

```
In [1]: #!unzip dank_data-master.zip  
#!pip install tensorflow_addons
```

```
In [2]: import glob  
import pandas as pd  
import warnings  
import logging  
warnings.filterwarnings("ignore")  
from tensorflow.keras.preprocessing.image import ImageDataGenerator  
from tensorflow.keras.layers import Dense, Input, Conv2D, MaxPool2D, Activation, Dropout, Flatten  
from tensorflow.keras.models import Model  
import tensorflow as tf  
import tensorflow_addons as tfa  
from tensorflow.keras.callbacks import LearningRateScheduler  
from tensorflow.keras.callbacks import ReduceLROnPlateau  
from tensorflow.keras.callbacks import ModelCheckpoint  
from tensorflow.keras.callbacks import EarlyStopping  
from sklearn.metrics import confusion_matrix, accuracy_score, f1_score  
from tensorflow.keras.applications.vgg19 import preprocess_input  
import seaborn as sns  
import numpy as np  
import matplotlib.pyplot as plt
```

```
In [3]: training='/content/dank_data-master/data/training/*'  
test='/content/dank_data-master/data/test/*'  
validation='/content/dank_data-master/data/validation/*'
```

```
In [4]: training = glob.glob(training)  
test = glob.glob(test)  
validation = glob.glob(validation)
```

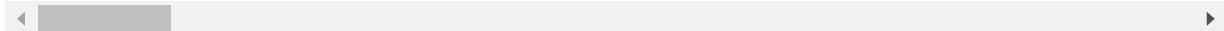
```
In [5]: final_dank=pd.read_csv('/content/dank_data-master/data/final_dank.csv')  
train_labels = [fn.split('/')[-1].split('.')[0].strip() for fn in training]  
validation_labels = [fn.split('/')[-1].split('.')[0].strip() for fn in validation]  
test_labels = [fn.split('/')[-1].split('.')[0].strip() for fn in test]
```

```
In [6]: for labels in train_labels:
    if labels==train_labels[0]:
        train_data =final_dank[final_dank['id']==labels]
    else :
        train_data =train_data.append(final_dank[final_dank['id']==labels],sort=False)
for labels in validation_labels:
    if labels==validation_labels[0]:
        val_data =final_dank[final_dank['id']==labels]
    else :
        val_data =val_data.append(final_dank[final_dank['id']==labels],sort=False)
for labels in test_labels:
    if labels==test_labels[0]:
        test_data =final_dank[final_dank['id']==labels]
    else :
        test_data =test_data.append(final_dank[final_dank['id']==labels],sort=False)
print(train_data.shape)
print(test_data.shape)
print(val_data.shape)
train_data.head(5)
```

(3405, 68)
(1719, 68)
(1688, 68)

Out[6]:

	Unnamed: 0	level_0	index	author	awards	processed_words	created_utc
53606	96606	1118.0	32771.0	SwiftScout4	[]	['dowk']	1.584914e+09
35469	61068	63931.0	63931.0	Captain_Trisl	[]	['boy', 'hang', 'quarantine', 'orona', 'extra']	1.584168e+09
11453	25253	26477.0	26477.0	Kenmoops	[]	['vehe', 'believ', 'lie', 'girl']	1.584383e+09
53276	96125	621.0	32274.0	fantastich_freidrich	[]	['human', 'come', 'futuretim', 'travel', 'trap...']	1.584917e+09
65568	129029	4832.0	66485.0	YashSSJB1	[]	['centr', 'attractionm', 'show', 'fulli', 'bui...']	1.584688e+09



```
In [7]: def file_extension(x):
    return x+".jpg"
train_data['id'] = train_data['id'].apply(file_extension)
val_data['id'] = val_data['id'].apply(file_extension)
test_data['id'] = test_data['id'].apply(file_extension)
```

```
In [8]: def numeric_to_string(x):
    if (x==1.0):
        return 'Not_dank'
    elif (x==0.0):
        return 'Dank'
train_data['dank_level_new'] = train_data['dank_level'].apply(numeric_to_string)
val_data['dank_level_new'] = val_data['dank_level'].apply(numeric_to_string)
test_data['dank_level_new'] = test_data['dank_level'].apply(numeric_to_string)
```

```
In [9]: logger = logging.getLogger()
logger.disabled = False
train_datagen = ImageDataGenerator(zoom_range=0.3, rotation_range=50,
                                   width_shift_range=0.2, height_shift_range=
0.2, shear_range=0.2,
                                   horizontal_flip=True, fill_mode='nearest')
train_generator = train_datagen.flow_from_dataframe(
    dataframe=train_data,
    directory="/content/dank_data-master/data/training/",
    x_col="id",
    y_col="dank_level_new",
    subset="training",
    batch_size=30,
    seed=42,
    class_mode="binary",
    target_size= (156,156))
predict_datagen = ImageDataGenerator(preprocessing_function=preprocess_input
)
train_prediction_generator = predict_datagen.flow_from_dataframe(
    dataframe=train_data,
    directory="/content/dank_data-master/data/training/",
    x_col="id",
    y_col="dank_level_new",
    batch_size=30,
    seed=42,
    class_mode="binary",
    shuffle=False,
    target_size= (156,156))
validation_prediction_generator = predict_datagen.flow_from_dataframe(
    dataframe=val_data,
    directory="/content/dank_data-master/data/validation/",
    x_col="id",
    y_col="dank_level_new",
    batch_size=30,
    seed=42,
    shuffle=False,
    class_mode="binary",
    target_size= (156,156))
test_prediction_generator = predict_datagen.flow_from_dataframe(
    dataframe=test_data,
    directory="/content/dank_data-master/data/test/",
    x_col="id",
    y_col="dank_level_new",
    batch_size=30,
    seed=42,
    shuffle=False,
    class_mode="binary",
    target_size= (156,156))
```

