

## بخش 1:

## (الف)

Dropout را به منظور کاهش پیچیدگی شبکه و جلوگیری از overfit شدن است، Dropout ها روی هر لایه اعمال می شوند و هدف آنها جلوگیری از لرن شدن بعضی نورون ها در هر تکرار است این و به اصطلاح این نورون ها از شبکه حذف می شوند همین امر باعث می شود تا همه ی نورن های یک لایه به اندازه کل تکرار ها روی داده آموزش داده نشوند، dropout بصورت تصادفی نورون ها را حذف میکند و با تنظیم پارامتر rate میزان حذف شدن نورون را نشان میدهد مثلا اگر مقدار آن برابر 0.2 یعنی 20 درصد نورون ها حذف می شوند.

## (ب)

رابطه عکس دارند به این منظور که هرچقدر میزان dropout بیشتر باشد در لایه نورون های بیشتر حذف می شوند در نتیجه ظرفیت مدل ما کاهش می یابد و پارامتر های کمتری در کل شبکه آموزش داده می شود

## بخش 2:

در لایه های fully connected فرض بر آن است که اجماع تمام نرون های قبلی یک ویژگی می سازد و هر نورون به تمام نورون های لایه قبلش متصل می شود و locally connected هر نورون به بخشی از نورون های قبلی برای استخراج یک ویژگی خاص به آن نورون ها متصل می شود . در locally connected هم میتوانیم اشتراک وزن را داشته باشیم هم نداشته باشیم چون ما میخواهیم از آن دسته نورون قبلی یک ویژگی استخراج کنیم ممکن است بخواهیم از وزن بدست آمده در مکان های دیگر نیز استفاده کنیم در این صورت وزن ها را به اشتراک میذاریم در غیر اینصورت از اشتراک وزن ها را نداریم که باعث افزایش چشمگیر پارامتر های یادگیریمان می شود.

## بخش 3:

(الف) image generator

```

from keras.preprocessing.image import ImageDataGenerator
batch_size = 16
train_datagen = ImageDataGenerator(
    rotation_range=25,
    vertical_flip=True,
    horizontal_flip=True,
    shear_range=0.1,
    width_shift_range=0.2,
    height_shift_range=0.2,
    validation_split= 0.1,
    fill_mode="wrap"
)

test_datagen = ImageDataGenerator()
train_datagen_wo_agum = ImageDataGenerator(validation_split= 0.1)

train_dir=os.path.normpath('./train')

train_generator = train_datagen.flow_from_directory(train_dir,seed=42, batch_size=batch_size, subset='training',class_mode='categorical')
validation_generator = train_datagen.flow_from_directory(train_dir,seed=42, batch_size=batch_size, subset='validation',class_mode='categorical')
#with out agumation
train_wo_agum = train_datagen_wo_agum.flow_from_directory(train_dir,seed=42, subset='training',class_mode='categorical', batch_size=batch_size)
validation_generator_wo_agum = train_datagen_wo_agum.flow_from_directory(train_dir,seed=42, subset='validation',class_mode='categorical', batch_size=batch_size)
#test
test_generator = test_datagen.flow_from_directory('./test',seed=42, batch_size=1)

```

داده افزایی بر روی داده های آموزشی انجام شده و 10 درصد داده برای validation انتخاب شده اند داده ها 25 درجه چرخش داده اند تصاویر بصورت افقی و عمودی flip می شوند داده ها نیز شیف داده شده اند و 10 درصد داده ها نیز shear شده اند همینطور داده ها به صورت رندوم چپ و راست و بالا و پایین شده اند

و همینطور یک مدل ImageDataGenerator به صورت خالی تعریف شده است و فقط 10 درصد داده ها برای validation انتخاب شده اند تا دیتای بدون داده افزایی نیز داشته باشیم

