به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین دوم

نام و نام خانوادگی	مهسا همت‌پناه – محمد هادی بابالو
شماره دانشجویی	۸۱۰۱۹۹۵۸۴ - ۸۱۰۱۹۹۳۸۰
تاریخ ارسال گزارش	۱۴۰۲,۰۹,۰۳

فهرست

شکل‌ها

جدول‌ها

پاسخ ۱. تجزیه وتحلیل احساسات صورت مبتنی بر CNN

۱-۱. معرفی مقاله

۱-۲. پیش پردازش تصاویر و Data Augmentation

برای پیش پردازش داده ها ابتدا همه تصاویر چهره به اندازه ۱۲۸×۱۲۸ پیکسل درآمده اند و تمام داده های تصویر از [۰، ۲۵۵] به [۰، ۱] نرمال می شده اند.

همچنین در این بخش از ۳ نوع data augmentation متفاوت برای تولید تصاویر بیشتر برای آموزش و افزایش تعمیم پذیری مدل استفاده شد.

- چرخش تصاویر تا ۲۰ درجه با استفاده از RandomRotation(degrees=20)
- Translation تا ۱۰٪ (در هر دو جهت y و x) با استفاده از

RandomAffine(degrees=0, translate=(0.1, 0.1))

- چرخش در جهت X با استفاده از RandomHorizontalFlip()

در ادامه نیز مجموعه داده به نسبت ۲۰٪ به مجموعه های آموزش و ارزیابی تقسیم شده است.

۱-۳. پیاده سازی مدل AlexNet

این معماری به شکل زیر است.

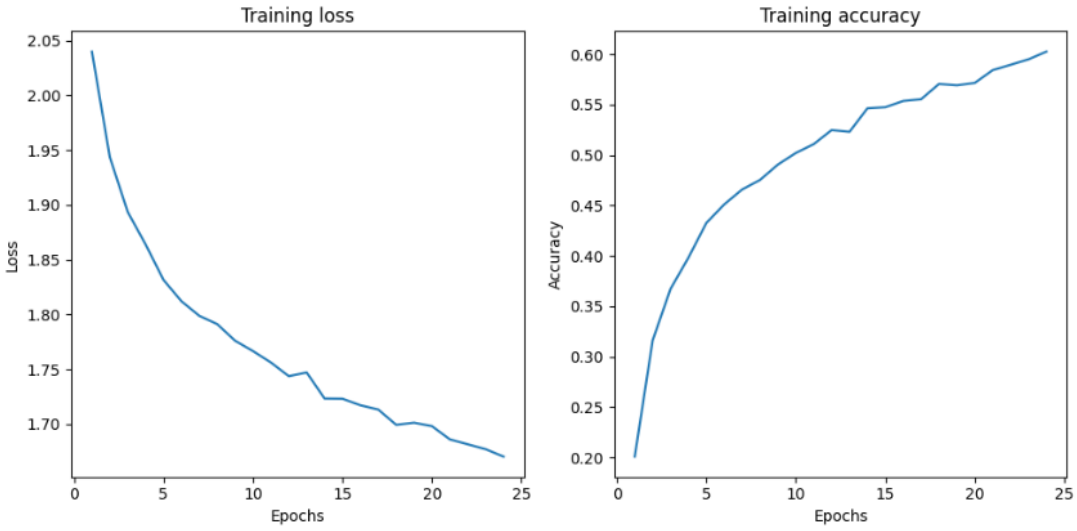
جدول ۱=۱ - معماری AlexNet

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

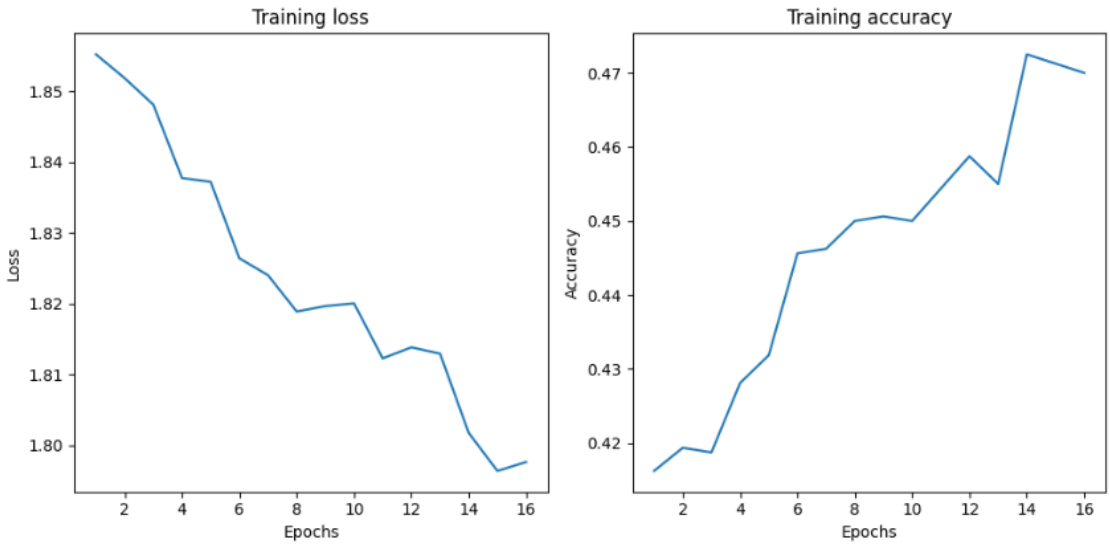
ابتدا مدل را آموزش می دهیم. برای این کار از پارامتر های ذکر شده در مقاله استفاده کرده ایم مانند استفاده از ۲۴ ایپاک برای آموزش و ۱۶ ایپاک برای fine-tune. برای محاسبه خطا از Weighted-loss و برای optimiser هم از Adam استفاده شده است. برای fine-tune کردن نیز از مدل ترین شده با همین پارامتر ها استفاده شده است. تنها برای دست یابی به نتیجه بهتر مقدار learning rate از 0.001 به 0.0001 کاهش یافته است.

همچنین برای fine-tune کردن مدل به این صورت عمل کردیم که مدل را بعد از آموزش با دستور torch.save(model, 'AlexNet.pth') ذخیره کرده و سپس برای fine-tune کردن لود کرده ایم و لایه‌های convolutional را freeze کرده و صرفا وزن‌های شبکه fully-connected را مورد آموزش بیشتر با مجموعه داده جدید tune قرار دادیم.

شکل شماره ۱-۱ نشان دهنده نمودارهای loss و accuracy در ایپوک های مختلف برای داده train است و شکل ۱-۲ نشان دهنده نمودارهای loss و accuracy در ایپوک های مختلف برای داده tune است .



شکل ۱-۱. **نمودارهای loss و accuracy در ایپوک های مختلف برای داده train**



شکل ۱-۲. **نمودارهای loss و accuracy در ایپوک های مختلف برای داده tune**

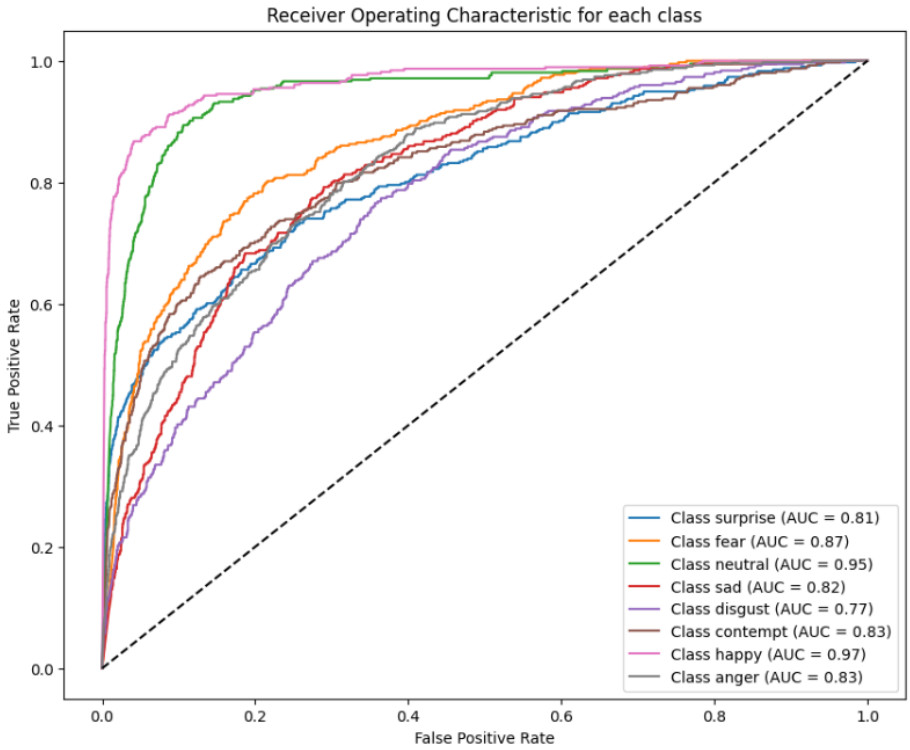
همانطور که در شکل ها مشخص است خطا و دقت با شیب و سرعت خوبی به ترتیب کاهش و افزایش می یابند و در ایپاک های آخر آموزش، متوقف می شود. در آخرین ایپاک به مقادیر زیر برای دقت و خطا برای داده های آموزش و tune می رسیم.

```
Train accuracy: 60.26%
Train loss: 1.6702

Tune accuracy: 47.00%
Tune loss: 1.7977
```

