

```
In [ ]: from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dropout, Flatten, Dense
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt
import os
```

```
In [ ]: data = '../input/covid-face-mask-detection-dataset/New Masks Dataset'
train = os.path.join(data, 'Train')
test = os.path.join(data, 'Test')
valid = os.path.join(data, 'Validation')

train_mask = os.path.join(train, 'Mask')
train_nomask = os.path.join(train, 'Non Mask')
train_mask = os.listdir(train_mask)
print(train_mask[15:25])

train_nomask_names = os.listdir(train_nomask)
print(train_nomask_names[15:25])

['0372.jpg', '1548.jpg', '0932.jpg', '0006.jpg', '0559.jpg', '1473.jpg', '1669.jpg', '0862.jpg', '1058.jpg', '1166.jpg']
['215.jpg', '185.jpg', '243.jpg', '153.jpg', '143.jpg', '327.jpg', '253.jpg', '115.jpg', '131.jpg', '151.jpg']
```

```
In [ ]: import matplotlib.image as mpimg
n_rows = 4
n_cols = 4
plt.figure(figsize=(12,12))
mask_picture = []
for i in train_mask[10:20]:
    mask_picture.append(os.path.join(train_mask,i))
nomask_pic = []
for i in train_nomask_names[10:20]:
    nomask_pic.append(os.path.join(train_nomask,i))
print(mask_picture)
print(nomask_pic)
merged_pics = mask_picture+nomask_pic
for i in range(0,len(merged_pics)):
    data = merged_pics[i].split('/')[4]
    sp = plt.subplot(4,4,i+1)
    sp.axis('Off')
    image = mpimg.imread(merged_pics[i])
    sp.set_title(data,fontsize=10)
    plt.imshow(image,cmap='gray')
plt.show()

['../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Mask/1037.png', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Mask/1488.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Mask/1539.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Mask/1409.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Mask/1154.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Mask/0372.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Mask/1548.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Mask/0932.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Mask/0006.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Mask/0559.jpg']
['../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Non Mask/109.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Non Mask/149.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Non Mask/187.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Non Mask/76.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Non Mask/270.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Non Mask/215.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Non Mask/185.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Non Mask/243.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Non Mask/153.jpg', '../input/covid-face-mask-detection-dataset/New Masks Dataset/Train/Non Mask/143.jpg']
```

```
ValueError                                Traceback (most recent call last)
/tmp/ipykernel_23/282198932.py in <module>
    19 for i in range(0,len(merged_pics)):
    20     data = merged_pics[i].split('/')[4]
--> 21     sp = plt.subplot(4,4,i+1)
    22     sp.axis('Off')
    23     image = mpimg.imread(merged_pics[i])

/opt/conda/lib/python3.7/site-packages/matplotlib/pyplot.py in subplot(*args, **kwargs)
   1285
   1286     # First, search for an existing subplot with a matching spec.
-> 1287     key = SubplotSpec._from_subplot_args(fig, args)
   1288
   1289     for ax in fig.axes:

/opt/conda/lib/python3.7/site-packages/matplotlib/gridspec.py in from_subplot_args(figure, args)
    607         if not isinstance(num, Integral) or num < 1 or num > rows*cols:
    608             raise ValueError(
--> 609                 f"num must be 1 <= num <= {rows*cols}, not {num!r}")
    610         i = j = num
    611         return gs[i-1:j]

ValueError: num must be 1 <= num <= 16, not 17
```

Train/Mask/1037.png



Train/Mask/1488.jpg



Train/Mask/1539.jpg



Train/Mask/1409.jpg



Train/Mask/1154.jpg



Train/Mask/0372.jpg



Train/Mask/1548.jpg



Train/Mask/0932.jpg



Train/Mask/0006.jpg



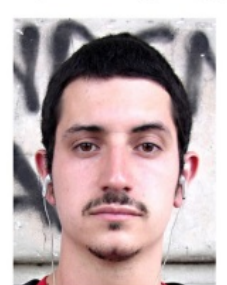
Train/Mask/0559.jpg



Train/Non Mask/109.jpg



Train/Non Mask/149.jpg



Train/Non Mask/187.jpg



Train/Non Mask/76.jpg



Train/Non Mask/270.jpg



Train/Non Mask/215.jpg



```
In [ ]: data_train = ImageDataGenerator(rescale=1./255, zoom_range = 0.2, rotation_range = 40, horizontal_flip = True)
data_test = ImageDataGenerator(rescale=1./255)
valid_data = ImageDataGenerator(rescale=1./255)
train_generator = data_train.flow_from_directory(train, target_size=(150,150), batch_size = 32, class_mode = 'binary')
test_generator = data_test.flow_from_directory(test, target_size=(150,150), batch_size = 32, class_mode = 'binary')
valid_generator = valid_data.flow_from_directory(valid, target_size=(150,150), batch_size = 32, class_mode = 'binary')
```

