

به نام خدا

استاد: دکتر مرضیه داود آبادی
درس مبانی یادگیری عمیق

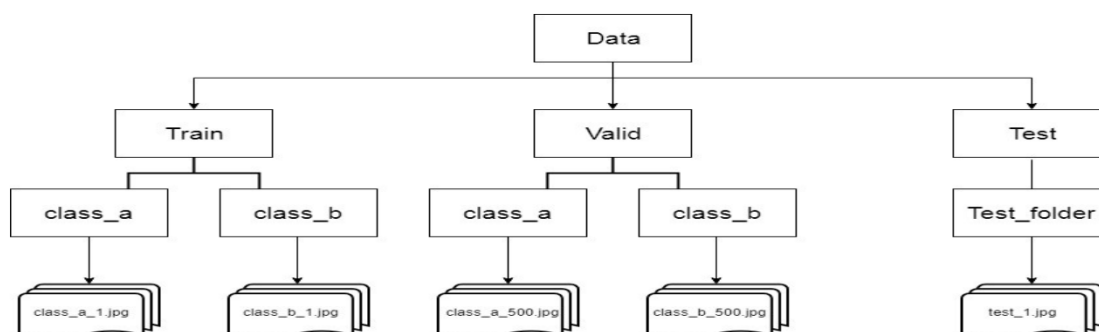
نام: فاطمه زهرا بخشنده
شماره دانشجویی: 98522157

گزارش تمرین 4:

سوال پنجم:

الف) ابتدا فابل دیتاست را در درایو آپلود کرده، درایو را در کولب mount می کنیم و دیتا ها را unzip می کنیم.

مراحل را طبق نوت های موجود در کولب انجام می دهیم و دیتاست را آماده می کنیم. 80 درصد داده ها را برای train، 10 درصد validation و 10 درصد برای test در نظر می گیریم. باید دقت کنیم برای استفاده از ImageDataGenerator، باید دیتاست خود را به شکل زیر طبقه بندی کنیم:



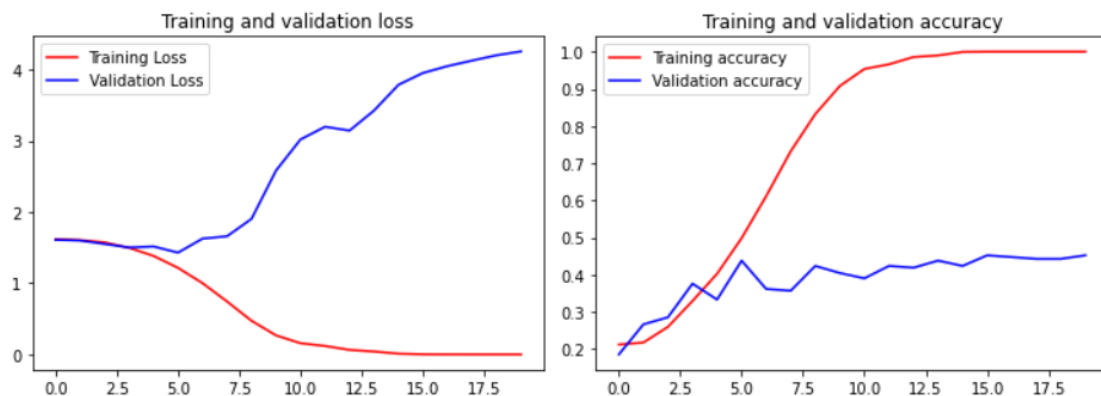
مدل خود را میسازیم. برای این مدل، سه لایه کانولوشنی با تعداد فیلتر 32، 64 و 128 به همراه max pooling در نظر گرفته، و در ادامه از یک لایه dense با تعداد نورون 512 و لایه آخر هم یک لایه dense با 5 نورون (به تعداد کلاس ها) در نظر گرفتیم. مدل را با بهینه ساز adam و تابع ضرر categorical_crossentropy کامپایل می کنیم.

summary مدل به صورت زیر است:

```
conv2d_15 (Conv2D)          (None, 111, 111, 32)      896
max_pooling2d_15 (MaxPoolin (None, 55, 55, 32)        0
g2D)
conv2d_16 (Conv2D)          (None, 27, 27, 64)      18496
max_pooling2d_16 (MaxPoolin (None, 13, 13, 64)        0
g2D)
conv2d_17 (Conv2D)          (None, 11, 11, 128)     73856
max_pooling2d_17 (MaxPoolin (None, 5, 5, 128)         0
g2D)
flatten_5 (Flatten)         (None, 3200)             0
dense_10 (Dense)            (None, 512)              1638912
dense_11 (Dense)            (None, 5)                2565
=====
Total params: 1,734,725
Trainable params: 1,734,725
Non-trainable params: 0
```

از سه callback پیشنهادی استفاده می کنیم.

