Modelling for Classification

Using wget to get the data directly from kaggle

In [1]:

```
!wget --header="Host: storage.googleapis.com" --header="User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.75 Safari/537.36" --header="Ac
cept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,ap
ation/signed-exchange;v=b3;q=0.9" --header="Accept-Language: en-US,en;q=0.9" --header="Referer: ht
tps://www.kaggle.com/" "https://storage.googleapis.com/kaggle-data-
sets/246422/519715/bundle/archive.zip?X-Goog-Algorithm=GOOG4-RSA-SHA256&X-Goog-Credential=gcp-kagg
le-com%40kaggle-161607.iam.gserviceaccount.com%2F20201026%2Fauto%2Fstorage%2Fgoog4_request&X-Goog-
Date=20201026T182225Z&X-Goog-Expires=259199&X-Goog-SignedHeaders=host&X-Goog-
Signature=8e9e9cf3ee34f89c77b410bfcb41565065ae8ed1a09b1418d24e345747f3f3885f5e559368be7b82acf2407aa
359108c06278e29adcebdde1b6c0be1e9cf5d6ca37e6e471951036d57ac11e9777c25ac61c565f07d2c0e936fd5871c8546
eccf43815490e7d7464ebef6001bfb425646ee21b26fc58f1981245a34e738c44323ae99b737e16f71081632f2e7b92a17f
baa935244e9670191b7c348816eab4ed3cb83cda741117984b1264fbd52f1775677df079865ab43d7fe8fd53b2a2632ee99
a996d9a3911d71f584232eb3b267498e0e2ebc16f64b665b477e11580269fee197358b0c2f3182f0926dd66beb42e032c6b
289c64c" -c -O 'archive.zip'
```

```
--2020-10-26 18:22:44--  https://storage.googleapis.com/kaggle-data-
sets/246422/519715/bundle/archive.zip?X-Goog-Algorithm=GOOG4-RSA-SHA256&X-Goog-Credential=gcp-kagg
le-com%40kaggle-161607.iam.gserviceaccount.com%2F20201026%2Fauto%2Fstorage%2Fgoog4_request&X-Goog-
Date=20201026T182225Z&X-Goog-Expires=259199&X-Goog-SignedHeaders=host&X-Goog-
Signature=8e9e9cf3ee34f89c77b410bfcb41565065ae8ed1a09b1418d24e345747f3f3885f5e559368be7b82acf2407aa
359108c06278e29adcebdde1b6c0be1e9cf5d6ca37e6e471951036d57ac11e9777c25ac61c565f07d2c0e936fd5871c8546
eccf43815490e7d7464ebef6001bfb425646ee21b26fc58f1981245a34e738c44323ae99b737e16f71081632f2e7b92a17f
baa935244e9670191b7c348816eab4ed3cb83cda741117984b1264fbd52f1775677df079865ab43d7fe8fd53b2a2632ee99
a996d9a3911d71f584232eb3b267498e0e2ebc16f64b665b477e11580269fee197358b0c2f3182f0926dd66beb42e032c6b
289c64c
Resolving storage.googleapis.com (storage.googleapis.com)... 64.233.184.128, 66.102.1.128,
64.233.167.128, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|64.233.184.128|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3256848012 (3.0G) [application/zip]
Saving to: ‘archive.zip’

archive.zip         100%[===================>]   3.03G  38.0MB/s    in 85s

2020-10-26 18:24:10 (36.4 MB/s) - ‘archive.zip’ saved [3256848012/3256848012]
```

Extracting the rar file

In [2]:

```python
from zipfile import ZipFile
zip_file = ZipFile('archive.zip','r')
zip_file.extractall()
```

Importing Libraries and reading the csv

In [4]:

```python
from tensorflow.keras import models, layers
from tensorflow.keras.models import Model
from tensorflow.keras.layers import BatchNormalization, Activation, Flatten
from tensorflow.keras.optimizers import Adam
from keras_preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import ReduceLROnPlateau,
ModelCheckpoint,EarlyStopping,LearningRateScheduler
import numpy as np
```

In [5]:

```
import tensorflow as tf
import tensorflow_io as tfio
```

In [3]:

```
!pip install -q tensorflow-io
!pip install pydicom
```

```
     |████████████████████████████████| 22.4MB 9.9MB/s
Collecting pydicom
  Downloading
https://files.pythonhosted.org/packages/d3/56/342e1f8ce5afe63bf65c23d0b2c1cd5a05600caad1c211c39725c
c56/pydicom-2.0.0-py3-none-any.whl (35.4MB)
     |████████████████████████████████| 35.5MB 74kB/s
Installing collected packages: pydicom
Successfully installed pydicom-2.0.0
```

In [6]:

```
import warnings
warnings.filterwarnings("ignore")
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import re
import os
import datetime as dt
from datetime import datetime
#!pip install pydicom
import pydicom as dicom
from tqdm.notebook import tqdm
from glob import glob
import pandas as pd
#reading all dcm files into train and text
train = sorted(glob("pneumothorax/dicom-images-train/*/*/*.dcm")) #There is an image after 2
subfolders . Rather than manually typing the entire path we are using glob to access the image wit
h ease
test = sorted(glob("pneumothorax/dicom-images-test/*/*/*.dcm"))

#reading the csv

dataset = pd.read_csv("pneumothorax/train-rle.csv", delimiter=",")
```

In [7]:

```
missing_images=0
train_df=[]
remove=[]
for i in tqdm(train):
  sample=dicom.dcmread(i) #reading each image
  train={}
  train["UID"]=sample.SOPInstanceUID
  try: #try and except to avoid throwing an error in case any file is missing
    encoded_pixels = dataset[dataset["ImageId"] == train["UID"]].values[0][1] #We are checking whea
ther each image(from the train) present has been mapped to the csv file given .
    train["EncodedPixels"]=encoded_pixels
  except:
    missing_images=missing_images+1
    remove.append("pneumothorax/dicom-images-train/" + sample.StudyInstanceUID + "/" + sample.Serie
sInstanceUID + "/" + sample.SOPInstanceUID + ".dcm")
     #if the image details are not present in the csv that means that the file is missing
  train["path"] = "pneumothorax/dicom-images-train/" + sample.StudyInstanceUID + "/" + sample.Serie
sInstanceUID + "/" + sample.SOPInstanceUID + ".dcm" #saving the path in csv for further reference
  train_df.append(train)

patients_train = pd.DataFrame(train_df,columns=["UID", "EncodedPixels","path"])

label=[]
for i in patients_train['EncodedPixels']:
  if str(i)==" -1":
    label.append(0)  #custom labelling based on the encoded pixels
    else:
        label.append(1)
patients_train['Label']=label
patients_train.head()
```

Out[7]:

	UID	EncodedPixels	path	Label
0	1.2.276.0.7230010.3.1.4.8323329.1000.151787516...	-1	pneumothorax/dicom-images-train/1.2.276.0.7230...	0
1	1.2.276.0.7230010.3.1.4.8323329.10000.15178752...	-1	pneumothorax/dicom-images-train/1.2.276.0.7230...	0
2	1.2.276.0.7230010.3.1.4.8323329.10001.15178752...	-1	pneumothorax/dicom-images-train/1.2.276.0.7230...	0
3	1.2.276.0.7230010.3.1.4.8323329.10002.15178752...	-1	pneumothorax/dicom-images-train/1.2.276.0.7230...	0
4	1.2.276.0.7230010.3.1.4.8323329.10003.15178752...	-1	pneumothorax/dicom-images-train/1.2.276.0.7230...	0

In [8]:

```
patients_train=patients_train.loc[~patients_train['path'].isin(remove)] #remove rows which do not h
ave images
```

In [ ]:

```
patients_train
```

Out[ ]:

	UID	EncodedPixels	path	Label
0	1.2.276.0.7230010.3.1.4.8323329.1000.151787516...	-1	pneumothorax/dicom-images-train/1.2.276.0.7230...	0
1	1.2.276.0.7230010.3.1.4.8323329.10000.15178752...	-1	pneumothorax/dicom-images-train/1.2.276.0.7230...	0
2	1.2.276.0.7230010.3.1.4.8323329.10001.15178752...	-1	pneumothorax/dicom-images-train/1.2.276.0.7230...	0
3	1.2.276.0.7230010.3.1.4.8323329.10002.15178752...	-1	pneumothorax/dicom-images-train/1.2.276.0.7230...	0
4	1.2.276.0.7230010.3.1.4.8323329.10003.15178752...	-1	pneumothorax/dicom-images-train/1.2.276.0.7230...	0
...	...	...	...	...
10707	1.2.276.0.7230010.3.1.4.8323329.5792.151787519...	-1	pneumothorax/dicom-images-train/1.2.276.0.7230...	0
10708	1.2.276.0.7230010.3.1.4.8323329.5793.151787519...	-1	pneumothorax/dicom-images-train/1.2.276.0.7230...	0
10709	1.2.276.0.7230010.3.1.4.8323329.5794.151787519...	-1	pneumothorax/dicom-images-train/1.2.276.0.7230...	0
10710	1.2.276.0.7230010.3.1.4.8323329.5795.151787519...	174459 17 982 47 952 76 943 79 936 83 937 83 ...	pneumothorax/dicom-images-train/1.2.276.0.7230...	1
10711	1.2.276.0.7230010.3.1.4.8323329.5796.151787519...	-1	pneumothorax/dicom-images-train/1.2.276.0.7230...	0

10675 rows × 4 columns

Preparing the Data

In [9]:

```
from tensorflow.keras.layers import Dense,Input,Conv2D,MaxPool2D,Activation,Dropout,Flatten
from tensorflow.keras.models import Model
import datetime
file_paths=patients_train['path'].values
labels=patients_train['Label'].values
```

In [10]:

```
list_ds = tf.data.Dataset.from_tensor_slices((file_paths,labels))
list_ds = list_ds.shuffle(len(patients_train),seed=42)
```

In [11]:

```
def decode_img(img):
  # convert the compressed string to a 3D uint8 tensor
  #image_bytes = tf.io.read_file(img)
  image = tfio.image.decode_dicom_image(img, dtype=tf.uint8,color_dim=True,scale='preserve')

  image = tf.image.convert_image_dtype(image, tf.float32)#converting the image to tf.float32
  image=tf.squeeze(image,[0]) #squeezing the image because the file is of the shape
(1,1024,1024,1) and we want (1024,1024,3)
  b = tf.constant([1,1,3], tf.int32)
  image=tf.tile(image,b)
  image=tf.image.resize(image,size=[256,256]) #the image is of the shape (1024,1024,1) to make it (
1024,1024,3) I am using tf.tile
  # resize the image to the desired size
  return image
```

In [12]:

```
def process_path(file_path,label):
  img = tf.io.read_file(file_path) #reading the image from the file path
  img = decode_img(img) #passing the image to the function
  return img,label
```

In [13]:

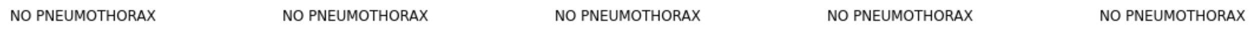
```
AUTOTUNE = tf.data.experimental.AUTOTUNE
list_ds = list_ds.map(process_path,num_parallel_calls=AUTOTUNE) #mapping the file paths to the
above function
val_size = int(len(patients_train) * 0.2) #splitting to 80-20 data
train_ds = list_ds.skip(val_size)
val_ds = list_ds.take(val_size)
```

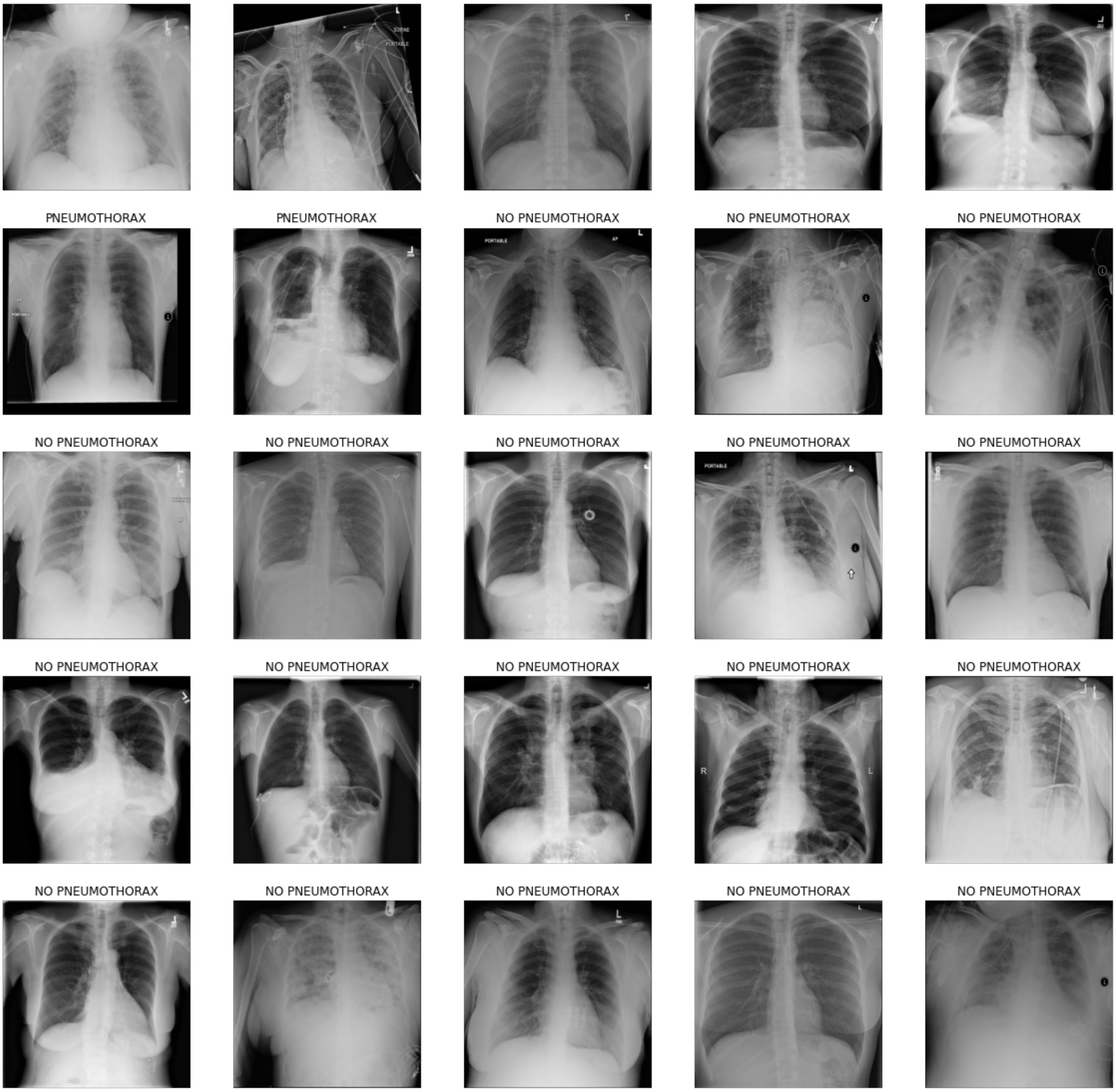
In [ ]:

```
val_ds = val_ds.batch(64, drop_remainder=True)
```

In [ ]:

```
plt.figure(figsize=(20,20))                    #plotting images from the train as a sanity check o
f whether the data has been properly converted or not
count=0
for i,j in tqdm(train_ds.take(25)):
  ax = plt.subplot(5,5,count+1)
  count=count+1
  if j==0:
    plt.title("NO PNEUMOTHORAX")
  else:
    plt.title("PNEUMOTHORAX")
  plt.imshow(i)
  plt.axis("off")
```





Training the model

In [14]:

```
%load_ext tensorboard
```

In [15]:

```
!rm -rf ./logs/
```

In [16]:

```
checkpoint_path = "training_1/cp.ckpt"
checkpoint_dir = os.path.dirname(checkpoint_path)
# Tensorbaord
! rm -rf ./logs/
logdir = os.path.join("logs", datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
%tensorboard --logdir $logdir
tensorboard_callback = tf.keras.callbacks.TensorBoard(logdir, histogram_freq=1)
```

In [23]:

```
import os
os.mkdir("model_save")
```

In [24]:

```
from tensorflow.keras.callbacks import ModelCheckpoint
filepath="model_save/weights-{epoch:02d}-{val_recall:.4f}.hdf5"
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_recall',verbose=1, save_best_only=True
, mode='max')
```

