

Neural Network Assignment (22AI509T)

Name: Tanmay Rathod

Enrollment: 23MAI007

Aim: Implementation of Transfer Learning with pre-trained Model for Pulmonary Image Classification

Reference

Dataset: [Fruits Dataset](#)

Git-Hub Code: [Assignment Code](#)

Subject Git-Hub: [22AI511T Subject Repository](#)

```
In [ ]: import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from tensorflow import keras

os.listdir("/content/drive/MyDrive/chest_xray/chest_xray")
```

```
Out[ ]: ['chest_xray', '__MACOSX', 'val', 'test', 'train']
```

```
In [ ]: len(os.listdir("/content/drive/MyDrive/chest_xray/chest_xray/train/PNEUMONIA"))
```

```
Out[ ]: 3875
```

The dataset is divided into three sets: 1) Train set 2) Validation set and 3) Test set.

Data Visualization

```
In [ ]: train_dir = "/content/drive/MyDrive/chest_xray/chest_xray/train"
test_dir = "/content/drive/MyDrive/chest_xray/chest_xray/test"
val_dir = "/content/drive/MyDrive/chest_xray/chest_xray/val"

print("Train set:\n====")
num_pneumonia = len(os.listdir(os.path.join(train_dir, 'PNEUMONIA'))))
num_normal = len(os.listdir(os.path.join(train_dir, 'NORMAL'))))
print(f"PNEUMONIA={num_pneumonia}")
print(f"NORMAL={num_normal}")

print("Test set:\n====")
print(f"PNEUMONIA={len(os.listdir(os.path.join(test_dir, 'PNEUMONIA'))))}")
```

```

print(f"NORMAL={len(os.listdir(os.path.join(test_dir, 'NORMAL')))}")\n\n
print("Validation set:\n=====\n")
print(f"PNEUMONIA={len(os.listdir(os.path.join(val_dir, 'PNEUMONIA')))}")
print(f"NORMAL={len(os.listdir(os.path.join(val_dir, 'NORMAL')))}")\n\n
pneumonia = os.listdir("/content/drive/MyDrive/chest_xray/chest_xray/train/PNEUMONIA")
pneumonia_dir = "/content/drive/MyDrive/chest_xray/chest_xray/train/PNEUMONIA"\n\n
plt.figure(figsize=(20, 10))\n\n
for i in range(9):\n    plt.subplot(3, 3, i + 1)\n    img = plt.imread(os.path.join(pneumonia_dir, pneumonia[i]))\n    plt.imshow(img, cmap='gray')\n    plt.axis('off')\n\nplt.tight_layout()

```

Train set:

PNEUMONIA=3875

NORMAL=1341

Test set:

PNEUMONIA=390

NORMAL=234

Validation set:

PNEUMONIA=8

NORMAL=8

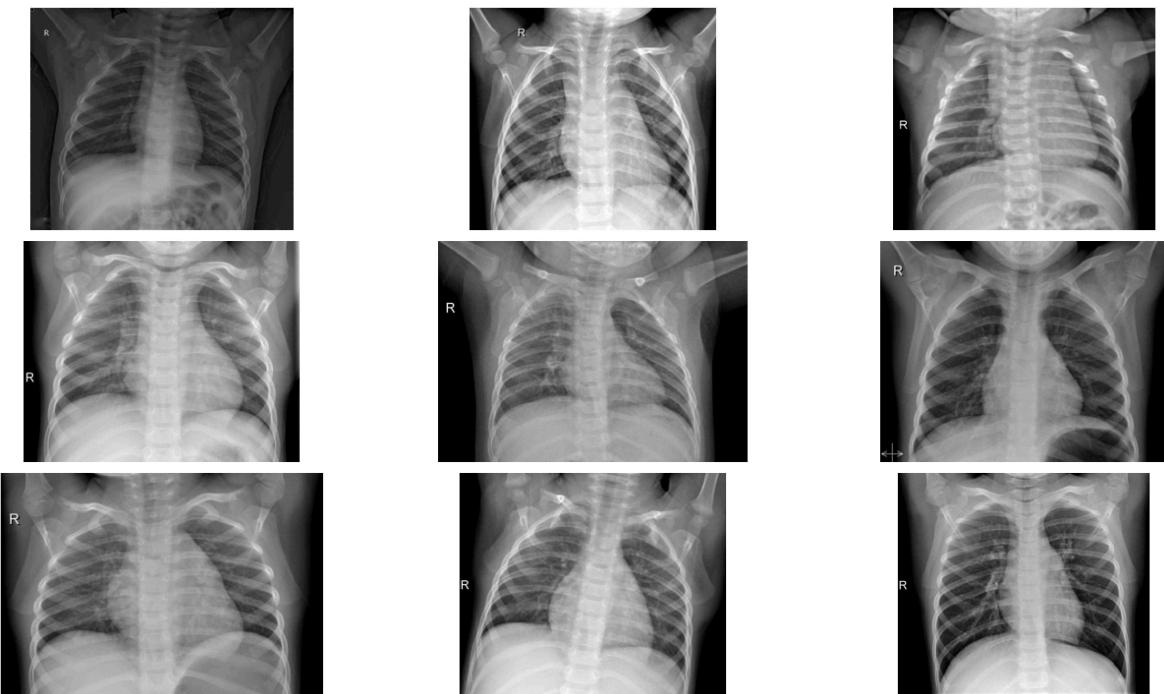


```

In [ ]: normal = os.listdir("/content/drive/MyDrive/chest_xray/chest_xray/train/NORMAL")
normal_dir = "/content/drive/MyDrive/chest_xray/chest_xray/train/NORMAL"\n\n
plt.figure(figsize=(20, 10))\n\n
for i in range(9):\n    plt.subplot(3, 3, i + 1)\n    img = plt.imread(os.path.join(normal_dir, normal[i]))\n    plt.imshow(img, cmap='gray')

```

```
plt.axis('off')  
plt.tight_layout()
```

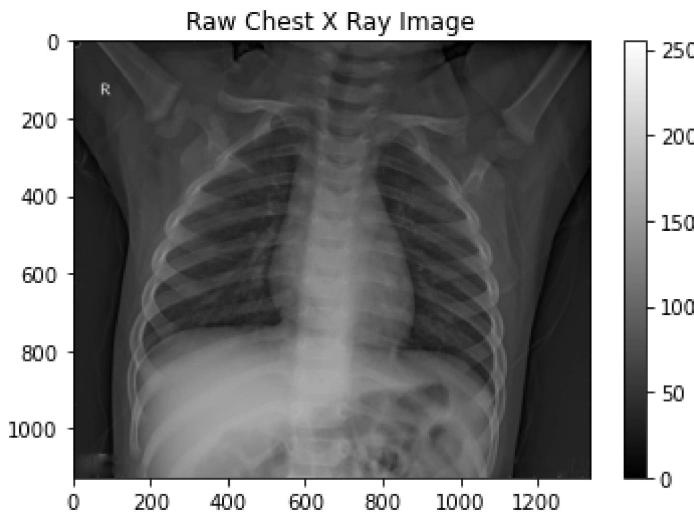


```
In [ ]: normal_img = os.listdir("/content/drive/MyDrive/chest_xray/chest_xray/train/NORMAL")  
normal_dir = "/content/drive/MyDrive/chest_xray/chest_xray/train/NORMAL"  
sample_img = plt.imread(os.path.join(normal_dir, normal_img))  
plt.imshow(sample_img, cmap='gray')  
plt.colorbar()  
plt.title('Raw Chest X Ray Image')  
  
print(f"The dimensions of the image are {sample_img.shape[0]} pixels width and {sample_img.shape[1]} pixels height, one single color channel.  
The maximum pixel value is {sample_img.max():.4f} and the minimum is {sample_img.min():.4f}  
The mean value of the pixels is {sample_img.mean():.4f} and the standard deviation is {sample_img.std():.4f}")
```

The dimensions of the image are 1128 pixels width and 1336 pixels height, one single color channel.

The maximum pixel value is 255.0000 and the minimum is 0.0000

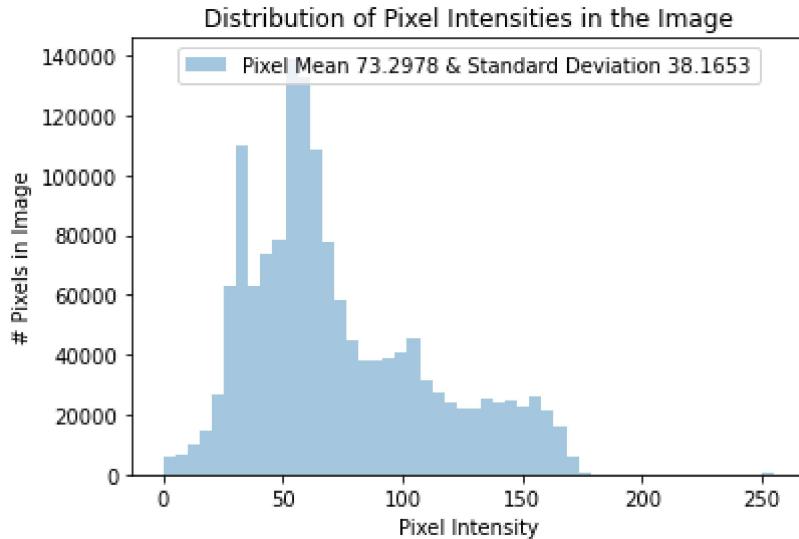
The mean value of the pixels is 73.2978 and the standard deviation is 38.1653



Investigate pixel value distribution

```
In [ ]: sns.distplot(sample_img.ravel(),
                     label=f"Pixel Mean {np.mean(sample_img):.4f} & Standard Deviation {n
                     plt.legend(loc='upper center')
                     plt.title('Distribution of Pixel Intensities in the Image')
                     plt.xlabel('Pixel Intensity')
                     plt.ylabel('# Pixels in Image')
```

```
Out[ ]: Text(0, 0.5, '# Pixels in Image')
```



2. Image Preprocessing

Before training, we'll first modify your images to be better suited for training a convolutional neural network. For this task we'll use the Keras ImageDataGenerator function to perform data preprocessing and data augmentation.

This class also provides support for basic data augmentation such as random horizontal flipping of images. We also use the generator to transform the values in each batch so that their mean is 0 and their standard deviation is 1 (this will facilitate model training by standardizing the input distribution). The generator also converts our single channel X-ray images (gray-scale) to a three-channel format by repeating the values in the image across all channels (we will want this because the pre-trained model that we'll use requires three-channel inputs).

```
In [ ]: from keras.preprocessing.image import ImageDataGenerator

image_generator = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    samplewise_center=True,
    samplewise_std_normalization=True
)
```

Build a separate generator for validation and test sets

Now we need to build a new generator for validation and testing data.

Why can't use the same generator as for the training data?

Look back at the generator we wrote for the training data.

It normalizes each image per batch, meaning that it uses batch statistics. We should not do this with the test and validation data, since in a real life scenario we don't process incoming images a batch at a time (we process one image at a time). Knowing the average per batch of test data would effectively give our model an advantage (The model should not have any information about the test data). What we need to do is to normalize incoming test data using the statistics computed from the training set.

```
In [ ]: train = image_generator.flow_from_directory(train_dir,
                                                 batch_size=8,
                                                 shuffle=True,
                                                 class_mode='binary',
                                                 target_size=(180, 180))

validation = image_generator.flow_from_directory(val_dir,
                                                 batch_size=1,
                                                 shuffle=False,
                                                 class_mode='binary',
                                                 target_size=(180, 180))

test = image_generator.flow_from_directory(test_dir,
                                           batch_size=1,
                                           shuffle=False,
                                           class_mode='binary',
                                           target_size=(180, 180))
```

Found 5216 images belonging to 2 classes.

Found 16 images belonging to 2 classes.

Found 624 images belonging to 2 classes.

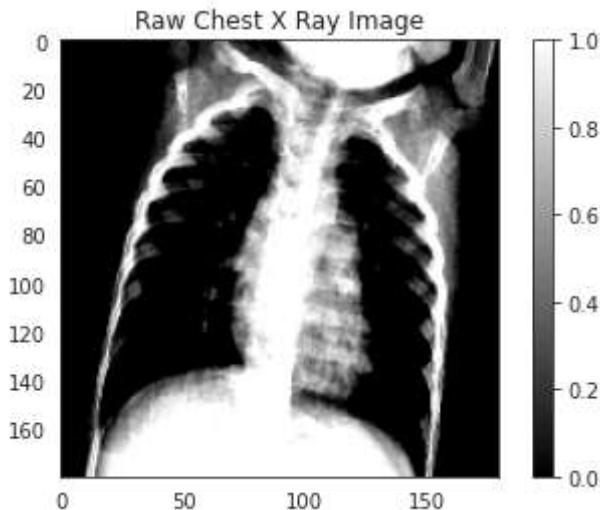
```
In [ ]: sns.set_style('white')
generated_image, label = train.__getitem__(0)
plt.imshow(generated_image[0], cmap='gray')
plt.colorbar()
plt.title('Raw Chest X Ray Image')

print(f"The dimensions of the image are {generated_image.shape[1]} pixels width")
print(f"The maximum pixel value is {generated_image.max():.4f} and the minimum is {generated_image.min():.4f}")
print(f"The mean value of the pixels is {generated_image.mean():.4f} and the standard deviation is {generated_image.std():.4f}")
```

The dimensions of the image are 180 pixels width and 180 pixels height, one single color channel.

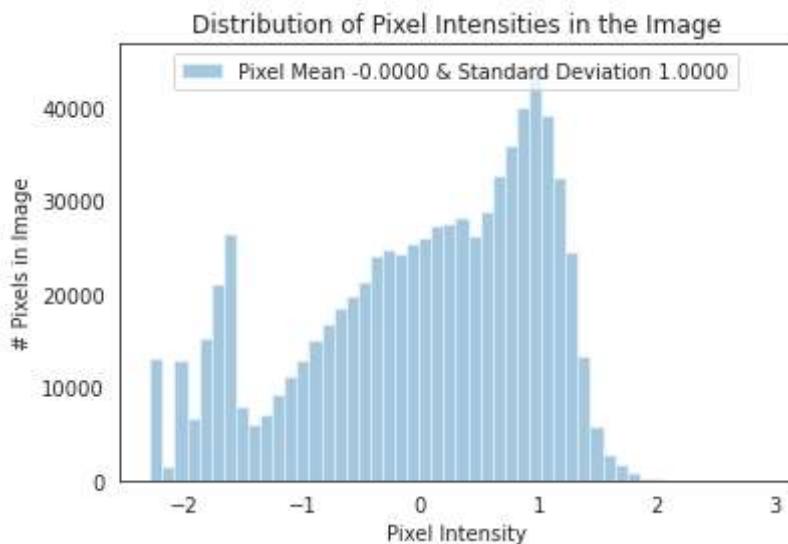
The maximum pixel value is 2.8855 and the minimum is -2.2819

The mean value of the pixels is -0.0000 and the standard deviation is 1.0000



```
In [ ]: sns.distplot(generated_image.ravel(),
                    label=f"Pixel Mean {np.mean(generated_image):.4f} & Standard Deviation {np.std(generated_image):.4f}")
                    plt.legend(loc='upper center')
                    plt.title('Distribution of Pixel Intensities in the Image')
                    plt.xlabel('Pixel Intensity')
                    plt.ylabel('# Pixels in Image')
```

Out[]: Text(0, 0.5, '# Pixels in Image')



```
In [ ]: # Class weights

weight_for_0 = num_pneumonia / (num_normal + num_pneumonia)
weight_for_1 = num_normal / (num_normal + num_pneumonia)

class_weight = {0: weight_for_0, 1: weight_for_1}

print(f"Weight for class 0: {weight_for_0:.2f}")
print(f"Weight for class 1: {weight_for_1:.2f}")
```

