به نام خدا
دانشگاه تهران
دانشکده مهندسی
برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق
تمرین دوم

نام و نام خانوادگی	**نام نام خانوادگی – نام نام خانوادگی**
شماره دانشجویی	**810199357 - 810199395**
تاریخ ارسال گزارش	**1402.09.04**

فهرست

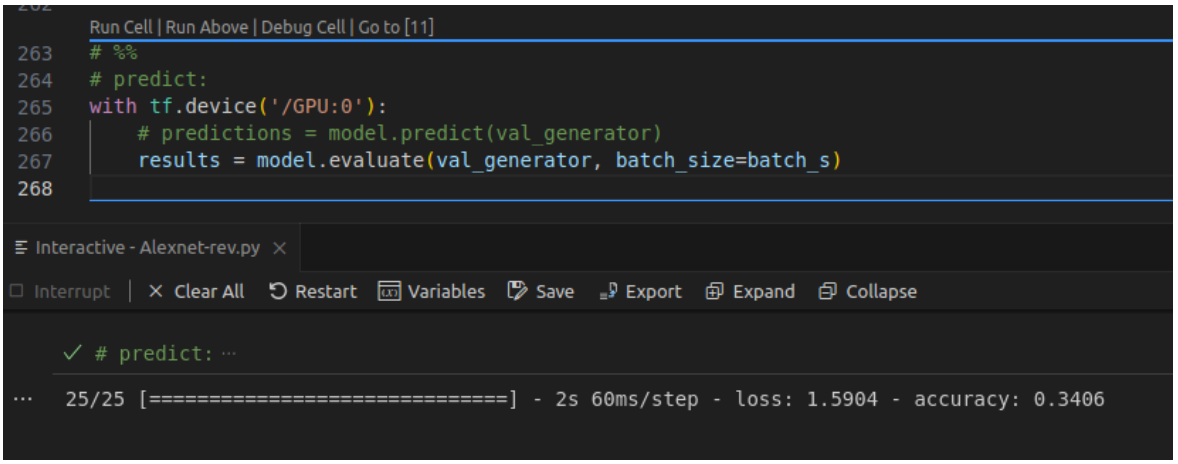
شکل‌ها

جدول‌ها

پاسخ 1. تجزیه و تحلیل احساسات صورت مبتنی بر CNN

AlextNet .1-1

:Tune and Train –



1.AlexNet accuracy on validation

پس از ۶۰ ایپاک ترین کردن به دقت ۳۵ روی داده validation رسیدیم. در اینجا validation داده ای است که آن را مدل ندیده است. و بر اساس آن نیز هیچ گونه hyper paramter tunninn صورت نگرفته است. مقادیر hyper paramter های ندل از روی مدل گرفته شده است.

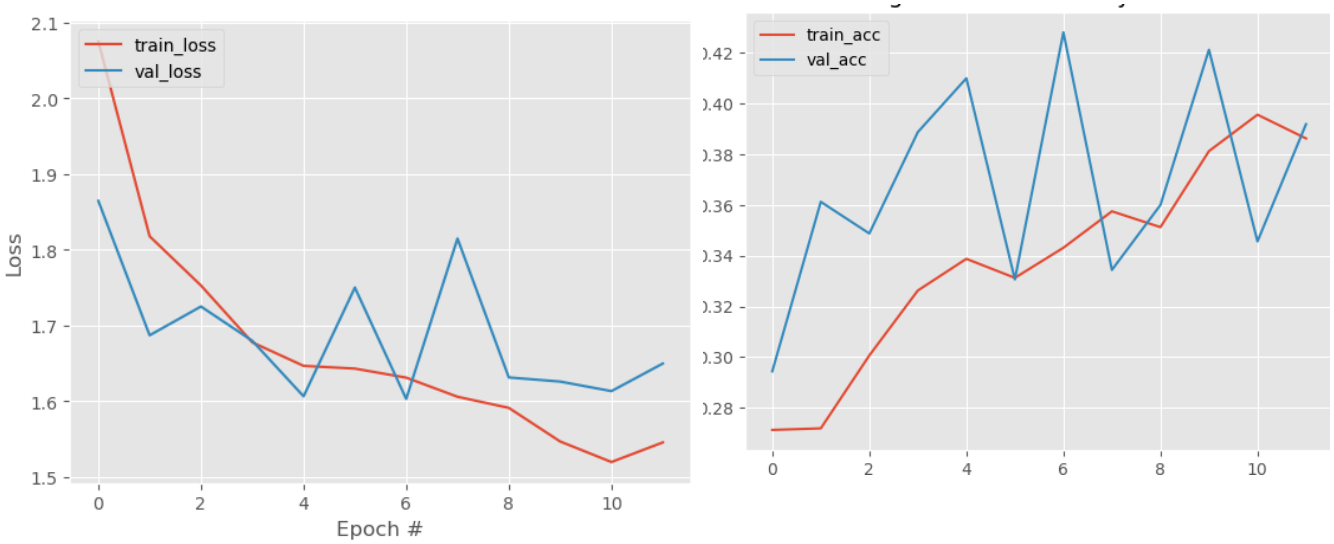
در tune کردن ما یک مدل pre-trianed شده داریم و میخواهیم آن را برای تسک جدید مون آپدیت کنیم. برای اینکار در صورت نیاز لایه های به اخر مدل pre-trained شده اضافه میکنیم و اصطلاحا لایه قبلی را freeze میکنیم. تا لایه های جدیدی که اضافه میکنیم learn بشوند. بعد از آن وارد فاز tunning میشود که بسته به شرایط کل یا قسمتی از لایه های قبلی را از حالت freeze در میاریم و مدل را برای چند ایپاک لرن میکنیم تا لایه های قبلی و لایه های جدید به توانند وزن مناسب را نسبت به یکدیگر در تسک مورد نظر بدست بیارند. لایه های pre-trained شده را نمیخواهیم از ابتدا لرن کنیم زیرا وزن آن ها برای بدست آوردن فیچر های عکسُ مناسب است زیرا از قبل رو تعدادی زیادی داده لرن شده اند و نمیخواهیم در ابتدای کار در ایپاک های اول وزن ها به طور قابل توجهی تغییر بکند.
در اینجا چون تسک عوض نمیشود پس لایه ای اضافه نمیشود و مستقیما دوباره مدل را روی داده های واقعی لرن میکنیم. و val_accuracy را بررسی میکنیم. تا از overfit جلو گیری کنیم.

مدل را اینقدر ران میکینم تا overfit به روی validation رخ بدهد. البته چون داده های validation کم هستند. ممکن است نمایانگر خوبی از واقعیت نباشند. با ۱۲ ایپاک ران کردن در مرحله tunning به ۴۰ accuracy درصد رسیدیم. اگر بیشتر از این مدل رای آموزش بدهیم به دقت های بالاتری مانند ۸۸ برای داده tune میرسیم ولی همان طور که بالاتر ذکر شد دچار اورفیت میشویم و به روی داده validation خوب عمل نخواهیم کرد.

```
loss: 1.6307 - accuracy: 0.3431 - val_loss: 1.6026 - val_accuracy: 0.4281

loss: 1.6054 - accuracy: 0.3575 - val_loss: 1.8144 - val_accuracy: 0.3344

loss: 1.5908 - accuracy: 0.3512 - val_loss: 1.6309 - val_accuracy: 0.3600

loss: 1.5464 - accuracy: 0.3812 - val_loss: 1.6255 - val_accuracy: 0.4212

loss: 1.5193 - accuracy: 0.3956 - val_loss: 1.6129 - val_accuracy: 0.3456

loss: 1.5451 - accuracy: 0.3862 - val_loss: 1.6493 - val_accuracy: 0.3919
```