Found 3405 validated image filenames belonging to 2 classes.
 Found 3405 validated image filenames belonging to 2 classes.
 Found 1688 validated image filenames belonging to 2 classes.
 Found 1719 validated image filenames belonging to 2 classes.

```
In [10]: input = Input(shape=(156,156, 3,))

#Conv Layer
Conv1 = Conv2D(filters=256,kernel_size=(3,3),padding='same',data_format='channels_last',
               activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0),name='Conv1')(input)
#MaxPool Layer
Pool1 = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='same',data_format='channels_last',name='Pool1')(Conv1)

#Conv Layer
Conv2 = Conv2D(filters=128,kernel_size=(3,3),padding='same',data_format='channels_last',
               activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0),name='Conv2')(Pool1)
#MaxPool Layer
Pool2 = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='same',data_format='channels_last',name='Pool2')(Conv2)

#Conv Layer
Conv3 = Conv2D(filters=64,kernel_size=(3,3),padding='same',data_format='channels_last',
               activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0),name='Conv3')(Pool2)
#MaxPool Layer
Pool3 = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='same',data_format='channels_last',name='Pool3')(Conv3)

#Conv Layer
Conv4 = Conv2D(filters=32,kernel_size=(3,3),padding='same',data_format='channels_last',
               activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0),name='Conv4')(Pool3)
#MaxPool Layer
Pool4 = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='same',data_format='channels_last',name='Pool4')(Conv4)

Conv5 = Conv2D(filters=16,kernel_size=(3,3),padding='same',data_format='channels_last',
               activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0),name='Conv5')(Pool4)
#MaxPool Layer
Pool5 = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='same',data_format='channels_last',name='Pool5')(Conv5)

Conv6 = Conv2D(filters=8,kernel_size=(3,3),padding='same',data_format='channels_last',
               activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0),name='Conv6')(Pool5)
#MaxPool Layer
Pool6 = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='same',data_format='channels_last',name='Pool6')(Conv6)

#Flatten
flatten = Flatten(data_format='channels_last',name='Flatten')(Pool6)
```

```
#FC Layer
FC1 = Dense(units=128,activation='relu',kernel_initializer=tf.keras.initializers.glorot_normal(seed=32),name='FC1')(flatten)

#FC Layer
FC2 = Dense(units=64,activation='relu',kernel_initializer=tf.keras.initializers.glorot_normal(seed=33),name='FC2')(FC1)

#output layer
Out = Dense(units=1,activation='sigmoid',kernel_initializer=tf.keras.initializers.glorot_normal(seed=3),name='Output')(FC2)

model = Model(inputs=input, outputs=Out)
model.summary()
```

Model: "model"

