

```
from google.colab import drive
drive.mount('/content/gdrive')
root_path = 'gdrive/My Drive/Face Mask Dataset/'
```

Mounted at /content/gdrive

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import os
import shutil
import glob
import cv2
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import AveragePooling2D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import to_categorical
```

```
baseModel = VGG16(weights="imagenet", include_top = False, input_tensor=Input(shape=(224, 224, 3)))
headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(128, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(2, activation="softmax")(headModel)
```

```
model = Model(inputs=baseModel.input, outputs=headModel)
for layer in baseModel.layers:
    layer.trainable = False
```

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels/58892288/58889256 [=====] - 1s 0us/step
 58900480/58889256 [=====] - 1s 0us/step

```
print(model.summary())
```

Layer (type)	Output Shape	Param #
=====		
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808

block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
average_pooling2d (AveragePooling2D)	(None, 1, 1, 512)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 128)	65664
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258

```

=====
Total params: 14,780,610
Trainable params: 65,922
Non-trainable params: 14,714,688

```

None

```

rescale = 1./255
target_size = (224, 224)
batch_size = 32
class_mode = "categorical"

```

```

train_datagen = ImageDataGenerator(
    rescale=rescale,
    shear_range=0.2,

```

```
zoom_range=0.2,
horizontal_flip=True)
train_generator = train_datagen.flow_from_directory('gdrive/My Drive/Face Mask Dataset/Train/',
target_size=target_size,
class_mode=class_mode,
batch_size=batch_size,
shuffle=True)

validation_datagen = ImageDataGenerator(rescale=rescale)
validation_generator = validation_datagen.flow_from_directory('gdrive/My Drive/Face Mask Dataset/Test/',
target_size=target_size,
class_mode=class_mode,
batch_size=batch_size,
shuffle = False)
```

```
Found 10992 images belonging to 2 classes.
Found 992 images belonging to 2 classes.
```

```
steps_per_epoch=len(train_generator)
validation_steps=len(validation_generator)
loss='categorical_crossentropy'
metrics=['accuracy']
epochs = 1

opt = Adam(lr=1e-3, decay=1e-3 / 10)
model.compile(optimizer=opt, loss=loss, metrics=metrics)

history = model.fit_generator(
    train_generator,
    steps_per_epoch = steps_per_epoch,
    epochs=epochs,
    verbose=1,
    # callbacks=callbacks,
    validation_data=validation_generator,
    validation_steps=validation_steps)
```

```
# class_weight=class_weight
)
```

```
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/adam.py:105: UserWarning: The `lr` argument is deprecated, use `learn
super(Adam, self).__init__(name, **kwargs)
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:17: UserWarning: `Model.fit_generator` is deprecated and will be r
344/344 [=====] - 1535s 4s/step - loss: 0.1822 - accuracy: 0.9334 - val_loss: 0.0661 - val_accuracy: 0
```

```
from sklearn.metrics import precision_score, recall_score, f1_score, classification_report
```

```
print("results for training")
result = model.evaluate_generator(train_generator, verbose=1)
print("%s%.2f  %" ("Loss      : ", result[0]))
print("%s%.2f%s%" ("Accuracy : ", result[1]*100, "%"))
classes = os.listdir('gdrive/My Drive/Face Mask Dataset/Train/')

print("results for test")
result = model.evaluate_generator(validation_generator, steps=len(validation_generator), verbose=1)
print("%s%.2f  %" ("Loss      : ", result[0]))
print("%s%.2f%s%" ("Accuracy : ", result[1]*100, "%"))
print("results for test prediction")
```

```
results for training
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: UserWarning: `Model.evaluate_generator` is deprecated and will
after removing the cwd from sys.path.
344/344 [=====] - 148s 430ms/step - loss: 0.0827 - accuracy: 0.9720
Loss      : 0.08
Accuracy : 97.20%
results for test
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:11: UserWarning: `Model.evaluate_generator` is deprecated and will
# This is added back by InteractiveShellApp.init_path()
31/31 [=====] - 5s 162ms/step - loss: 0.0661 - accuracy: 0.9788
Loss      : 0.07
```

Accuracy : 97.88%
results for test prediction

```
y_pred = model.predict_generator(validation_generator, steps=len(validation_generator), verbose=1)
y_pred = y_pred.argmax(axis=-1)
y_true=validation_generator.classes
precision = precision_score(y_true, y_pred)
recall = recall_score(y_true, y_pred)
f1 = f1_score(y_true, y_pred)
print("-"*90)
print("Derived Report")
print("-"*90)
print("%s%.2f%s" % ("Precision      : ", precision*100, "%"))
print("%s%.2f%s" % ("Recall        : ", recall*100, "%"))
print("%s%.2f%s" % ("F1-Score     : ", f1*100, "%"))
print("-"*90)
print("\n\n")
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.predict_generator` is deprecated and will be removed in a future version.
"""Entry point for launching an IPython kernel.

31/31 [=====] - 5s 160ms/step

Derived Report

Precision	: 97.66%
Recall	: 98.23%
F1-Score	: 97.94%

-Made By Manish Kumar

✓ 0s completed at 2:11 PM

