Self Case Study-2_Severstal: Steel Defect Detection - Can you detect and classify defects in steel?

Section-II

4. Preprocessing, Feature Engineering and applying various models

4.1 Preprocessing and Feature Engineering

In [1]:
```
#importing usefull library
import pandas as pd
import numpy as np
import os
import cv2
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
import pandas_profiling as pp
from tqdm import tqdm
import sys
from PIL import Image, ImageDraw
from PIL import ImagePath
import urllib
import tensorflow as tf
# import matplotlib.pyplot as plt
from sklearn import preprocessing
from numpy import save ,load
from keras.layers.pooling import MaxPooling2D
from tensorflow.keras.layers import concatenate
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Conv2D, BatchNormalization, Activation, MaxPool2D, Conv2DTranspose, Concatenate, Input
from tensorflow.keras.models import Model
from tensorflow.keras.layers import GlobalAveragePooling2D, Dense, Conv2D, BatchNormalization, Dropout, Input, MaxPool2D , Flatten
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Input In [1], in <cell line: 5>()
      3 import numpy as np
      4 import os
----> 5 import cv2
      6 get_ipython().run_line_magic('matplotlib', 'inline')
      7 import matplotlib.pyplot as plt

ModuleNotFoundError: No module named 'cv2'
```

In [ ]:
```
# os.chdir('C:/Users/Vikrant Mohite/Desktop/Applied AI/Case Study-2')
# os.getcwd()
```

In [2]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [3]:
```
path1 = "/content/drive/MyDrive/Case Study-2/train_images.zip"
```

In [4]:
```
!pip install pyunpack
!pip install patool
```

```
Looking in indexes: https://pypi.org/simple, (https://pypi.org/simple,) https://us-python.pkg.dev/colab-wheels/public/simple/ (https://us-python.pkg.dev/colab-wheels/public/simple/)
Collecting pyunpack
  Downloading pyunpack-0.3-py2.py3-none-any.whl (4.1 kB)
Collecting entrypoint2
  Downloading entrypoint2-1.1-py2.py3-none-any.whl (9.9 kB)
Collecting easyprocess
  Downloading EasyProcess-1.1-py3-none-any.whl (8.7 kB)
Installing collected packages: entrypoint2, easyprocess, pyunpack
Successfully installed easyprocess-1.1 entrypoint2-1.1 pyunpack-0.3
Looking in indexes: https://pypi.org/simple, (https://pypi.org/simple,) https://us-python.pkg.dev/colab-wheels/public/simple/ (https://us-python.pkg.dev/colab-wheels/public/simple/)
Collecting patool
  Downloading patool-1.12-py2.py3-none-any.whl (77 kB)
     |████████████████████████████████| 77 kB 2.2 MB/s
Installing collected packages: patool
Successfully installed patool-1.12
```

In [5]:
```
mkdir train_images
```

In [6]:
```
#extracting images
from pyunpack import Archive
Archive(path1).extractall('/content/train_images')
```

In [7]:
```
pd.options.display.max_columns=50
train_df = pd.read_csv("/content/drive/MyDrive/Case Study-2/train.csv")
# train_df = pd.read_csv("train.csv")
train_df.shape
```

Out[7]:
```
(7095, 3)
```

In [ ]:
```
train_df.head(5)
```

Out[8]:

	ImageId	ClassId	EncodedPixels
0	0002cc93b.jpg	1	29102 12 29346 24 29602 24 29858 24 30114 24 3...
1	0007a71bf.jpg	3	18661 28 18863 82 19091 110 19347 110 19603 11...
2	000a4bcdd.jpg	1	37607 3 37858 8 38108 14 38359 20 38610 25 388...
3	000f6bf48.jpg	4	131973 1 132228 4 132483 6 132738 8 132993 11 ...
4	0014fce06.jpg	3	229501 11 229741 33 229981 55 230221 77 230468...

In [8]:
```
# train_images = os.listdir("C:/Users/Vikrant Mohite/Desktop/Applied AI/Case Study-2/train_images")
train_images = os.listdir("/content/train_images")
len(train_images)
```

Out[8]:
```
12568
```

In [9]:
```
train_images_df = pd.DataFrame(train_images,columns =['ImageId'])
train_images_df.head()
```

Out[9]:

	ImageId
0	bac68f2c3.jpg
1	c2234a956.jpg
2	d7aa8391c.jpg
3	fe56055d0.jpg
4	49d28224c.jpg

In [10]:
```
#mearging train_df & train_images_df
train_df_mearged = pd.merge(train_df,train_images_df,how = 'outer',on = ['ImageId','ImageId'])
train_df_mearged.head(5)
```

Out[10]:

	ImageId	ClassId	EncodedPixels
0	0002cc93b.jpg	1.0	29102 12 29346 24 29602 24 29858 24 30114 24 3...
1	0007a71bf.jpg	3.0	18661 28 18863 82 19091 110 19347 110 19603 11...
2	000a4bcdd.jpg	1.0	37607 3 37858 8 38108 14 38359 20 38610 25 388...
3	000f6bf48.jpg	4.0	131973 1 132228 4 132483 6 132738 8 132993 11 ...
4	0014fce06.jpg	3.0	229501 11 229741 33 229981 55 230221 77 230468...

