۲۳۵۱۴۷۹ پارامتر است اما به غیر از لایه کلسیفایر که ۶۱۴۷ پارامتر دارد همه را فریز کرده ایم. و بهترین خروجی گرفته شده با پارامتر های زیر بوده است

dynamic ۰.۰۱ : Learning Rate •

SGD : Optimizer •

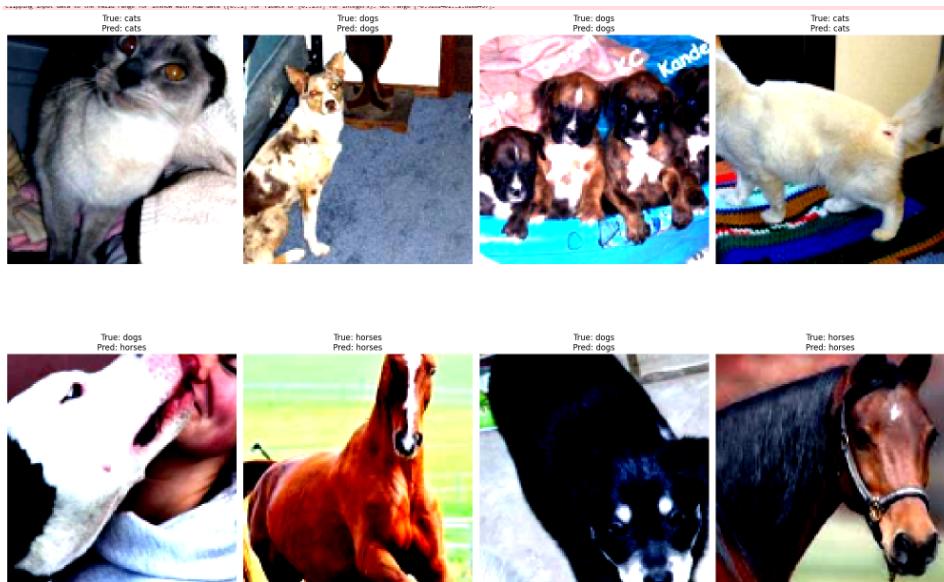
۰.۰۰۰۱ : decay weight •

نمودار ها



در نمودار ها کاملا مشخص است که مدل توانسته سریعاً به درصدی خوبی بررسه و دلیل آن پری ترین شدن این مدل ها با دیتا ست ها بسیار بزرگ و کلاس های متنوع است.
با توجه به ماتریس گمراهی و نمودار ROC کلاس بندی برای گربه به خوبی انجام شده است همچنین برای اسب و سگ هم با درصد خطای بسیار اندکی میتوان آموزش خوبی بر روی این کلاس ها متصور شد.

ارزیابی



شکل ۱۰:

ارزیابی مدل :

۹۴ : Score F1 .

۹۴ : Recall .

۹۴ : Precision .

۹۱ : Validation Accuracy .

۹۹ Training: Acuracy .

همانطور که از این ۱۰ پیش بینی پیداست مشهود است که گربه ها را به خوبی تشخیص داده است اما در تشخیص سگ و اسب در این پوزیشن خاص خیلی خیلی ضعیف عمل کرده است
: MobileNetV2