In [ ]:

```
tf.keras.backend.clear_session()
checkpoint_path = "training_1/cp.ckpt"
checkpoint_dir = os.path.dirname(checkpoint_path)
# Tensorbaord
! rm -rf ./logs/
logdir = os.path.join("logs", datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
%tensorboard --logdir $logdir
tensorboard_callback = tf.keras.callbacks.TensorBoard(logdir, histogram_freq=1)

model_1 =tf.keras.applications.vgg16.VGG16(weights = "imagenet", include_top=False, input_shape = (
256,256,3))
for i in model_1.layers:
  i.trainable=False
model=model_1.output
model=Conv2D(32, (3, 3))(model)
model=(Activation('relu'))(model)
model=(MaxPool2D(pool_size=(2, 2)))(model)
model=Flatten()(model)
model = Dense(256, activation="relu")(model)
model = Dense(128, activation="relu")(model)
output_layer = Dense(1, activation="sigmoid")(model)
model1 = Model(model_1.input,output_layer)
model1.compile(loss = "binary_crossentropy", optimizer =tf.keras.optimizers.Adam(lr=0.0001), metric
s=["accuracy",
tf.keras.metrics.Precision(name='precision'),tf.keras.metrics.Recall(name='recall')])
model1.fit(train_ds,epochs=30,verbose=True,validation_data=val_ds,batch_size=64,callbacks=[checkpoi
nt,tensorboard_callback])
```

```
Epoch 1/30
  1/133 [..............................] - ETA: 0s - loss: 0.7870 - accuracy: 0.1875 - precision:
0.1875 - recall: 1.0000WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/summary_ops_v2.py:1277: stop (from
tensorflow.python.eager.profiler) is deprecated and will be removed after 2020-07-01.
Instructions for updating:
use `tf.profiler.experimental.stop` instead.
133/133 [==============================] - ETA: 0s - loss: 0.4963 - accuracy: 0.7709 - precision:
0.3179 - recall: 0.0253
Epoch 00001: val_recall improved from -inf to 0.10448, saving model to model_save/weights-01-0.104
5.hdf5
133/133 [==============================] - 234s 2s/step - loss: 0.4963 - accuracy: 0.7709 - precis
ion: 0.3179 - recall: 0.0253 - val_loss: 0.4525 - val_accuracy: 0.7902 - val_precision: 0.6806 - v
al_recall: 0.1045
Epoch 2/30
133/133 [==============================] - ETA: 0s - loss: 0.4287 - accuracy: 0.8013 - precision:
0.6625 - recall: 0.2214
Epoch 00002: val_recall improved from 0.10448 to 0.33191, saving model to model_save/weights-02-0.
3319.hdf5
133/133 [==============================] - 233s 2s/step - loss: 0.4287 - accuracy: 0.8013 - precis
ion: 0.6625 - recall: 0.2214 - val_loss: 0.3946 - val_accuracy: 0.8153 - val_precision: 0.6652 - v
al_recall: 0.3319
Epoch 3/30
133/133 [==============================] - ETA: 0s - loss: 0.3972 - accuracy: 0.8146 - precision:
0.6680 - recall: 0.3438
Epoch 00003: val_recall improved from 0.33191 to 0.33962, saving model to model_save/weights-03-0.
3396.hdf5
133/133 [==============================] - 235s 2s/step - loss: 0.3972 - accuracy: 0.8146 - precis
ion: 0.6680 - recall: 0.3438 - val_loss: 0.3732 - val_accuracy: 0.8215 - val_precision: 0.7232 - v
al_recall: 0.3396
Epoch 4/30
133/133 [==============================] - ETA: 0s - loss: 0.3802 - accuracy: 0.8265 - precision:
0.6886 - recall: 0.4205
Epoch 00004: val_recall improved from 0.33962 to 0.41043, saving model to model_save/weights-04-0.
```

```
4104.hdf5
133/133 [==============================] - 234s 2s/step - loss: 0.3802 - accuracy: 0.8265 - precis
ion: 0.6886 - recall: 0.4205 - val_loss: 0.3473 - val_accuracy: 0.8466 - val_precision: 0.7388 - v
al_recall: 0.4104
Epoch 5/30
133/133 [==============================] - ETA: 0s - loss: 0.3610 - accuracy: 0.8394 - precision:
0.7139 - recall: 0.4572
Epoch 00005: val_recall did not improve from 0.41043
133/133 [==============================] - 239s 2s/step - loss: 0.3610 - accuracy: 0.8394 - precis
ion: 0.7139 - recall: 0.4572 - val_loss: 0.3376 - val_accuracy: 0.8414 - val_precision: 0.7882 - v
al_recall: 0.3540
Epoch 6/30
133/133 [==============================] - ETA: 0s - loss: 0.3479 - accuracy: 0.8475 - precision:
0.7319 - recall: 0.5000
Epoch 00006: val_recall improved from 0.41043 to 0.48315, saving model to model_save/weights-06-0.
4831.hdf5
133/133 [==============================] - 237s 2s/step - loss: 0.3479 - accuracy: 0.8475 - precis
ion: 0.7319 - recall: 0.5000 - val_loss: 0.3161 - val_accuracy: 0.8646 - val_precision: 0.7934 - v
al_recall: 0.4831
Epoch 7/30
133/133 [==============================] - ETA: 0s - loss: 0.3306 - accuracy: 0.8564 - precision:
0.7502 - recall: 0.5465
Epoch 00007: val_recall improved from 0.48315 to 0.48454, saving model to model_save/weights-07-0.
4845.hdf5
133/133 [==============================] - 238s 2s/step - loss: 0.3306 - accuracy: 0.8564 - precis
ion: 0.7502 - recall: 0.5465 - val_loss: 0.3204 - val_accuracy: 0.8532 - val_precision: 0.7966 - v
al_recall: 0.4845
Epoch 8/30
133/133 [==============================] - ETA: 0s - loss: 0.3140 - accuracy: 0.8694 - precision:
0.7758 - recall: 0.5798
Epoch 00008: val_recall did not improve from 0.48454
133/133 [==============================] - 240s 2s/step - loss: 0.3140 - accuracy: 0.8694 - precis
ion: 0.7758 - recall: 0.5798 - val_loss: 0.2902 - val_accuracy: 0.8655 - val_precision: 0.8583 - v
al_recall: 0.4599
Epoch 9/30
133/133 [==============================] - ETA: 0s - loss: 0.2967 - accuracy: 0.8770 - precision:
0.7875 - recall: 0.6188
Epoch 00009: val_recall improved from 0.48454 to 0.60613, saving model to model_save/weights-09-0.
6061.hdf5
133/133 [==============================] - 240s 2s/step - loss: 0.2967 - accuracy: 0.8770 - precis
ion: 0.7875 - recall: 0.6188 - val_loss: 0.2679 - val_accuracy: 0.8864 - val_precision: 0.8220 - v
al_recall: 0.6061
Epoch 10/30
133/133 [==============================] - ETA: 0s - loss: 0.2710 - accuracy: 0.8924 - precision:
0.8222 - recall: 0.6614
Epoch 00010: val_recall improved from 0.60613 to 0.73181, saving model to model_save/weights-10-0.
7318.hdf5
133/133 [==============================] - 239s 2s/step - loss: 0.2710 - accuracy: 0.8924 - precis
ion: 0.8222 - recall: 0.6614 - val_loss: 0.2306 - val_accuracy: 0.9205 - val_precision: 0.9003 - v
al_recall: 0.7318
Epoch 11/30
133/133 [==============================] - ETA: 0s - loss: 0.2649 - accuracy: 0.8927 - precision:
0.8184 - recall: 0.6670
Epoch 00011: val_recall improved from 0.73181 to 0.84989, saving model to model_save/weights-11-0.
8499.hdf5
133/133 [==============================] - 237s 2s/step - loss: 0.2649 - accuracy: 0.8927 - precis
ion: 0.8184 - recall: 0.6670 - val_loss: 0.2558 - val_accuracy: 0.9048 - val_precision: 0.7556 - v
al_recall: 0.8499
Epoch 12/30
133/133 [==============================] - ETA: 0s - loss: 0.2452 - accuracy: 0.9024 - precision:
0.8293 - recall: 0.7105
Epoch 00012: val_recall did not improve from 0.84989
133/133 [==============================] - 236s 2s/step - loss: 0.2452 - accuracy: 0.9024 - precis
ion: 0.8293 - recall: 0.7105 - val_loss: 0.2234 - val_accuracy: 0.9186 - val_precision: 0.8075 - v
al_recall: 0.8316
Epoch 13/30
133/133 [==============================] - ETA: 0s - loss: 0.2240 - accuracy: 0.9133 - precision:
0.8527 - recall: 0.7373
Epoch 00013: val_recall did not improve from 0.84989
133/133 [==============================] - 237s 2s/step - loss: 0.2240 - accuracy: 0.9133 - precis
ion: 0.8527 - recall: 0.7373 - val_loss: 0.1906 - val_accuracy: 0.9347 - val_precision: 0.8949 - v
al_recall: 0.8046
Epoch 14/30
133/133 [==============================] - ETA: 0s - loss: 0.2014 - accuracy: 0.9243 - precision:
0.8761 - recall: 0.7689
Epoch 00014: val_recall did not improve from 0.84989
133/133 [==============================] - 237s 2s/step - loss: 0.2014 - accuracy: 0.9243 - precis
```

```
ion: 0.8761 - recall: 0.7689 - val_loss: 0.1930 - val_accuracy: 0.9313 - val_precision: 0.8619 - v
al_recall: 0.8391
Epoch 15/30
133/133 [==============================] - ETA: 0s - loss: 0.1842 - accuracy: 0.9307 - precision:
0.8871 - recall: 0.7847
Epoch 00015: val_recall did not improve from 0.84989
133/133 [==============================] - 238s 2s/step - loss: 0.1842 - accuracy: 0.9307 - precis
ion: 0.8871 - recall: 0.7847 - val_loss: 0.1637 - val_accuracy: 0.9389 - val_precision: 0.9467 - v
al_recall: 0.7755
Epoch 16/30
133/133 [==============================] - ETA: 0s - loss: 0.1650 - accuracy: 0.9423 - precision:
0.9058 - recall: 0.8256
Epoch 00016: val_recall did not improve from 0.84989
133/133 [==============================] - 241s 2s/step - loss: 0.1650 - accuracy: 0.9423 - precis
ion: 0.9058 - recall: 0.8256 - val_loss: 0.1403 - val_accuracy: 0.9531 - val_precision: 0.9509 - v
al_recall: 0.8392
Epoch 17/30
133/133 [==============================] - ETA: 0s - loss: 0.1486 - accuracy: 0.9483 - precision:
0.9195 - recall: 0.8435
Epoch 00017: val_recall improved from 0.84989 to 0.85776, saving model to model_save/weights-17-0.
8578.hdf5
133/133 [==============================] - 241s 2s/step - loss: 0.1486 - accuracy: 0.9483 - precis
ion: 0.9195 - recall: 0.8435 - val_loss: 0.1331 - val_accuracy: 0.9564 - val_precision: 0.9387 - v
al_recall: 0.8578
Epoch 18/30
133/133 [==============================] - ETA: 0s - loss: 0.1351 - accuracy: 0.9535 - precision:
0.9239 - recall: 0.8600
Epoch 00018: val_recall did not improve from 0.85776
133/133 [==============================] - 238s 2s/step - loss: 0.1351 - accuracy: 0.9535 - precis
ion: 0.9239 - recall: 0.8600 - val_loss: 0.1095 - val_accuracy: 0.9631 - val_precision: 0.9786 - v
al_recall: 0.8565
Epoch 19/30
133/133 [==============================] - ETA: 0s - loss: 0.1202 - accuracy: 0.9616 - precision:
0.9367 - recall: 0.8878
Epoch 00019: val_recall improved from 0.85776 to 0.97083, saving model to model_save/weights-19-0.
9708.hdf5
133/133 [==============================] - 234s 2s/step - loss: 0.1202 - accuracy: 0.9616 - precis
ion: 0.9367 - recall: 0.8878 - val_loss: 0.1150 - val_accuracy: 0.9688 - val_precision: 0.8996 - v
al_recall: 0.9708
Epoch 20/30
133/133 [==============================] - ETA: 0s - loss: 0.1004 - accuracy: 0.9690 - precision:
0.9504 - recall: 0.9083
Epoch 00020: val_recall did not improve from 0.97083
133/133 [==============================] - 231s 2s/step - loss: 0.1004 - accuracy: 0.9690 - precis
ion: 0.9504 - recall: 0.9083 - val_loss: 0.0824 - val_accuracy: 0.9811 - val_precision: 0.9670 - v
al_recall: 0.9461
Epoch 21/30
133/133 [==============================] - ETA: 0s - loss: 0.0815 - accuracy: 0.9780 - precision:
0.9712 - recall: 0.9284
Epoch 00021: val_recall did not improve from 0.97083
133/133 [==============================] - 230s 2s/step - loss: 0.0815 - accuracy: 0.9780 - precis
ion: 0.9712 - recall: 0.9284 - val_loss: 0.0719 - val_accuracy: 0.9815 - val_precision: 0.9910 - v
al_recall: 0.9262
Epoch 22/30
133/133 [==============================] - ETA: 0s - loss: 0.0764 - accuracy: 0.9785 - precision:
0.9626 - recall: 0.9410
Epoch 00022: val_recall improved from 0.97083 to 0.97737, saving model to model_save/weights-22-0.
9774.hdf5
133/133 [==============================] - 229s 2s/step - loss: 0.0764 - accuracy: 0.9785 - precis
ion: 0.9626 - recall: 0.9410 - val_loss: 0.0572 - val_accuracy: 0.9886 - val_precision: 0.9734 - v
al_recall: 0.9774
Epoch 23/30
133/133 [==============================] - ETA: 0s - loss: 0.0631 - accuracy: 0.9867 - precision:
0.9811 - recall: 0.9588
Epoch 00023: val_recall did not improve from 0.97737
133/133 [==============================] - 228s 2s/step - loss: 0.0631 - accuracy: 0.9867 - precis
ion: 0.9811 - recall: 0.9588 - val_loss: 0.0509 - val_accuracy: 0.9924 - val_precision: 0.9921 - v
al_recall: 0.9767
Epoch 24/30
133/133 [==============================] - ETA: 0s - loss: 0.0521 - accuracy: 0.9891 - precision:
0.9854 - recall: 0.9651
Epoch 00024: val_recall improved from 0.97737 to 0.99142, saving model to model_save/weights-24-0.
9914.hdf5
133/133 [==============================] - 228s 2s/step - loss: 0.0521 - accuracy: 0.9891 - precis
ion: 0.9854 - recall: 0.9651 - val_loss: 0.0420 - val_accuracy: 0.9948 - val_precision: 0.9851 - v
al_recall: 0.9914
Epoch 25/30
```

```
133/133 [==============================] - ETA: 0s - loss: 0.0437 - accuracy: 0.9924 - precision:
0.9877 - recall: 0.9778
Epoch 00025: val_recall did not improve from 0.99142
133/133 [==============================] - 229s 2s/step - loss: 0.0437 - accuracy: 0.9924 - precis
ion: 0.9877 - recall: 0.9778 - val_loss: 0.0305 - val_accuracy: 0.9957 - val_precision: 0.9891 - v
al_recall: 0.9913
Epoch 26/30
133/133 [==============================] - ETA: 0s - loss: 0.0374 - accuracy: 0.9938 - precision:
0.9905 - recall: 0.9816
Epoch 00026: val_recall improved from 0.99142 to 0.99360, saving model to model_save/weights-26-0.
9936.hdf5
133/133 [==============================] - 227s 2s/step - loss: 0.0374 - accuracy: 0.9938 - precis
ion: 0.9905 - recall: 0.9816 - val_loss: 0.0306 - val_accuracy: 0.9976 - val_precision: 0.9957 - v
al_recall: 0.9936
Epoch 27/30
133/133 [==============================] - ETA: 0s - loss: 0.0320 - accuracy: 0.9947 - precision:
0.9904 - recall: 0.9857
Epoch 00027: val_recall improved from 0.99360 to 1.00000, saving model to model_save/weights-27-1.
0000.hdf5
133/133 [==============================] - 227s 2s/step - loss: 0.0320 - accuracy: 0.9947 - precis
ion: 0.9904 - recall: 0.9857 - val_loss: 0.0239 - val_accuracy: 0.9995 - val_precision: 0.9980 - v
al_recall: 1.0000
Epoch 28/30
133/133 [==============================] - ETA: 0s - loss: 0.0248 - accuracy: 0.9966 - precision:
0.9930 - recall: 0.9914
Epoch 00028: val_recall did not improve from 1.00000
133/133 [==============================] - 229s 2s/step - loss: 0.0248 - accuracy: 0.9966 - precis
ion: 0.9930 - recall: 0.9914 - val_loss: 0.0177 - val_accuracy: 1.0000 - val_precision: 1.0000 - v
al_recall: 1.0000
Epoch 29/30
133/133 [==============================] - ETA: 0s - loss: 0.0216 - accuracy: 0.9975 - precision:
0.9973 - recall: 0.9915
Epoch 00029: val_recall did not improve from 1.00000
133/133 [==============================] - 229s 2s/step - loss: 0.0216 - accuracy: 0.9975 - precis
ion: 0.9973 - recall: 0.9915 - val_loss: 0.0160 - val_accuracy: 0.9995 - val_precision: 1.0000 - v
al_recall: 0.9979
Epoch 30/30
133/133 [==============================] - ETA: 0s - loss: 0.0179 - accuracy: 0.9984 - precision:
0.9974 - recall: 0.9953
Epoch 00030: val_recall did not improve from 1.00000
133/133 [==============================] - 228s 2s/step - loss: 0.0179 - accuracy: 0.9984 - precis
ion: 0.9974 - recall: 0.9953 - val_loss: 0.0127 - val_accuracy: 0.9991 - val_precision: 0.9957 - v
al_recall: 1.0000
```

Out[ ]:

```
<tensorflow.python.keras.callbacks.History at 0x7fecac056748>
```

In [ ]:

```
from keras.models import load_model

model1.save('weights-30-1.0000.h5') # creates a HDF5 file 'my_model.h5' #chose model at epoch 30 b
ecause it looked reasonable
```

In [ ]:

```
model1.evaluate(val_ds)
```

```
33/33 [==============================] - 47s 1s/step - loss: 0.0135 - accuracy: 0.9991 -
precision: 0.9979 - recall: 0.9979
```

Out[ ]:

```
[0.01349087804555893,
 0.9990530014038086,
 0.9979423880577087,
 0.9979423880577087]
```

In [ ]:

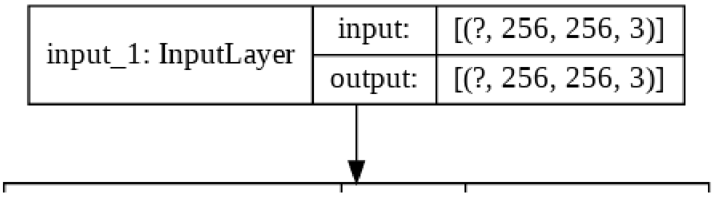
```
model1.summary()
```

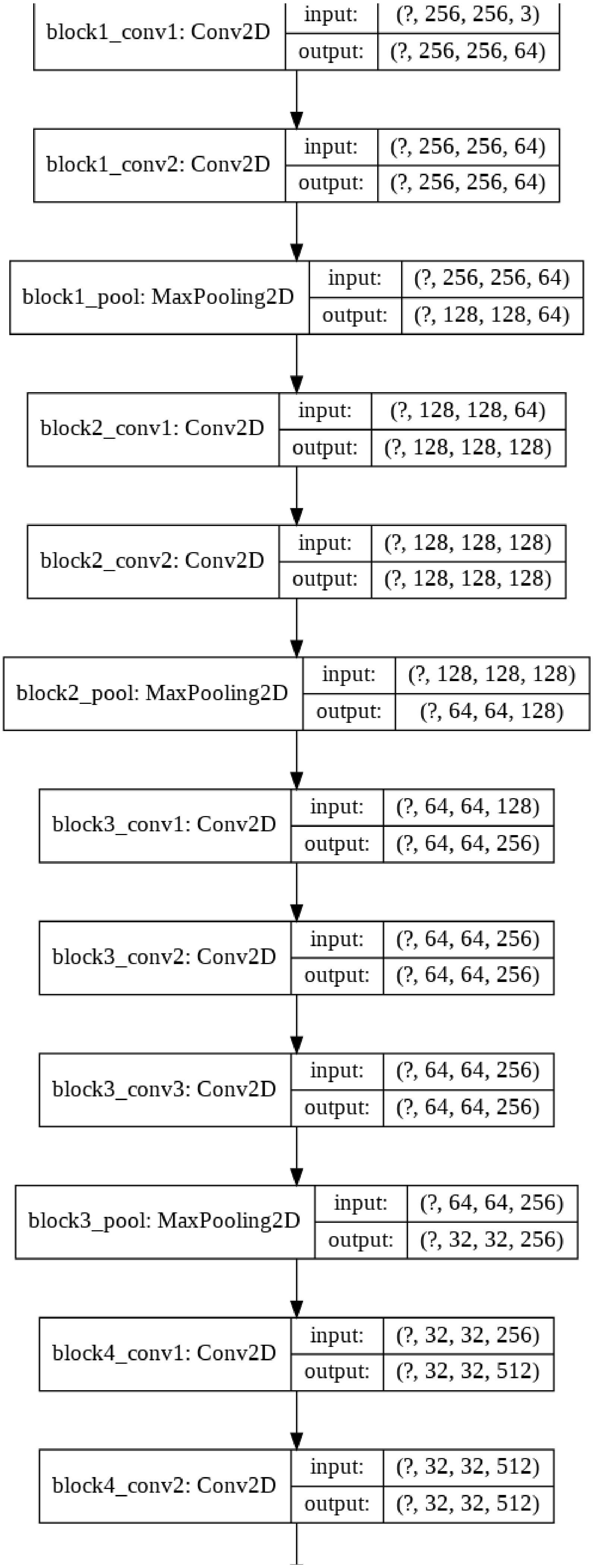
```
Model: "functional_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_1 (InputLayer)         [(None, 256, 256, 3)]     0         
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 256, 256, 64)      1792      
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 256, 256, 64)      36928     
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 128, 128, 64)      0         
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 128, 128, 128)     73856     
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 128, 128, 128)     147584    
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 64, 64, 128)       0         
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 64, 64, 256)       295168    
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 64, 64, 256)       590080    
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 64, 64, 256)       590080    
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 32, 32, 256)       0         
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 32, 32, 512)       1180160   
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 32, 32, 512)       2359808   
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 32, 32, 512)       2359808   
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 16, 16, 512)       0         
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 16, 16, 512)       2359808   
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 16, 16, 512)       2359808   
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 16, 16, 512)       2359808   
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 8, 8, 512)         0         
_________________________________________________________________
conv2d (Conv2D)              (None, 6, 6, 32)          147488    
_________________________________________________________________
activation (Activation)      (None, 6, 6, 32)          0         
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 3, 3, 32)          0         
_________________________________________________________________
flatten (Flatten)            (None, 288)               0         
_________________________________________________________________
dense (Dense)                (None, 256)               73984     
_________________________________________________________________
dense_1 (Dense)              (None, 128)               32896     
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 129       
=================================================================
Total params: 14,969,185
Trainable params: 254,497
Non-trainable params: 14,714,688
_________________________________________________________________
```

In [ ]:

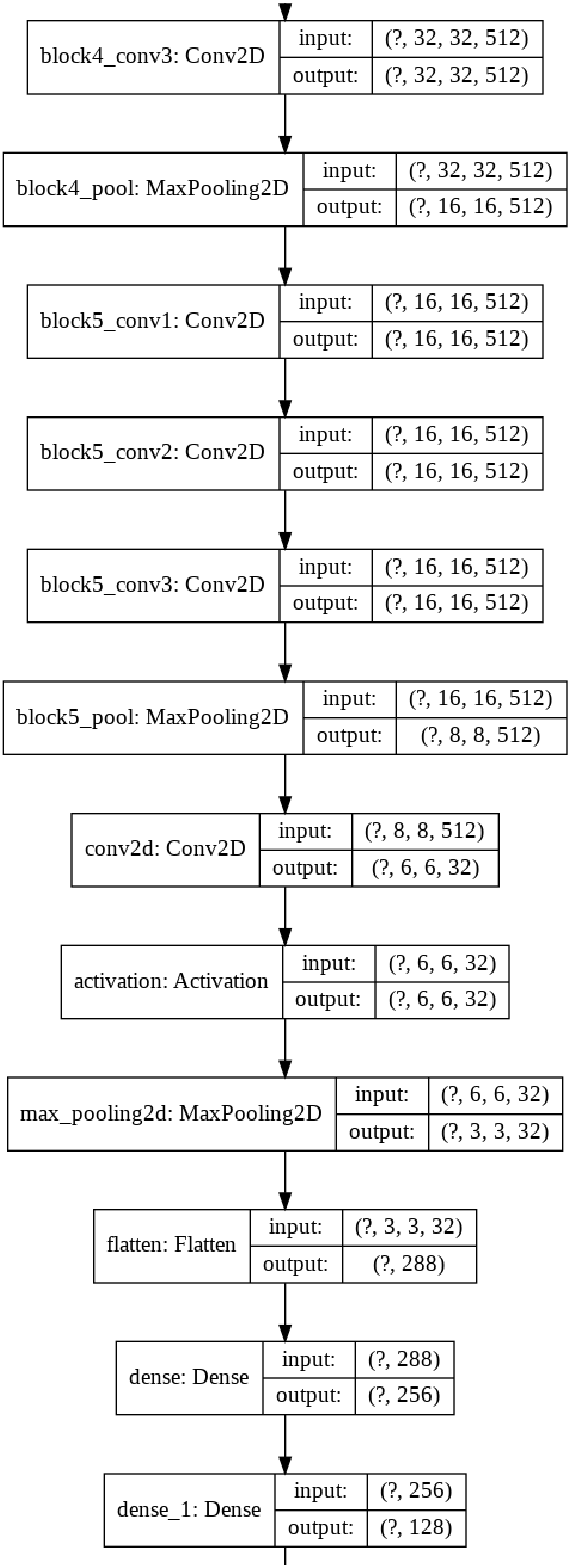
```python
from tensorflow.keras.utils import plot_model
plot_model(model1, 'model1.png', show_shapes=True)
```