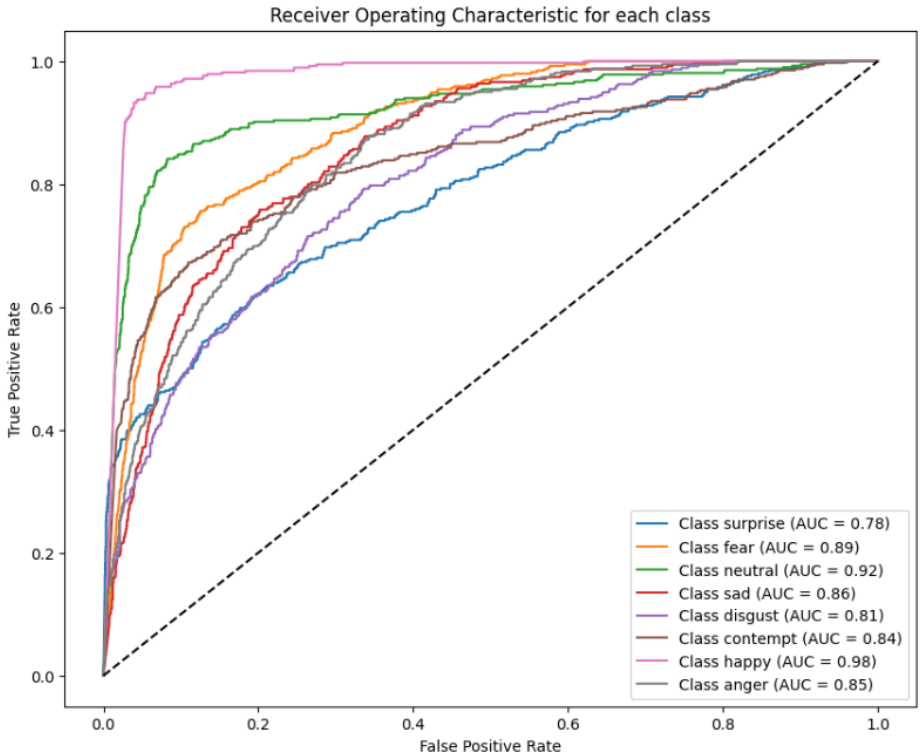
برای داده های تست نیز به ترتیب بعد از آموزش و fine-tune به مقادیر زیر برای دقت و خطا می رسیم. مقدار دقت داده تست بعد از انجام fine-tune حدود ۳ درصد افزایش داشته است و خطا نیز کاهش یافته است.

```
Test accuracy after training: 54.39%
Test loss after training: 1.7235

Test accuracy after finetuning: 57.84%
Test loss after finetuning: 1.6898
```

شکل شماره ۱-۳ و ۱-۴ نشان دهنده نمودار ROC برای کلاس های مختلف به ترتیب بعد از انجام training و finetuning هستند.



شکل ۱-۳. **نمودار ROC پس از آموزش مدل**



شکل ۱-۴. **نمودار ROC پس از fine-tune مدل**

همانطور که در شکل ها مشخص است در هر دو نمودار احساسات خوشحالی (happy) بهترین عملکرد را دارد و اندازه عددی AUC و یا همان مساحت زیر نمودار ROC در بعد از fine-tune کردن مدل به عدد 0.98 رسیده است که بیانگر آن است که تست ما تا حد زیادی دارای عملکرد مناسبی در شناسایی این احساس از سایر احساسات است. در مقابل احساسات surprise و disgust دارای کمترین AUC هستند.

جدول ۱-۲ و ۱-۳ نشان دهنده مقادیر Precision, Recall و F1 برای هر کلاس در بعد از انجام training و finetuning هستند. احساسات به شکل روبرو هستند: Neutral, Happy, Sad, Surprised, Afraid, Disgusted, Angryو Contemptuous.

جدول ۱-۲ – مقدار معیار ها پس از آموزش مدل

	precision	recall	f1-score	support
Su	0.81	0.34	0.48	416
Af	0.56	0.61	0.58	394
N	0.67	0.79	0.72	416
Sa	0.47	0.51	0.49	385
D	0.44	0.38	0.41	376
C	0.55	0.62	0.58	403
H	0.70	0.95	0.80	385
An	0.50	0.43	0.46	422
accuracy			0.58	3197
macro avg	0.59	0.58	0.57	3197
weighted avg	0.59	0.58	0.57	3197

جدول ۱-۳ – مقدار معیار ها پس از fine-tune مدل

	precision	recall	f1-score	support
Su	0.74	0.41	0.53	416
Af	0.52	0.58	0.55	394
N	0.64	0.80	0.71	416
Sa	0.36	0.55	0.43	385
D	0.48	0.22	0.30	376
C	0.54	0.53	0.53	403
H	0.91	0.72	0.80	385
An	0.41	0.54	0.47	422
accuracy			0.54	3197
macro avg	0.57	0.54	0.54	3197
weighted avg	0.58	0.54	0.54	3197

- Precision

این مقدار نسبت true positives به مجموع true positives و false positives است و هرچه مواردی که مدل به غلط پیش بینی کرده است که نسبت به پیش بینی‌های درست بیشتر باشد مقدار Precision کمتر خواهد شد. این مقدار در هر دو حالت برای کلاس های surprise و happy با اختلاف زیادی، بیشتر از بقیه کلاس ها است. این نشان دهنده این است که این مدل به ندرت احساسی را به‌عنوان surprise و happy طبقه‌بندی می‌کند، در حالی که آنها کلاس درست داده دسته بندی شده نیستند .کمترین مقدار بعد از آموزش برای احساس disgust با تنها 0.44 و بعد از finetuning برای احساس sad با مقدار 0.38 است.

- Recall

این معیار به دنبال محاسبه‌ی پوشش بر روی کل داده‌هاست و نسبت true positives به مجموع true positives و false negatives است. کلاس happy برای مدل آموزش داده شده با ۹۱ درصد بهترین عمکرد را داشته است و کلاس neutral برای مدل fine-tuned با ۸۰ درصد و بعد از آن کلاس happy با مقدار ۷۲ درصد است. پس می توان نتیجه گرفت این مدل بخش زیادی از داده های happy و neutral را به درستی شناسایی می کند.

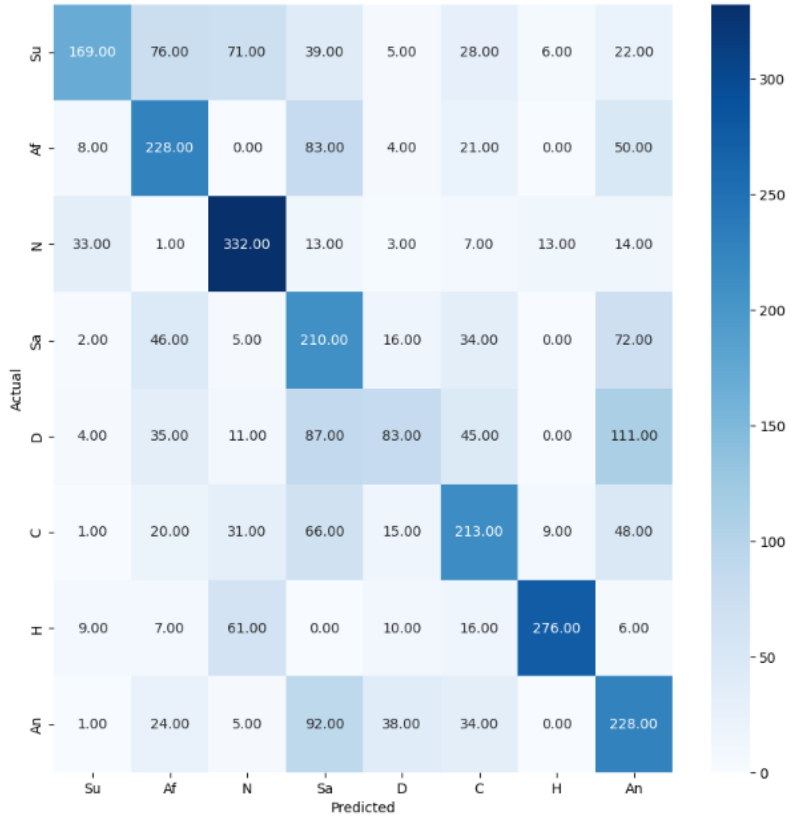
دو کلاس surprised و disgusted نیز کمترین درصد را برای recall دارند که مدل تعداد زیادی از داده های این کلاس ها را به اشتباه در کلاس های دیگری جز disgusted پیش بینی کرده است.