Weight for class 0: 0.74
Weight for class 1: 0.26

```
In [ ]: from keras.models import Sequential
        from keras.layers import Dense, Conv2D, MaxPool2D, Dropout, Flatten, BatchNormalization

model = Sequential()
```

```
model.add(Conv2D(filters=32, kernel_size=(3, 3), input_shape=(180, 180, 3), activation='relu'))
model.add(BatchNormalization())
model.add(Conv2D(filters=32, kernel_size=(3, 3), input_shape=(180, 180, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

In []: model.summary()

Model: "sequential"

Layer (type)	Output Shape	Param #
<hr/>		
conv2d (Conv2D)	(None, 178, 178, 32)	896
<hr/>		
batch_normalization (BatchNo	(None, 178, 178, 32)	128
<hr/>		
conv2d_1 (Conv2D)	(None, 176, 176, 32)	9248
<hr/>		
batch_normalization_1 (Batch	(None, 176, 176, 32)	128
<hr/>		
max_pooling2d (MaxPooling2D)	(None, 88, 88, 32)	0
<hr/>		
conv2d_2 (Conv2D)	(None, 86, 86, 64)	18496
<hr/>		
batch_normalization_2 (Batch	(None, 86, 86, 64)	256
<hr/>		
conv2d_3 (Conv2D)	(None, 84, 84, 64)	36928
<hr/>		
batch_normalization_3 (Batch	(None, 84, 84, 64)	256
<hr/>		
max_pooling2d_1 (MaxPooling2	(None, 42, 42, 64)	0
<hr/>		
conv2d_4 (Conv2D)	(None, 40, 40, 128)	73856
<hr/>		
batch_normalization_4 (Batch	(None, 40, 40, 128)	512
<hr/>		
conv2d_5 (Conv2D)	(None, 38, 38, 128)	147584
<hr/>		
batch_normalization_5 (Batch	(None, 38, 38, 128)	512
<hr/>		
max_pooling2d_2 (MaxPooling2	(None, 19, 19, 128)	0
<hr/>		
flatten (Flatten)	(None, 46208)	0
<hr/>		
dense (Dense)	(None, 128)	5914752
<hr/>		
dropout (Dropout)	(None, 128)	0
<hr/>		
dense_1 (Dense)	(None, 1)	129
<hr/>		
Total params: 6,203,681		
Trainable params: 6,202,785		
Non-trainable params: 896		

```
In [ ]: r = model.fit(
    train,
    epochs=10,
    validation_data=validation,
    class_weight=class_weight,
    steps_per_epoch=100,
    validation_steps=25,
)
```

```

Epoch 1/10
100/100 [=====] - 22s 221ms/step - loss: 1.0302 - accuracy: 0.8075 - val_loss: 47.6744 - val_accuracy: 0.5000
Epoch 2/10
100/100 [=====] - 20s 205ms/step - loss: 0.2636 - accuracy: 0.8275 - val_loss: 58.9051 - val_accuracy: 0.5000
Epoch 3/10
100/100 [=====] - 19s 193ms/step - loss: 0.1774 - accuracy: 0.8500 - val_loss: 11.9920 - val_accuracy: 0.5000
Epoch 4/10
100/100 [=====] - 20s 195ms/step - loss: 0.1473 - accuracy: 0.8675 - val_loss: 8.3084 - val_accuracy: 0.5000
Epoch 5/10
100/100 [=====] - 18s 185ms/step - loss: 0.1619 - accuracy: 0.8813 - val_loss: 1.1438 - val_accuracy: 0.5625
Epoch 6/10
100/100 [=====] - 19s 188ms/step - loss: 0.1226 - accuracy: 0.8800 - val_loss: 1.3685 - val_accuracy: 0.6875
Epoch 7/10
100/100 [=====] - 18s 179ms/step - loss: 0.1062 - accuracy: 0.9000 - val_loss: 0.2966 - val_accuracy: 0.8750
Epoch 8/10
100/100 [=====] - 18s 176ms/step - loss: 0.1107 - accuracy: 0.8913 - val_loss: 1.1703 - val_accuracy: 0.5625
Epoch 9/10
100/100 [=====] - 16s 164ms/step - loss: 0.1166 - accuracy: 0.9087 - val_loss: 0.7184 - val_accuracy: 0.5625
Epoch 10/10
100/100 [=====] - 17s 175ms/step - loss: 0.0977 - accuracy: 0.9050 - val_loss: 0.5407 - val_accuracy: 0.8125

```

```
In [ ]: plt.figure(figsize=(12, 8))
```

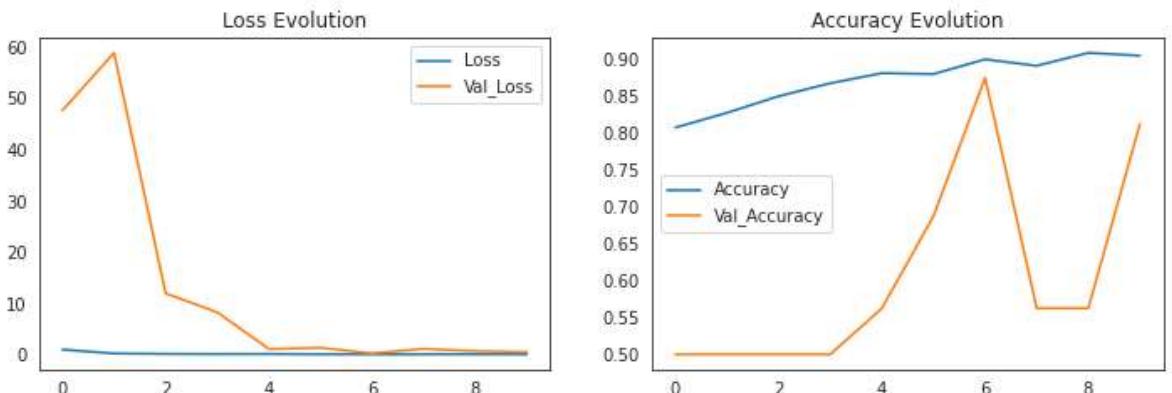
```

plt.subplot(2, 2, 1)
plt.plot(r.history['loss'], label='Loss')
plt.plot(r.history['val_loss'], label='Val_Loss')
plt.legend()
plt.title('Loss Evolution')

plt.subplot(2, 2, 2)
plt.plot(r.history['accuracy'], label='Accuracy')
plt.plot(r.history['val_accuracy'], label='Val_Accuracy')
plt.legend()
plt.title('Accuracy Evolution')

```

```
Out[ ]: Text(0.5, 1.0, 'Accuracy Evolution')
```



```
In [ ]: evaluation = model.evaluate(test)
print(f"Test Accuracy: {evaluation[1] * 100:.2f}%")

evaluation = model.evaluate(train)
print(f"Train Accuracy: {evaluation[1] * 100:.2f}%")

624/624 [=====] - 14s 23ms/step - loss: 0.4296 - accuracy: 0.8462
Test Accuracy: 84.62%
652/652 [=====] - 103s 159ms/step - loss: 0.3312 - accuracy: 0.8969
Train Accuracy: 89.69%
```

```
In [ ]: from sklearn.metrics import confusion_matrix, classification_report

pred = model.predict(test)

print(confusion_matrix(test.classes, pred > 0.5))
pd.DataFrame(classification_report(test.classes, pred > 0.5, output_dict=True))

[[181  53]
 [ 46 344]]
```

```
Out[ ]:
```

	0	1	accuracy	macro avg	weighted avg
precision	0.797357	0.866499	0.841346	0.831928	0.840571
recall	0.773504	0.882051	0.841346	0.827778	0.841346
f1-score	0.785249	0.874206	0.841346	0.829728	0.840847
support	234.000000	390.000000	0.841346	624.000000	624.000000

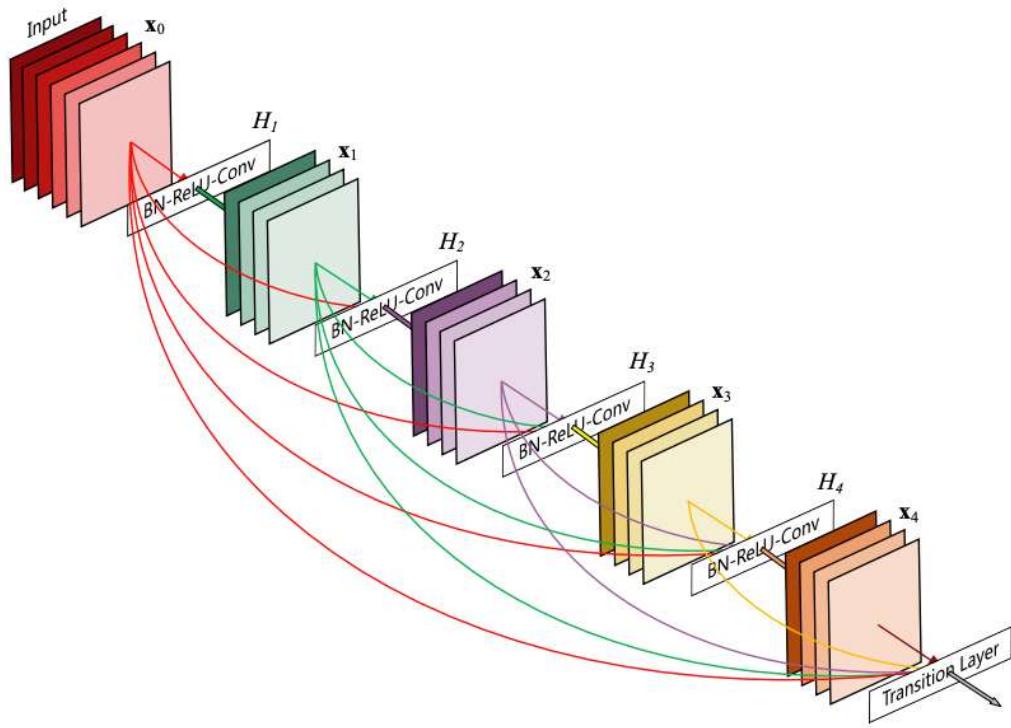
```
In [ ]: print(confusion_matrix(test.classes, pred > 0.7))
pd.DataFrame(classification_report(test.classes, pred > 0.7, output_dict=True))
```

Transfer Learning

DenseNet

Densenet is a convolutional network where each layer is connected to all other layers that are deeper in the network:

- The first layer is connected to the 2nd, 3rd, 4th etc.
- The second layer is conected to the 3rd, 4th, 5th etc.



for more information about the DenseNet Architecture visit this website :
<https://keras.io/api/applications/densenet/>

```
In [ ]: from keras.applications.densenet import DenseNet121
from keras.layers import Dense, GlobalAveragePooling2D
from keras.models import Model
from keras import backend as K

base_model = DenseNet121(input_shape=(180, 180, 3), include_top=False, weights='

