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

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=9473189

Enter your authorization code:
··········
Mounted at /content/drive
```

New Section

New Section

```
!unzip -uq "/content/drive/My Drive/10-monkey-species.zip" -d "/content/drive/My Drive/"
```

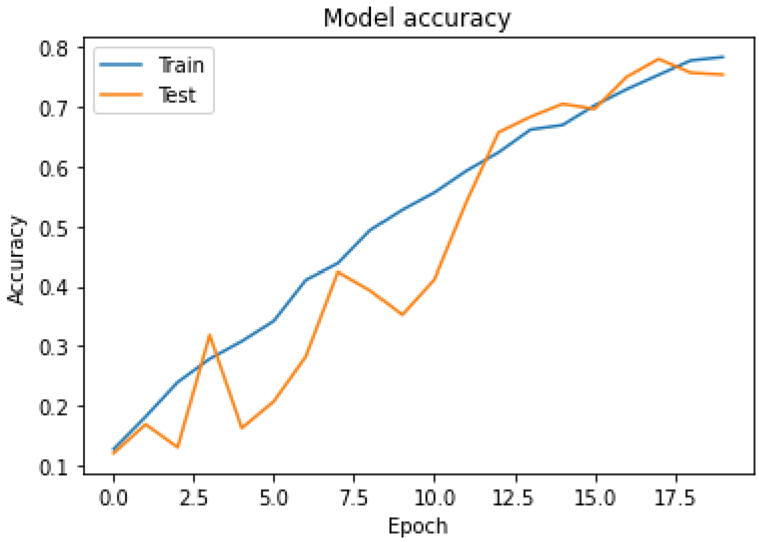
```
import keras
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dense, Flatten, BatchNormalization
from keras.layers import Dropout
```

```
#VGG16
model = Sequential()

#1st Layer
model.add(Conv2D(96,(3,3),input_shape=(128,128,3),activation="relu"))

#2nd Layer
model.add(Conv2D(128,(3,3), activation="relu"))
model.add(MaxPooling2D(pool_size=(2,2)))

#3rd Layer
model.add(Conv2D(128,(3,3),activation="relu"))

#4th Layer
model.add(Conv2D(256,(3,3), activation="relu"))
model.add(MaxPooling2D(pool_size=(2,2)))

#5th Layer
model.add(Conv2D(128,(3,3), activation="relu"))

#6th Layer
model.add(Conv2D(96,(3,3), activation="relu"))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(BatchNormalization())

#Flattening
model.add(Flatten())
model.add(Dense(units=64,activation="relu"))
model.add(Dropout(rate=0.))
model.add(Dense(units=32,activation="relu"))
model.add(Dropout(rate=0.2))
model.add(Dense(units=10, activation="softmax"))

from keras import optimizers
opt=optimizers.Adam(lr=0.00005)

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

from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2,
                                 zoom_range=0.2, horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)

training_set=train_datagen.flow_from_directory('/content/drive/My Drive/train_data',target_si
                                               batch_size=16,class_mode='categorical')
test_set=test_datagen.flow_from_directory('/content/drive/My Drive/test_data',target_size=(12
                                          batch_size=16,class_mode='categorical')

model.fit_generator(training_set,steps_per_epoch=5200/16,epochs=20,
                         validation_data=test_set,validation_steps=1300/16)
```

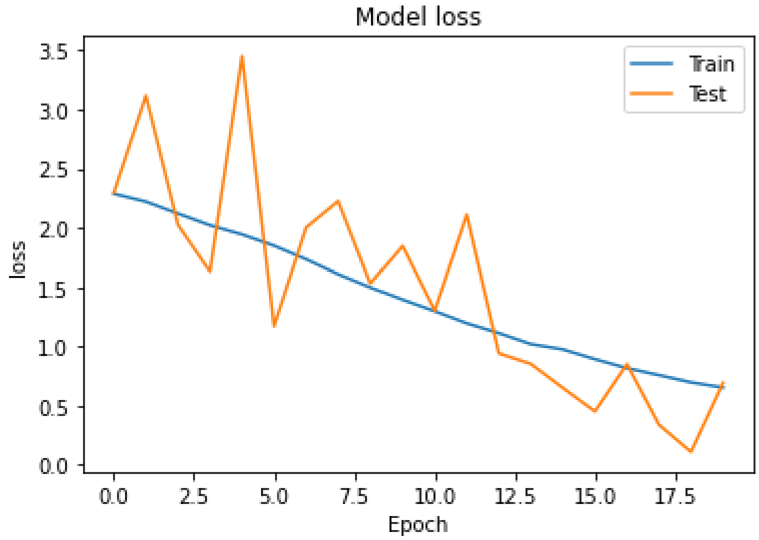
```
Found 5200 images belonging to 10 classes.
Found 1300 images belonging to 10 classes.
Epoch 1/20
325/325 [==============================] - 1956s 6s/step - loss: 2.2874 - accuracy: 0.12
Epoch 2/20
325/325 [==============================] - 285s 877ms/step - loss: 2.2209 - accuracy: 0.
Epoch 3/20
325/325 [==============================] - 286s 879ms/step - loss: 2.1206 - accuracy: 0.
Epoch 4/20
325/325 [==============================] - 285s 878ms/step - loss: 2.0218 - accuracy: 0.
Epoch 5/20
325/325 [==============================] - 285s 877ms/step - loss: 1.9444 - accuracy: 0.
Epoch 6/20
325/325 [==============================] - 285s 876ms/step - loss: 1.8499 - accuracy: 0.
Epoch 7/20
325/325 [==============================] - 285s 878ms/step - loss: 1.7357 - accuracy: 0.
Epoch 8/20
325/325 [==============================] - 287s 882ms/step - loss: 1.6093 - accuracy: 0.
Epoch 9/20
325/325 [==============================] - 287s 882ms/step - loss: 1.4975 - accuracy: 0.
Epoch 10/20
325/325 [==============================] - 282s 866ms/step - loss: 1.3945 - accuracy: 0.
Epoch 11/20
325/325 [==============================] - 289s 889ms/step - loss: 1.2978 - accuracy: 0.
Epoch 12/20
325/325 [==============================] - 281s 865ms/step - loss: 1.1960 - accuracy: 0.
Epoch 13/20
325/325 [==============================] - 282s 869ms/step - loss: 1.1127 - accuracy: 0.
Epoch 14/20
325/325 [==============================] - 284s 873ms/step - loss: 1.0200 - accuracy: 0.
Epoch 15/20
325/325 [==============================] - 283s 871ms/step - loss: 0.9751 - accuracy: 0.
Epoch 16/20
325/325 [==============================] - 284s 875ms/step - loss: 0.8927 - accuracy: 0.
Epoch 17/20
325/325 [==============================] - 283s 872ms/step - loss: 0.8168 - accuracy: 0.
Epoch 18/20
325/325 [==============================] - 282s 869ms/step - loss: 0.7579 - accuracy: 0.
Epoch 19/20
325/325 [==============================] - 282s 866ms/step - loss: 0.6969 - accuracy: 0.
Epoch 20/20
325/325 [==============================] - 284s 875ms/step - loss: 0.6530 - accuracy: 0.
<keras.callbacks.callbacks.History at 0x7f98c120bda0>
```

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

```
history=model.history
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```



```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper right')
plt.show()
```


Model loss
loss
Train
Test
Epoch