- F1

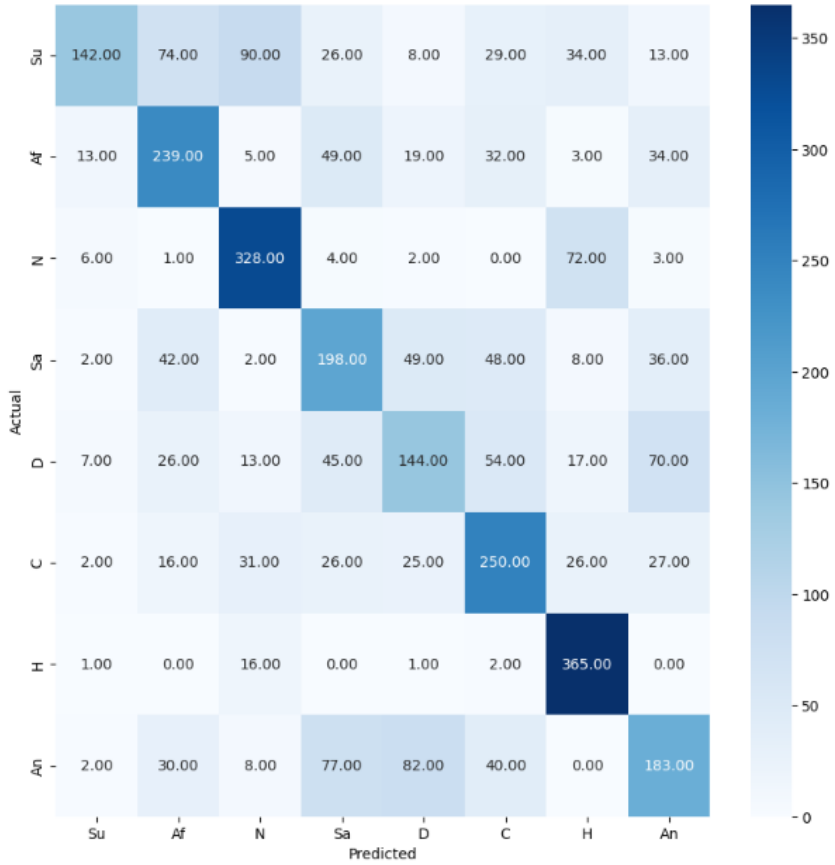
این معیار ارزیابی، میانگین هارمونیک از دو مورد قبلی یعنی Recall و Precision است. این معیار برای زمانی که تعداد داده های کلاس ها یکسان نیست بسیار کاربردی است. در این تمرین نیز تعداد داده ها در کلاس ها برابر نبوده اند. در این معیار نیز مانند معیار قبل مدل بهترین عملکرد را در neutral و happy دارد و بدترین عملکرد را در تشخیص کلاس های angry و sad.

جدول ۱-۴و ۱-۵ نشان دهنده ماتریس confusion برای هر کلاس در بعد از انجام training و finetuning هستند.

جدول ۱-۴ – ماتریس confusion پس از آموزش مدل



جدول ۱-۵ – ماتریس confusion پس از fine-tune مدل



همان طور که در جدول ها مشخص است و در معیار های قبلی نیز مشاهده شد دو مدل برای احساسات neutral و happy بهترین عملکرد را دارد و داده های این کلاس ها را با درصد بالایی به درستی دسته بندی می کند. در مقابل تشخیص احساسات surprised وdisgusted و تفکیک آن ها از احساسات دیگر برای مدل دشوار است. با توجه به نتایج به نظر میرسد تشخیص احساس disgust و angry برای مدل کار دشواری است و این دو احساس را در بسیاری از موارد به جای هم تشخیص داده است. همانطور که در مقاله هم به این مورد اشاره شده است ، خشم و انزجار به دلیل واحدهای کنش(action unit) صورت مشترکی که دارند، اشتباه گرفته می شوند.

همچنین با توجه به جدول های بالا، به طور کلی عملکرد مدل در تشخیص درست کلاس ها و دسته بندی درست پس از fine-tuning بهتر شده است.

پاسخ ۲ – پیاده‌سازی مدل VGGNet

۲-۱. مدل VGGNet

معماری این مدل به شکل زیر است.

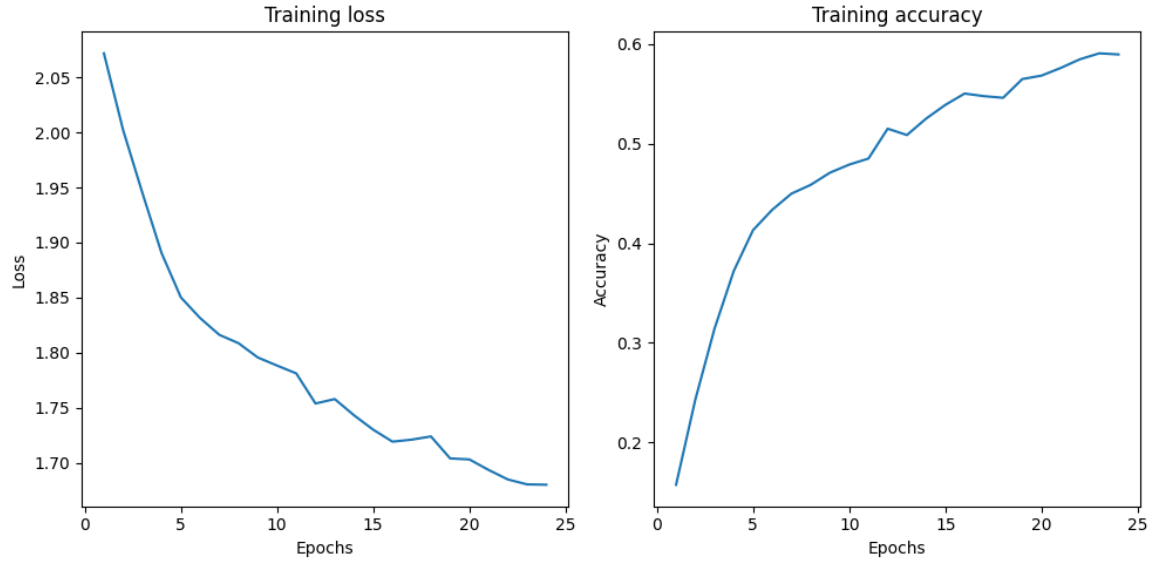
جدول ۲=۱ - معماری **VGGNet**

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

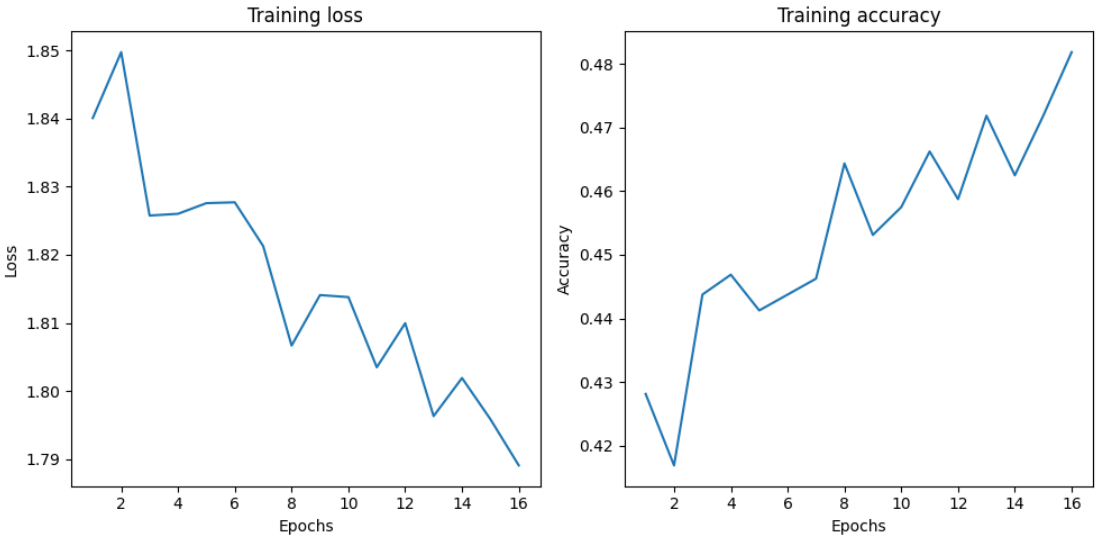
در این تمرین نیز ابتدا مدل را آموزش می دهیم. برای این کار از پارامتر های ذکر شده در مقاله استفاده کرده ایم مانند استفاده از ۲۴ ایپاک برای آموزش و ۱۶ ایپاک برای fine-tune. برای محاسبه خطا از Weighted-loss و برای optimiser هم از Adam استفاده شده است. برای fine-tune کردن نیز از مدل ترین شده با همین پارامتر ها استفاده شده است. تنها برای دست یابی به نتیجه بهتر مقدار learning rate از 0.001 به 0.0001 کاهش یافته است.