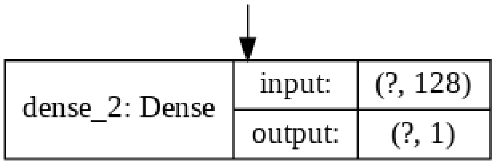
Out[ ]:





block1_conv1: Conv2D
input: (?, 256, 256, 3)
output: (?, 256, 256, 64)
block1_conv2: Conv2D
input: (?, 256, 256, 64)
output: (?, 256, 256, 64)
block1_pool: MaxPooling2D
input: (?, 256, 256, 64)
output: (?, 128, 128, 64)
block2_conv1: Conv2D
input: (?, 128, 128, 64)
output: (?, 128, 128, 128)
block2_conv2: Conv2D
input: (?, 128, 128, 128)
output: (?, 128, 128, 128)
block2_pool: MaxPooling2D
input: (?, 128, 128, 128)
output: (?, 64, 64, 128)
block3_conv1: Conv2D
input: (?, 64, 64, 128)
output: (?, 64, 64, 256)
block3_conv2: Conv2D
input: (?, 64, 64, 256)
output: (?, 64, 64, 256)
block3_conv3: Conv2D
input: (?, 64, 64, 256)
output: (?, 64, 64, 256)
block3_pool: MaxPooling2D
input: (?, 64, 64, 256)
output: (?, 32, 32, 256)
block4_conv1: Conv2D
input: (?, 32, 32, 256)
output: (?, 32, 32, 512)
block4_conv2: Conv2D
input: (?, 32, 32, 512)
output: (?, 32, 32, 512)


block4_conv3: Conv2D | input: (?, 32, 32, 512) | output: (?, 32, 32, 512)
block4_pool: MaxPooling2D | input: (?, 32, 32, 512) | output: (?, 16, 16, 512)
block5_conv1: Conv2D | input: (?, 16, 16, 512) | output: (?, 16, 16, 512)
block5_conv2: Conv2D | input: (?, 16, 16, 512) | output: (?, 16, 16, 512)
block5_conv3: Conv2D | input: (?, 16, 16, 512) | output: (?, 16, 16, 512)
block5_pool: MaxPooling2D | input: (?, 16, 16, 512) | output: (?, 8, 8, 512)
conv2d: Conv2D | input: (?, 8, 8, 512) | output: (?, 6, 6, 32)
activation: Activation | input: (?, 6, 6, 32) | output: (?, 6, 6, 32)
max_pooling2d: MaxPooling2D | input: (?, 6, 6, 32) | output: (?, 3, 3, 32)
flatten: Flatten | input: (?, 3, 3, 32) | output: (?, 288)
dense: Dense | input: (?, 288) | output: (?, 256)
dense_1: Dense | input: (?, 256) | output: (?, 128)



In [29]:

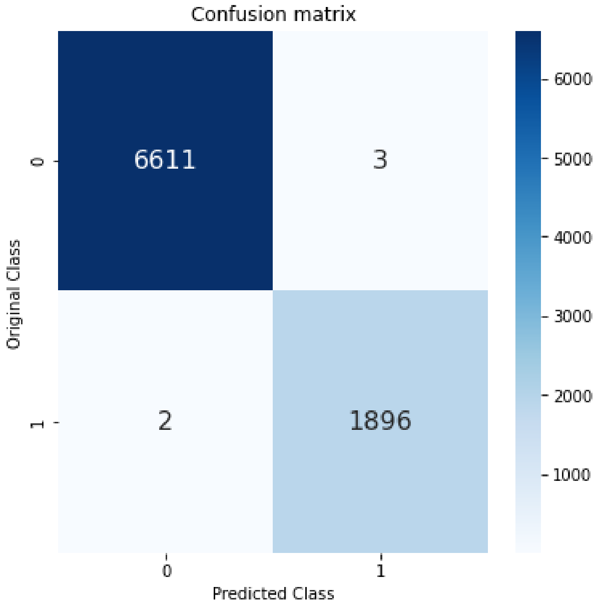
```
from sklearn.metrics import confusion_matrix
def confusion_mat(test_y,predict_y):
    ''' Function to Visualize the Confusion Matrix'''

    labels = [0,1]
    plt.figure(figsize=(6,6))
    cmap=sns.light_palette("blue")
    C = confusion_matrix(test_y, predict_y)
    print("Percentage of misclassified points ",(len(test_y)-np.trace(C))/len(test_y)*100)
    sns.heatmap(C, cmap="Blues",annot=True,annot_kws={"size": 16},fmt='g')
    plt.xlabel('Predicted Class')
    plt.ylabel('Original Class')
    plt.title('Confusion matrix')

    plt.show()
```

Checking confusion Matrix on train Data with different thresholds

In [ ]:

```
y_pred_1=[] #array to store predicted label
y_true=[] #array to store the ground truth
for i,j in tqdm(train_ds.take(8512)):
  y_pred_1.extend(model1.predict(i)) #predicting each batch
  y_true.extend(j)
y_pred=[]
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a
custom threshold (0.5 is the default threshold)
  if i[0]>=0.5: #setting threshold
    y_pred.append(1)
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
confusion_mat(y_true,y_pred)
```

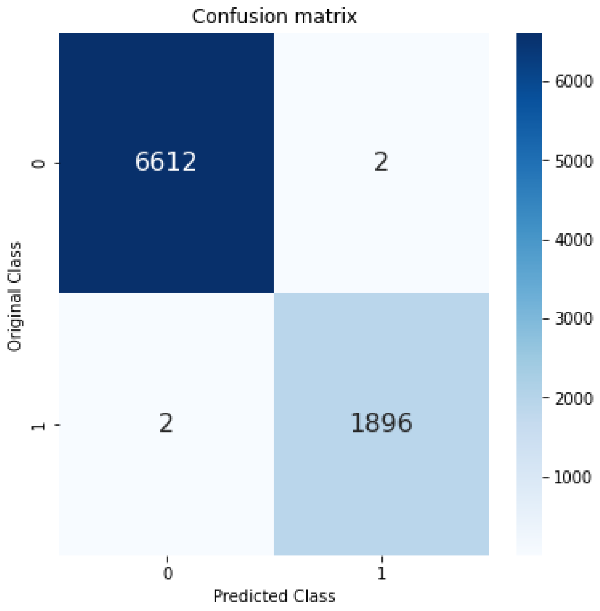
```
Percentage of misclassified points  0.05874060150375594
```



In [ ]:

```
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a custom threshold (0.5 is the default threshold)
  if i[0]>=0.6: #setting threshold
    y_pred.append(1)
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
confusion_mat(y_true,y_pred)
```

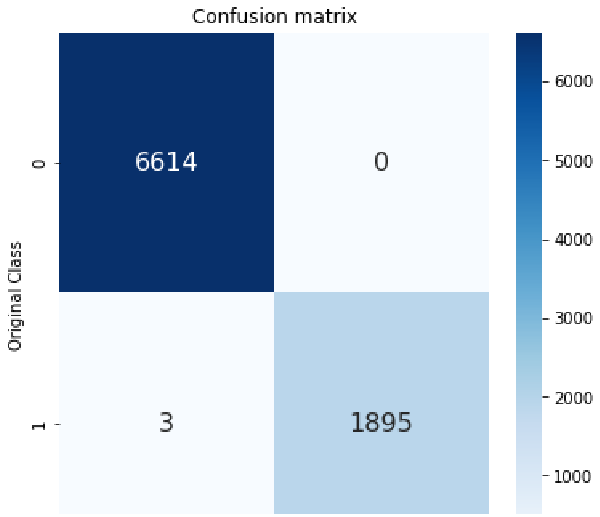
```
Percentage of misclassified points  0.046992481203007516
```

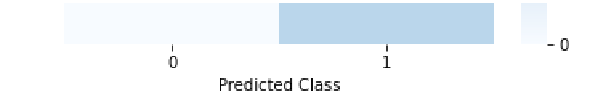


In [ ]:

```
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a custom threshold (0.5 is the default threshold)
  if i[0]>=0.7: #setting threshold
    y_pred.append(1)
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
confusion_mat(y_true,y_pred)
```

```
Percentage of misclassified points  0.035244360902255634
```

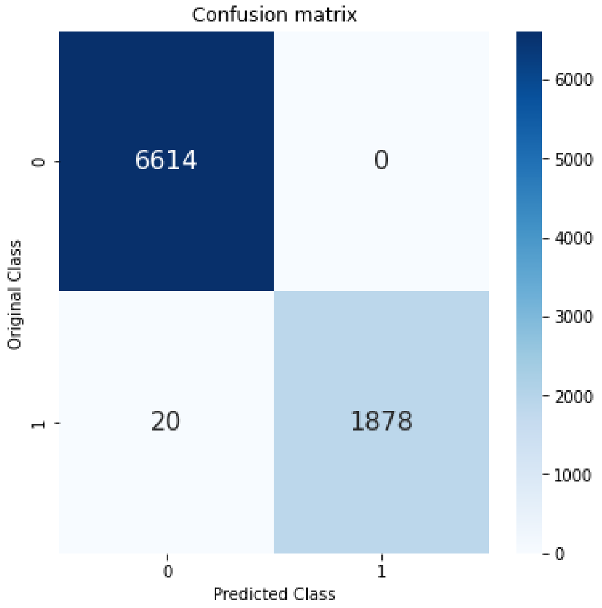




In [ ]:

```
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a custom threshold (0.5 is the default threshold)
  if i[0]>=0.8: #setting threshold
    y_pred.append(1)
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
confusion_mat(y_true,y_pred)
```

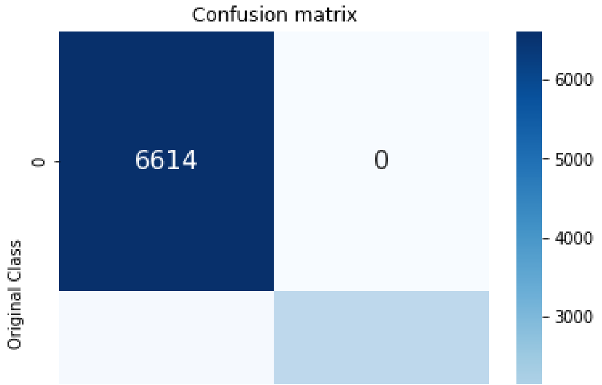
```
Percentage of misclassified points  0.2349624060150376
```

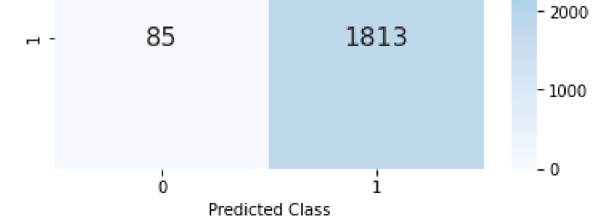


In [ ]:

```
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a custom threshold (0.5 is the default threshold)
  if i[0]>=0.8: #setting threshold
    y_pred.append(1)
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
confusion_mat(y_true,y_pred)
```

```
Percentage of misclassified points  0.9985902255639099
```



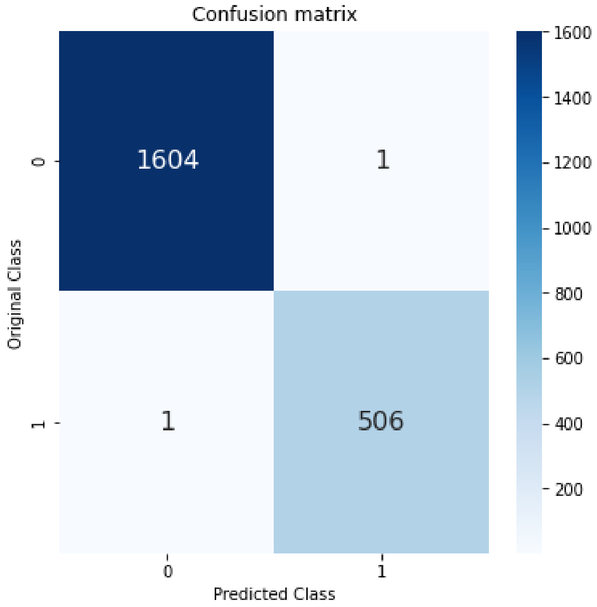


Checking confusion Matrix on test Data with different thresholds

In [ ]:

```
y_pred_1=[] #array to store predicted label
y_true=[] #array to store the ground truth
for i,j in tqdm(val_ds.take(2112)):
  y_pred_1.extend(model1.predict(i)) #predicting each batch
  y_true.extend(j)
y_pred=[]
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a custom threshold (0.5 is the default threshold)
  if i[0]>=0.5: #setting threshold
    y_pred.append(1)
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
confusion_mat(y_true,y_pred)
```

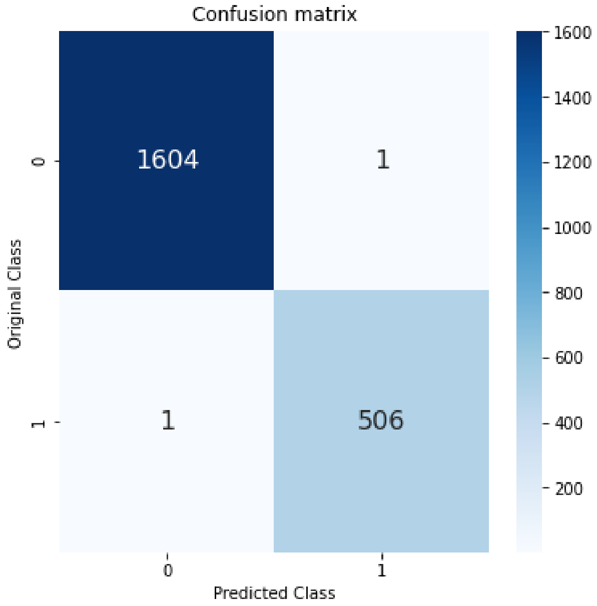
```
Percentage of misclassified points  0.0946969696969697
```



In [ ]:

```
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a custom threshold (0.5 is the default threshold)
  if i[0]>=0.6: #setting threshold
    y_pred.append(1)
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
confusion_mat(y_true,y_pred)
```

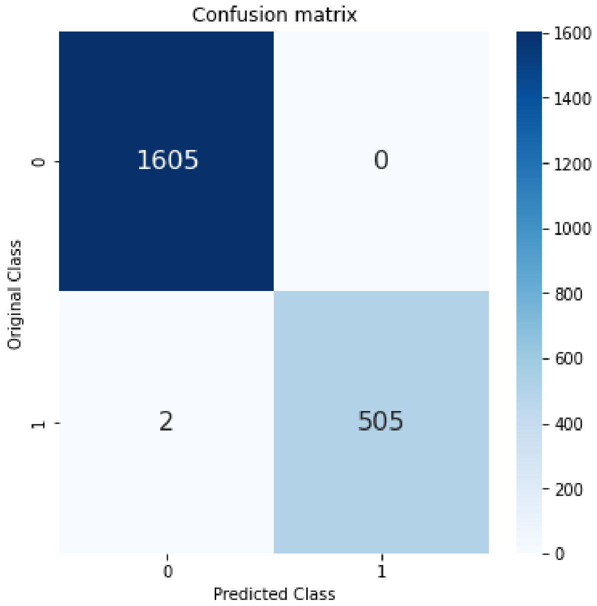
```
Percentage of misclassified points  0.0946969696969697
```



In [ ]:

```
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a custom threshold (0.5 is the default threshold)
  if i[0]>=0.7: #setting threshold
    y_pred.append(1)
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
confusion_mat(y_true,y_pred)
```

```
Percentage of misclassified points  0.0946969696969697
```

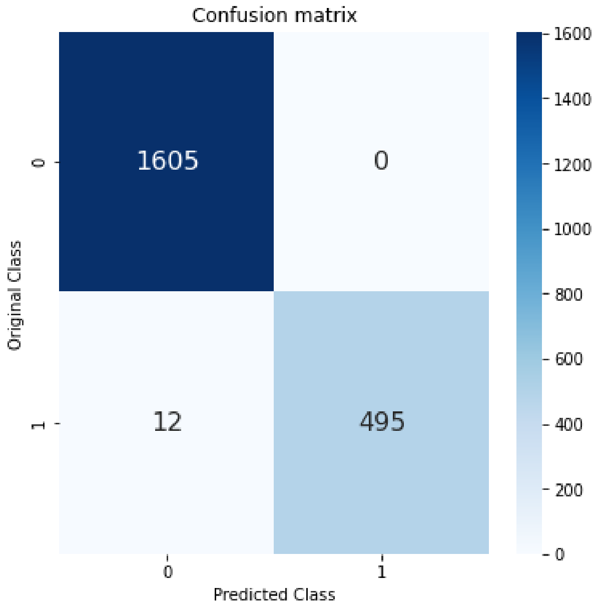


In [ ]:

```
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a custom threshold (0.5 is the default threshold)
  if i[0]>=0.8: #setting threshold
    y_pred.append(1)
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
```

```
confusion_mat(y_true,y_pred)
```

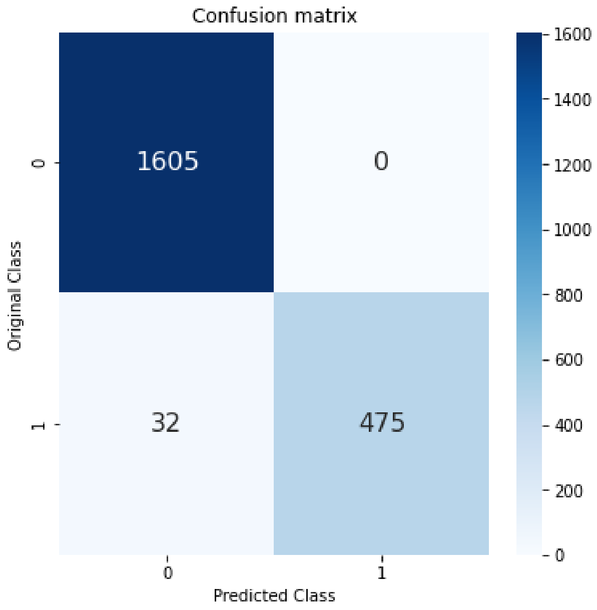
```
Percentage of misclassified points  0.5681818181818182
```



In [ ]:

```
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a custom threshold (0.5 is the default threshold)
  if i[0]>=0.9: #setting threshold
    y_pred.append(1)
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
confusion_mat(y_true,y_pred)
```

```
Percentage of misclassified points  1.5151515151515151
```



Observations:

1. 0.5 as the threshold is better than the other thresholds for reducing False negatives
2. As the threshold increases the number of False negatives increases

2. As the threshold increases the number of False negatives increases

Custom 5 fold cross validation approach :

Approach : We shall take the entire training data and divide it into 5 equal parts . We shall train 5 models, each model will be trained on the 4 parts and validated on the remaining 1 part. We shall then take the best model out of the 5 models under consideration

Note : Unlike the first model this might have a slight decrease in performance due to decrease in data for training . There are approximately 10100 images . We have already split the data into 80-20 for train and test. Our previous model was built by training the model with 80% of the data . But here we are training the model with only 60% of the total data and using 40% of the data for cross validation and test. We are training the model model 2000+ images less than the above model. If we get reasonable metrics closer to the first model then we can tell the model is working decently .

Splitting the training data into 5 equal parts

In [14]:

```
val_size = int(8512 * 0.2)    #20% of the training data
cv1=train_ds.take(val_size) #taking the 20% of training data into cv1
temp1=train_ds.skip(val_size) #temporarily storing the remaining data in temp1



cv2=temp1.take(val_size) #taking another 20% of the training data from temp1
temp2=temp1.skip(val_size) #temporarily storing the remaining data in temp2


cv3=temp2.take(val_size)#taking another 20% of the training data from temp2
temp3=temp2.skip(val_size)#temporarily storing the remaining data in temp3


cv4=temp3.take(val_size)#taking another 20% of the training data from temp3
temp4=temp3.skip(val_size)#temporarily storing the remaining data in temp4
cv5=temp4.take(val_size)#As there is only 20% of the data left we are storing it into cv5
```

In [15]:

```
train1=cv1.concatenate(cv2)   #concatenating cv1,cv2,cv3 and cv4 for training the model
train1=train1.concatenate(cv3)
train1=train1.concatenate(cv4)


train2=cv2.concatenate(cv3)      #concatenating cv5,cv2,cv3 and cv4 for training the model
train2=train2.concatenate(cv4)
train2=train2.concatenate(cv5)


train3=cv3.concatenate(cv4)        #concatenating cv1,cv5,cv3 and cv4 for training the model
train3=train3.concatenate(cv1)
train3=train3.concatenate(cv5)

train4=cv4.concatenate(cv5)        #concatenating cv1,cv2,cv5 and cv4 for training the model
train4=train4.concatenate(cv1)
train4=train4.concatenate(cv2)


train5=cv5.concatenate(cv1)        #concatenating cv1,cv2,cv3 and cv5 for training the model
train5=train5.concatenate(cv2)
train5=train5.concatenate(cv3)
```

In [16]:

```
#preparing the training data into batches for training
train1 = train1.batch(64, drop_remainder=True)
train2 = train2.batch(64, drop_remainder=True)
train3 = train3.batch(64, drop_remainder=True)
train4 = train4.batch(64, drop_remainder=True)
train5 = train5.batch(64, drop_remainder=True)
```

