IMAGE AUGUMENTATION

```
!pip install split-folders
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting split-folders
  Downloading split_folders-0.5.1-py3-none-any.whl (8.4 kB)
Installing collected packages: split-folders
Successfully installed split-folders-0.5.1
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import splitfolders
# To only split into training and validation set, set a tuple to `ratio`, i.e, `(.8, .2)`.
splitfolders.ratio("/content/drive/MyDrive/flowers", output="/content/drive/MyDrive/flowers",seed=1337, ratio=(.8, .1, .1), group_prefix=None, move=False)
```

```
Copying files: 4317 files [01:17, 55.49 files/s]
```

```
import keras
from keras.preprocessing.image import ImageDataGenerator
```

```
def getdata():
  train_datagen=ImageDataGenerator (rescale=1./255,
  shear_range=0.2,
  rotation_range=180,
  zoom_range=0.2,
  horizontal_flip=True)
  test_datagen=ImageDataGenerator (rescale=1./255)
  x_train = train_datagen.flow_from_directory
(r'/content/drive/MyDrive/flowers/train', target_size = (64,64), batch_size = 32, class_mode= 'categorical')
  x_test = 
test_datagen.flow_from_directory(r'/content/drive/MyDrive/flowers/test', target_size = (64,64), batch_size = 32, class_mode= 'categorical')
  return x_train,x_test;
```

CREATE MODEL

```
#To define linear intialisan
from keras.models import Sequential
#To add Layers import Dense
from keras.layers import Dense
#To create Convolution kernel import Convolution2D
from keras.layers import Convolution2D
```

```python
#import Maxpooling layer
from keras.layers import MaxPooling2D
#import Flatten Layer
from keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')
```

```python
model=Sequential()
```

ADDING LAYERS

```python
#add convolutional layer
model.add(Convolution2D(112, (7,7),input_shape=(64,64,3),
activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
```

```python
#add hidden layer
model.add(Dense(32,activation='relu'))
model.add(Dense(64,activation='relu'))
```

COMPILE THE MODEL

```python
#add output Layer
model.add(Dense(5, activation='softmax'))
model.compile(loss = 'categorical_crossentropy', optimizer = "adam",
metrics = ["accuracy"])
```

```python
model.summary()
```

```
Model: "sequential"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 58, 58, 112)       16576

 max_pooling2d (MaxPooling2D  (None, 29, 29, 112)      0
 )

 flatten (Flatten)           (None, 94192)             0

 dense (Dense)               (None, 32)                3014176

 dense_1 (Dense)             (None, 64)                2112

 dense_2 (Dense)             (None, 5)                 325

=================================================================
Total params: 3,033,189
Trainable params: 3,033,189
Non-trainable params: 0
_________________________________________________________________
```

FIT THE MODEL

```
x_train,x_test=getdata()
tom=model.fit_generator(x_train , steps_per_epoch=len(x_train),
epochs=100, validation_data=x_test, validation_steps=len(x_test))
```

```
Found 3452 images belonging to 5 classes.
Found 435 images belonging to 5 classes.