همچنین برای fine-tune کردن مدل به این صورت عمل کردیم که مدل را بعد از آموزش با دستور torch.save(model, 'VGGNet.pth') ذخیره کرده و سپس برای fine-tune کردن لود کرده ایم و لایه‌های convolutional را freeze کرده و صرفا وزن‌های شبکه fully-connected را مورد آموزش بیشتر با مجموعه داده جدید tune قرار دادیم.

شکل شماره ۲-۱ نشان دهنده نمودارهای loss و accuracy در ایپوک های مختلف برای داده train است و شکل ۲-۲ نشان دهنده نمودارهای loss و accuracy در ایپوک های مختلف برای داده tune است .



شکل ۲-۱. **نمودارهای loss و accuracy در ایپوک های مختلف برای داده train**



شکل۲-۲. **نمودارهای loss و accuracy در ایپوک های مختلف برای داده tune**

همانطور که در شکل ها مشخص است خطا و دقت با شیب و سرعت خوبی به ترتیب کاهش و افزایش می یابند و در ایپاک های آخر آموزش، متوقف می شود. در آخرین ایپاک به مقادیر زیر برای دقت و خطا برای داده های آموزش و tune می رسیم.

```
Train accuracy: 58.97%
Train loss: 1.6800
```

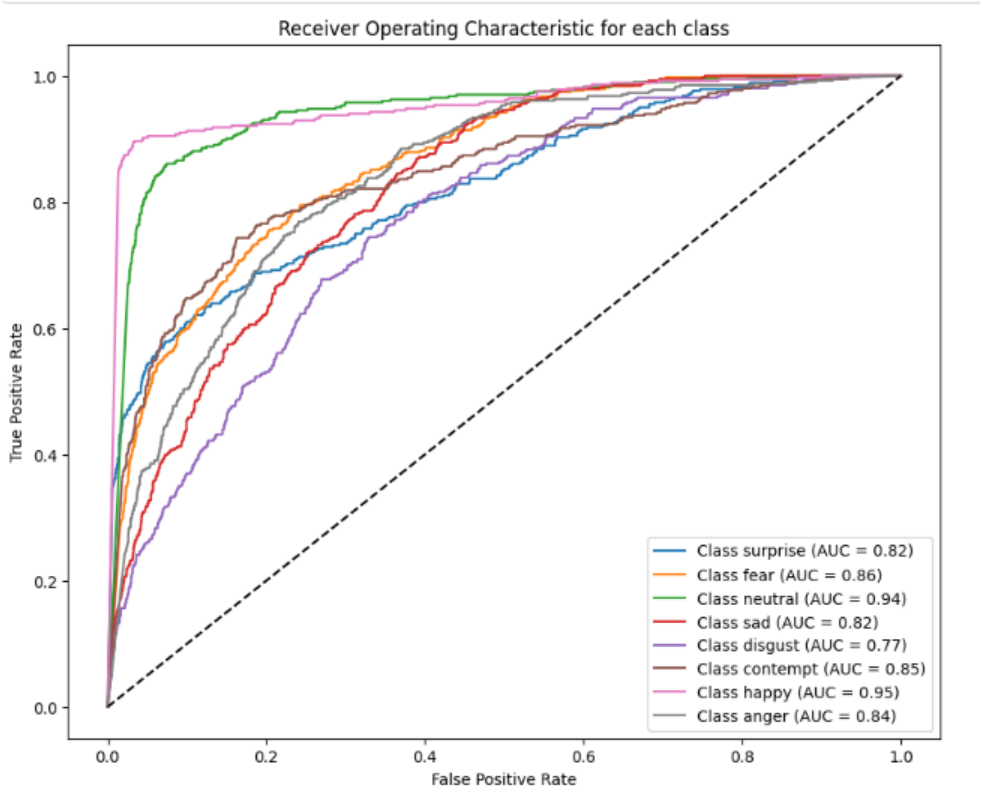
```
Tune accuracy: 48.19%
Tune loss: 1.7891
```

برای داده های تست نیز به ترتیب بعد از آموزش و fine-tune به مقادیر زیر برای دقت و خطا می رسیم.
مقدار دقت داده تست بعد از انجام fine-tune درصد کمی افزایش داشته است و خطا نیز کاهش یافته است

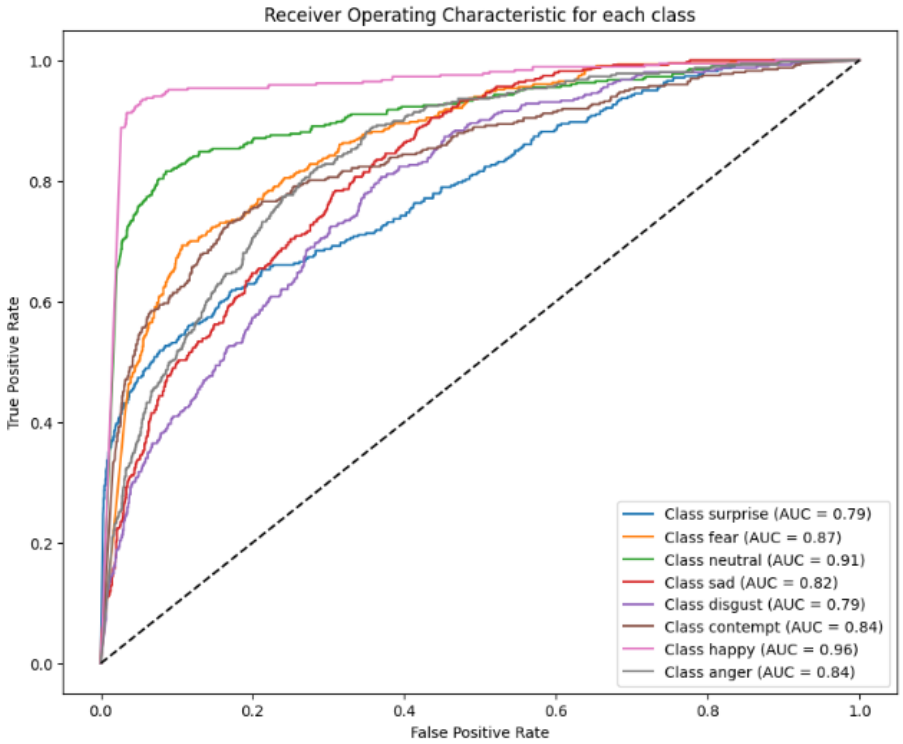
همچنین در مقایسه با مدل AlexNet به دقت بالاتر و خطای کمتری برای داده تست برای مدل آموزش داده رسیدیم و در مدل fine-tuned نتایج تقریبا مشابه هستند.

```
Test accuracy after training: 56.08%
Test loss after training: 1.7118

Test accuracy after finetuning: 56.49%
Test loss after finetuning: 1.7074
```

شکل شماره ۲-۳ و ۲-۴ نشان دهنده نمودار ROC برای کلاس های مختلف به ترتیب بعد از انجام training و finetuning هستند.



شکل ۲-۳. **نمودار ROC پس از آموزش مدل**



شکل ۲-۴. **نمودار ROC پس از fine-tune مدل**

همانطور که در شکل ها مشخص است در هر دو نمودار احساسات خوشحالی (happy) بهترین عملکرد را دارد و نمودار صورتی مربوط به این دسته بندی بیشترین شیب را دارد. همچنین اندازه عددی AUC و یا همان مساحت زیر نمودار ROC در بعد از fine-tune کردن مدل به عدد 0.96 رسیده است که بیانگر آن است که تست ما تا حد زیادی دارای عملکرد مناسبی در شناسایی این احساس از سایر احساسات است. در مقابل احساسات surprise و disgust دارای کمترین AUC هستند.

جدول ۲-۲ و ۲-۳ نشان دهنده مقادیر Precision, Recall و F1 برای هر کلاس در بعد از انجام training و finetuning هستند. احساسات به شکل روبرو هستند: Neutral, Happy, Sad, Surprised, Afraid Disgusted, Angryو Contemptuous.

جدول ۲-۲ – مقدار معیار ها پس از آموزش مدل

	precision	recall	f1-score	support
Su	0.72	0.47	0.57	380
Af	0.60	0.50	0.55	430
N	0.66	0.84	0.74	402
Sa	0.33	0.59	0.42	388
D	0.47	0.25	0.32	403
C	0.59	0.54	0.56	397
H	0.78	0.90	0.83	367
An	0.47	0.44	0.46	407
accuracy			0.56	3174
macro avg	0.58	0.56	0.56	3174
weighted avg	0.58	0.56	0.55	3174

جدول ۲-۳ – مقدار معیار ها پس از fine-tune مدل

	precision	recall	f1-score	support
Su	0.73	0.39	0.51	380
Af	0.54	0.63	0.58	430
N	0.61	0.79	0.69	402
Sa	0.42	0.48	0.45	388
D	0.48	0.31	0.38	403
C	0.60	0.55	0.57	397
H	0.69	0.93	0.79	367
An	0.50	0.44	0.47	407
accuracy			0.56	3174
macro avg	0.57	0.57	0.55	3174
weighted avg	0.57	0.56	0.55	3174

- Precision

این مقدار نسبت true positives به مجموع true positives و false positives است و صحت بالا برای یک کلاس به این معنی است که مدل، پیش‌بینی‌های false positives کمتری انجام می‌دهد. این مقدار در هر دو حالت برای کلاس های surprise و happy بیشتر از بقیه کلاس ها است.

