

HW4

بابک بهکام کیا

سوال 5)

قبل از هر کاری دیتای داده شده را به 3 دسته train, test, validation تقسیم می کنیم.

در ابتدا متغیر های گلوبال را تعریف می کنیم.

```
1 num_classes = 5
2 EPOCHS = 150
3 BATCH_SIZE = 20
4
5 BASE_FOLDER = './data/'
```

سپس مدل خود را تعریف می کنیم.

```
19
20 model = Sequential([
21     layers.Conv2D(16,3,padding='SAME',activation="relu",input_shape=(224,224,3)),
22     layers.MaxPooling2D(2),
23     layers.Conv2D(32,3,padding='SAME',activation="relu"),
24     layers.MaxPooling2D(2),
25     # layers.Conv2D(64,3,padding='SAME',activation="relu"),
26     # layers.MaxPooling2D(2),
27     layers.Flatten(),
28     layers.Dense(32,activation="relu"),
29     layers.Dense(5,activation="softmax")
30 ])
31
32
33 model.compile(
34     optimizer='adam',
35     loss='categorical_crossentropy',
36     metrics=['accuracy']
37 )
38
39 print(model.summary())
40
```

در مرحله بعد کالیک های گفته شده را به لیست callbacks اضافه می کنیم.

```

14 ##### Your implementation here #####
15 callback = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=3)
16 reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2,
17                               patience=5, min_lr=0.001)
18 model_checkpoint_callback = tf.keras.callbacks.ModelCheckpoint(
19     'models/cnn_model.hdf5',
20     monitor="val_accuracy",
21     save_best_only=True)
22
23 callbacks = [
24     callback,
25     reduce_lr,
26     model_checkpoint_callback
27 ]
28
29

```

در نهایت اندازه step در هر epoch را محاسبه می کنیم و تابع fit را صدا می زنیم.

```

41
42 model = create_cnn_model()
43
44 STEP_SIZE_TRAIN=train_generator.n//train_generator.batch_size
45 STEP_SIZE_VALID=validation_generator.n//validation_generator.batch_size
46
47
48 history = model.fit_generator(generator=train_generator,
49                             steps_per_epoch=STEP_SIZE_TRAIN,
50                             validation_data=validation_generator,
51                             validation_steps=STEP_SIZE_VALID,
52                             epochs=50,
53                             verbose=2,
54                             callbacks=create_callbacks()
55                             )
56

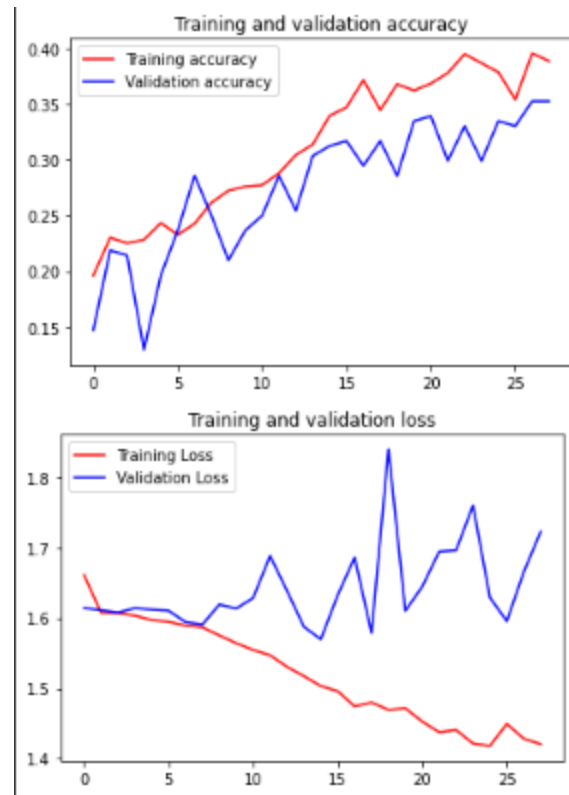
```

و نتایج زیر به دست آمد.

```

50/50 - 30s - loss: 1.4409 - accuracy: 0.3950 - val_loss: 1.6960 - val_accuracy: 0.3304 - lr: 0.0010 - 30s/epoch - 591ms/ste
Epoch 24/50
50/50 - 31s - loss: 1.4217 - accuracy: 0.3868 - val_loss: 1.7598 - val_accuracy: 0.2991 - lr: 0.0010 - 31s/epoch - 616ms/ste
Epoch 25/50
50/50 - 30s - loss: 1.4183 - accuracy: 0.3786 - val_loss: 1.6290 - val_accuracy: 0.3348 - lr: 0.0010 - 30s/epoch - 592ms/ste
Epoch 26/50
50/50 - 36s - loss: 1.4495 - accuracy: 0.3541 - val_loss: 1.5954 - val_accuracy: 0.3304 - lr: 0.0010 - 36s/epoch - 711ms/ste
Epoch 27/50
50/50 - 30s - loss: 1.4285 - accuracy: 0.3956 - val_loss: 1.6651 - val_accuracy: 0.3527 - lr: 0.0010 - 30s/epoch - 590ms/ste
Epoch 28/50
50/50 - 31s - loss: 1.4208 - accuracy: 0.3887 - val_loss: 1.7225 - val_accuracy: 0.3527 - lr: 0.0010 - 31s/epoch - 629ms/ste

```



در قسمت بعدی این سوال مدل MobileNetV2 را load می کنیم و لایه های آخر آن را عوض می کنیم تا classification را 5 کلاسه بکنیم.