Epoch 1/100
108/108 [==============================] - 45s 407ms/step - loss:
1.4229 - accuracy: 0.3827 - val_loss: 1.2983 - val_accuracy: 0.4368
Epoch 2/100
108/108 [==============================] - 41s 383ms/step - loss:
1.2115 - accuracy: 0.4780 - val_loss: 1.1736 - val_accuracy: 0.5034
Epoch 3/100
108/108 [==============================] - 42s 387ms/step - loss:
1.1918 - accuracy: 0.5046 - val_loss: 1.1473 - val_accuracy: 0.5379
Epoch 4/100
108/108 [==============================] - 42s 386ms/step - loss:
1.1423 - accuracy: 0.5330 - val_loss: 1.1095 - val_accuracy: 0.5471
Epoch 5/100
108/108 [==============================] - 42s 386ms/step - loss:
1.1290 - accuracy: 0.5406 - val_loss: 1.1051 - val_accuracy: 0.5517
Epoch 6/100
108/108 [==============================] - 42s 387ms/step - loss:
1.1000 - accuracy: 0.5576 - val_loss: 1.1469 - val_accuracy: 0.5103
Epoch 7/100
108/108 [==============================] - 42s 387ms/step - loss:
1.0895 - accuracy: 0.5646 - val_loss: 1.1434 - val_accuracy: 0.5540
Epoch 8/100
108/108 [==============================] - 42s 390ms/step - loss:
1.0595 - accuracy: 0.5733 - val_loss: 1.1650 - val_accuracy: 0.5609
Epoch 9/100
108/108 [==============================] - 42s 386ms/step - loss:
1.0429 - accuracy: 0.5756 - val_loss: 1.0629 - val_accuracy: 0.5977
Epoch 10/100
108/108 [==============================] - 42s 388ms/step - loss:
1.0427 - accuracy: 0.5808 - val_loss: 1.0503 - val_accuracy: 0.5701
Epoch 11/100
108/108 [==============================] - 42s 388ms/step - loss:
1.0409 - accuracy: 0.5765 - val_loss: 1.0563 - val_accuracy: 0.5632
Epoch 12/100
108/108 [==============================] - 42s 388ms/step - loss:
1.0167 - accuracy: 0.5956 - val_loss: 1.0337 - val_accuracy: 0.6000
Epoch 13/100
108/108 [==============================] - 42s 385ms/step - loss:
0.9996 - accuracy: 0.5918 - val_loss: 1.0376 - val_accuracy: 0.6000
Epoch 14/100
108/108 [==============================] - 42s 387ms/step - loss:
0.9955 - accuracy: 0.5895 - val_loss: 1.0509 - val_accuracy: 0.5839
Epoch 15/100
```

```
108/108 [==============================] - 42s 388ms/step - loss:
0.9781 - accuracy: 0.6133 - val_loss: 1.0002 - val_accuracy: 0.5908
Epoch 16/100
108/108 [==============================] - 42s 389ms/step - loss:
0.9801 - accuracy: 0.6020 - val_loss: 1.0043 - val_accuracy: 0.6253
Epoch 17/100
108/108 [==============================] - 42s 385ms/step - loss:
0.9676 - accuracy: 0.6023 - val_loss: 1.0117 - val_accuracy: 0.6069
Epoch 18/100
108/108 [==============================] - 42s 387ms/step - loss:
0.9504 - accuracy: 0.6156 - val_loss: 1.0148 - val_accuracy: 0.5954
Epoch 19/100
108/108 [==============================] - 42s 386ms/step - loss:
0.9374 - accuracy: 0.6266 - val_loss: 0.9903 - val_accuracy: 0.6345
Epoch 20/100
108/108 [==============================] - 42s 388ms/step - loss:
0.9304 - accuracy: 0.6211 - val_loss: 0.9755 - val_accuracy: 0.6345
Epoch 21/100
108/108 [==============================] - 42s 387ms/step - loss:
0.9121 - accuracy: 0.6280 - val_loss: 0.9511 - val_accuracy: 0.6575