کمترین مقدار بعد از آموزش برای احساس sad با تنها 0.33 و بعد از finetuning با مقدار 0.42 است. این بدین معنا است که مدل تعداد زیادی داده را در این کلاس پیش بینی می کند در حالی که دسته بندی درست این داده ها کلاس sad نبوده است.

- Recall

این معیار به دنبال محاسبه‌ی پوشش بر روی کل داده‌هاست و نسبت true positives به مجموع true positives و false negatives است. کلاس happy برای مدل آموزش داده شده مدل fine-tuned به ترتیب با ۹۰ درصد و ۹۳ درصد بهترین عمکرد را داشته است. پس می توان نتیجه گرفت این مدل بخش زیادی از داده های happy را به درستی شناسایی می کند.

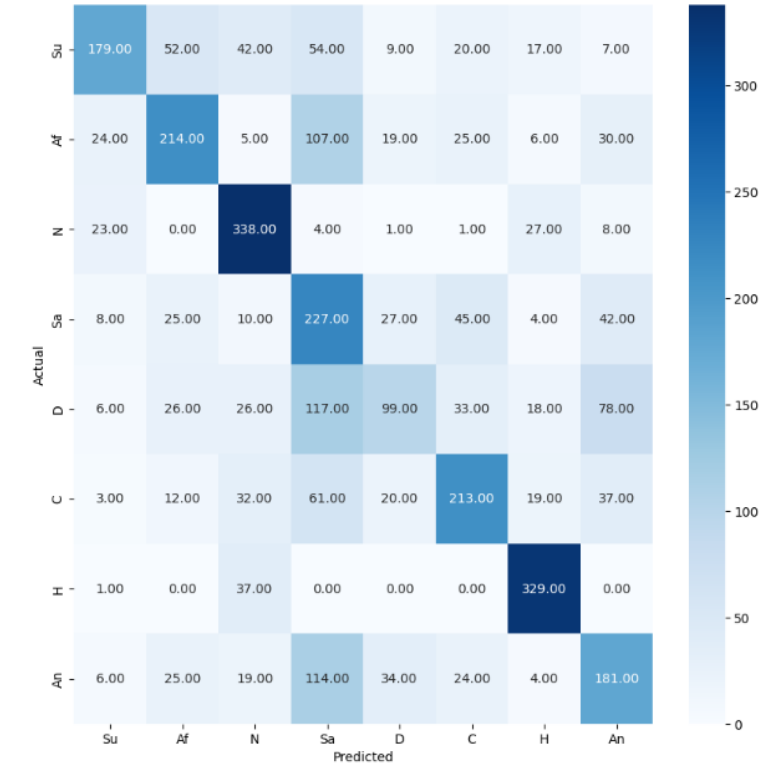
کلاس disgusted نیز کمترین درصد را برای recall دارد که بدین معنا است که مدل تعداد زیادی از داده های این کلاس ها را به اشتباه در کلاس های دیگری جز disgusted پیش بینی کرده است.

- F1

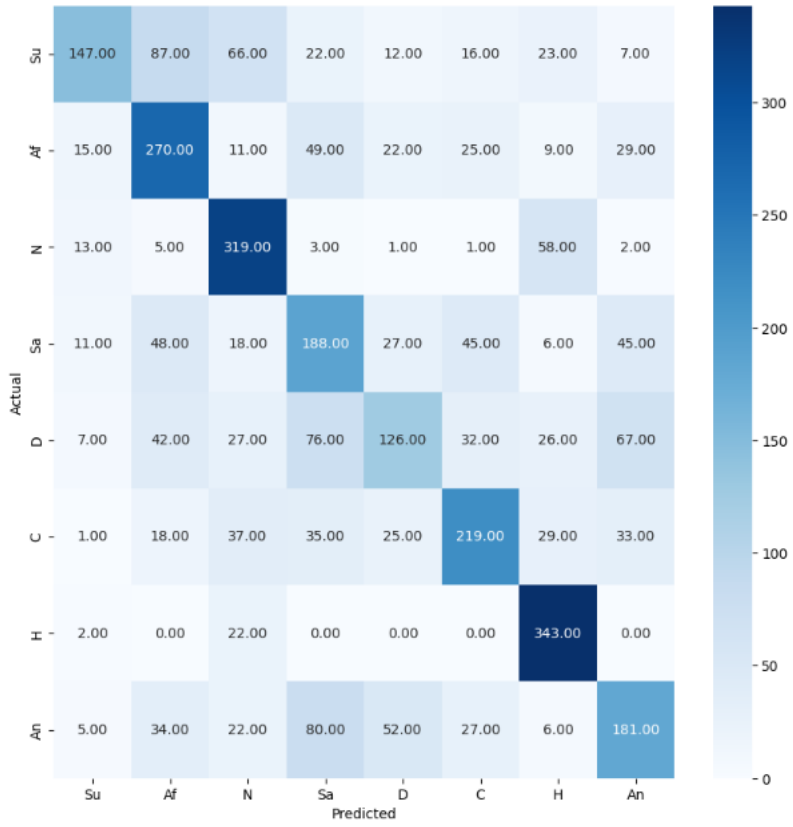
این معیار ارزیابی، میانگین هارمونیک از دو مورد قبلی یعنی Recall و Precision است. این معیار برای زمانی که تعداد داده های کلاس ها یکسان نیست بسیار کاربردی است. در این تمرین نیز تعداد داده ها در کلاس ها برابر نبوده اند. در این معیار نیز مانند معیار قبل مدل بهترین عملکرد را در neutral و happy دارد و بدترین عملکرد را در تشخیص کلاس disgusted.

جدول ۲-۴و ۲-۵ نشان دهنده ماتریس confusion برای هر کلاس در بعد از انجام training و finetuning هستند.

جدول ۲-۴ – ماتریس confusion پس از آموزش مدل



جدول ۲-۵ – ماتریس confusion پس از fine-tune مدل



همان طور که در جدول ها مشخص است و در معیار های قبلی نیز مشاهده شد دو مدل برای احساسات neutral و happy بهترین عملکرد را دارد و داده های این کلاس ها را با درصد بالایی به درستی دسته بندی می کند. در مقابل تشخیص احساسات surprised وdisgusted و تفکیک آن ها از احساسات دیگر برای مدل دشوار است. با توجه به نتایج به نظر میرسد تشخیص احساس disgust از sad و angry برای مدل کار دشواری است و این احساس ها را در بسیاری از موارد به جای هم تشخیص داده است. همانطور که در مقاله هم به این مورد اشاره شده است ، خشم و انزجار به دلیل واحدهای کنش(action unit) صورت مشترکی که دارند، اشتباه گرفته می شوند.

در کل ساختار این دو مدل بسیار شبیه به هم است اما تفاوت های هم دارند. از تفاوت های آن ها می توان به تعداد لایه ها اشاره کرد. مدل AlexNet از ۸ لایه تشکیل شده است که ۵ تای آن لایه convolutionalهستند و ۳ تای آن لایهfully-connected است. در مقابل VGGNet از ۱۳ لایه دارد که ۱۰ تای آن لایه convolutional هستند و ۳ تای آن لایهfully-connected است. مدل VGGNet ساختار یکدست تری دارد زیرا سایز کرنل در مدل AlexNet در لایه های convolutional کاهش می یابد ولی در مدل VGGNet لایه های convolutional ساختار و کرنل سایز مشابهی دارند. همچنین در مدل VGGNet بعد از ۲ لایه convolutional پشت هم pooling اعمال می شود.

مدل VGGNet با استفاده از کرنل های 3 × 3 stacked سعی در گرفتن ساختارهای بزرگتر تصویر دارد.

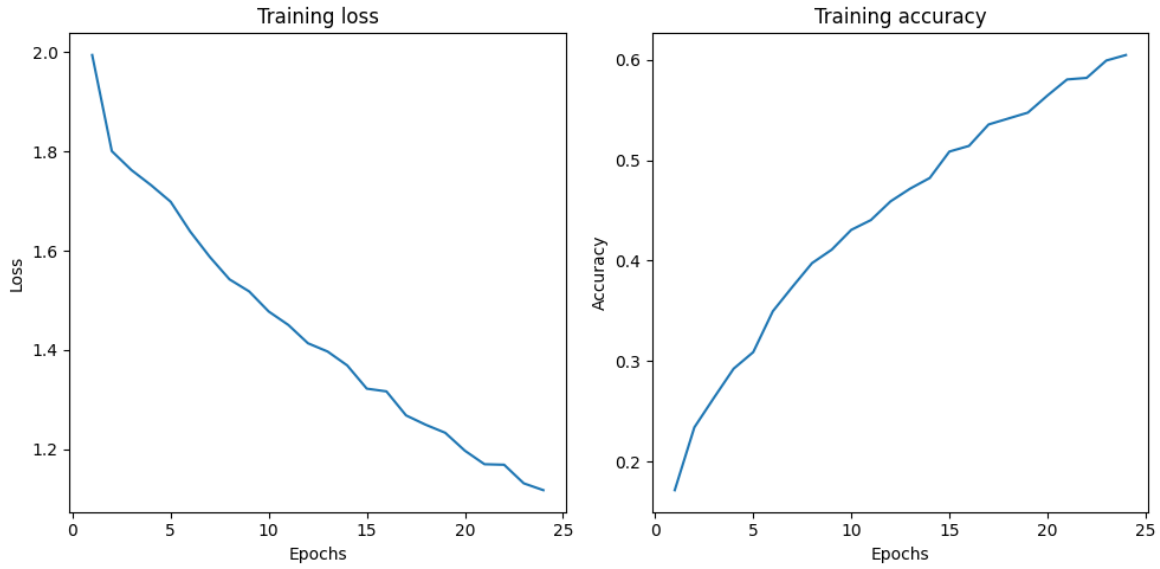
۲-۲. مدل MobileNet

معماری این مدل در جدول ۲-۶ قابل مشاهده است.