base_model.summary()
```

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/densenet/densenet121_weights_tf_dim_ordering_tf_kernels_notop.h5
29089792/29084464 [=====] - 1s 0us/step
Model: "densenet121"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 180, 180, 3) 0]		
zero_padding2d (ZeroPadding2D)	(None, 186, 186, 3) 0		input_1[0][0]
conv1/conv (Conv2D)	(None, 90, 90, 64) 9408		zero_padding2d[0][0]
conv1/bn (BatchNormalization)	(None, 90, 90, 64) 256		conv1/conv[0][0]
conv1/relu (Activation)	(None, 90, 90, 64) 0		conv1/bn[0][0]
zero_padding2d_1 (ZeroPadding2D)	(None, 92, 92, 64) 0		conv1/relu[0][0]
pool1 (MaxPooling2D)	(None, 45, 45, 64) 0		zero_padding2d_1[0][0]
conv2_block1_0_bn (BatchNormali	(None, 45, 45, 64) 256		pool1[0][0]
conv2_block1_0_relu (Activation	(None, 45, 45, 64) 0		conv2_block1_0_bn[0][0]
conv2_block1_1_conv (Conv2D)	(None, 45, 45, 128) 8192		conv2_block1_0_relu[0][0]
conv2_block1_1_bn (BatchNormali	(None, 45, 45, 128) 512		conv2_block1_1_conv[0][0]
conv2_block1_1_relu (Activation	(None, 45, 45, 128) 0		conv2_block1_1_bn[0][0]
conv2_block1_2_conv (Conv2D)	(None, 45, 45, 32) 36864		conv2_block1_1_relu[0][0]
conv2_block1_concat (Concatenat	(None, 45, 45, 96) 0		pool1[0][0] conv2_block1_2_c onv[0][0]

conv2_block2_0_bn (BatchNormali (None, 45, 45, 96) 384	conv2_block1_cat[0][0]	
conv2_block2_0_relu (Activation (None, 45, 45, 96) 0	conv2_block2_0_bn[0][0]	
conv2_block2_1_conv (Conv2D) (None, 45, 45, 128) 12288	conv2_block2_0_relu[0][0]	
conv2_block2_1_bn (BatchNormali (None, 45, 45, 128) 512	conv2_block2_1_conv[0][0]	
conv2_block2_1_relu (Activation (None, 45, 45, 128) 0	conv2_block2_1_bn[0][0]	
conv2_block2_2_conv (Conv2D) (None, 45, 45, 32) 36864	conv2_block2_1_relu[0][0]	
conv2_block2_concat (Concatenat (None, 45, 45, 128) 0	conv2_block1_cat[0][0]	conv2_block2_2_conv[0][0]
conv2_block3_0_bn (BatchNormali (None, 45, 45, 128) 512	conv2_block2_concat[0][0]	conv2_block2_conv[0][0]
conv2_block3_0_relu (Activation (None, 45, 45, 128) 0	conv2_block3_0_bn[0][0]	conv2_block3_0_b[0][0]
conv2_block3_1_conv (Conv2D) (None, 45, 45, 128) 16384	conv2_block3_0_relu[0][0]	conv2_block3_0_r[0][0]
conv2_block3_1_bn (BatchNormali (None, 45, 45, 128) 512	conv2_block3_1_conv[0][0]	conv2_block3_1_c[0][0]
conv2_block3_1_relu (Activation (None, 45, 45, 128) 0	conv2_block3_1_bn[0][0]	conv2_block3_1_b[0][0]
conv2_block3_2_conv (Conv2D) (None, 45, 45, 32) 36864	conv2_block3_1_relu[0][0]	conv2_block3_1_r[0][0]
conv2_block3_concat (Concatenat (None, 45, 45, 160) 0	conv2_block2_cat[0][0]	conv2_block2_conv[0][0]
	conv2_block3_concat[0][0]	conv2_block3_2_c[0][0]

conv2_block4_0_bn (BatchNormali (None, 45, 45, 160) 640 cat[0][0]			conv2_block3_con
conv2_block4_0_relu (Activation (None, 45, 45, 160) 0 n[0][0]			conv2_block4_0_b
conv2_block4_1_conv (Conv2D) (None, 45, 45, 128) 20480 elu[0][0]			conv2_block4_0_r
conv2_block4_1_bn (BatchNormali (None, 45, 45, 128) 512 onv[0][0]			conv2_block4_1_c
conv2_block4_1_relu (Activation (None, 45, 45, 128) 0 n[0][0]			conv2_block4_1_b
conv2_block4_2_conv (Conv2D) (None, 45, 45, 32) 36864 elu[0][0]			conv2_block4_1_r
conv2_block4_concat (Concatenat (None, 45, 45, 192) 0 cat[0][0]			conv2_block3_con
			conv2_block4_2_c
conv2_block4_concat (Concatenat (None, 45, 45, 192) 0 onv[0][0]			
conv2_block5_0_bn (BatchNormali (None, 45, 45, 192) 768 cat[0][0]			conv2_block4_con
conv2_block5_0_relu (Activation (None, 45, 45, 192) 0 n[0][0]			conv2_block5_0_b
conv2_block5_1_conv (Conv2D) (None, 45, 45, 128) 24576 elu[0][0]			conv2_block5_0_r
conv2_block5_1_bn (BatchNormali (None, 45, 45, 128) 512 onv[0][0]			conv2_block5_1_c
conv2_block5_1_relu (Activation (None, 45, 45, 128) 0 n[0][0]			conv2_block5_1_b
conv2_block5_2_conv (Conv2D) (None, 45, 45, 32) 36864 elu[0][0]			conv2_block5_1_r
conv2_block5_concat (Concatenat (None, 45, 45, 224) 0 cat[0][0]			conv2_block4_con
			conv2_block5_2_c
conv2_block5_concat (Concatenat (None, 45, 45, 224) 0 onv[0][0]			

conv2_block6_0_bn (BatchNormali (None, 45, 45, 224) 896 cat[0][0]		conv2_block5_con
conv2_block6_0_relu (Activation (None, 45, 45, 224) 0 n[0][0]		conv2_block6_0_b
conv2_block6_1_conv (Conv2D) (None, 45, 45, 128) 28672 elu[0][0]		conv2_block6_0_r
conv2_block6_1_bn (BatchNormali (None, 45, 45, 128) 512 onv[0][0]		conv2_block6_1_c
conv2_block6_1_relu (Activation (None, 45, 45, 128) 0 n[0][0]		conv2_block6_1_b
conv2_block6_2_conv (Conv2D) (None, 45, 45, 32) 36864 elu[0][0]		conv2_block6_1_r
conv2_block6_concat (Concatenat (None, 45, 45, 256) 0 cat[0][0]		conv2_block5_con
		conv2_block6_2_c
onv[0][0]		
pool2_bn (BatchNormalization) (None, 45, 45, 256) 1024 cat[0][0]		conv2_block6_con
pool2_relu (Activation) (None, 45, 45, 256) 0 pool2_bn[0][0]		
pool2_conv (Conv2D) (None, 45, 45, 128) 32768 pool2_relu[0][0]		
pool2_pool (AveragePooling2D) (None, 22, 22, 128) 0 pool2_conv[0][0]		
conv3_block1_0_bn (BatchNormali (None, 22, 22, 128) 512 pool2_pool[0][0]		
conv3_block1_0_relu (Activation (None, 22, 22, 128) 0 n[0][0]		conv3_block1_0_b
conv3_block1_1_conv (Conv2D) (None, 22, 22, 128) 16384 elu[0][0]		conv3_block1_0_r
conv3_block1_1_bn (BatchNormali (None, 22, 22, 128) 512 onv[0][0]		conv3_block1_1_c
conv3_block1_1_relu (Activation (None, 22, 22, 128) 0 n[0][0]		conv3_block1_1_b

conv3_block1_2_conv (Conv2D) (None, 22, 22, 32) 36864	conv3_block1_1_r elu[0][0]
conv3_block1_concat (Concatenat (None, 22, 22, 160) 0	pool2_pool[0][0] conv3_block1_2_c onv[0][0]
conv3_block2_0_bn (BatchNormali (None, 22, 22, 160) 640	conv3_block1_con cat[0][0]
conv3_block2_0_relu (Activation (None, 22, 22, 160) 0	conv3_block2_0_b n[0][0]
conv3_block2_1_conv (Conv2D) (None, 22, 22, 128) 20480	conv3_block2_0_r elu[0][0]
conv3_block2_1_bn (BatchNormali (None, 22, 22, 128) 512	conv3_block2_1_c onv[0][0]
conv3_block2_1_relu (Activation (None, 22, 22, 128) 0	conv3_block2_1_b n[0][0]
conv3_block2_2_conv (Conv2D) (None, 22, 22, 32) 36864	conv3_block2_1_r elu[0][0]
conv3_block2_concat (Concatenat (None, 22, 22, 192) 0	conv3_block1_con cat[0][0] conv3_block2_2_c onv[0][0]
conv3_block3_0_bn (BatchNormali (None, 22, 22, 192) 768	conv3_block2_con cat[0][0]
conv3_block3_0_relu (Activation (None, 22, 22, 192) 0	conv3_block3_0_b n[0][0]
conv3_block3_1_conv (Conv2D) (None, 22, 22, 128) 24576	conv3_block3_0_r elu[0][0]
conv3_block3_1_bn (BatchNormali (None, 22, 22, 128) 512	conv3_block3_1_c onv[0][0]
conv3_block3_1_relu (Activation (None, 22, 22, 128) 0	conv3_block3_1_b n[0][0]

conv3_block3_2_conv (Conv2D) (None, 22, 22, 32) 36864	conv3_block3_1_r
elu[0][0]	
conv3_block3_concat (Concatenat (None, 22, 22, 224) 0	conv3_block2_con
cat[0][0]	
conv3_block3_conv[0][0]	conv3_block3_2_c
conv3_block4_0_bn (BatchNormali (None, 22, 22, 224) 896	conv3_block3_con
cat[0][0]	
conv3_block4_0_relu (Activation (None, 22, 22, 224) 0	conv3_block4_0_b
bn[0][0]	
conv3_block4_1_conv (Conv2D) (None, 22, 22, 128) 28672	conv3_block4_0_r
elu[0][0]	
conv3_block4_1_bn (BatchNormali (None, 22, 22, 128) 512	conv3_block4_1_c
conv[0][0]	
conv3_block4_1_relu (Activation (None, 22, 22, 128) 0	conv3_block4_1_b
bn[0][0]	
conv3_block4_2_conv (Conv2D) (None, 22, 22, 32) 36864	conv3_block4_1_r
elu[0][0]	
conv3_block4_concat (Concatenat (None, 22, 22, 256) 0	conv3_block3_con
cat[0][0]	
conv3_block4_conv[0][0]	conv3_block4_2_c
conv3_block5_0_bn (BatchNormali (None, 22, 22, 256) 1024	conv3_block4_con
cat[0][0]	
conv3_block5_0_relu (Activation (None, 22, 22, 256) 0	conv3_block5_0_b
bn[0][0]	
conv3_block5_1_conv (Conv2D) (None, 22, 22, 128) 32768	conv3_block5_0_r
elu[0][0]	
conv3_block5_1_bn (BatchNormali (None, 22, 22, 128) 512	conv3_block5_1_c
conv[0][0]	
conv3_block5_1_relu (Activation (None, 22, 22, 128) 0	conv3_block5_1_b
bn[0][0]	

conv3_block5_2_conv (Conv2D) (None, 22, 22, 32) 36864	conv3_block5_1_r elu[0][0]	
conv3_block5_concat (Concatenat (None, 22, 22, 288) 0 cat[0][0] conv[0][0]		conv3_block4_con conv3_block5_2_c
conv3_block6_0_bn (BatchNormali (None, 22, 22, 288) 1152 cat[0][0]		conv3_block5_con
conv3_block6_0_relu (Activation (None, 22, 22, 288) 0 n[0][0]		conv3_block6_0_b n[0][0]
conv3_block6_1_conv (Conv2D) (None, 22, 22, 128) 36864	conv3_block6_0_r elu[0][0]	
conv3_block6_1_bn (BatchNormali (None, 22, 22, 128) 512 conv[0][0]		conv3_block6_1_c
conv3_block6_1_relu (Activation (None, 22, 22, 128) 0 n[0][0]		conv3_block6_1_b n[0][0]
conv3_block6_2_conv (Conv2D) (None, 22, 22, 32) 36864	conv3_block6_1_r elu[0][0]	
conv3_block6_concat (Concatenat (None, 22, 22, 320) 0 cat[0][0] conv[0][0]		conv3_block5_con conv3_block6_2_c
conv3_block7_0_bn (BatchNormali (None, 22, 22, 320) 1280 cat[0][0]		conv3_block6_con
conv3_block7_0_relu (Activation (None, 22, 22, 320) 0 n[0][0]		conv3_block7_0_b n[0][0]
conv3_block7_1_conv (Conv2D) (None, 22, 22, 128) 40960	conv3_block7_0_r elu[0][0]	
conv3_block7_1_bn (BatchNormali (None, 22, 22, 128) 512 conv[0][0]		conv3_block7_1_c
conv3_block7_1_relu (Activation (None, 22, 22, 128) 0 n[0][0]		conv3_block7_1_b n[0][0]

conv3_block7_2_conv (Conv2D) (None, 22, 22, 32) 36864	conv3_block7_1_r elu[0][0]
conv3_block7_concat (Concatenat (None, 22, 22, 352) 0	conv3_block6_con cat[0][0]
conv3_block7_2_conv [0][0]	conv3_block7_2_c
conv3_block8_0_bn (BatchNormali (None, 22, 22, 352) 1408	conv3_block7_con cat[0][0]
conv3_block8_0_relu (Activation (None, 22, 22, 352) 0	conv3_block8_0_b n[0][0]
conv3_block8_1_conv (Conv2D) (None, 22, 22, 128) 45056	conv3_block8_0_r elu[0][0]
conv3_block8_1_bn (BatchNormali (None, 22, 22, 128) 512	conv3_block8_1_c onv[0][0]
conv3_block8_1_relu (Activation (None, 22, 22, 128) 0	conv3_block8_1_b n[0][0]
conv3_block8_2_conv (Conv2D) (None, 22, 22, 32) 36864	conv3_block8_1_r elu[0][0]
conv3_block8_concat (Concatenat (None, 22, 22, 384) 0	conv3_block7_con cat[0][0]
conv3_block8_2_conv [0][0]	conv3_block8_2_c
conv3_block9_0_bn (BatchNormali (None, 22, 22, 384) 1536	conv3_block8_con cat[0][0]
conv3_block9_0_relu (Activation (None, 22, 22, 384) 0	conv3_block9_0_b n[0][0]
conv3_block9_1_conv (Conv2D) (None, 22, 22, 128) 49152	conv3_block9_0_r elu[0][0]
conv3_block9_1_bn (BatchNormali (None, 22, 22, 128) 512	conv3_block9_1_c onv[0][0]
conv3_block9_1_relu (Activation (None, 22, 22, 128) 0	conv3_block9_1_b n[0][0]

conv3_block9_2_conv (Conv2D) (None, 22, 22, 32) 36864	conv3_block9_1_r elu[0][0]
conv3_block9_concat (Concatenat (None, 22, 22, 416) 0 cat[0][0]	conv3_block8_con conv3_block9_2_c onv[0][0]
conv3_block10_0_bn (BatchNormal (None, 22, 22, 416) 1664 cat[0][0]	conv3_block9_con cat[0][0]
conv3_block10_0_relu (Activatio (None, 22, 22, 416) 0 bn[0][0]	conv3_block10_0_ bn[0][0]
conv3_block10_1_conv (Conv2D) (None, 22, 22, 128) 53248	conv3_block10_0_ relu[0][0]
conv3_block10_1_bn (BatchNormal (None, 22, 22, 128) 512 conv[0][0]	conv3_block10_1_ conv[0][0]
conv3_block10_1_relu (Activatio (None, 22, 22, 128) 0 bn[0][0]	conv3_block10_1_ bn[0][0]
conv3_block10_2_conv (Conv2D) (None, 22, 22, 32) 36864	conv3_block10_1_ relu[0][0]
conv3_block10_concat (Concatena (None, 22, 22, 448) 0 cat[0][0]	conv3_block9_con cat[0][0]
conv3_block10_2_conv (Conv2D) (None, 22, 22, 32) 36864	conv3_block10_2_ conv[0][0]
conv3_block11_0_bn (BatchNormal (None, 22, 22, 448) 1792 ncat[0][0]	conv3_block10_co ncat[0][0]
conv3_block11_0_relu (Activatio (None, 22, 22, 448) 0 bn[0][0]	conv3_block11_0_ bn[0][0]
conv3_block11_1_conv (Conv2D) (None, 22, 22, 128) 57344	conv3_block11_0_ relu[0][0]
conv3_block11_1_bn (BatchNormal (None, 22, 22, 128) 512 conv[0][0]	conv3_block11_1_ conv[0][0]
conv3_block11_1_relu (Activatio (None, 22, 22, 128) 0 bn[0][0]	conv3_block11_1_ bn[0][0]

conv3_block11_2_conv (Conv2D) (None, 22, 22, 32) 36864	conv3_block11_1_relu[0][0]	
conv3_block11_concat (Concatenation) (None, 22, 22, 480) 0 ncat[0][0]	conv3_block10_co	conv3_block11_2_conv[0][0]
conv3_block12_0_bn (BatchNormal) (None, 22, 22, 480) 1920 ncat[0][0]	conv3_block11_co	
conv3_block12_0_relu (Activation) (None, 22, 22, 480) 0 bn[0][0]	conv3_block12_0_bn	
conv3_block12_1_conv (Conv2D) (None, 22, 22, 128) 61440 relu[0][0]	conv3_block12_0_re	
conv3_block12_1_bn (BatchNormal) (None, 22, 22, 128) 512 conv[0][0]	conv3_block12_1_co	
conv3_block12_1_relu (Activation) (None, 22, 22, 128) 0 bn[0][0]	conv3_block12_1_re	
conv3_block12_2_conv (Conv2D) (None, 22, 22, 32) 36864 relu[0][0]	conv3_block12_1_re	
conv3_block12_concat (Concatenation) (None, 22, 22, 512) 0 ncat[0][0]	conv3_block11_co	conv3_block12_2_conv[0][0]
pool3_bn (BatchNormalization) (None, 22, 22, 512) 2048 ncat[0][0]	conv3_block12_co	
pool3_relu (Activation) (None, 22, 22, 512) 0	pool3_bn[0][0]	