In [18]:

```
#preparing the cross validation data into batches for validation dataset
cv1 = cv1.batch(64, drop_remainder=True)
cv2 = cv2.batch(64, drop_remainder=True)
cv3 = cv3.batch(64, drop_remainder=True)
cv4 = cv4.batch(64, drop_remainder=True)
cv5 = cv5.batch(64, drop_remainder=True)
```

cross validation-1

In [ ]:

```
tf.keras.backend.clear_session()
checkpoint_path = "training_1/cp.ckpt"
checkpoint_dir = os.path.dirname(checkpoint_path)
# Tensorbaord
! rm -rf ./logs/
logdir = os.path.join("logs", datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
%tensorboard --logdir $logdir
tensorboard_callback = tf.keras.callbacks.TensorBoard(logdir, histogram_freq=1)

model_1 =tf.keras.applications.vgg16.VGG16(weights = "imagenet", include_top=False, input_shape = (
256,256,3))
for i in model_1.layers:
  i.trainable=False
model=model_1.output
model=Conv2D(32, (3, 3))(model)
model=(Activation('relu'))(model)
model=(MaxPool2D(pool_size=(2, 2)))(model)
model=Flatten()(model)
model = Dense(256, activation="relu")(model)
model = Dense(128, activation="relu")(model)
output_layer = Dense(1, activation="sigmoid")(model)
model1 = Model(model_1.input,output_layer)
model1.compile(loss = "binary_crossentropy", optimizer =tf.keras.optimizers.Adam(lr=0.0001), metric
s=["accuracy",
tf.keras.metrics.Precision(name='precision'),tf.keras.metrics.Recall(name='recall')])
model1.fit(train1,epochs=30,verbose=True,validation_data=cv5,batch_size=64,callbacks=[checkpoint,te
nsorboard_callback])
```

```
Epoch 1/30
  1/106 [..............................] - ETA: 0s - loss: 0.6660 - accuracy: 0.6406 - precision:
0.2000 - recall: 0.0500WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/summary_ops_v2.py:1277: stop (from
tensorflow.python.eager.profiler) is deprecated and will be removed after 2020-07-01.
Instructions for updating:
use `tf.profiler.experimental.stop` instead.
106/106 [==============================] - ETA: 0s - loss: 0.5101 - accuracy: 0.7767 - precision:
0.2000 - recall: 6.6138e-04
Epoch 00001: val_recall improved from -inf to 0.00000, saving model to model_save/weights-01-0.000
0.hdf5
106/106 [==============================] - 809s 8s/step - loss: 0.5101 - accuracy: 0.7767 - precis
ion: 0.2000 - recall: 6.6138e-04 - val_loss: 0.4760 - val_accuracy: 0.7843 - val_precision: 0.0000
e+00 - val_recall: 0.0000e+00
Epoch 2/30
106/106 [==============================] - ETA: 0s - loss: 0.4397 - accuracy: 0.7945 - precision:
0.6579 - recall: 0.1184
Epoch 00002: val_recall improved from 0.00000 to 0.18407, saving model to model_save/weights-02-0.
1841.hdf5
106/106 [==============================] - 815s 8s/step - loss: 0.4397 - accuracy: 0.7945 - precis
ion: 0.6579 - recall: 0.1184 - val_loss: 0.4269 - val_accuracy: 0.7945 - val_precision: 0.5982 - v
al_recall: 0.1841
Epoch 3/30
106/106 [==============================] - ETA: 0s - loss: 0.4189 - accuracy: 0.8062 - precision:
0.6431 - recall: 0.2732
Epoch 00003: val_recall improved from 0.18407 to 0.24619, saving model to model_save/weights-03-0.
2462.hdf5
106/106 [==============================] - 809s 8s/step - loss: 0.4189 - accuracy: 0.8062 - precis
ion: 0.6431 - recall: 0.2732 - val_loss: 0.4128 - val_accuracy: 0.7981 - val_precision: 0.7132 - v
al_recall: 0.2462
Epoch 4/30
106/106 [==============================] - ETA: 0s - loss: 0.3910 - accuracy: 0.8147 - precision:
0.6592 - recall: 0.3492
```

```
0.6592 - recall: 0.3492
Epoch 00004: val_recall improved from 0.24619 to 0.54286, saving model to model_save/weights-04-0.
5429.hdf5
106/106 [==============================] - 808s 8s/step - loss: 0.3910 - accuracy: 0.8147 - precis
ion: 0.6592 - recall: 0.3492 - val_loss: 0.3982 - val_accuracy: 0.8263 - val_precision: 0.6491 - v
al_recall: 0.5429
Epoch 5/30
106/106 [==============================] - ETA: 0s - loss: 0.3792 - accuracy: 0.8284 - precision:
0.6990 - recall: 0.4475
Epoch 00005: val_recall improved from 0.54286 to 0.57821, saving model to model_save/weights-05-0.
5782.hdf5
106/106 [==============================] - 816s 8s/step - loss: 0.3792 - accuracy: 0.8284 - precis
ion: 0.6990 - recall: 0.4475 - val_loss: 0.3828 - val_accuracy: 0.8365 - val_precision: 0.6311 - v
al_recall: 0.5782
Epoch 6/30
106/106 [==============================] - ETA: 0s - loss: 0.3822 - accuracy: 0.8240 - precision:
0.6701 - recall: 0.4317
Epoch 00006: val_recall did not improve from 0.57821
106/106 [==============================] - 811s 8s/step - loss: 0.3822 - accuracy: 0.8240 - precis
ion: 0.6701 - recall: 0.4317 - val_loss: 0.3620 - val_accuracy: 0.8311 - val_precision: 0.7724 - v
al_recall: 0.2730
Epoch 7/30
106/106 [==============================] - ETA: 0s - loss: 0.3551 - accuracy: 0.8436 - precision:
0.7206 - recall: 0.4694
Epoch 00007: val_recall did not improve from 0.57821
106/106 [==============================] - 815s 8s/step - loss: 0.3551 - accuracy: 0.8436 - precis
ion: 0.7206 - recall: 0.4694 - val_loss: 0.3477 - val_accuracy: 0.8377 - val_precision: 0.7126 - v
al_recall: 0.4693
Epoch 8/30
106/106 [==============================] - ETA: 0s - loss: 0.3398 - accuracy: 0.8476 - precision:
0.7283 - recall: 0.5344
Epoch 00008: val_recall did not improve from 0.57821
106/106 [==============================] - 809s 8s/step - loss: 0.3398 - accuracy: 0.8476 - precis
ion: 0.7283 - recall: 0.5344 - val_loss: 0.3293 - val_accuracy: 0.8594 - val_precision: 0.7663 - v
al_recall: 0.5362
Epoch 9/30
106/106 [==============================] - ETA: 0s - loss: 0.3230 - accuracy: 0.8616 - precision:
0.7520 - recall: 0.5704
Epoch 00009: val_recall did not improve from 0.57821
106/106 [==============================] - 812s 8s/step - loss: 0.3230 - accuracy: 0.8616 - precis
ion: 0.7520 - recall: 0.5704 - val_loss: 0.3496 - val_accuracy: 0.8389 - val_precision: 0.8397 - v
al_recall: 0.3503
Epoch 10/30
106/106 [==============================] - ETA: 0s - loss: 0.3208 - accuracy: 0.8619 - precision:
0.7612 - recall: 0.5797
Epoch 00010: val_recall did not improve from 0.57821
106/106 [==============================] - 817s 8s/step - loss: 0.3208 - accuracy: 0.8619 - precis
ion: 0.7612 - recall: 0.5797 - val_loss: 0.3016 - val_accuracy: 0.8648 - val_precision: 0.8333 - v
al_recall: 0.5486
Epoch 11/30
106/106 [==============================] - ETA: 0s - loss: 0.2993 - accuracy: 0.8697 - precision:
0.7752 - recall: 0.5883
Epoch 00011: val_recall improved from 0.57821 to 0.70000, saving model to model_save/weights-11-0.
7000.hdf5
106/106 [==============================] - 813s 8s/step - loss: 0.2993 - accuracy: 0.8697 - precis
ion: 0.7752 - recall: 0.5883 - val_loss: 0.3036 - val_accuracy: 0.8768 - val_precision: 0.7337 - v
al_recall: 0.7000
Epoch 12/30
106/106 [==============================] - ETA: 0s - loss: 0.2922 - accuracy: 0.8757 - precision:
0.7737 - recall: 0.6132
Epoch 00012: val_recall did not improve from 0.70000
106/106 [==============================] - 814s 8s/step - loss: 0.2922 - accuracy: 0.8757 - precis
ion: 0.7737 - recall: 0.6132 - val_loss: 0.2942 - val_accuracy: 0.8558 - val_precision: 0.8611 - v
al_recall: 0.4189
Epoch 13/30
106/106 [==============================] - ETA: 0s - loss: 0.2686 - accuracy: 0.8930 - precision:
0.8229 - recall: 0.6816
Epoch 00013: val_recall improved from 0.70000 to 0.73846, saving model to model_save/weights-13-0.
7385.hdf5
106/106 [==============================] - 815s 8s/step - loss: 0.2686 - accuracy: 0.8930 - precis
ion: 0.8229 - recall: 0.6816 - val_loss: 0.2680 - val_accuracy: 0.8906 - val_precision: 0.7826 - v
al_recall: 0.7385
Epoch 14/30
106/106 [==============================] - ETA: 0s - loss: 0.2687 - accuracy: 0.8861 - precision:
0.8040 - recall: 0.6516
Epoch 00014: val_recall did not improve from 0.73846
106/106 [==============================] - 819s 8s/step - loss: 0.2687 - accuracy: 0.8861 - precis
ion: 0.8040 - recall: 0.6516 - val_loss: 0.2459 - val_accuracy: 0.9038 - val_precision: 0.8897 - v
```

```
ion: 0.8040 - recall: 0.6516 - val_loss: 0.2459 - val_accuracy: 0.9038 - val_precision: 0.8897 - v
al_recall: 0.6684
Epoch 15/30
106/106 [==============================] - ETA: 0s - loss: 0.2493 - accuracy: 0.9006 - precision:
0.8316 - recall: 0.7122
Epoch 00015: val_recall did not improve from 0.73846
106/106 [==============================] - 816s 8s/step - loss: 0.2493 - accuracy: 0.9006 - precis
ion: 0.8316 - recall: 0.7122 - val_loss: 0.2624 - val_accuracy: 0.8876 - val_precision: 0.9182 - v
al_recall: 0.5995
Epoch 16/30
106/106 [==============================] - ETA: 0s - loss: 0.2455 - accuracy: 0.9018 - precision:
0.8290 - recall: 0.7135
Epoch 00016: val_recall did not improve from 0.73846
106/106 [==============================] - 815s 8s/step - loss: 0.2455 - accuracy: 0.9018 - precis
ion: 0.8290 - recall: 0.7135 - val_loss: 0.2204 - val_accuracy: 0.9135 - val_precision: 0.8673 - v
al_recall: 0.7083
Epoch 17/30
106/106 [==============================] - ETA: 0s - loss: 0.2141 - accuracy: 0.9173 - precision:
0.8611 - recall: 0.7392
Epoch 00017: val_recall improved from 0.73846 to 0.76039, saving model to model_save/weights-17-0.
7604.hdf5
106/106 [==============================] - 809s 8s/step - loss: 0.2141 - accuracy: 0.9173 - precis
ion: 0.8611 - recall: 0.7392 - val_loss: 0.2189 - val_accuracy: 0.9213 - val_precision: 0.9041 - v
al_recall: 0.7604
Epoch 18/30
106/106 [==============================] - ETA: 0s - loss: 0.2049 - accuracy: 0.9226 - precision:
0.8670 - recall: 0.7653
Epoch 00018: val_recall improved from 0.76039 to 0.80124, saving model to model_save/weights-18-0.
8012.hdf5
106/106 [==============================] - 817s 8s/step - loss: 0.2049 - accuracy: 0.9226 - precis
ion: 0.8670 - recall: 0.7653 - val_loss: 0.1722 - val_accuracy: 0.9417 - val_precision: 0.8866 - v
al_recall: 0.8012
Epoch 19/30
106/106 [==============================] - ETA: 0s - loss: 0.1934 - accuracy: 0.9310 - precision:
0.8876 - recall: 0.7916
Epoch 00019: val_recall improved from 0.80124 to 0.80267, saving model to model_save/weights-19-0.
8027.hdf5
106/106 [==============================] - 815s 8s/step - loss: 0.1934 - accuracy: 0.9310 - precis
ion: 0.8876 - recall: 0.7916 - val_loss: 0.1887 - val_accuracy: 0.9351 - val_precision: 0.8985 - v
al_recall: 0.8027
Epoch 20/30
106/106 [==============================] - ETA: 0s - loss: 0.1745 - accuracy: 0.9365 - precision:
0.9078 - recall: 0.7917
Epoch 00020: val_recall improved from 0.80267 to 0.88971, saving model to model_save/weights-20-0.
8897.hdf5
106/106 [==============================] - 814s 8s/step - loss: 0.1745 - accuracy: 0.9365 - precis
ion: 0.9078 - recall: 0.7917 - val_loss: 0.1579 - val_accuracy: 0.9543 - val_precision: 0.9213 - v
al_recall: 0.8897
Epoch 21/30
106/106 [==============================] - ETA: 0s - loss: 0.1699 - accuracy: 0.9360 - precision:
0.8967 - recall: 0.8036
Epoch 00021: val_recall did not improve from 0.88971
106/106 [==============================] - 811s 8s/step - loss: 0.1699 - accuracy: 0.9360 - precis
ion: 0.8967 - recall: 0.8036 - val_loss: 0.1677 - val_accuracy: 0.9453 - val_precision: 0.8867 - v
al_recall: 0.8652
Epoch 22/30
106/106 [==============================] - ETA: 0s - loss: 0.1523 - accuracy: 0.9460 - precision:
0.9135 - recall: 0.8441
Epoch 00022: val_recall did not improve from 0.88971
106/106 [==============================] - 813s 8s/step - loss: 0.1523 - accuracy: 0.9460 - precis
ion: 0.9135 - recall: 0.8441 - val_loss: 0.1398 - val_accuracy: 0.9471 - val_precision: 0.9690 - v
al_recall: 0.7806
Epoch 23/30
106/106 [==============================] - ETA: 0s - loss: 0.1349 - accuracy: 0.9530 - precision:
0.9261 - recall: 0.8567
Epoch 00023: val_recall improved from 0.88971 to 0.93939, saving model to model_save/weights-23-0.
9394.hdf5
106/106 [==============================] - 816s 8s/step - loss: 0.1349 - accuracy: 0.9530 - precis
ion: 0.9261 - recall: 0.8567 - val_loss: 0.1281 - val_accuracy: 0.9639 - val_precision: 0.8997 - v
al_recall: 0.9394
Epoch 24/30
106/106 [==============================] - ETA: 0s - loss: 0.1236 - accuracy: 0.9584 - precision:
0.9340 - recall: 0.8727
Epoch 00024: val_recall did not improve from 0.93939
106/106 [==============================] - 812s 8s/step - loss: 0.1236 - accuracy: 0.9584 - precis
ion: 0.9340 - recall: 0.8727 - val_loss: 0.1132 - val_accuracy: 0.9573 - val_precision: 0.9658 - v
al_recall: 0.8222
Epoch 25/30
Epoch 25/30
106/106 [==============================] - ETA: 0s - loss: 0.1125 - accuracy: 0.9646 - precision:
0.9487 - recall: 0.8879
Epoch 00025: val_recall did not improve from 0.93939
106/106 [==============================] - 814s 8s/step - loss: 0.1125 - accuracy: 0.9646 - precis
ion: 0.9487 - recall: 0.8879 - val_loss: 0.1155 - val_accuracy: 0.9645 - val_precision: 0.9188 - v
al_recall: 0.9306
Epoch 26/30
106/106 [==============================] - ETA: 0s - loss: 0.1077 - accuracy: 0.9671 - precision:
0.9493 - recall: 0.9011
Epoch 00026: val_recall did not improve from 0.93939
106/106 [==============================] - 818s 8s/step - loss: 0.1077 - accuracy: 0.9671 - precis
ion: 0.9493 - recall: 0.9011 - val_loss: 0.0919 - val_accuracy: 0.9730 - val_precision: 0.9891 - v
al_recall: 0.8988
Epoch 27/30
106/106 [==============================] - ETA: 0s - loss: 0.0954 - accuracy: 0.9704 - precision:
0.9541 - recall: 0.9085
Epoch 00027: val_recall did not improve from 0.93939
106/106 [==============================] - 816s 8s/step - loss: 0.0954 - accuracy: 0.9704 - precis
ion: 0.9541 - recall: 0.9085 - val_loss: 0.0960 - val_accuracy: 0.9663 - val_precision: 0.9907 - v
al_recall: 0.8579
Epoch 28/30
106/106 [==============================] - ETA: 0s - loss: 0.0869 - accuracy: 0.9760 - precision:
0.9681 - recall: 0.9226
Epoch 00028: val_recall did not improve from 0.93939
106/106 [==============================] - 815s 8s/step - loss: 0.0869 - accuracy: 0.9760 - precis
ion: 0.9681 - recall: 0.9226 - val_loss: 0.0766 - val_accuracy: 0.9778 - val_precision: 0.9855 - v
al_recall: 0.9137
Epoch 29/30
106/106 [==============================] - ETA: 0s - loss: 0.0759 - accuracy: 0.9816 - precision:
0.9735 - recall: 0.9433
Epoch 00029: val_recall improved from 0.93939 to 0.94986, saving model to model_save/weights-29-0.
9499.hdf5
106/106 [==============================] - 812s 8s/step - loss: 0.0759 - accuracy: 0.9816 - precis
ion: 0.9735 - recall: 0.9433 - val_loss: 0.0634 - val_accuracy: 0.9850 - val_precision: 0.9799 - v
al_recall: 0.9499
Epoch 30/30
106/106 [==============================] - ETA: 0s - loss: 0.0712 - accuracy: 0.9801 - precision:
0.9736 - recall: 0.9375
Epoch 00030: val_recall improved from 0.94986 to 0.97035, saving model to model_save/weights-30-0.
9704.hdf5
106/106 [==============================] - 815s 8s/step - loss: 0.0712 - accuracy: 0.9801 - precis
ion: 0.9736 - recall: 0.9375 - val_loss: 0.0647 - val_accuracy: 0.9850 - val_precision: 0.9626 - v
al_recall: 0.9704
```