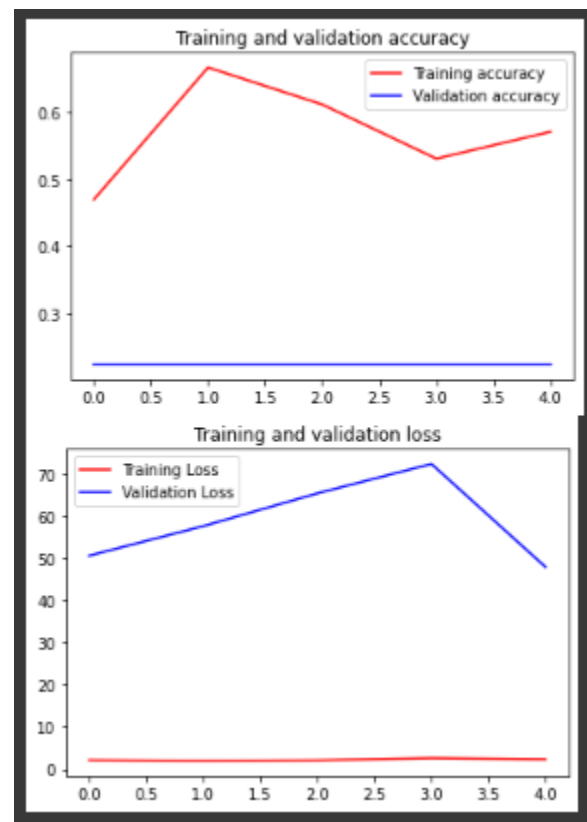
```

19 MobileNetV2 = tf.keras.applications.MobileNetV2(input_shape=(224, 224, 3),
20                                                    alpha=1.0,
21                                                    include_top=False,
22                                                    weights="imagenet",
23                                                    input_tensor=None,
24                                                    pooling=None,
25                                                    classes=5,
26                                                    classifier_activation="softmax",
27                                                    )
28
29 model = Sequential()
30 model.add(MobileNetV2)
31 model.add(layers.Flatten())
32 model.add(layers.Dense(5, activation='softmax'))
33
34
35 model.compile(
36     optimizer='adam',
37     loss='categorical_crossentropy',
38     metrics=['accuracy']
39 )
40
41 print(model.summary())
42

```

و نتایج زیر به دست آمد.

```
Epoch 1/10
50/50 - 41s - loss: 2.6772 - accuracy: 0.3755 - val_loss: 46.3469 - val_accuracy: 0.2188 - lr: 0.0010 - 41s/epoch - 820ms/step
Epoch 2/10
50/50 - 35s - loss: 1.5812 - accuracy: 0.6094 - val_loss: 23.6510 - val_accuracy: 0.2098 - lr: 0.0010 - 35s/epoch - 702ms/step
Epoch 3/10
50/50 - 35s - loss: 1.9329 - accuracy: 0.5736 - val_loss: 19.4586 - val_accuracy: 0.1339 - lr: 0.0010 - 35s/epoch - 704ms/step
Epoch 4/10
50/50 - 34s - loss: 1.5215 - accuracy: 0.6478 - val_loss: 62.0084 - val_accuracy: 0.2232 - lr: 0.0010 - 34s/epoch - 683ms/step
Epoch 5/10
50/50 - 34s - loss: 1.3423 - accuracy: 0.6899 - val_loss: 39.5137 - val_accuracy: 0.1741 - lr: 0.0010 - 34s/epoch - 683ms/step
Epoch 6/10
50/50 - 35s - loss: 1.5198 - accuracy: 0.7164 - val_loss: 27.1219 - val_accuracy: 0.1562 - lr: 0.0010 - 35s/epoch - 703ms/step
Epoch 7/10
50/50 - 35s - loss: 1.6598 - accuracy: 0.6937 - val_loss: 53.9764 - val_accuracy: 0.1786 - lr: 0.0010 - 35s/epoch - 703ms/step
Epoch 8/10
50/50 - 35s - loss: 1.2666 - accuracy: 0.6692 - val_loss: 89.3754 - val_accuracy: 0.1384 - lr: 0.0010 - 35s/epoch - 690ms/step
Epoch 9/10
50/50 - 35s - loss: 1.1956 - accuracy: 0.6811 - val_loss: 65.8798 - val_accuracy: 0.1607 - lr: 0.0010 - 35s/epoch - 702ms/step
Epoch 10/10
50/50 - 35s - loss: 0.8100 - accuracy: 0.7547 - val_loss: 50.5562 - val_accuracy: 0.1607 - lr: 0.0010 - 35s/epoch - 704ms/step
```



لینک های استفاده شده:

<https://keras.io/api/applications/mobilenet/>

https://keras.io/api/callbacks/early_stopping/

https://keras.io/api/callbacks/reduce_lr_on_plateau/

https://keras.io/api/callbacks/model_checkpoint/

<https://keras.io/api/applications/mobilenet/>

<https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/>

<https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>

https://keras.io/guides/transfer_learning/

<https://stackoverflow.com/questions/64984888/add-layers-into-keras-pretrained-model>

https://www.tensorflow.org/api_docs/python/tf/keras/metrics/categorical_crossentropy

<https://pavisj.medium.com/convolutions-and-backpropagations-46026a8f5d2c>

<https://medium.com/featurepreneur/image-data-augmentation-using-keras-imagedatagenerator-1cee60255ea8>

<https://stackoverflow.com/questions/49535561/imagedatagenerator-on-a-folder>

<https://keras.io/api/preprocessing/image/>

<https://vijayabhaskar96.medium.com/tutorial-image-classification-with-keras-flow-from-directory-and-generators-95f75ebe5720>

<https://pythontutorials.eu/deep-learning/image-classification/>