

```

from keras.applications.vgg19 import VGG19
from keras import Sequential
from keras.layers import Flatten, Dense
from keras.preprocessing.image import ImageDataGenerator

```

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

```

Cloning into 'mask-detection'...
remote: Enumerating objects: 11644, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 11644 (delta 0), reused 4 (delta 0), pack-reused 11638
Receiving objects: 100% (11644/11644), 400.24 MiB | 14.13 MiB/s, done.
Checking out files: 100% (11800/11800), 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, shear_range=0.2, rotation_range=15)
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_range=0.2, rotation_range=15)
val_generator = val_datagen.flow_from_directory(directory=val_dir, target_size=(128, 128))

```

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

```

# load VGG19 architecture
vgg19 = VGG19(weights='imagenet', include_top=False, input_shape=(128, 128, 3))

```

```

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg19
80142336/80134624 [=====] - 3s 0us/step
80150528/80134624 [=====] - 3s 0us/step

```



```

# build the model architecture and add some layers
model = Sequential()
model.add(vgg19)
model.add(Flatten())
model.add(Dense(100, 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,
                             )

```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
vgg19 (Functional)	(None, 4, 4, 512)	20024384
flatten_1 (Flatten)	(None, 8192)	0
dense_2 (Dense)	(None, 100)	819300
dense_3 (Dense)	(None, 2)	202

```

=====
Total params: 20,843,886
Trainable params: 20,843,886
Non-trainable params: 0

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:10: UserWarning: `Model
# Remove the CWD from sys.path while we load stuff.

```

Epoch 1/20

313/313 [=====] - 120s 335ms/step - loss: 0.7936 - accuracy

Epoch 2/20

313/313 [=====] - 100s 320ms/step - loss: 0.2124 - accuracy

Epoch 3/20

313/313 [=====] - 101s 321ms/step - loss: 0.1293 - accuracy

Epoch 4/20

313/313 [=====] - 101s 321ms/step - loss: 0.1223 - accuracy

Epoch 5/20

313/313 [=====] - 100s 320ms/step - loss: 0.1218 - accuracy

Epoch 6/20

313/313 [=====] - 100s 320ms/step - loss: 0.1225 - accuracy

Epoch 7/20

313/313 [=====] - 100s 320ms/step - loss: 0.0869 - accuracy

Epoch 8/20

313/313 [=====] - 100s 320ms/step - loss: 0.0796 - accuracy

Epoch 9/20

313/313 [=====] - 100s 319ms/step - loss: 0.0688 - accuracy

Epoch 10/20

313/313 [=====] - 100s 319ms/step - loss: 0.0657 - accuracy

Epoch 11/20

313/313 [=====] - 100s 319ms/step - loss: 0.1217 - accuracy

Epoch 12/20

313/313 [=====] - 100s 318ms/step - loss: 0.1009 - accuracy

Epoch 13/20

313/313 [=====] - 100s 318ms/step - loss: 0.0672 - accuracy

Epoch 14/20

313/313 [=====] - 100s 318ms/step - loss: 0.0583 - accuracy

Epoch 15/20

313/313 [=====] - 100s 318ms/step - loss: 0.0679 - accuracy

Epoch 16/20

```
313/313 [=====] - 100s 318ms/step - loss: 0.0641 - accuracy  
Epoch 17/20  
313/313 [=====] - 100s 318ms/step - loss: 0.0513 - accuracy  
Epoch 18/20  
313/313 [=====] - 101s 322ms/step - loss: 0.0532 - accuracy  
Epoch 19/20  
313/313 [=====] - 100s 319ms/step - loss: 0.0534 - accuracy
```

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

✓ 34m 26s completed at 4:45 PM



Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.