2.AlexNet accuracy on validation after tunning

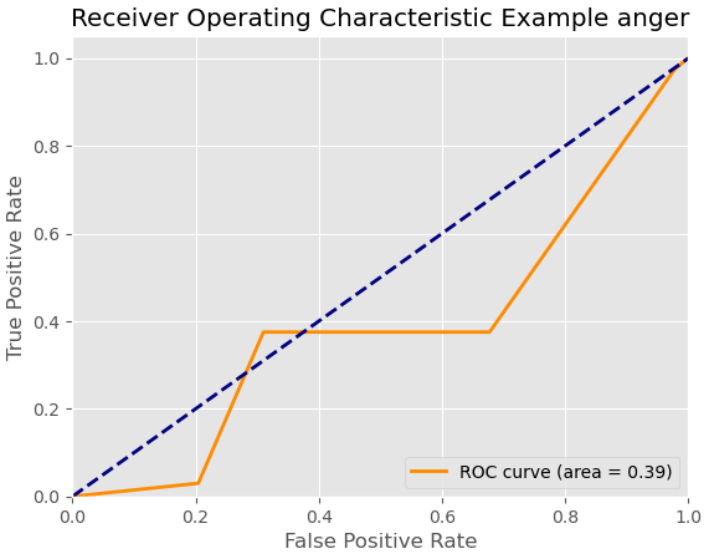
Loss & acuuracy –

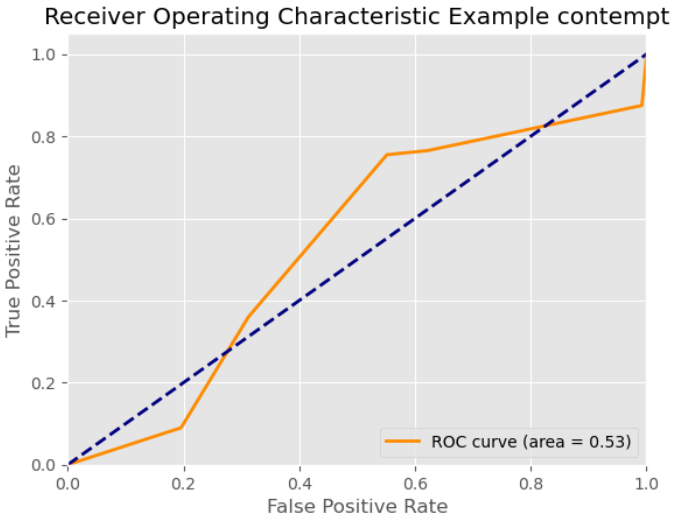


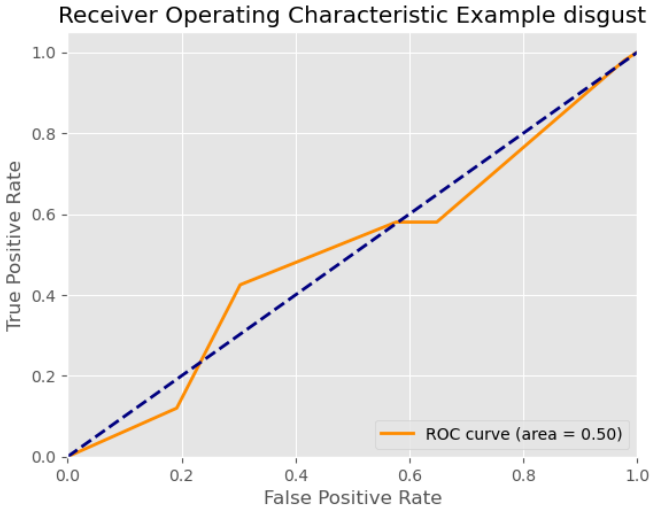
3.AlexNet accuracy & loss

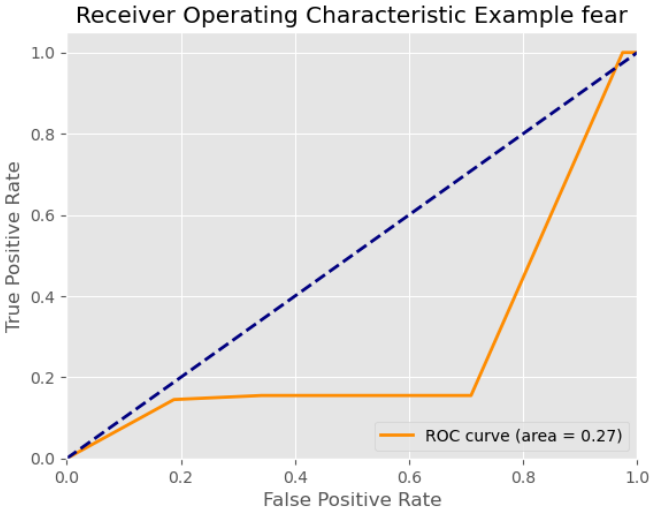
اوسیلیشن به ری داده های validation به خاطر این هست که تعداد داده های کم است. و در کل روند کاهشی loss برای validation را مشاهده میکنیم تا ایپاک ۱۰ ام و بعد از مدل شروع به اورفیت میکند. برای داده های tune که مدل در حاضر به روی آن ها اموزش میبیند برای loss که روند کاملا کاهشی هست و در نتیجه accuracy هم کاملا به صورت صعودی \یش میرود همان طور که گفته شد اگر بیشتر از آموزش بدهیم به روی داده های tune خیلی بهتر عمل خواهیم کرد ولی عملکرد مان به روی داده های validation بدتر میشود. این پدیده را(overfitting) میتوان از فاصله نمودار های خطا یا validation مدل فهمید؛ به طوری که روند نمودار loss هم به روی validation و هم به روی tune به صورت نزولی هست و هر دو باهم دارند کاهش پیدا میکنند و به اصطلاح converge میکنند. اما بعد از یک نقطه ای همچنان loss برای داده اموزش کاهشی است؛اما به صورت صعودی برای validation رو به افزایش میباشد.

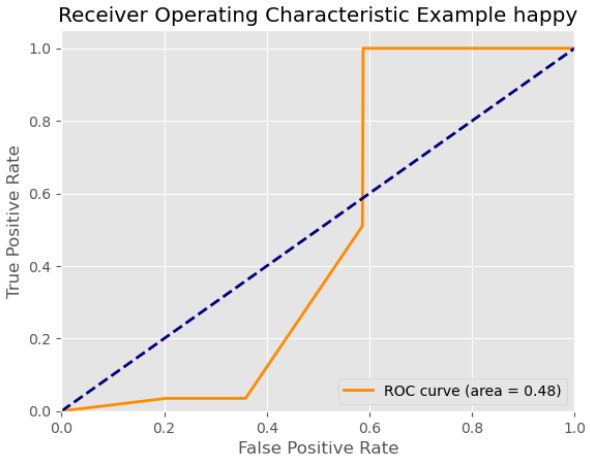
ROC –

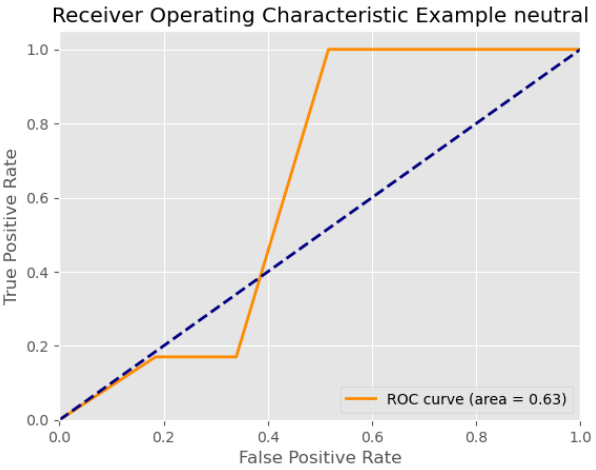

Receiver Operating Characteristic Example anger
True Positive Rate
False Positive Rate
ROC curve (area = 0.39)

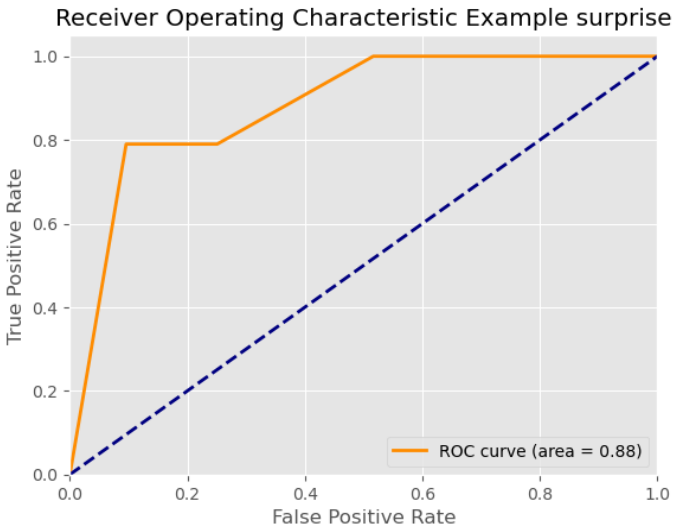

Receiver Operating Characteristic Example contempt
True Positive Rate
False Positive Rate
ROC curve (area = 0.53)

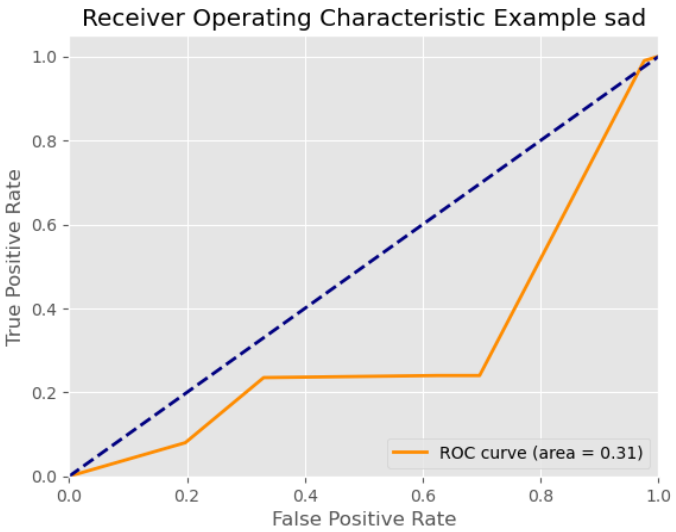

Receiver Operating Characteristic Example disgust
True Positive Rate
False Positive Rate
ROC curve (area = 0.50)


Receiver Operating Characteristic Example fear
True Positive Rate
False Positive Rate
ROC curve (area = 0.27)


Receiver Operating Characteristic Example happy
True Positive Rate
False Positive Rate
ROC curve (area = 0.48)


Receiver Operating Characteristic Example neutral
True Positive Rate
False Positive Rate
ROC curve (area = 0.63)





4.AlexNet ROCs

نمودار roc بر اساس confusion matrix کار میکند. در واقع فرمول هر محور آن به صورت زیر است:

$$\text{True Positive Rate (TPR)} \quad \textit{also called sensitivity/recall/hit rate} \quad = \frac{TP}{P} = \frac{TP}{TP + FN}$$

$$\text{False Positive Rate (FPR)} \quad \textit{also called fall out} \quad = \frac{FP}{N} = \frac{FP}{FP + TN}$$

اگر رندوم مدل ما predict بکند؛ انگاه با خط y=x که به صورت خط چین آبی در صفحه کشیده شده است برابر است. و اگر ۱۰۰ تخمین بزند آنگاه به خط y = 1 نزدیک تر میشود. و اگر هم برعکس تخمین به زند به خط ۰ نزدیکتر میشود. هر چقدر مدل ما توانسته باشد در softmax نهایی خود بیشتر کلاس ها از یکدیگر متمایز بکند. در اینجا auc بیشتر میشود. auc مخفف area under curve هست. منظور از curve هم roc میباشد.

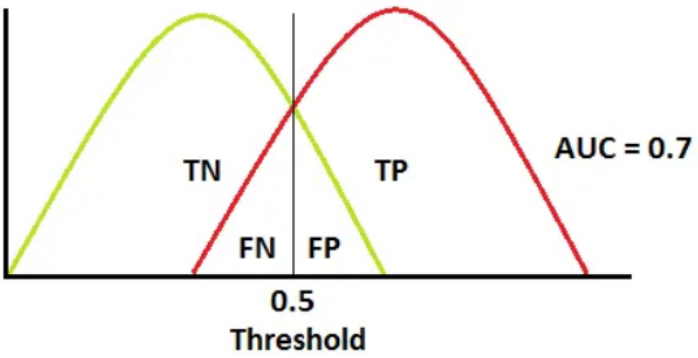
در واقع: AUC: Area Under the ROC Curve

نحوه محاسبه roc هم به این صورت هست که هر کلاس را به صورت one vs all مقایسه میکنند. برای مثال کلاس suprise را در نظر بگیرید. و دو فرمول بالا را برای آن محسابه میکنند و نمودار را رسم میکند. مدل به ازای هر عکس یک عدد بین ۰ و ۱ تولید میکند. هر چقدر فاصله عدد که مدل به هر عکس اختصاص میدهد برای دو کتگوری positive , negetive در اینجا عکس suprise هست یا نه؛ از هم بیشتر باشد. auc بیشتر میشود. اگر auc؛ ۱ باشد. به معنی این است که مدل دارد به خوبی تخمین میزند و اگر ۰.۵ باشد یعنی رندوم است و اگر ۰ باشد یه دارد برعکس تخمین میزند. در اینجا برای هر مدل roc,auc را رسم کردیم و با توجه به توضیحات داه شده؛ مدل برای مثال یکسری از کتگوری ها را به خوبی از بقیه تشخیص میدهد.مانند suprise؛ و happy نیز recall خوبی دارد. به طور هرچه زودتر مدل به y=1 برسد؛ آن مدل recall بهتری خواهد داشت.