جدول ۲-۶. **معماری شبکه MobileNet**

Type	Shape	Stride	Output
Conv	$3 \times 3 \times 32$	2	$64 \times 64 \times 32$
DConv	$3 \times 3 \times 64$	1	$64 \times 64 \times 64$
DConv	$3 \times 3 \times 128$	2	$32 \times 32 \times 128$
DConv	$3 \times 3 \times 128$	1	$32 \times 32 \times 128$
DConv	$3 \times 3 \times 256$	2	$16 \times 16 \times 256$
DConv	$3 \times 3 \times 256$	1	$16 \times 16 \times 256$
DConv	$3 \times 3 \times 512$	2	$8 \times 8 \times 512$
5×DConv	$3 \times 3 \times 512$	1	$8 \times 8 \times 512$
DConv	$3 \times 3 \times 1024$	2	$4 \times 4 \times 1024$
DConv	$3 \times 3 \times 1024$	1	$4 \times 4 \times 1024$
GlobalAvePool	1024	–	–
Dense	8 or 2	–	1 label or 2 floats

پس از آموزش مدل با استفاده از روش‌ها و مقادیری که در مقاله ذکر شده است، نمودار accuracy و loss ما در حین آموزش به شکل زیر در می‌آید.

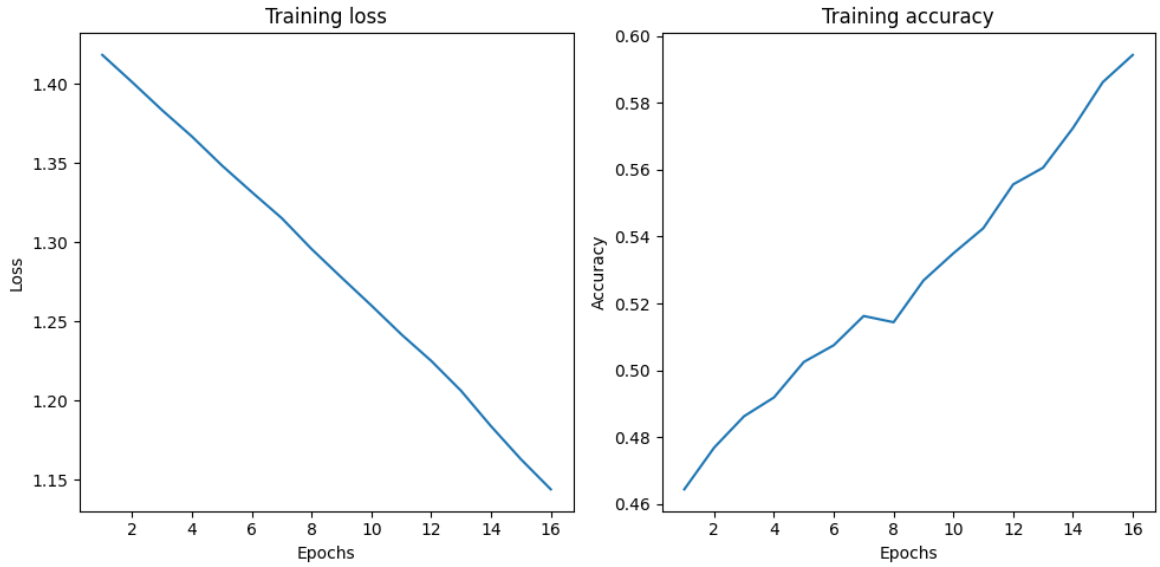


شکل ۲-۵. نمودار accuracy و loss در حین آموزش

همانطور که در نمودارها می‌بینیم، در طی فرایند آموزش مدل دقت و عملکرد مدل روی مجموعه داده آموزش به طور مداوم بهبود می‌یابد و پیشرفت مدل قابل قبول است.

همچنین دقت به دست آمده بر روی مجموعه داده ارزیابی پس از اتمام فرایند آموزش به ۴۴,۴۶ درصد رسید.

برای fine-tune کردن مدل، به این صورت عمل کردیم که لایه‌های convolutional را freeze کرده و صرفا وزن‌های شبکه fully-connected را مورد آموزش بیشتر با مجموعه داده جدید قرار دادیم. در این صورت نمودارهای accuracy و loss در حین fine-tune کردن به صورت زیر در می‌آیند.



شکل ۲-۶. نمودار accuracy و loss در حین fine-tune کردن مدل

همچنین دقت نهایی مدل پس از fine-tune کردن آن روی مجموعه داده تست ما در حدود ۴۲,۲۴ درصد شد.

یکی از تفاوت‌های این مدل با دو مدل AlexNet و VGGNet در این است که مدل MobileNet به طور خاص به منظور عملکرد خوب در حین سبک‌وزن بودن مدل طراحی و ساخته شده است و از ایده تجزیه کردن عملیات convolution عادی به دو بخش depthwise convolution و pointwise convolution استفاده می‌کند. در حالی که مدل VGGNet ساختار کلاسیک‌تری دارد و از فیلترهای ۳ در ۳ استک شده روی هم استفاده می‌کند. مدل AlexNet هم از پیشگامان CNN ها و جزو اولین مدل‌هایی است که در امر پردازش تصویر بسیار خوب عمل کرد. همچنین تعداد پارامترهای این مدل کمتر از دو مدل دیگر است.

در کل مدل MobileNet بیشتر برای استفاده در دستگاه‌های Mobile و یا Embedded طراحی شده است و در همین راستا هم مدل سبکی از لحاظ محاسباتی و حجم است و نسبت به دو مدل دیگر تعداد پارامترهای کمتری هم دارد.

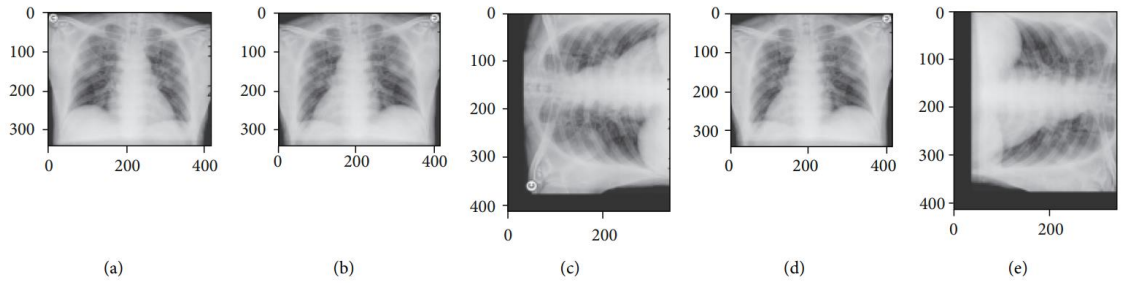
پاسخ ۳ – تشخیص بیماران مبتلا به کووید با استفاده از عکس ریه

۳-۱. معرفی مقاله

۳-۲. جمع‌آوری داده و پیش‌پردازش تصویر

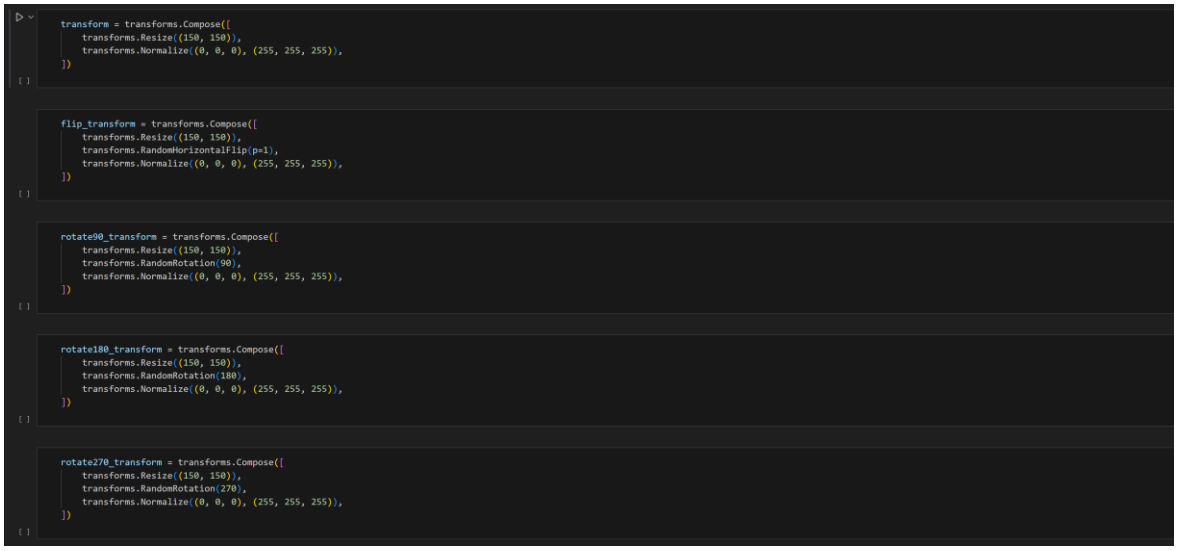
در این بخش از ۴ نوع data augmentation متفاوت برای تولید تصاویر بیشتر برای آموزش مدل استفاده شد.