In [11]:
```
train_df_mearged.shape
```

Out[11]:
```
(12997, 3)
```

In [12]:
```
train_df_mearged.isnull().sum()
```

Out[12]:
```
ImageId            0
ClassId         5902
EncodedPixels   5902
dtype: int64
```

In [ ]:
```
7095+5902
```

Out[13]:
```
12997
```

In [13]:
```
# replacing 'NaN' values of class ID of non-defective images with '0'
train_df_mearged['ClassId'].fillna(value=0,inplace=True)
train_df_mearged.head()
```

Out[13]:

	ImageId	ClassId	EncodedPixels
0	0002cc93b.jpg	1.0	29102 12 29346 24 29602 24 29858 24 30114 24 3...
1	0007a71bf.jpg	3.0	18661 28 18863 82 19091 110 19347 110 19603 11...
2	000a4bcdd.jpg	1.0	37607 3 37858 8 38108 14 38359 20 38610 25 388...
3	000f6bf48.jpg	4.0	131973 1 132228 4 132483 6 132738 8 132993 11 ...
4	0014fce06.jpg	3.0	229501 11 229741 33 229981 55 230221 77 230468...

In [14]:
```
train_df_mearged.tail()
```

Out[14]:

	ImageId	ClassId	EncodedPixels
12992	ceae95bea.jpg	0.0	NaN
12993	288314368.jpg	0.0	NaN
12994	ae765c316.jpg	0.0	NaN
12995	034b387c3.jpg	0.0	NaN
12996	57461a858.jpg	0.0	NaN

The below function creates the dataframe of paths of all images of given dataset

In [15]:
```
# The below function creates the dataframe of path of the images of given dataset
def return_file_names_df(root_dir):
    lst1 = []
    for path, dirc, files in os.walk(root_dir):
        for name in files:
            if name.endswith('jpg'):
                lst1.append(path + '/' + name)
    data_df = pd.DataFrame((lst1), columns = ["image_path"])
    pd.set_option('display.max_colwidth', None)
#     pd.set_option("display.max_rows", None)
    return data_df
```

In [16]:
```
# train_images_path = "C:/Users/Vikrant Mohite/Desktop/Applied AI/Case Study-2/train_images"
train_images_path = "/content/train_images"
rfn_df = return_file_names_df(train_images_path)
```

In [17]:
```
rfn_df.head()
```

Out[17]:

	image_path
0	/content/train_images/bac68f2c3.jpg
1	/content/train_images/c2234a956.jpg
2	/content/train_images/d7aa8391c.jpg
3	/content/train_images/fe56055d0.jpg
4	/content/train_images/49d28224c.jpg

In [18]:
```
rfn_df.shape
```

Out[18]:
```
(12568, 1)
```

The below function creates dataframe of paths of the defective images from the dataset

In [19]:
```
def return_file_names_df1(root_dir):
    list1 = []
    list2 = []
    for path, dirc, files in os.walk(root_dir):
        for name in files:
            list1.append(name)
    for index, row in train_df.iterrows():
        if row.ImageId in list1 and row.ClassId !=0:
            list2.append(path + '/' + row.ImageId)
    data_df = pd.DataFrame((list2), columns = ["image_path"])
    pd.set_option('display.max_colwidth', None)
    return data_df
```

In [20]:
```
df1 = return_file_names_df1(train_images_path)
df1.head()
```

Out[20]:

	image_path
0	/content/train_images/0002cc93b.jpg
1	/content/train_images/0007a71bf.jpg
2	/content/train_images/000a4bcdd.jpg
3	/content/train_images/000f6bf48.jpg
4	/content/train_images/0014fce06.jpg

In [21]:
```
df1.shape
```

Out[21]: (7095, 1)

Defining colour coding for given classes.

In [ ]:
```
colourmap = [[0, 0, 0], [255, 105, 180], [ 180,255,105],[ 105, 180,255], [255, 255,105]]
classes = [0,1,2,3,4]
classes_tocolour_ =   dict({0: [0, 0, 0], 1: [255, 105, 180], 2:  [180,255,105], 3:[105, 180,255], 4: [ 255, 255,105]})
```

In [ ]:
```

```

Below code generates numpy array of shape of the given images from the EncodedPixels given in 'train' csv file

In [ ]:
```
train_numpy_masks = []
for index, row in tqdm(train_df_mearged.iterrows()):
    img = np.zeros(1600*256, dtype=np.uint8)
    if row.ClassId !=0:
        p = row.EncodedPixels.split()
        starts, lengths = [np.asarray(x, dtype=int) for x in (p[0::2], p[1::2])]
        starts -= 1
        ends = starts + lengths
        for lo, hi in zip(starts, ends):
            img[lo:hi] = row.ClassId
        img = img.reshape(1600,256).T
        train_numpy_masks.append(img)
```

```
12997it [00:16, 775.07it/s]
```