(ب)

```

def build_model():
    model = Sequential()
    model.add(layers.Conv2D(64, (3,3), input_shape=(256, 256, 3)))
    model.add(layers.Conv2D(32, (3,3), activation='relu'))
    model.add(layers.Conv2D(32, (3,3), activation='relu'))
    model.add(layers.Conv2D(16, (3,3), activation='relu'))

    model.add(layers.Flatten())
    model.add(layers.Dense(128, activation='relu'))
    model.add(layers.Dense(5, activation='softmax'))
    return model

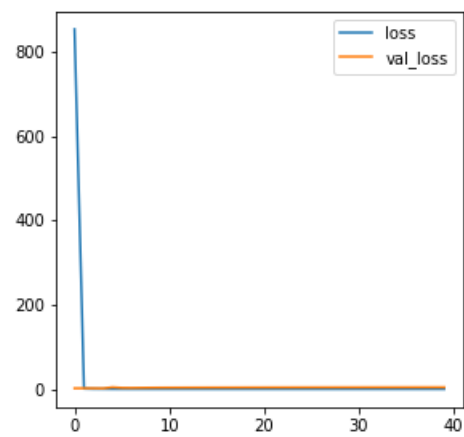
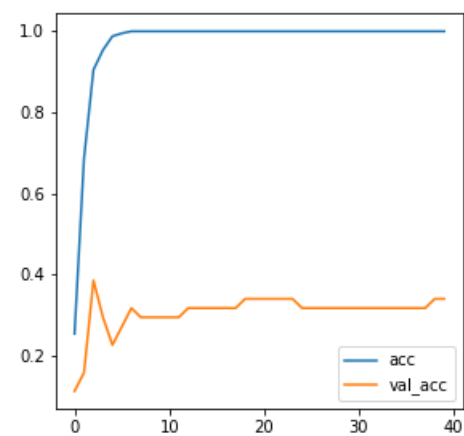
model = build_model()
loss = 'categorical_crossentropy'
optimizer = 'adam'
model.compile(loss= loss, optimizer= optimizer, metrics=['accuracy'])

```

مدل ای که طبق گفته ی سوال تعریف شده است و 4 لایه کانولوشنی و دو لایه dense نیز تعریف شده اند چون دسته بندی هست از categorical\_crossentropy استفاده کردیم و از بهینه ساز adam نیز استفاده کردیم

```
history = model.fit(
    train_wo_agum,
    epochs=25,
    batch_size=batch_size,
    validation_data=validation_generator_wo_agum
)
```

مدل بر روی دیتاها عادی آموزش داده شد



مدل آموزش داده شد و overfit شد

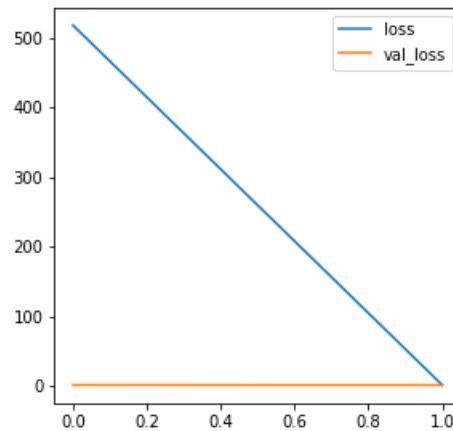
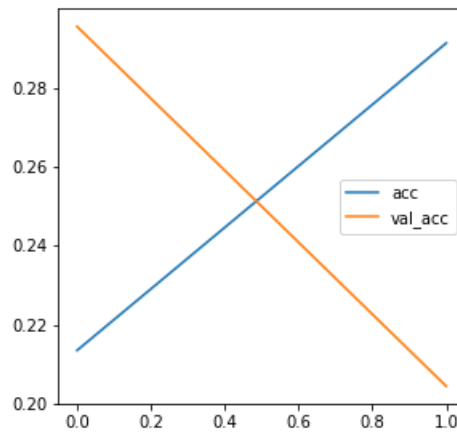
(پ)

```

1 model = build_model()
  loss = 'categorical_crossentropy'
  optimizer = 'adam'
  model.compile(loss= loss, optimizer= optimizer, metrics=['accuracy'])
  agu_history = model.fit(
    train_generator,
    epochs=40,
    batch_size=batch_size,
    validation_data=validation_generator,
    callbacks=[callbacks.TensorBoard('logs/with_augmentation'),es]
  )