Layer (type)	Output Shape	Param #
<hr/>		
input_1 (InputLayer)	[(None, 156, 156, 3)]	0
Conv1 (Conv2D)	(None, 156, 156, 256)	7168
Pool1 (MaxPooling2D)	(None, 78, 78, 256)	0
Conv2 (Conv2D)	(None, 78, 78, 128)	295040
Pool2 (MaxPooling2D)	(None, 39, 39, 128)	0
Conv3 (Conv2D)	(None, 39, 39, 64)	73792
Pool3 (MaxPooling2D)	(None, 20, 20, 64)	0
Conv4 (Conv2D)	(None, 20, 20, 32)	18464
Pool4 (MaxPooling2D)	(None, 10, 10, 32)	0
Conv5 (Conv2D)	(None, 10, 10, 16)	4624
Pool5 (MaxPooling2D)	(None, 5, 5, 16)	0
Conv6 (Conv2D)	(None, 5, 5, 8)	1160
Pool6 (MaxPooling2D)	(None, 3, 3, 8)	0
Flatten (Flatten)	(None, 72)	0
FC1 (Dense)	(None, 128)	9344
FC2 (Dense)	(None, 64)	8256
Output (Dense)	(None, 1)	65
<hr/>		
Total params: 417,913		
Trainable params: 417,913		
Non-trainable params: 0		

```
In [11]: def scheduler(epoch,lr):
    if((epoch+1)%3==0):
        lr=lr*0.95
        return lr
    else:
        return lr
```

```
In [12]: filepath="model_save/weights-{epoch:02d}-{val_accuracy:.4f}.h5"
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_accuracy', save_be-
st_only=True, mode='auto')

lrschedule = tf.keras.callbacks.LearningRateScheduler(scheduler,verbose=0.1)

#stop the training if your validation accuracy is not increased in last 2 epochs.
early_stop= EarlyStopping(monitor='val_accuracy', patience=3,verbose=1)

#If your validation accuracy at that epoch is Less than previous epoch accuracy, you have to decrease the
#Learning rate by 10%
reduce_lr = ReduceLROnPlateau(monitor='val_accuracy', factor=0.75,
                             patience=3, min_lr=0.001,verbose=1)

model.compile(
    loss='binary_crossentropy',
    optimizer=tf.keras.optimizers.RMSprop(lr=1e-5),
    metrics=[ 'accuracy',tf.keras.metrics.Precision(),tf.keras.metrics.Recall(),t
fa.metrics.F1Score(num_classes=1)]
)
```

```
In [13]: history=model.fit_generator(train_generator,steps_per_epoch=len(train_generator),epochs=50,validation_data=test_prediction_generator,validation_steps=len(test_prediction_generator),use_multiprocessing=False,workers=12,callbacks=[lrschedule,checkpoint,reduce_lr])
```

Epoch 1/50

Epoch 00001: LearningRateScheduler reducing learning rate to 9.99999974737875
2e-06.

114/114 [=====] - 118s 927ms/step - loss: 14.1985 -
accuracy: 0.4801 - precision: 0.4794 - recall: 0.6946 - f1_score: 0.6538 - va
l_loss: 11.0671 - val_accuracy: 0.4817 - val_precision: 0.4754 - val_recall:
0.6980 - val_f1_score: 0.6465

Epoch 2/50

Epoch 00002: LearningRateScheduler reducing learning rate to 9.99999974737875
2e-06.

114/114 [=====] - 113s 910ms/step - loss: 5.1444 - a
ccuracy: 0.4982 - precision: 0.4914 - recall: 0.5088 - f1_score: 0.6597 - val
_loss: 8.8184 - val_accuracy: 0.4788 - val_precision: 0.4708 - val_recall: 0.
6306 - val_f1_score: 0.6465

Epoch 3/50

Epoch 00003: LearningRateScheduler reducing learning rate to 9.49999976000981
3e-06.

114/114 [=====] - 115s 908ms/step - loss: 3.4053 - a
ccuracy: 0.5131 - precision: 0.5161 - recall: 0.5217 - f1_score: 0.6662 - val
_loss: 7.8826 - val_accuracy: 0.4764 - val_precision: 0.4706 - val_recall: 0.
6643 - val_f1_score: 0.6524

Epoch 4/50

Epoch 00004: LearningRateScheduler reducing learning rate to 9.49999957811087
4e-06.

114/114 [=====] - 116s 920ms/step - loss: 3.1539 - a
ccuracy: 0.4869 - precision: 0.4751 - recall: 0.4812 - f1_score: 0.6568 - val
_loss: 5.6958 - val_accuracy: 0.4945 - val_precision: 0.4802 - val_recall: 0.
5535 - val_f1_score: 0.6472

Epoch 5/50

Epoch 00005: LearningRateScheduler reducing learning rate to 9.49999957811087
4e-06.

114/114 [=====] - 112s 939ms/step - loss: 2.5646 - a
ccuracy: 0.5100 - precision: 0.4942 - recall: 0.4981 - f1_score: 0.6530 - val
_loss: 5.3180 - val_accuracy: 0.4904 - val_precision: 0.4787 - val_recall: 0.
6089 - val_f1_score: 0.6499

Epoch 6/50

Epoch 00006: LearningRateScheduler reducing learning rate to 9.02499959920533
e-06.

114/114 [=====] - 112s 919ms/step - loss: 2.2885 - a
ccuracy: 0.5070 - precision: 0.5089 - recall: 0.5234 - f1_score: 0.6683 - val
_loss: 4.3982 - val_accuracy: 0.4927 - val_precision: 0.4771 - val_recall: 0.
5150 - val_f1_score: 0.6502

Epoch 7/50

Epoch 00007: LearningRateScheduler reducing learning rate to 9.02499959920533
e-06.