pool3_conv (Conv2D) (None, 22, 22, 256) 131072	pool3_relu[0][0]	
pool3_pool (AveragePooling2D) (None, 11, 11, 256) 0	pool3_conv[0][0]	
conv4_block1_0_bn (BatchNormal) (None, 11, 11, 256) 1024	pool3_pool[0][0]	
conv4_block1_0_relu (Activation) (None, 11, 11, 256) 0 n[0][0]	conv4_block1_0_b	

conv4_block1_1_conv (Conv2D) (None, 11, 11, 128) 32768	conv4_block1_0_r elu[0][0]
conv4_block1_1_bn (BatchNormali (None, 11, 11, 128) 512	conv4_block1_1_c onv[0][0]
conv4_block1_1_relu (Activation (None, 11, 11, 128) 0	conv4_block1_1_b n[0][0]
conv4_block1_2_conv (Conv2D) (None, 11, 11, 32) 36864	conv4_block1_1_r elu[0][0]
conv4_block1_concat (Concatenat (None, 11, 11, 288) 0	pool3_pool[0][0] conv4_block1_2_c onv[0][0]
conv4_block2_0_bn (BatchNormali (None, 11, 11, 288) 1152	conv4_block1_con cat[0][0]
conv4_block2_0_relu (Activation (None, 11, 11, 288) 0	conv4_block2_0_b n[0][0]
conv4_block2_1_conv (Conv2D) (None, 11, 11, 128) 36864	conv4_block2_0_r elu[0][0]
conv4_block2_1_bn (BatchNormali (None, 11, 11, 128) 512	conv4_block2_1_c onv[0][0]
conv4_block2_1_relu (Activation (None, 11, 11, 128) 0	conv4_block2_1_b n[0][0]
conv4_block2_2_conv (Conv2D) (None, 11, 11, 32) 36864	conv4_block2_1_r elu[0][0]
conv4_block2_concat (Concatenat (None, 11, 11, 320) 0	conv4_block1_con cat[0][0]
conv4_block2_concat (Concatenat (None, 11, 11, 320) 0	conv4_block2_2_c onv[0][0]
conv4_block3_0_bn (BatchNormali (None, 11, 11, 320) 1280	conv4_block2_con cat[0][0]
conv4_block3_0_relu (Activation (None, 11, 11, 320) 0	conv4_block3_0_b n[0][0]

conv4_block3_1_conv (Conv2D) (None, 11, 11, 128) 40960	conv4_block3_0_r elu[0][0]
conv4_block3_1_bn (BatchNormali (None, 11, 11, 128) 512	conv4_block3_1_c onv[0][0]
conv4_block3_1_relu (Activation (None, 11, 11, 128) 0	conv4_block3_1_b n[0][0]
conv4_block3_2_conv (Conv2D) (None, 11, 11, 32) 36864	conv4_block3_1_r elu[0][0]
conv4_block3_concat (Concatenat (None, 11, 11, 352) 0	conv4_block2_con cat[0][0]
conv4_block3_2_bn (BatchNormali (None, 11, 11, 352) 1408	conv4_block3_2_c onv[0][0]
conv4_block4_0_relu (Activation (None, 11, 11, 352) 0	conv4_block4_0_b n[0][0]
conv4_block4_1_conv (Conv2D) (None, 11, 11, 128) 45056	conv4_block4_0_r elu[0][0]
conv4_block4_1_bn (BatchNormali (None, 11, 11, 128) 512	conv4_block4_1_c onv[0][0]
conv4_block4_1_relu (Activation (None, 11, 11, 128) 0	conv4_block4_1_b n[0][0]
conv4_block4_2_conv (Conv2D) (None, 11, 11, 32) 36864	conv4_block4_1_r elu[0][0]
conv4_block4_concat (Concatenat (None, 11, 11, 384) 0	conv4_block3_con cat[0][0]
conv4_block4_2_bn (BatchNormali (None, 11, 11, 384) 1536	conv4_block4_2_c onv[0][0]
conv4_block5_0_relu (Activation (None, 11, 11, 384) 0	conv4_block5_0_b n[0][0]

conv4_block5_1_conv (Conv2D) (None, 11, 11, 128) 49152	conv4_block5_0_r elu[0][0]
conv4_block5_1_bn (BatchNormali (None, 11, 11, 128) 512	conv4_block5_1_c onv[0][0]
conv4_block5_1_relu (Activation (None, 11, 11, 128) 0	conv4_block5_1_b n[0][0]
conv4_block5_2_conv (Conv2D) (None, 11, 11, 32) 36864	conv4_block5_1_r elu[0][0]
conv4_block5_concat (Concatenat (None, 11, 11, 416) 0	conv4_block4_con cat[0][0]
conv4_block5_concat (Concatenat (None, 11, 11, 416) 0	conv4_block5_2_c onv[0][0]
conv4_block6_0_bn (BatchNormali (None, 11, 11, 416) 1664	conv4_block5_con cat[0][0]
conv4_block6_0_relu (Activation (None, 11, 11, 416) 0	conv4_block6_0_b n[0][0]
conv4_block6_1_conv (Conv2D) (None, 11, 11, 128) 53248	conv4_block6_0_r elu[0][0]
conv4_block6_1_bn (BatchNormali (None, 11, 11, 128) 512	conv4_block6_1_c onv[0][0]
conv4_block6_1_relu (Activation (None, 11, 11, 128) 0	conv4_block6_1_b n[0][0]
conv4_block6_2_conv (Conv2D) (None, 11, 11, 32) 36864	conv4_block6_1_r elu[0][0]
conv4_block6_concat (Concatenat (None, 11, 11, 448) 0	conv4_block5_con cat[0][0]
conv4_block6_concat (Concatenat (None, 11, 11, 448) 0	conv4_block6_2_c onv[0][0]
conv4_block7_0_bn (BatchNormali (None, 11, 11, 448) 1792	conv4_block6_con cat[0][0]
conv4_block7_0_relu (Activation (None, 11, 11, 448) 0	conv4_block7_0_b n[0][0]

conv4_block7_1_conv (Conv2D) (None, 11, 11, 128) 57344	conv4_block7_0_r elu[0][0]
conv4_block7_1_bn (BatchNormali (None, 11, 11, 128) 512	conv4_block7_1_c onv[0][0]
conv4_block7_1_relu (Activation (None, 11, 11, 128) 0	conv4_block7_1_b n[0][0]
conv4_block7_2_conv (Conv2D) (None, 11, 11, 32) 36864	conv4_block7_1_r elu[0][0]
conv4_block7_concat (Concatenat (None, 11, 11, 480) 0	conv4_block6_con cat[0][0]
conv4_block7_2_bn (BatchNormali (None, 11, 11, 480) 1920	conv4_block7_2_c onv[0][0]
conv4_block8_0_relu (Activation (None, 11, 11, 480) 0	conv4_block8_0_b n[0][0]
conv4_block8_1_conv (Conv2D) (None, 11, 11, 128) 61440	conv4_block8_0_r elu[0][0]
conv4_block8_1_bn (BatchNormali (None, 11, 11, 128) 512	conv4_block8_1_c onv[0][0]
conv4_block8_1_relu (Activation (None, 11, 11, 128) 0	conv4_block8_1_b n[0][0]
conv4_block8_2_conv (Conv2D) (None, 11, 11, 32) 36864	conv4_block8_1_r elu[0][0]
conv4_block8_concat (Concatenat (None, 11, 11, 512) 0	conv4_block7_con cat[0][0]
conv4_block8_2_bn (BatchNormali (None, 11, 11, 512) 2048	conv4_block8_2_c onv[0][0]
conv4_block9_0_relu (Activation (None, 11, 11, 512) 0	conv4_block9_0_b n[0][0]

conv4_block9_1_conv (Conv2D) (None, 11, 11, 128) 65536	elu[0][0]	conv4_block9_0_r
conv4_block9_1_bn (BatchNormal) (None, 11, 11, 128) 512	bn[0][0]	conv4_block9_1_c
conv4_block9_1_relu (Activation) (None, 11, 11, 128) 0	n[0][0]	conv4_block9_1_b
conv4_block9_2_conv (Conv2D) (None, 11, 11, 32) 36864	elu[0][0]	conv4_block9_1_r
conv4_block9_concat (Concatenat) (None, 11, 11, 544) 0	cat[0][0]	conv4_block8_con
conv4_block9_concat (Concatenat) (None, 11, 11, 544) 0	onv[0][0]	conv4_block9_2_c
conv4_block10_0_bn (BatchNormal) (None, 11, 11, 544) 2176	cat[0][0]	conv4_block9_con
conv4_block10_0_relu (Activation) (None, 11, 11, 544) 0	bn[0][0]	conv4_block10_0_b
conv4_block10_1_conv (Conv2D) (None, 11, 11, 128) 69632	relu[0][0]	conv4_block10_0_r
conv4_block10_1_bn (BatchNormal) (None, 11, 11, 128) 512	conv[0][0]	conv4_block10_1_b
conv4_block10_1_relu (Activation) (None, 11, 11, 128) 0	bn[0][0]	conv4_block10_1_c
conv4_block10_2_conv (Conv2D) (None, 11, 11, 32) 36864	relu[0][0]	conv4_block10_1_r
conv4_block10_concat (Concatena) (None, 11, 11, 576) 0	cat[0][0]	conv4_block9_con
conv4_block10_concat (Concatena) (None, 11, 11, 576) 0	conv[0][0]	conv4_block10_2_c
conv4_block11_0_bn (BatchNormal) (None, 11, 11, 576) 2304	ncat[0][0]	conv4_block10_co
conv4_block11_0_relu (Activation) (None, 11, 11, 576) 0	bn[0][0]	conv4_block11_0_b

conv4_block11_1_conv (Conv2D) (None, 11, 11, 128) 73728	conv4_block11_0_relu[0][0]
conv4_block11_1_bn (BatchNormal (None, 11, 11, 128) 512	conv4_block11_1_conv[0][0]
conv4_block11_1_relu (Activatio (None, 11, 11, 128) 0	conv4_block11_1_bn[0][0]
conv4_block11_2_conv (Conv2D) (None, 11, 11, 32) 36864	conv4_block11_1_relu[0][0]
conv4_block11_concat (Concatena (None, 11, 11, 608) 0	conv4_block10_cat[0][0]
conv4_block11_2_conv[0][0]	conv4_block11_2_conv[0][0]
conv4_block12_0_bn (BatchNormal (None, 11, 11, 608) 2432	conv4_block11_cat[0][0]
conv4_block12_0_relu (Activatio (None, 11, 11, 608) 0	conv4_block12_0_bn[0][0]
conv4_block12_1_conv (Conv2D) (None, 11, 11, 128) 77824	conv4_block12_0_relu[0][0]
conv4_block12_1_bn (BatchNormal (None, 11, 11, 128) 512	conv4_block12_1_conv[0][0]
conv4_block12_1_relu (Activatio (None, 11, 11, 128) 0	conv4_block12_1_bn[0][0]
conv4_block12_2_conv (Conv2D) (None, 11, 11, 32) 36864	conv4_block12_1_relu[0][0]
conv4_block12_concat (Concatena (None, 11, 11, 640) 0	conv4_block11_cat[0][0]
conv4_block12_2_conv[0][0]	conv4_block12_2_conv[0][0]
conv4_block13_0_bn (BatchNormal (None, 11, 11, 640) 2560	conv4_block12_cat[0][0]
conv4_block13_0_relu (Activatio (None, 11, 11, 640) 0	conv4_block13_0_bn[0][0]

conv4_block13_1_conv (Conv2D) (None, 11, 11, 128) 81920	relu[0][0]	conv4_block13_0_
conv4_block13_1_bn (BatchNormal (None, 11, 11, 128) 512	conv[0][0]	conv4_block13_1_
conv4_block13_1_relu (Activatio (None, 11, 11, 128) 0	bn[0][0]	conv4_block13_1_
conv4_block13_2_conv (Conv2D) (None, 11, 11, 32) 36864	relu[0][0]	conv4_block13_1_
conv4_block13_concat (Concatena (None, 11, 11, 672) 0	ncat[0][0]	conv4_block12_co
conv4_block13_2_conv[0][0]		conv4_block13_2_
conv4_block14_0_bn (BatchNormal (None, 11, 11, 672) 2688	ncat[0][0]	conv4_block13_co
conv4_block14_0_relu (Activatio (None, 11, 11, 672) 0	bn[0][0]	conv4_block14_0_
conv4_block14_1_conv (Conv2D) (None, 11, 11, 128) 86016	relu[0][0]	conv4_block14_0_
conv4_block14_1_bn (BatchNormal (None, 11, 11, 128) 512	conv[0][0]	conv4_block14_1_
conv4_block14_1_relu (Activatio (None, 11, 11, 128) 0	bn[0][0]	conv4_block14_1_
conv4_block14_2_conv (Conv2D) (None, 11, 11, 32) 36864	relu[0][0]	conv4_block14_1_
conv4_block14_concat (Concatena (None, 11, 11, 704) 0	ncat[0][0]	conv4_block13_co
conv4_block14_2_conv[0][0]		conv4_block14_2_
conv4_block15_0_bn (BatchNormal (None, 11, 11, 704) 2816	ncat[0][0]	conv4_block14_co
conv4_block15_0_relu (Activatio (None, 11, 11, 704) 0	bn[0][0]	conv4_block15_0_

conv4_block15_1_conv (Conv2D) (None, 11, 11, 128) 90112	relu[0][0]	conv4_block15_0_
conv4_block15_1_bn (BatchNormal (None, 11, 11, 128) 512	conv[0][0]	conv4_block15_1_
conv4_block15_1_relu (Activatio (None, 11, 11, 128) 0	bn[0][0]	conv4_block15_1_
conv4_block15_2_conv (Conv2D) (None, 11, 11, 32) 36864	relu[0][0]	conv4_block15_1_
conv4_block15_concat (Concatena (None, 11, 11, 736) 0	ncat[0][0]	conv4_block14_co
conv4_block15_2_conv [0][0]		conv4_block15_2_
conv4_block16_0_bn (BatchNormal (None, 11, 11, 736) 2944	conv[0][0]	conv4_block15_co
conv4_block16_0_relu (Activatio (None, 11, 11, 736) 0	bn[0][0]	conv4_block16_0_
conv4_block16_1_conv (Conv2D) (None, 11, 11, 128) 94208	relu[0][0]	conv4_block16_0_
conv4_block16_1_bn (BatchNormal (None, 11, 11, 128) 512	conv[0][0]	conv4_block16_1_
conv4_block16_1_relu (Activatio (None, 11, 11, 128) 0	bn[0][0]	conv4_block16_1_
conv4_block16_2_conv (Conv2D) (None, 11, 11, 32) 36864	relu[0][0]	conv4_block16_1_
conv4_block16_concat (Concatena (None, 11, 11, 768) 0	ncat[0][0]	conv4_block15_co
conv4_block16_2_conv [0][0]		conv4_block16_2_
conv4_block17_0_bn (BatchNormal (None, 11, 11, 768) 3072	conv[0][0]	conv4_block16_co
conv4_block17_0_relu (Activatio (None, 11, 11, 768) 0	bn[0][0]	conv4_block17_0_

conv4_block17_1_conv (Conv2D) (None, 11, 11, 128) 98304	conv4_block17_0_relu[0][0]
conv4_block17_1_bn (BatchNormal (None, 11, 11, 128) 512	conv4_block17_1_conv[0][0]
conv4_block17_1_relu (Activatio (None, 11, 11, 128) 0	conv4_block17_1_bn[0][0]
conv4_block17_2_conv (Conv2D) (None, 11, 11, 32) 36864	conv4_block17_1_relu[0][0]
conv4_block17_concat (Concatena (None, 11, 11, 800) 0	conv4_block16_co ncat[0][0]
conv4_block17_2_conv[0][0]	conv4_block17_2_
conv4_block18_0_bn (BatchNormal (None, 11, 11, 800) 3200	conv4_block17_co ncat[0][0]
conv4_block18_0_relu (Activatio (None, 11, 11, 800) 0	conv4_block18_0_bn[0][0]
conv4_block18_1_conv (Conv2D) (None, 11, 11, 128) 102400	conv4_block18_0_relu[0][0]
conv4_block18_1_bn (BatchNormal (None, 11, 11, 128) 512	conv4_block18_1_conv[0][0]
conv4_block18_1_relu (Activatio (None, 11, 11, 128) 0	conv4_block18_1_bn[0][0]
conv4_block18_2_conv (Conv2D) (None, 11, 11, 32) 36864	conv4_block18_1_relu[0][0]
conv4_block18_concat (Concatena (None, 11, 11, 832) 0	conv4_block17_co ncat[0][0]
conv4_block18_2_conv[0][0]	conv4_block18_2_
conv4_block19_0_bn (BatchNormal (None, 11, 11, 832) 3328	conv4_block18_co ncat[0][0]
conv4_block19_0_relu (Activatio (None, 11, 11, 832) 0	conv4_block19_0_bn[0][0]

conv4_block19_1_conv (Conv2D) (None, 11, 11, 128)	106496	conv4_block19_0_relu[0][0]
conv4_block19_1_bn (BatchNormal (None, 11, 11, 128))	512	conv4_block19_1_conv[0][0]
conv4_block19_1_relu (Activation (None, 11, 11, 128))	0	conv4_block19_1_bn[0][0]
conv4_block19_2_conv (Conv2D) (None, 11, 11, 32)	36864	conv4_block19_1_relu[0][0]
conv4_block19_concat (Concatenation (None, 11, 11, 864))	0	conv4_block18_cat[0][0]
conv4_block19_2_relu (Activation (None, 11, 11, 864))	0	conv4_block19_2_conv[0][0]
conv4_block20_0_bn (BatchNormal (None, 11, 11, 864))	3456	conv4_block19_cat[0][0]
conv4_block20_0_relu (Activation (None, 11, 11, 864))	0	conv4_block20_0_bn[0][0]
conv4_block20_1_conv (Conv2D) (None, 11, 11, 128)	110592	conv4_block20_0_relu[0][0]
conv4_block20_1_bn (BatchNormal (None, 11, 11, 128))	512	conv4_block20_1_conv[0][0]
conv4_block20_1_relu (Activation (None, 11, 11, 128))	0	conv4_block20_1_bn[0][0]
conv4_block20_2_conv (Conv2D) (None, 11, 11, 32)	36864	conv4_block20_1_relu[0][0]