نوع اول روی تصاویر اصلی ما عملیات Horizontal Flip را انجام داده و به همان تعداد تصاویر جدید ایجاد میکند. نوع دوم، سوم و چهارم هم هر کدام به ترتیب ۹۰، ۱۸۰ و ۲۷۰ درجه عکس را می‌چرخانند و عکس‌های جدید تولید می‌کنند. نمونه عکس‌های تولید شده توسط این عملیات‌ها را می‌توان در شکل ۳-۱ مشاهده کرد.



شکل ۳-۱. نمونه تصاویر استفاده شده برای augmentation

این عملیات‌ها به صورت مرحله به مرحله روی مجموعه داده اصلی ما اعمال شده و در نهایت منجر به تولید شدن مجموعه دادن آموزش به اندازه ۵ برابر مجموعه داده اصلی می‌شود که ۲۵ درصد آن برای validation و ۷۵ درصد ان برای آموزش مدل مورد استفاده قرار می‌گیرد.



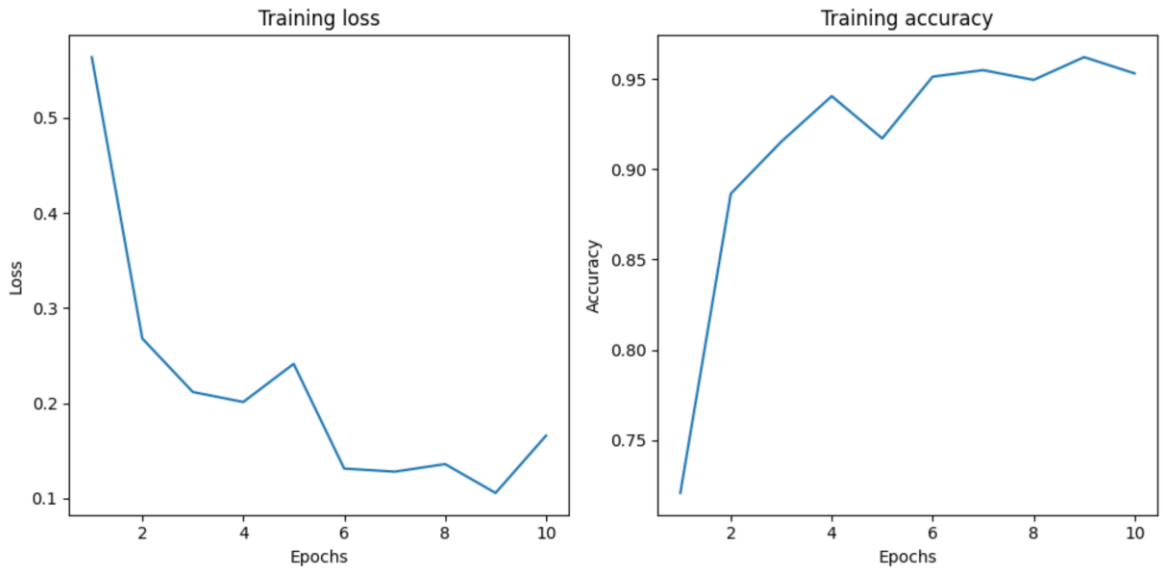
شکل ۳-۲. تعریف کردن transformهای مورد نیاز

```
# modify the root path based on your environment
root_dir = '/kaggle/input/xray-dataset-covid19/Covid_Dataset/xray_dataset_covid19/train'
original_dataset = ImagesDataset(root_dir, transform=transform)
flipped_dataset = ImagesDataset(root_dir, transform=flip_transform)
rotated90_dataset = ImagesDataset(root_dir, transform=rotate90_transform)
rotated180_dataset = ImagesDataset(root_dir, transform=rotate180_transform)
rotated270_dataset = ImagesDataset(root_dir, transform=rotate270_transform)
```

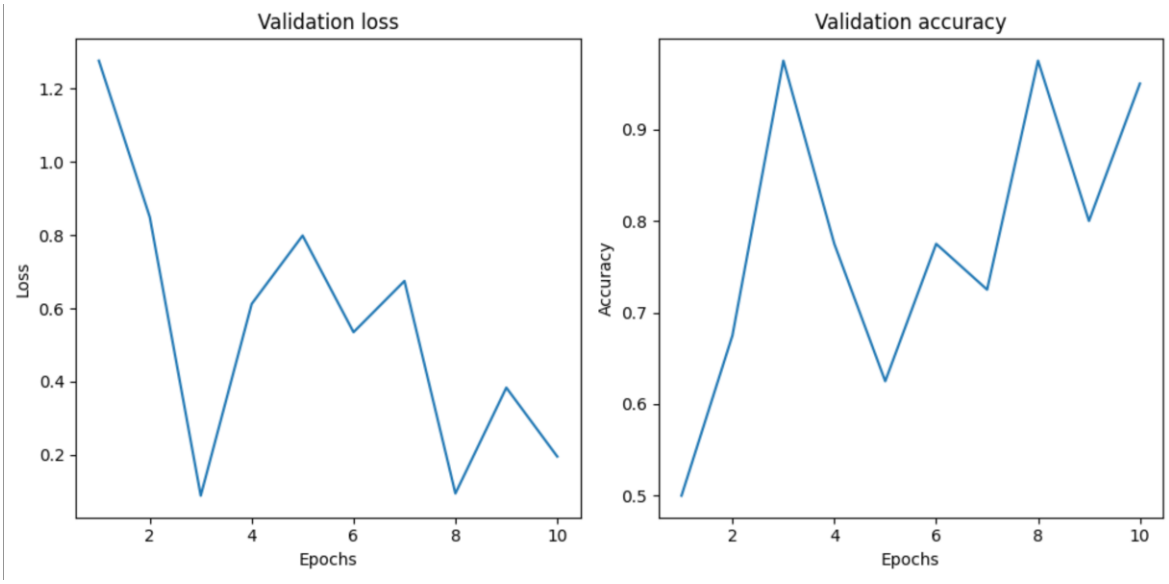
شکل ۳-۳. تولید تصاویر و مجموعه داده augment شده

۳-۳. آموزش شبکه

مدل ارائه شده دقیقا به صورت توضیح داده شده در مقاله ذکر شده پیاده‌سازی شد و نمودارهای loss و accuracy در حین آموزش مدل برای مجموعه داده augment شده رسم شده است. نتایج به دست آمده از این نمودارها را می‌توان در شکل‌های زیر مشاهده کرد.