برای disgust کاملا رندوم است زیرا برابر با خط x=y میباشد.

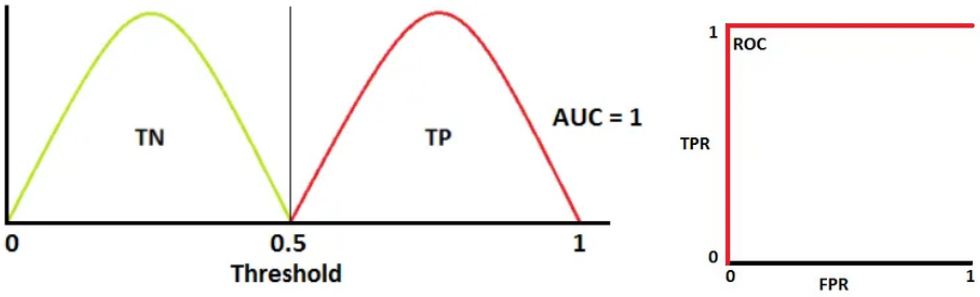
اما fear تعداد fp هایش زیاد است؛ به خاطر اینکه نمودار آن به سمت این محور کشیده شده است و این مسله در قسمت بعدی (recall, precision)نیز به طور واضح تری قابل بررسی است؛زیرا که این مسله باعث کاهش precision میشود.

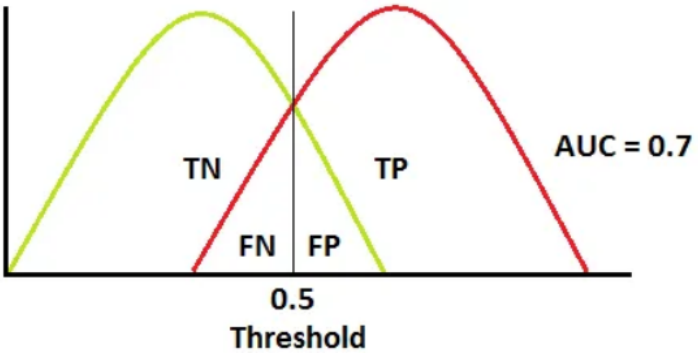
موارد گفته شده در confusion matrix ای که در قسمت توضیح داده خواهد شد نیز معلوم است.علت این که نمودار در بعضی کتگوری ها به خوبی عمل نمیکند میتواند کمبود داده ی ورودی باشد. زیرا مدل نتوانسته به خوبی generalize بکند. و همین طور کیفیت داده نیز در بعضی از موادر تاثیر گذار است.

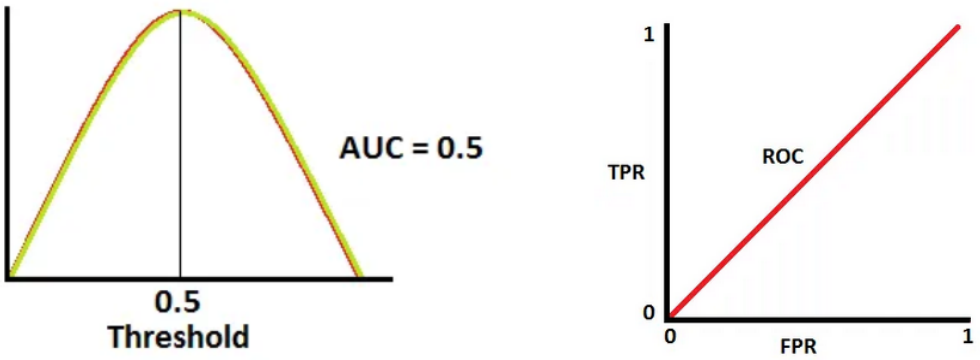


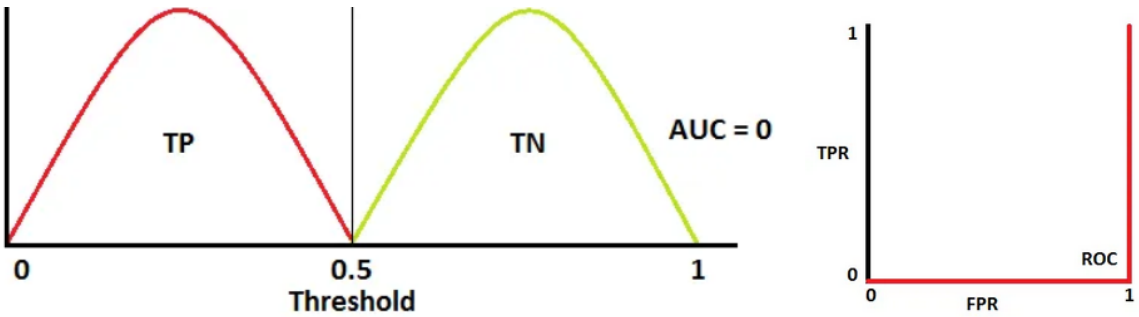
در واقع roc؛ treshold را از ۰ تا ۱ جا به جا میکنید و x,y را براساس فرمول هایی که بالا گفته شده مخاسبه میکند و بدین شکل یک نقطه محاسبه میشود. این فرانید را تکراری میکند و نقاط را بهم وصل میکند. هر چقدر که دو کلاس positive و negetive بیشتر از یکدیگر جدا باشند؛ auc به ۱ بیشتر نزدیک میشود. در واقع ما داریم رابطه بین tp که محور عمودی هست را با fp که محوری افقی هستُ محاسبه میکنیم. هر چقدر از بین آن هایی که ما میگوییم positive؛ تعداد بیشتری واقعا درست باشند و تعداد کمتری رو اشتباها درست بگوییم auc بیشتری خواهیم داشت.

برای توضیح بیشتر مقادیر و معنای مختلف roc , auc ؛ میتوان از عکس های زیر استفاده نمود:









F1, Recall, Precision –

	precision	recall	f1-score	support
anger	0.00	0.00	0.00	200
contempt	1.00	0.01	0.01	200
disgust	0.11	0.02	0.03	200
fear	0.31	0.84	0.46	200
happy	0.98	0.49	0.65	200
neutral	0.40	0.83	0.54	200
sad	0.14	0.15	0.15	200
surprise	0.54	0.79	0.64	200
accuracy			0.39	1600
macro avg	0.44	0.39	0.31	1600
weighted avg	0.44	0.39	0.31	1600

5.AlexNet Recall,F1,Precision

suprise همان طور که در قسمت قبل هم با توجه به نمودار ها roc اشاره شد؛ عملکرد خوبی دارد. و بالاترین recall برای suprise,happy میباشد. happy دقت بالایی دارد. زیرا ستون مربوط به happy در cm؛ فقط در سطر مروبط به همین کتگوری مقدار دارد و سطر های دیگر این ستون بسیار کم هستند.

اما recall پایینی دارد؛ زیرا در سطر مربوط به این کتگوری علاوه بر happy؛ عکس هایی که واقعا happy بودن به ستون های دیگری ها هم اختصاص داده شدن (fn). به خاطر همین recall کاهش یافته است. برای بقیه کتگوری نیز میتوان به طور مشابه استدلال نمود.

F1 = 2 * (precision * recall) / (precision + recall)

فرمول f1 بدین شکل است تا در صورت کاهش شدید هز کدام از پارامتر ها مقدار این معیار به شپت کاهش یابد علت این امر هم ضربی هست که در صورت کسر وجود دارد.

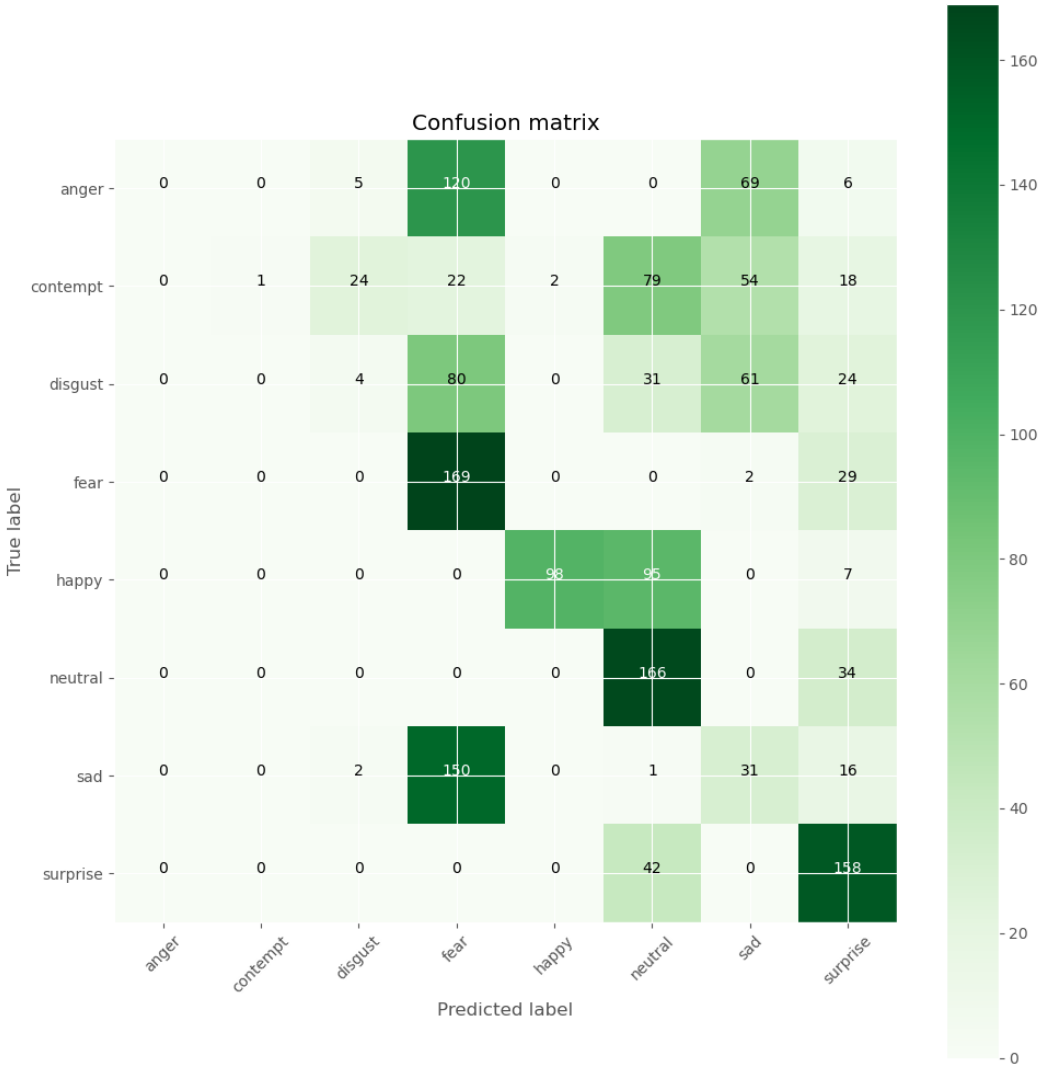
support به معنی وزن یا تعداد اون کتگوری میباشد. accuracy کلی همان طور که انتظار میرفت نزدیک به ۴۰(39) میباشد. میانگین وزن دار معیار های PRECISION , RECALL در سطر اخر مشاهده میشود.