```

/usr/local/lib/python3.7/dist-packages/PIL/Image.py:960: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images  
 "Palette images with Transparency expressed in bytes should be "  
 Epoch 1/40  
 26/26 [=====] - 12s 450ms/step - loss: 517.7046 - accuracy: 0.2136 - val\_loss: 1.7239 - val\_accuracy: 0.2955  
 Epoch 2/40  
 26/26 [=====] - 11s 432ms/step - loss: 1.6361 - accuracy: 0.2913 - val\_loss: 1.6142 - val\_accuracy: 0.2045



با اضافه کرده داده هایی که تغییرات داده افزایی در ان اعمال شده مدل overfit نشد اما به accuracy خوبی نرسید.

(ت)

Dropout با مقدار های 0.3,0.5,0.7 تست شدند

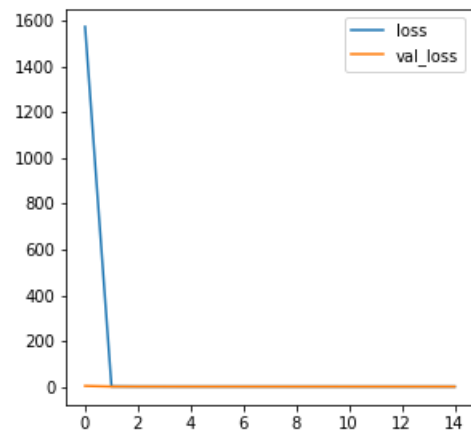
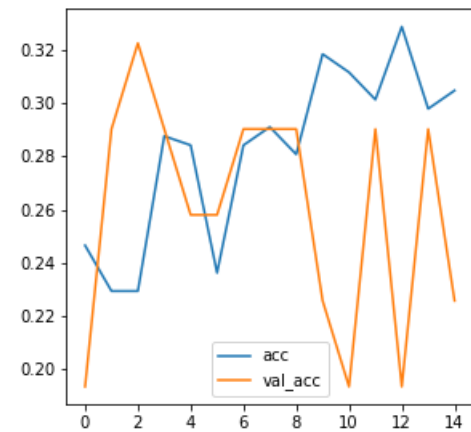
```
drop_history = []
summary = []
for rate in [0.3,0.5,0.7]:
    model = build_model(rate)
    summary.append(model.summary())
    loss = 'categorical_crossentropy'
    optimizer = 'adam'
    model.compile(loss= loss, optimizer= optimizer, metrics=['accuracy'])
    drop_history.append(model.fit(
        train_generator,
        epochs=15,
        batch_size=batch_size,
        validation_data=validation_generator,
        # callbacks=[es]
    ))
```

(0.3

```

Epoch 9/15
19/19 [=====] - 8s 395ms/step - loss: 1.5444 - accuracy: 0.2808 - val_loss: 1.5442 - val_accuracy: 0.2903
Epoch 10/15
19/19 [=====] - 7s 391ms/step - loss: 1.5611 - accuracy: 0.3185 - val_loss: 1.5786 - val_accuracy: 0.2258
Epoch 11/15
19/19 [=====] - 8s 395ms/step - loss: 1.5163 - accuracy: 0.3116 - val_loss: 1.5632 - val_accuracy: 0.1935
Epoch 12/15
19/19 [=====] - 8s 389ms/step - loss: 1.5074 - accuracy: 0.3014 - val_loss: 1.5656 - val_accuracy: 0.2903
Epoch 13/15
19/19 [=====] - 7s 390ms/step - loss: 1.5016 - accuracy: 0.3288 - val_loss: 1.5581 - val_accuracy: 0.1935
Epoch 14/15
19/19 [=====] - 7s 392ms/step - loss: 1.5241 - accuracy: 0.2979 - val_loss: 1.4974 - val_accuracy: 0.2903
Epoch 15/15
19/19 [=====] - 7s 390ms/step - loss: 1.4901 - accuracy: 0.3048 - val_loss: 1.5303 - val_accuracy: 0.2258