Epoch 22/100
108/108 [==============================] - 42s 386ms/step - loss:
0.9137 - accuracy: 0.6350 - val_loss: 1.0227 - val_accuracy: 0.5954
Epoch 23/100
108/108 [==============================] - 42s 386ms/step - loss:
0.9222 - accuracy: 0.6272 - val_loss: 0.9923 - val_accuracy: 0.6276
Epoch 24/100
108/108 [==============================] - 42s 390ms/step - loss:
0.9099 - accuracy: 0.6324 - val_loss: 0.9688 - val_accuracy: 0.6230
Epoch 25/100
108/108 [==============================] - 42s 385ms/step - loss:
0.9057 - accuracy: 0.6298 - val_loss: 0.9611 - val_accuracy: 0.6276
Epoch 26/100
108/108 [==============================] - 42s 385ms/step - loss:
0.8986 - accuracy: 0.6347 - val_loss: 0.9766 - val_accuracy: 0.6161
Epoch 27/100
108/108 [==============================] - 42s 387ms/step - loss:
0.8949 - accuracy: 0.6446 - val_loss: 0.9586 - val_accuracy: 0.6552
Epoch 28/100
108/108 [==============================] - 42s 387ms/step - loss:
0.8864 - accuracy: 0.6448 - val_loss: 0.9768 - val_accuracy: 0.6483
Epoch 29/100
108/108 [==============================] - 42s 386ms/step - loss:
0.8910 - accuracy: 0.6448 - val_loss: 0.9375 - val_accuracy: 0.6207
Epoch 30/100
108/108 [==============================] - 42s 386ms/step - loss:
0.8929 - accuracy: 0.6489 - val_loss: 0.9930 - val_accuracy: 0.6161
Epoch 31/100
108/108 [==============================] - 42s 387ms/step - loss:
0.8956 - accuracy: 0.6390 - val_loss: 0.9831 - val_accuracy: 0.6253
```

```
Epoch 32/100
108/108 [==============================] - 42s 386ms/step - loss:
0.8755 - accuracy: 0.6515 - val_loss: 0.9529 - val_accuracy: 0.6529
Epoch 33/100
108/108 [==============================] - 42s 385ms/step - loss:
0.8500 - accuracy: 0.6585 - val_loss: 1.0159 - val_accuracy: 0.6138
Epoch 34/100
108/108 [==============================] - 42s 387ms/step - loss:
0.8550 - accuracy: 0.6643 - val_loss: 1.0299 - val_accuracy: 0.6207
Epoch 35/100
108/108 [==============================] - 42s 385ms/step - loss:
0.8451 - accuracy: 0.6541 - val_loss: 0.9103 - val_accuracy: 0.6644
Epoch 36/100
108/108 [==============================] - 42s 384ms/step - loss:
0.8464 - accuracy: 0.6596 - val_loss: 0.9457 - val_accuracy: 0.6368
Epoch 37/100
108/108 [==============================] - 42s 383ms/step - loss:
0.8487 - accuracy: 0.6553 - val_loss: 1.0188 - val_accuracy: 0.6161
Epoch 38/100
108/108 [==============================] - 41s 382ms/step - loss:
0.8561 - accuracy: 0.6535 - val_loss: 0.9220 - val_accuracy: 0.6483
Epoch 39/100
108/108 [==============================] - 42s 384ms/step - loss:
0.8189 - accuracy: 0.6758 - val_loss: 0.8935 - val_accuracy: 0.6552
Epoch 40/100
108/108 [==============================] - 42s 384ms/step - loss:
0.8322 - accuracy: 0.6674 - val_loss: 0.9167 - val_accuracy: 0.6575
Epoch 41/100
108/108 [==============================] - 42s 384ms/step - loss:
0.8259 - accuracy: 0.6590 - val_loss: 0.9213 - val_accuracy: 0.6529
Epoch 42/100
108/108 [==============================] - 42s 387ms/step - loss:
0.8228 - accuracy: 0.6695 - val_loss: 0.9271 - val_accuracy: 0.6437
Epoch 43/100
108/108 [==============================] - 42s 384ms/step - loss:
0.8298 - accuracy: 0.6640 - val_loss: 0.9307 - val_accuracy: 0.6345
Epoch 44/100
108/108 [==============================] - 42s 387ms/step - loss:
0.8096 - accuracy: 0.6686 - val_loss: 0.9410 - val_accuracy: 0.6391
Epoch 45/100
108/108 [==============================] - 42s 386ms/step - loss:
0.8053 - accuracy: 0.6773 - val_loss: 0.9283 - val_accuracy: 0.6667
Epoch 46/100
108/108 [==============================] - 42s 385ms/step - loss:
0.8024 - accuracy: 0.6767 - val_loss: 0.9461 - val_accuracy: 0.6368
Epoch 47/100
108/108 [==============================] - 42s 386ms/step - loss:
0.8011 - accuracy: 0.6819 - val_loss: 0.9182 - val_accuracy: 0.6552
Epoch 48/100
108/108 [==============================] - 42s 389ms/step - loss:
```

```
0.7995 - accuracy: 0.6811 - val_loss: 0.8911 - val_accuracy: 0.6644
Epoch 49/100
108/108 [==============================] - 42s 388ms/step - loss:
0.7891 - accuracy: 0.6868 - val_loss: 1.0326 - val_accuracy: 0.6230
Epoch 50/100
108/108 [==============================] - 42s 388ms/step - loss:
0.8387 - accuracy: 0.6724 - val_loss: 0.9104 - val_accuracy: 0.6322
Epoch 51/100
108/108 [==============================] - 42s 388ms/step - loss:
0.8088 - accuracy: 0.6811 - val_loss: 0.9043 - val_accuracy: 0.6552
Epoch 52/100
108/108 [==============================] - 42s 387ms/step - loss:
0.7938 - accuracy: 0.6805 - val_loss: 0.9212 - val_accuracy: 0.6552
Epoch 53/100
108/108 [==============================] - 42s 387ms/step - loss:
0.8039 - accuracy: 0.6732 - val_loss: 0.9538 - val_accuracy: 0.6345
Epoch 54/100
108/108 [==============================] - 42s 388ms/step - loss:
0.7884 - accuracy: 0.6831 - val_loss: 0.8745 - val_accuracy: 0.6805
Epoch 55/100
108/108 [==============================] - 42s 387ms/step - loss:
0.7703 - accuracy: 0.6915 - val_loss: 0.9312 - val_accuracy: 0.6552
Epoch 56/100
108/108 [==============================] - 42s 387ms/step - loss:
0.7690 - accuracy: 0.6938 - val_loss: 0.9271 - val_accuracy: 0.6575
Epoch 57/100
108/108 [==============================] - 42s 386ms/step - loss:
0.7840 - accuracy: 0.6892 - val_loss: 0.9243 - val_accuracy: 0.6713
Epoch 58/100
108/108 [==============================] - 42s 385ms/step - loss:
0.7724 - accuracy: 0.6906 - val_loss: 0.9665 - val_accuracy: 0.6322
Epoch 59/100
108/108 [==============================] - 42s 388ms/step - loss:
0.7733 - accuracy: 0.6935 - val_loss: 0.9118 - val_accuracy: 0.6736
Epoch 60/100
108/108 [==============================] - 42s 386ms/step - loss:
0.7708 - accuracy: 0.6970 - val_loss: 0.9365 - val_accuracy: 0.6460
Epoch 61/100
108/108 [==============================] - 42s 386ms/step - loss:
0.7737 - accuracy: 0.6932 - val_loss: 0.9820 - val_accuracy: 0.6276
Epoch 62/100
108/108 [==============================] - 42s 388ms/step - loss:
0.7663 - accuracy: 0.6912 - val_loss: 0.9206 - val_accuracy: 0.6529
Epoch 63/100
108/108 [==============================] - 42s 385ms/step - loss:
0.7767 - accuracy: 0.6883 - val_loss: 0.9583 - val_accuracy: 0.6506
Epoch 64/100
108/108 [==============================] - 42s 385ms/step - loss:
0.7475 - accuracy: 0.7068 - val_loss: 0.9767 - val_accuracy: 0.6437
Epoch 65/100
```

```
108/108 [==============================] - 42s 387ms/step - loss:
0.7329 - accuracy: 0.7054 - val_loss: 0.9316 - val_accuracy: 0.6552
Epoch 66/100
108/108 [==============================] - 42s 386ms/step - loss:
0.7576 - accuracy: 0.6947 - val_loss: 1.0120 - val_accuracy: 0.6529
Epoch 67/100
108/108 [==============================] - 42s 387ms/step - loss:
0.7622 - accuracy: 0.6958 - val_loss: 1.0015 - val_accuracy: 0.6391
Epoch 68/100
108/108 [==============================] - 42s 384ms/step - loss:
0.7684 - accuracy: 0.6889 - val_loss: 0.9465 - val_accuracy: 0.6713
Epoch 69/100
108/108 [==============================] - 42s 386ms/step - loss:
0.7465 - accuracy: 0.7048 - val_loss: 0.9293 - val_accuracy: 0.6736
Epoch 70/100
108/108 [==============================] - 42s 387ms/step - loss:
0.7479 - accuracy: 0.7039 - val_loss: 0.9391 - val_accuracy: 0.6598
Epoch 71/100
108/108 [==============================] - 42s 386ms/step - loss:
0.7321 - accuracy: 0.7109 - val_loss: 0.9749 - val_accuracy: 0.6506
Epoch 72/100
108/108 [==============================] - 42s 387ms/step - loss:
0.7356 - accuracy: 0.7013 - val_loss: 0.9331 - val_accuracy: 0.6391
Epoch 73/100
108/108 [==============================] - 42s 386ms/step - loss:
0.7472 - accuracy: 0.6999 - val_loss: 0.9373 - val_accuracy: 0.6667
Epoch 74/100
108/108 [==============================] - 42s 387ms/step - loss:
0.7545 - accuracy: 0.7037 - val_loss: 0.8956 - val_accuracy: 0.6667
Epoch 75/100
108/108 [==============================] - 42s 387ms/step - loss:
0.7488 - accuracy: 0.7022 - val_loss: 0.9172 - val_accuracy: 0.6598
Epoch 76/100
108/108 [==============================] - 42s 387ms/step - loss:
0.7312 - accuracy: 0.7129 - val_loss: 0.8934 - val_accuracy: 0.6736
Epoch 77/100
108/108 [==============================] - 42s 384ms/step - loss:
0.7181 - accuracy: 0.7077 - val_loss: 0.9299 - val_accuracy: 0.6713
Epoch 78/100
108/108 [==============================] - 42s 386ms/step - loss:
0.7273 - accuracy: 0.7097 - val_loss: 0.9268 - val_accuracy: 0.6667
Epoch 79/100
108/108 [==============================] - 42s 386ms/step - loss:
0.7127 - accuracy: 0.7144 - val_loss: 0.9365 - val_accuracy: 0.6644
Epoch 80/100
108/108 [==============================] - 42s 387ms/step - loss:
0.7324 - accuracy: 0.7054 - val_loss: 0.9378 - val_accuracy: 0.6621
Epoch 81/100
108/108 [==============================] - 42s 387ms/step - loss:
0.7037 - accuracy: 0.7207 - val_loss: 0.9389 - val_accuracy: 0.6506
```

```
Epoch 82/100
108/108 [==============================] - 42s 388ms/step - loss:
0.7174 - accuracy: 0.7112 - val_loss: 0.9787 - val_accuracy: 0.6483
Epoch 83/100
108/108 [==============================] - 42s 386ms/step - loss:
0.7061 - accuracy: 0.7158 - val_loss: 0.9303 - val_accuracy: 0.6667
Epoch 84/100