114/114 [=====] - 112s 926ms/step - loss: 1.9571 - a
ccuracy: 0.5125 - precision: 0.4956 - recall: 0.4979 - f1_score: 0.6517 - val
_loss: 3.5913 - val_accuracy: 0.5143 - val_precision: 0.4980 - val_recall: 0.
5884 - val_f1_score: 0.6507

Epoch 8/50

Epoch 00008: LearningRateScheduler reducing learning rate to 9.02499959920533e-06.
114/114 [=====] - 112s 938ms/step - loss: 1.7794 - accuracy: 0.4980 - precision: 0.4985 - recall: 0.5191 - f1_score: 0.6670 - val_loss: 3.1786 - val_accuracy: 0.5102 - val_precision: 0.4939 - val_recall: 0.5355 - val_f1_score: 0.6512
Epoch 9/50

Epoch 00009: LearningRateScheduler reducing learning rate to 8.573749619245064e-06.
114/114 [=====] - 110s 913ms/step - loss: 1.6769 - accuracy: 0.4888 - precision: 0.4760 - recall: 0.4814 - f1_score: 0.6551 - val_loss: 2.8202 - val_accuracy: 0.4997 - val_precision: 0.4827 - val_recall: 0.4874 - val_f1_score: 0.6518
Epoch 10/50

Epoch 00010: LearningRateScheduler reducing learning rate to 8.573749255447183e-06.
114/114 [=====] - 116s 925ms/step - loss: 1.4971 - accuracy: 0.4887 - precision: 0.4720 - recall: 0.4831 - f1_score: 0.6518 - val_loss: 2.5490 - val_accuracy: 0.4991 - val_precision: 0.4803 - val_recall: 0.4404 - val_f1_score: 0.6518
Epoch 11/50

Epoch 00011: LearningRateScheduler reducing learning rate to 8.573749255447183e-06.
114/114 [=====] - 111s 931ms/step - loss: 1.3973 - accuracy: 0.4986 - precision: 0.4916 - recall: 0.4906 - f1_score: 0.6605 - val_loss: 2.2986 - val_accuracy: 0.5236 - val_precision: 0.5083 - val_recall: 0.4440 - val_f1_score: 0.6518
Epoch 12/50

Epoch 00012: LearningRateScheduler reducing learning rate to 8.145061792674824e-06.
114/114 [=====] - 111s 920ms/step - loss: 1.2985 - accuracy: 0.5125 - precision: 0.5152 - recall: 0.5309 - f1_score: 0.6703 - val_loss: 2.1980 - val_accuracy: 0.5201 - val_precision: 0.5039 - val_recall: 0.4645 - val_f1_score: 0.6518
Epoch 13/50

Epoch 00013: LearningRateScheduler reducing learning rate to 8.145061656250618e-06.
114/114 [=====] - 112s 907ms/step - loss: 1.2591 - accuracy: 0.4875 - precision: 0.4856 - recall: 0.4841 - f1_score: 0.6651 - val_loss: 2.0729 - val_accuracy: 0.5218 - val_precision: 0.5063 - val_recall: 0.4332 - val_f1_score: 0.6518
Epoch 14/50

Epoch 00014: LearningRateScheduler reducing learning rate to 8.145061656250618e-06.
114/114 [=====] - 118s 937ms/step - loss: 1.0762 - accuracy: 0.5042 - precision: 0.4953 - recall: 0.4903 - f1_score: 0.6587 - val_loss: 1.9759 - val_accuracy: 0.5201 - val_precision: 0.5043 - val_recall: 0.4236 - val_f1_score: 0.6518
Epoch 15/50

Epoch 00015: LearningRateScheduler reducing learning rate to 7.73780857343808
7e-06.

114/114 [=====] - 112s 923ms/step - loss: 1.1070 - accuracy: 0.4938 - precision: 0.4853 - recall: 0.4979 - f1_score: 0.6587 - val_loss: 1.9978 - val_accuracy: 0.5148 - val_precision: 0.4977 - val_recall: 0.3827 - val_f1_score: 0.6518

Epoch 16/50

Epoch 00016: LearningRateScheduler reducing learning rate to 7.73780811869073
7e-06.

114/114 [=====] - 110s 924ms/step - loss: 1.0810 - accuracy: 0.5038 - precision: 0.4948 - recall: 0.4929 - f1_score: 0.6589 - val_loss: 1.9475 - val_accuracy: 0.5305 - val_precision: 0.5189 - val_recall: 0.3959 - val_f1_score: 0.6518

Epoch 17/50

Epoch 00017: LearningRateScheduler reducing learning rate to 7.73780811869073
7e-06.