```



مدل به دقت خوبی نرسید اما از overfit شدن جلوگیری شد

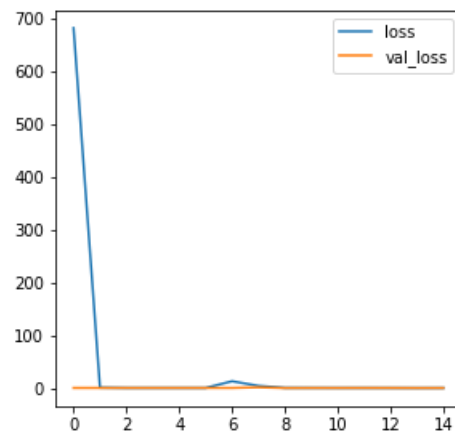
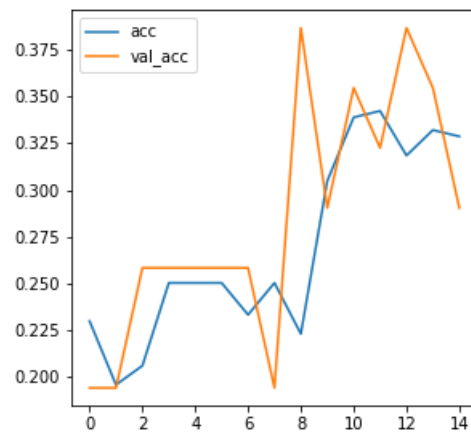
با داده افزایی:

(0.5

```

19/19 [=====] - 7s 391ms/step - loss: 1.5099 - accuracy: 0.3185 - val_loss: 1.5833 - val_accuracy: 0.3871
Epoch 14/15
19/19 [=====] - 8s 408ms/step - loss: 1.4701 - accuracy: 0.3322 - val_loss: 1.4607 - val_accuracy: 0.3548
Epoch 15/15
19/19 [=====] - 8s 394ms/step - loss: 1.5097 - accuracy: 0.3288 - val_loss: 1.5118 - val_accuracy: 0.2903

```



مدل به دقت خوبی نرسید اما از overfit شدن جلوگیری شد و val loss , loss

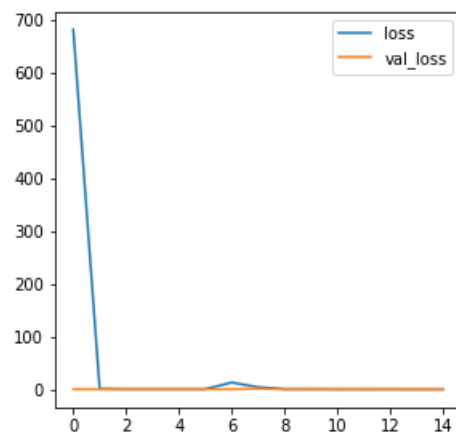
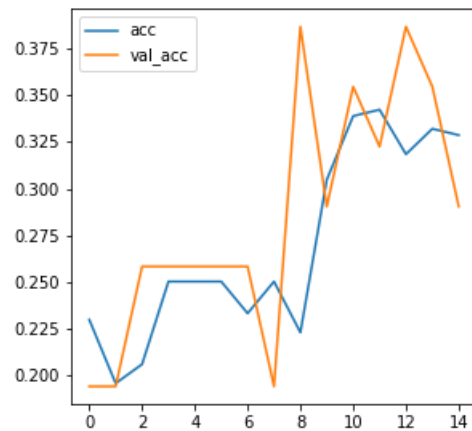
به هم نزدیک تر شدند که نشان از همگرا شدن آنهاست