Out[ ]:

```
<tensorflow.python.keras.callbacks.History at 0x7f5d136a3278>
```

cross validation-2

In [ ]:

```
tf.keras.backend.clear_session()
checkpoint_path = "training_1/cp.ckpt"
checkpoint_dir = os.path.dirname(checkpoint_path)
# Tensorbaord
! rm -rf ./logs/
logdir = os.path.join("logs", datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
%tensorboard --logdir $logdir
tensorboard_callback = tf.keras.callbacks.TensorBoard(logdir, histogram_freq=1)

model_1 =tf.keras.applications.vgg16.VGG16(weights = "imagenet", include_top=False, input_shape = (
256,256,3))
for i in model_1.layers:
  i.trainable=False
model=model_1.output
model=Conv2D(32, (3, 3))(model)
model=(Activation('relu'))(model)
model=(MaxPool2D(pool_size=(2, 2)))(model)
model=Flatten()(model)
model = Dense(256, activation="relu")(model)
model = Dense(128, activation="relu")(model)
output_layer = Dense(1, activation="sigmoid")(model)
model1 = Model(model_1.input,output_layer)
```

```
model1 = Model(model_1.input,output_layer)
model1.compile(loss = "binary_crossentropy", optimizer =tf.keras.optimizers.Adam(lr=0.0001), metric
s=["accuracy",
tf.keras.metrics.Precision(name='precision'),tf.keras.metrics.Recall(name='recall')])
model1.fit(train2,epochs=30,verbose=True,validation_data=cv1,batch_size=64,callbacks=[checkpoint,te
nsorboard_callback])
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-
applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58892288/58889256 [==============================] - 1s 0us/step
Epoch 1/30
  1/106 [..............................] - ETA: 0s - loss: 0.5431 - accuracy: 0.7656 - precision:
0.0000e+00 - recall: 0.0000e+00WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/summary_ops_v2.py:1277: stop (from
tensorflow.python.eager.profiler) is deprecated and will be removed after 2020-07-01.
Instructions for updating:
use `tf.profiler.experimental.stop` instead.
106/106 [==============================] - ETA: 0s - loss: 0.4929 - accuracy: 0.7839 - precision:
0.5000 - recall: 6.8213e-04
Epoch 00001: val_recall improved from -inf to 0.08282, saving model to model_save/weights-01-0.082
8.hdf5
106/106 [==============================] - 697s 7s/step - loss: 0.4929 - accuracy: 0.7839 - precis
ion: 0.5000 - recall: 6.8213e-04 - val_loss: 0.4327 - val_accuracy: 0.8161 - val_precision: 0.7941
- val_recall: 0.0828
Epoch 2/30
106/106 [==============================] - ETA: 0s - loss: 0.4422 - accuracy: 0.7911 - precision:
0.6475 - recall: 0.1687
Epoch 00002: val_recall improved from 0.08282 to 0.34694, saving model to model_save/weights-02-0.
3469.hdf5
106/106 [==============================] - 699s 7s/step - loss: 0.4422 - accuracy: 0.7911 - precis
ion: 0.6475 - recall: 0.1687 - val_loss: 0.4125 - val_accuracy: 0.8131 - val_precision: 0.5777 - v
al_recall: 0.3469
Epoch 3/30
106/106 [==============================] - ETA: 0s - loss: 0.4201 - accuracy: 0.8037 - precision:
0.6504 - recall: 0.3242
Epoch 00003: val_recall improved from 0.34694 to 0.40227, saving model to model_save/weights-03-0.
4023.hdf5
106/106 [==============================] - 662s 6s/step - loss: 0.4201 - accuracy: 0.8037 - precis
ion: 0.6504 - recall: 0.3242 - val_loss: 0.3749 - val_accuracy: 0.8377 - val_precision: 0.7065 - v
al_recall: 0.4023
Epoch 4/30
106/106 [==============================] - ETA: 0s - loss: 0.3960 - accuracy: 0.8157 - precision:
0.6706 - recall: 0.3362
Epoch 00004: val_recall improved from 0.40227 to 0.41885, saving model to model_save/weights-04-0.
4188.hdf5
106/106 [==============================] - 670s 6s/step - loss: 0.3960 - accuracy: 0.8157 - precis
ion: 0.6706 - recall: 0.3362 - val_loss: 0.3628 - val_accuracy: 0.8383 - val_precision: 0.7729 - v
al_recall: 0.4188
Epoch 5/30
106/106 [==============================] - ETA: 0s - loss: 0.3748 - accuracy: 0.8268 - precision:
0.6893 - recall: 0.4298
Epoch 00005: val_recall improved from 0.41885 to 0.57632, saving model to model_save/weights-05-0.
5763.hdf5
106/106 [==============================] - 673s 6s/step - loss: 0.3748 - accuracy: 0.8268 - precis
ion: 0.6893 - recall: 0.4298 - val_loss: 0.3497 - val_accuracy: 0.8534 - val_precision: 0.7252 - v
al_recall: 0.5763
Epoch 6/30
106/106 [==============================] - ETA: 0s - loss: 0.3566 - accuracy: 0.8377 - precision:
0.7093 - recall: 0.4671
Epoch 00006: val_recall did not improve from 0.57632
106/106 [==============================] - 699s 7s/step - loss: 0.3566 - accuracy: 0.8377 - precis
ion: 0.7093 - recall: 0.4671 - val_loss: 0.3467 - val_accuracy: 0.8468 - val_precision: 0.8008 - v
al_recall: 0.4761
Epoch 7/30
106/106 [==============================] - ETA: 0s - loss: 0.3377 - accuracy: 0.8494 - precision:
0.7314 - recall: 0.5129
Epoch 00007: val_recall did not improve from 0.57632
106/106 [==============================] - 697s 7s/step - loss: 0.3377 - accuracy: 0.8494 - precis
ion: 0.7314 - recall: 0.5129 - val_loss: 0.3403 - val_accuracy: 0.8540 - val_precision: 0.8073 - v
al_recall: 0.4668
Epoch 8/30
106/106 [==============================] - ETA: 0s - loss: 0.3406 - accuracy: 0.8486 - precision:
0.7355 - recall: 0.5189
Epoch 00008: val_recall improved from 0.57632 to 0.68022, saving model to model_save/weights-08-0.
6802.hdf5
106/106 [==============================] - 674s 6s/step - loss: 0.3406 - accuracy: 0.8486 - precis
ion: 0.7355 - recall: 0.5189 - val_loss: 0.3291 - val_accuracy: 0.8696 - val_precision: 0.7171 - v
```

```
ion: 0.7355 - recall: 0.5189 - val_loss: 0.3291 - val_accuracy: 0.8696 - val_precision: 0.7171 - v
al_recall: 0.6802
Epoch 9/30
106/106 [==============================] - ETA: 0s - loss: 0.3180 - accuracy: 0.8650 - precision:
0.7771 - recall: 0.5545
Epoch 00009: val_recall did not improve from 0.68022
106/106 [==============================] - 704s 7s/step - loss: 0.3180 - accuracy: 0.8650 - precis
ion: 0.7771 - recall: 0.5545 - val_loss: 0.3048 - val_accuracy: 0.8684 - val_precision: 0.8353 - v
al_recall: 0.5462
Epoch 10/30
106/106 [==============================] - ETA: 0s - loss: 0.3124 - accuracy: 0.8659 - precision:
0.7648 - recall: 0.6070
Epoch 00010: val_recall did not improve from 0.68022
106/106 [==============================] - 727s 7s/step - loss: 0.3124 - accuracy: 0.8659 - precis
ion: 0.7648 - recall: 0.6070 - val_loss: 0.2874 - val_accuracy: 0.8834 - val_precision: 0.7847 - v
al_recall: 0.6313
Epoch 11/30
106/106 [==============================] - ETA: 0s - loss: 0.2949 - accuracy: 0.8721 - precision:
0.7801 - recall: 0.5797
Epoch 00011: val_recall did not improve from 0.68022
106/106 [==============================] - 728s 7s/step - loss: 0.2949 - accuracy: 0.8721 - precis
ion: 0.7801 - recall: 0.5797 - val_loss: 0.2760 - val_accuracy: 0.8852 - val_precision: 0.8669 - v
al_recall: 0.5938
Epoch 12/30
106/106 [==============================] - ETA: 0s - loss: 0.2718 - accuracy: 0.8943 - precision:
0.8290 - recall: 0.6458
Epoch 00012: val_recall did not improve from 0.68022
106/106 [==============================] - 720s 7s/step - loss: 0.2718 - accuracy: 0.8943 - precis
ion: 0.8290 - recall: 0.6458 - val_loss: 0.2736 - val_accuracy: 0.8774 - val_precision: 0.8929 - v
al_recall: 0.5263
Epoch 13/30
106/106 [==============================] - ETA: 0s - loss: 0.2592 - accuracy: 0.8952 - precision:
0.8295 - recall: 0.6710
Epoch 00013: val_recall improved from 0.68022 to 0.82143, saving model to model_save/weights-13-0.
8214.hdf5
106/106 [==============================] - 726s 7s/step - loss: 0.2592 - accuracy: 0.8952 - precis
ion: 0.8295 - recall: 0.6710 - val_loss: 0.2725 - val_accuracy: 0.8948 - val_precision: 0.7311 - v
al_recall: 0.8214
Epoch 14/30
106/106 [==============================] - ETA: 0s - loss: 0.2472 - accuracy: 0.9012 - precision:
0.8312 - recall: 0.6995
Epoch 00014: val_recall did not improve from 0.82143
106/106 [==============================] - 729s 7s/step - loss: 0.2472 - accuracy: 0.9012 - precis
ion: 0.8312 - recall: 0.6995 - val_loss: 0.2525 - val_accuracy: 0.8936 - val_precision: 0.8812 - v
al_recall: 0.6544
Epoch 15/30
106/106 [==============================] - ETA: 0s - loss: 0.2256 - accuracy: 0.9138 - precision:
0.8595 - recall: 0.7248
Epoch 00015: val_recall did not improve from 0.82143
106/106 [==============================] - 723s 7s/step - loss: 0.2256 - accuracy: 0.9138 - precis
ion: 0.8595 - recall: 0.7248 - val_loss: 0.2386 - val_accuracy: 0.8972 - val_precision: 0.8907 - v
al_recall: 0.6044
Epoch 16/30
106/106 [==============================] - ETA: 0s - loss: 0.2240 - accuracy: 0.9132 - precision:
0.8547 - recall: 0.7370
Epoch 00016: val_recall did not improve from 0.82143
106/106 [==============================] - 728s 7s/step - loss: 0.2240 - accuracy: 0.9132 - precis
ion: 0.8547 - recall: 0.7370 - val_loss: 0.2309 - val_accuracy: 0.9093 - val_precision: 0.8257 - v
al_recall: 0.7625
Epoch 17/30
106/106 [==============================] - ETA: 0s - loss: 0.2010 - accuracy: 0.9228 - precision:
0.8734 - recall: 0.7618
Epoch 00017: val_recall improved from 0.82143 to 0.83287, saving model to model_save/weights-17-0.
8329.hdf5
106/106 [==============================] - 725s 7s/step - loss: 0.2010 - accuracy: 0.9228 - precis
ion: 0.8734 - recall: 0.7618 - val_loss: 0.1748 - val_accuracy: 0.9375 - val_precision: 0.8717 - v
al_recall: 0.8329
Epoch 18/30
106/106 [==============================] - ETA: 0s - loss: 0.2004 - accuracy: 0.9225 - precision:
0.8718 - recall: 0.7691
Epoch 00018: val_recall did not improve from 0.83287
106/106 [==============================] - 716s 7s/step - loss: 0.2004 - accuracy: 0.9225 - precis
ion: 0.8718 - recall: 0.7691 - val_loss: 0.1838 - val_accuracy: 0.9255 - val_precision: 0.9604 - v
al_recall: 0.7026
Epoch 19/30
106/106 [==============================] - ETA: 0s - loss: 0.1768 - accuracy: 0.9360 - precision:
0.8909 - recall: 0.7885
Epoch 00019: val_recall did not improve from 0.83287
Epoch 00019: val_recall did not improve from 0.83287
106/106 [==============================] - 698s 7s/step - loss: 0.1768 - accuracy: 0.9360 - precis
ion: 0.8909 - recall: 0.7885 - val_loss: 0.2137 - val_accuracy: 0.9153 - val_precision: 0.9763 - v
al_recall: 0.6466
Epoch 20/30
106/106 [==============================] - ETA: 0s - loss: 0.1701 - accuracy: 0.9394 - precision:
0.9064 - recall: 0.8191
Epoch 00020: val_recall improved from 0.83287 to 0.83632, saving model to model_save/weights-20-0.
8363.hdf5
106/106 [==============================] - 704s 7s/step - loss: 0.1701 - accuracy: 0.9394 - precis
ion: 0.9064 - recall: 0.8191 - val_loss: 0.1471 - val_accuracy: 0.9507 - val_precision: 0.9478 - v
al_recall: 0.8363
Epoch 21/30
106/106 [==============================] - ETA: 0s - loss: 0.1537 - accuracy: 0.9474 - precision:
0.9181 - recall: 0.8419
Epoch 00021: val_recall improved from 0.83632 to 0.87395, saving model to model_save/weights-21-0.
8739.hdf5
106/106 [==============================] - 688s 6s/step - loss: 0.1537 - accuracy: 0.9474 - precis
ion: 0.9181 - recall: 0.8419 - val_loss: 0.1172 - val_accuracy: 0.9657 - val_precision: 0.9630 - v
al_recall: 0.8739
Epoch 22/30
106/106 [==============================] - ETA: 0s - loss: 0.1371 - accuracy: 0.9522 - precision:
0.9245 - recall: 0.8509
Epoch 00022: val_recall did not improve from 0.87395
106/106 [==============================] - 685s 6s/step - loss: 0.1371 - accuracy: 0.9522 - precis
ion: 0.9245 - recall: 0.8509 - val_loss: 0.1176 - val_accuracy: 0.9573 - val_precision: 0.9660 - v
al_recall: 0.8391
Epoch 23/30
106/106 [==============================] - ETA: 0s - loss: 0.1297 - accuracy: 0.9592 - precision:
0.9415 - recall: 0.8693
Epoch 00023: val_recall did not improve from 0.87395
106/106 [==============================] - 712s 7s/step - loss: 0.1297 - accuracy: 0.9592 - precis
ion: 0.9415 - recall: 0.8693 - val_loss: 0.1093 - val_accuracy: 0.9639 - val_precision: 0.9664 - v
al_recall: 0.8658
Epoch 24/30
106/106 [==============================] - ETA: 0s - loss: 0.1116 - accuracy: 0.9649 - precision:
0.9445 - recall: 0.8995
Epoch 00024: val_recall did not improve from 0.87395
106/106 [==============================] - 717s 7s/step - loss: 0.1116 - accuracy: 0.9649 - precis
ion: 0.9445 - recall: 0.8995 - val_loss: 0.1225 - val_accuracy: 0.9573 - val_precision: 0.9778 - v
al_recall: 0.8486
Epoch 25/30
106/106 [==============================] - ETA: 0s - loss: 0.1069 - accuracy: 0.9662 - precision:
0.9469 - recall: 0.8968
Epoch 00025: val_recall improved from 0.87395 to 0.96226, saving model to model_save/weights-25-0.
9623.hdf5
106/106 [==============================] - 719s 7s/step - loss: 0.1069 - accuracy: 0.9662 - precis
ion: 0.9469 - recall: 0.8968 - val_loss: 0.0907 - val_accuracy: 0.9778 - val_precision: 0.9395 - v
al_recall: 0.9623
Epoch 26/30
106/106 [==============================] - ETA: 0s - loss: 0.0886 - accuracy: 0.9761 - precision:
0.9670 - recall: 0.9209
Epoch 00026: val_recall did not improve from 0.96226
106/106 [==============================] - 721s 7s/step - loss: 0.0886 - accuracy: 0.9761 - precis
ion: 0.9670 - recall: 0.9209 - val_loss: 0.0749 - val_accuracy: 0.9826 - val_precision: 0.9864 - v
al_recall: 0.9381
Epoch 27/30
106/106 [==============================] - ETA: 0s - loss: 0.0779 - accuracy: 0.9805 - precision:
0.9723 - recall: 0.9405
Epoch 00027: val_recall did not improve from 0.96226
106/106 [==============================] - 722s 7s/step - loss: 0.0779 - accuracy: 0.9805 - precis
ion: 0.9723 - recall: 0.9405 - val_loss: 0.0745 - val_accuracy: 0.9796 - val_precision: 0.9613 - v
al_recall: 0.9457
Epoch 28/30
106/106 [==============================] - ETA: 0s - loss: 0.0727 - accuracy: 0.9816 - precision:
0.9738 - recall: 0.9458
Epoch 00028: val_recall did not improve from 0.96226
106/106 [==============================] - 706s 7s/step - loss: 0.0727 - accuracy: 0.9816 - precis
ion: 0.9738 - recall: 0.9458 - val_loss: 0.0659 - val_accuracy: 0.9802 - val_precision: 0.9733 - v
al_recall: 0.9318
Epoch 29/30
106/106 [==============================] - ETA: 0s - loss: 0.0610 - accuracy: 0.9842 - precision:
0.9735 - recall: 0.9547
Epoch 00029: val_recall did not improve from 0.96226
106/106 [==============================] - 712s 7s/step - loss: 0.0610 - accuracy: 0.9842 - precis
ion: 0.9735 - recall: 0.9547 - val_loss: 0.0547 - val_accuracy: 0.9880 - val_precision: 0.9943 - v
al_recall: 0.9508
```

```
Epoch 30/30
106/106 [==============================] - ETA: 0s - loss: 0.0568 - accuracy: 0.9854 - precision:
0.9811 - recall: 0.9512
Epoch 00030: val_recall improved from 0.96226 to 0.97832, saving model to model_save/weights-30-0.
9783.hdf5
106/106 [==============================] - 718s 7s/step - loss: 0.0568 - accuracy: 0.9854 - precis
ion: 0.9811 - recall: 0.9512 - val_loss: 0.0470 - val_accuracy: 0.9916 - val_precision: 0.9837 - v
al_recall: 0.9783
```