این نشان دهنده این است؛ مدل اکثرا دارد FEAR,SUPRISE,NEUTRAL پیش بینی میکند؛ زیرا recall این کتگوری ها بالا میباشد اما دقت کمی دارند و گروه دیگر recall پایینی دارند که به معنی این هست که عکس های مربوط به این گروه ها به طور اشتباهی در ۳ گروه که بالاتر گفته شد؛ predict میشود.

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

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

Confusion Matrix –



6.AlexNet CM

هر چقدر رنگ خانه های قطر اصلی ماتریس پر رنگ تر باشد و در واقع مقدار آن ها بیشتر باشد؛ این بدین معناست که مدل ما دارد بهتر تخمین میزند. در اینجا suprise و neutral, fear را مدل از همه بهتر تشخیص میدهد. از هر کتگوری ۲۰۰ تا داریم. سطر افقی بیانگر لیبل واقعی داده های میباشد و مجموع هر سطر با یکدیگر برابر و برابر با ۲۰۰ میباشد. برای مثل می توان فهمید که مدل ما اکثرا دارد fear تخمین میزند به خاطر همین recall خوبی دارد. ولی precision خوبی ندارد. ولی برای مثال suprise هم recall و هم precision خوبی دارد. که ای مسئله باعث میشود که همزمان هم tp بالایی داشته باشیم و هم fp کمی؛ به خاطر همین auc این کتگوری بالا بود. موارد گفته شده در قسمت قبلی هم برقرار بود.

– Tune and Train:

همانند واقعیت و برای بهینگی مدل را یکبار با تعداد ایپاک مناسب ترین کردیم و سپس آن را ذخیره کردیم و بعد از آن: ابتدا مدل را لود میکنیم و سپس؛ evalution یا tuning انجام میدهیم.

```
# evalute after train:
with tf.device('/GPU:0'):
    results = model.evaluate(val_generator)
✓ 1.9s
25/25 [==============================] - 2s 71ms/step - loss: 0.4216 - accuracy: 0.8838
```

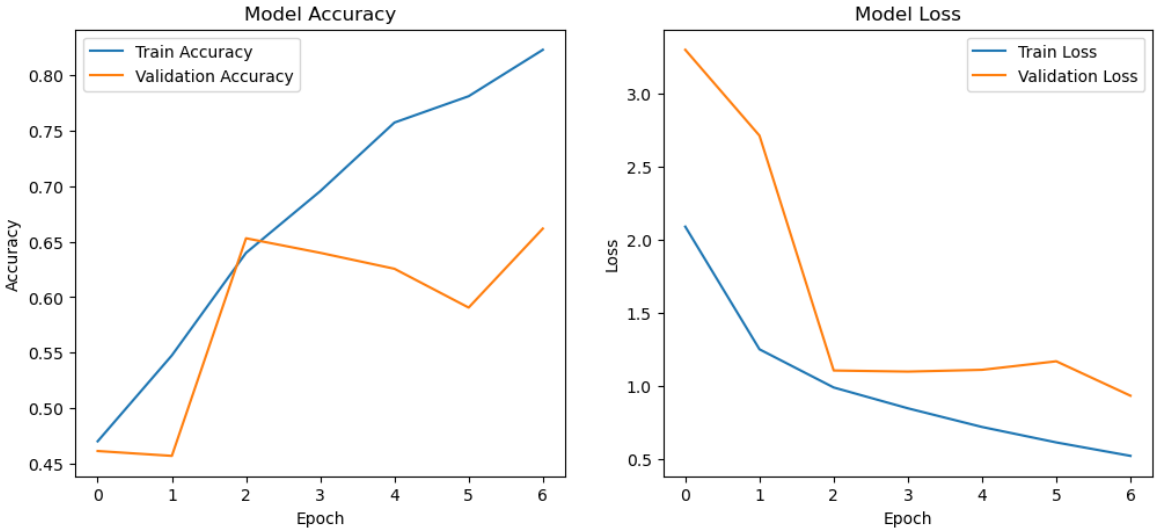
7.VGGNet accuracy on validation

```
- loss: 2.0904 - accuracy: 0.4700 - val_loss: 3.2990 - val_accuracy: 0.4613
- loss: 1.2522 - accuracy: 0.5475 - val_loss: 2.7123 - val_accuracy: 0.4569
- loss: 0.9914 - accuracy: 0.6400 - val_loss: 1.1075 - val_accuracy: 0.6531
- loss: 0.8490 - accuracy: 0.6956 - val_loss: 1.0999 - val_accuracy: 0.6400
- loss: 0.7214 - accuracy: 0.7575 - val_loss: 1.1119 - val_accuracy: 0.6256
- loss: 0.6160 - accuracy: 0.7812 - val_loss: 1.1704 - val_accuracy: 0.5906
- loss: 0.5239 - accuracy: 0.8231 - val_loss: 0.9351 - val_accuracy: 0.6619
```

۸.VGGNet accuracy on validation after tune

در اینجا نیز validation_accuracy مدل به ۷۰ رسید و بعد از پدیده overfit در حال رخ دادن است؛ پس متوقف میکنیم و بیشتر از آموزش نمیدهیم.

Loss and Accuracy –



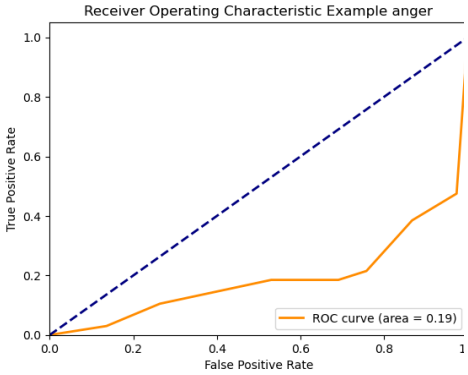
۹.VGGNet Loos & Accuracy

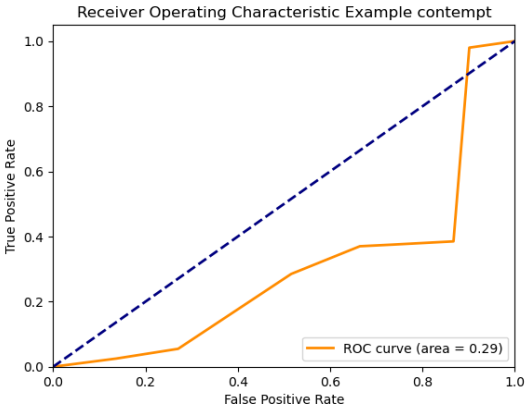
علت اینکهvalidation_accuracy پایین است یا oscillate میکند میتواند کمبود داده باشد.

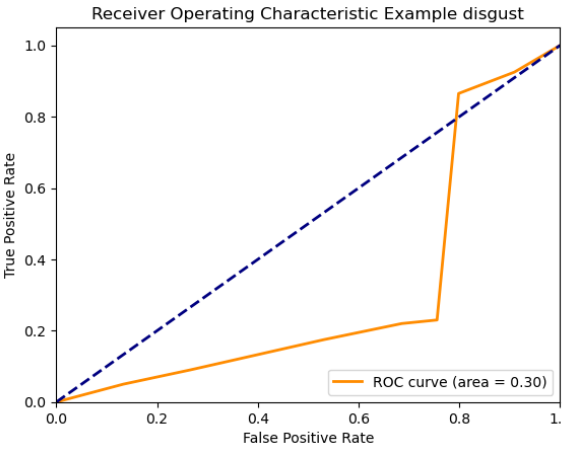
ولی همان طور که در نمودار مشاهده میکنید؛ قبل از اینکه پدیده overfit رخ بدهدُ مدل را متوقف میکنیم؛ زیرا اگر ادامه بدیم روی داده ی tune بخ دقت ۹۰۹ خواهیم رسید ولی val به ۵۰ و کمتر کاهش خواهد یافت(overfit)

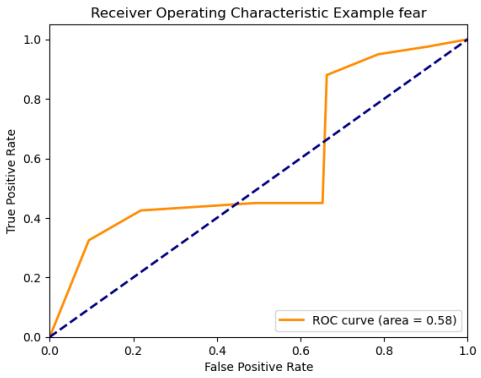
پس تا جایی که همروند هست و converge میکنند؛ ما آموزش را ادامه میدهیم. ادامه دادن آموزش باعث میشود loss داده validation بیشتر شود.