conv4_block20_concat (Concatenation (None, 11, 11, 896))	0	conv4_block19_cat[0][0]
conv4_block20_2_relu (Activation (None, 11, 11, 896))	0	conv4_block20_2_conv[0][0]
conv4_block21_0_bn (BatchNormal (None, 11, 11, 896))	3584	conv4_block20_cat[0][0]
conv4_block21_0_relu (Activation (None, 11, 11, 896))	0	conv4_block21_0_bn[0][0]

conv4_block21_1_conv (Conv2D) (None, 11, 11, 128) 114688	conv4_block21_0_relu[0][0]
conv4_block21_1_bn (BatchNormal (None, 11, 11, 128) 512	conv4_block21_1_conv[0][0]
conv4_block21_1_relu (Activatio (None, 11, 11, 128) 0	conv4_block21_1_bn[0][0]
conv4_block21_2_conv (Conv2D) (None, 11, 11, 32) 36864	conv4_block21_1_relu[0][0]
conv4_block21_concat (Concatena (None, 11, 11, 928) 0	conv4_block20_cat[0][0]
conv4_block21_2_conv[0][0]	conv4_block21_2_conv[0][0]
conv4_block22_0_bn (BatchNormal (None, 11, 11, 928) 3712	conv4_block21_cat[0][0]
conv4_block22_0_relu (Activatio (None, 11, 11, 928) 0	conv4_block22_0_bn[0][0]
conv4_block22_1_conv (Conv2D) (None, 11, 11, 128) 118784	conv4_block22_0_relu[0][0]
conv4_block22_1_bn (BatchNormal (None, 11, 11, 128) 512	conv4_block22_1_conv[0][0]
conv4_block22_1_relu (Activatio (None, 11, 11, 128) 0	conv4_block22_1_bn[0][0]
conv4_block22_2_conv (Conv2D) (None, 11, 11, 32) 36864	conv4_block22_1_relu[0][0]
conv4_block22_concat (Concatena (None, 11, 11, 960) 0	conv4_block21_cat[0][0]
conv4_block22_2_conv[0][0]	conv4_block22_2_conv[0][0]
conv4_block23_0_bn (BatchNormal (None, 11, 11, 960) 3840	conv4_block22_cat[0][0]
conv4_block23_0_relu (Activatio (None, 11, 11, 960) 0	conv4_block23_0_bn[0][0]

conv4_block23_1_conv (Conv2D) (None, 11, 11, 128)	122880	conv4_block23_0_relu[0][0]
conv4_block23_1_bn (BatchNormal (None, 11, 11, 128)	512	conv4_block23_1_conv[0][0]
conv4_block23_1_relu (Activatio (None, 11, 11, 128)	0	conv4_block23_1_bn[0][0]
conv4_block23_2_conv (Conv2D) (None, 11, 11, 32)	36864	conv4_block23_1_relu[0][0]
conv4_block23_concat (Concatena (None, 11, 11, 992)	0	conv4_block22_c0ncat[0][0]
conv4_block23_2_conv[0][0]		conv4_block23_2_conv[0][0]
conv4_block24_0_bn (BatchNormal (None, 11, 11, 992)	3968	conv4_block23_c0ncat[0][0]
conv4_block24_0_relu (Activatio (None, 11, 11, 992)	0	conv4_block24_0_bn[0][0]
conv4_block24_1_conv (Conv2D) (None, 11, 11, 128)	126976	conv4_block24_0_relu[0][0]
conv4_block24_1_bn (BatchNormal (None, 11, 11, 128)	512	conv4_block24_1_conv[0][0]
conv4_block24_1_relu (Activatio (None, 11, 11, 128)	0	conv4_block24_1_bn[0][0]
conv4_block24_2_conv (Conv2D) (None, 11, 11, 32)	36864	conv4_block24_1_relu[0][0]
conv4_block24_concat (Concatena (None, 11, 11, 1024)	0	conv4_block23_c0ncat[0][0]
conv4_block24_2_conv[0][0]		conv4_block24_2_conv[0][0]
pool4_bn (BatchNormalization) (None, 11, 11, 1024)	4096	conv4_block24_c0ncat[0][0]
pool4_relu (Activation)	(None, 11, 11, 1024) 0	pool4_bn[0][0]
pool4_conv (Conv2D)	(None, 11, 11, 512)	524288
		pool4_relu[0][0]

pool4_pool (AveragePooling2D)	(None, 5, 5, 512)	0	pool4_conv[0][0]
conv5_block1_0_bn (BatchNormali	(None, 5, 5, 512)	2048	pool4_pool[0][0]
conv5_block1_0_relu (Activation	(None, 5, 5, 512)	0	conv5_block1_0_b n[0][0]
conv5_block1_1_conv (Conv2D)	(None, 5, 5, 128)	65536	conv5_block1_0_r elu[0][0]
conv5_block1_1_bn (BatchNormali	(None, 5, 5, 128)	512	conv5_block1_1_c onv[0][0]
conv5_block1_1_relu (Activation	(None, 5, 5, 128)	0	conv5_block1_1_b n[0][0]
conv5_block1_2_conv (Conv2D)	(None, 5, 5, 32)	36864	conv5_block1_1_r elu[0][0]
conv5_block1_concat (Concatenat	(None, 5, 5, 544)	0	pool4_pool[0][0] conv5_block1_2_c onv[0][0]
conv5_block2_0_bn (BatchNormali	(None, 5, 5, 544)	2176	conv5_block1_con cat[0][0]
conv5_block2_0_relu (Activation	(None, 5, 5, 544)	0	conv5_block2_0_b n[0][0]
conv5_block2_1_conv (Conv2D)	(None, 5, 5, 128)	69632	conv5_block2_0_r elu[0][0]
conv5_block2_1_bn (BatchNormali	(None, 5, 5, 128)	512	conv5_block2_1_c onv[0][0]
conv5_block2_1_relu (Activation	(None, 5, 5, 128)	0	conv5_block2_1_b n[0][0]
conv5_block2_2_conv (Conv2D)	(None, 5, 5, 32)	36864	conv5_block2_1_r elu[0][0]
conv5_block2_concat (Concatenat	(None, 5, 5, 576)	0	conv5_block1_con cat[0][0] conv5_block2_2_c

onv[0][0]

conv5_block3_0_bn (BatchNormali (None, 5, 5, 576) 2304 conv5_block2_con
cat[0][0]

conv5_block3_0_relu (Activation (None, 5, 5, 576) 0 conv5_block3_0_b
n[0][0]

conv5_block3_1_conv (Conv2D) (None, 5, 5, 128) 73728 conv5_block3_0_r
elu[0][0]

conv5_block3_1_bn (BatchNormali (None, 5, 5, 128) 512 conv5_block3_1_c
onv[0][0]

conv5_block3_1_relu (Activation (None, 5, 5, 128) 0 conv5_block3_1_b
n[0][0]

conv5_block3_2_conv (Conv2D) (None, 5, 5, 32) 36864 conv5_block3_1_r
elu[0][0]

conv5_block3_concat (Concatenat (None, 5, 5, 608) 0 conv5_block2_con
cat[0][0]
conv5_block3_2_c
onv[0][0]

conv5_block4_0_bn (BatchNormali (None, 5, 5, 608) 2432 conv5_block3_con
cat[0][0]

conv5_block4_0_relu (Activation (None, 5, 5, 608) 0 conv5_block4_0_b
n[0][0]

conv5_block4_1_conv (Conv2D) (None, 5, 5, 128) 77824 conv5_block4_0_r
elu[0][0]

conv5_block4_1_bn (BatchNormali (None, 5, 5, 128) 512 conv5_block4_1_c
onv[0][0]

conv5_block4_1_relu (Activation (None, 5, 5, 128) 0 conv5_block4_1_b
n[0][0]

conv5_block4_2_conv (Conv2D) (None, 5, 5, 32) 36864 conv5_block4_1_r
elu[0][0]

conv5_block4_concat (Concatenat (None, 5, 5, 640) 0 conv5_block3_con
cat[0][0]
conv5_block4_2_c

onv[0][0]

conv5_block5_0_bn (BatchNormali (None, 5, 5, 640) 2560 conv5_block4_con
cat[0][0]

conv5_block5_0_relu (Activation (None, 5, 5, 640) 0 conv5_block5_0_b
n[0][0]

conv5_block5_1_conv (Conv2D) (None, 5, 5, 128) 81920 conv5_block5_0_r
elu[0][0]

conv5_block5_1_bn (BatchNormali (None, 5, 5, 128) 512 conv5_block5_1_c
onv[0][0]

conv5_block5_1_relu (Activation (None, 5, 5, 128) 0 conv5_block5_1_b
n[0][0]

conv5_block5_2_conv (Conv2D) (None, 5, 5, 32) 36864 conv5_block5_1_r
elu[0][0]

conv5_block5_concat (Concatenat (None, 5, 5, 672) 0 conv5_block4_con
cat[0][0]
conv5_block5_2_c
onv[0][0]

conv5_block6_0_bn (BatchNormali (None, 5, 5, 672) 2688 conv5_block5_con
cat[0][0]

conv5_block6_0_relu (Activation (None, 5, 5, 672) 0 conv5_block6_0_b
n[0][0]

conv5_block6_1_conv (Conv2D) (None, 5, 5, 128) 86016 conv5_block6_0_r
elu[0][0]

conv5_block6_1_bn (BatchNormali (None, 5, 5, 128) 512 conv5_block6_1_c
onv[0][0]

conv5_block6_1_relu (Activation (None, 5, 5, 128) 0 conv5_block6_1_b
n[0][0]

conv5_block6_2_conv (Conv2D) (None, 5, 5, 32) 36864 conv5_block6_1_r
elu[0][0]

conv5_block6_concat (Concatenat (None, 5, 5, 704) 0 conv5_block5_con
cat[0][0]
conv5_block6_2_c

onv[0][0]

conv5_block7_0_bn (BatchNormali (None, 5, 5, 704) 2816 conv5_block6_con
cat[0][0]

conv5_block7_0_relu (Activation (None, 5, 5, 704) 0 conv5_block7_0_b
n[0][0]

conv5_block7_1_conv (Conv2D) (None, 5, 5, 128) 90112 conv5_block7_0_r
elu[0][0]

conv5_block7_1_bn (BatchNormali (None, 5, 5, 128) 512 conv5_block7_1_c
onv[0][0]

conv5_block7_1_relu (Activation (None, 5, 5, 128) 0 conv5_block7_1_b
n[0][0]

conv5_block7_2_conv (Conv2D) (None, 5, 5, 32) 36864 conv5_block7_1_r
elu[0][0]

conv5_block7_concat (Concatenat (None, 5, 5, 736) 0 conv5_block6_con
cat[0][0]

conv5_block7_2_c

onv[0][0]

conv5_block8_0_bn (BatchNormali (None, 5, 5, 736) 2944 conv5_block7_con
cat[0][0]

conv5_block8_0_relu (Activation (None, 5, 5, 736) 0 conv5_block8_0_b
n[0][0]

conv5_block8_1_conv (Conv2D) (None, 5, 5, 128) 94208 conv5_block8_0_r
elu[0][0]

conv5_block8_1_bn (BatchNormali (None, 5, 5, 128) 512 conv5_block8_1_c
onv[0][0]

conv5_block8_1_relu (Activation (None, 5, 5, 128) 0 conv5_block8_1_b
n[0][0]

conv5_block8_2_conv (Conv2D) (None, 5, 5, 32) 36864 conv5_block8_1_r
elu[0][0]

conv5_block8_concat (Concatenat (None, 5, 5, 768) 0 conv5_block7_con
cat[0][0]

conv5_block8_2_c

onv[0][0]

conv5_block9_0_bn (BatchNormali (None, 5, 5, 768) 3072 conv5_block8_con
cat[0][0]

conv5_block9_0_relu (Activation (None, 5, 5, 768) 0 conv5_block9_0_b
n[0][0]

conv5_block9_1_conv (Conv2D) (None, 5, 5, 128) 98304 conv5_block9_0_r
elu[0][0]

conv5_block9_1_bn (BatchNormali (None, 5, 5, 128) 512 conv5_block9_1_c
onv[0][0]

conv5_block9_1_relu (Activation (None, 5, 5, 128) 0 conv5_block9_1_b
n[0][0]

conv5_block9_2_conv (Conv2D) (None, 5, 5, 32) 36864 conv5_block9_1_r
elu[0][0]

conv5_block9_concat (Concatenat (None, 5, 5, 800) 0 conv5_block8_con
cat[0][0]
conv5_block9_2_c
onv[0][0]

conv5_block10_0_bn (BatchNormal (None, 5, 5, 800) 3200 conv5_block9_con
cat[0][0]

conv5_block10_0_relu (Activatio (None, 5, 5, 800) 0 conv5_block10_0_
bn[0][0]

conv5_block10_1_conv (Conv2D) (None, 5, 5, 128) 102400 conv5_block10_0_
relu[0][0]

conv5_block10_1_bn (BatchNormal (None, 5, 5, 128) 512 conv5_block10_1_
conv[0][0]

conv5_block10_1_relu (Activatio (None, 5, 5, 128) 0 conv5_block10_1_
bn[0][0]

conv5_block10_2_conv (Conv2D) (None, 5, 5, 32) 36864 conv5_block10_1_
relu[0][0]

conv5_block10_concat (Concatena (None, 5, 5, 832) 0 conv5_block9_con
cat[0][0]
conv5_block10_2_

conv[0][0]

conv5_block11_0_bn (BatchNormal (None, 5, 5, 832) 3328 conv5_block10_co
ncat[0][0]

conv5_block11_0_relu (Activatio (None, 5, 5, 832) 0 conv5_block11_0_
bn[0][0]

conv5_block11_1_conv (Conv2D) (None, 5, 5, 128) 106496 conv5_block11_0_
relu[0][0]

conv5_block11_1_bn (BatchNormal (None, 5, 5, 128) 512 conv5_block11_1_
conv[0][0]

conv5_block11_1_relu (Activatio (None, 5, 5, 128) 0 conv5_block11_1_
bn[0][0]

conv5_block11_2_conv (Conv2D) (None, 5, 5, 32) 36864 conv5_block11_1_
relu[0][0]

conv5_block11_concat (Concatena (None, 5, 5, 864) 0 conv5_block10_co
ncat[0][0]
conv5_block11_2_
conv[0][0]

conv5_block12_0_bn (BatchNormal (None, 5, 5, 864) 3456 conv5_block11_co
ncat[0][0]

conv5_block12_0_relu (Activatio (None, 5, 5, 864) 0 conv5_block12_0_
bn[0][0]

conv5_block12_1_conv (Conv2D) (None, 5, 5, 128) 110592 conv5_block12_0_
relu[0][0]

conv5_block12_1_bn (BatchNormal (None, 5, 5, 128) 512 conv5_block12_1_
conv[0][0]

conv5_block12_1_relu (Activatio (None, 5, 5, 128) 0 conv5_block12_1_
bn[0][0]

conv5_block12_2_conv (Conv2D) (None, 5, 5, 32) 36864 conv5_block12_1_
relu[0][0]

conv5_block12_concat (Concatena (None, 5, 5, 896) 0 conv5_block11_co
ncat[0][0]
conv5_block12_2_

conv[0][0]

conv5_block13_0_bn (BatchNormal (None, 5, 5, 896) 3584 conv5_block12_co
ncat[0][0]

conv5_block13_0_relu (Activatio (None, 5, 5, 896) 0 conv5_block13_0_
bn[0][0]

conv5_block13_1_conv (Conv2D) (None, 5, 5, 128) 114688 conv5_block13_0_
relu[0][0]

conv5_block13_1_bn (BatchNormal (None, 5, 5, 128) 512 conv5_block13_1_
conv[0][0]

conv5_block13_1_relu (Activatio (None, 5, 5, 128) 0 conv5_block13_1_
bn[0][0]

conv5_block13_2_conv (Conv2D) (None, 5, 5, 32) 36864 conv5_block13_1_
relu[0][0]

conv5_block13_concat (Concatena (None, 5, 5, 928) 0 conv5_block12_co
ncat[0][0]
conv5_block13_2_
conv[0][0]

conv5_block14_0_bn (BatchNormal (None, 5, 5, 928) 3712 conv5_block13_co
ncat[0][0]

conv5_block14_0_relu (Activatio (None, 5, 5, 928) 0 conv5_block14_0_
bn[0][0]

conv5_block14_1_conv (Conv2D) (None, 5, 5, 128) 118784 conv5_block14_0_
relu[0][0]

conv5_block14_1_bn (BatchNormal (None, 5, 5, 128) 512 conv5_block14_1_
conv[0][0]

conv5_block14_1_relu (Activatio (None, 5, 5, 128) 0 conv5_block14_1_
bn[0][0]

conv5_block14_2_conv (Conv2D) (None, 5, 5, 32) 36864 conv5_block14_1_
relu[0][0]

conv5_block14_concat (Concatena (None, 5, 5, 960) 0 conv5_block13_co
ncat[0][0]
conv5_block14_2_

conv[0][0]

conv5_block15_0_bn (BatchNormal (None, 5, 5, 960) 3840 conv5_block14_co
ncat[0][0]

conv5_block15_0_relu (Activatio (None, 5, 5, 960) 0 conv5_block15_0_
bn[0][0]

conv5_block15_1_conv (Conv2D) (None, 5, 5, 128) 122880 conv5_block15_0_
relu[0][0]

conv5_block15_1_bn (BatchNormal (None, 5, 5, 128) 512 conv5_block15_1_
conv[0][0]

conv5_block15_1_relu (Activatio (None, 5, 5, 128) 0 conv5_block15_1_
bn[0][0]

conv5_block15_2_conv (Conv2D) (None, 5, 5, 32) 36864 conv5_block15_1_
relu[0][0]

conv5_block15_concat (Concatena (None, 5, 5, 992) 0 conv5_block14_co
ncat[0][0]
conv5_block15_2_
conv[0][0]

conv5_block16_0_bn (BatchNormal (None, 5, 5, 992) 3968 conv5_block15_co
ncat[0][0]

conv5_block16_0_relu (Activatio (None, 5, 5, 992) 0 conv5_block16_0_
bn[0][0]

conv5_block16_1_conv (Conv2D) (None, 5, 5, 128) 126976 conv5_block16_0_
relu[0][0]

conv5_block16_1_bn (BatchNormal (None, 5, 5, 128) 512 conv5_block16_1_
conv[0][0]

conv5_block16_1_relu (Activatio (None, 5, 5, 128) 0 conv5_block16_1_
bn[0][0]

conv5_block16_2_conv (Conv2D) (None, 5, 5, 32) 36864 conv5_block16_1_
relu[0][0]

conv5_block16_concat (Concatena (None, 5, 5, 1024) 0 conv5_block15_co
ncat[0][0]
conv5_block16_2_