Out[ ]:

```
<tensorflow.python.keras.callbacks.History at 0x7f76b9f64cf8>
```

cross validation-3

In [ ]:

```
tf.keras.backend.clear_session()
checkpoint_path = "training_1/cp.ckpt"
checkpoint_dir = os.path.dirname(checkpoint_path)
# Tensorbaord
! rm -rf ./logs/
logdir = os.path.join("logs", datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
%tensorboard --logdir $logdir
tensorboard_callback = tf.keras.callbacks.TensorBoard(logdir, histogram_freq=1)

model_1 =tf.keras.applications.vgg16.VGG16(weights = "imagenet", include_top=False, input_shape = (
256,256,3))
for i in model_1.layers:
  i.trainable=False
model=model_1.output
model=Conv2D(32, (3, 3))(model)
model=(Activation('relu'))(model)
model=(MaxPool2D(pool_size=(2, 2)))(model)
model=Flatten()(model)
model = Dense(256, activation="relu")(model)
model = Dense(128, activation="relu")(model)
output_layer = Dense(1, activation="sigmoid")(model)
model1 = Model(model_1.input,output_layer)
model1.compile(loss = "binary_crossentropy", optimizer =tf.keras.optimizers.Adam(lr=0.0001), metric
s=["accuracy",
tf.keras.metrics.Precision(name='precision'),tf.keras.metrics.Recall(name='recall')])
model1.fit(train3,epochs=30,verbose=True,validation_data=cv2,batch_size=64,callbacks=[checkpoint,te
nsorboard_callback])
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-
applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58892288/58889256 [==============================] - 1s 0us/step
Epoch 1/30
  1/106 [..............................] - ETA: 0s - loss: 0.6799 - accuracy: 0.6406 - precision:
0.2727 - recall: 0.4615WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/summary_ops_v2.py:1277: stop (from
tensorflow.python.eager.profiler) is deprecated and will be removed after 2020-07-01.
Instructions for updating:
use `tf.profiler.experimental.stop` instead.
106/106 [==============================] - ETA: 0s - loss: 0.5171 - accuracy: 0.7754 - precision:
0.2727 - recall: 0.0040
Epoch 00001: val_recall improved from -inf to 0.00266, saving model to model_save/weights-01-0.002
7.hdf5
106/106 [==============================] - 615s 6s/step - loss: 0.5171 - accuracy: 0.7754 - precis
ion: 0.2727 - recall: 0.0040 - val_loss: 0.4840 - val_accuracy: 0.7740 - val_precision: 0.5000 - v
al_recall: 0.0027
Epoch 2/30
106/106 [==============================] - ETA: 0s - loss: 0.4589 - accuracy: 0.7826 - precision:
0.6506 - recall: 0.1045
Epoch 00002: val_recall improved from 0.00266 to 0.21488, saving model to model_save/weights-02-0.
2149.hdf5
106/106 [==============================] - 609s 6s/step - loss: 0.4589 - accuracy: 0.7826 - precis
ion: 0.6506 - recall: 0.1045 - val_loss: 0.4123 - val_accuracy: 0.8095 - val_precision: 0.7091 - v
al_recall: 0.2149
Epoch 3/30
106/106 [==============================] - ETA: 0s - loss: 0.4175 - accuracy: 0.8062 - precision:
0.6667 - recall: 0.2686
Epoch 00003: val_recall improved from 0.21488 to 0.38950, saving model to model_save/weights-03-0.
Epoch 00003: val_recall improved from 0.21488 to 0.38950, saving model to model_save/weights-03-0.
3895.hdf5
106/106 [==============================] - 611s 6s/step - loss: 0.4175 - accuracy: 0.8062 - precis
ion: 0.6667 - recall: 0.2686 - val_loss: 0.4040 - val_accuracy: 0.8167 - val_precision: 0.6267 - v
al_recall: 0.3895
Epoch 4/30
106/106 [==============================] - ETA: 0s - loss: 0.3987 - accuracy: 0.8125 - precision:
0.6610 - recall: 0.3327
Epoch 00004: val_recall did not improve from 0.38950
106/106 [==============================] - 610s 6s/step - loss: 0.3987 - accuracy: 0.8125 - precis
ion: 0.6610 - recall: 0.3327 - val_loss: 0.3758 - val_accuracy: 0.8239 - val_precision: 0.7353 - v
al_recall: 0.2801
Epoch 5/30
106/106 [==============================] - ETA: 0s - loss: 0.3788 - accuracy: 0.8258 - precision:
0.6872 - recall: 0.4054
Epoch 00005: val_recall improved from 0.38950 to 0.57031, saving model to model_save/weights-05-0.
5703.hdf5
106/106 [==============================] - 595s 6s/step - loss: 0.3788 - accuracy: 0.8258 - precis
ion: 0.6872 - recall: 0.4054 - val_loss: 0.3825 - val_accuracy: 0.8257 - val_precision: 0.6366 - v
al_recall: 0.5703
Epoch 6/30
106/106 [==============================] - ETA: 0s - loss: 0.3671 - accuracy: 0.8390 - precision:
0.7201 - recall: 0.4457
Epoch 00006: val_recall did not improve from 0.57031
106/106 [==============================] - 608s 6s/step - loss: 0.3671 - accuracy: 0.8390 - precis
ion: 0.7201 - recall: 0.4457 - val_loss: 0.3493 - val_accuracy: 0.8456 - val_precision: 0.7532 - v
al_recall: 0.4652
Epoch 7/30
106/106 [==============================] - ETA: 0s - loss: 0.3458 - accuracy: 0.8436 - precision:
0.7189 - recall: 0.4561
Epoch 00007: val_recall did not improve from 0.57031
106/106 [==============================] - 610s 6s/step - loss: 0.3458 - accuracy: 0.8436 - precis
ion: 0.7189 - recall: 0.4561 - val_loss: 0.3244 - val_accuracy: 0.8582 - val_precision: 0.7696 - v
al_recall: 0.4538
Epoch 8/30
106/106 [==============================] - ETA: 0s - loss: 0.3470 - accuracy: 0.8483 - precision:
0.7371 - recall: 0.4993
Epoch 00008: val_recall did not improve from 0.57031
106/106 [==============================] - 603s 6s/step - loss: 0.3470 - accuracy: 0.8483 - precis
ion: 0.7371 - recall: 0.4993 - val_loss: 0.3225 - val_accuracy: 0.8606 - val_precision: 0.7590 - v
al_recall: 0.5612
Epoch 9/30
106/106 [==============================] - ETA: 0s - loss: 0.3144 - accuracy: 0.8635 - precision:
0.7641 - recall: 0.5341
Epoch 00009: val_recall improved from 0.57031 to 0.58333, saving model to model_save/weights-09-0.
5833.hdf5
106/106 [==============================] - 607s 6s/step - loss: 0.3144 - accuracy: 0.8635 - precis
ion: 0.7641 - recall: 0.5341 - val_loss: 0.3148 - val_accuracy: 0.8732 - val_precision: 0.7949 - v
al_recall: 0.5833
Epoch 10/30
106/106 [==============================] - ETA: 0s - loss: 0.3283 - accuracy: 0.8589 - precision:
0.7635 - recall: 0.5490
Epoch 00010: val_recall improved from 0.58333 to 0.69600, saving model to model_save/weights-10-0.
6960.hdf5
106/106 [==============================] - 625s 6s/step - loss: 0.3283 - accuracy: 0.8589 - precis
ion: 0.7635 - recall: 0.5490 - val_loss: 0.3163 - val_accuracy: 0.8720 - val_precision: 0.7250 - v
al_recall: 0.6960
Epoch 11/30
106/106 [==============================] - ETA: 0s - loss: 0.3001 - accuracy: 0.8725 - precision:
0.7849 - recall: 0.5836
Epoch 00011: val_recall did not improve from 0.69600
106/106 [==============================] - 614s 6s/step - loss: 0.3001 - accuracy: 0.8725 - precis
ion: 0.7849 - recall: 0.5836 - val_loss: 0.2861 - val_accuracy: 0.8810 - val_precision: 0.8838 - v
al_recall: 0.5000
Epoch 12/30
106/106 [==============================] - ETA: 0s - loss: 0.2815 - accuracy: 0.8819 - precision:
0.8094 - recall: 0.6491
Epoch 00012: val_recall did not improve from 0.69600
106/106 [==============================] - 613s 6s/step - loss: 0.2815 - accuracy: 0.8819 - precis
ion: 0.8094 - recall: 0.6491 - val_loss: 0.2768 - val_accuracy: 0.8786 - val_precision: 0.8482 - v
al_recall: 0.5722
Epoch 13/30
106/106 [==============================] - ETA: 0s - loss: 0.2646 - accuracy: 0.8922 - precision:
0.8149 - recall: 0.6604
Epoch 00013: val_recall improved from 0.69600 to 0.82961, saving model to model_save/weights-13-0.
8296.hdf5
106/106 [==============================] - 610s 6s/step - loss: 0.2646 - accuracy: 0.8922 - precis
ion: 0.8149 - recall: 0.6604 - val_loss: 0.2644 - val_accuracy: 0.9014 - val_precision: 0.7425 - v
```

```
ion: 0.8149 - recall: 0.6604 - val_loss: 0.2644 - val_accuracy: 0.9014 - val_precision: 0.7425 - v
al_recall: 0.8296
Epoch 14/30
106/106 [==============================] - ETA: 0s - loss: 0.2674 - accuracy: 0.8931 - precision:
0.8342 - recall: 0.6768
Epoch 00014: val_recall did not improve from 0.82961
106/106 [==============================] - 608s 6s/step - loss: 0.2674 - accuracy: 0.8931 - precis
ion: 0.8342 - recall: 0.6768 - val_loss: 0.2422 - val_accuracy: 0.8990 - val_precision: 0.8607 - v
al_recall: 0.6514
Epoch 15/30
106/106 [==============================] - ETA: 0s - loss: 0.2369 - accuracy: 0.9065 - precision:
0.8462 - recall: 0.7157
Epoch 00015: val_recall did not improve from 0.82961
106/106 [==============================] - 606s 6s/step - loss: 0.2369 - accuracy: 0.9065 - precis
ion: 0.8462 - recall: 0.7157 - val_loss: 0.2255 - val_accuracy: 0.9105 - val_precision: 0.9222 - v
al_recall: 0.6475
Epoch 16/30
106/106 [==============================] - ETA: 0s - loss: 0.2308 - accuracy: 0.9091 - precision:
0.8513 - recall: 0.7243
Epoch 00016: val_recall improved from 0.82961 to 0.85359, saving model to model_save/weights-16-0.
8536.hdf5
106/106 [==============================] - 605s 6s/step - loss: 0.2308 - accuracy: 0.9091 - precis
ion: 0.8513 - recall: 0.7243 - val_loss: 0.2146 - val_accuracy: 0.9213 - val_precision: 0.7984 - v
al_recall: 0.8536
Epoch 17/30
106/106 [==============================] - ETA: 0s - loss: 0.2197 - accuracy: 0.9133 - precision:
0.8593 - recall: 0.7356
Epoch 00017: val_recall did not improve from 0.85359
106/106 [==============================] - 604s 6s/step - loss: 0.2197 - accuracy: 0.9133 - precis
ion: 0.8593 - recall: 0.7356 - val_loss: 0.1960 - val_accuracy: 0.9345 - val_precision: 0.8732 - v
al_recall: 0.8177
Epoch 18/30
106/106 [==============================] - ETA: 0s - loss: 0.1968 - accuracy: 0.9248 - precision:
0.8792 - recall: 0.7640
Epoch 00018: val_recall did not improve from 0.85359
106/106 [==============================] - 625s 6s/step - loss: 0.1968 - accuracy: 0.9248 - precis
ion: 0.8792 - recall: 0.7640 - val_loss: 0.1921 - val_accuracy: 0.9321 - val_precision: 0.9162 - v
al_recall: 0.7905
Epoch 19/30
106/106 [==============================] - ETA: 0s - loss: 0.1831 - accuracy: 0.9328 - precision:
0.8928 - recall: 0.7969
Epoch 00019: val_recall did not improve from 0.85359
106/106 [==============================] - 675s 6s/step - loss: 0.1831 - accuracy: 0.9328 - precis
ion: 0.8928 - recall: 0.7969 - val_loss: 0.1846 - val_accuracy: 0.9267 - val_precision: 0.9322 - v
al_recall: 0.7294
Epoch 20/30
106/106 [==============================] - ETA: 0s - loss: 0.1772 - accuracy: 0.9350 - precision:
0.8963 - recall: 0.8025
Epoch 00020: val_recall improved from 0.85359 to 0.85753, saving model to model_save/weights-20-0.
8575.hdf5
106/106 [==============================] - 653s 6s/step - loss: 0.1772 - accuracy: 0.9350 - precis
ion: 0.8963 - recall: 0.8025 - val_loss: 0.1630 - val_accuracy: 0.9477 - val_precision: 0.9037 - v
al_recall: 0.8575
Epoch 21/30
106/106 [==============================] - ETA: 0s - loss: 0.1594 - accuracy: 0.9452 - precision:
0.9141 - recall: 0.8368
Epoch 00021: val_recall improved from 0.85753 to 0.89855, saving model to model_save/weights-21-0.
8986.hdf5
106/106 [==============================] - 646s 6s/step - loss: 0.1594 - accuracy: 0.9452 - precis
ion: 0.9141 - recall: 0.8368 - val_loss: 0.1436 - val_accuracy: 0.9519 - val_precision: 0.8732 - v
al_recall: 0.8986
Epoch 22/30
106/106 [==============================] - ETA: 0s - loss: 0.1421 - accuracy: 0.9518 - precision:
0.9247 - recall: 0.8495
Epoch 00022: val_recall did not improve from 0.89855
106/106 [==============================] - 642s 6s/step - loss: 0.1421 - accuracy: 0.9518 - precis
ion: 0.9247 - recall: 0.8495 - val_loss: 0.1377 - val_accuracy: 0.9585 - val_precision: 0.9484 - v
al_recall: 0.8747
Epoch 23/30
106/106 [==============================] - ETA: 0s - loss: 0.1337 - accuracy: 0.9556 - precision:
0.9309 - recall: 0.8767
Epoch 00023: val_recall did not improve from 0.89855
106/106 [==============================] - 637s 6s/step - loss: 0.1337 - accuracy: 0.9556 - precis
ion: 0.9309 - recall: 0.8767 - val_loss: 0.1264 - val_accuracy: 0.9627 - val_precision: 0.9524 - v
al_recall: 0.8831
Epoch 24/30
106/106 [==============================] - ETA: 0s - loss: 0.1256 - accuracy: 0.9598 - precision:
```

```
0.9280 - recall: 0.8849
Epoch 00024: val_recall did not improve from 0.89855
106/106 [==============================] - 638s 6s/step - loss: 0.1256 - accuracy: 0.9598 - precis
ion: 0.9280 - recall: 0.8849 - val_loss: 0.1053 - val_accuracy: 0.9633 - val_precision: 0.9669 - v
al_recall: 0.8652
Epoch 25/30
106/106 [==============================] - ETA: 0s - loss: 0.1088 - accuracy: 0.9676 - precision:
0.9500 - recall: 0.9041
Epoch 00025: val_recall improved from 0.89855 to 0.94545, saving model to model_save/weights-25-0.
9455.hdf5
106/106 [==============================] - 644s 6s/step - loss: 0.1088 - accuracy: 0.9676 - precis
ion: 0.9500 - recall: 0.9041 - val_loss: 0.0898 - val_accuracy: 0.9772 - val_precision: 0.9554 - v
al_recall: 0.9455
Epoch 26/30
106/106 [==============================] - ETA: 0s - loss: 0.1024 - accuracy: 0.9682 - precision:
0.9486 - recall: 0.9109
Epoch 00026: val_recall improved from 0.94545 to 0.95238, saving model to model_save/weights-26-0.
9524.hdf5
106/106 [==============================] - 645s 6s/step - loss: 0.1024 - accuracy: 0.9682 - precis
ion: 0.9486 - recall: 0.9109 - val_loss: 0.0968 - val_accuracy: 0.9754 - val_precision: 0.9275 - v
al_recall: 0.9524
Epoch 27/30
106/106 [==============================] - ETA: 0s - loss: 0.0882 - accuracy: 0.9764 - precision:
0.9616 - recall: 0.9328
Epoch 00027: val_recall did not improve from 0.95238
106/106 [==============================] - 640s 6s/step - loss: 0.0882 - accuracy: 0.9764 - precis
ion: 0.9616 - recall: 0.9328 - val_loss: 0.0736 - val_accuracy: 0.9784 - val_precision: 0.9702 - v
al_recall: 0.9261
Epoch 28/30
106/106 [==============================] - ETA: 0s - loss: 0.0845 - accuracy: 0.9764 - precision:
0.9661 - recall: 0.9284
Epoch 00028: val_recall did not improve from 0.95238
106/106 [==============================] - 641s 6s/step - loss: 0.0845 - accuracy: 0.9764 - precis
ion: 0.9661 - recall: 0.9284 - val_loss: 0.0723 - val_accuracy: 0.9826 - val_precision: 0.9837 - v
al_recall: 0.9404
Epoch 29/30
106/106 [==============================] - ETA: 0s - loss: 0.0706 - accuracy: 0.9823 - precision:
0.9748 - recall: 0.9452
Epoch 00029: val_recall did not improve from 0.95238
106/106 [==============================] - 685s 6s/step - loss: 0.0706 - accuracy: 0.9823 - precis
ion: 0.9748 - recall: 0.9452 - val_loss: 0.0675 - val_accuracy: 0.9814 - val_precision: 0.9834 - v
al_recall: 0.9344
Epoch 30/30
106/106 [==============================] - ETA: 0s - loss: 0.0589 - accuracy: 0.9872 - precision:
0.9845 - recall: 0.9555
Epoch 00030: val_recall did not improve from 0.95238
106/106 [==============================] - 687s 6s/step - loss: 0.0589 - accuracy: 0.9872 - precis
ion: 0.9845 - recall: 0.9555 - val_loss: 0.0514 - val_accuracy: 0.9862 - val_precision: 0.9886 - v
al_recall: 0.9481
```

Out[ ]:

```
<tensorflow.python.keras.callbacks.History at 0x7f2065ef1550>
```

This model can certainly be improved by training for a few more epochs so training for 2 more epochs would be fine to get the best model

In [ ]:

```
%tensorboard --logdir $logdir
tensorboard_callback = tf.keras.callbacks.TensorBoard(logdir, histogram_freq=1)
model1.fit(train3,initial_epoch=30,epochs=32,verbose=True,validation_data=cv2,batch_size=64,callbac
ks=[checkpoint,tensorboard_callback])
```

```
Epoch 31/32
106/106 [==============================] - ETA: 0s - loss: 0.0583 - accuracy: 0.9869 - precision:
0.9820 - recall: 0.9596
Epoch 00031: val_recall improved from 0.95238 to 0.96535, saving model to model_save/weights-31-0.
9653.hdf5
106/106 [==============================] - 691s 7s/step - loss: 0.0583 - accuracy: 0.9869 - precis
ion: 0.9820 - recall: 0.9596 - val_loss: 0.0512 - val_accuracy: 0.9904 - val_precision: 0.9949 - v
al_recall: 0.9653
Epoch 32/32
106/106 [==============================] - ETA: 0s - loss: 0.0474 - accuracy: 0.9909 - precision:
```

```
0.9824 - recall: 0.9774
Epoch 00032: val_recall improved from 0.96535 to 0.97837, saving model to model_save/weights-32-0.
9784.hdf5
106/106 [==============================] - 700s 7s/step - loss: 0.0474 - accuracy: 0.9909 - precis
ion: 0.9824 - recall: 0.9774 - val_loss: 0.0409 - val_accuracy: 0.9928 - val_precision: 0.9927 - v
al_recall: 0.9784
```

Out[ ]:

```
<tensorflow.python.keras.callbacks.History at 0x7f1e147cae48>
```

cross validation-4

In [ ]:

```
tf.keras.backend.clear_session()
checkpoint_path = "training_1/cp.ckpt"
checkpoint_dir = os.path.dirname(checkpoint_path)
# Tensorbaord
! rm -rf ./logs/
logdir = os.path.join("logs", datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
%tensorboard --logdir $logdir
tensorboard_callback = tf.keras.callbacks.TensorBoard(logdir, histogram_freq=1)

model_1 =tf.keras.applications.vgg16.VGG16(weights = "imagenet", include_top=False, input_shape = (
256,256,3))
for i in model_1.layers:
  i.trainable=False
model=model_1.output
model=Conv2D(32, (3, 3))(model)
model=(Activation('relu'))(model)
model=(MaxPool2D(pool_size=(2, 2)))(model)
model=Flatten()(model)
model = Dense(256, activation="relu")(model)
model = Dense(128, activation="relu")(model)
output_layer = Dense(1, activation="sigmoid")(model)
model1 = Model(model_1.input,output_layer)
model1.compile(loss = "binary_crossentropy", optimizer =tf.keras.optimizers.Adam(lr=0.0001), metric
s=["accuracy",
tf.keras.metrics.Precision(name='precision'),tf.keras.metrics.Recall(name='recall')])
model1.fit(train4,epochs=30,verbose=True,validation_data=cv3,batch_size=64,callbacks=[checkpoint,te
nsorboard_callback])
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-
applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58892288/58889256 [==============================] - 0s 0us/step
Epoch 1/30
  1/106 [..............................] - ETA: 0s - loss: 0.8885 - accuracy: 0.2031 - precision:
0.2031 - recall: 1.0000WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/summary_ops_v2.py:1277: stop (from
tensorflow.python.eager.profiler) is deprecated and will be removed after 2020-07-01.
Instructions for updating:
use `tf.profiler.experimental.stop` instead.
106/106 [==============================] - ETA: 0s - loss: 0.5298 - accuracy: 0.7585 - precision:
0.2695 - recall: 0.0288
Epoch 00001: val_recall improved from -inf to 0.00000, saving model to model_save/weights-01-0.000
0.hdf5
106/106 [==============================] - 649s 6s/step - loss: 0.5298 - accuracy: 0.7585 - precis
ion: 0.2695 - recall: 0.0288 - val_loss: 0.4931 - val_accuracy: 0.7686 - val_precision: 0.0000e+00
- val_recall: 0.0000e+00
Epoch 2/30
106/106 [==============================] - ETA: 0s - loss: 0.4521 - accuracy: 0.7897 - precision:
0.6667 - recall: 0.1049
Epoch 00002: val_recall improved from 0.00000 to 0.29412, saving model to model_save/weights-02-0.
2941.hdf5
106/106 [==============================] - 649s 6s/step - loss: 0.4521 - accuracy: 0.7897 - precis
ion: 0.6667 - recall: 0.1049 - val_loss: 0.4217 - val_accuracy: 0.8137 - val_precision: 0.6442 - v
al_recall: 0.2941
Epoch 3/30
106/106 [==============================] - ETA: 0s - loss: 0.4231 - accuracy: 0.8057 - precision:
0.6898 - recall: 0.2965
Epoch 00003: val_recall improved from 0.29412 to 0.31481, saving model to model_save/weights-03-0.
3148.hdf5
```

```
106/106 [==============================] - 648s 6s/step - loss: 0.4231 - accuracy: 0.8057 - precis
ion: 0.6898 - recall: 0.2965 - val_loss: 0.4135 - val_accuracy: 0.8017 - val_precision: 0.6263 - v
al_recall: 0.3148
Epoch 4/30
106/106 [==============================] - ETA: 0s - loss: 0.4107 - accuracy: 0.8135 - precision:
0.6663 - recall: 0.3589
Epoch 00004: val_recall improved from 0.31481 to 0.43094, saving model to model_save/weights-04-0.
4309.hdf5
106/106 [==============================] - 645s 6s/step - loss: 0.4107 - accuracy: 0.8135 - precis
ion: 0.6663 - recall: 0.3589 - val_loss: 0.3871 - val_accuracy: 0.8281 - val_precision: 0.6610 - v
al_recall: 0.4309
Epoch 5/30
106/106 [==============================] - ETA: 0s - loss: 0.3810 - accuracy: 0.8256 - precision:
0.6862 - recall: 0.3846
Epoch 00005: val_recall did not improve from 0.43094
106/106 [==============================] - 644s 6s/step - loss: 0.3810 - accuracy: 0.8256 - precis
ion: 0.6862 - recall: 0.3846 - val_loss: 0.3434 - val_accuracy: 0.8510 - val_precision: 0.7892 - v
al_recall: 0.3808
Epoch 6/30
106/106 [==============================] - ETA: 0s - loss: 0.3713 - accuracy: 0.8373 - precision:
0.7344 - recall: 0.4650
Epoch 00006: val_recall improved from 0.43094 to 0.51928, saving model to model_save/weights-06-0.
5193.hdf5
106/106 [==============================] - 647s 6s/step - loss: 0.3713 - accuracy: 0.8373 - precis
ion: 0.7344 - recall: 0.4650 - val_loss: 0.3669 - val_accuracy: 0.8347 - val_precision: 0.6966 - v
al_recall: 0.5193
Epoch 7/30
106/106 [==============================] - ETA: 0s - loss: 0.3646 - accuracy: 0.8367 - precision:
0.7020 - recall: 0.4574
Epoch 00007: val_recall did not improve from 0.51928
106/106 [==============================] - 651s 6s/step - loss: 0.3646 - accuracy: 0.8367 - precis
ion: 0.7020 - recall: 0.4574 - val_loss: 0.3416 - val_accuracy: 0.8444 - val_precision: 0.8155 - v
al_recall: 0.4319
Epoch 8/30
106/106 [==============================] - ETA: 0s - loss: 0.3443 - accuracy: 0.8427 - precision:
0.7067 - recall: 0.4627
Epoch 00008: val_recall did not improve from 0.51928
106/106 [==============================] - 639s 6s/step - loss: 0.3443 - accuracy: 0.8427 - precis
ion: 0.7067 - recall: 0.4627 - val_loss: 0.3272 - val_accuracy: 0.8413 - val_precision: 0.7632 - v
al_recall: 0.3984
Epoch 9/30
106/106 [==============================] - ETA: 0s - loss: 0.3232 - accuracy: 0.8591 - precision:
0.7524 - recall: 0.5277
Epoch 00009: val_recall improved from 0.51928 to 0.56955, saving model to model_save/weights-09-0.
5696.hdf5
106/106 [==============================] - 642s 6s/step - loss: 0.3232 - accuracy: 0.8591 - precis
ion: 0.7524 - recall: 0.5277 - val_loss: 0.3093 - val_accuracy: 0.8642 - val_precision: 0.7778 - v
al_recall: 0.5696
Epoch 10/30
106/106 [==============================] - ETA: 0s - loss: 0.3163 - accuracy: 0.8660 - precision:
0.7685 - recall: 0.5668
Epoch 00010: val_recall improved from 0.56955 to 0.57453, saving model to model_save/weights-10-0.
5745.hdf5
106/106 [==============================] - 639s 6s/step - loss: 0.3163 - accuracy: 0.8660 - precis
ion: 0.7685 - recall: 0.5668 - val_loss: 0.3126 - val_accuracy: 0.8672 - val_precision: 0.7681 - v
al_recall: 0.5745
Epoch 11/30
106/106 [==============================] - ETA: 0s - loss: 0.3003 - accuracy: 0.8706 - precision:
0.7739 - recall: 0.5993
Epoch 00011: val_recall did not improve from 0.57453
106/106 [==============================] - 634s 6s/step - loss: 0.3003 - accuracy: 0.8706 - precis
ion: 0.7739 - recall: 0.5993 - val_loss: 0.2935 - val_accuracy: 0.8642 - val_precision: 0.8384 - v
al_recall: 0.5039
Epoch 12/30
106/106 [==============================] - ETA: 0s - loss: 0.2976 - accuracy: 0.8759 - precision:
0.7818 - recall: 0.6272
Epoch 00012: val_recall improved from 0.57453 to 0.78151, saving model to model_save/weights-12-0.
7815.hdf5
106/106 [==============================] - 639s 6s/step - loss: 0.2976 - accuracy: 0.8759 - precis
ion: 0.7818 - recall: 0.6272 - val_loss: 0.3086 - val_accuracy: 0.8720 - val_precision: 0.6739 - v
al_recall: 0.7815
Epoch 13/30
106/106 [==============================] - ETA: 0s - loss: 0.2817 - accuracy: 0.8813 - precision:
0.7903 - recall: 0.6374
Epoch 00013: val_recall did not improve from 0.78151
106/106 [==============================] - 642s 6s/step - loss: 0.2817 - accuracy: 0.8813 - precis
ion: 0.7903 - recall: 0.6374 - val_loss: 0.2845 - val_accuracy: 0.8792 - val_precision: 0.8643 - v
```

```
al_recall: 0.5733
Epoch 14/30
106/106 [==============================] - ETA: 0s - loss: 0.2692 - accuracy: 0.8878 - precision:
0.8144 - recall: 0.6580
Epoch 00014: val_recall did not improve from 0.78151
106/106 [==============================] - 649s 6s/step - loss: 0.2692 - accuracy: 0.8878 - precis
ion: 0.8144 - recall: 0.6580 - val_loss: 0.2699 - val_accuracy: 0.8930 - val_precision: 0.7714 - v
al_recall: 0.7337
Epoch 15/30
106/106 [==============================] - ETA: 0s - loss: 0.2518 - accuracy: 0.8965 - precision:
0.8191 - recall: 0.6923
Epoch 00015: val_recall did not improve from 0.78151
106/106 [==============================] - 649s 6s/step - loss: 0.2518 - accuracy: 0.8965 - precis
ion: 0.8191 - recall: 0.6923 - val_loss: 0.2355 - val_accuracy: 0.9111 - val_precision: 0.8492 - v
al_recall: 0.7175
Epoch 16/30
106/106 [==============================] - ETA: 0s - loss: 0.2385 - accuracy: 0.9067 - precision:
0.8418 - recall: 0.7047
Epoch 00016: val_recall did not improve from 0.78151
106/106 [==============================] - 639s 6s/step - loss: 0.2385 - accuracy: 0.9067 - precis
ion: 0.8418 - recall: 0.7047 - val_loss: 0.2244 - val_accuracy: 0.9135 - val_precision: 0.8514 - v
al_recall: 0.7641
Epoch 17/30
106/106 [==============================] - ETA: 0s - loss: 0.2287 - accuracy: 0.9119 - precision:
0.8633 - recall: 0.7167
Epoch 00017: val_recall did not improve from 0.78151
106/106 [==============================] - 638s 6s/step - loss: 0.2287 - accuracy: 0.9119 - precis
ion: 0.8633 - recall: 0.7167 - val_loss: 0.2169 - val_accuracy: 0.9141 - val_precision: 0.8908 - v
al_recall: 0.7016
Epoch 18/30
106/106 [==============================] - ETA: 0s - loss: 0.2103 - accuracy: 0.9173 - precision:
0.8528 - recall: 0.7508
Epoch 00018: val_recall improved from 0.78151 to 0.79088, saving model to model_save/weights-18-0.
7909.hdf5
106/106 [==============================] - 632s 6s/step - loss: 0.2103 - accuracy: 0.9173 - precis
ion: 0.8528 - recall: 0.7508 - val_loss: 0.2197 - val_accuracy: 0.9165 - val_precision: 0.8287 - v
al_recall: 0.7909
Epoch 19/30
106/106 [==============================] - ETA: 0s - loss: 0.1994 - accuracy: 0.9244 - precision:
0.8755 - recall: 0.7800
Epoch 00019: val_recall did not improve from 0.79088
106/106 [==============================] - 633s 6s/step - loss: 0.1994 - accuracy: 0.9244 - precis
ion: 0.8755 - recall: 0.7800 - val_loss: 0.2041 - val_accuracy: 0.9117 - val_precision: 0.9686 - v
al_recall: 0.6067
Epoch 20/30
106/106 [==============================] - ETA: 0s - loss: 0.1791 - accuracy: 0.9365 - precision:
0.8933 - recall: 0.8027
Epoch 00020: val_recall did not improve from 0.79088
106/106 [==============================] - 631s 6s/step - loss: 0.1791 - accuracy: 0.9365 - precis
ion: 0.8933 - recall: 0.8027 - val_loss: 0.1708 - val_accuracy: 0.9399 - val_precision: 0.9119 - v
al_recall: 0.7843
Epoch 21/30
106/106 [==============================] - ETA: 0s - loss: 0.1749 - accuracy: 0.9363 - precision:
0.8934 - recall: 0.8182
Epoch 00021: val_recall improved from 0.79088 to 0.85825, saving model to model_save/weights-21-0.
8582.hdf5
106/106 [==============================] - 630s 6s/step - loss: 0.1749 - accuracy: 0.9363 - precis
ion: 0.8934 - recall: 0.8182 - val_loss: 0.1666 - val_accuracy: 0.9447 - val_precision: 0.9000 - v
al_recall: 0.8582
Epoch 22/30
106/106 [==============================] - ETA: 0s - loss: 0.1587 - accuracy: 0.9458 - precision:
0.9131 - recall: 0.8287
Epoch 00022: val_recall did not improve from 0.85825
106/106 [==============================] - 627s 6s/step - loss: 0.1587 - accuracy: 0.9458 - precis
ion: 0.9131 - recall: 0.8287 - val_loss: 0.1572 - val_accuracy: 0.9399 - val_precision: 0.9091 - v
al_recall: 0.7955
Epoch 23/30
106/106 [==============================] - ETA: 0s - loss: 0.1484 - accuracy: 0.9483 - precision:
0.9235 - recall: 0.8375
Epoch 00023: val_recall improved from 0.85825 to 0.93872, saving model to model_save/weights-23-0.
9387.hdf5
106/106 [==============================] - 624s 6s/step - loss: 0.1484 - accuracy: 0.9483 - precis
ion: 0.9235 - recall: 0.8375 - val_loss: 0.1320 - val_accuracy: 0.9645 - val_precision: 0.9011 - v
al_recall: 0.9387
Epoch 24/30
106/106 [==============================] - ETA: 0s - loss: 0.1409 - accuracy: 0.9540 - precision:
0.9302 - recall: 0.8574
```

```
Epoch 00024: val_recall did not improve from 0.93872
106/106 [==============================] - 619s 6s/step - loss: 0.1409 - accuracy: 0.9540 - precis
ion: 0.9302 - recall: 0.8574 - val_loss: 0.1179 - val_accuracy: 0.9621 - val_precision: 0.9580 - v
al_recall: 0.8668
Epoch 25/30
106/106 [==============================] - ETA: 0s - loss: 0.1152 - accuracy: 0.9646 - precision:
0.9506 - recall: 0.8817
Epoch 00025: val_recall did not improve from 0.93872
106/106 [==============================] - 619s 6s/step - loss: 0.1152 - accuracy: 0.9646 - precis
ion: 0.9506 - recall: 0.8817 - val_loss: 0.1261 - val_accuracy: 0.9537 - val_precision: 0.9713 - v
al_recall: 0.8346
Epoch 26/30
106/106 [==============================] - ETA: 0s - loss: 0.1087 - accuracy: 0.9654 - precision:
0.9511 - recall: 0.8919
Epoch 00026: val_recall improved from 0.93872 to 0.95640, saving model to model_save/weights-26-0.
9564.hdf5
106/106 [==============================] - 620s 6s/step - loss: 0.1087 - accuracy: 0.9654 - precis
ion: 0.9511 - recall: 0.8919 - val_loss: 0.1061 - val_accuracy: 0.9748 - val_precision: 0.9242 - v
al_recall: 0.9564
Epoch 27/30
106/106 [==============================] - ETA: 0s - loss: 0.1058 - accuracy: 0.9648 - precision:
0.9483 - recall: 0.8915
Epoch 00027: val_recall did not improve from 0.95640
106/106 [==============================] - 621s 6s/step - loss: 0.1058 - accuracy: 0.9648 - precis
ion: 0.9483 - recall: 0.8915 - val_loss: 0.1403 - val_accuracy: 0.9423 - val_precision: 1.0000 - v
al_recall: 0.7588
Epoch 28/30
106/106 [==============================] - ETA: 0s - loss: 0.0888 - accuracy: 0.9755 - precision:
0.9635 - recall: 0.9233
Epoch 00028: val_recall did not improve from 0.95640
106/106 [==============================] - 622s 6s/step - loss: 0.0888 - accuracy: 0.9755 - precis
ion: 0.9635 - recall: 0.9233 - val_loss: 0.0746 - val_accuracy: 0.9784 - val_precision: 0.9437 - v
al_recall: 0.9544
Epoch 29/30
106/106 [==============================] - ETA: 0s - loss: 0.0760 - accuracy: 0.9800 - precision:
0.9665 - recall: 0.9437
Epoch 00029: val_recall improved from 0.95640 to 0.96875, saving model to model_save/weights-29-0.
9688.hdf5
106/106 [==============================] - 623s 6s/step - loss: 0.0760 - accuracy: 0.9800 - precis
ion: 0.9665 - recall: 0.9437 - val_loss: 0.0668 - val_accuracy: 0.9850 - val_precision: 0.9662 - v
al_recall: 0.9688
Epoch 30/30
106/106 [==============================] - ETA: 0s - loss: 0.0668 - accuracy: 0.9830 - precision:
0.9785 - recall: 0.9460
Epoch 00030: val_recall improved from 0.96875 to 0.97051, saving model to model_save/weights-30-0.
9705.hdf5
106/106 [==============================] - 619s 6s/step - loss: 0.0668 - accuracy: 0.9830 - precis
ion: 0.9785 - recall: 0.9460 - val_loss: 0.0575 - val_accuracy: 0.9874 - val_precision: 0.9731 - v
al_recall: 0.9705
```