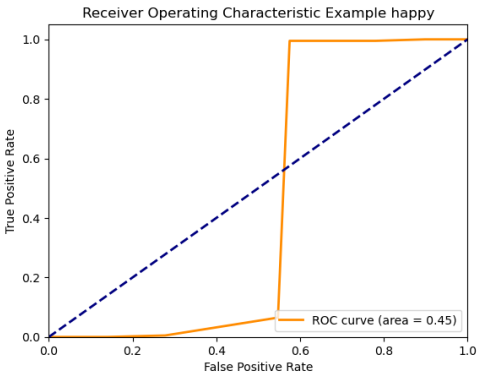
ROC –

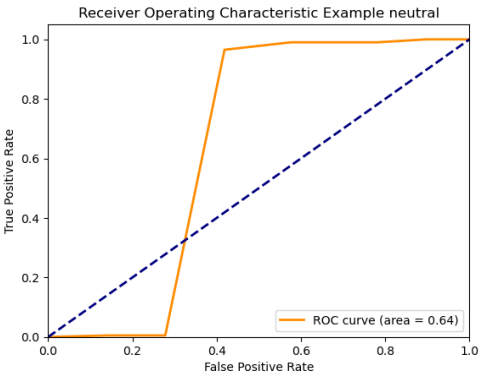


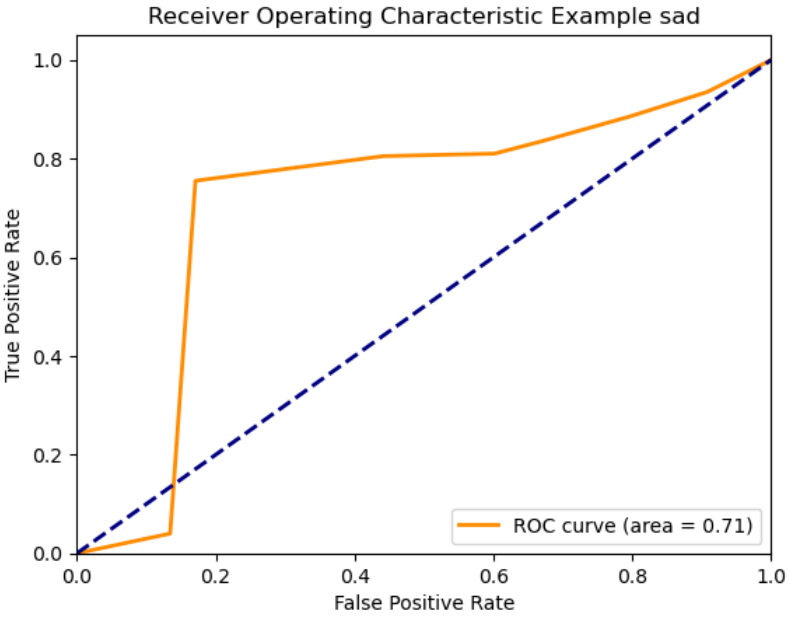


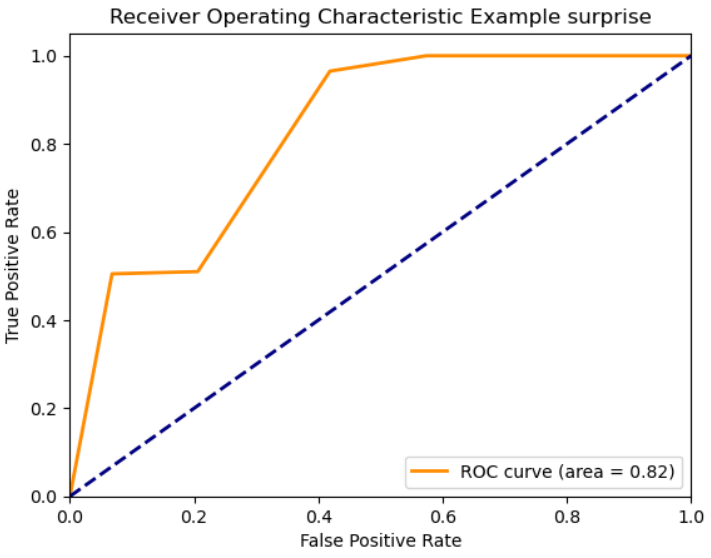












10.VGGNet ROC

برای مثال در اینجا مساحت زیر نمودار یا همان auc برای sad نزدیک به ۱ میباشد. کلا هر چقدر از خط y=x که بیانگر تخمین تصادفی هست فاصله داشته باشیم؛ نشان دهنده عملکر بهتر مدل ما میباشد. یعنی مدل توانسته است این کتگوری را از بقیه تشخیص دهد. انتظار میرود مدل suprise را هم به خوبی مدل کند. زیرا auc بالایی دارد نزدیک به ۱.

و همین طور ممکن است که happy را مدلی خیلی خوب بتواند classify کند. علت آن این است که در میانه ی بازه مدل جهش داشته و به شدت به خط y=1 نزدیک شده است. این بدین معنا است که قبل از اینکه treshold پایین بیاید بیش از حد مدل توانسته اکثر اعضای این گروه را تشخیص دهد. بدون آنکه fp (عامل کاهش precision)آن زیاد بشود. این یعنی اینکه هم recall و هم presicion بالایی خواهیم داشت.

- **F1, Recall, Precision**

	precision	recall	f1-score	support
anger	0.74	0.53	0.61	200
contempt	0.71	0.59	0.65	200
disgust	0.68	0.64	0.66	200
fear	0.86	0.43	0.57	200
happy	0.83	0.93	0.88	200
neutral	0.49	0.96	0.65	200
sad	0.74	0.71	0.73	200
surprise	0.52	0.51	0.51	200
accuracy			0.66	1600
macro avg	0.70	0.66	0.66	1600
weighted avg	0.70	0.66	0.66	1600

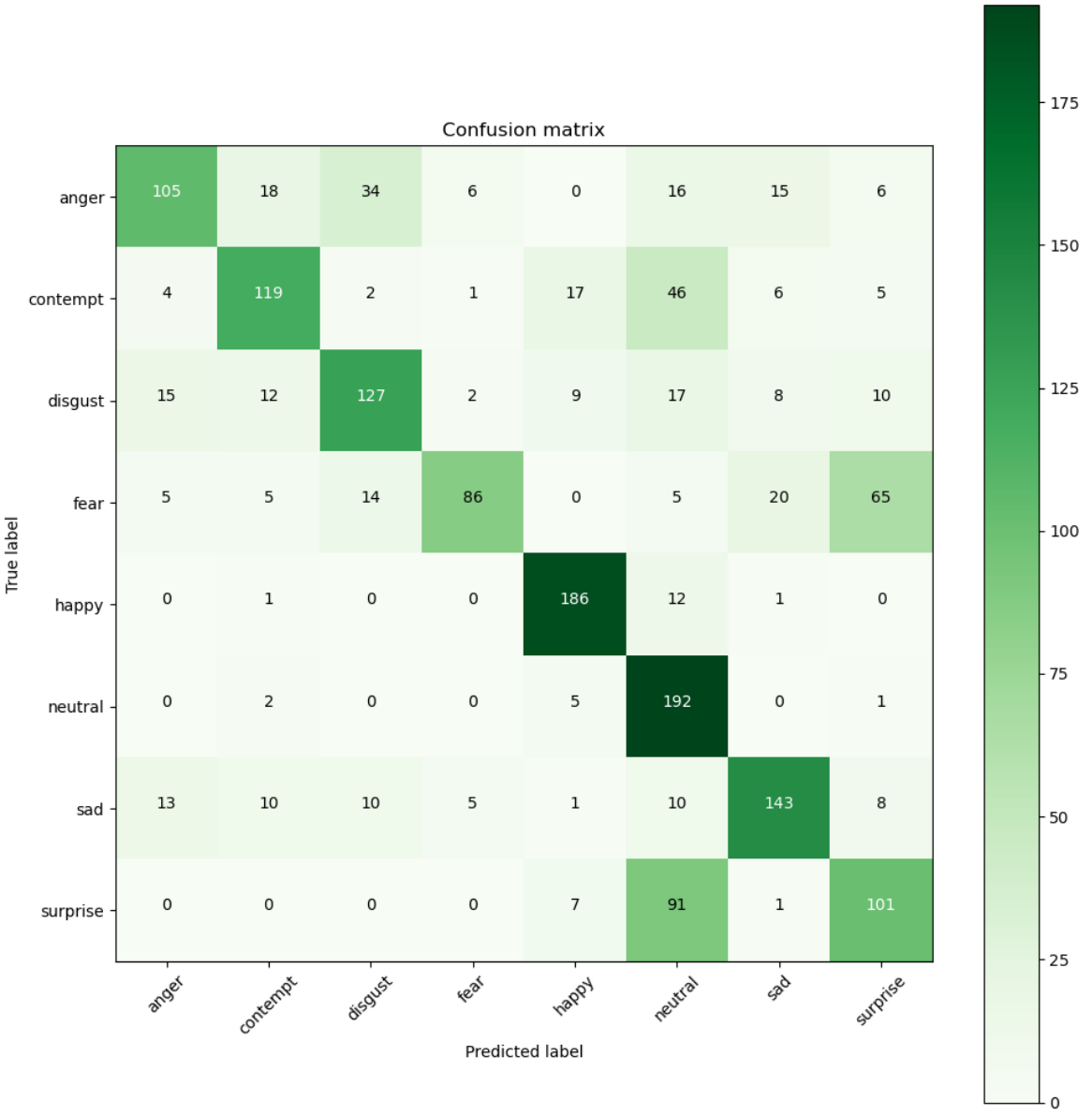
۱۱.VGGNet classification report

در اینجا fear بیشترین دقت را دارد: این بدین معنا استُ که اگر مدل میگویید fear به احتمال 86 درصد مدل درست تشخیص داده است. مدل اشتباهی عکس های دیگر را fear تشخیص نمیدهد. اما از طرقی recall اش پایین تر است. این نشان دهنده این است که مدل در تشخیص fear محتاطانه عمل مکیند و همه fear ها نمیتواند تشخیص بدهد و کلاس دیگری ب یک fear اختصاص میدهد. اما از طرفی fear دقت بالایی دارد.

بقیه کتگوری ها را میتوان بدین شکل استدلال کرد. کلا بالا بردن همزمان این دو پارامتر باهمدیگر کار سختی معمولا یکی را باید فدای دیگری کنیم بسته به شرایط. که میخواهیم مدل دقیق باشد یا recall بالایی داشته باشد. برای مثال میخواهیم یک تست همه بیماران را تشخیص دهد ولی اگر فردی مریض هم نبودُ بیمار تشخیص داده بشود برای ما ایرادی نداردو در اینجا ما recall بالا برای بیمار میخواهیم.تا همه آن ها را تشخیص دهیم.

در اینجا همان طور که گفته شد happy به خوبی عمل کرده است و f1 آن 0.88 میباشد.