(0.7

```

19/19 [=====] - 8s 397ms/step - loss: 1.5886 - accuracy: 0.2534 - val_loss: 1.5874 - val_accuracy: 0.2581
Epoch 13/15
19/19 [=====] - 8s 395ms/step - loss: 1.5834 - accuracy: 0.2568 - val_loss: 1.5867 - val_accuracy: 0.2581
Epoch 14/15
19/19 [=====] - 8s 393ms/step - loss: 1.5873 - accuracy: 0.2534 - val_loss: 1.5846 - val_accuracy: 0.2581
Epoch 15/15
19/19 [=====] - 8s 409ms/step - loss: 1.5953 - accuracy: 0.2534 - val_loss: 1.5833 - val_accuracy: 0.2581

```



مدل به دقت خوبی نرسید اما از overfit شدن جلوگیری شد و loss , val loss نسبت به 0.5

به هم نزدیک تر شدند که نشان از همگرا شدن آنهاست و اما دقت کمتری بدست آمد

بدون داده افزایی:

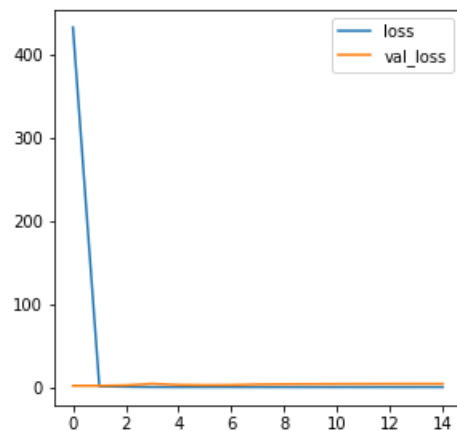
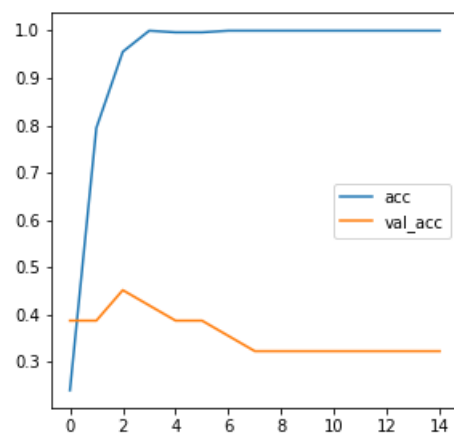
:0.3



```

Epoch 13/15
19/19 [=====] - 4s 207ms/step - loss: 6.8754e-05 - accuracy: 1.0000 - val_loss: 3.8323 - val_accuracy: 0.3226
Epoch 14/15
19/19 [=====] - 4s 203ms/step - loss: 2.2621e-05 - accuracy: 1.0000 - val_loss: 3.8701 - val_accuracy: 0.3226
Epoch 15/15
19/19 [=====] - 4s 200ms/step - loss: 4.2763e-05 - accuracy: 1.0000 - val_loss: 3.8957 - val_accuracy: 0.3226

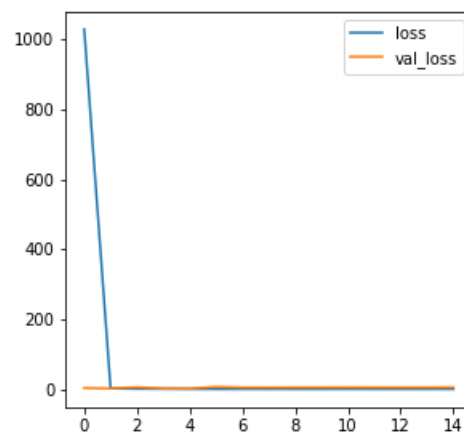
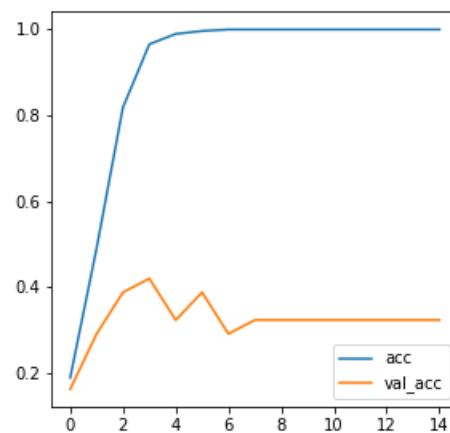
```