در نهایت در تابع train_model_naive_split از ImageDataGenerator استفاده کرده و مدل را fit می کنیم.



```
loss: 4.2808e-04 - accuracy: 1.0000 - val_loss: 4.2559 - val_accuracy: 0.4524
```

می بینیم که دقت مدل روی داده train زود به 1 می رسد، در حالی رو روی داده validation پیشرفت زیادی نمی کند. در واقع مدل روی داده train، overfit شده است. دقت آن روی داده train خیلی بهتر از دقت آن روی داده validation است.

نتیجه store prediction: مشخص است دقت مدل روی داده test خوب نیست و درست predict نمی کند.

```

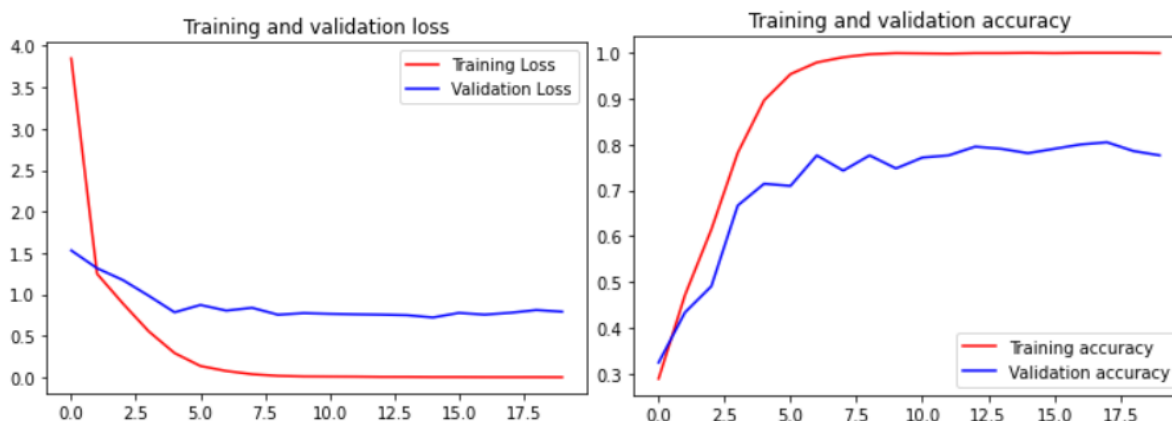
image_id      label
iranKhodro_dena/111.jpg    0
iranKhodro_dena/114.jpg    4
iranKhodro_dena/122.jpg    4
iranKhodro_dena/130.jpg    0
iranKhodro_dena/157.jpg    4
Writing submission

```

استفاده از transfer learning:

این دفعه از مدل MobileNetV2 با وزن های از پیش آموخته imagenet استفاده می کنیم. این مدل را به جز لایه های dense آخر آن، لود می کنیم. و سپس لایه های dense خودمان را در ادامه آن اضافه می کنیم. ویژگی trainable لایه های مدل MobileNetV2 را False کرده و در واقع آن ها را فریز می کنیم. مدل را کامپایل کرده و شبکه را آموزش می دهیم.

Layer (type)	Output Shape	Param #
mobilenetv2_1.00_224 (Functional)	(None, 1280)	2257984
dropout_3 (Dropout)	(None, 1280)	0
flatten_1 (Flatten)	(None, 1280)	0
dense_6 (Dense)	(None, 128)	163968
dense_7 (Dense)	(None, 5)	645
=====		
Total params: 2,422,597		
Trainable params: 164,613		
Non-trainable params: 2,257,984		



```
loss: 0.1014 - accuracy: 0.9732 - val_loss: 0.4745 - val_accuracy: 0.8476
```

می بینیم که این دفعه مدل سریع آموزش می بیند و علاوه بر train accuracy، دقت validation نیز به نحوه خوبی افزایش می یابد. در آخر val accuracy به 85% رسیده و تقریباً دو برابر دقت مدل قبلی می شود!

نتیجه store prediction:

image_id	label
iranKhodro_dena/109.jpg	0
iranKhodro_dena/123.jpg	0
iranKhodro_dena/133.jpg	0
iranKhodro_dena/143.jpg	0
iranKhodro_dena/165.jpg	0
Writing submission	

همانطور که می بینیم داده های test را هم به خوبی predict می کند. پس با استفاده از مدل pretrained کمبود داده در تسک جدید را جبران کردیم. دقت تا حد خوبی پیشرفت کرده و توانستیم از overfitting هم جلوگیری کنیم.

استفاده از Grad Cam:

مراحل مربوط به نمایش heat map را با کمک این [منبع](#) و اسلاید ها می نویسیم. در اینجا چون می خواهیم مدل خود را predict کنیم و این مدل در نهایت 5 نرون خروجی دارد، باید تابع decode_predictions را خودمان با توجه به تسک 5 کلاسه مان پیاده سازی کرده و نمی توانیم از تابع آماده آن استفاده کنیم. پس از پیاده سازی توابع مورد نظر، یک دیتای رندوم را از دیتاست گرفته، مدل pretrained که مدل بهتری بود و با کمک callback checkpoint در my_own_model.h5 ذخیره شده است را لود کرده و مراحل را روی آن اجرا می کنیم تا grad cam heat map را بسازیم.

منابع: [لینک](#) و [لینک](#) و [لینک](#)