114/114 [=====] - 111s 918ms/step - loss: 0.9915 - accuracy: 0.5249 - precision: 0.5258 - recall: 0.5320 - f1_score: 0.6677 - val_loss: 1.8158 - val_accuracy: 0.5282 - val_precision: 0.5155 - val_recall: 0.4007 - val_f1_score: 0.6518

Epoch 18/50

Epoch 00018: LearningRateScheduler reducing learning rate to 7.3509177127562e
-06.

114/114 [=====] - 112s 924ms/step - loss: 0.9850 - accuracy: 0.4893 - precision: 0.4751 - recall: 0.4792 - f1_score: 0.6542 - val_loss: 2.3560 - val_accuracy: 0.5044 - val_precision: 0.4634 - val_recall: 0.1600 - val_f1_score: 0.6518

Epoch 19/50

Epoch 00019: LearningRateScheduler reducing learning rate to 7.3509177127562e
-06.

114/114 [=====] - 111s 918ms/step - loss: 0.9768 - accuracy: 0.4939 - precision: 0.5033 - recall: 0.4950 - f1_score: 0.6743 - val_loss: 1.7811 - val_accuracy: 0.5265 - val_precision: 0.5156 - val_recall: 0.3381 - val_f1_score: 0.6518

Epoch 20/50

Epoch 00020: LearningRateScheduler reducing learning rate to 7.3509177127562e
-06.

114/114 [=====] - 111s 918ms/step - loss: 0.9210 - accuracy: 0.4985 - precision: 0.4911 - recall: 0.5082 - f1_score: 0.6595 - val_loss: 1.7278 - val_accuracy: 0.5271 - val_precision: 0.5169 - val_recall: 0.3321 - val_f1_score: 0.6518

Epoch 21/50

Epoch 00021: LearningRateScheduler reducing learning rate to 6.98337182711838
9e-06.

114/114 [=====] - 110s 920ms/step - loss: 0.9312 - accuracy: 0.4984 - precision: 0.5125 - recall: 0.5281 - f1_score: 0.6801 - val_loss: 1.8029 - val_accuracy: 0.5323 - val_precision: 0.5337 - val_recall: 0.2575 - val_f1_score: 0.6518

Epoch 22/50

Epoch 00022: LearningRateScheduler reducing learning rate to 6.98337180438102

2e-06.
114/114 [=====] - 116s 931ms/step - loss: 0.8903 - accuracy: 0.4879 - precision: 0.4796 - recall: 0.4460 - f1_score: 0.6603 - val_loss: 1.6707 - val_accuracy: 0.5230 - val_precision: 0.5105 - val_recall: 0.3225 - val_f1_score: 0.6518
Epoch 23/50

Epoch 00023: LearningRateScheduler reducing learning rate to 6.98337180438102 2e-06.
114/114 [=====] - 111s 925ms/step - loss: 0.8753 - accuracy: 0.5099 - precision: 0.5167 - recall: 0.5104 - f1_score: 0.6722 - val_loss: 1.6302 - val_accuracy: 0.5253 - val_precision: 0.5149 - val_recall: 0.3117 - val_f1_score: 0.6518
Epoch 24/50

Epoch 00024: LearningRateScheduler reducing learning rate to 6.63420321416197 04e-06.
114/114 [=====] - 110s 912ms/step - loss: 0.8474 - accuracy: 0.5227 - precision: 0.5065 - recall: 0.5066 - f1_score: 0.6517 - val_loss: 1.6507 - val_accuracy: 0.5334 - val_precision: 0.5283 - val_recall: 0.3261 - val_f1_score: 0.6518
Epoch 25/50

Epoch 00025: LearningRateScheduler reducing learning rate to 6.63420314594986 85e-06.
114/114 [=====] - 110s 921ms/step - loss: 0.8458 - accuracy: 0.5043 - precision: 0.4892 - recall: 0.4864 - f1_score: 0.6533 - val_loss: 1.7054 - val_accuracy: 0.5236 - val_precision: 0.5140 - val_recall: 0.2659 - val_f1_score: 0.6518
Epoch 26/50

Epoch 00026: LearningRateScheduler reducing learning rate to 6.63420314594986 85e-06.
114/114 [=====] - 118s 941ms/step - loss: 0.8278 - accuracy: 0.5327 - precision: 0.5204 - recall: 0.5092 - f1_score: 0.6552 - val_loss: 1.5590 - val_accuracy: 0.5416 - val_precision: 0.5468 - val_recall: 0.3020 - val_f1_score: 0.6518
Epoch 27/50