108/108 [==============================] - 42s 385ms/step - loss:
0.7308 - accuracy: 0.7092 - val_loss: 0.9532 - val_accuracy: 0.6621
Epoch 85/100
108/108 [==============================] - 42s 386ms/step - loss:
0.7252 - accuracy: 0.7129 - val_loss: 0.9492 - val_accuracy: 0.6368
Epoch 86/100
108/108 [==============================] - 42s 386ms/step - loss:
0.7115 - accuracy: 0.7187 - val_loss: 0.9897 - val_accuracy: 0.6391
Epoch 87/100
108/108 [==============================] - 42s 387ms/step - loss:
0.7083 - accuracy: 0.7164 - val_loss: 0.9388 - val_accuracy: 0.6460
Epoch 88/100
108/108 [==============================] - 42s 388ms/step - loss:
0.6887 - accuracy: 0.7233 - val_loss: 0.9965 - val_accuracy: 0.6621
Epoch 89/100
108/108 [==============================] - 42s 387ms/step - loss:
0.7146 - accuracy: 0.7118 - val_loss: 0.9266 - val_accuracy: 0.6828
Epoch 90/100
108/108 [==============================] - 42s 387ms/step - loss:
0.7030 - accuracy: 0.7196 - val_loss: 1.0215 - val_accuracy: 0.6460
Epoch 91/100
108/108 [==============================] - 42s 387ms/step - loss:
0.7047 - accuracy: 0.7181 - val_loss: 0.9960 - val_accuracy: 0.6621
Epoch 92/100
108/108 [==============================] - 42s 386ms/step - loss:
0.6951 - accuracy: 0.7199 - val_loss: 0.9756 - val_accuracy: 0.6552
Epoch 93/100
108/108 [==============================] - 42s 386ms/step - loss:
0.6975 - accuracy: 0.7207 - val_loss: 0.9423 - val_accuracy: 0.6621
Epoch 94/100
108/108 [==============================] - 42s 388ms/step - loss:
0.6998 - accuracy: 0.7210 - val_loss: 0.9770 - val_accuracy: 0.6506
Epoch 95/100
108/108 [==============================] - 42s 386ms/step - loss:
0.6922 - accuracy: 0.7262 - val_loss: 0.9723 - val_accuracy: 0.6759
Epoch 96/100
108/108 [==============================] - 42s 388ms/step - loss:
0.7057 - accuracy: 0.7196 - val_loss: 0.9675 - val_accuracy: 0.6644
Epoch 97/100
108/108 [==============================] - 42s 386ms/step - loss:
0.7245 - accuracy: 0.7097 - val_loss: 1.0012 - val_accuracy: 0.6621
Epoch 98/100
108/108 [==============================] - 42s 387ms/step - loss:
```

```
0.6791 - accuracy: 0.7251 - val_loss: 0.9566 - val_accuracy: 0.6782
Epoch 99/100
108/108 [==============================] - 42s 389ms/step - loss:
0.6794 - accuracy: 0.7231 - val_loss: 1.0116 - val_accuracy: 0.6782
Epoch 100/100
108/108 [==============================] - 42s 389ms/step - loss:
0.6852 - accuracy: 0.7225 - val_loss: 1.0187 - val_accuracy: 0.6598
```

SAVE THE MODEL

```
model.save("/content/drive/MyDrive/flowermodel.h5")
```

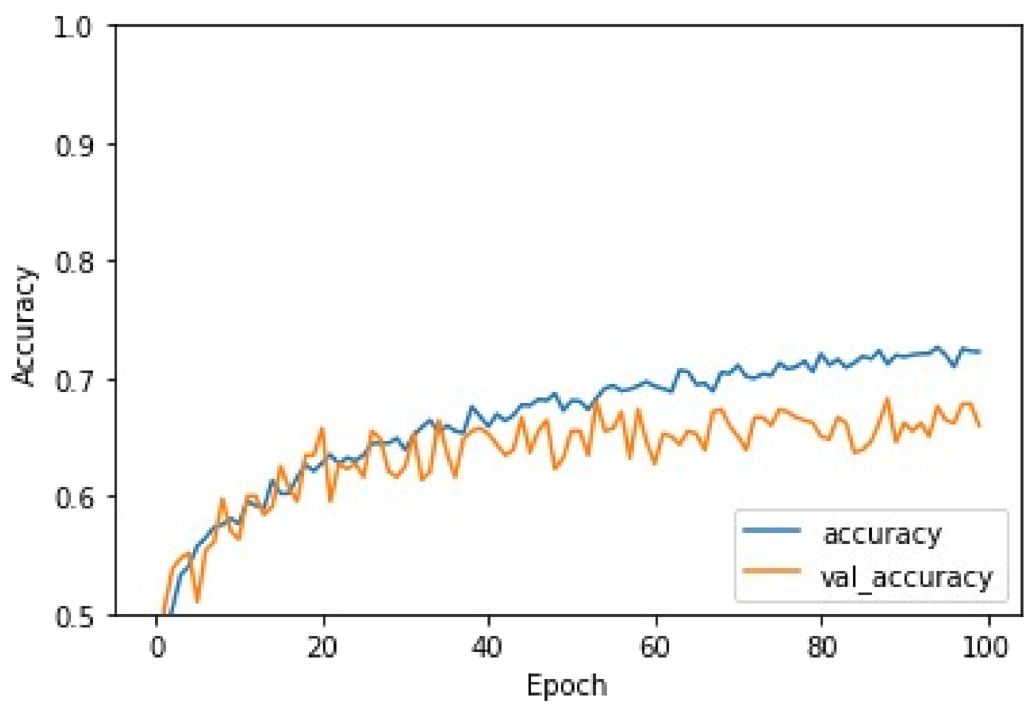
TEST THE MODEL

```
import matplotlib.pyplot as plt
```

```
plt.plot(tom.history['accuracy'], label='accuracy')
plt.plot(tom.history['val_accuracy'], label = 'val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0.5, 1])
plt.legend(loc='lower right')

test_loss, test_acc = model.evaluate(x_test, verbose=2)
```

```
14/14 - 3s - loss: 1.0187 - accuracy: 0.6598 - 3s/epoch - 250ms/step
```




```
print(test_acc)
```

```
0.659770131111145
```