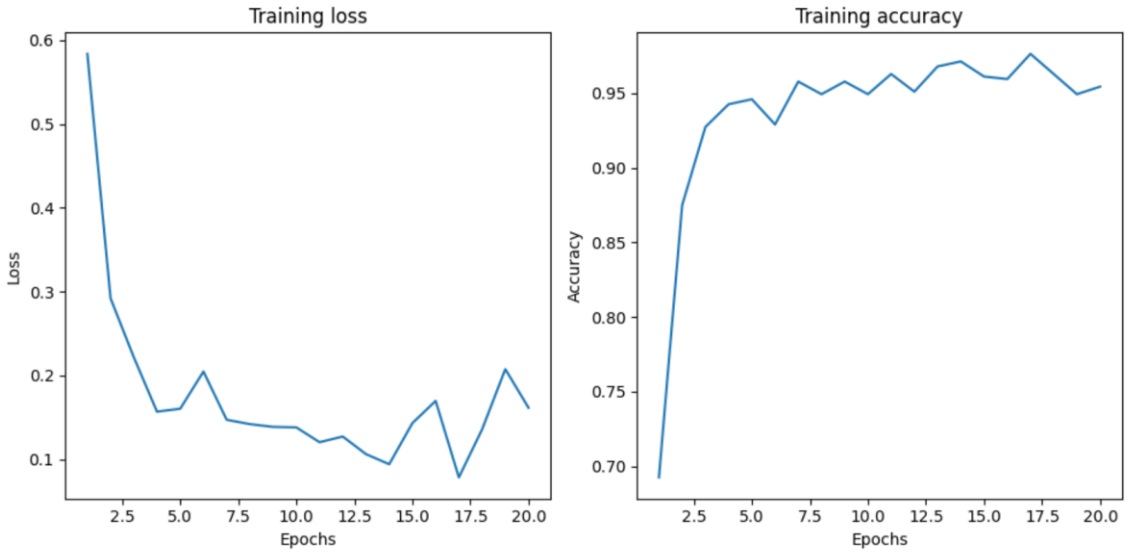


شکل ۳-۴. نمودار مربوط به مجموعه داده آموزش

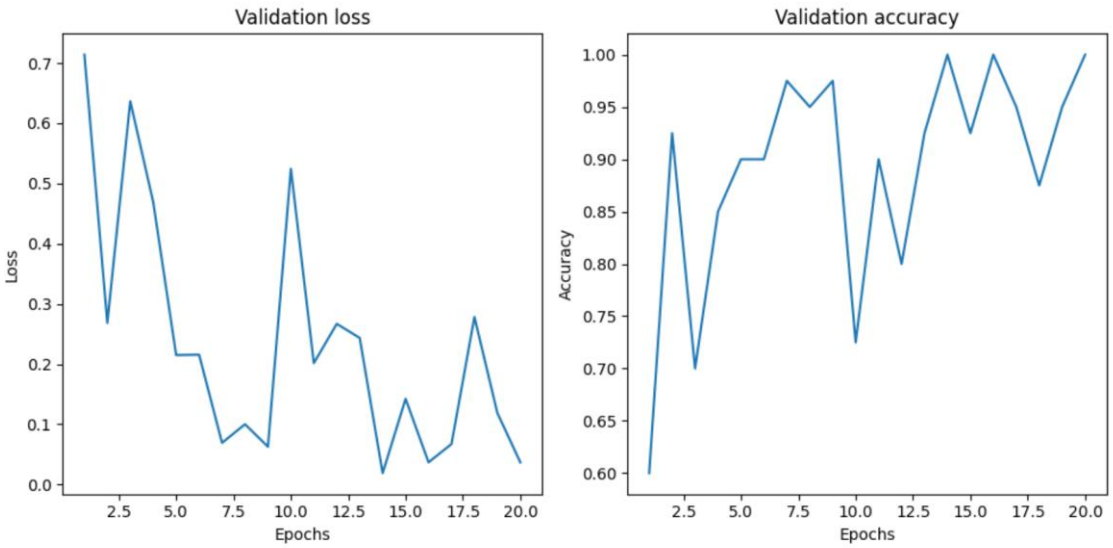


شکل ۳-۵. نمودار مربوط به مجموعه داده اعتبارسنجی

همانطور که می‌بینیم، به دقت خوبی هم روی مجموعه داده آموزش و هم روی مجموعه داده اعتبارسنجی رسیده‌ایم. با ادامه دادن روند آموزش باز هم ممکن است که عملکرد مدل بهتر شود ولی خطر overfitting هم وجود دارد. نتایج متناظر در صورت آموزش مدل در ۲۰ epoch هم در زیر آورده شده است.



شکل ۳-۶. نمودار مربوط به مجموعه داده آموزش



شکل ۳-۷. نمودار مربوط به مجموعه داده اعتبارسنجی

۳-۴. ارزیابی شبکه

طبق نتایج به دست آمده که به صورت کامل در notebook ضمیمه شده وجود دارند، در صورت آموزش مدل با داده augment شده، مقادیر به دست آمده برای معیارهای ما روی مجموعه داده ارزیابی طبق شکل ۳-۸ خواهند بود.

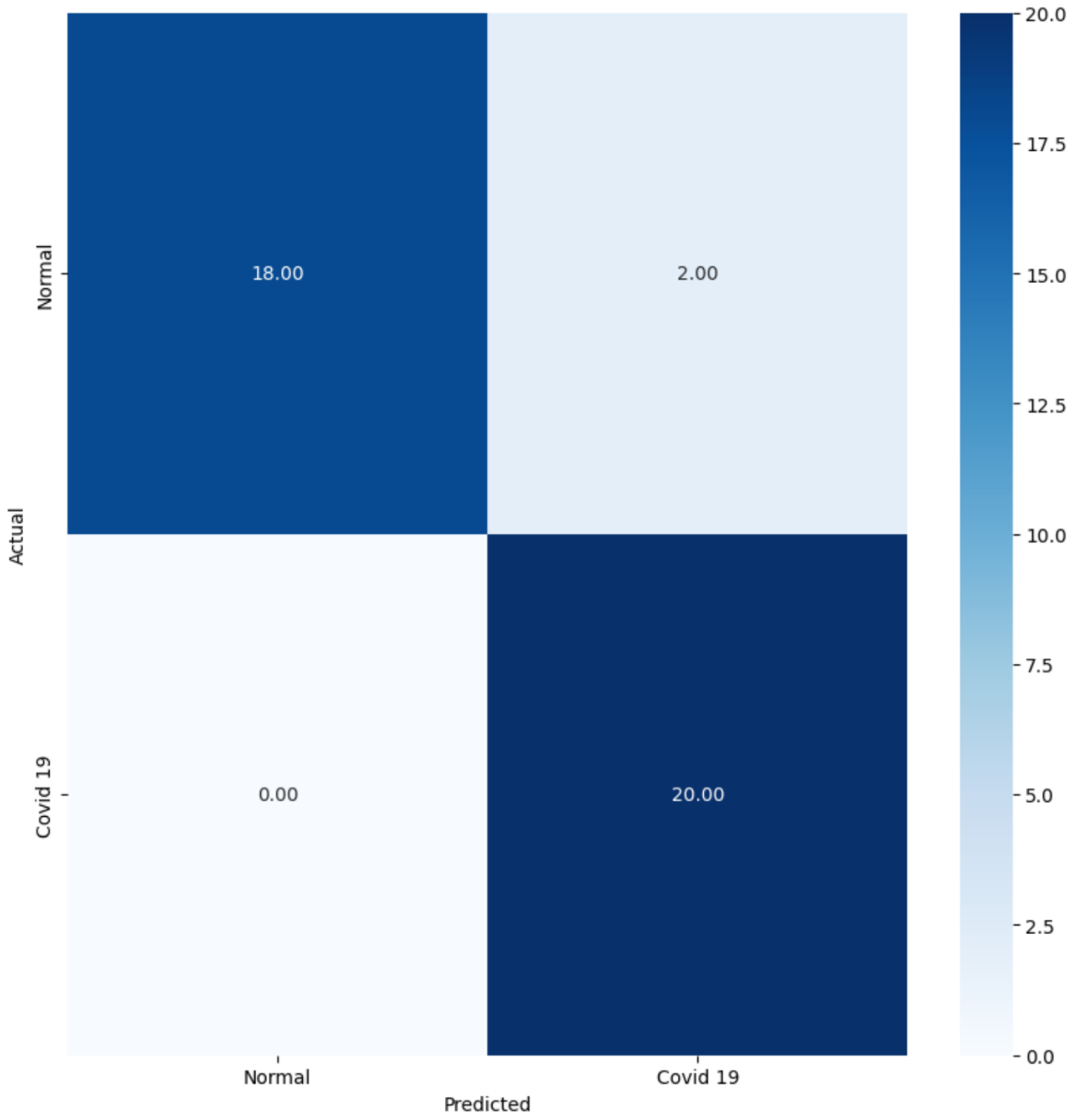
```
calculate_metrics(labels_test, preds_test)
```

	Value
Accuracy	0.950000
F1 Score	0.952381
Precision	0.909091
Sensitivity (Recall)	1.000000
Specificity	0.900000

شکل ۳-۸. مقادیر معیارهای طبقه‌بندی بر روی مجموعه داده ارزیابی

از آنجا که تشخیص نادرست افرادی که واقعا مبتلا به کرونا هستند در این مساله برای ما هزینه زیادی دارد و مهم است، بالا بودن معیار Sensitivity برای ما مهم است که با این مدل برآورده نیز شده است.

همچنین ماتریس آشفتگی نیز در شکل ۳-۹. قابل مشاهده می‌باشد.



شکل ۳-۹. ماتریس آشفتگی

با آموزش مدل بر روی epoch ۲۰، می‌توان حتی به دقت ۱۰۰ درصد رسید که البته با توجه به اندازه کوچک مجموعه داده ارزیابی ما ممکن است زیاد قابل اعتماد نباشد و در مجموعه داده بزرگتر با مشکل مواجه شود. نتایج مربوط به این مدل در شکل ۳-۱۰ قابل مشاهده است.

In [51]:
```
calculate_metrics(dataset3_labels_test, dataset3_preds_test)
```

Out[51]:

	Value
Accuracy	1.0
F1 Score	1.0
Precision	1.0
Sensitivity (Recall)	1.0
Specificity	1.0

شکل ۳-۱۰. نتایج معیارهای طبقه‌بندی برای مدل ثانویه

.