Epoch 00027: LearningRateScheduler reducing learning rate to 6.30249298865237 4e-06.
114/114 [=====] - 110s 913ms/step - loss: 0.8332 - accuracy: 0.5048 - precision: 0.5022 - recall: 0.4839 - f1_score: 0.6644 - val_loss: 1.4928 - val_accuracy: 0.5317 - val_precision: 0.5252 - val_recall: 0.3261 - val_f1_score: 0.6518
Epoch 28/50

Epoch 00028: LearningRateScheduler reducing learning rate to 6.30249314781394 8e-06.
114/114 [=====] - 112s 922ms/step - loss: 0.8448 - accuracy: 0.5160 - precision: 0.5039 - recall: 0.4836 - f1_score: 0.6562 - val_loss: 1.5360 - val_accuracy: 0.5358 - val_precision: 0.5347 - val_recall: 0.3057 - val_f1_score: 0.6518
Epoch 29/50

Epoch 00029: LearningRateScheduler reducing learning rate to 6.30249314781394 8e-06.

```
114/114 [=====] - 111s 913ms/step - loss: 0.7990 - accuracy: 0.5085 - precision: 0.5010 - recall: 0.5131 - f1_score: 0.6603 - val_loss: 1.4287 - val_accuracy: 0.5457 - val_precision: 0.5465 - val_recall: 0.3538 - val_f1_score: 0.6518
Epoch 30/50

Epoch 00030: LearningRateScheduler reducing learning rate to 5.98736849042325e-06.
114/114 [=====] - 110s 914ms/step - loss: 0.8328 - accuracy: 0.4846 - precision: 0.4717 - recall: 0.4870 - f1_score: 0.6542 - val_loss: 1.4598 - val_accuracy: 0.5282 - val_precision: 0.5202 - val_recall: 0.3105 - val_f1_score: 0.6518
Epoch 31/50

Epoch 00031: LearningRateScheduler reducing learning rate to 5.98736869505955e-06.
114/114 [=====] - 118s 941ms/step - loss: 0.7964 - accuracy: 0.5160 - precision: 0.5126 - recall: 0.5101 - f1_score: 0.6631 - val_loss: 1.4601 - val_accuracy: 0.5282 - val_precision: 0.5200 - val_recall: 0.3129 - val_f1_score: 0.6518
Epoch 32/50

Epoch 00032: LearningRateScheduler reducing learning rate to 5.98736869505955e-06.
114/114 [=====] - 112s 926ms/step - loss: 0.7801 - accuracy: 0.5123 - precision: 0.4985 - recall: 0.5081 - f1_score: 0.6544 - val_loss: 1.6609 - val_accuracy: 0.5369 - val_precision: 0.5464 - val_recall: 0.2479 - val_f1_score: 0.6518
Epoch 33/50

Epoch 00033: LearningRateScheduler reducing learning rate to 5.68800026030658e-06.
114/114 [=====] - 110s 908ms/step - loss: 0.8088 - accuracy: 0.4950 - precision: 0.4922 - recall: 0.4962 - f1_score: 0.6631 - val_loss: 1.5810 - val_accuracy: 0.5369 - val_precision: 0.5422 - val_recall: 0.2708 - val_f1_score: 0.6518
Epoch 34/50

Epoch 00034: LearningRateScheduler reducing learning rate to 5.68800032851868e-06.
114/114 [=====] - 110s 912ms/step - loss: 0.7806 - accuracy: 0.5051 - precision: 0.4893 - recall: 0.4668 - f1_score: 0.6534 - val_loss: 1.6962 - val_accuracy: 0.5369 - val_precision: 0.5467 - val_recall: 0.2467 - val_f1_score: 0.6518
Epoch 35/50

Epoch 00035: LearningRateScheduler reducing learning rate to 5.68800032851868e-06.
114/114 [=====] - 110s 917ms/step - loss: 0.7821 - accuracy: 0.5019 - precision: 0.4812 - recall: 0.4792 - f1_score: 0.6491 - val_loss: 1.6931 - val_accuracy: 0.5265 - val_precision: 0.5219 - val_recall: 0.2443 - val_f1_score: 0.6518
Epoch 36/50

Epoch 00036: LearningRateScheduler reducing learning rate to 5.403600312092749e-06.
114/114 [=====] - 110s 910ms/step - loss: 0.7895 - a
```

```
accuracy: 0.5037 - precision: 0.4972 - recall: 0.5368 - f1_score: 0.6619 - val_loss: 1.8062 - val_accuracy: 0.5276 - val_precision: 0.5262 - val_recall: 0.2298 - val_f1_score: 0.6518
Epoch 37/50

Epoch 00037: LearningRateScheduler reducing learning rate to 5.40360042577958e-06.
114/114 [=====] - 108s 908ms/step - loss: 0.7688 - accuracy: 0.5248 - precision: 0.5131 - recall: 0.5425 - f1_score: 0.6565 - val_loss: 1.8262 - val_accuracy: 0.5212 - val_precision: 0.5121 - val_recall: 0.2034 - val_f1_score: 0.6518
Epoch 38/50

Epoch 00038: LearningRateScheduler reducing learning rate to 5.40360042577958e-06.
114/114 [=====] - 110s 904ms/step - loss: 0.7870 - accuracy: 0.4821 - precision: 0.4580 - recall: 0.4362 - f1_score: 0.6498 - val_loss: 1.6471 - val_accuracy: 0.5381 - val_precision: 0.5466 - val_recall: 0.2611 - val_f1_score: 0.6518
Epoch 39/50

Epoch 00039: LearningRateScheduler reducing learning rate to 5.13342040449060e-06.