```

conv[0][0]

bn (BatchNormalization)      (None, 5, 5, 1024)  4096    conv5_block16_co
ncat[0][0]

relu (Activation)          (None, 5, 5, 1024)  0        bn[0][0]

avg_pool (GlobalAveragePooling2 (None, 1024)      0        relu[0][0]
=====
=====

Total params: 7,037,504
Trainable params: 6,953,856
Non-trainable params: 83,648

```

```
In [ ]: layers = base_model.layers
print(f"The model has {len(layers)} layers")
```

The model has 428 layers

```
In [ ]: print(f"The input shape {base_model.input}")
print(f"The output shape {base_model.output}")
```

The input shape Tensor("input_1:0", shape=(None, 180, 180, 3), dtype=float32)
The output shape Tensor("avg_pool/Identity:0", shape=(None, 1024), dtype=float32)

```
#model = Sequential()
base_model = DenseNet121(include_top=False, weights='imagenet')
x = base_model.output

x = GlobalAveragePooling2D()(x)

predictions = Dense(1, activation="sigmoid")(x)

model = Model(inputs=base_model.input, outputs=predictions)
#model.add(base_model)
#model.add(GlobalAveragePooling2D())
#model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

```
In [ ]: r = model.fit(
    train,
    epochs=10,
    validation_data=validation,
    class_weight=class_weight,
    steps_per_epoch=100,
    validation_steps=25,
)
```

```

Epoch 1/10
100/100 [=====] - 21s 207ms/step - loss: 0.1705 - accuracy: 0.8275 - val_loss: 8.3540 - val_accuracy: 0.5000
Epoch 2/10
100/100 [=====] - 17s 169ms/step - loss: 0.1189 - accuracy: 0.8725 - val_loss: 2.6481 - val_accuracy: 0.6250
Epoch 3/10
100/100 [=====] - 18s 179ms/step - loss: 0.0975 - accuracy: 0.8875 - val_loss: 1.5859 - val_accuracy: 0.5000
Epoch 4/10
100/100 [=====] - 18s 176ms/step - loss: 0.1059 - accuracy: 0.8988 - val_loss: 0.9687 - val_accuracy: 0.5625
Epoch 5/10
100/100 [=====] - 18s 179ms/step - loss: 0.0798 - accuracy: 0.9112 - val_loss: 1.3149 - val_accuracy: 0.6875
Epoch 6/10
100/100 [=====] - 19s 187ms/step - loss: 0.0721 - accuracy: 0.9212 - val_loss: 2.3806 - val_accuracy: 0.5000
Epoch 7/10
100/100 [=====] - 17s 171ms/step - loss: 0.1056 - accuracy: 0.8975 - val_loss: 0.8888 - val_accuracy: 0.7500
Epoch 8/10
100/100 [=====] - 18s 181ms/step - loss: 0.0762 - accuracy: 0.9162 - val_loss: 1.3945 - val_accuracy: 0.6250
Epoch 9/10
100/100 [=====] - 17s 173ms/step - loss: 0.0845 - accuracy: 0.9150 - val_loss: 0.5547 - val_accuracy: 0.8750
Epoch 10/10
100/100 [=====] - 18s 184ms/step - loss: 0.0721 - accuracy: 0.9200 - val_loss: 0.3499 - val_accuracy: 0.9375

```

In []: `plt.figure(figsize=(12, 8))`

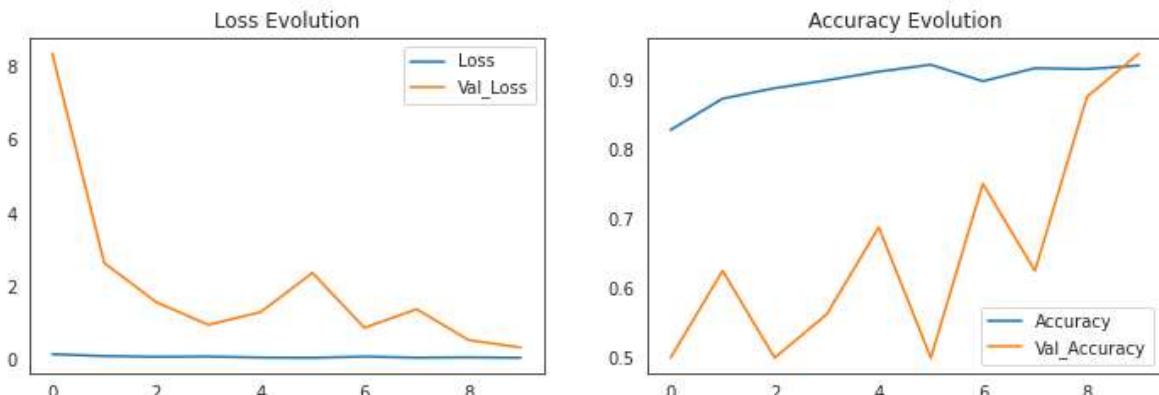
```

plt.subplot(2, 2, 1)
plt.plot(r.history['loss'], label='Loss')
plt.plot(r.history['val_loss'], label='Val_Loss')
plt.legend()
plt.title('Loss Evolution')

plt.subplot(2, 2, 2)
plt.plot(r.history['accuracy'], label='Accuracy')
plt.plot(r.history['val_accuracy'], label='Val_Accuracy')
plt.legend()
plt.title('Accuracy Evolution')

```

Out[]: `Text(0.5, 1.0, 'Accuracy Evolution')`



```
In [ ]: evaluation = model.evaluate(test)
print(f"Test Accuracy: {evaluation[1] * 100:.2f}%")

evaluation = model.evaluate(train)
print(f"Train Accuracy: {evaluation[1] * 100:.2f}%")

624/624 [=====] - 15s 24ms/step - loss: 0.3812 - accuracy: 0.8446
Test Accuracy: 84.46%
652/652 [=====] - 97s 148ms/step - loss: 0.2113 - accuracy: 0.9245
Train Accuracy: 92.45%
```

Evaluation

```
In [ ]: predicted_vals = model.predict(test, steps=len(test))

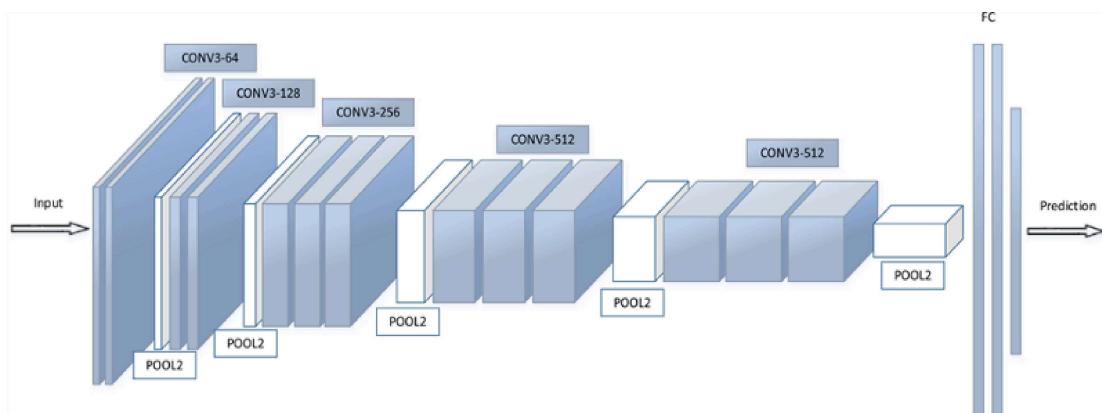
In [ ]: print(confusion_matrix(test.classes, predicted_vals > 0.5))
pd.DataFrame(classification_report(test.classes, predicted_vals > 0.5, output_dict=True))

[[175  59]
 [ 42 348]]
```

	0	1	accuracy	macro avg	weighted avg
precision	0.806452	0.855037	0.838141	0.830744	0.836817
recall	0.747863	0.892308	0.838141	0.820085	0.838141
f1-score	0.776053	0.873275	0.838141	0.824664	0.836817
support	234.000000	390.000000	0.838141	624.000000	624.000000

VGG16

Presented in 2014, VGG16 has a very simple and classical architecture, with blocks of 2 or 3 convolutional layers followed by a pooling layer, plus a final dense network composed of 2 hidden layers (of 4096 nodes each) and one output layer (of 1000 nodes). Only 3x3 filters are used.



```
In [ ]: from keras.models import Sequential
from keras.layers import GlobalAveragePooling2D
from keras.applications import VGG16

vgg16_base_model = VGG16(input_shape=(180,180,3),include_top=False,weights='imagine' )
```

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

```
In [ ]: vgg16_base_model.summary()
```

Model: "vgg16"

Layer (type)	Output Shape	Param #
=====		
input_3 (InputLayer)	[None, 180, 180, 3]	0
block1_conv1 (Conv2D)	(None, 180, 180, 64)	1792
block1_conv2 (Conv2D)	(None, 180, 180, 64)	36928
block1_pool (MaxPooling2D)	(None, 90, 90, 64)	0
block2_conv1 (Conv2D)	(None, 90, 90, 128)	73856
block2_conv2 (Conv2D)	(None, 90, 90, 128)	147584
block2_pool (MaxPooling2D)	(None, 45, 45, 128)	0
block3_conv1 (Conv2D)	(None, 45, 45, 256)	295168
block3_conv2 (Conv2D)	(None, 45, 45, 256)	590080
block3_conv3 (Conv2D)	(None, 45, 45, 256)	590080
block3_pool (MaxPooling2D)	(None, 22, 22, 256)	0
block4_conv1 (Conv2D)	(None, 22, 22, 512)	1180160
block4_conv2 (Conv2D)	(None, 22, 22, 512)	2359808
block4_conv3 (Conv2D)	(None, 22, 22, 512)	2359808
block4_pool (MaxPooling2D)	(None, 11, 11, 512)	0
block5_conv1 (Conv2D)	(None, 11, 11, 512)	2359808
block5_conv2 (Conv2D)	(None, 11, 11, 512)	2359808
block5_conv3 (Conv2D)	(None, 11, 11, 512)	2359808
block5_pool (MaxPooling2D)	(None, 5, 5, 512)	0
=====		
Total params:	14,714,688	
Trainable params:	14,714,688	
Non-trainable params:	0	

```
In [ ]: vgg16_model = tf.keras.Sequential([
    vgg16_base_model,
    GlobalAveragePooling2D(),
    Dense(512, activation="relu"),
    BatchNormalization(),
    Dropout(0.6),
    Dense(128, activation="relu"),
    BatchNormalization(),
    Dropout(0.4),
    Dense(64, activation="relu"),
    BatchNormalization(),
    Dropout(0.3),
```

```
Dense(1,activation="sigmoid")  
])
```

```
In [ ]: opt = tf.keras.optimizers.Adam(learning_rate=0.001)  
METRICS = [  
    'accuracy',  
    tf.keras.metrics.Precision(name='precision'),  
    tf.keras.metrics.Recall(name='recall')  
]  
vgg16_model.compile(optimizer=opt,loss='binary_crossentropy',metrics=METRICS)
```

```
In [ ]: r = vgg16_model.fit(train,  
                           epochs=10,  
                           validation_data=validation,  
                           class_weight=class_weight,  
                           steps_per_epoch=100,  
                           validation_steps=25)
```

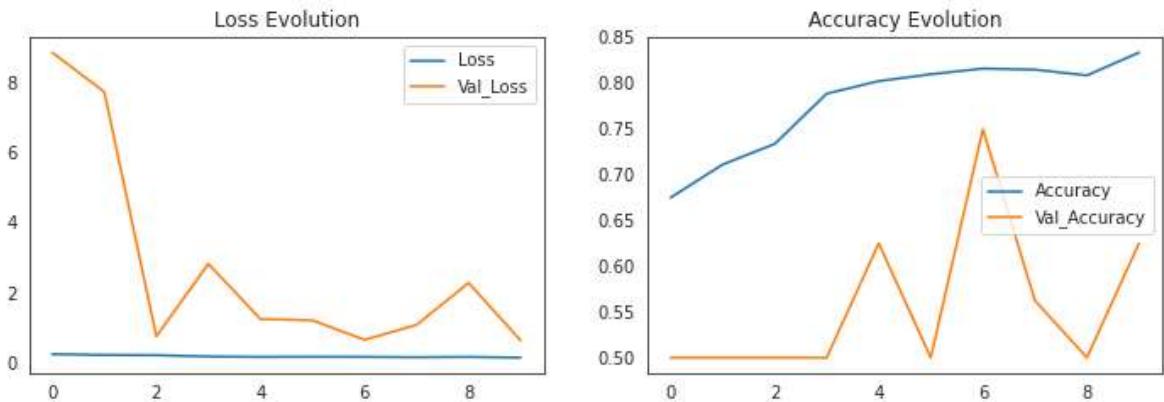
```
Epoch 1/10  
100/100 [=====] - 17s 171ms/step - loss: 0.2456 - accuracy: 0.6750 - precision: 0.8983 - recall: 0.6231 - val_loss: 8.8623 - val_accuracy: 0.5000 - val_precision: 0.0000e+00 - val_recall: 0.0000e+00  
Epoch 2/10  
100/100 [=====] - 18s 176ms/step - loss: 0.2209 - accuracy: 0.7113 - precision: 0.9064 - recall: 0.6949 - val_loss: 7.7409 - val_accuracy: 0.5000 - val_precision: 0.0000e+00 - val_recall: 0.0000e+00  
Epoch 3/10  
100/100 [=====] - 16s 160ms/step - loss: 0.2153 - accuracy: 0.7337 - precision: 0.9087 - recall: 0.7097 - val_loss: 0.7526 - val_accuracy: 0.5000 - val_precision: 0.5000 - val_recall: 0.1250  
Epoch 4/10  
100/100 [=====] - 18s 178ms/step - loss: 0.1775 - accuracy: 0.7887 - precision: 0.9350 - recall: 0.7637 - val_loss: 2.8258 - val_accuracy: 0.5000 - val_precision: 0.5000 - val_recall: 1.0000  
Epoch 5/10  
100/100 [=====] - 16s 164ms/step - loss: 0.1652 - accuracy: 0.8025 - precision: 0.9455 - recall: 0.7813 - val_loss: 1.2478 - val_accuracy: 0.6250 - val_precision: 0.5714 - val_recall: 1.0000  
Epoch 6/10  
100/100 [=====] - 18s 175ms/step - loss: 0.1692 - accuracy: 0.8100 - precision: 0.9398 - recall: 0.7932 - val_loss: 1.2110 - val_accuracy: 0.5000 - val_precision: 0.0000e+00 - val_recall: 0.0000e+00  
Epoch 7/10  
100/100 [=====] - 16s 164ms/step - loss: 0.1659 - accuracy: 0.8163 - precision: 0.9328 - recall: 0.8068 - val_loss: 0.6551 - val_accuracy: 0.7500 - val_precision: 0.7000 - val_recall: 0.8750  
Epoch 8/10  
100/100 [=====] - 17s 172ms/step - loss: 0.1571 - accuracy: 0.8150 - precision: 0.9482 - recall: 0.7883 - val_loss: 1.0808 - val_accuracy: 0.5625 - val_precision: 0.5333 - val_recall: 1.0000  
Epoch 9/10  
100/100 [=====] - 17s 166ms/step - loss: 0.1659 - accuracy: 0.8087 - precision: 0.9453 - recall: 0.7875 - val_loss: 2.2807 - val_accuracy: 0.5000 - val_precision: 0.0000e+00 - val_recall: 0.0000e+00  
Epoch 10/10  
100/100 [=====] - 17s 170ms/step - loss: 0.1448 - accuracy: 0.8338 - precision: 0.9598 - recall: 0.8176 - val_loss: 0.6366 - val_accuracy: 0.6250 - val_precision: 0.7500 - val_recall: 0.3750
```

```
In [ ]: plt.figure(figsize=(12, 8))

plt.subplot(2, 2, 1)
plt.plot(r.history['loss'], label='Loss')
plt.plot(r.history['val_loss'], label='Val_Loss')
plt.legend()
plt.title('Loss Evolution')

plt.subplot(2, 2, 2)
plt.plot(r.history['accuracy'], label='Accuracy')
plt.plot(r.history['val_accuracy'], label='Val_Accuracy')
plt.legend()
plt.title('Accuracy Evolution')
```

Out[]: Text(0.5, 1.0, 'Accuracy Evolution')



```
In [ ]: evaluation = vgg16_model.evaluate(test)
print(f"Test Accuracy: {evaluation[1] * 100:.2f}%")
```

```
evaluation = vgg16_model.evaluate(train)
print(f"Train Accuracy: {evaluation[1] * 100:.2f}%")
```

```
624/624 [=====] - 13s 21ms/step - loss: 0.8073 - accuracy: 0.6571 - precision: 0.9272 - recall: 0.4897
Test Accuracy: 65.71%
652/652 [=====] - 99s 152ms/step - loss: 0.9043 - accuracy: 0.6181 - precision: 0.9963 - recall: 0.4877
Train Accuracy: 61.81%
```