In [ ]:
```
del train_numpy_masks
```

In [ ]:
```
# len(train_numpy_masks)
```

In [ ]:
```
train_numpy_masks[0].shape
```

The below code adds the color map into numpy array generated above

In [ ]:
```
RGB_image_list = []
for img in tqdm(train_numpy_masks):
    RGB_image = []
    for i in img :
        lst1 = []
        for j in i :
            lst1.append(classes_tocolour.get(j))
        RGB_image.append(lst1)
    c = np.array(RGB_image)
    RGB_image_list.append(c)
```

```
100%|██████████████████████████████████████████████████████████████████████████████| 7095/7095 [1:10:00<00:00,  1.69it/s]
```

In [ ]:
```
# RGB_image_df = pd.DataFrame(RGB_image_list, dtype=np.int8)
```

In [ ]:
```
RGB_image_arr = np.array(RGB_image_list, dtype=np.int8)
```

In [ ]:
```
sys.getsizeof(RGB_image_arr)
```

Out[30]:
```
160
```

saving 'RGB_image_arr' into pickel format

In [ ]:
```
# # saving 'RGB_image_arr' into pickel file
# import pickle
# with open('RGB_image_arr.pkl', 'wb') as f:
#     pickle.dump(RGB_image_arr, f)
```

loading previously saved pickle file

In [ ]:
```
import pickle
file = open("RGB_image_arr.pkl",'rb')
# file = open("/content/drive/MyDrive/Case_Study-2/RGB_image_arr.pkl",'rb')
RGB_image_arr = pickle.load(file)
file.close()
```

In [ ]:
```
RGB_image_arr.shape
```

Out[28]:
```
(7095, 256, 1600, 3)
```

In [ ]:
```
print(RGB_image_arr.__sizeof__())
```

```
160
```

Generating ans saving mask images for given images of dataset

In [ ]:
```
mkdir output
```

In [ ]:
```
# p = "C:/Users/Vikrant Mohite/Desktop/Applied AI/Case Study-2/output/"
p = "/content/output/"
for k, (i, j) in enumerate(zip(RGB_image_arr, train_df['ImageId'].values)):
    mask_img = Image.fromarray(i.astype(np.uint8))
    mask_name = j.split('.')[0] + '_' + str(k) + '_mask.jpg'
    mask_path = os.path.join(p, mask_name)
    mask_img.save(mask_path)
```

In [ ]:
```
mkdir output_5d
```

In [ ]:
```
q = "C:/Users/Vikrant Mohite/Desktop/Applied AI/Case Study-2/output_5d/"
```

In [ ]:
```
colourmap = [[0, 0, 0], [255, 105, 180], [ 180,255,105],[ 105, 180,255], [255, 255,105]]

for k, (mask, j) in enumerate(zip(RGB_image_arr, train_df['ImageId'].values)):
    output_mask = []
    for i , color in enumerate(colourmap):
        cmap = np.all(np.equal(mask , color ), axis = -1)
        cmap.astype(int)
        output_mask.append(cmap)
    output_mask = np.stack(output_mask , axis = -1)
    output_mask = output_mask.astype(np.uint8)

    mask_name = j.split('.')[0] + '_' + str(k) + '.npy'
    mask_path = os.path.join(q, mask_name)
    save(mask_path, output_mask)
```

In [ ]:
```
train_masks_5d = os.listdir("C:/Users/Vikrant Mohite/Desktop/Applied AI/Case Study-2/output_5d")
len(train_masks_5d)
```

Out[40]:
```
7095
```

In [ ]:

Extracting mask images stored previously

In [22]:
```
mkdir mask_images1
```

In [23]:
```
# extracting mask images saved previously
# path2 = "/content/output1.zip"
path2 = "/content/drive/MyDrive/Case Study-2/output1.zip"

from pyunpack import Archive
Archive(path2).extractall('/content/mask_images1')
```

In [24]:
```
# train_masks = os.listdir("C:/Users/Vikrant Mohite/Desktop/Applied AI/Case Study-2/output1")
train_masks = os.listdir("/content/mask_images1/output1")
len(train_masks)
```

Out[24]: 7095

In [25]:
```
df1['mask_path'] = train_masks
df1.head()
```

Out[25]:

	image_path	mask_path
0	/content/train_images/0002cc93b.jpg	84e51db1c_3651_mask.png
1	/content/train_images/0007a71bf.jpg	db62a970a_6107_mask.png
2	/content/train_images/000a4bcdd.jpg	4ef6c54c9_2147_mask.png
3	/content/train_images/000f6bf48.jpg	684c99351_2823_mask.png
4	/content/train_images/0014fce06.jpg	570cb5438_2343_mask.png

In [26]:
```
# df1['mask_path'] = "C:/Users/Vikrant Mohite/Desktop/Applied AI/Case Study-2/output/" + df1['mask_path']
df1['mask_path'] = "/content/mask_images1/output1/" + df1['mask_path']
```