Found 600 images belonging to 2 classes.
 Found 100 images belonging to 2 classes.
 Found 306 images belonging to 2 classes.

```
In [ ]: model = Sequential()
model.add(Conv2D(32, (3,3), padding='SAME', activation='relu', input_shape=(150,150,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.5))

model.add(Conv2D(64, (3,3), padding='SAME', activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.5))

model.add(Flatten())

model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 150, 150, 32)	896
max_pooling2d (MaxPooling2D)	(None, 75, 75, 32)	0
dropout (Dropout)	(None, 75, 75, 32)	0
conv2d_1 (Conv2D)	(None, 75, 75, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 37, 37, 64)	0
dropout_1 (Dropout)	(None, 37, 37, 64)	0
flatten (Flatten)	(None, 87616)	0
dense (Dense)	(None, 256)	22429952
dropout_2 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 1)	257

=====
Total params: 22,449,601
Trainable params: 22,449,601
Non-trainable params: 0
=====

In []:

```
In [ ]: model.compile(Adam(lr=0.001),loss='binary_crossentropy',metrics=['accuracy'])
history = model.fit(train_generator,
                    epochs = 30,
                    validation_data = valid_generator)
```

Epoch 1/30

2023-02-26 18:37:36.855618: E tensorflow/core/grappler/optimizers/meta_optimizer.cc:954] layout failed: INVALID_ARGUMENT: Size of values 0 does not match size of permutation 4 @ fanin shape insequential/dropout/dropout/SelectV2-2-TransposeNHWCToNCHW-LayoutOptimizer

19/19 [=====] - 29s 1s/step - loss: 3.2225 - accuracy: 0.5117 - val_loss: 0.6895 - val_accuracy: 0.5000

Epoch 2/30

19/19 [=====] - 14s 735ms/step - loss: 0.6169 - accuracy: 0.6583 - val_loss: 0.6118 - val_accuracy: 0.7484

Epoch 3/30

19/19 [=====] - 14s 738ms/step - loss: 0.4569 - accuracy: 0.8150 - val_loss: 0.5496 - val_accuracy: 0.7876

Epoch 4/30

19/19 [=====] - 15s 784ms/step - loss: 0.3713 - accuracy: 0.8650 - val_loss: 0.4006 - val_accuracy: 0.8954

Epoch 5/30

19/19 [=====] - 14s 736ms/step - loss: 0.3395 - accuracy: 0.8733 - val_loss: 0.4136 - val_accuracy: 0.8758

Epoch 6/30

19/19 [=====] - 14s 770ms/step - loss: 0.2896 - accuracy: 0.8950 - val_loss: 0.3012 - val_accuracy: 0.9020

Epoch 7/30

19/19 [=====] - 14s 729ms/step - loss: 0.2623 - accuracy: 0.9117 - val_loss: 0.3023 - val_accuracy: 0.8954

Epoch 8/30

19/19 [=====] - 14s 765ms/step - loss: 0.2809 - accuracy: 0.9000 - val_loss: 0.3261 - val_accuracy: 0.9085

Epoch 9/30

19/19 [=====] - 14s 726ms/step - loss: 0.2689 - accuracy: 0.8983 - val_loss: 0.3134 - val_accuracy: 0.8889

Epoch 10/30

19/19 [=====] - 14s 718ms/step - loss: 0.2739 - accuracy: 0.9117 - val_loss: 0.3328 - val_accuracy: 0.8627

Epoch 11/30

19/19 [=====] - 14s 764ms/step - loss: 0.2388 - accuracy: 0.9150 - val_loss: 0.2821 - val_accuracy: 0.8987

Epoch 12/30

19/19 [=====] - 14s 742ms/step - loss: 0.2369 - accuracy: 0.9167 - val_loss: 0.2713 - val_accuracy: 0.9118

Epoch 13/30

19/19 [=====] - 14s 766ms/step - loss: 0.2213 - accuracy: 0.9183 - val_loss: 0.2613 - val_accuracy: 0.9020

Epoch 14/30

19/19 [=====] - 14s 720ms/step - loss: 0.2143 - accuracy: 0.9200 - val_loss: 0.2682 - val_accuracy: 0.8987

Epoch 15/30

19/19 [=====] - 19s 1s/step - loss: 0.2107 - accuracy: 0.9233 - val_loss: 0.2756 - val_accuracy: 0.8922

Epoch 16/30