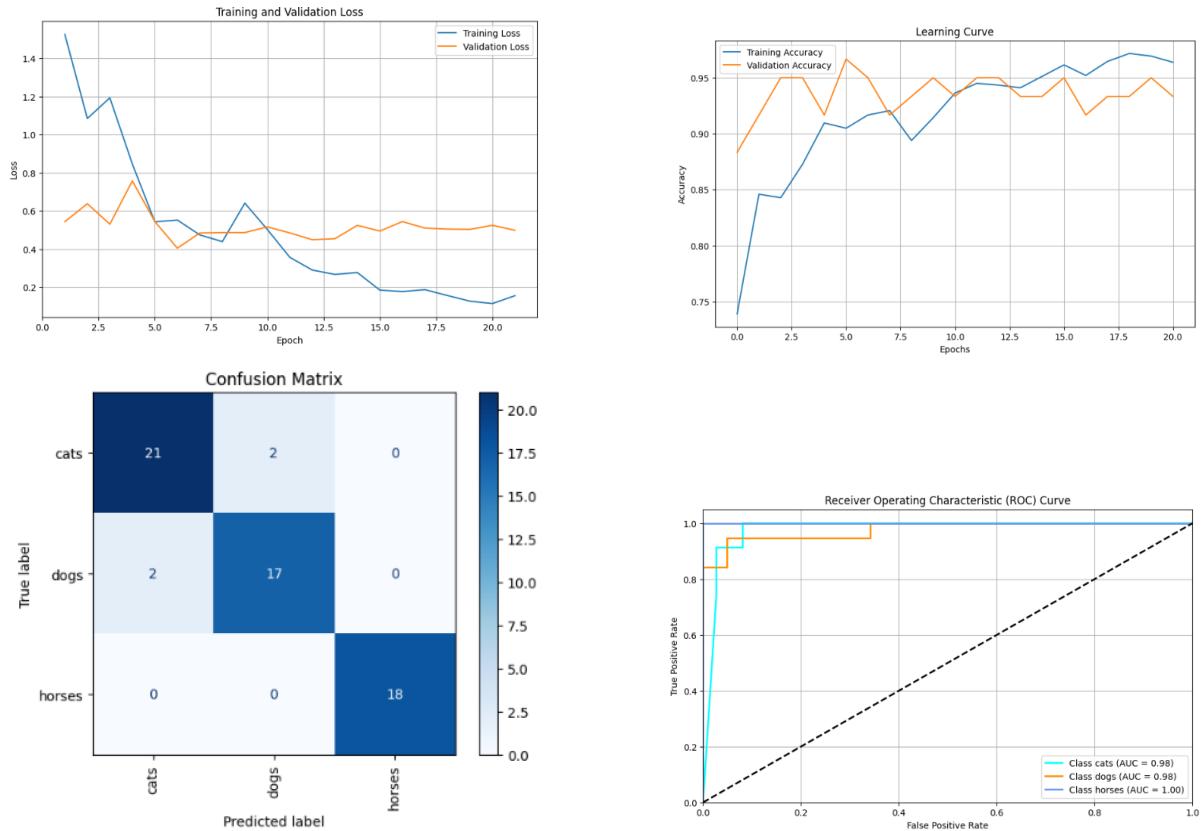
این مدل شامل ۲۲۲۷۷۱۵ پارامتر است اما به غیر از لایه کلسیفایر که ۳۸۴۳ پارامتر دارد همه را فریز کرده ایم. و بهترین خروجی گرفته شده با پارامتر های زیر بوده است

dynamic ۰.۰۱ : Learning Rate .

SGD : Optimizer .

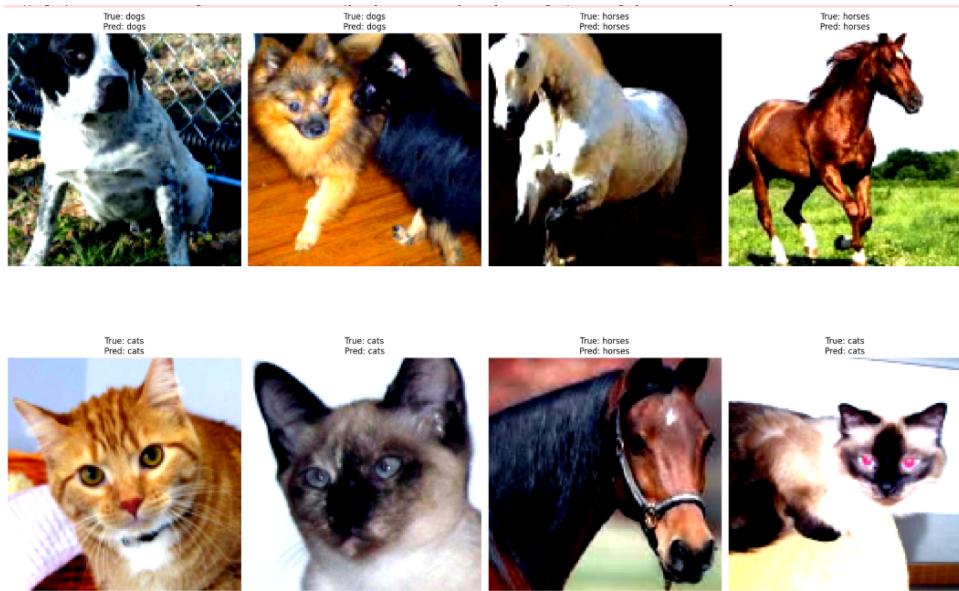
۰.۰۰۰۱ : decay weight .