114/114 [=====] - 110s 903ms/step - loss: 0.7679 - accuracy: 0.4989 - precision: 0.4975 - recall: 0.5091 - f1_score: 0.6651 - val_loss: 1.5560 - val_accuracy: 0.5375 - val_precision: 0.5511 - val_recall: 0.2335 - val_f1_score: 0.6518
Epoch 40/50

Epoch 00040: LearningRateScheduler reducing learning rate to 5.13342047270271e-06.
114/114 [=====] - 110s 906ms/step - loss: 0.7786 - accuracy: 0.5057 - precision: 0.5168 - recall: 0.4976 - f1_score: 0.6765 - val_loss: 1.6059 - val_accuracy: 0.5294 - val_precision: 0.5314 - val_recall: 0.2238 - val_f1_score: 0.6518
Epoch 41/50

Epoch 00041: LearningRateScheduler reducing learning rate to 5.13342047270271e-06.
114/114 [=====] - 110s 908ms/step - loss: 0.7755 - accuracy: 0.4909 - precision: 0.4872 - recall: 0.4654 - f1_score: 0.6652 - val_loss: 1.5963 - val_accuracy: 0.5247 - val_precision: 0.5185 - val_recall: 0.2359 - val_f1_score: 0.6518
Epoch 42/50

Epoch 00042: LearningRateScheduler reducing learning rate to 4.876749449067574e-06.
114/114 [=====] - 110s 907ms/step - loss: 0.7811 - accuracy: 0.5079 - precision: 0.5055 - recall: 0.5350 - f1_score: 0.6647 - val_loss: 1.6568 - val_accuracy: 0.5271 - val_precision: 0.5269 - val_recall: 0.2118 - val_f1_score: 0.6518
Epoch 43/50

Epoch 00043: LearningRateScheduler reducing learning rate to 4.8767492444312666e-06.
114/114 [=====] - 109s 917ms/step - loss: 0.7763 - accuracy: 0.5024 - precision: 0.4919 - recall: 0.4786 - f1_score: 0.6577 - val
```

```
_loss: 1.4588 - val_accuracy: 0.5259 - val_precision: 0.5193 - val_recall: 0.  
2587 - val_f1_score: 0.6518  
Epoch 44/50  
  
Epoch 00044: LearningRateScheduler reducing learning rate to 4.87674924443126  
66e-06.  
114/114 [=====] - 109s 908ms/step - loss: 0.7670 - a  
ccuracy: 0.5030 - precision: 0.4948 - recall: 0.5130 - f1_score: 0.6588 - val  
_loss: 1.5834 - val_accuracy: 0.5236 - val_precision: 0.5158 - val_recall: 0.  
2359 - val_f1_score: 0.6518  
Epoch 45/50  
  
Epoch 00045: LearningRateScheduler reducing learning rate to 4.63291178220970  
3e-06.  
114/114 [=====] - 112s 937ms/step - loss: 0.7484 - a  
ccuracy: 0.5073 - precision: 0.4985 - recall: 0.5338 - f1_score: 0.6583 - val  
_loss: 1.5230 - val_accuracy: 0.5305 - val_precision: 0.5297 - val_recall: 0.  
2575 - val_f1_score: 0.6518  
Epoch 46/50  
  
Epoch 00046: LearningRateScheduler reducing learning rate to 4.63291189589654  
1e-06.  
114/114 [=====] - 113s 936ms/step - loss: 0.7637 - a  
ccuracy: 0.4950 - precision: 0.4897 - recall: 0.5113 - f1_score: 0.6617 - val  
_loss: 1.4453 - val_accuracy: 0.5282 - val_precision: 0.5226 - val_recall: 0.  
2780 - val_f1_score: 0.6518  
Epoch 47/50  
  
Epoch 00047: LearningRateScheduler reducing learning rate to 4.63291189589654  
1e-06.  
114/114 [=====] - 112s 935ms/step - loss: 0.7601 - a  
ccuracy: 0.5235 - precision: 0.5092 - recall: 0.5101 - f1_score: 0.6533 - val  
_loss: 1.4725 - val_accuracy: 0.5271 - val_precision: 0.5220 - val_recall: 0.  
2575 - val_f1_score: 0.6518  
Epoch 48/50  
  
Epoch 00048: LearningRateScheduler reducing learning rate to 4.40126630110171  
36e-06.  
114/114 [=====] - 110s 914ms/step - loss: 0.7560 - a  
ccuracy: 0.5051 - precision: 0.5089 - recall: 0.5450 - f1_score: 0.6713 - val  
_loss: 1.5819 - val_accuracy: 0.5259 - val_precision: 0.5222 - val_recall: 0.  
2262 - val_f1_score: 0.6518  
Epoch 49/50  
  
Epoch 00049: LearningRateScheduler reducing learning rate to 4.40126632383908  
15e-06.  
114/114 [=====] - 116s 931ms/step - loss: 0.7457 - a  
ccuracy: 0.5192 - precision: 0.5114 - recall: 0.5307 - f1_score: 0.6597 - val  
_loss: 1.3054 - val_accuracy: 0.5329 - val_precision: 0.5289 - val_recall: 0.  
3081 - val_f1_score: 0.6518  
Epoch 50/50  
  
Epoch 00050: LearningRateScheduler reducing learning rate to 4.40126632383908  
15e-06.  
114/114 [=====] - 110s 903ms/step - loss: 0.7377 - a  
ccuracy: 0.5138 - precision: 0.5020 - recall: 0.5051 - f1_score: 0.6560 - val
```

```
_loss: 1.3106 - val_accuracy: 0.5259 - val_precision: 0.5167 - val_recall: 0.
2972 - val_f1_score: 0.6518
```