Out[ ]:

```
<tensorflow.python.keras.callbacks.History at 0x7f6eb60a3f28>
```

cross validation-5

In [25]:

```
tf.keras.backend.clear_session()
checkpoint_path = "training_1/cp.ckpt"
checkpoint_dir = os.path.dirname(checkpoint_path)
# Tensorbaord
! rm -rf ./logs/
logdir = os.path.join("logs", datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
%tensorboard --logdir $logdir
tensorboard_callback = tf.keras.callbacks.TensorBoard(logdir, histogram_freq=1)

model_1 =tf.keras.applications.vgg16.VGG16(weights = "imagenet", include_top=False, input_shape = (
256,256,3))
for i in model_1.layers:
  i.trainable=False
model=model_1.output
model=Conv2D(32, (3, 3))(model)
model=(Activation('relu'))(model)
model=(MaxPool2D(pool_size=(2, 2)))(model)
```

```
model=(MaxPool2D(pool_size=(2, 2)))(model)
model=Flatten()(model)
model = Dense(256, activation="relu")(model)
model = Dense(128, activation="relu")(model)
output_layer = Dense(1, activation="sigmoid")(model)
model1 = Model(model_1.input,output_layer)
model1.compile(loss = "binary_crossentropy", optimizer =tf.keras.optimizers.Adam(lr=0.0001), metric
s=["accuracy",
tf.keras.metrics.Precision(name='precision'),tf.keras.metrics.Recall(name='recall')])
model1.fit(train5,epochs=30,verbose=True,validation_data=cv4,batch_size=64,callbacks=[checkpoint,te
nsorboard_callback])
```

```
Epoch 1/30
  1/106 [..............................] - ETA: 0s - loss: 0.5951 - accuracy: 0.7812 - precision:
0.0000e+00 - recall: 0.0000e+00WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/summary_ops_v2.py:1277: stop (from
tensorflow.python.eager.profiler) is deprecated and will be removed after 2020-07-01.
Instructions for updating:
use `tf.profiler.experimental.stop` instead.
106/106 [==============================] - ETA: 0s - loss: 0.5128 - accuracy: 0.7748 - precision:
0.3333 - recall: 6.5488e-04
Epoch 00001: val_recall improved from -inf to 0.04810, saving model to model_save/weights-01-0.048
1.hdf5
106/106 [==============================] - 508s 5s/step - loss: 0.5128 - accuracy: 0.7748 - precis
ion: 0.3333 - recall: 6.5488e-04 - val_loss: 0.4984 - val_accuracy: 0.7668 - val_precision: 0.6129
- val_recall: 0.0481
Epoch 2/30
106/106 [==============================] - ETA: 0s - loss: 0.4575 - accuracy: 0.7910 - precision:
0.6730 - recall: 0.1173
Epoch 00002: val_recall improved from 0.04810 to 0.13830, saving model to model_save/weights-02-0.
1383.hdf5
106/106 [==============================] - 505s 5s/step - loss: 0.4575 - accuracy: 0.7910 - precis
ion: 0.6730 - recall: 0.1173 - val_loss: 0.4335 - val_accuracy: 0.7945 - val_precision: 0.7429 - v
al_recall: 0.1383
Epoch 3/30
106/106 [==============================] - ETA: 0s - loss: 0.4136 - accuracy: 0.8078 - precision:
0.6481 - recall: 0.2601
Epoch 00003: val_recall improved from 0.13830 to 0.30245, saving model to model_save/weights-03-0.
3025.hdf5
106/106 [==============================] - 512s 5s/step - loss: 0.4136 - accuracy: 0.8078 - precis
ion: 0.6481 - recall: 0.2601 - val_loss: 0.3957 - val_accuracy: 0.8131 - val_precision: 0.6687 - v
al_recall: 0.3025
Epoch 4/30
106/106 [==============================] - ETA: 0s - loss: 0.3964 - accuracy: 0.8190 - precision:
0.6683 - recall: 0.3567
Epoch 00004: val_recall improved from 0.30245 to 0.57967, saving model to model_save/weights-04-0.
5797.hdf5
106/106 [==============================] - 507s 5s/step - loss: 0.3964 - accuracy: 0.8190 - precis
ion: 0.6683 - recall: 0.3567 - val_loss: 0.3899 - val_accuracy: 0.8383 - val_precision: 0.6453 - v
al_recall: 0.5797
Epoch 5/30
106/106 [==============================] - ETA: 0s - loss: 0.3748 - accuracy: 0.8320 - precision:
0.7009 - recall: 0.3870
Epoch 00005: val_recall improved from 0.57967 to 0.62169, saving model to model_save/weights-05-0.
6217.hdf5
106/106 [==============================] - 505s 5s/step - loss: 0.3748 - accuracy: 0.8320 - precis
ion: 0.7009 - recall: 0.3870 - val_loss: 0.4044 - val_accuracy: 0.8119 - val_precision: 0.5802 - v
al_recall: 0.6217
Epoch 6/30
106/106 [==============================] - ETA: 0s - loss: 0.3651 - accuracy: 0.8377 - precision:
0.6994 - recall: 0.4073
Epoch 00006: val_recall did not improve from 0.62169
106/106 [==============================] - 512s 5s/step - loss: 0.3651 - accuracy: 0.8377 - precis
ion: 0.6994 - recall: 0.4073 - val_loss: 0.3530 - val_accuracy: 0.8395 - val_precision: 0.7730 - v
al_recall: 0.3539
Epoch 7/30
106/106 [==============================] - ETA: 0s - loss: 0.3586 - accuracy: 0.8393 - precision:
0.7136 - recall: 0.4615
Epoch 00007: val_recall did not improve from 0.62169
106/106 [==============================] - 503s 5s/step - loss: 0.3586 - accuracy: 0.8393 - precis
ion: 0.7136 - recall: 0.4615 - val_loss: 0.3267 - val_accuracy: 0.8546 - val_precision: 0.6926 - v
al_recall: 0.5584
Epoch 8/30
106/106 [==============================] - ETA: 0s - loss: 0.3486 - accuracy: 0.8501 - precision:
0.7348 - recall: 0.5165
Epoch 00008: val_recall improved from 0.62169 to 0.71892, saving model to model_save/weights-08-0.
```

```
7189.hdf5
106/106 [==============================] - 507s 5s/step - loss: 0.3486 - accuracy: 0.8501 - precis
ion: 0.7348 - recall: 0.5165 - val_loss: 0.3639 - val_accuracy: 0.8522 - val_precision: 0.6520 - v
al_recall: 0.7189
Epoch 9/30
106/106 [==============================] - ETA: 0s - loss: 0.3301 - accuracy: 0.8552 - precision:
0.7364 - recall: 0.5157
Epoch 00009: val_recall did not improve from 0.71892
106/106 [==============================] - 501s 5s/step - loss: 0.3301 - accuracy: 0.8552 - precis
ion: 0.7364 - recall: 0.5157 - val_loss: 0.3312 - val_accuracy: 0.8552 - val_precision: 0.7773 - v
al_recall: 0.4581
Epoch 10/30
106/106 [==============================] - ETA: 0s - loss: 0.3171 - accuracy: 0.8647 - precision:
0.7527 - recall: 0.5643
Epoch 00010: val_recall did not improve from 0.71892
106/106 [==============================] - 510s 5s/step - loss: 0.3171 - accuracy: 0.8647 - precis
ion: 0.7527 - recall: 0.5643 - val_loss: 0.3283 - val_accuracy: 0.8522 - val_precision: 0.8095 - v
al_recall: 0.4215
Epoch 11/30
106/106 [==============================] - ETA: 0s - loss: 0.3089 - accuracy: 0.8675 - precision:
0.7725 - recall: 0.5867
Epoch 00011: val_recall did not improve from 0.71892
106/106 [==============================] - 503s 5s/step - loss: 0.3089 - accuracy: 0.8675 - precis
ion: 0.7725 - recall: 0.5867 - val_loss: 0.2955 - val_accuracy: 0.8690 - val_precision: 0.7842 - v
al_recall: 0.5798
Epoch 12/30
106/106 [==============================] - ETA: 0s - loss: 0.3100 - accuracy: 0.8665 - precision:
0.7604 - recall: 0.5675
Epoch 00012: val_recall did not improve from 0.71892
106/106 [==============================] - 497s 5s/step - loss: 0.3100 - accuracy: 0.8665 - precis
ion: 0.7604 - recall: 0.5675 - val_loss: 0.2836 - val_accuracy: 0.8876 - val_precision: 0.8134 - v
al_recall: 0.6329
Epoch 13/30
106/106 [==============================] - ETA: 0s - loss: 0.2875 - accuracy: 0.8809 - precision:
0.7920 - recall: 0.6237
Epoch 00013: val_recall did not improve from 0.71892
106/106 [==============================] - 487s 5s/step - loss: 0.2875 - accuracy: 0.8809 - precis
ion: 0.7920 - recall: 0.6237 - val_loss: 0.3015 - val_accuracy: 0.8678 - val_precision: 0.9067 - v
al_recall: 0.3977
Epoch 14/30
106/106 [==============================] - ETA: 0s - loss: 0.2831 - accuracy: 0.8807 - precision:
0.8010 - recall: 0.6167
Epoch 00014: val_recall improved from 0.71892 to 0.73964, saving model to model_save/weights-14-0.
7396.hdf5
106/106 [==============================] - 491s 5s/step - loss: 0.2831 - accuracy: 0.8807 - precis
ion: 0.8010 - recall: 0.6167 - val_loss: 0.2406 - val_accuracy: 0.9255 - val_precision: 0.8741 - v
al_recall: 0.7396
Epoch 15/30
106/106 [==============================] - ETA: 0s - loss: 0.2673 - accuracy: 0.8948 - precision:
0.8248 - recall: 0.6741
Epoch 00015: val_recall did not improve from 0.73964
106/106 [==============================] - 492s 5s/step - loss: 0.2673 - accuracy: 0.8948 - precis
ion: 0.8248 - recall: 0.6741 - val_loss: 0.2489 - val_accuracy: 0.8954 - val_precision: 0.8846 - v
al_recall: 0.6150
Epoch 16/30
106/106 [==============================] - ETA: 0s - loss: 0.2596 - accuracy: 0.8920 - precision:
0.8106 - recall: 0.6715
Epoch 00016: val_recall did not improve from 0.73964
106/106 [==============================] - 498s 5s/step - loss: 0.2596 - accuracy: 0.8920 - precis
ion: 0.8106 - recall: 0.6715 - val_loss: 0.2592 - val_accuracy: 0.8888 - val_precision: 0.9261 - v
al_recall: 0.5591
Epoch 17/30
106/106 [==============================] - ETA: 0s - loss: 0.2338 - accuracy: 0.9080 - precision:
0.8507 - recall: 0.7214
Epoch 00017: val_recall did not improve from 0.73964
106/106 [==============================] - 504s 5s/step - loss: 0.2338 - accuracy: 0.9080 - precis
ion: 0.8507 - recall: 0.7214 - val_loss: 0.2335 - val_accuracy: 0.9123 - val_precision: 0.8609 - v
al_recall: 0.7143
Epoch 18/30
106/106 [==============================] - ETA: 0s - loss: 0.2311 - accuracy: 0.9068 - precision:
0.8464 - recall: 0.7213
Epoch 00018: val_recall improved from 0.73964 to 0.75661, saving model to model_save/weights-18-0.
7566.hdf5
106/106 [==============================] - 510s 5s/step - loss: 0.2311 - accuracy: 0.9068 - precis
ion: 0.8464 - recall: 0.7213 - val_loss: 0.2070 - val_accuracy: 0.9237 - val_precision: 0.8910 - v
al_recall: 0.7566
Epoch 19/30
```

```
106/106 [==============================] - ETA: 0s - loss: 0.2152 - accuracy: 0.9170 - precision:
0.8645 - recall: 0.7590
Epoch 00019: val_recall did not improve from 0.75661
106/106 [==============================] - 513s 5s/step - loss: 0.2152 - accuracy: 0.9170 - precis
ion: 0.8645 - recall: 0.7590 - val_loss: 0.1879 - val_accuracy: 0.9273 - val_precision: 0.9051 - v
al_recall: 0.7230
Epoch 20/30
106/106 [==============================] - ETA: 0s - loss: 0.1977 - accuracy: 0.9241 - precision:
0.8801 - recall: 0.7743
Epoch 00020: val_recall did not improve from 0.75661
106/106 [==============================] - 507s 5s/step - loss: 0.1977 - accuracy: 0.9241 - precis
ion: 0.8801 - recall: 0.7743 - val_loss: 0.2054 - val_accuracy: 0.9201 - val_precision: 0.9206 - v
al_recall: 0.6967
Epoch 21/30
106/106 [==============================] - ETA: 0s - loss: 0.1928 - accuracy: 0.9285 - precision:
0.8903 - recall: 0.7809
Epoch 00021: val_recall did not improve from 0.75661
106/106 [==============================] - 505s 5s/step - loss: 0.1928 - accuracy: 0.9285 - precis
ion: 0.8903 - recall: 0.7809 - val_loss: 0.1784 - val_accuracy: 0.9309 - val_precision: 0.9460 - v
al_recall: 0.7525
Epoch 22/30
106/106 [==============================] - ETA: 0s - loss: 0.1745 - accuracy: 0.9351 - precision:
0.8970 - recall: 0.8121
Epoch 00022: val_recall improved from 0.75661 to 0.87324, saving model to model_save/weights-22-0.
8732.hdf5
106/106 [==============================] - 503s 5s/step - loss: 0.1745 - accuracy: 0.9351 - precis
ion: 0.8970 - recall: 0.8121 - val_loss: 0.1589 - val_accuracy: 0.9507 - val_precision: 0.8934 - v
al_recall: 0.8732
Epoch 23/30
106/106 [==============================] - ETA: 0s - loss: 0.1521 - accuracy: 0.9468 - precision:
0.9174 - recall: 0.8314
Epoch 00023: val_recall did not improve from 0.87324
106/106 [==============================] - 498s 5s/step - loss: 0.1521 - accuracy: 0.9468 - precis
ion: 0.9174 - recall: 0.8314 - val_loss: 0.1533 - val_accuracy: 0.9477 - val_precision: 0.9210 - v
al_recall: 0.8324
Epoch 24/30
106/106 [==============================] - ETA: 0s - loss: 0.1532 - accuracy: 0.9449 - precision:
0.9074 - recall: 0.8345
Epoch 00024: val_recall did not improve from 0.87324
106/106 [==============================] - 497s 5s/step - loss: 0.1532 - accuracy: 0.9449 - precis
ion: 0.9074 - recall: 0.8345 - val_loss: 0.1307 - val_accuracy: 0.9603 - val_precision: 0.9651 - v
al_recall: 0.8468
Epoch 25/30
106/106 [==============================] - ETA: 0s - loss: 0.1353 - accuracy: 0.9517 - precision:
0.9224 - recall: 0.8536
Epoch 00025: val_recall did not improve from 0.87324
106/106 [==============================] - 500s 5s/step - loss: 0.1353 - accuracy: 0.9517 - precis
ion: 0.9224 - recall: 0.8536 - val_loss: 0.1239 - val_accuracy: 0.9651 - val_precision: 0.9591 - v
al_recall: 0.8714
Epoch 26/30
106/106 [==============================] - ETA: 0s - loss: 0.1240 - accuracy: 0.9586 - precision:
0.9367 - recall: 0.8727
Epoch 00026: val_recall improved from 0.87324 to 0.93899, saving model to model_save/weights-26-0.
9390.hdf5
106/106 [==============================] - 517s 5s/step - loss: 0.1240 - accuracy: 0.9586 - precis
ion: 0.9367 - recall: 0.8727 - val_loss: 0.1074 - val_accuracy: 0.9681 - val_precision: 0.9219 - v
al_recall: 0.9390
Epoch 27/30
106/106 [==============================] - ETA: 0s - loss: 0.1157 - accuracy: 0.9617 - precision:
0.9361 - recall: 0.8768
Epoch 00027: val_recall did not improve from 0.93899
106/106 [==============================] - 531s 5s/step - loss: 0.1157 - accuracy: 0.9617 - precis
ion: 0.9361 - recall: 0.8768 - val_loss: 0.0948 - val_accuracy: 0.9712 - val_precision: 0.9382 - v
al_recall: 0.9220
Epoch 28/30
106/106 [==============================] - ETA: 0s - loss: 0.1133 - accuracy: 0.9623 - precision:
0.9397 - recall: 0.8821
Epoch 00028: val_recall did not improve from 0.93899
106/106 [==============================] - 536s 5s/step - loss: 0.1133 - accuracy: 0.9623 - precis
ion: 0.9397 - recall: 0.8821 - val_loss: 0.1011 - val_accuracy: 0.9730 - val_precision: 0.9469 - v
al_recall: 0.9346
Epoch 29/30
106/106 [==============================] - ETA: 0s - loss: 0.0925 - accuracy: 0.9733 - precision:
0.9563 - recall: 0.9194
Epoch 00029: val_recall did not improve from 0.93899
106/106 [==============================] - 537s 5s/step - loss: 0.0925 - accuracy: 0.9733 - precis
ion: 0.9563 - recall: 0.9194 - val_loss: 0.0869 - val_accuracy: 0.9724 - val_precision: 0.9855 - v
```

```
al_recall: 0.8924
Epoch 30/30
106/106 [==============================] - ETA: 0s - loss: 0.0824 - accuracy: 0.9749 - precision:
0.9558 - recall: 0.9289
Epoch 00030: val_recall did not improve from 0.93899
106/106 [==============================] - 543s 5s/step - loss: 0.0824 - accuracy: 0.9749 - precis
ion: 0.9558 - recall: 0.9289 - val_loss: 0.0895 - val_accuracy: 0.9651 - val_precision: 0.9937 - v
al_recall: 0.8499
```

Out[25]:

```
<tensorflow.python.keras.callbacks.History at 0x7f460c333a58>
```

This model can certainly be improved by training for a few more epochs so training for 2 more epochs would be fine to get the best model

In [27]:

```
%tensorboard --logdir $logdir
tensorboard_callback = tf.keras.callbacks.TensorBoard(logdir, histogram_freq=1)
model1.fit(train5,initial_epoch=30,epochs=32,verbose=True,validation_data=cv4,batch_size=64,callbac
ks=[checkpoint,tensorboard_callback])
```

```
Reusing TensorBoard on port 6008 (pid 281), started 5:37:20 ago. (Use '!kill 281' to kill it.)
```

```
Epoch 31/32
106/106 [==============================] - ETA: 0s - loss: 0.0756 - accuracy: 0.9794 - precision:
0.9711 - recall: 0.9371
Epoch 00031: val_recall improved from 0.93899 to 0.95798, saving model to model_save/weights-31-0.
9580.hdf5
106/106 [==============================] - 531s 5s/step - loss: 0.0756 - accuracy: 0.9794 - precis
ion: 0.9711 - recall: 0.9371 - val_loss: 0.0658 - val_accuracy: 0.9868 - val_precision: 0.9799 - v
al_recall: 0.9580
Epoch 32/32
106/106 [==============================] - ETA: 0s - loss: 0.0692 - accuracy: 0.9832 - precision:
0.9725 - recall: 0.9502
Epoch 00032: val_recall improved from 0.95798 to 0.97375, saving model to model_save/weights-32-0.
9738.hdf5
106/106 [==============================] - 540s 5s/step - loss: 0.0692 - accuracy: 0.9832 - precis
ion: 0.9725 - recall: 0.9502 - val_loss: 0.0537 - val_accuracy: 0.9910 - val_precision: 0.9867 - v
al_recall: 0.9738
```

Out[27]:

```
<tensorflow.python.keras.callbacks.History at 0x7f457c5795f8>
```

Checking average metrics of all the best models (5-fold Cross Validation)

In [20]:

```
from keras.models import load_model
best_model=['weights-30-0.9704.hdf5','weights-30-0.9705.hdf5','weights-30-0.9783.hdf5','weights-32
-0.9738.hdf5','weights-32-0.9784.hdf5']
loss=[]
accuracy=[]
precision=[]
recall=[]
for i in tqdm(best_model):
  model = load_model(i)
  a,b,c,d=model.evaluate(val_ds)
  loss.append(a)
  accuracy.append(b)
  precision.append(c)
  recall.append(d)
print("The average loss of all the models is",np.mean(loss))
print("The average accuracy of all the models is",np.mean(accuracy))
print("The average precision of all the models is",np.mean(precision))
print("The average recall of all the models is",np.mean(recall))
```

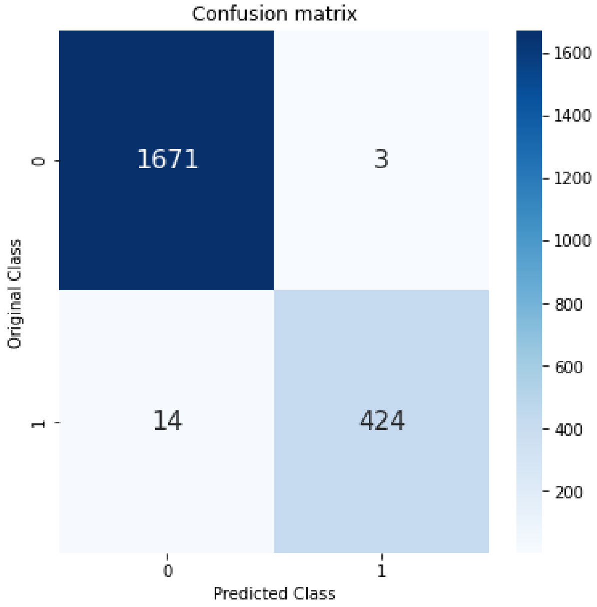
```
33/33 [==============================] - 45s 1s/step - loss: 0.0664 - accuracy: 0.9877 -
precision: 0.9804 - recall: 0.9637
33/33 [==============================] - 45s 1s/step - loss: 0.0601 - accuracy: 0.9853 -
precision: 0.9770 - recall: 0.9589
33/33 [==============================] - 46s 1s/step - loss: 0.0506 - accuracy: 0.9901 -
precision: 0.9812 - recall: 0.9752
33/33 [==============================] - 46s 1s/step - loss: 0.0590 - accuracy: 0.9844 -
precision: 0.9868 - recall: 0.9430
33/33 [==============================] - 46s 1s/step - loss: 0.0443 - accuracy: 0.9891 -
precision: 0.9836 - recall: 0.9632

The average loss of all the models is 0.05608991980552673
The average accuracy of all the models is 0.9873106122016907
The average precision of all the models is 0.981799042224884
The average recall of all the models is 0.9608038187026977
```

Based on cross validation data(not on final validation data) 'weights-32-0.9784.hdf5' is our best model

In [31]:

```
model1 = load_model('weights-32-0.9784.hdf5')
y_pred_1=[] #array to store predicted label
y_true=[] #array to store the ground truth
for i,j in tqdm(val_ds.take(2112)):
  y_pred_1.extend(model1.predict(i)) #predicting each batch
  y_true.extend(j)
y_pred=[]
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a
custom threshold (0.5 is the default threshold)
  if i[0]>=0.5: #setting threshold
    y_pred.append(1)
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
confusion_mat(y_true,y_pred)
```

```
Percentage of misclassified points  0.8049242424242424
```

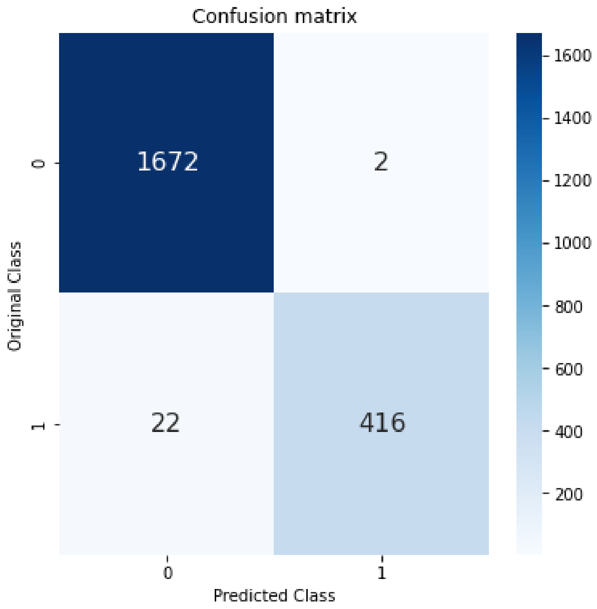


In [33]:

```
y_pred=[]
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a
custom threshold (0.5 is the default threshold)
  if i[0]>=0.6: #setting threshold
    y_pred.append(1)
  else:
```

```
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
confusion_mat(y_true,y_pred)
```

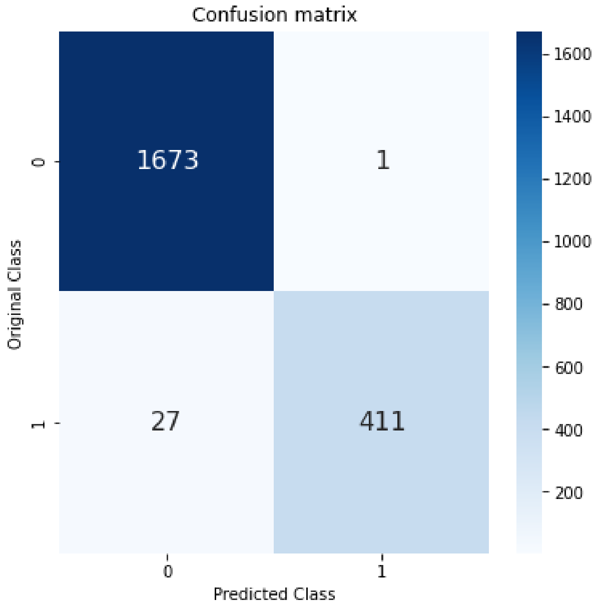
```
Percentage of misclassified points  1.1363636363636365
```



In [40]:

```
y_pred=[]
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a
custom threshold (0.5 is the default threshold)
  if i[0]>=0.7: #setting threshold
    y_pred.append(1)
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
confusion_mat(y_true,y_pred)
```

```
Percentage of misclassified points  1.3257575757575757
```

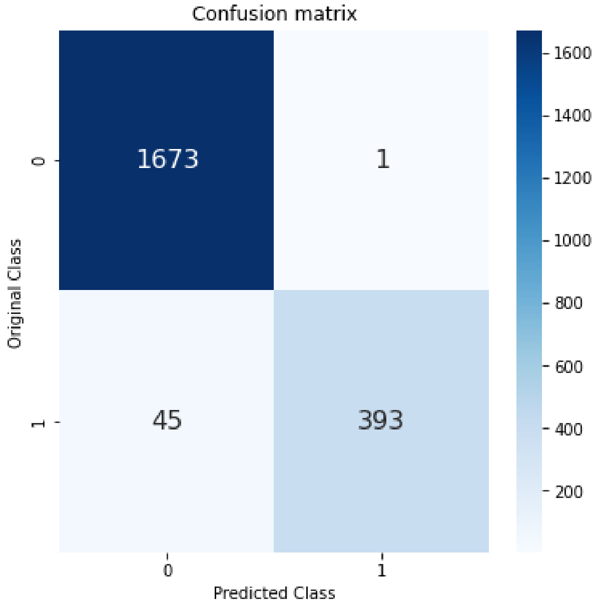


In [41]:

In [41]:

```
y_pred=[]
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a custom threshold (0.5 is the default threshold)
  if i[0]>=0.8: #setting threshold
    y_pred.append(1)
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
confusion_mat(y_true,y_pred)
```

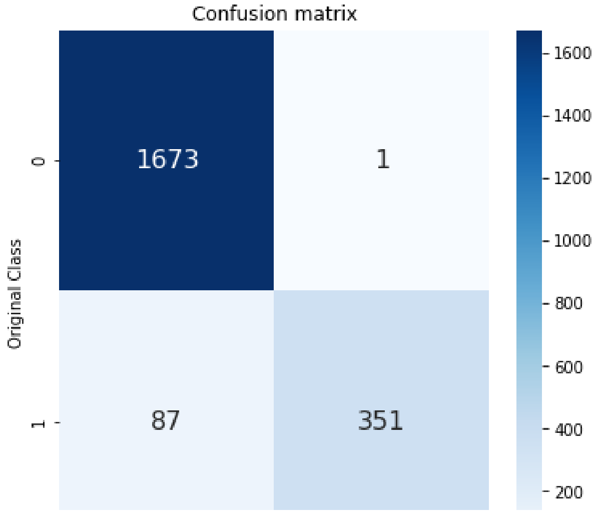
```
Percentage of misclassified points  2.178030303030303
```

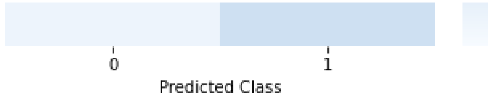


In [42]:

```
y_pred=[]
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a custom threshold (0.5 is the default threshold)
  if i[0]>=0.9: #setting threshold
    y_pred.append(1)
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
confusion_mat(y_true,y_pred)
```

```
Percentage of misclassified points  4.166666666666666
```





Having a look at the images which are wrongly classified

In [44]:

```
y_pred=[]
for i in y_pred_1: #the values are in probabilities and hence we are going to classify based on a custom threshold (0.5 is the default threshold)
  if i[0]>=0.5: #setting threshold
    y_pred.append(1)
  else:
    y_pred.append(0)
y_true=np.array(y_true) #converting the array for into numpy
y_pred=np.array(y_pred).reshape(1,-1)
y_pred=y_pred[0]
count=[] #stores the position of the wrongly classified point
count1=0
for i,j in zip(y_true,y_pred):
  if i!=j: #checking if the true and predicted class label
    count.append(count1)
    count1=count1+1
  else:
    count1=count1+1
```

In [67]:

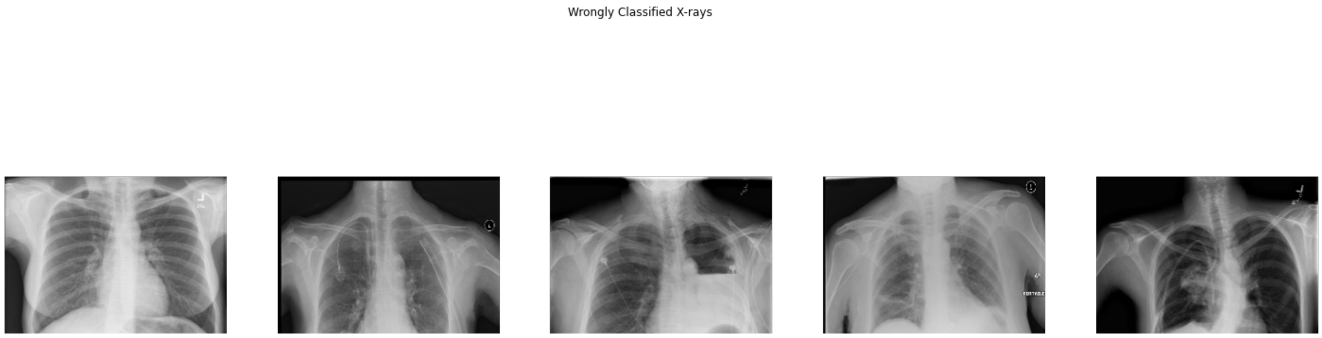
```
val=[]
for i,j in tqdm(val_ds.take(2112)): #storing all the validation data points in val array
  for k in i:
    val.append(k)
```

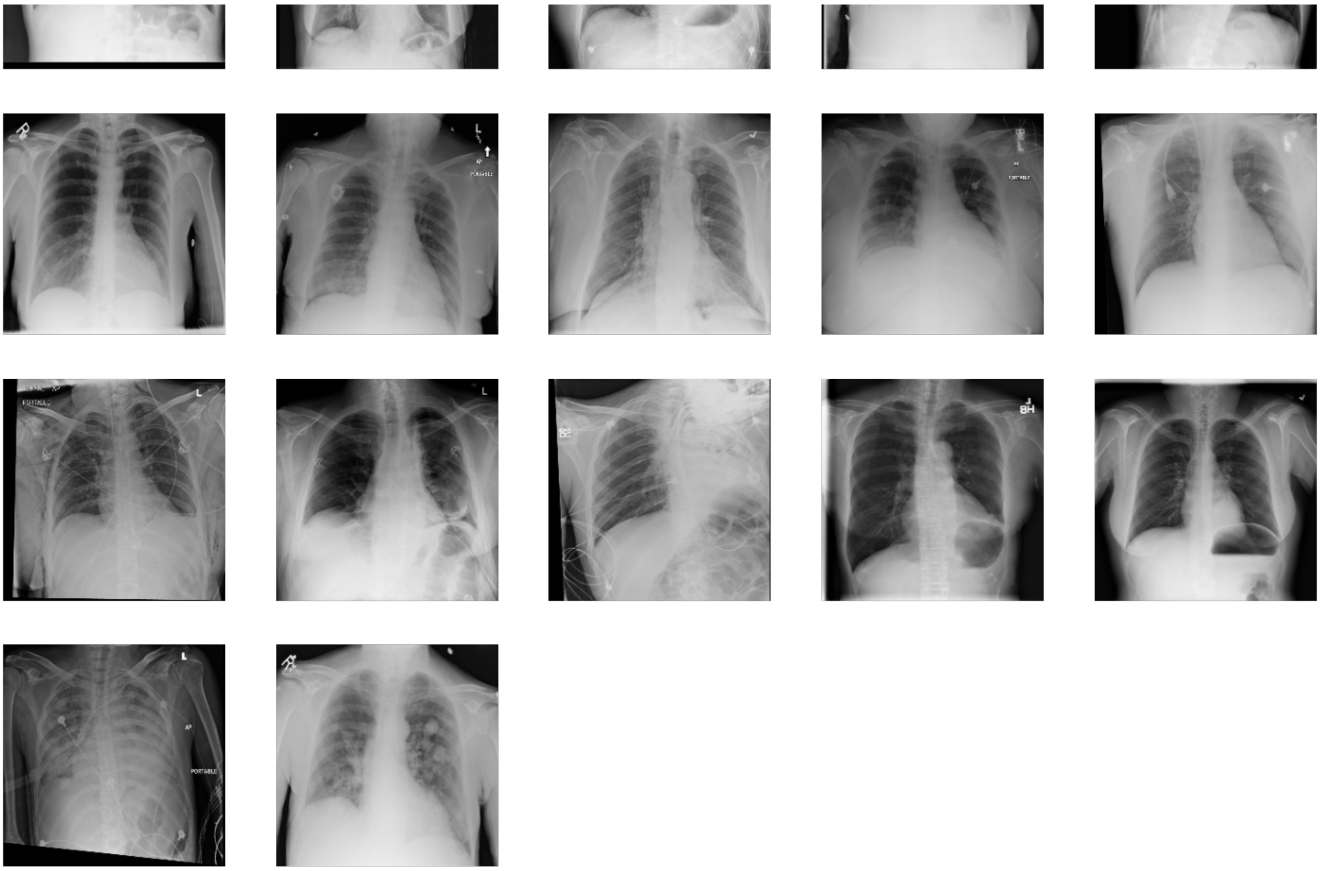
In [68]:

```
wrong_pred=[]
for i,j in tqdm(enumerate(val)):  #As the position of the wrongly predicted data points are in the count array we shall use that array to get the image
  if i+1 in count:
    wrong_pred.append(j)
```

In [76]:

```
plt.figure(figsize=(25,25))
                      #plotting images which are wrongly predicted
count=0
for i in tqdm(wrong_pred):
  ax = plt.subplot(5,5,count+1)
  count=count+1
  plt.imshow(i)
  plt.axis("off")
  plt.suptitle("Wrongly Classified X-rays")
```





Observations:

1. The cross validations models perform decently but do not perform as well as the first model using 80-20 split
2. From the images we can see that most of the wrongly classified points have issues .
3. Few X-ray images have only a single lung . Few of them are have the backgrounds to be very bright .
4. We have trained the model 2000+ images less than the first model and yet it performed decently .
5. More the the training data the better the model performs . The above model can be improved if more number of images are fed to the model for training
6. Both the models(with cross validation and without cross validation) have performed really well but the model trained without cross validation outperforms the other because of the number of images the model has been trained on.