Confusion Matrix –



۱۲.VggNet CM

هر چقدر accuracy مدل به طور کلی بالاتر میرود. قطر اصلی پر رنگ تر میشود. در واقع مجموع اعدادی که در قطر اصلی حضور دارند؛ برابر با تعداد عکس هایی هست که مدل آن ها را به درستی تشخیص داده است. وقتی مدل f1 بالایی برای یک کتگوری دارد. در cm تمام سطر و ستون مربوط به کتگوری خالی هست به جز خانه ای که مربوط به قطر اصلی است. همان طور که مشاهده میکنید happy دقیقا چنین ویژگی را دارد و از ۲۰۰ تا ۱۸۶ را تشخیص داده است. بقیه کتگوری ها مانند استدلالی که در سوال قبل کردیم میتوان تحلیل کرد. و همچنین neutral؛ recall بالایی دارد(0.96) میتوان این را در نمودار دید؛ در سطر مربوط به آن؛ ۱۹۲ تا ۲۰۰ ها را تشخیص داده است. و این بیانگر موضوع گفته شده است. کلا سطر بیانگر recall و ستون بیانگر precision میباشد. و هر چقدر تمرکز روی کتگوری مربوطه بیشتر باشد؛ بهتر است.

VGGNet vs AlexNet –

	precision	recall	f1-score	support
anger	0.74	0.53	0.61	200
contempt	0.71	0.59	0.65	200
disgust	0.68	0.64	0.66	200
fear	0.86	0.43	0.57	200
happy	0.83	0.93	0.88	200
neutral	0.49	0.96	0.65	200
sad	0.74	0.71	0.73	200
surprise	0.52	0.51	0.51	200
accuracy			0.66	1600
macro avg	0.70	0.66	0.66	1600
weighted avg	0.70	0.66	0.66	1600

	precision	recall	f1-score	support
anger	0.00	0.00	0.00	200
contempt	1.00	0.01	0.01	200
disgust	0.11	0.02	0.03	200
fear	0.31	0.84	0.46	200
happy	0.98	0.49	0.65	200
neutral	0.40	0.83	0.54	200
sad	0.14	0.15	0.15	200
surprise	0.54	0.79	0.64	200
accuracy			0.39	1600
macro avg	0.44	0.39	0.31	1600
weighted avg	0.44	0.39	0.31	1600

13.VggNet vs AlexNet

همان طور که مشاهده میکنید؛ این مدل نسبت به قبلی بهتر عمل کنید. تقریبا accuracy آن ۱.۸ برابر alexnet هست.

علت این در تفاوت در ساختار این دو مدل هست. مدل vgg پارامتر های بیشتری دارد و عمق بیشتری دارد. این دو باعث میشود که مدل ظرفیت بالاتری را داشته باشد. و نکته دیگر استفاده کردن از relu هست.(البته در این پیاده سازی alexnet مقاله نیز از relu استفاده شده است). vgg به علت اینکه پارامتر های زیادی دارد ممکن است overfit کند و این به طور کلی از مشکلات مدلی هایی هست که ظرفیت بالایی دارند. که البته با data aument و روش دیگر این مشکل را میتوان حل کرد. پس در اینجا (همانند case های دیگر) مدل vgg خیلی بهتر از alexnet عمل میکند و کاملا این قابلیت را دارد روی داده ترین به ۱۰۰ درصد برسد. که این خطر overfit را دارد ولی همان طور گفته شد؛ میتوان با روش هایی این مسله را هندل کرد و به نتایج خیلی خوبی با این دست یافت.

Type	Shape	Output
2×Conv	$3 \times 3 \times 16$	$128 \times 128 \times 16$
MaxPool	2×2	$64 \times 64 \times 16$
2×Conv	$3 \times 3 \times 32$	$64 \times 64 \times 32$
MaxPool	2×2	$32 \times 32 \times 32$
2×Conv	$3 \times 3 \times 64$	$32 \times 32 \times 64$
MaxPool	2×2	$16 \times 16 \times 64$
2×Conv	$3 \times 3 \times 128$	$16 \times 16 \times 128$
MaxPool	2×2	$8 \times 8 \times 128$
2×Conv	$3 \times 3 \times 128$	$8 \times 8 \times 128$
MaxPool	2×2	$4 \times 4 \times 128$
Flatten	2048	–
2×Dense	1024	–
Dense	8 or 2	1 label or 2 floats

Type	Shape	Output
Conv	$9 \times 9 \times 16$	$128 \times 128 \times 16$
MaxPool	2×2	$64 \times 64 \times 16$
Conv	$7 \times 7 \times 32$	$64 \times 64 \times 32$
MaxPool	2×2	$32 \times 32 \times 32$
Conv	$5 \times 5 \times 64$	$32 \times 32 \times 64$
MaxPool	2×2	$16 \times 16 \times 64$
Conv	$3 \times 3 \times 128$	$16 \times 16 \times 128$
MaxPool	2×2	$8 \times 8 \times 128$
Conv	$3 \times 3 \times 128$	$8 \times 8 \times 128$
MaxPool	2×2	$4 \times 4 \times 128$
Flatten	2048	–
2×Dense	1024	–
Dense	8 or 2	1 label or 2 floats

14.VggNet(left) vs Alexnet

:Tune and Train –

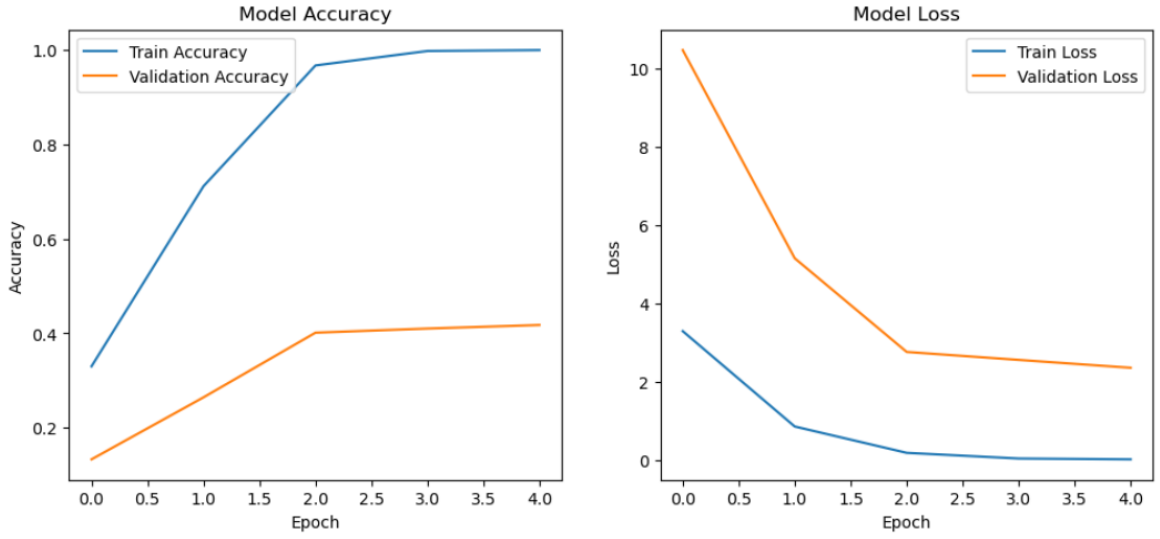
```
# evalute after train:
with tf.device('/GPU:0'):
    results = model.evaluate(val_generator)
```

```
25/25 [==============================] - 2s 77ms/step - loss: 1.0509 - accuracy: 0.7781
```

```
 - loss: 3.2895 - accuracy: 0.3300 - val_loss: 10.4672 - val_accuracy: 0.1331
 - loss: 0.8540 - accuracy: 0.7113 - val_loss: 5.1511 - val_accuracy: 0.2644
 - loss: 0.1812 - accuracy: 0.9669 - val_loss: 2.7584 - val_accuracy: 0.4013
 - loss: 0.0367 - accuracy: 0.9975 - val_loss: 2.5550 - val_accuracy: 0.4100
 - loss: 0.0154 - accuracy: 0.9994 - val_loss: 2.3540 - val_accuracy: 0.4175
```

15.MobileNet accuracy after tuning and training

Loss and Accuracy –

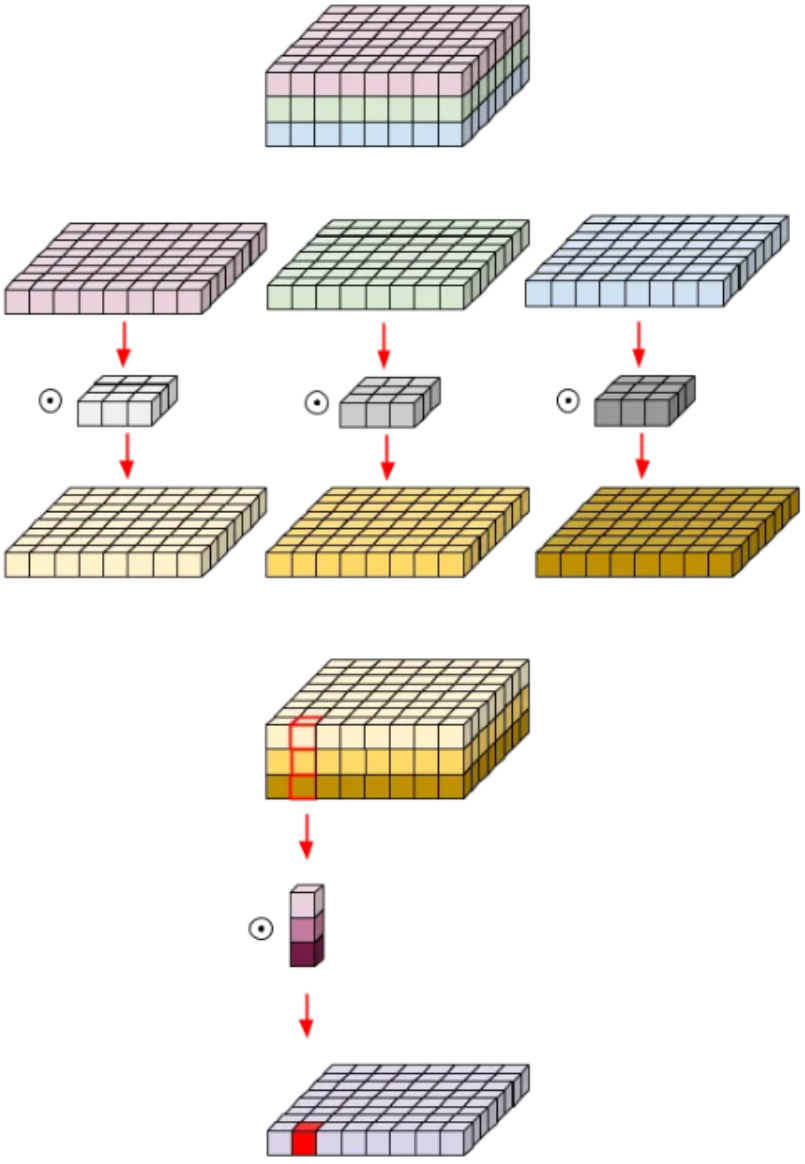


16.MobileNet accuracy

MobileNet vs VGGNet & AlexNet –

تفاوت این مدل در نحوه محاسبه کانولوشن میباشد. در واقع ما به جای استفاده از لایه conv معمولی از لایه

dconv استفاده میکنیم و همان طوری که در این لینک توضیح داده شده است. این روش باعث کاهش پارامتر ها میشود. و از overfiting جلو گیری میکند. زیرا لایه conv معمولی تعداد بسیاری پارامتر دارد و این مسئله باعث افزایش احتمال overfit، میشود. Depth-wise convolution یا depth-wise separable در واقع اسم کامل لایه dconv میباشد. و فرایند محاسبه کانولوشن توسط این لایه در عکس زیر آورده شده است. این معماری عمق بیشتری نسبت به لایه های قبلی دارد همین باعث شده است که مدل به خوبی یاد بگیرد. از انجایی که این مدل پارامتر های کمتری دارد در نتیجه آموزش آن اسان تر است. و محاسبات کمتری نیاز دارد. ولی همان طوری که گفته شد پارامتر های کمتری دارد شاید این مسئله در موادری باعث شود که از لحاظ ظزفیت از vgg کمتر باشد و کم پایین تر از آن عمل بکند.