ResNet

See the full explanation and schemes in the Research Paper on Deep Residual Learning (<https://arxiv.org/pdf/1512.03385.pdf>)

```
In [ ]: from keras.applications import ResNet50

resnet_base_model = ResNet50(input_shape=(180,180,3), include_top=False, weights
```

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
94773248/94765736 [=====] - 1s 0us/step

```
In [ ]: resnet_base_model.summary()
```

Model: "resnet50"

Layer (type)	Output Shape	Param #	Connected to
<hr/>			
input_4 (InputLayer)	[(None, 180, 180, 3) 0		
conv1_pad (ZeroPadding2D)	(None, 186, 186, 3) 0		input_4[0][0]
conv1_conv (Conv2D)	(None, 90, 90, 64)	9472	conv1_pad[0][0]
conv1_bn (BatchNormalization)	(None, 90, 90, 64)	256	conv1_conv[0][0]
conv1_relu (Activation)	(None, 90, 90, 64) 0		conv1_bn[0][0]
pool1_pad (ZeroPadding2D)	(None, 92, 92, 64) 0		conv1_relu[0][0]
pool1_pool (MaxPooling2D)	(None, 45, 45, 64) 0		pool1_pad[0][0]
conv2_block1_1_conv (Conv2D)	(None, 45, 45, 64)	4160	pool1_pool[0][0]
conv2_block1_1_bn (BatchNormali onv[0][0]	(None, 45, 45, 64)	256	conv2_block1_1_c
conv2_block1_1_relu (Activation n[0][0]	(None, 45, 45, 64) 0		conv2_block1_1_b
conv2_block1_2_conv (Conv2D) elu[0][0]	(None, 45, 45, 64)	36928	conv2_block1_1_r
conv2_block1_2_bn (BatchNormali onv[0][0]	(None, 45, 45, 64)	256	conv2_block1_2_c
conv2_block1_2_relu (Activation n[0][0]	(None, 45, 45, 64) 0		conv2_block1_2_b
conv2_block1_0_conv (Conv2D)	(None, 45, 45, 256)	16640	pool1_pool[0][0]
conv2_block1_3_conv (Conv2D) elu[0][0]	(None, 45, 45, 256)	16640	conv2_block1_2_r
conv2_block1_0_bn (BatchNormali onv[0][0]	(None, 45, 45, 256)	1024	conv2_block1_0_c

conv2_block1_3_bn	(BatchNormali onv[0][0]	(None, 45, 45, 256)	1024	conv2_block1_3_c
conv2_block1_add	(Add) n[0][0]	(None, 45, 45, 256)	0	conv2_block1_0_b conv2_block1_3_b
conv2_block1_out	(Activation) [0][0]	(None, 45, 45, 256)	0	conv2_block1_add
conv2_block2_1_conv	(Conv2D) [0][0]	(None, 45, 45, 64)	16448	conv2_block1_out
conv2_block2_1_bn	(BatchNormali onv[0][0]	(None, 45, 45, 64)	256	conv2_block2_1_c
conv2_block2_1_relu	(Activation (None, 45, 45, 64) n[0][0]	0		conv2_block2_1_b
conv2_block2_2_conv	(Conv2D) elu[0][0]	(None, 45, 45, 64)	36928	conv2_block2_1_r
conv2_block2_2_bn	(BatchNormali onv[0][0]	(None, 45, 45, 64)	256	conv2_block2_2_c
conv2_block2_2_relu	(Activation (None, 45, 45, 64) n[0][0]	0		conv2_block2_2_b
conv2_block2_3_conv	(Conv2D) elu[0][0]	(None, 45, 45, 256)	16640	conv2_block2_2_r
conv2_block2_3_bn	(BatchNormali onv[0][0]	(None, 45, 45, 256)	1024	conv2_block2_3_c
conv2_block2_add	(Add) n[0][0]	(None, 45, 45, 256)	0	conv2_block1_out conv2_block2_3_b
conv2_block2_out	(Activation) [0][0]	(None, 45, 45, 256)	0	conv2_block2_add
conv2_block3_1_conv	(Conv2D) [0][0]	(None, 45, 45, 64)	16448	conv2_block2_out

conv2_block3_1_bn (BatchNormali (None, 45, 45, 64) 256 onv[0][0]		conv2_block3_1_c
conv2_block3_1_relu (Activation (None, 45, 45, 64) 0 n[0][0]		conv2_block3_1_b
conv2_block3_2_conv (Conv2D) (None, 45, 45, 64) 36928 elu[0][0]		conv2_block3_1_r
conv2_block3_2_bn (BatchNormali (None, 45, 45, 64) 256 onv[0][0]		conv2_block3_2_c
conv2_block3_2_relu (Activation (None, 45, 45, 64) 0 n[0][0]		conv2_block3_2_b
conv2_block3_3_conv (Conv2D) (None, 45, 45, 256) 16640 elu[0][0]		conv2_block3_2_r
conv2_block3_3_bn (BatchNormali (None, 45, 45, 256) 1024 onv[0][0]		conv2_block3_3_c
conv2_block3_add (Add) (None, 45, 45, 256) 0 [0][0]		conv2_block2_out
n[0][0]		conv2_block3_3_b
conv2_block3_out (Activation) (None, 45, 45, 256) 0 [0][0]		conv2_block3_add
conv3_block1_1_conv (Conv2D) (None, 23, 23, 128) 32896 [0][0]		conv2_block3_out
conv3_block1_1_bn (BatchNormali (None, 23, 23, 128) 512 onv[0][0]		conv3_block1_1_c
conv3_block1_1_relu (Activation (None, 23, 23, 128) 0 n[0][0]		conv3_block1_1_b
conv3_block1_2_conv (Conv2D) (None, 23, 23, 128) 147584 elu[0][0]		conv3_block1_1_r
conv3_block1_2_bn (BatchNormali (None, 23, 23, 128) 512 onv[0][0]		conv3_block1_2_c
conv3_block1_2_relu (Activation (None, 23, 23, 128) 0 conv3_block1_2_b		conv3_block1_2_b

n[0][0]

conv3_block1_0_conv (Conv2D) (None, 23, 23, 512) 131584 conv2_block3_out
[0][0]

conv3_block1_3_conv (Conv2D) (None, 23, 23, 512) 66048 conv3_block1_2_r
elu[0][0]

conv3_block1_0_bn (BatchNormali (None, 23, 23, 512) 2048 conv3_block1_0_c
onv[0][0]

conv3_block1_3_bn (BatchNormali (None, 23, 23, 512) 2048 conv3_block1_3_c
onv[0][0]

conv3_block1_add (Add) (None, 23, 23, 512) 0 conv3_block1_0_b
n[0][0] conv3_block1_3_b
n[0][0]

conv3_block1_out (Activation) (None, 23, 23, 512) 0 conv3_block1_add
[0][0]

conv3_block2_1_conv (Conv2D) (None, 23, 23, 128) 65664 conv3_block1_out
[0][0]

conv3_block2_1_bn (BatchNormali (None, 23, 23, 128) 512 conv3_block2_1_c
onv[0][0]

conv3_block2_1_relu (Activation (None, 23, 23, 128) 0 conv3_block2_1_b
n[0][0]

conv3_block2_2_conv (Conv2D) (None, 23, 23, 128) 147584 conv3_block2_1_r
elu[0][0]

conv3_block2_2_bn (BatchNormali (None, 23, 23, 128) 512 conv3_block2_2_c
onv[0][0]

conv3_block2_2_relu (Activation (None, 23, 23, 128) 0 conv3_block2_2_b
n[0][0]

conv3_block2_3_conv (Conv2D) (None, 23, 23, 512) 66048 conv3_block2_2_r
elu[0][0]

conv3_block2_3_bn (BatchNormali (None, 23, 23, 512) 2048 conv3_block2_3_c
onv[0][0]

conv3_block2_add	(Add) [0][0]	(None, 23, 23, 512) 0	conv3_block1_out conv3_block2_3_b
conv3_block2_out	(Activation) [0][0]	(None, 23, 23, 512) 0	conv3_block2_add
conv3_block3_1_conv	(Conv2D) [0][0]	(None, 23, 23, 128) 65664	conv3_block2_out
conv3_block3_1_bn	(BatchNormali onv[0][0]	(None, 23, 23, 128) 512	conv3_block3_1_c
conv3_block3_1_relu	(Activation n[0][0]	(None, 23, 23, 128) 0	conv3_block3_1_b
conv3_block3_2_conv	(Conv2D) elu[0][0]	(None, 23, 23, 128) 147584	conv3_block3_1_r
conv3_block3_2_bn	(BatchNormali onv[0][0]	(None, 23, 23, 128) 512	conv3_block3_2_c
conv3_block3_2_relu	(Activation n[0][0]	(None, 23, 23, 128) 0	conv3_block3_2_b
conv3_block3_3_conv	(Conv2D) elu[0][0]	(None, 23, 23, 512) 66048	conv3_block3_2_r
conv3_block3_3_bn	(BatchNormali onv[0][0]	(None, 23, 23, 512) 2048	conv3_block3_3_c
conv3_block3_add	(Add) n[0][0]	(None, 23, 23, 512) 0	conv3_block2_out conv3_block3_3_b
conv3_block3_out	(Activation) [0][0]	(None, 23, 23, 512) 0	conv3_block3_add
conv3_block4_1_conv	(Conv2D) [0][0]	(None, 23, 23, 128) 65664	conv3_block3_out
conv3_block4_1_bn	(BatchNormali onv[0][0]	(None, 23, 23, 128) 512	conv3_block4_1_c

conv3_block4_1_relu (Activation (None, 23, 23, 128) 0 n[0][0]		conv3_block4_1_b
conv3_block4_2_conv (Conv2D) (None, 23, 23, 128) 147584 elu[0][0]		conv3_block4_1_r
conv3_block4_2_bn (BatchNormali (None, 23, 23, 128) 512 onv[0][0]		conv3_block4_2_c
conv3_block4_2_relu (Activation (None, 23, 23, 128) 0 n[0][0]		conv3_block4_2_b
conv3_block4_3_conv (Conv2D) (None, 23, 23, 512) 66048 elu[0][0]		conv3_block4_2_r
conv3_block4_3_bn (BatchNormali (None, 23, 23, 512) 2048 onv[0][0]		conv3_block4_3_c
conv3_block4_add (Add) (None, 23, 23, 512) 0 [0][0]		conv3_block3_out
n[0][0]		conv3_block4_3_b
conv3_block4_out (Activation) (None, 23, 23, 512) 0 [0][0]		conv3_block4_add
conv4_block1_1_conv (Conv2D) (None, 12, 12, 256) 131328 [0][0]		conv3_block4_out
conv4_block1_1_bn (BatchNormali (None, 12, 12, 256) 1024 onv[0][0]		conv4_block1_1_c
conv4_block1_1_relu (Activation (None, 12, 12, 256) 0 n[0][0]		conv4_block1_1_b
conv4_block1_2_conv (Conv2D) (None, 12, 12, 256) 590080 elu[0][0]		conv4_block1_1_r
conv4_block1_2_bn (BatchNormali (None, 12, 12, 256) 1024 onv[0][0]		conv4_block1_2_c
conv4_block1_2_relu (Activation (None, 12, 12, 256) 0 n[0][0]		conv4_block1_2_b
conv4_block1_0_conv (Conv2D) (None, 12, 12, 1024) 525312 [0][0]		conv3_block4_out

[0][0]

conv4_block1_3_conv (Conv2D) (None, 12, 12, 1024) 263168 conv4_block1_2_r
elu[0][0]

conv4_block1_0_bn (BatchNormali (None, 12, 12, 1024) 4096 conv4_block1_0_c
onv[0][0]

conv4_block1_3_bn (BatchNormali (None, 12, 12, 1024) 4096 conv4_block1_3_c
onv[0][0]

conv4_block1_add (Add) (None, 12, 12, 1024) 0 conv4_block1_0_b
n[0][0] conv4_block1_3_b
n[0][0]

conv4_block1_out (Activation) (None, 12, 12, 1024) 0 conv4_block1_add
[0][0]

conv4_block2_1_conv (Conv2D) (None, 12, 12, 256) 262400 conv4_block1_out
[0][0]

conv4_block2_1_bn (BatchNormali (None, 12, 12, 256) 1024 conv4_block2_1_c
onv[0][0]

conv4_block2_1_relu (Activation (None, 12, 12, 256) 0 conv4_block2_1_b
n[0][0]

conv4_block2_2_conv (Conv2D) (None, 12, 12, 256) 590080 conv4_block2_1_r
elu[0][0]

conv4_block2_2_bn (BatchNormali (None, 12, 12, 256) 1024 conv4_block2_2_c
onv[0][0]

conv4_block2_2_relu (Activation (None, 12, 12, 256) 0 conv4_block2_2_b
n[0][0]

conv4_block2_3_conv (Conv2D) (None, 12, 12, 1024) 263168 conv4_block2_2_r
elu[0][0]

conv4_block2_3_bn (BatchNormali (None, 12, 12, 1024) 4096 conv4_block2_3_c
onv[0][0]

conv4_block2_add (Add) (None, 12, 12, 1024) 0 conv4_block1_out
[0][0] conv4_block2_3_b

n[0][0]

conv4_block2_out (Activation) (None, 12, 12, 1024) 0 conv4_block2_add [0][0]

conv4_block3_1_conv (Conv2D) (None, 12, 12, 256) 262400 conv4_block2_out [0][0]

conv4_block3_1_bn (BatchNormali (None, 12, 12, 256) 1024 conv4_block3_1_c onv[0][0]

conv4_block3_1_relu (Activation (None, 12, 12, 256) 0 conv4_block3_1_b n[0][0]

conv4_block3_2_conv (Conv2D) (None, 12, 12, 256) 590080 conv4_block3_1_r elu[0][0]

conv4_block3_2_bn (BatchNormali (None, 12, 12, 256) 1024 conv4_block3_2_c onv[0][0]

conv4_block3_2_relu (Activation (None, 12, 12, 256) 0 conv4_block3_2_b n[0][0]

conv4_block3_3_conv (Conv2D) (None, 12, 12, 1024) 263168 conv4_block3_2_r elu[0][0]

conv4_block3_3_bn (BatchNormali (None, 12, 12, 1024) 4096 conv4_block3_3_c onv[0][0]

conv4_block3_add (Add) (None, 12, 12, 1024) 0 conv4_block2_out [0][0] conv4_block3_3_b n[0][0]

conv4_block3_out (Activation) (None, 12, 12, 1024) 0 conv4_block3_add [0][0]

conv4_block4_1_conv (Conv2D) (None, 12, 12, 256) 262400 conv4_block3_out [0][0]

conv4_block4_1_bn (BatchNormali (None, 12, 12, 256) 1024 conv4_block4_1_c onv[0][0]

conv4_block4_1_relu (Activation (None, 12, 12, 256) 0 conv4_block4_1_b n[0][0]

conv4_block4_2_conv (Conv2D) (None, 12, 12, 256) 590080	conv4_block4_1_r elu[0][0]
conv4_block4_2_bn (BatchNormali (None, 12, 12, 256) 1024	conv4_block4_2_c onv[0][0]
conv4_block4_2_relu (Activation (None, 12, 12, 256) 0	conv4_block4_2_b n[0][0]
conv4_block4_3_conv (Conv2D) (None, 12, 12, 1024) 263168	conv4_block4_2_r elu[0][0]
conv4_block4_3_bn (BatchNormali (None, 12, 12, 1024) 4096	conv4_block4_3_c onv[0][0]
conv4_block4_add (Add) (None, 12, 12, 1024) 0	conv4_block3_out [0][0]
	conv4_block4_3_b n[0][0]
conv4_block4_out (Activation) (None, 12, 12, 1024) 0	conv4_block4_add [0][0]
conv4_block5_1_conv (Conv2D) (None, 12, 12, 256) 262400	conv4_block4_out [0][0]
conv4_block5_1_bn (BatchNormali (None, 12, 12, 256) 1024	conv4_block5_1_c onv[0][0]
conv4_block5_1_relu (Activation (None, 12, 12, 256) 0	conv4_block5_1_b n[0][0]
conv4_block5_2_conv (Conv2D) (None, 12, 12, 256) 590080	conv4_block5_1_r elu[0][0]
conv4_block5_2_bn (BatchNormali (None, 12, 12, 256) 1024	conv4_block5_2_c onv[0][0]
conv4_block5_2_relu (Activation (None, 12, 12, 256) 0	conv4_block5_2_b n[0][0]
conv4_block5_3_conv (Conv2D) (None, 12, 12, 1024) 263168	conv4_block5_2_r elu[0][0]
conv4_block5_3_bn (BatchNormali (None, 12, 12, 1024) 4096	conv4_block5_3_c onv[0][0]

onv[0][0]			
conv4_block5_add [0][0]	(Add) (None, 12, 12, 1024) 0	conv4_block4_out	conv4_block5_3_b
n[0][0]			
conv4_block5_out [0][0]	(Activation) (None, 12, 12, 1024) 0	conv4_block5_add	
conv4_block6_1_conv [0][0]	(Conv2D) (None, 12, 12, 256) 262400	conv4_block5_out	
conv4_block6_1_bn onv[0][0]	(BatchNormali (None, 12, 12, 256) 1024	conv4_block6_1_c	
conv4_block6_1_relu n[0][0]	(Activation (None, 12, 12, 256) 0	conv4_block6_1_b	
conv4_block6_2_conv elu[0][0]	(Conv2D) (None, 12, 12, 256) 590080	conv4_block6_1_r	
conv4_block6_2_bn onv[0][0]	(BatchNormali (None, 12, 12, 256) 1024	conv4_block6_2_c	
conv4_block6_2_relu n[0][0]	(Activation (None, 12, 12, 256) 0	conv4_block6_2_b	
conv4_block6_3_conv elu[0][0]	(Conv2D) (None, 12, 12, 1024) 263168	conv4_block6_2_r	
conv4_block6_3_bn onv[0][0]	(BatchNormali (None, 12, 12, 1024) 4096	conv4_block6_3_c	
conv4_block6_add [0][0]	(Add) (None, 12, 12, 1024) 0	conv4_block5_out	
n[0][0]			conv4_block6_3_b
conv4_block6_out [0][0]	(Activation) (None, 12, 12, 1024) 0	conv4_block6_add	
conv5_block1_1_conv [0][0]	(Conv2D) (None, 6, 6, 512) 524800	conv4_block6_out	
conv5_block1_1_bn onv[0][0]	(BatchNormali (None, 6, 6, 512) 2048	conv5_block1_1_c	

onv[0][0]

conv5_block1_1_relu (Activation (None, 6, 6, 512) 0 conv5_block1_1_b
n[0][0]

conv5_block1_2_conv (Conv2D) (None, 6, 6, 512) 2359808 conv5_block1_1_r
elu[0][0]

conv5_block1_2_bn (BatchNormali (None, 6, 6, 512) 2048 conv5_block1_2_c
onv[0][0]

conv5_block1_2_relu (Activation (None, 6, 6, 512) 0 conv5_block1_2_b
n[0][0]

conv5_block1_0_conv (Conv2D) (None, 6, 6, 2048) 2099200 conv4_block6_out
[0][0]

conv5_block1_3_conv (Conv2D) (None, 6, 6, 2048) 1050624 conv5_block1_2_r
elu[0][0]

conv5_block1_0_bn (BatchNormali (None, 6, 6, 2048) 8192 conv5_block1_0_c
onv[0][0]

conv5_block1_3_bn (BatchNormali (None, 6, 6, 2048) 8192 conv5_block1_3_c
onv[0][0]

conv5_block1_add (Add) (None, 6, 6, 2048) 0 conv5_block1_0_b
n[0][0] conv5_block1_3_b
n[0][0]

conv5_block1_out (Activation) (None, 6, 6, 2048) 0 conv5_block1_add
[0][0]

conv5_block2_1_conv (Conv2D) (None, 6, 6, 512) 1049088 conv5_block1_out
[0][0]

conv5_block2_1_bn (BatchNormali (None, 6, 6, 512) 2048 conv5_block2_1_c
onv[0][0]

conv5_block2_1_relu (Activation (None, 6, 6, 512) 0 conv5_block2_1_b
n[0][0]

conv5_block2_2_conv (Conv2D) (None, 6, 6, 512) 2359808 conv5_block2_1_r
elu[0][0]

conv5_block2_2_bn (BatchNormali onv[0][0]	(None, 6, 6, 512)	2048	conv5_block2_2_c
conv5_block2_2_relu (Activation n[0][0]	(None, 6, 6, 512)	0	conv5_block2_2_b
conv5_block2_3_conv (Conv2D elu[0][0]	(None, 6, 6, 2048)	1050624	conv5_block2_2_r
conv5_block2_3_bn (BatchNormali onv[0][0]	(None, 6, 6, 2048)	8192	conv5_block2_3_c
conv5_block2_add (Add n[0][0]	(None, 6, 6, 2048)	0	conv5_block1_out conv5_block2_3_b
conv5_block2_out (Activation) [0][0]	(None, 6, 6, 2048)	0	conv5_block2_add
conv5_block3_1_conv (Conv2D [0][0]	(None, 6, 6, 512)	1049088	conv5_block2_out
conv5_block3_1_bn (BatchNormali onv[0][0]	(None, 6, 6, 512)	2048	conv5_block3_1_c
conv5_block3_1_relu (Activation n[0][0]	(None, 6, 6, 512)	0	conv5_block3_1_b
conv5_block3_2_conv (Conv2D elu[0][0]	(None, 6, 6, 512)	2359808	conv5_block3_1_r
conv5_block3_2_bn (BatchNormali onv[0][0]	(None, 6, 6, 512)	2048	conv5_block3_2_c
conv5_block3_2_relu (Activation n[0][0]	(None, 6, 6, 512)	0	conv5_block3_2_b
conv5_block3_3_conv (Conv2D elu[0][0]	(None, 6, 6, 2048)	1050624	conv5_block3_2_r
conv5_block3_3_bn (BatchNormali onv[0][0]	(None, 6, 6, 2048)	8192	conv5_block3_3_c
conv5_block3_add (Add)	(None, 6, 6, 2048)	0	conv5_block2_out

```
[0][0]                                     conv5_block3_3_b
n[0][0]



---


conv5_block3_out (Activation)  (None, 6, 6, 2048)  0           conv5_block3_add
[0][0]
=====
=====
Total params: 23,587,712
Trainable params: 23,534,592
Non-trainable params: 53,120


---


```

```
In [ ]: resnet_model = tf.keras.Sequential([
    resnet_base_model,
    GlobalAveragePooling2D(),
    Dense(512, activation="relu"),
    BatchNormalization(),
    Dropout(0.6),
    Dense(128, activation="relu"),
    BatchNormalization(),
    Dropout(0.4),
    Dense(64, activation="relu"),
    BatchNormalization(),
    Dropout(0.3),
    Dense(1,activation="sigmoid")
])

opt = tf.keras.optimizers.Adam(learning_rate=0.001)
METRICS = [
    'accuracy',
    tf.keras.metrics.Precision(name='precision'),
    tf.keras.metrics.Recall(name='recall')
]
resnet_model.compile(optimizer=opt, loss='binary_crossentropy', metrics=METRIC
```

```
In [ ]: r = resnet_model.fit(train,
                           epochs=10,
                           validation_data=validation,
                           class_weight=class_weight,
                           steps_per_epoch=100,
                           validation_steps=25)
```

```
Epoch 1/10
100/100 [=====] - 18s 175ms/step - loss: 0.2575 - accuracy: 0.6625 - precision: 0.8758 - recall: 0.6486 - val_loss: 5.4126 - val_accuracy: 0.5000 - val_precision: 0.0000e+00 - val_recall: 0.0000e+00
Epoch 2/10
100/100 [=====] - 18s 180ms/step - loss: 0.2642 - accuracy: 0.6662 - precision: 0.8556 - recall: 0.6650 - val_loss: 0.6702 - val_accuracy: 0.6250 - val_precision: 0.6250 - val_recall: 0.6250
Epoch 3/10
100/100 [=====] - 16s 163ms/step - loss: 0.1924 - accuracy: 0.7713 - precision: 0.9170 - recall: 0.7613 - val_loss: 0.9461 - val_accuracy: 0.5000 - val_precision: 0.5000 - val_recall: 1.0000
Epoch 4/10
100/100 [=====] - 18s 184ms/step - loss: 0.1954 - accuracy: 0.8000 - precision: 0.9138 - recall: 0.7958 - val_loss: 1.2042 - val_accuracy: 0.5000 - val_precision: 0.5000 - val_recall: 1.0000
Epoch 5/10
100/100 [=====] - 17s 168ms/step - loss: 0.1631 - accuracy: 0.8325 - precision: 0.9504 - recall: 0.8146 - val_loss: 0.8847 - val_accuracy: 0.5000 - val_precision: 0.5000 - val_recall: 1.0000
Epoch 6/10
100/100 [=====] - 18s 178ms/step - loss: 0.1678 - accuracy: 0.8037 - precision: 0.9219 - recall: 0.8014 - val_loss: 1.3853 - val_accuracy: 0.5000 - val_precision: 0.5000 - val_recall: 1.0000
Epoch 7/10
100/100 [=====] - 17s 170ms/step - loss: 0.1518 - accuracy: 0.8163 - precision: 0.9479 - recall: 0.8036 - val_loss: 7.8778 - val_accuracy: 0.5000 - val_precision: 0.5000 - val_recall: 1.0000
Epoch 8/10
100/100 [=====] - 18s 177ms/step - loss: 0.1586 - accuracy: 0.8163 - precision: 0.9369 - recall: 0.8113 - val_loss: 1.4738 - val_accuracy: 0.6250 - val_precision: 0.5714 - val_recall: 1.0000
Epoch 9/10
100/100 [=====] - 17s 172ms/step - loss: 0.1392 - accuracy: 0.8475 - precision: 0.9467 - recall: 0.8409 - val_loss: 1.2015 - val_accuracy: 0.5625 - val_precision: 0.5333 - val_recall: 1.0000
Epoch 10/10
100/100 [=====] - 18s 177ms/step - loss: 0.1313 - accuracy: 0.8537 - precision: 0.9673 - recall: 0.8342 - val_loss: 0.7883 - val_accuracy: 0.6875 - val_precision: 0.6364 - val_recall: 0.8750
```

```
In [ ]: plt.figure(figsize=(12, 8))

plt.subplot(2, 2, 1)
plt.plot(r.history['loss'], label='Loss')
plt.plot(r.history['val_loss'], label='Val_Loss')
plt.legend()
plt.title('Loss Evolution')

plt.subplot(2, 2, 2)
plt.plot(r.history['accuracy'], label='Accuracy')
plt.plot(r.history['val_accuracy'], label='Val_Accuracy')
plt.legend()
plt.title('Accuracy Evolution')
```

```
Out[ ]: Text(0.5, 1.0, 'Accuracy Evolution')
```



```

    class_weight=class_weight,
    steps_per_epoch=100,
    validation_steps=25)

Epoch 1/10
100/100 [=====] - 18s 182ms/step - loss: 0.2331 - accuracy: 0.7113 - precision: 0.8994 - recall: 0.6954 - val_loss: 0.4899 - val_accuracy: 0.7500 - val_precision: 0.7000 - val_recall: 0.8750
Epoch 2/10
100/100 [=====] - 19s 191ms/step - loss: 0.1994 - accuracy: 0.7850 - precision: 0.9289 - recall: 0.7694 - val_loss: 8.5411 - val_accuracy: 0.5000 - val_precision: 0.0000e+00 - val_recall: 0.0000e+00
Epoch 3/10
100/100 [=====] - 16s 158ms/step - loss: 0.1820 - accuracy: 0.7763 - precision: 0.9300 - recall: 0.7635 - val_loss: 12.8820 - val_accuracy: 0.6250 - val_precision: 0.7500 - val_recall: 0.3750
Epoch 4/10
100/100 [=====] - 20s 195ms/step - loss: 0.1638 - accuracy: 0.8062 - precision: 0.9461 - recall: 0.7874 - val_loss: 2.0006 - val_accuracy: 0.3125 - val_precision: 0.3333 - val_recall: 0.3750
Epoch 5/10
100/100 [=====] - 16s 165ms/step - loss: 0.1921 - accuracy: 0.7987 - precision: 0.9320 - recall: 0.7858 - val_loss: 55.1592 - val_accuracy: 0.4375 - val_precision: 0.4000 - val_recall: 0.2500
Epoch 6/10
100/100 [=====] - 19s 187ms/step - loss: 0.1999 - accuracy: 0.7738 - precision: 0.9234 - recall: 0.7621 - val_loss: 0.6640 - val_accuracy: 0.6250 - val_precision: 0.6667 - val_recall: 0.5000
Epoch 7/10
100/100 [=====] - 17s 168ms/step - loss: 0.1390 - accuracy: 0.8450 - precision: 0.9477 - recall: 0.8408 - val_loss: 5.5447 - val_accuracy: 0.6250 - val_precision: 0.6000 - val_recall: 0.7500
Epoch 8/10
100/100 [=====] - 18s 183ms/step - loss: 0.1661 - accuracy: 0.7987 - precision: 0.9437 - recall: 0.7864 - val_loss: 59.6781 - val_accuracy: 0.5000 - val_precision: 0.5000 - val_recall: 1.0000
Epoch 9/10
100/100 [=====] - 17s 166ms/step - loss: 0.1854 - accuracy: 0.8087 - precision: 0.9187 - recall: 0.8154 - val_loss: 2.0644 - val_accuracy: 0.5000 - val_precision: 0.5000 - val_recall: 0.8750
Epoch 10/10
100/100 [=====] - 19s 192ms/step - loss: 0.1718 - accuracy: 0.8100 - precision: 0.9247 - recall: 0.8091 - val_loss: 0.6752 - val_accuracy: 0.6875 - val_precision: 0.8000 - val_recall: 0.5000

```

```

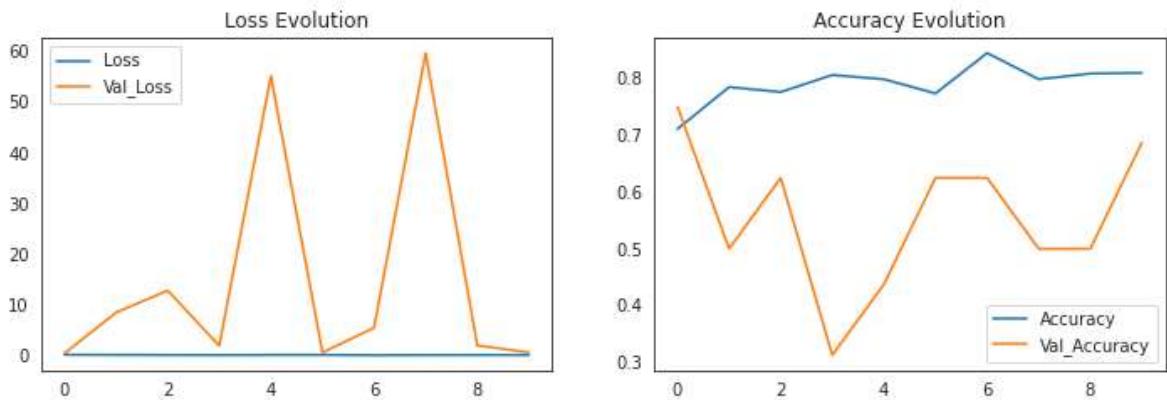
In [ ]: plt.figure(figsize=(12, 8))

plt.subplot(2, 2, 1)
plt.plot(r.history['loss'], label='Loss')
plt.plot(r.history['val_loss'], label='Val_Loss')
plt.legend()
plt.title('Loss Evolution')

plt.subplot(2, 2, 2)
plt.plot(r.history['accuracy'], label='Accuracy')
plt.plot(r.history['val_accuracy'], label='Val_Accuracy')
plt.legend()
plt.title('Accuracy Evolution')

```

Out[]: Text(0.5, 1.0, 'Accuracy Evolution')



```
In [ ]: evaluation = inception_model.evaluate(test)
print(f"Test Accuracy: {evaluation[1] * 100:.2f}%")

evaluation = inception_model.evaluate(train)
print(f"Train Accuracy: {evaluation[1] * 100:.2f}%")
```

624/624 [=====] - 16s 25ms/step - loss: 0.6258 - accuracy: 0.7051 - precision: 0.9120 - recall: 0.5846
Test Accuracy: 70.51%
652/652 [=====] - 103s 159ms/step - loss: 0.6924 - accuracy: 0.6904 - precision: 0.9922 - recall: 0.5879
Train Accuracy: 69.04%

```
In [ ]:
```