نمودار ها



تحلیل های پارت قبل قبیل هم صدق میکند اما همانطور که در نمودار ها مشخص است،، روند ترین smooth تر پیش رفته است

ارزیابی



شکل ۱۱

ارزیابی مدل :

۹۴ : Score F1 •

۹۴ : Recall •

۹۴ : Precision •

۹۳ : Validation Accuracy •

۹۶ Training: Accuracy •

همانطور که از این ۱۰ پیش بینی پیداست مشهود است پیش بینی خوبی توانسته است مدل داشته باشد که با توجه به accuracy میتوان همین انتظار را داشت.

همچنین دقت ترین نسبت به پارت کمتر است که میتوان دید این مدل بهتر جنرالایز شده و حتی دقت تست هم بیشتر است و در کل عملکرد بهتری داشته است.

: EfficientNetB0

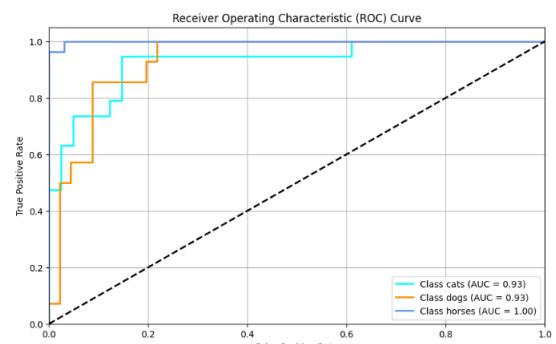
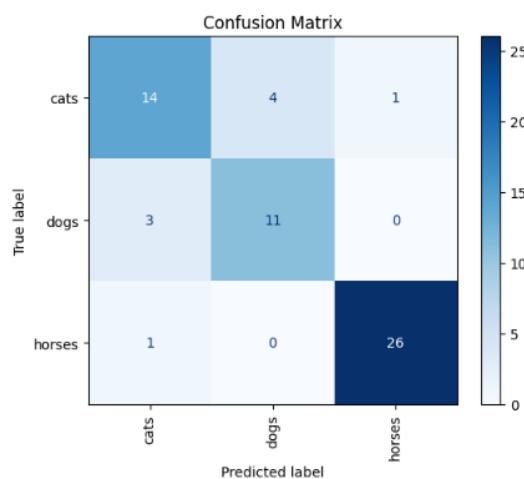
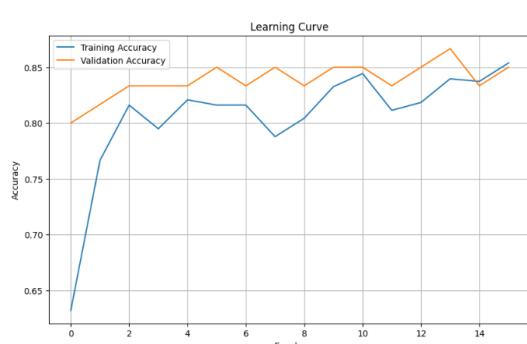
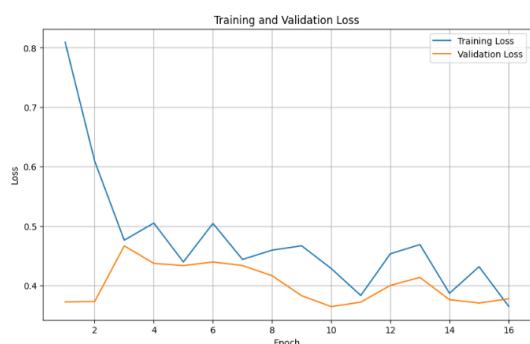
این مدل شامل ۴۰۱۱۳۹۱ پارامتر است اما به غیر از لایه کلسیفایر که ۳۸۴۳ پارامتر دارد همه را فریز کرده ایم. و بهترین خروجی گرفته شده با پارامتر های زیر بوده است

dynamic ۰.۰۱ : Learning Rate •

SGD : Optimizer •

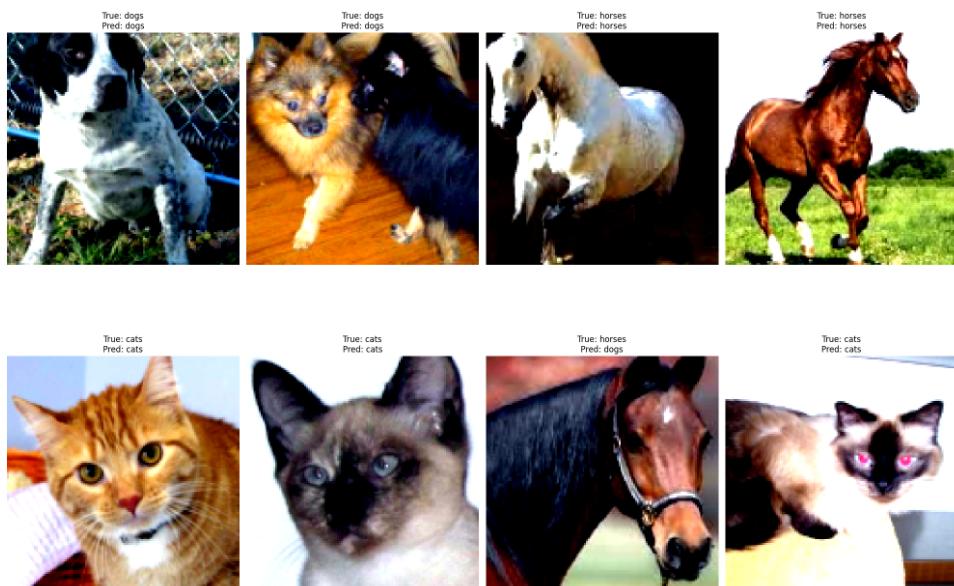
۰.۰۰۰۱ : decay weight •

نمودار ها



در نمودار ها کاملا مشخص است که این مدل نتوانسته مانند دو مدل قبل به درصد خوبی برسد و کمی درصد ها پایین آمده است اما باز هم با توجه به ماتریس گمراهی و نمودار ROC کلاس بندی برای اسب به خوبی انجام شده است همچنین برای گربه و سگ هم با درصد خطای بسیار اندکی میتوان آموزش خوبی بر روی این کلاس ها متصور شد.

ارزیابی



شکل ۱۲:

ارزیابی مدل :

۸۳٪ : Score F1 •

۸۳٪ : Recall •

۸۳٪ : Precision •

۸۵٪ : Validation Accuracy •

۸۵٪ Training: Accuracy •

همانطور که از این ۱۰ پیش بینی پیداست مشهود است که گربه ها را به خوبی تشخیص داده است اما در تشخیص سگ و اسب در این پوزیشن خاص خیلی خیلی اندک ضعیف عمل کرده است در این مدل کمترین فاصله بین تست و ترین و البته پایین ترین دقت رو نسبت به دو مدل قبل داشته ایم.

بررسی پارامتر ها :

در این قسمت هر ۴ مدل را مجدد بدون اعمال offline data agumentarion هم چک کردیم اما نتایج افت داشته بود برای هر مدل نتایج به صورت زیر بوده است :

• بهترین مدل انتخابی از فاز ۳ : ترین ۹۱ درصد ولیدیشن ۸۶ درصد

• : ترین ۸۹ درصد ولیدیشن ۸۸ درصد ResNet۵۰

• : ترین ۸۴ درصد ولیدیشن ۸۸ درصد MobileNetV۲

• : ترین ۸۵ ولیدیشن ۸۶ EfficientNetB۰