```

19/19 [=====] - 14s 738ms/step - loss: 0.2120 - accuracy: 0.9333 - val_loss: 0.2495 -
val_accuracy: 0.8954
Epoch 17/30
19/19 [=====] - 14s 762ms/step - loss: 0.2124 - accuracy: 0.9250 - val_loss: 0.2390 -
val_accuracy: 0.9085
Epoch 18/30
19/19 [=====] - 14s 735ms/step - loss: 0.2016 - accuracy: 0.9250 - val_loss: 0.2399 -
val_accuracy: 0.9118
Epoch 19/30
19/19 [=====] - 14s 772ms/step - loss: 0.1940 - accuracy: 0.9317 - val_loss: 0.2446 -
val_accuracy: 0.9020
Epoch 20/30
19/19 [=====] - 14s 730ms/step - loss: 0.1896 - accuracy: 0.9267 - val_loss: 0.2154 -
val_accuracy: 0.9150
Epoch 21/30
19/19 [=====] - 14s 738ms/step - loss: 0.1921 - accuracy: 0.9317 - val_loss: 0.2840 -
val_accuracy: 0.8922
Epoch 22/30
19/19 [=====] - 14s 744ms/step - loss: 0.1966 - accuracy: 0.9183 - val_loss: 0.2454 -
val_accuracy: 0.8987
Epoch 23/30
19/19 [=====] - 13s 720ms/step - loss: 0.1678 - accuracy: 0.9333 - val_loss: 0.2236 -
val_accuracy: 0.9150
Epoch 24/30
19/19 [=====] - 14s 753ms/step - loss: 0.1769 - accuracy: 0.9350 - val_loss: 0.2261 -
val_accuracy: 0.9118
Epoch 25/30
19/19 [=====] - 14s 737ms/step - loss: 0.2136 - accuracy: 0.9133 - val_loss: 0.2432 -
val_accuracy: 0.9052
Epoch 26/30
19/19 [=====] - 14s 758ms/step - loss: 0.1939 - accuracy: 0.9317 - val_loss: 0.2632 -
val_accuracy: 0.8987
Epoch 27/30
19/19 [=====] - 13s 714ms/step - loss: 0.2045 - accuracy: 0.9250 - val_loss: 0.2260 -
val_accuracy: 0.9248
Epoch 28/30
19/19 [=====] - 14s 747ms/step - loss: 0.1600 - accuracy: 0.9367 - val_loss: 0.2407 -
val_accuracy: 0.9052
Epoch 29/30
19/19 [=====] - 14s 745ms/step - loss: 0.1564 - accuracy: 0.9450 - val_loss: 0.2461 -
val_accuracy: 0.9052
Epoch 30/30
19/19 [=====] - 14s 725ms/step - loss: 0.1853 - accuracy: 0.9200 - val_loss: 0.2513 -
val_accuracy: 0.8954

```

```
In [ ]: history.history.keys()
```

```
Out[ ]: dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

```
In [ ]: plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.legend(['trainig', 'validation'])
plt.title('Training and validation loss')
plt.xlabel('epoch')
```

```
Out[ ]: Text(0.5, 0, 'epoch')
```



```
In [ ]: plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
```

```
plt.legend(['trainig','validation'])
plt.title('Training and validation accuracy')
plt.xlabel('epoch')
```

Out[]: Text(0.5, 0, 'epoch')



```
In [ ]: test_loss , test_acc = model.evaluate(test_generator)
print('test acc :{} test loss:{}'.format(test_acc,test_loss))
```

4/4 [=====] - 2s 548ms/step - loss: 0.2142 - accuracy: 0.9000
test acc :0.8999999761581421 test loss:0.2141808718442917

```
In [ ]: import numpy as np
from google.colab import files
from keras.preprocessing import image

upload_file = files.upload()
for i in upload_file.keys():
    image_path='/content/'+i
    img = image.load_img(image_path , target_size=(150,150))
    images = image.img_to_array(img)
    images=np.expand_dims(images,axis=0)
    prediction = model.predict(images)
    print(i)
    if prediction==0:
        print('mask')
    else:
        print('nomask')
```