به دلیل حجم داده های کم و ظرفیت بالای مدل مدل روی داده های آموزشی overfit شد

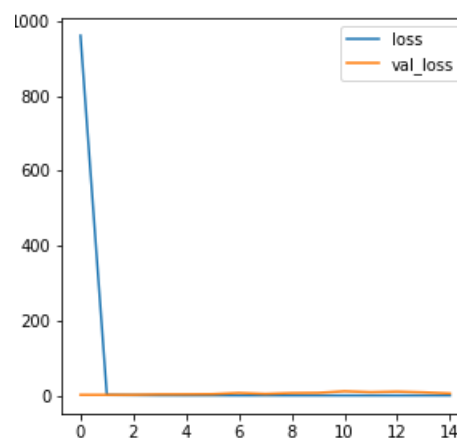
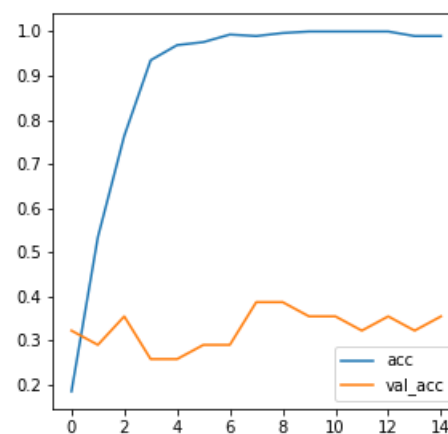
(0.5

```
Epoch 12/15
19/19 [=====] - 4s 207ms/step - loss: 1.4473e-05 - accuracy: 1.0000 - val_loss: 4.7149 - val_accuracy: 0.3226
Epoch 13/15
19/19 [=====] - 4s 203ms/step - loss: 3.2264e-05 - accuracy: 1.0000 - val_loss: 4.5687 - val_accuracy: 0.3226
Epoch 14/15
19/19 [=====] - 4s 206ms/step - loss: 1.5962e-04 - accuracy: 1.0000 - val_loss: 4.6518 - val_accuracy: 0.3226
Epoch 15/15
19/19 [=====] - 4s 205ms/step - loss: 4.3629e-06 - accuracy: 1.0000 - val_loss: 4.9618 - val_accuracy: 0.3226
```



(0.7

```
19/19 [=====] - 4s 202ms/step - loss: 2.1171e-04 - accuracy: 1.0000 - val_loss: 11.1212 - val_accuracy: 0.3548
Epoch 12/15
19/19 [=====] - 4s 206ms/step - loss: 8.6651e-05 - accuracy: 1.0000 - val_loss: 8.7927 - val_accuracy: 0.3226
Epoch 13/15
19/19 [=====] - 5s 259ms/step - loss: 1.6416e-04 - accuracy: 1.0000 - val_loss: 10.1559 - val_accuracy: 0.3548
Epoch 14/15
19/19 [=====] - 4s 201ms/step - loss: 0.0556 - accuracy: 0.9897 - val_loss: 8.2841 - val_accuracy: 0.3226
Epoch 15/15
19/19 [=====] - 4s 206ms/step - loss: 0.0253 - accuracy: 0.9897 - val_loss: 5.3143 - val_accuracy: 0.3548
```



با افزایش مقدار dropout مدل باز overfit شد اما مقدار validation loss کمتر شد و درصد خطای دیتاهای valid بهتر شد و در تلاش بود تا جلوی overfit شدن را بگیرد