In [27]:
```
df1.head(10)
```

Out[27]:

	image_path	mask_path
0	/content/train_images/0002cc93b.jpg	/content/mask_images1/output1/84e51db1c_3651_mask.png
1	/content/train_images/0007a71bf.jpg	/content/mask_images1/output1/db62a970a_6107_mask.png
2	/content/train_images/000a4bcdd.jpg	/content/mask_images1/output1/4ef6c54c9_2147_mask.png
3	/content/train_images/000f6bf48.jpg	/content/mask_images1/output1/684c99351_2823_mask.png
4	/content/train_images/0014fce06.jpg	/content/mask_images1/output1/570cb5438_2343_mask.png
5	/content/train_images/0025bde0c.jpg	/content/mask_images1/output1/8f81e6d35_3949_mask.png
6	/content/train_images/0025bde0c.jpg	/content/mask_images1/output1/9c005a770_4270_mask.png
7	/content/train_images/002af848d.jpg	/content/mask_images1/output1/a1f5481df_4442_mask.png
8	/content/train_images/002fc4e19.jpg	/content/mask_images1/output1/0bb37f2aa_323_mask.png
9	/content/train_images/002fc4e19.jpg	/content/mask_images1/output1/35a8b85a8_1459_mask.png

In [28]:
```
df1['image_path'] = sorted(df1['image_path'])
df1['mask_path'] = sorted(df1['mask_path'])
df1
```

Out[28]:

	image_path	mask_path
0	/content/train_images/0002cc93b.jpg	/content/mask_images1/output1/0002cc93b_0_mask.png
1	/content/train_images/0007a71bf.jpg	/content/mask_images1/output1/0007a71bf_1_mask.png
2	/content/train_images/000a4bcdd.jpg	/content/mask_images1/output1/000a4bcdd_2_mask.png
3	/content/train_images/000f6bf48.jpg	/content/mask_images1/output1/000f6bf48_3_mask.png
4	/content/train_images/0014fce06.jpg	/content/mask_images1/output1/0014fce06_4_mask.png
...	...	...
7090	/content/train_images/ffcf72ecf.jpg	/content/mask_images1/output1/ffcf72ecf_7090_mask.png
7091	/content/train_images/fff02e9c5.jpg	/content/mask_images1/output1/fff02e9c5_7091_mask.png
7092	/content/train_images/fffe98443.jpg	/content/mask_images1/output1/fffe98443_7092_mask.png
7093	/content/train_images/ffff4eaa8.jpg	/content/mask_images1/output1/ffff4eaa8_7093_mask.png
7094	/content/train_images/fffffd67df.jpg	/content/mask_images1/output1/fffffd67df_7094_mask.png

7095 rows × 2 columns

In [ ]:
```
mkdir mask_5d
```

In [ ]:
```
q = '/content/mask_5d'
```

In [ ]:
```
colourmap = [[0, 0, 0], [255, 105, 180], [ 180,255,105],[ 105, 180,255], [255, 255,105]]

for k, (mask, j) in enumerate(zip(df1['mask_path'], train_df['ImageId'].values)):
    output_mask = []
    for i , color in enumerate(colourmap):
        image = cv2.imread(mask, cv2.IMREAD_UNCHANGED)
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        cmap = np.all(np.equal(image , color ), axis = -1)
        cmap.astype(int)
        output_mask.append(cmap)
    output_mask = np.stack(output_mask , axis = -1)
    output_mask = output_mask.astype(np.uint8)

    mask_name = j.split('.')[0] + '_' + str(k) + '.npy'
    mask_path1 = os.path.join(q, mask_name)
    save(mask_path1, output_mask)
```

In [ ]:
```
!zip -r /content/mask_5d.zip /content/mask_5d
```

In [29]:
```
mkdir mask_images3
```

In [30]:
```
# extracting mask images saved previously
# path3 = "/content/output_5d.zip"
# path3 = "/content/drive/MyDrive/Case_Study-2/output_5d.zip"
path3 = "/content/drive/MyDrive/Case Study-2/mask_5d.zip"
from pyunpack import Archive
Archive(path3).extractall('/content/mask_images3')
```

In [31]:
```
# train_masks_5d = os.listdir("/content/mask_images2/output_5d")
# train_masks_5d = os.listdir("/content/mask_5d")
train_masks_5d = os.listdir("/content/mask_images3/mask_5d")
len(train_masks_5d)
```

Out[31]:
```
7095
```

In [32]:
```
df1['mask_path_5d'] = train_masks_5d
df1.head()
```

Out[32]:

	image_path	mask_path	mask_path_5d
0	/content/train_images/0002cc93b.jpg	/content/mask_images1/output1/0002cc93b_0_mask.png	ab13f368e_4703.npy
1	/content/train_images/0007a71bf.jpg	/content/mask_images1/output1/0007a71bf_1_mask.png	29124169b_1131.npy
2	/content/train_images/000a4bcdd.jpg	/content/mask_images1/output1/000a4bcdd_2_mask.png	2e4fefc28_1268.npy
3	/content/train_images/000f6bf48.jpg	/content/mask_images1/output1/000f6bf48_3_mask.png	86aae8998_3700.npy
4	/content/train_images/0014fce06.jpg	/content/mask_images1/output1/0014fce06_4_mask.png	7de00739f_3445.npy

