```
from keras.applications.vgg19 import VGG19
from keras import Sequential
from keras.layers import Flatten, Dense
from keras.preprocessing.image import ImageDataGenerator
import tensorflow as tf
```

```
! git clone https://github.com/Faris-ML/mask-detection.git
```

```
Cloning into 'mask-detection'...
remote: Enumerating objects: 11672, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 11672 (delta 11), reused 33 (delta 10), pack-reused 11638
Receiving objects: 100% (11672/11672), 400.38 MiB | 19.71 MiB/s, done.
Resolving deltas: 100% (11/11), done.
Checking out files: 100% (11804/11804), done.
```

```
# the data path -change the path for your own data-
train_dir = '/content/mask-detection/Face Mask Dataset/Train'
val_dir = '/content/mask-detection/Face Mask Dataset/Validation'
```

```
# make a augmentation generator for training data
train_datagen = ImageDataGenerator(rescale=1.0/255, horizontal_flip=True, zoom_range=0.2,shea
train_generator = train_datagen.flow_from_directory(directory=train_dir,target_size=(128, 128
```

```
Found 10000 images belonging to 2 classes.
```

```
# make a augmentation generator for validation data
val_datagen = ImageDataGenerator(rescale=1.0/255,horizontal_flip=True, zoom_range=0.2,shear_r
val_generator = train_datagen.flow_from_directory(directory=val_dir,target_size=(128, 128,),c
checkpoint = tf.keras.callbacks.ModelCheckpoint(
        "best_model",
        monitor="val_accuracy",
        mode="max",
        save_best_only=True)
```

```
Found 800 images belonging to 2 classes.
```

```
# load VVG19 architecture
vgg19 = VGG19(weights='imagenet', include_top=False, input_shape=(128, 128, 3))
for layer in vgg19.layers:
    layer.trainable = False
```

```
# build the model architecture and add some layers
model = Sequential()
model.add(vgg19)
```

```
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add (Dense(64,activation='relu'))
model.add (Dense(32,activation='relu'))
model.add (Dense(16,activation='relu'))
model.add (Dense(8,activation='relu'))
model.add (Dense(4,activation='relu'))
model.add(Dense(2,activation='softmax'))
model.summary()
model.compile(optimizer="adam",loss="categorical_crossentropy",metrics ="accuracy")
history = model.fit_generator(generator=train_generator,
                              epochs=20,validation_data=val_generator,callbacks=[checkpoint]
                              )
```

```
 dense_15 (Dense)            (None, 2)                 10

=================================================================
Total params: 21,084,134
Trainable params: 1,059,750
Non-trainable params: 20,024,384
_________________________________________________________________
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:15: UserWarning: `Model.
  from ipykernel import kernelapp as app
Epoch 1/20
313/313 [==============================] - ETA: 0s - loss: 0.1455 - accuracy: 0.9341I
313/313 [==============================] - 81s 254ms/step - loss: 0.1455 - accuracy:
Epoch 2/20
313/313 [==============================] - ETA: 0s - loss: 0.0671 - accuracy: 0.9751I
313/313 [==============================] - 78s 249ms/step - loss: 0.0671 - accuracy:
Epoch 3/20
313/313 [==============================] - 73s 234ms/step - loss: 0.0398 - accuracy:
Epoch 4/20
313/313 [==============================] - ETA: 0s - loss: 0.0417 - accuracy: 0.9843I
313/313 [==============================] - 78s 249ms/step - loss: 0.0417 - accuracy:
Epoch 5/20
313/313 [==============================] - ETA: 0s - loss: 0.0381 - accuracy: 0.9860I
313/313 [==============================] - 78s 249ms/step - loss: 0.0381 - accuracy:
Epoch 6/20
313/313 [==============================] - 73s 232ms/step - loss: 0.0379 - accuracy:
Epoch 7/20
313/313 [==============================] - 72s 231ms/step - loss: 0.0349 - accuracy:
Epoch 8/20
313/313 [==============================] - 73s 231ms/step - loss: 0.0346 - accuracy:
Epoch 9/20
313/313 [==============================] - ETA: 0s - loss: 0.0324 - accuracy: 0.9881I
313/313 [==============================] - 77s 245ms/step - loss: 0.0324 - accuracy:
Epoch 10/20
313/313 [==============================] - 72s 230ms/step - loss: 0.0269 - accuracy:
Epoch 11/20
313/313 [==============================] - 72s 229ms/step - loss: 0.0229 - accuracy:
Epoch 12/20
313/313 [==============================] - 72s 230ms/step - loss: 0.0294 - accuracy:
Epoch 13/20
313/313 [==============================] - 72s 230ms/step - loss: 0.0243 - accuracy:
Epoch 14/20
```

```
313/313 [==============================] - 72s 231ms/step - loss: 0.0273 - accuracy:
Epoch 15/20
313/313 [==============================] - ETA: 0s - loss: 0.0207 - accuracy: 0.9926I
313/313 [==============================] - 77s 246ms/step - loss: 0.0207 - accuracy:
Epoch 16/20
313/313 [==============================] - ETA: 0s - loss: 0.0203 - accuracy: 0.9928I
313/313 [==============================] - 77s 245ms/step - loss: 0.0203 - accuracy:
Epoch 17/20
313/313 [==============================] - 73s 234ms/step - loss: 0.0234 - accuracy:
Epoch 18/20
313/313 [==============================] - 73s 233ms/step - loss: 0.0261 - accuracy:
Epoch 19/20
313/313 [==============================] - 73s 232ms/step - loss: 0.0224 - accuracy:
Epoch 20/20
313/313 [==============================] - 73s 232ms/step - loss: 0.0219 - accuracy:
```

```
model.save('masknet.h5')
```