In [16]:

```
model_checkpoint = Model(inputs=input, outputs=Out)
model_checkpoint.load_weights('/content/model_save/weights-07-0.5143.h5')
model_checkpoint.save('simeconv.h5')
new_model = tf.keras.models.load_model('simeconv.h5')
```

WARNING:tensorflow:No training configuration found in the save file, so the model was *not* compiled. Compile it manually.

In [18]:

```
test_prediction=new_model.predict_generator(test_prediction_generator,steps=len(test_prediction_generator),workers=12)
test_prediction=((test_prediction > 0.5)+0).ravel()
print(test_prediction.shape)
y_test =tf.keras.utils.to_categorical(test_data['dank_level'].values,2)
y_test=np.argmax(y_test,axis=-1)
y_test.shape
```

(1719,)

Out[18]: (1719,)

In [19]:

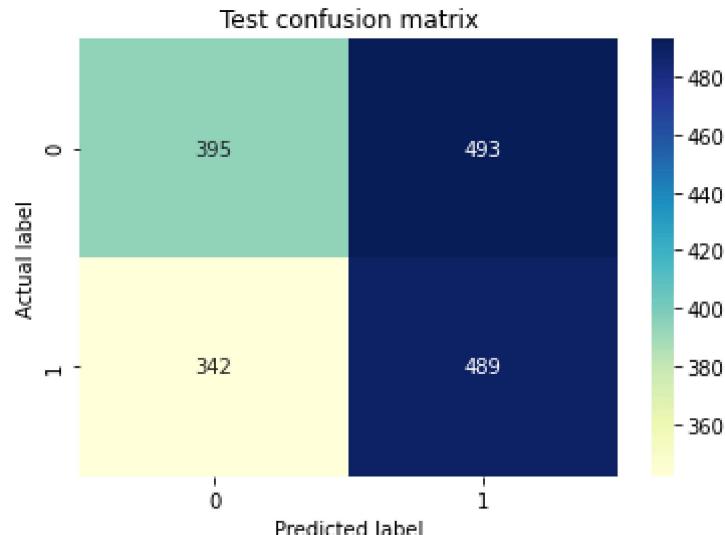
```
accuracy=accuracy_score(y_test,test_prediction)
print("Test accuracy_score",accuracy)
f1_test_score=f1_score(y_test,test_prediction)
print("Test F1_score",f1_test_score)
print("Test confusion matrix")
cnf_matrix2=confusion_matrix(y_test,test_prediction)
p = sns.heatmap(pd.DataFrame(cnf_matrix2), annot=True, cmap="YlGnBu" ,fmt='g')
plt.title('Test confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

Test accuracy_score 0.5142524723676556

Test F1_score 0.5394373965802538

Test confusion matrix

Out[19]: Text(0.5, 15.0, 'Predicted label')



```
In [20]: file = '/content/model_1.png'  
tf.keras.utils.plot_model(model,to_file=file, show_shapes=True)
```

Out[20]:

