dataset = Crack 500

Loading the important libraries and Dataset

In [1]:

```
! nvidia-smi
```

```
Thu Mar 31 21:30:19 2022
+----------------------------------------------------------------------------
-+
| NVIDIA-SMI 470.103.01   Driver Version: 470.103.01   CUDA Version: 11.4
|
|-------------------------------+----------------------+---------------------
-+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC
|
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M.
|
|                               |                      |               MIG M.
|
|===============================+======================+=====================
=|
|   0  NVIDIA GeForce ...  Off  | 00000000:02:00.0 N/A |                  N/A
|
| 40%   33C    P0    N/A /  N/A |    348MiB /  4041MiB |     N/A      Default
|
|                               |                      |                  N/A
|
+-------------------------------+----------------------+---------------------
-+

+----------------------------------------------------------------------------
-+
| Processes:
|
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory
|
|        ID   ID                                                   Usage
|
|============================================================================
=|
|  No running processes found
|
+----------------------------------------------------------------------------
-+
```

In [2]:

```
import os
import cv2
import shutil
import math
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
sns.set()
```

In [3]:

```
import tensorflow as tf
from tensorflow import keras
import tensorflow.keras.backend as K
from tensorflow.keras.utils import Sequence
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Conv2D, BatchNormalization, Activa
from tensorflow.keras.losses import binary_crossentropy
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
```

In [4]:

```
from sklearn.metrics import classification_report, roc_auc_score, accuracy_sc
from albumentations import Compose, OneOf, Flip, Rotate, RandomContrast, Ranc
```

In [5]:

```
from tensorflow.keras.losses import binary_crossentropy
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from skimage.transform import resize
from sklearn.metrics import classification_report
```

In [6]:

```
import os
import zipfile
```

Loading the data and splitting it into training and validation set.

In [7]:

```
train_image_dir = r'/home/ubuntu/Desktop/NNDL Project/CRACK500/traincrop'
# train_mask_dir = r'/content/CRACK500/traindata/mask'

valid_image_dir = '/home/ubuntu/Desktop/NNDL Project/CRACK500/valcrop'
# valid_mask_dir = '/content/CRACK500/valdata/mask'

test_image_dir = '/home/ubuntu/Desktop/NNDL Project/CRACK500/testcrop'
# test_mask_dir = '/content/CRACK500/testdata/mask'
```

In [8]:

```
# test_image_dir = '/content/CRACK500_CROP/testcrop/image'
# test_mask_dir = '/content/CRACK500_CROP/testcrop/mask'
train_image_paths = sorted([os.path.join(train_image_dir, fname) for fname ir
train_mask_paths = sorted([os.path.join(train_image_dir, fname) for fname in
print("Number of training images : ", len(train_image_paths))
print("Number of training masks : ", len(train_mask_paths))

valid_image_paths = sorted([os.path.join(valid_image_dir, fname) for fname ir
valid_mask_paths = sorted([os.path.join(valid_image_dir, fname) for fname in
print("Number of validation images : ", len(valid_image_paths))
print("Number of validation masks : ", len(valid_mask_paths))

test_image_paths = sorted([os.path.join(test_image_dir, fname) for fname in c
test_mask_paths = sorted([os.path.join(test_image_dir, fname) for fname in os
print("Number of testing images : ", len(test_image_paths))
print("Number of testing masks : ", len(test_mask_paths))
```

```
Number of training images :  1896
Number of training masks :  1896
Number of validation images :  348
Number of validation masks :  348
Number of testing images :  1124
Number of testing masks :  1124
```

In [9]:

```
# Shuffle
import random
combined = list(zip(train_image_paths, train_mask_paths))
random.shuffle(combined)
train_image_paths[:], train_mask_paths[:] = zip(*combined)
```

In [10]:

```
# Splitting
train_image_files = train_image_paths
train_mask_files = train_mask_paths

valid_image_files = valid_image_paths
valid_mask_files = valid_mask_paths

print(len(train_image_files), len(train_mask_files))
print(len(valid_image_files), len(valid_mask_files))
```

```
1896 1896
348 348
```

In [11]:

```
batch_size = 5
img_dim=(256, 256)
```

Generator to load and augment the image batch wise

In [12]:

```
class Generator(Sequence):

  def __init__(self, x_set, y_set, batch_size=5, img_dim=(128, 128), augment=
      self.x = x_set
      self.y = y_set
      self.batch_size = batch_size
      self.img_dim = img_dim
      self.augment = augment

  def __len__(self):
      return math.ceil(len(self.x) / self.batch_size)

  augmentations = Compose(
    [
      Flip(p=0.7),
      Rotate(p=0.7),
      OneOf([
              RandomContrast(),
              RandomGamma(),
              RandomBrightness()
            ], p=0.3),
      OneOf([
              ElasticTransform(alpha=120, sigma=120 * 0.05, alpha_affine=120
              GridDistortion(),
              OpticalDistortion(distort_limit=2, shift_limit=0.5)
            ], p=0.3),
    ])

  def __getitem__(self, idx):
      batch_x = self.x[idx * self.batch_size:(idx + 1) * self.batch_size]
      batch_y = self.y[idx * self.batch_size:(idx + 1) * self.batch_size]

      batch_x = np.array([cv2.resize(cv2.cvtColor(cv2.imread(file_name, -1),
      batch_y = np.array([(cv2.resize(cv2.imread(file_name, -1), (self.img_di

      if self.augment is True:
        aug = [self.augmentations(image=i, mask=j) for i, j in zip(batch_x, b
        batch_x = np.array([i['image'] for i in aug])
        batch_y = np.array([j['mask'] for j in aug])
      batch_y = np.expand_dims(batch_y, -1)

      return batch_x/255, batch_y/1
```

```
/home/ubuntu/anaconda3/envs/nndl/lib/python3.9/site-packages/albumentations/a
ugmentations/transforms.py:1826: FutureWarning: This class has been deprecate
d. Please use RandomBrightnessContrast
  warnings.warn(
/home/ubuntu/anaconda3/envs/nndl/lib/python3.9/site-packages/albumentations/a
ugmentations/transforms.py:1800: FutureWarning: This class has been deprecate
d. Please use RandomBrightnessContrast
  warnings.warn(
```

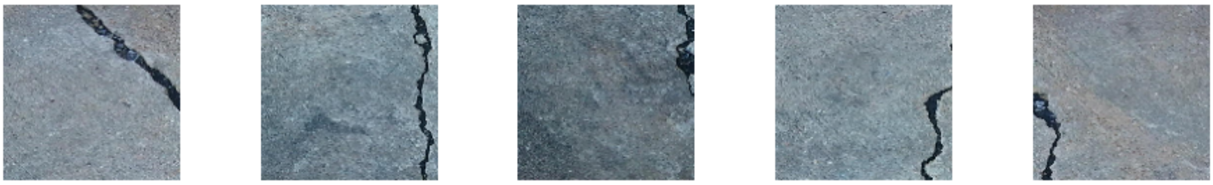
In [13]:

```
test1_generator=Generator(test_image_paths,test_mask_paths)
```

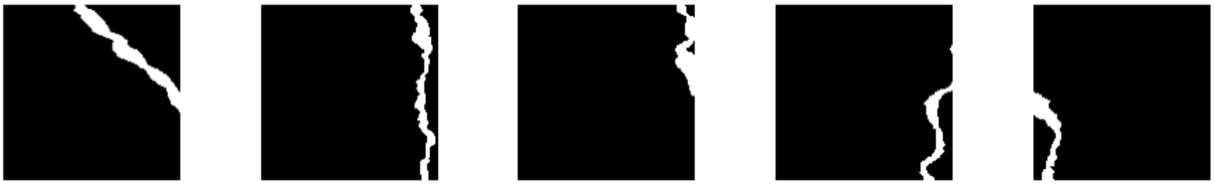
In [14]:

```
# Validation generator samples (Un-augmented)
for i, j in test1_generator:
    break

fig, axes = plt.subplots(1, 5, figsize=(13,2.5))
fig.suptitle('Original Images', fontsize=15)
axes = axes.flatten()
for img, ax in zip(i[:5], axes[:5]):
    ax.imshow(img)
    ax.axis('off')
plt.tight_layout()
plt.show()

fig, axes = plt.subplots(1, 5, figsize=(13,2.5))
fig.suptitle('Original Masks', fontsize=15)
axes = axes.flatten()
for img, ax in zip(j[:5], axes[:5]):
    ax.imshow(np.squeeze(img, -1), cmap='gray')
    ax.axis('off')
plt.tight_layout()
plt.show()
```

Original Images



Original Masks



In [15]:

```
train_generator = Generator(train_image_files, train_mask_files)
validation_generator = Generator(valid_image_files, valid_mask_files)
```

In [16]:

```
for i, j in train_generator:
  break
```

```
print(i.shape)
print(j.shape)
```

```
(5, 128, 128, 3)
(5, 128, 128, 1)
```

In [17]:

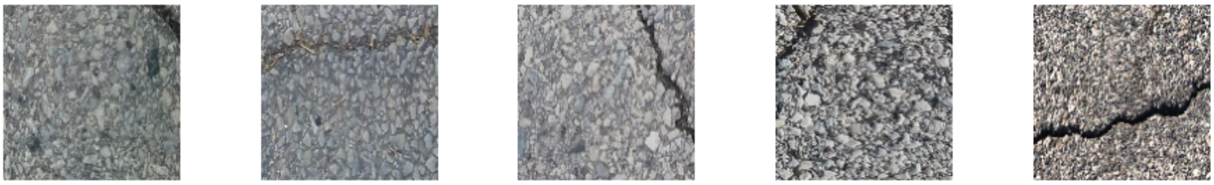
```
for i, j in validation_generator:
  break

print(i.shape)
print(j.shape)
```

```
(5, 128, 128, 3)
(5, 128, 128, 1)
```

In [18]:

```
# Train generator samples (Un-augmented)
for i, j in train_generator:
    break

fig, axes = plt.subplots(1, 5, figsize=(13,2.5))
fig.suptitle('Original Images', fontsize=15)
axes = axes.flatten()
for img, ax in zip(i[:5], axes[:5]):
    ax.imshow(img)
    ax.axis('off')
plt.tight_layout()
plt.show()

fig, axes = plt.subplots(1, 5, figsize=(13,2.5))
fig.suptitle('Original Masks', fontsize=15)
axes = axes.flatten()
for img, ax in zip(j[:5], axes[:5]):
    ax.imshow(np.squeeze(img, -1), cmap='gray')
    ax.axis('off')
plt.tight_layout()
plt.show()
```

Original Images



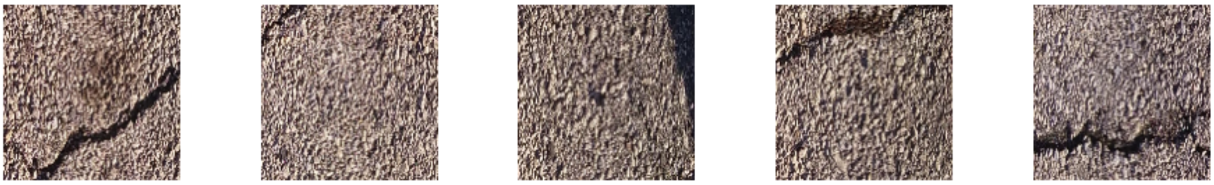
Original Masks

In [19]:

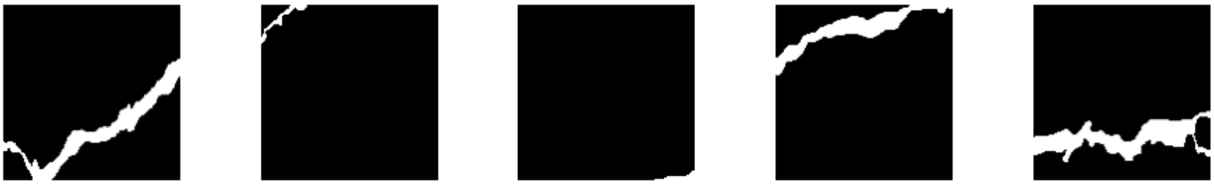
```
# Validation generator samples (Un-augmented)
for i, j in validation_generator:
    break

fig, axes = plt.subplots(1, 5, figsize=(13,2.5))
fig.suptitle('Original Images', fontsize=15)
axes = axes.flatten()
```

```
for img, ax in zip(i[:5], axes[:5]):
    ax.imshow(img)
    ax.axis('off')
plt.tight_layout()
plt.show()

fig, axes = plt.subplots(1, 5, figsize=(13,2.5))
fig.suptitle('Original Masks', fontsize=15)
axes = axes.flatten()
for img, ax in zip(j[:5], axes[:5]):
    ax.imshow(np.squeeze(img, -1), cmap='gray')
    ax.axis('off')
plt.tight_layout()
plt.show()
```

Original Images



Original Masks



In [20]:

```
tg = Generator(train_image_files, train_mask_files, batch_size, img_dim, augm
vg = Generator(valid_image_files, valid_mask_files, batch_size, img_dim, augm
```

In [21]:

```
for i, j in tg:
  break

print(i.shape)
print(j.shape)
```

```
(5, 256, 256, 3)
(5, 256, 256, 1)
```

In [22]:

```
for i, j in vg:
  break

print(i.shape)
print(j.shape)
```

```
(5, 256, 256, 3)
(5, 256, 256, 1)
```

In [23]:

```
# Augmented train
for i, j in tg:
    break

fig, axes = plt.subplots(1, 5, figsize=(13,2.5))
fig.suptitle('Augmented Images', fontsize=15)
axes = axes.flatten()
for img, ax in zip(i[:5], axes[:5]):
    ax.imshow(img)
    ax.axis('off')
plt.tight_layout()
plt.show()

fig, axes = plt.subplots(1, 5, figsize=(13,2.5))
fig.suptitle('Augmented Masks', fontsize=15)
axes = axes.flatten()
for img, ax in zip(j[:5], axes[:5]):
    ax.imshow(np.squeeze(img, -1), cmap='gray')
    ax.axis('off')
plt.tight_layout()
plt.show()
```

Augmented Images

Augmented Masks

Model

In [24]:

```
import numpy as np
from tensorflow.keras.backend import int_shape
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Conv2D, MaxPooling2D, UpSampling2D, Add,
from tensorflow.keras.regularizers import l2
```

In [25]:

```
# BatchNormalization and Activation
def BN_Act(x, act = True):
    x = BatchNormalization()(x)
    if act == True:
        x = Activation("relu")(x)
    return x
```

In [26]:

```
#conv2d block
def conv2d_block(x, filters, kernel_size = (3, 3), padding = "same", strides
    conv = BN_Act(x)
    conv = Conv2D(filters, kernel_size, padding = padding, strides = strides)
    return conv
```

In [27]:

```
#Fixed layer.
def stem(x, filters, kernel_size=(3, 3), padding="same", strides=1):
    conv = Conv2D(filters, kernel_size, padding = padding, strides = strides)
    conv = conv2d_block(conv, filters, kernel_size = kernel_size, padding = p

    #skip
    shortcut = Conv2D(filters, kernel_size = (1, 1), padding = padding, stric
    shortcut = BN_Act(shortcut, act = False) # No activation in skip connecti

    output = Add()([conv, shortcut])
    return output
```

In [28]:

```
# Residual Block
def residual_block(x, filters, kernel_size = (3, 3), padding = "same", stride
    res = conv2d_block(x, filters, kernel_size = kernel_size, padding = paddi
    res = conv2d_block(res, filters, kernel_size = kernel_size, padding = pac

    shortcut = Conv2D(filters, kernel_size = (1, 1), padding = padding, stric
    shortcut = BN_Act(shortcut, act = False) # No activation in skip connecti

    output = Add()([shortcut, res])
    return output
```

In [29]:

```
# Upsampling Concatenation block
def upsample_concat_block(x, xskip):
    u = UpSampling2D((2, 2))(x)
    c = Concatenate()([u, xskip])
    return c
```

In [30]:

```
# MODEL
def ResUNet():
    f = [16, 32, 64, 128, 256]
    inputs = Input((img_dim[0], img_dim[1], 3))

    ## Encoder/downsampling/contracting path
    e0 = inputs
    e1 = stem(e0, f[0])
    e2 = residual_block(e1, f[1], strides = 2)
    e3 = residual_block(e2, f[2], strides = 2)
    e4 = residual_block(e3, f[3], strides = 2)
    e5 = residual_block(e4, f[4], strides = 2)

    ## Bridge/Bottleneck
    b0 = conv2d_block(e5, f[4], strides = 1)
    b1 = conv2d_block(b0, f[4], strides = 1)

    ## Decoder/upsampling/expansive path
    u1 = upsample_concat_block(b1, e4)
    d1 = residual_block(u1, f[4])

    u2 = upsample_concat_block(d1, e3)
    d2 = residual_block(u2, f[3])

    u3 = upsample_concat_block(d2, e2)
    d3 = residual_block(u3, f[2])

    u4 = upsample_concat_block(d3, e1)
    d4 = residual_block(u4, f[1])

    outputs = Conv2D(1, (1, 1), padding = "same", activation = "sigmoid")(d4)
    model = Model(inputs, outputs)
    return model
```

In [32]:

```
K.clear_session()
model = ResUNet()
```

```
2022-03-31 21:31:44.369177: I tensorflow/compiler/jit/xla_cpu_device.cc:41] N
ot creating XLA devices, tf_xla_enable_xla_devices not set
2022-03-31 21:31:44.374157: I tensorflow/core/platform/cpu_feature_guard.cc:1
42] This TensorFlow binary is optimized with oneAPI Deep Neural Network Libra
ry (oneDNN) to use the following CPU instructions in performance-critical ope
rations:  SSE4.1 SSE4.2 AVX AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate c
ompiler flags.
2022-03-31 21:31:44.377238: I tensorflow/core/common_runtime/process_util.cc:
146] Creating new thread pool with default inter op setting: 2. Tune using in
ter_op_parallelism_threads for best performance.
```

In [33]:

```
model.summary()
```

```
Model: "model"
_____________________________________________________________________________
_____________________
Layer (type)                    Output Shape         Param #     Connected to
=============================================================================
=====================
input_1 (InputLayer)            [(None, 256, 256, 3) 0
_____________________________________________________________________________
_____________________
conv2d (Conv2D)                 (None, 256, 256, 16) 448         input_1[0]
[0]
_____________________________________________________________________________
_____________________
batch_normalization (BatchNorma (None, 256, 256, 16) 64          conv2d[0][0]
_____________________________________________________________________________
_____________________
activation (Activation)         (None, 256, 256, 16) 0           batch_normal
ization[0][0]
_____________________________________________________________________________
_____________________
conv2d_2 (Conv2D)               (None, 256, 256, 16) 64          input_1[0]
[0]
_____________________________________________________________________________
_____________________
conv2d_1 (Conv2D)               (None, 256, 256, 16) 2320        activation
[0][0]
_____________________________________________________________________________
_____________________
batch_normalization_1 (BatchNor (None, 256, 256, 16) 64          conv2d_2[0]
[0]
_____________________________________________________________________________
_____________________
add (Add)                       (None, 256, 256, 16) 0           conv2d_1[0]
[0]
                                                                 batch_normal
ization_1[0][0]
_____________________________________________________________________________
_____________________
batch_normalization_2 (BatchNor (None, 256, 256, 16) 64          add[0][0]
_____________________________________________________________________________
_____________________
activation_1 (Activation)       (None, 256, 256, 16) 0           batch_normal
ization_2[0][0]
_____________________________________________________________________________
_____________________
conv2d_3 (Conv2D)               (None, 128, 128, 32) 4640        activation_1
[0][0]
_____________________________________________________________________________
```

```
_____________________
batch_normalization_3 (BatchNor (None, 128, 128, 32) 128         conv2d_3[0]
[0]

_____________________________________________________________________________
_____________________
conv2d_5 (Conv2D)               (None, 128, 128, 32) 544         add[0][0]

_____________________________________________________________________________
_____________________
activation_2 (Activation)       (None, 128, 128, 32) 0           batch_normal
ization_3[0][0]

_____________________________________________________________________________
_____________________
batch_normalization_4 (BatchNor (None, 128, 128, 32) 128         conv2d_5[0]
[0]

_____________________________________________________________________________
_____________________
conv2d_4 (Conv2D)               (None, 128, 128, 32) 9248        activation_2
[0][0]

_____________________________________________________________________________
_____________________
add_1 (Add)                     (None, 128, 128, 32) 0           batch_normal
ization_4[0][0]
                                                                 conv2d_4[0]
[0]

_____________________________________________________________________________
_____________________
batch_normalization_5 (BatchNor (None, 128, 128, 32) 128         add_1[0][0]

_____________________________________________________________________________
_____________________
activation_3 (Activation)       (None, 128, 128, 32) 0           batch_normal
ization_5[0][0]

_____________________________________________________________________________
_____________________
conv2d_6 (Conv2D)               (None, 64, 64, 64)   18496       activation_3
[0][0]

_____________________________________________________________________________
_____________________
batch_normalization_6 (BatchNor (None, 64, 64, 64)   256         conv2d_6[0]
[0]

_____________________________________________________________________________
_____________________
conv2d_8 (Conv2D)               (None, 64, 64, 64)   2112        add_1[0][0]

_____________________________________________________________________________
_____________________
activation_4 (Activation)       (None, 64, 64, 64)   0           batch_normal
ization_6[0][0]

_____________________________________________________________________________
_____________________
batch_normalization_7 (BatchNor (None, 64, 64, 64)   256         conv2d_8[0]
[0]

_____________________________________________________________________________
_____________________
conv2d_7 (Conv2D)               (None, 64, 64, 64)   36928       activation_4
[0][0]

_____________________________________________________________________________
_____________________
add_2 (Add)                     (None, 64, 64, 64)   0           batch_normal
ization_7[0][0]
                                                                 conv2d_7[0]
[0]

_____________________________________________________________________________
_____________________
batch_normalization_8 (BatchNor (None, 64, 64, 64)   256         add_2[0][0]

_____________________________________________________________________________
```

```
_____________________
activation_5 (Activation)       (None, 64, 64, 64)   0           batch_normal
ization_8[0][0]
_____________________________________________________________________________
_____________________
conv2d_9 (Conv2D)               (None, 32, 32, 128)  73856       activation_5
[0][0]
_____________________________________________________________________________
_____________________
batch_normalization_9 (BatchNor (None, 32, 32, 128)  512         conv2d_9[0]
[0]
_____________________________________________________________________________
_____________________
conv2d_11 (Conv2D)              (None, 32, 32, 128)  8320        add_2[0][0]
_____________________________________________________________________________
_____________________
activation_6 (Activation)       (None, 32, 32, 128)  0           batch_normal
ization_9[0][0]
_____________________________________________________________________________
_____________________
batch_normalization_10 (BatchNo (None, 32, 32, 128)  512         conv2d_11[0]
[0]
_____________________________________________________________________________
_____________________
conv2d_10 (Conv2D)              (None, 32, 32, 128)  147584      activation_6
[0][0]
_____________________________________________________________________________
_____________________
add_3 (Add)                     (None, 32, 32, 128)  0           batch_normal
ization_10[0][0]
                                                                 conv2d_10[0]
[0]
_____________________________________________________________________________
_____________________
batch_normalization_11 (BatchNo (None, 32, 32, 128)  512         add_3[0][0]
_____________________________________________________________________________
_____________________
activation_7 (Activation)       (None, 32, 32, 128)  0           batch_normal
ization_11[0][0]
_____________________________________________________________________________
_____________________
conv2d_12 (Conv2D)              (None, 16, 16, 256)  295168      activation_7
[0][0]
_____________________________________________________________________________
_____________________
batch_normalization_12 (BatchNo (None, 16, 16, 256)  1024        conv2d_12[0]
[0]
_____________________________________________________________________________
_____________________
conv2d_14 (Conv2D)              (None, 16, 16, 256)  33024       add_3[0][0]
_____________________________________________________________________________
_____________________
activation_8 (Activation)       (None, 16, 16, 256)  0           batch_normal
ization_12[0][0]
_____________________________________________________________________________
_____________________
batch_normalization_13 (BatchNo (None, 16, 16, 256)  1024        conv2d_14[0]
[0]
_____________________________________________________________________________
_____________________
conv2d_13 (Conv2D)              (None, 16, 16, 256)  590080      activation_8
[0][0]
_____________________________________________________________________________
_____________________
```

```
add_4 (Add)                     (None, 16, 16, 256)  0           batch_normal
ization_13[0][0]
                                                                 conv2d_13[0]
[0]
_____________________________________________________________________________
_____________________
batch_normalization_14 (BatchNo (None, 16, 16, 256)  1024        add_4[0][0]
_____________________________________________________________________________
_____________________
activation_9 (Activation)       (None, 16, 16, 256)  0           batch_normal
ization_14[0][0]
_____________________________________________________________________________
_____________________
conv2d_15 (Conv2D)              (None, 16, 16, 256)  590080      activation_9
[0][0]
_____________________________________________________________________________
_____________________
batch_normalization_15 (BatchNo (None, 16, 16, 256)  1024        conv2d_15[0]
[0]
_____________________________________________________________________________
_____________________
activation_10 (Activation)      (None, 16, 16, 256)  0           batch_normal
ization_15[0][0]
_____________________________________________________________________________
_____________________
conv2d_16 (Conv2D)              (None, 16, 16, 256)  590080      activation_1
0[0][0]
_____________________________________________________________________________
_____________________
up_sampling2d (UpSampling2D)    (None, 32, 32, 256)  0           conv2d_16[0]
[0]
_____________________________________________________________________________
_____________________
concatenate (Concatenate)       (None, 32, 32, 384)  0           up_sampling2
d[0][0]
                                                                 add_3[0][0]
_____________________________________________________________________________
_____________________
batch_normalization_16 (BatchNo (None, 32, 32, 384)  1536        concatenate
[0][0]
_____________________________________________________________________________
_____________________
activation_11 (Activation)      (None, 32, 32, 384)  0           batch_normal
ization_16[0][0]
_____________________________________________________________________________
_____________________
conv2d_17 (Conv2D)              (None, 32, 32, 256)  884992      activation_1
1[0][0]
_____________________________________________________________________________
_____________________
batch_normalization_17 (BatchNo (None, 32, 32, 256)  1024        conv2d_17[0]
[0]
_____________________________________________________________________________
_____________________
conv2d_19 (Conv2D)              (None, 32, 32, 256)  98560       concatenate
[0][0]
_____________________________________________________________________________
_____________________
activation_12 (Activation)      (None, 32, 32, 256)  0           batch_normal
ization_17[0][0]
_____________________________________________________________________________
_____________________
batch_normalization_18 (BatchNo (None, 32, 32, 256)  1024        conv2d_19[0]
[0]
```

```
_____________________________________________________________________________
_____________________
conv2d_18 (Conv2D)              (None, 32, 32, 256)  590080      activation_1
2[0][0]

_____________________________________________________________________________
_____________________
add_5 (Add)                     (None, 32, 32, 256)  0           batch_normal
ization_18[0][0]
                                                                 conv2d_18[0]
[0]

_____________________________________________________________________________
_____________________
up_sampling2d_1 (UpSampling2D)  (None, 64, 64, 256)  0           add_5[0][0]

_____________________________________________________________________________
_____________________
concatenate_1 (Concatenate)     (None, 64, 64, 320)  0           up_sampling2
d_1[0][0]
                                                                 add_2[0][0]

_____________________________________________________________________________
_____________________
batch_normalization_19 (BatchNo (None, 64, 64, 320)  1280        concatenate_
1[0][0]

_____________________________________________________________________________
_____________________
activation_13 (Activation)      (None, 64, 64, 320)  0           batch_normal
ization_19[0][0]

_____________________________________________________________________________
_____________________
conv2d_20 (Conv2D)              (None, 64, 64, 128)  368768      activation_1
3[0][0]

_____________________________________________________________________________
_____________________
batch_normalization_20 (BatchNo (None, 64, 64, 128)  512         conv2d_20[0]
[0]

_____________________________________________________________________________
_____________________
conv2d_22 (Conv2D)              (None, 64, 64, 128)  41088       concatenate_
1[0][0]

_____________________________________________________________________________
_____________________
activation_14 (Activation)      (None, 64, 64, 128)  0           batch_normal
ization_20[0][0]

_____________________________________________________________________________
_____________________
batch_normalization_21 (BatchNo (None, 64, 64, 128)  512         conv2d_22[0]
[0]

_____________________________________________________________________________
_____________________
conv2d_21 (Conv2D)              (None, 64, 64, 128)  147584      activation_1
4[0][0]

_____________________________________________________________________________
_____________________
add_6 (Add)                     (None, 64, 64, 128)  0           batch_normal
ization_21[0][0]
                                                                 conv2d_21[0]
[0]

_____________________________________________________________________________
_____________________
up_sampling2d_2 (UpSampling2D)  (None, 128, 128, 128 0           add_6[0][0]

_____________________________________________________________________________
_____________________
concatenate_2 (Concatenate)     (None, 128, 128, 160 0           up_sampling2
d_2[0][0]
                                                                 add_1[0][0]
```

```
_____________________________________________________________________________
_____________________
batch_normalization_22 (BatchNo (None, 128, 128, 160 640         concatenate_
2[0][0]

_____________________________________________________________________________
_____________________
activation_15 (Activation)      (None, 128, 128, 160 0           batch_normal
ization_22[0][0]

_____________________________________________________________________________
_____________________
conv2d_23 (Conv2D)              (None, 128, 128, 64) 92224       activation_1
5[0][0]

_____________________________________________________________________________
_____________________
batch_normalization_23 (BatchNo (None, 128, 128, 64) 256         conv2d_23[0]
[0]

_____________________________________________________________________________
_____________________
conv2d_25 (Conv2D)              (None, 128, 128, 64) 10304       concatenate_
2[0][0]

_____________________________________________________________________________
_____________________
activation_16 (Activation)      (None, 128, 128, 64) 0           batch_normal
ization_23[0][0]

_____________________________________________________________________________
_____________________
batch_normalization_24 (BatchNo (None, 128, 128, 64) 256         conv2d_25[0]
[0]

_____________________________________________________________________________
_____________________
conv2d_24 (Conv2D)              (None, 128, 128, 64) 36928       activation_1
6[0][0]

_____________________________________________________________________________
_____________________
add_7 (Add)                     (None, 128, 128, 64) 0           batch_normal
ization_24[0][0]
                                                                 conv2d_24[0]
[0]

_____________________________________________________________________________
_____________________
up_sampling2d_3 (UpSampling2D)  (None, 256, 256, 64) 0           add_7[0][0]

_____________________________________________________________________________
_____________________
concatenate_3 (Concatenate)     (None, 256, 256, 80) 0           up_sampling2
d_3[0][0]
                                                                 add[0][0]

_____________________________________________________________________________
_____________________
batch_normalization_25 (BatchNo (None, 256, 256, 80) 320         concatenate_
3[0][0]

_____________________________________________________________________________
_____________________
activation_17 (Activation)      (None, 256, 256, 80) 0           batch_normal
ization_25[0][0]

_____________________________________________________________________________
_____________________
conv2d_26 (Conv2D)              (None, 256, 256, 32) 23072       activation_1
7[0][0]

_____________________________________________________________________________
_____________________
batch_normalization_26 (BatchNo (None, 256, 256, 32) 128         conv2d_26[0]
[0]

_____________________________________________________________________________
_____________________
```

```
conv2d_28 (Conv2D)              (None, 256, 256, 32) 2592        concatenate_
3[0][0]

_____________________________________________________________________________
_____________________
activation_18 (Activation)      (None, 256, 256, 32) 0           batch_normal
ization_26[0][0]

_____________________________________________________________________________
_____________________
batch_normalization_27 (BatchNo (None, 256, 256, 32) 128         conv2d_28[0]
[0]

_____________________________________________________________________________
_____________________
conv2d_27 (Conv2D)              (None, 256, 256, 32) 9248        activation_1
8[0][0]

_____________________________________________________________________________
_____________________
add_8 (Add)                     (None, 256, 256, 32) 0           batch_normal
ization_27[0][0]
                                                                 conv2d_27[0]
[0]

_____________________________________________________________________________
_____________________
conv2d_29 (Conv2D)              (None, 256, 256, 1)  33          add_8[0][0]
=============================================================================
=====================
Total params: 4,723,057
Trainable params: 4,715,761
Non-trainable params: 7,296
_____________________________________________________________________________
_____________________
```

In [115…

```
# from tensorflow.keras.utils import  plot_model/home/ubuntu/Desktop/NNDL Pr

# plot_model(
#     model,
#     to_file="model.png",
#     show_shapes=True,
#     show_layer_names=True,
#     rankdir="TB",
#     expand_nested=True,
#     dpi=100,
# )
```

Loss & Compile

In [34]:

```
smooth = 1.

def dice_coef(y_true, y_pred):
    y_true_f = K.flatten(y_true)
    y_pred_f = K.flatten(y_pred)
    intersection = tf.reduce_sum(y_true_f * y_pred_f)
    return (2. * intersection + smooth) / (tf.reduce_sum(y_true_f) + tf.reduc

def dice_coef_loss(y_true, y_pred):
    return 1.0 - dice_coef(y_true, y_pred)

def IOU(y_true, y_pred):

    y_true = K.flatten(y_true)
    y_pred = K.flatten(y_pred)

    thresh = 0.5

    y_true = K.cast(K.greater_equal(y_true, thresh), 'float32')
    y_pred = K.cast(K.greater_equal(y_pred, thresh), 'float32')

    union = K.sum(K.maximum(y_true, y_pred)) + K.epsilon()
    intersection = K.sum(K.minimum(y_true, y_pred)) + K.epsilon()

    iou = intersection/union

    return iou
```

In [35]:

```
def lr_schedule(epoch):

    lr =0.0035
    if epoch >150:
        lr *=2**-1
    elif epoch >80:
        lr *=2**(-1)
    elif epoch >50:
        lr *=2**(-1)
    elif epoch >30:
        lr *=2**(-1)

    print('Learning rate: ', lr)
    return lr
```

In [36]:

```
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.callbacks import LearningRateScheduler
from tensorflow.keras.optimizers import SGD
```

In [37]:

```
import time

start_time = time.time()

# Prepare callbacks for model saving and for learning rate adjustment.
lr_scheduler = LearningRateScheduler(lr_schedule)

lr_reducer = ReduceLROnPlateau(factor=np.sqrt(0.1),
                               cooldown=0,
                               patience=5,
                               min_lr=0.5e-6)

callbacks = [lr_reducer, lr_scheduler]
```

In [38]:

```
import tensorflow as tf
optimiser=tf.keras.optimizers.Adam(
    learning_rate=lr_schedule(0),
    beta_1=0.9,
    beta_2=0.999,
    epsilon=1e-07,
    amsgrad=True,
    name="Adam"
)
model.compile(optimizer =optimiser , loss = dice_coef_loss, metrics = ['accur
```

```
Learning rate:  0.0035
```

Training

In [39]:

```
train_steps = len(train_image_files)//batch_size
valid_steps = len(valid_image_files)//batch_size

history = model.fit(
    tg,
    steps_per_epoch=train_steps,
    initial_epoch = 0,
    epochs=50,
    validation_data = vg,
    validation_steps = valid_steps,callbacks=callbacks)
```

```
2022-03-31 21:32:50.958835: I tensorflow/compiler/mlir/mlir_graph_optimizatio
n_pass.cc:116] None of the MLIR optimization passes are enabled (registered
2)
2022-03-31 21:32:50.990826: I tensorflow/core/platform/profile_utils/cpu_util
s.cc:112] CPU Frequency: 3799900000 Hz
Epoch 1/50
Learning rate:  0.0035
379/379 [==============================] - 1619s 4s/step - loss: 0.4148 - acc
uracy: 0.9342 - IOU: 0.4316 - dice_coef: 0.5851 - val_loss: 0.4197 - val_accu
racy: 0.9203 - val_IOU: 0.4403 - val_dice_coef: 0.5803
Epoch 2/50
Learning rate:  0.0035
379/379 [==============================] - 1629s 4s/step - loss: 0.3397 - acc
uracy: 0.9550 - IOU: 0.5013 - dice_coef: 0.6591 - val_loss: 0.3759 - val_accu
racy: 0.9603 - val_IOU: 0.4790 - val_dice_coef: 0.6241
Epoch 3/50
Learning rate:  0.0035
379/379 [==============================] - 1704s 4s/step - loss: 0.3349 - acc
uracy: 0.9550 - IOU: 0.5073 - dice_coef: 0.6650 - val_loss: 0.3991 - val_accu
racy: 0.9587 - val_IOU: 0.4666 - val_dice_coef: 0.6009
Epoch 4/50
Learning rate:  0.0035
379/379 [==============================] - 1719s 5s/step - loss: 0.3018 - acc
uracy: 0.9630 - IOU: 0.5456 - dice_coef: 0.6978 - val_loss: 0.3079 - val_accu
racy: 0.9630 - val_IOU: 0.5476 - val_dice_coef: 0.6921
Epoch 5/50
Learning rate:  0.0035
379/379 [==============================] - 1717s 5s/step - loss: 0.3067 - acc
uracy: 0.9603 - IOU: 0.5387 - dice_coef: 0.6929 - val_loss: 0.3666 - val_accu
racy: 0.9238 - val_IOU: 0.5007 - val_dice_coef: 0.6334
Epoch 6/50
Learning rate:  0.0035
379/379 [==============================] - 1727s 5s/step - loss: 0.3009 - acc
uracy: 0.9628 - IOU: 0.5457 - dice_coef: 0.6990 - val_loss: 0.2805 - val_accu
racy: 0.9636 - val_IOU: 0.5827 - val_dice_coef: 0.7195
Epoch 7/50
Learning rate:  0.0035
379/379 [==============================] - 1601s 4s/step - loss: 0.2894 - acc
uracy: 0.9614 - IOU: 0.5597 - dice_coef: 0.7101 - val_loss: 0.2565 - val_accu
```

```
racy: 0.9703 - val_IOU: 0.6083 - val_dice_coef: 0.7435
Epoch 8/50
Learning rate:  0.0035
379/379 [==============================] - 1583s 4s/step - loss: 0.2802 - acc
uracy: 0.9635 - IOU: 0.5700 - dice_coef: 0.7197 - val_loss: 0.2745 - val_accu
racy: 0.9693 - val_IOU: 0.5862 - val_dice_coef: 0.7255
Epoch 9/50
Learning rate:  0.0035
379/379 [==============================] - 1597s 4s/step - loss: 0.2814 - acc
uracy: 0.9625 - IOU: 0.5688 - dice_coef: 0.7186 - val_loss: 0.8248 - val_accu
racy: 0.9368 - val_IOU: 0.1091 - val_dice_coef: 0.1752
Epoch 10/50
Learning rate:  0.0035
379/379 [==============================] - 1598s 4s/step - loss: 0.2882 - acc
uracy: 0.9634 - IOU: 0.5615 - dice_coef: 0.7115 - val_loss: 0.2881 - val_accu
racy: 0.9667 - val_IOU: 0.5774 - val_dice_coef: 0.7119
Epoch 11/50
Learning rate:  0.0035
379/379 [==============================] - 1595s 4s/step - loss: 0.2856 - acc
uracy: 0.9647 - IOU: 0.5643 - dice_coef: 0.7141 - val_loss: 0.2881 - val_accu
racy: 0.9672 - val_IOU: 0.5789 - val_dice_coef: 0.7119
Epoch 12/50
Learning rate:  0.0035
379/379 [==============================] - 1596s 4s/step - loss: 0.2685 - acc
uracy: 0.9658 - IOU: 0.5844 - dice_coef: 0.7315 - val_loss: 0.3054 - val_accu
racy: 0.9380 - val_IOU: 0.5665 - val_dice_coef: 0.6946
Epoch 13/50
Learning rate:  0.0035
379/379 [==============================] - 1603s 4s/step - loss: 0.2748 - acc
uracy: 0.9649 - IOU: 0.5771 - dice_coef: 0.7247 - val_loss: 0.2864 - val_accu
racy: 0.9511 - val_IOU: 0.5811 - val_dice_coef: 0.7136
Epoch 14/50
Learning rate:  0.0035
379/379 [==============================] - 1602s 4s/step - loss: 0.2607 - acc
uracy: 0.9665 - IOU: 0.5936 - dice_coef: 0.7392 - val_loss: 0.2744 - val_accu
racy: 0.9548 - val_IOU: 0.5971 - val_dice_coef: 0.7256
Epoch 15/50
Learning rate:  0.0035
379/379 [==============================] - 1599s 4s/step - loss: 0.2635 - acc
uracy: 0.9665 - IOU: 0.5899 - dice_coef: 0.7364 - val_loss: 0.2356 - val_accu
racy: 0.9743 - val_IOU: 0.6318 - val_dice_coef: 0.7644
Epoch 16/50
Learning rate:  0.0035
379/379 [==============================] - 1582s 4s/step - loss: 0.2665 - acc
uracy: 0.9660 - IOU: 0.5859 - dice_coef: 0.7335 - val_loss: 0.2471 - val_accu
racy: 0.9708 - val_IOU: 0.6253 - val_dice_coef: 0.7529
Epoch 17/50
Learning rate:  0.0035
379/379 [==============================] - 1585s 4s/step - loss: 0.2594 - acc
uracy: 0.9668 - IOU: 0.5946 - dice_coef: 0.7404 - val_loss: 0.2391 - val_accu
racy: 0.9752 - val_IOU: 0.6302 - val_dice_coef: 0.7609
Epoch 18/50
Learning rate:  0.0035
379/379 [==============================] - 1595s 4s/step - loss: 0.2687 - acc
uracy: 0.9668 - IOU: 0.5853 - dice_coef: 0.7313 - val_loss: 0.3185 - val_accu
racy: 0.9434 - val_IOU: 0.5498 - val_dice_coef: 0.6815
Epoch 19/50
Learning rate:  0.0035
379/379 [==============================] - 1597s 4s/step - loss: 0.2579 - acc
uracy: 0.9680 - IOU: 0.5984 - dice_coef: 0.7419 - val_loss: 0.2303 - val_accu
racy: 0.9765 - val_IOU: 0.6415 - val_dice_coef: 0.7697
Epoch 20/50
Learning rate:  0.0035
379/379 [==============================] - 1608s 4s/step - loss: 0.2450 - acc
```

```
uracy: 0.9685 - IOU: 0.6127 - dice_coef: 0.7550 - val_loss: 0.2710 - val_accu
racy: 0.9629 - val_IOU: 0.5938 - val_dice_coef: 0.7290
Epoch 21/50
Learning rate:  0.0035
379/379 [==============================] - 1603s 4s/step - loss: 0.2627 - acc
uracy: 0.9670 - IOU: 0.5911 - dice_coef: 0.7373 - val_loss: 0.2464 - val_accu
racy: 0.9727 - val_IOU: 0.6246 - val_dice_coef: 0.7536
Epoch 22/50
Learning rate:  0.0035
379/379 [==============================] - 1602s 4s/step - loss: 0.2581 - acc
uracy: 0.9672 - IOU: 0.5963 - dice_coef: 0.7419 - val_loss: 0.2317 - val_accu
racy: 0.9754 - val_IOU: 0.6400 - val_dice_coef: 0.7683
Epoch 23/50
Learning rate:  0.0035
379/379 [==============================] - 1591s 4s/step - loss: 0.2516 - acc
uracy: 0.9675 - IOU: 0.6045 - dice_coef: 0.7482 - val_loss: 0.2448 - val_accu
racy: 0.9690 - val_IOU: 0.6252 - val_dice_coef: 0.7552
Epoch 24/50
Learning rate:  0.0035
379/379 [==============================] - 1599s 4s/step - loss: 0.2592 - acc
uracy: 0.9681 - IOU: 0.5956 - dice_coef: 0.7408 - val_loss: 0.2503 - val_accu
racy: 0.9733 - val_IOU: 0.6163 - val_dice_coef: 0.7497
Epoch 25/50
Learning rate:  0.0035
379/379 [==============================] - 1658s 4s/step - loss: 0.2543 - acc
uracy: 0.9673 - IOU: 0.6012 - dice_coef: 0.7456 - val_loss: 0.2790 - val_accu
racy: 0.9668 - val_IOU: 0.5902 - val_dice_coef: 0.7210
Epoch 26/50
Learning rate:  0.0035
379/379 [==============================] - 1642s 4s/step - loss: 0.2446 - acc
uracy: 0.9678 - IOU: 0.6126 - dice_coef: 0.7553 - val_loss: 0.2342 - val_accu
racy: 0.9760 - val_IOU: 0.6376 - val_dice_coef: 0.7658
Epoch 27/50
Learning rate:  0.0035
379/379 [==============================] - 1650s 4s/step - loss: 0.2514 - acc
uracy: 0.9693 - IOU: 0.6044 - dice_coef: 0.7483 - val_loss: 0.2358 - val_accu
racy: 0.9744 - val_IOU: 0.6350 - val_dice_coef: 0.7642
Epoch 28/50
Learning rate:  0.0035
379/379 [==============================] - 1651s 4s/step - loss: 0.2476 - acc
uracy: 0.9689 - IOU: 0.6091 - dice_coef: 0.7521 - val_loss: 0.2303 - val_accu
racy: 0.9759 - val_IOU: 0.6424 - val_dice_coef: 0.7697
Epoch 29/50
Learning rate:  0.0035
379/379 [==============================] - 1648s 4s/step - loss: 0.2423 - acc
uracy: 0.9690 - IOU: 0.6157 - dice_coef: 0.7576 - val_loss: 0.2432 - val_accu
racy: 0.9715 - val_IOU: 0.6261 - val_dice_coef: 0.7568
Epoch 30/50
Learning rate:  0.0035
379/379 [==============================] - 1650s 4s/step - loss: 0.2457 - acc
uracy: 0.9682 - IOU: 0.6110 - dice_coef: 0.7539 - val_loss: 0.2279 - val_accu
racy: 0.9749 - val_IOU: 0.6412 - val_dice_coef: 0.7721
Epoch 31/50
Learning rate:  0.0035
379/379 [==============================] - 1656s 4s/step - loss: 0.2481 - acc
uracy: 0.9684 - IOU: 0.6091 - dice_coef: 0.7519 - val_loss: 0.2358 - val_accu
racy: 0.9746 - val_IOU: 0.6362 - val_dice_coef: 0.7642
Epoch 32/50
Learning rate:  0.00175
379/379 [==============================] - 1648s 4s/step - loss: 0.2414 - acc
uracy: 0.9691 - IOU: 0.6173 - dice_coef: 0.7582 - val_loss: 0.2311 - val_accu
racy: 0.9755 - val_IOU: 0.6433 - val_dice_coef: 0.7689
Epoch 33/50
Learning rate:  0.00175
```

```
379/379 [==============================] - 1658s 4s/step - loss: 0.2345 - acc
uracy: 0.9708 - IOU: 0.6255 - dice_coef: 0.7654 - val_loss: 0.2308 - val_accu
racy: 0.9757 - val_IOU: 0.6424 - val_dice_coef: 0.7692
Epoch 34/50
Learning rate:  0.00175
379/379 [==============================] - 1646s 4s/step - loss: 0.2364 - acc
uracy: 0.9691 - IOU: 0.6226 - dice_coef: 0.7635 - val_loss: 0.2271 - val_accu
racy: 0.9764 - val_IOU: 0.6475 - val_dice_coef: 0.7729
Epoch 35/50
Learning rate:  0.00175
379/379 [==============================] - 1649s 4s/step - loss: 0.2351 - acc
uracy: 0.9702 - IOU: 0.6249 - dice_coef: 0.7649 - val_loss: 0.2140 - val_accu
racy: 0.9775 - val_IOU: 0.6614 - val_dice_coef: 0.7860
Epoch 36/50
Learning rate:  0.00175
379/379 [==============================] - 1656s 4s/step - loss: 0.2370 - acc
uracy: 0.9702 - IOU: 0.6217 - dice_coef: 0.7624 - val_loss: 0.2187 - val_accu
racy: 0.9768 - val_IOU: 0.6558 - val_dice_coef: 0.7813
Epoch 37/50
Learning rate:  0.00175
379/379 [==============================] - 1653s 4s/step - loss: 0.2480 - acc
uracy: 0.9699 - IOU: 0.6103 - dice_coef: 0.7519 - val_loss: 0.2205 - val_accu
racy: 0.9774 - val_IOU: 0.6541 - val_dice_coef: 0.7795
Epoch 38/50
Learning rate:  0.00175
379/379 [==============================] - 1650s 4s/step - loss: 0.2334 - acc
uracy: 0.9696 - IOU: 0.6273 - dice_coef: 0.7664 - val_loss: 0.2348 - val_accu
racy: 0.9739 - val_IOU: 0.6408 - val_dice_coef: 0.7652
Epoch 39/50
Learning rate:  0.00175
379/379 [==============================] - 1662s 4s/step - loss: 0.2434 - acc
uracy: 0.9698 - IOU: 0.6153 - dice_coef: 0.7565 - val_loss: 0.2181 - val_accu
racy: 0.9768 - val_IOU: 0.6572 - val_dice_coef: 0.7819
Epoch 40/50
Learning rate:  0.00175
379/379 [==============================] - 1627s 4s/step - loss: 0.2411 - acc
uracy: 0.9696 - IOU: 0.6182 - dice_coef: 0.7589 - val_loss: 0.2208 - val_accu
racy: 0.9777 - val_IOU: 0.6558 - val_dice_coef: 0.7792
Epoch 41/50
Learning rate:  0.00175
379/379 [==============================] - 1617s 4s/step - loss: 0.2361 - acc
uracy: 0.9702 - IOU: 0.6235 - dice_coef: 0.7637 - val_loss: 0.2166 - val_accu
racy: 0.9770 - val_IOU: 0.6576 - val_dice_coef: 0.7834
Epoch 42/50
Learning rate:  0.00175
379/379 [==============================] - 1622s 4s/step - loss: 0.2301 - acc
uracy: 0.9706 - IOU: 0.6308 - dice_coef: 0.7698 - val_loss: 0.2263 - val_accu
racy: 0.9747 - val_IOU: 0.6437 - val_dice_coef: 0.7737
Epoch 43/50
Learning rate:  0.00175
379/379 [==============================] - 1624s 4s/step - loss: 0.2398 - acc
uracy: 0.9708 - IOU: 0.6194 - dice_coef: 0.7602 - val_loss: 0.2350 - val_accu
racy: 0.9708 - val_IOU: 0.6381 - val_dice_coef: 0.7650
Epoch 44/50
Learning rate:  0.00175
379/379 [==============================] - 1730s 5s/step - loss: 0.2311 - acc
uracy: 0.9712 - IOU: 0.6296 - dice_coef: 0.7687 - val_loss: 0.2427 - val_accu
racy: 0.9724 - val_IOU: 0.6298 - val_dice_coef: 0.7573
Epoch 45/50
Learning rate:  0.00175
379/379 [==============================] - 1669s 4s/step - loss: 0.2346 - acc
uracy: 0.9698 - IOU: 0.6261 - dice_coef: 0.7653 - val_loss: 0.2185 - val_accu
racy: 0.9765 - val_IOU: 0.6549 - val_dice_coef: 0.7815
Epoch 46/50
```

```
Learning rate:  0.00175
379/379 [==============================] - 1599s 4s/step - loss: 0.2364 - acc
uracy: 0.9695 - IOU: 0.6234 - dice_coef: 0.7631 - val_loss: 0.2201 - val_accu
racy: 0.9772 - val_IOU: 0.6563 - val_dice_coef: 0.7799
Epoch 47/50
Learning rate:  0.00175
379/379 [==============================] - 1604s 4s/step - loss: 0.2362 - acc
uracy: 0.9704 - IOU: 0.6231 - dice_coef: 0.7634 - val_loss: 0.2302 - val_accu
racy: 0.9744 - val_IOU: 0.6443 - val_dice_coef: 0.7698
Epoch 48/50
Learning rate:  0.00175
379/379 [==============================] - 1607s 4s/step - loss: 0.2346 - acc
uracy: 0.9704 - IOU: 0.6250 - dice_coef: 0.7652 - val_loss: 0.2233 - val_accu
racy: 0.9770 - val_IOU: 0.6534 - val_dice_coef: 0.7767
Epoch 49/50
Learning rate:  0.00175
379/379 [==============================] - 1615s 4s/step - loss: 0.2363 - acc
uracy: 0.9701 - IOU: 0.6232 - dice_coef: 0.7634 - val_loss: 0.2289 - val_accu
racy: 0.9762 - val_IOU: 0.6445 - val_dice_coef: 0.7711
Epoch 50/50
Learning rate:  0.00175
379/379 [==============================] - 1597s 4s/step - loss: 0.2377 - acc
uracy: 0.9698 - IOU: 0.6220 - dice_coef: 0.7622 - val_loss: 0.2203 - val_accu
racy: 0.9764 - val_IOU: 0.6549 - val_dice_coef: 0.7797
```

In [40]:

```
# save model
model.save('crack500.h5')
print('Model Saved!')
```

```
Model Saved!
```

In [41]:

```
# saving and loading the model weights

# save model
model.save_weights('crack500_weights')
print('Model Saved!')

# load model
# savedModel = model.load_weights('gfgModelWeights')
# print('Model Loaded!')
```

```
Model Saved!
```

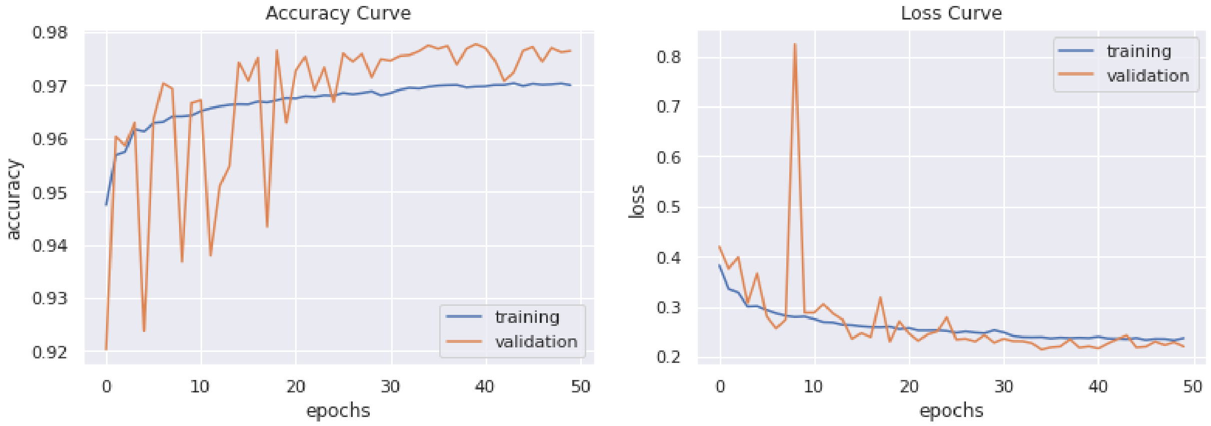
In [42]:

```
train_loss = history.history['loss']
valid_loss = history.history['val_loss']

train_acc = history.history['accuracy']
valid_acc = history.history['val_accuracy']
```

In [43]:

```
fig, axes = plt.subplots(1, 2, figsize=(13,4))
axes = axes.flatten()

axes[0].plot(train_acc, label='training')
axes[0].plot(valid_acc, label='validation')
axes[0].set_title('Accuracy Curve')
axes[0].set_xlabel('epochs')
axes[0].set_ylabel('accuracy')
axes[0].legend()


axes[1].plot(train_loss, label='training')
```

```
axes[1].plot(valid_loss, label='validation')
axes[1].set_title('Loss Curve')
axes[1].set_xlabel('epochs')
axes[1].set_ylabel('loss')
axes[1].legend()

plt.show()
```

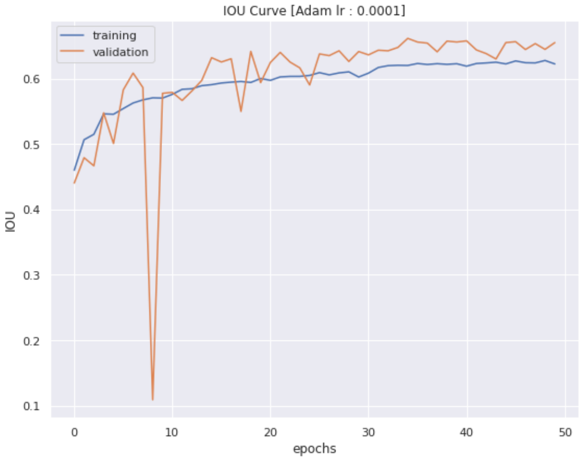


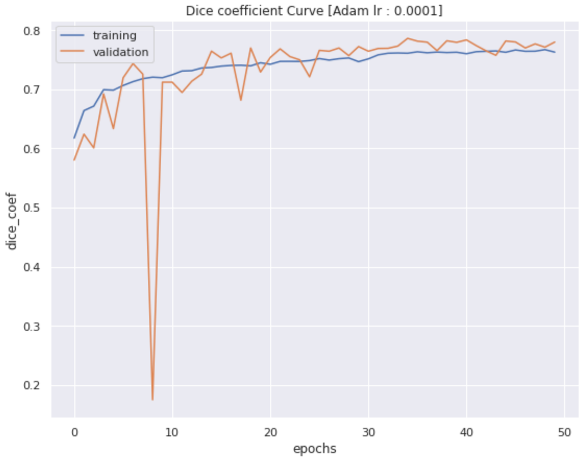
In [44]:

```
train_dice = history.history['dice_coef']
valid_dice = history.history['val_dice_coef']


train_IOU = history.history['IOU']
valid_IOU = history.history['val_IOU']
```

In [45]:

```
fig, axes = plt.subplots(1, 2, figsize=(20,7))
axes = axes.flatten()


axes[0].plot(train_IOU, label='training')
axes[0].plot(valid_IOU, label='validation')
axes[0].set_title('IOU Curve [Adam lr : 0.0001]')
axes[0].set_xlabel('epochs')
axes[0].set_ylabel('IOU')
axes[0].legend()


axes[1].plot(train_dice, label='training')
axes[1].plot(valid_dice, label='validation')
axes[1].set_title('Dice coefficient Curve [Adam lr : 0.0001]')
axes[1].set_xlabel('epochs')
axes[1].set_ylabel('dice_coef')
axes[1].legend()

plt.show()
```





Testing

In [46]:

```
for i, j in test1_generator:
  break

print(i.shape)
print(j.shape)
```

```
(5, 128, 128, 3)
(5, 128, 128, 1)
```

In [47]:

```
ttg = Generator(test_image_paths,test_mask_paths, batch_size, img_dim, augmer
```

In [48]:

```
for i, j in ttg:
  break

print(i.shape)
print(j.shape)
```

```
(5, 256, 256, 3)
(5, 256, 256, 1)
```

In [66]:

```
test_generator = Generator(test_image_paths,test_mask_paths,1124, img_dim)

for x_test, y_test in test_generator:
  break

print(x_test.shape)
print(y_test.shape)

y_pred = model.predict(x_test)

yy_true = (y_test>0.5).flatten()
yy_pred = (y_pred>0.5).flatten()

print(yy_true.shape)
print(yy_pred.shape)
```

```
(1124, 256, 256, 3)
(1124, 256, 256, 1)
(73662464,)
(73662464,)
```

In [60]:

```
report = classification_report(yy_true, yy_pred, output_dict=True)
```

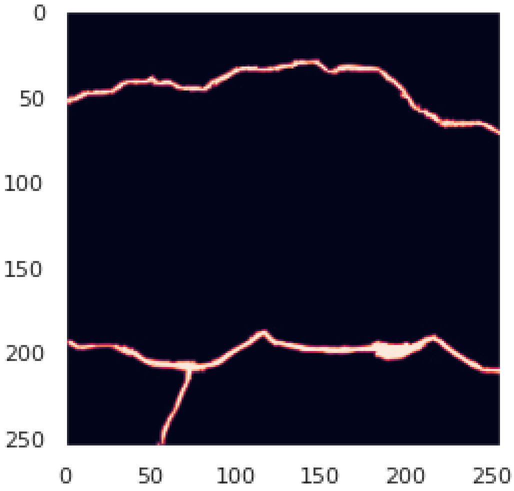
```
Accuracy = accuracy_score(yy_true, yy_pred)

Precision = report['True']['precision']
Recall = report['True']['recall']
F1_score = report['True']['f1-score']

Sensitivity = Recall
Specificity = report['False']['recall']

AUC = roc_auc_score(y_test.flatten(), y_pred.flatten())

IOU = (Precision*Recall)/(Precision+Recall-Precision*Recall)

print("Accuracy: {0:.4f}\n".format(Accuracy))
print("Precision: {0:.4f}\n".format(Precision))
print("Recall: {0:.4f}\n".format(Recall))
print("F1-Score: {0:.4f}\n".format(F1_score))
print("Sensitivity: {0:.4f}\n".format(Sensitivity))
print("Specificity: {0:.4f}\n".format(Specificity))
print("AUC: {0:.4f}\n".format(AUC))
print("IOU: {0:.4f}\n".format(IOU))
print('-'*50,'\n')
print(classification_report(yy_true, yy_pred))
```

```
Accuracy: 0.9662

Precision: 0.6917

Recall: 0.7746

F1-Score: 0.7308

Sensitivity: 0.7746

Specificity: 0.9783

AUC: 0.9145

IOU: 0.5758

--------------------------------------------------

              precision    recall  f1-score   support

       False       0.99      0.98      0.98  69301712
        True       0.69      0.77      0.73   4360752

    accuracy                           0.97  73662464
   macro avg       0.84      0.88      0.86  73662464
weighted avg       0.97      0.97      0.97  73662464
```

In [77]:

```
print('train image and label size')
img_t1 = cv2.imread('/home/ubuntu/Desktop/NNDL Project/CRACK500/traincrop/201
img_l1 = cv2.imread('/home/ubuntu/Desktop/NNDL Project/CRACK500/traincrop/201
print(img_t1.shape)
print(img_l1.shape)
print('validation image and label size')
img_t2 = cv2.imread('/home/ubuntu/Desktop/NNDL Project/CRACK500/valcrop/2016
img_l2 = cv2.imread('/home/ubuntu/Desktop/NNDL Project/CRACK500/valcrop/2016
print(img_t2.shape)
print(img_l2.shape)
print('test image and label size')
img_t3 = cv2.imread('/home/ubuntu/Desktop/NNDL Project/CRACK500/testcrop/2016
img_l3 = cv2.imread('/home/ubuntu/Desktop/NNDL Project/CRACK500/testcrop/2016
print(img_t3.shape)
print(img_l3.shape)
print('our test image and label size')
img_t4 = cv2.imread('/home/ubuntu/Desktop/NNDL Project/PIPE NEW 256/Pipe (1).
img_l4 = cv2.imread('/home/ubuntu/Desktop/Pipe (1) GT.png', -1)
print(img_t4.shape)
print(img_l4[:,:,0].shape)

plt.grid(False)
plt.imshow(img_l4[:,:,0])
plt.show()
```

```
train image and label size
(360, 640, 3)
(360, 640)
validation image and label size
(360, 640, 3)
(360, 640)
test image and label size
(640, 360, 3)
(640, 360)
our test image and label size
(256, 256, 3)
(256, 256)
```



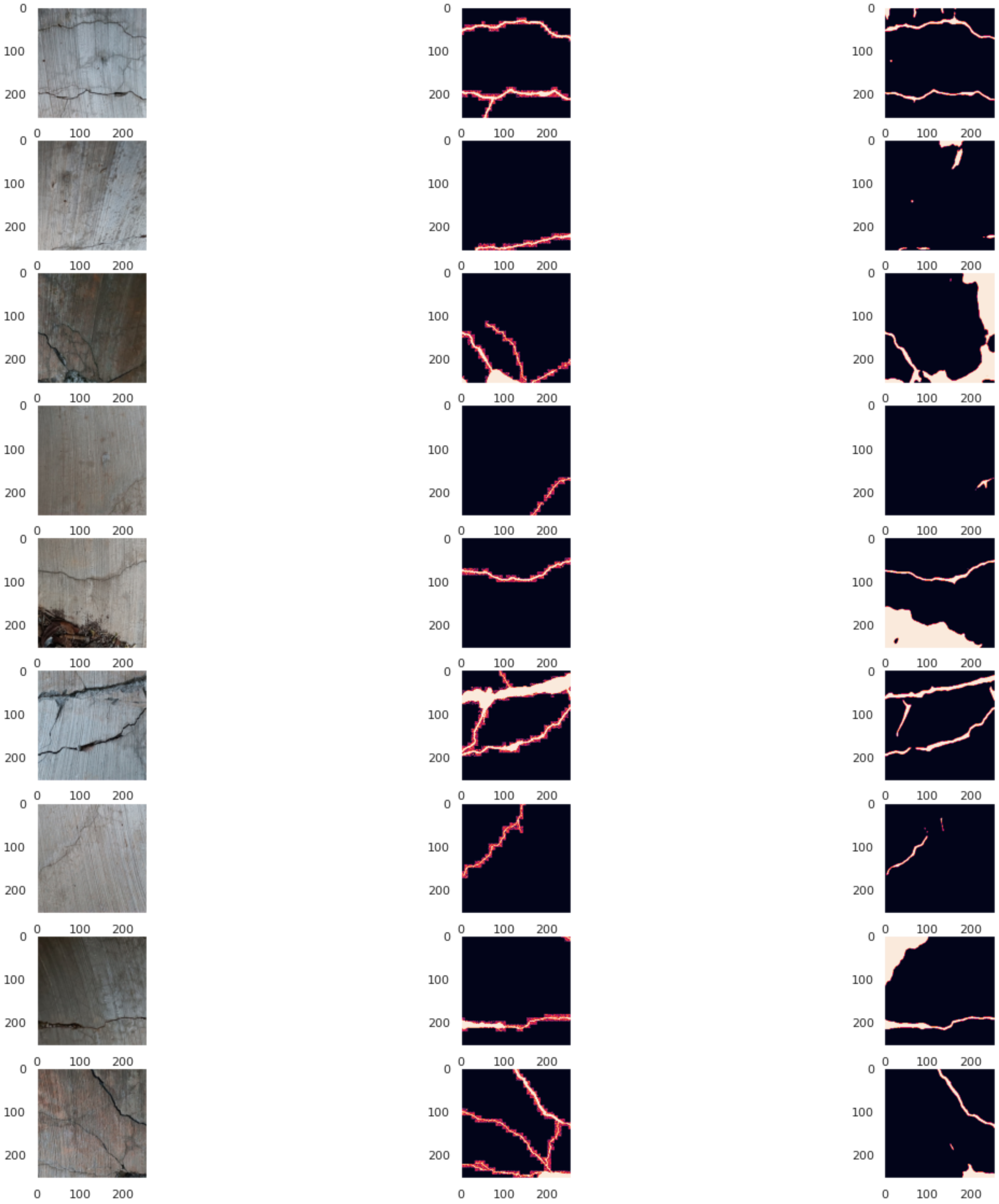
In [80]:

```
path = r'/home/ubuntu/Desktop/NNDL Project/PIPE NEW 256'
pipe_image_paths = sorted([os.path.join(path, fname) for fname in os.listdir(

test_lab_path = r'/home/ubuntu/Desktop/NNDL Project/Pipe GT'
pipe_lab_paths = sorted([os.path.join(test_lab_path, fname) for fname in os.l


pipe = Generator(pipe_image_paths, pipe_lab_paths, 9, img_dim, augment = Fals
for p_test, y_test in pipe:
    break
print(p_test.shape)
print(y_test.shape)
y_test = y_test.reshape(y_test.shape[0],y_test.shape[1],y_test.shape[2],y_tes
print(y_test.shape)
y_test = np.expand_dims(y_test[:,:,:,0], axis = -1)
print(y_test.shape)

y_pred = model.predict(p_test)
```

```python
yy_true = (y_test>0.5).flatten()
yy_pred = (y_pred>0.5).flatten()

print(yy_true.shape)
print(yy_pred.shape)

print((y_test.flatten().shape))
print((y_pred.flatten().shape))

report = classification_report(yy_true, yy_pred, output_dict=True)

Accuracy = accuracy_score(yy_true, yy_pred)

Precision = report['True']['precision']
Recall = report['True']['recall']
F1_score = report['True']['f1-score']

Sensitivity = Recall
Specificity = report['False']['recall']

AUC = roc_auc_score(y_test.flatten(), y_pred.flatten())

IOU = (Precision*Recall)/(Precision+Recall-Precision*Recall)

print("Accuracy: {0:.4f}\n".format(Accuracy))
print("Precision: {0:.4f}\n".format(Precision))
print("Recall: {0:.4f}\n".format(Recall))
print("F1-Score: {0:.4f}\n".format(F1_score))
print("Sensitivity: {0:.4f}\n".format(Sensitivity))
print("Specificity: {0:.4f}\n".format(Specificity))
print("AUC: {0:.4f}\n".format(AUC))
print("IOU: {0:.4f}\n".format(IOU))
print('-'*50,'\n')
print(classification_report(yy_true, yy_pred))

n=len(p_test)

plt.figure(figsize=(20, 20))

for i in range(n):
    # display image
    plt.grid(False)
    plt.subplot(n,3,3*i+1)
    plt.imshow(p_test[i])

    plt.grid(False)
    plt.subplot(n,3,3*i+2)
    plt.imshow(y_test[i])

    # display reconstructed (after noise removed) image
    plt.grid(False)
    plt.subplot(n, 3,  3*(i+1))
    plt.imshow(y_pred[i])

plt.grid(False)
plt.show()
```

```
(9, 256, 256, 3)
(9, 256, 256, 3, 1)
(9, 256, 256, 3)
(9, 256, 256, 1)
(589824,)
(589824,)
(589824,)
```

```
(589824,)
Accuracy: 0.9067

Precision: 0.3449

Recall: 0.4165

F1-Score: 0.3773

Sensitivity: 0.4165

Specificity: 0.9424

AUC: 0.7195

IOU: 0.2325

--------------------------------------------------

              precision    recall  f1-score   support

       False       0.96      0.94      0.95    549788
        True       0.34      0.42      0.38     40036

    accuracy                           0.91    589824
   macro avg       0.65      0.68      0.66    589824
weighted avg       0.92      0.91      0.91    589824
```



```
In [ ]:
```