In [33]:
```
# df1['mask_path_5d'] = "/content/mask_images2/output_5d/" + df1['mask_path_5d']

df1['mask_path_5d'] = "/content/mask_images3/mask_5d/" + df1['mask_path_5d']
```

In [34]:
```
df1
```

Out[34]:

	image_path	mask_path	mask_path_5d
0	/content/train_images/0002cc93b.jpg	/content/mask_images1/output1/0002cc93b_0_mask.png	/content/mask_images3/mask_5d/ab13f368e_4703.npy
1	/content/train_images/0007a71bf.jpg	/content/mask_images1/output1/0007a71bf_1_mask.png	/content/mask_images3/mask_5d/29124169b_1131.npy
2	/content/train_images/000a4bcdd.jpg	/content/mask_images1/output1/000a4bcdd_2_mask.png	/content/mask_images3/mask_5d/2e4fefc28_1268.npy
3	/content/train_images/000f6bf48.jpg	/content/mask_images1/output1/000f6bf48_3_mask.png	/content/mask_images3/mask_5d/86aae8998_3700.npy
4	/content/train_images/0014fce06.jpg	/content/mask_images1/output1/0014fce06_4_mask.png	/content/mask_images3/mask_5d/7de00739f_3445.npy
...	...	...	...
7090	/content/train_images/ffcf72ecf.jpg	/content/mask_images1/output1/ffcf72ecf_7090_mask.png	/content/mask_images3/mask_5d/74cd21b41_3207.npy
7091	/content/train_images/fff02e9c5.jpg	/content/mask_images1/output1/fff02e9c5_7091_mask.png	/content/mask_images3/mask_5d/def2ff0c5_6212.npy
7092	/content/train_images/fffe98443.jpg	/content/mask_images1/output1/fffe98443_7092_mask.png	/content/mask_images3/mask_5d/f2444c793_6723.npy
7093	/content/train_images/ffff4eaa8.jpg	/content/mask_images1/output1/ffff4eaa8_7093_mask.png	/content/mask_images3/mask_5d/ba54b162d_5166.npy
7094	/content/train_images/ffffd67df.jpg	/content/mask_images1/output1/ffffd67df_7094_mask.png	/content/mask_images3/mask_5d/012a9a4c7_36.npy

7095 rows × 3 columns

In [35]:
```
df1['image_path'] = sorted(df1['image_path'])
df1['mask_path'] = sorted(df1['mask_path'])
df1['mask_path_5d'] = sorted(df1['mask_path_5d'])
df1
```

Out[35]:

	image_path	mask_path	mask_path_5d
0	/content/train_images/0002cc93b.jpg	/content/mask_images1/output1/0002cc93b_0_mask.png	/content/mask_images3/mask_5d/0002cc93b_0.npy
1	/content/train_images/0007a71bf.jpg	/content/mask_images1/output1/0007a71bf_1_mask.png	/content/mask_images3/mask_5d/0007a71bf_1.npy
2	/content/train_images/000a4bcdd.jpg	/content/mask_images1/output1/000a4bcdd_2_mask.png	/content/mask_images3/mask_5d/000a4bcdd_2.npy
3	/content/train_images/000f6bf48.jpg	/content/mask_images1/output1/000f6bf48_3_mask.png	/content/mask_images3/mask_5d/000f6bf48_3.npy
4	/content/train_images/0014fce06.jpg	/content/mask_images1/output1/0014fce06_4_mask.png	/content/mask_images3/mask_5d/0014fce06_4.npy
...	...	...	...
7090	/content/train_images/ffcf72ecf.jpg	/content/mask_images1/output1/ffcf72ecf_7090_mask.png	/content/mask_images3/mask_5d/ffcf72ecf_7090.npy
7091	/content/train_images/fff02e9c5.jpg	/content/mask_images1/output1/fff02e9c5_7091_mask.png	/content/mask_images3/mask_5d/fff02e9c5_7091.npy
7092	/content/train_images/fffe98443.jpg	/content/mask_images1/output1/fffe98443_7092_mask.png	/content/mask_images3/mask_5d/fffe98443_7092.npy
7093	/content/train_images/ffff4eaa8.jpg	/content/mask_images1/output1/ffff4eaa8_7093_mask.png	/content/mask_images3/mask_5d/ffff4eaa8_7093.npy
7094	/content/train_images/fffffd67df.jpg	/content/mask_images1/output1/fffffd67df_7094_mask.png	/content/mask_images3/mask_5d/fffffd67df_7094.npy

7095 rows × 3 columns

In [36]:
```
# d = '/content/mask_5d'
img1 = load(df1['mask_path_5d'][1960])
```

In [37]:
```
np.sum(img1)
```

Out[37]:
```
409600
```

In [ ]:
```
!zip -r /content/mask_5d.zip /content/mask_5d
```

In [ ]:
```
image = cv2.imread(df1['mask_path'][500], cv2.IMREAD_UNCHANGED)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
```

In [ ]:
```
colourmap = [[0, 0, 0], [255, 105, 180], [ 180,255,105],[ 105, 180,255], [255, 255,105]]


output_mask = []
for i , color in enumerate(colourmap):
    cmap = np.all(np.equal(image , color ), axis = -1)
    cmap.astype(int)
    output_mask.append(cmap)
output_mask = np.stack(output_mask , axis = -1)
output_mask = output_mask.astype(np.uint8)
```

In [ ]:
```
np.sum(output_mask)
```

Out[123]:
```
409600
```

In [ ]:
```
mkdir mask_5d
```

In [ ]:
```
q = '/content/mask_5d'
```

In [ ]:
```
colourmap = [[0, 0, 0], [255, 105, 180], [ 180,255,105],[ 105, 180,255], [255, 255,105]]

for k, (mask, j) in enumerate(zip(df1['mask_path'], train_df['ImageId'].values)):
    output_mask = []
    for i , color in enumerate(colourmap):
        image = cv2.imread(mask, cv2.IMREAD_UNCHANGED)
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        cmap = np.all(np.equal(image , color ), axis = -1)
        cmap.astype(int)
        output_mask.append(cmap)
    output_mask = np.stack(output_mask , axis = -1)
    output_mask = output_mask.astype(np.uint8)

    mask_name = j.split('.')[0] + '_' + str(k) + '.npy'
    mask_path1 = os.path.join(q, mask_name)
    save(mask_path1, output_mask)
```

In [ ]:
```
image = cv2.imread(df1['mask_path'][500], cv2.IMREAD_UNCHANGED)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
```

In [ ]:
```
image = cv2.imread(df1['mask_path'][500], cv2.IMREAD_UNCHANGED)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
```

In [ ]:
```

```

In [ ]:
```
img = np.argmax(img1,axis= -1) #256 X 1600
# RGB_image = []
# for outer in img1 :
#   col = []
#   for inner in outer :
#     col.append(classes_tocolour_.get(inner))
#   RGB_image.append(col)
# return np.array(RGB_image) #256 X 1600 X 3
```

In [ ]:
```
np.sum(np.argmax(img1,axis= -1))
```

Out[53]:
```
0
```

In [ ]:

In [38]:
```
from sklearn.model_selection import train_test_split
train, test = train_test_split(df1, test_size=0.3, random_state=42)
train.shape, test.shape
```

Out[38]:
```
((4966, 3), (2129, 3))
```

In [39]:
```
from sklearn.model_selection import train_test_split
train, test = train_test_split(df1, test_size=0.3, random_state=42)
test, validation = train_test_split(test, test_size=0.3, random_state=42)
test.shape, validation.shape
```

Out[39]:
```
((1490, 3), (639, 3))
```

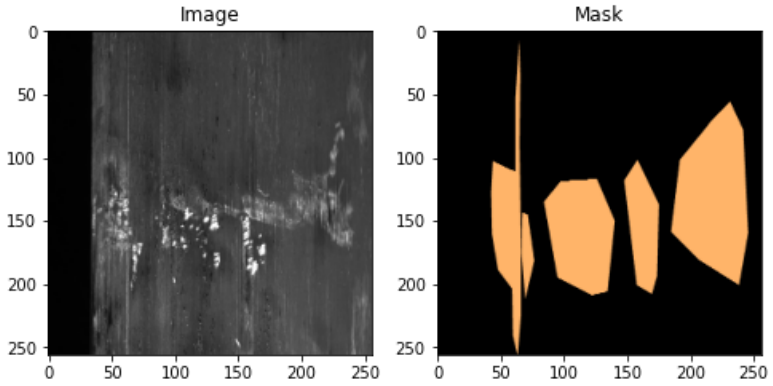
Visualizing the train images and the equivalent masks.

In [ ]:
```
images_ = train['image_path'].values
masks_ = train["mask_path"].values
lst1 = np.arange(len(images_))
len(lst1)
np.random.choice(lst1, size = 5, replace = False)
```

Out[67]:
```
array([3679, 3191, 5110, 1582, 4825])
```

In [ ]:
```
%matplotlib inline
import matplotlib.pyplot as plt
```

In [ ]:
```
# Visualizing the train images and the equivalent masks.
import random
ids = np.random.choice(lst1, size = 5, replace = False)
for i in ids:
  image = cv2.imread(images_[i], cv2.IMREAD_UNCHANGED)
  image = cv2.resize(image, (256, 256),  interpolation = cv2.INTER_AREA)
  image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
  image = np.expand_dims(image, axis=0)

  mask  = cv2.imread(masks_[i], cv2.IMREAD_UNCHANGED)
  mask = cv2.resize(mask, (256, 256),  interpolation = cv2.INTER_AREA)

  fig = plt.figure(figsize=(8,4))

  ax1 = fig.add_subplot(1, 2, 1)
  ax1.imshow(image[0,:,:])
  ax2=fig.add_subplot(1, 2, 2)
  ax2.imshow(mask)
  ax1.title.set_text('Image')
  ax2.title.set_text('Mask')
  plt.show()
```



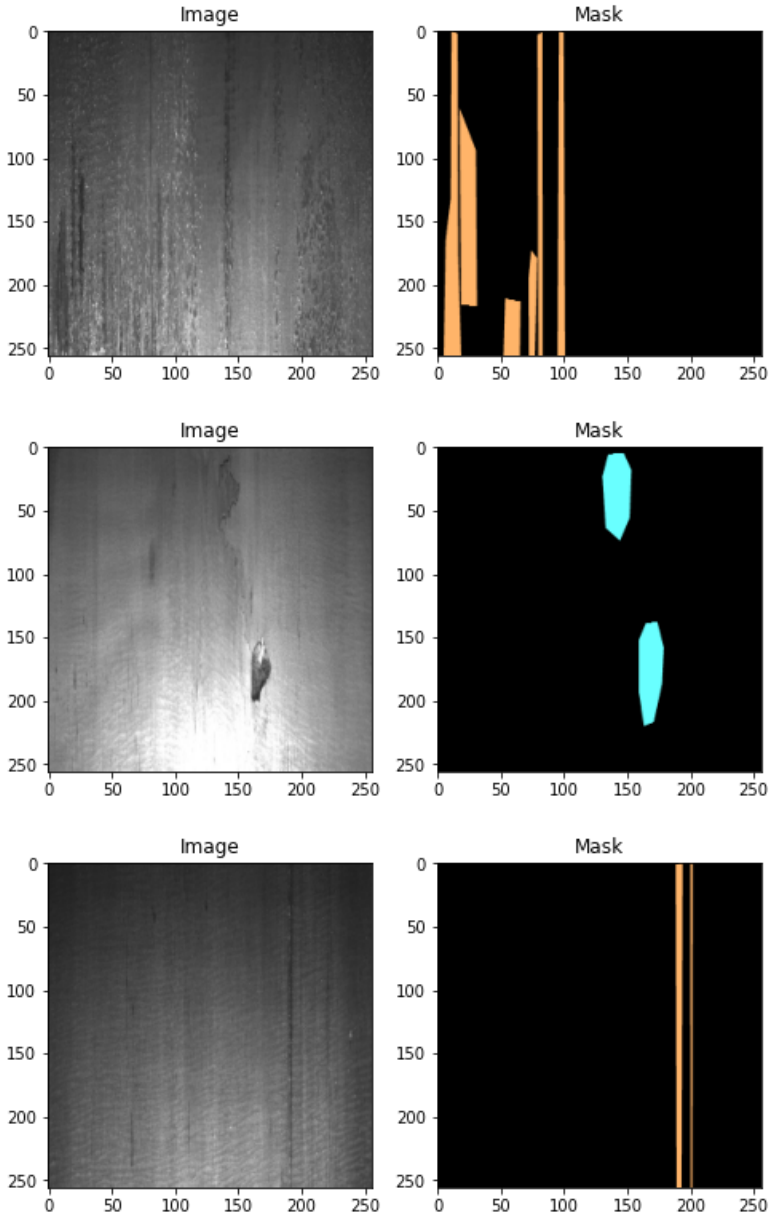
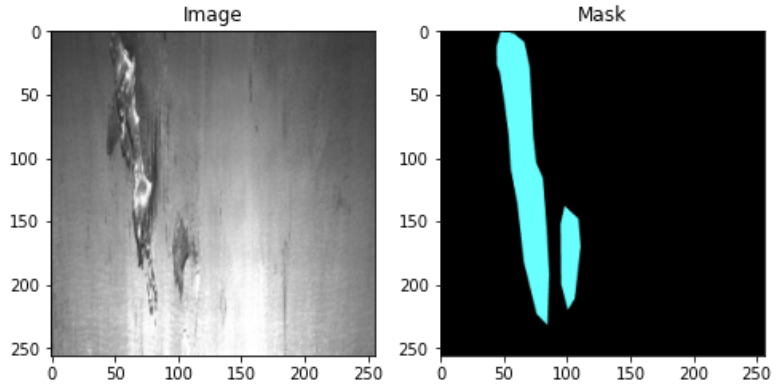


Image
Mask
Image
Mask
Image
Mask



4.2 Generating Train & Test datasets for further training

In [46]:
```
import imgaug.augmenters as iaa
# check the imgaug documentations for more augmentations
aug2 = iaa.Fliplr(1)
aug3 = iaa.Flipud(1)
aug4 = iaa.Emboss(alpha=(1), strength=1)
aug5 = iaa.DirectedEdgeDetect(alpha=(0.8), direction=(1.0))
aug6 = iaa.Sharpen(alpha=(1.0), lightness=(1.5))
```

In [ ]:
```
# def get_key(val, my_dict):
#   for key, value in my_dict.items():
#     if (val == value).all():
#       return key
```

In [ ]:
```
classes_tocolour_ =   dict({0: [0, 0, 0], 1: [255, 105, 180], 2:  [180,255,105], 3:[105, 180,255], 4: [ 255, 255,105]})
```