17.Dconv

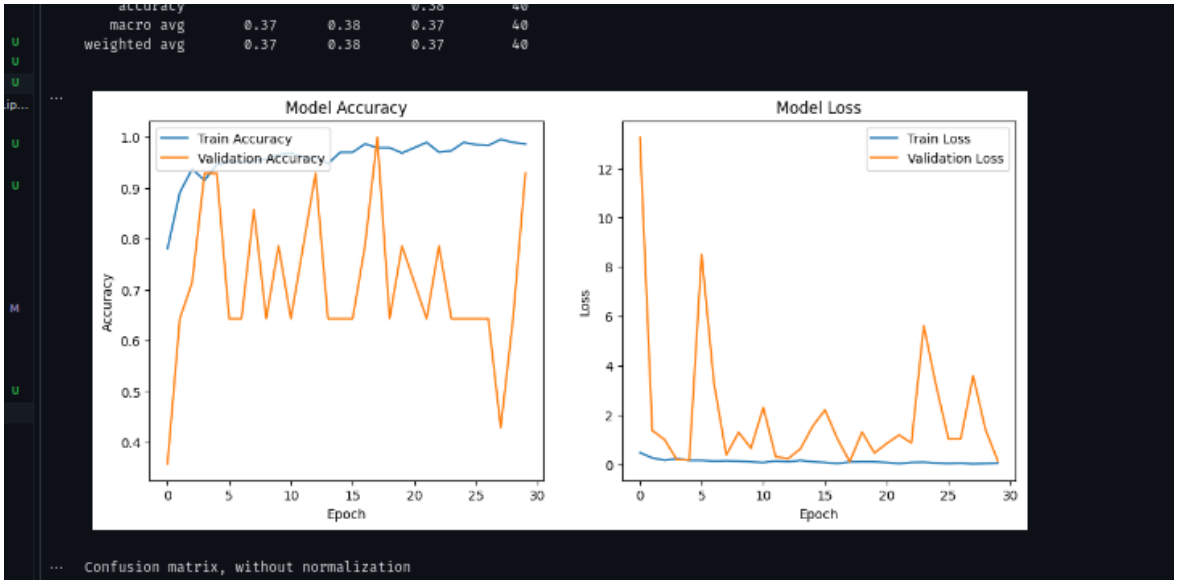
پاسخ۲-تشخیص بیماران مبتالا به کووید با استفاده از عکس ریه

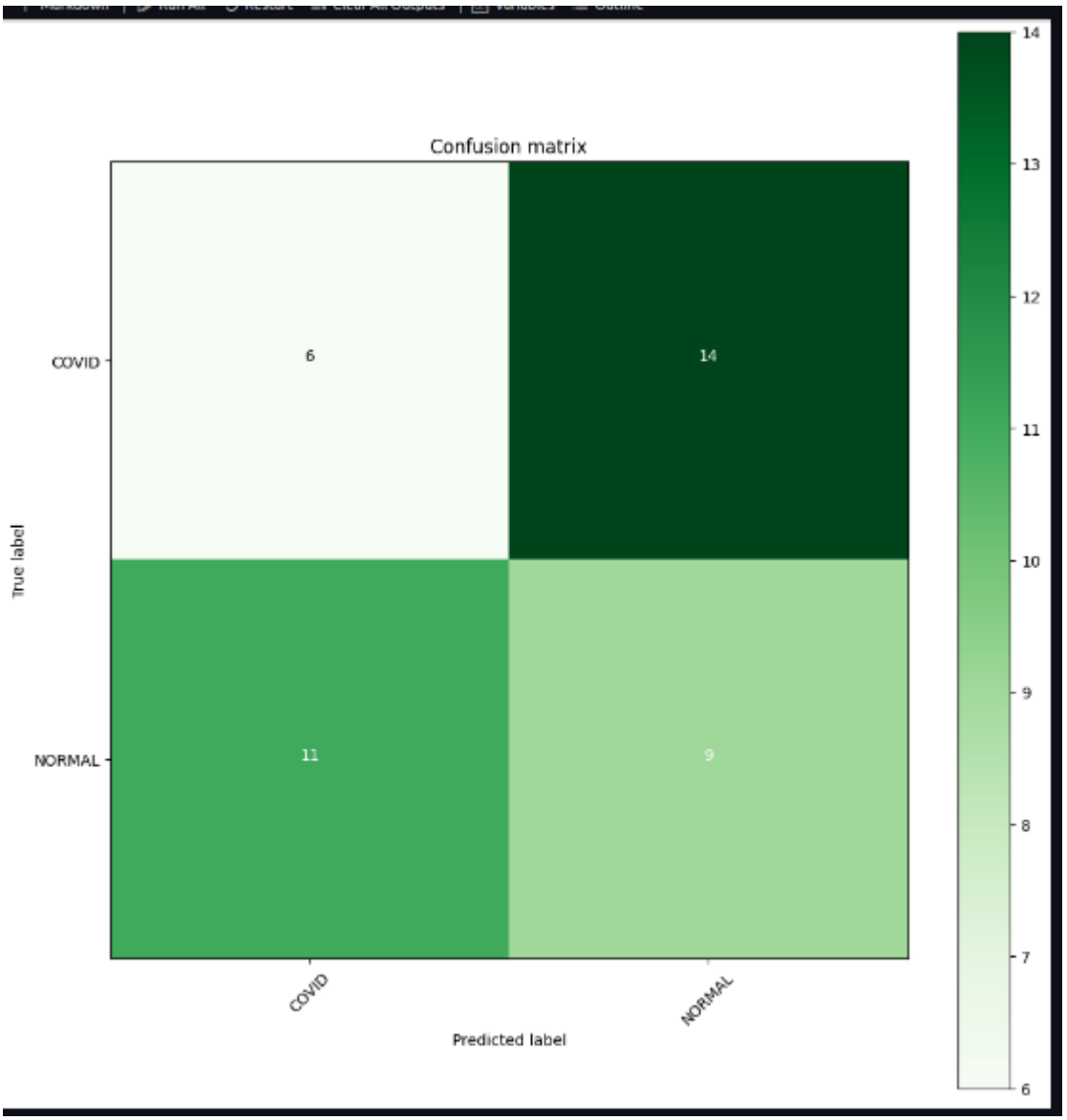
2-1.Data Preprocessing

مستقیم همه نوع augmentation پیشنهاد شده در مقاله را انجام داده ایم که شامل یک Horizental Flip و همچنین ۳ rotation است که زاویه های ۹۰ و ۱۸۰ و ۲۷۰ را شامل می شود.

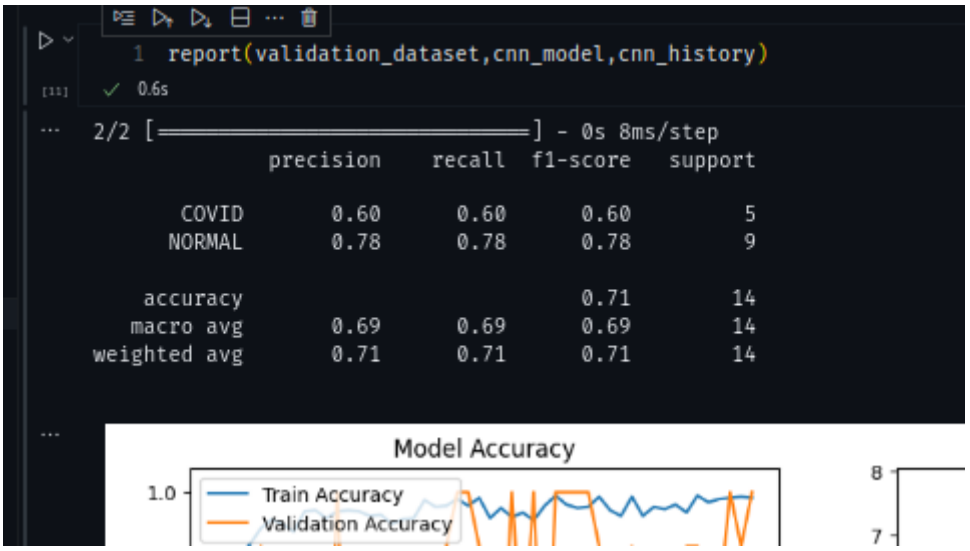
2-2.Train

نایج روی



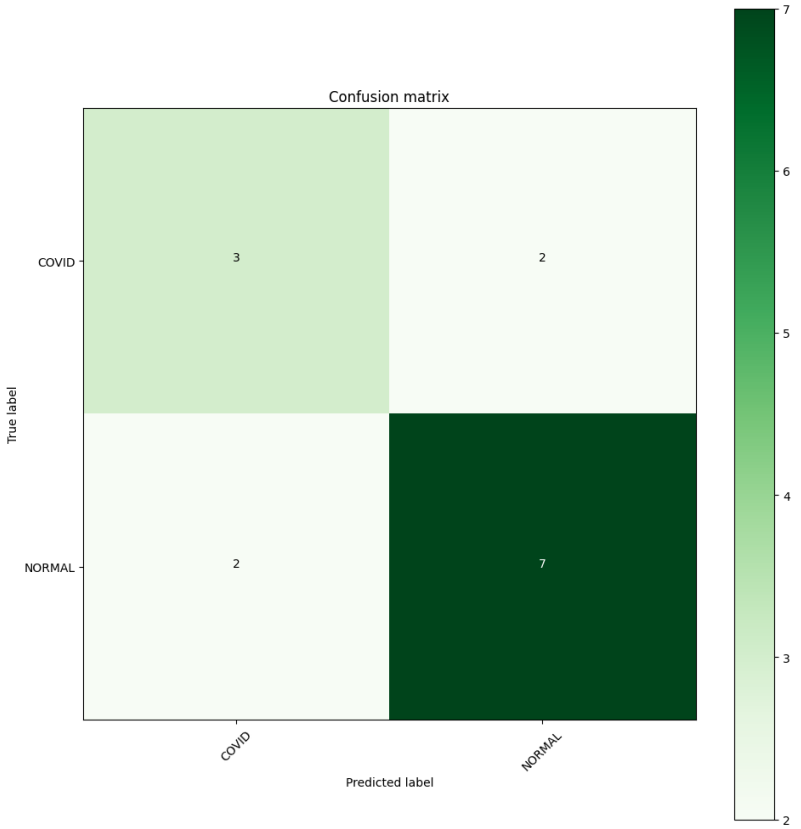


در زیر نتایج بر روی دادگان validation دیده می شود

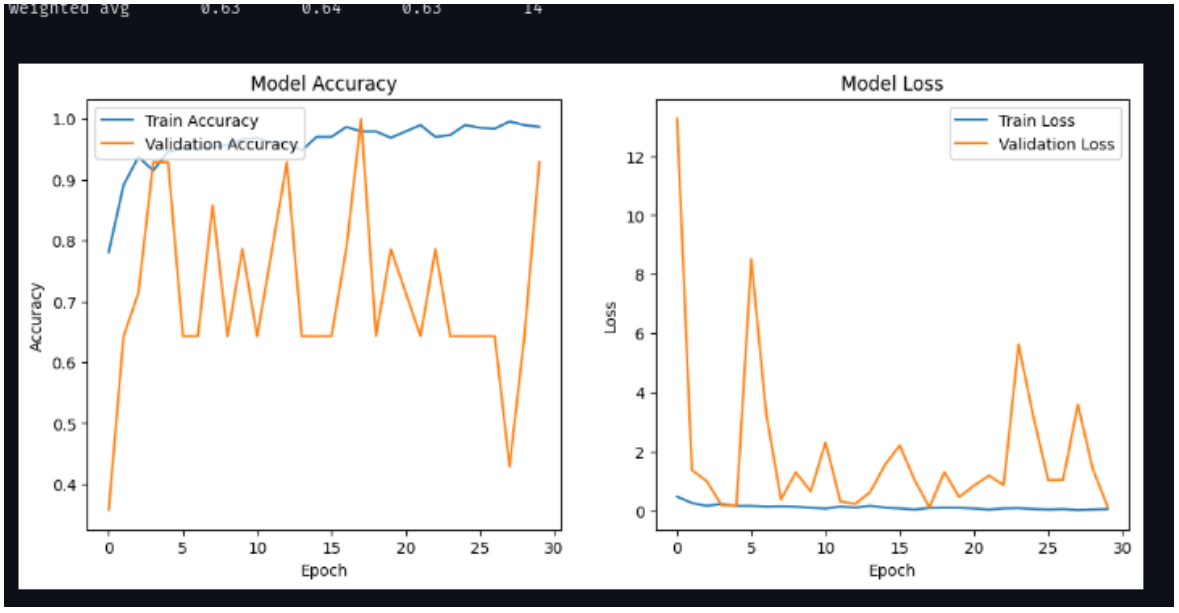


16.Covid 19 Model Accuracy

در عکس زیر Confusion Matrix مربوط به Validation Dataset را مشاهده می کنیم



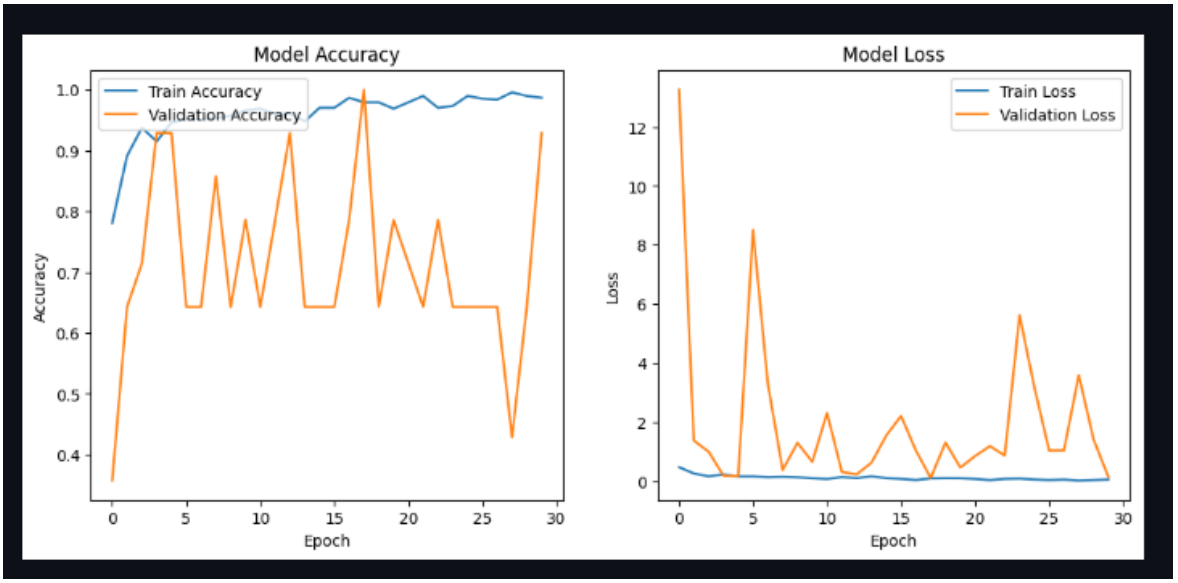
16.Covid 19 Validation Confusion Matrix



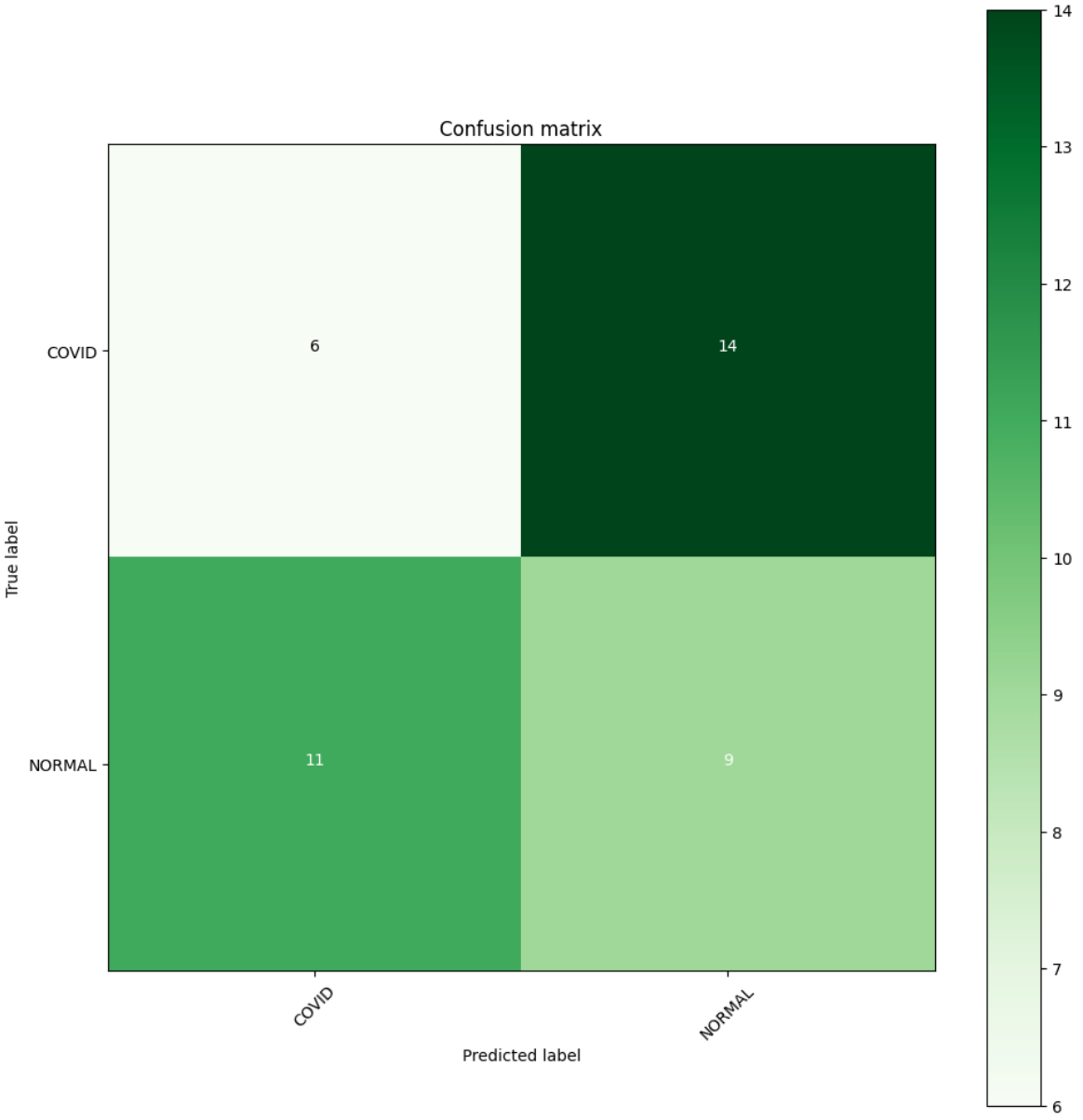
.17

2-3.**Test**

عملکرد روی داده های تست




```
[24]  ✓ 0.7s

...  4/4 [==============================] - 0s 9ms/step
                precision    recall  f1-score   support

         COVID       0.35      0.30      0.32        20
        NORMAL       0.39      0.45      0.42        20

      accuracy                           0.38        40
     macro avg       0.37      0.38      0.37        40
  weighted avg       0.37      0.38      0.37        40
```



قابل برداشت است confusion matrix از روی specificity