In [ ]:
```
# def NormalizeData(data):
#   c =  (data - np.min(data)) / (np.max(data) - np.min(data))
#   return c
```

In [ ]:
```
# class Dataset:
#     # we will be modifying this CLASSES according to your data/problems
#     CLASSES = [0,1,2,3,4]

#     classes_tocolour_ =   dict({0: [0, 0, 0], 1: [255, 105, 180], 2:  [180,255,105], 3:[105, 180,255], 4: [ 255, 255,105]})

#     # the parameters needs to changed based on your requirements
#     # here we are collecting the file_names because in our dataset, both our images and maks will have same file name
#     # ex: fil_name.jpg   file_name.mask.jpg
#     def __init__(self, file_names , w, h, classes=None):

#         self.ids = file_names
#         # the paths of images
#         self.images_fps   = [image_id for image_id in self.ids["image_path"]]
#         # the paths of segmentation images
#         self.masks_fps    = [image_id for image_id in self.ids["mask_path"]]
#         # giving labels for each class
#         self.class_values = [self.CLASSES.index(cls) for cls in classes]
#         self.w = w
#         self.h = h


#     def __getitem__(self, i): #https://omkarpathak.in/2018/04/11/python-getitem-and-setitem/

#         # read data
#         image = cv2.imread(self.images_fps[i], cv2.IMREAD_UNCHANGED)
#         image = cv2.resize(image, (self.w, self.h),  interpolation = cv2.INTER_AREA)
#         image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB).astype('float')
#         # image = NormalizeData(image)


#         mask  = cv2.imread(self.masks_fps[i])
#         mask = cv2.resize(mask, (self.w, self.h),  interpolation = cv2.INTER_AREA)
#         mask = cv2.cvtColor(mask, cv2.COLOR_BGR2RGB)

#         s = []
#         for d in mask:
#           k = []
#           for j in d:
#             p = get_key(j, classes_tocolour_)
#             k.append(p)
#           s.append(k)
#         mask = np.array(s)



#         image_mask = mask


#         image_masks = [(image_mask == v) for v in self.class_values]

#         image_mask = np.stack(image_masks, axis=-1).astype('float')
```

```
#         # image_mask = cv2.resize(image_mask, (self.w, self.h))

#         a = np.random.uniform()
#         if a<0.2:
#             image = aug2.augment_image(image)
#             image_mask = aug2.augment_image(image_mask)
#         elif a<0.4:
#             image = aug3.augment_image(image)
#             image_mask = aug3.augment_image(image_mask)
#         elif a<0.6:
#             image = aug4.augment_image(image)
#             image_mask = aug4.augment_image(image_mask)
#         elif a<0.8:
#             image = aug5.augment_image(image)
#             image_mask = image_mask
#         else:
#             image = aug6.augment_image(image)
#             image_mask = aug6.augment_image(image_mask)
#         # mask = NormalizeData(mask)
#         return image, image_mask

#     def __len__(self):
#         return len(self.ids)
```

In [40]:
```
class Dataset:

    def __init__(self, data_df):

        self.images_fps   = data_df['image_path'].values

        self.masks_fps  = data_df['mask_path_5d'].values

#         self.w = 256
#         self.h = 1600


    def __getitem__(self, i): #https://omkarpathak.in/2018/04/11/python-getitem-and-setitem/

        # read data
        image = cv2.imread(self.images_fps[i], cv2.IMREAD_UNCHANGED)
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB).astype('float')
        # image = NormalizeData(image)

        image_mask = load(self.masks_fps[i])

        image_mask = image_mask.astype('float')

        a = np.random.uniform()
        if a<0.2:
            image = aug2.augment_image(image)
            image_mask = aug2.augment_image(image_mask)
        elif a<0.4:
            image = aug3.augment_image(image)
            image_mask = aug3.augment_image(image_mask)
        elif a<0.6:
            image = aug4.augment_image(image)
            image_mask = aug4.augment_image(image_mask)
        elif a<0.8:
            image = aug5.augment_image(image)
            image_mask = image_mask
        else:
            image = aug6.augment_image(image)
            image_mask = aug6.augment_image(image_mask)
        # mask = NormalizeData(mask)
        return image, image_mask

    def __len__(self):
        return len(self.images_fps)
```

In [41]:
```
class Dataloder(tf.keras.utils.Sequence):
    def __init__(self, dataset, batch_size=1, shuffle=False):
        self.dataset = dataset
        self.batch_size = batch_size
        self.shuffle = shuffle
        self.indexes = np.arange(len(dataset))

    def __getitem__(self, i):

        # collect batch data
        start = i * self.batch_size
        stop = (i + 1) * self.batch_size
        data = []
        for j in range(start, stop):
            data.append(self.dataset[j])

        batch = [np.stack(samples, axis=0) for samples in zip(*data)]

        return tuple(batch)

    def __len__(self):
        return len(self.indexes) // self.batch_size

    def on_epoch_end(self):
        if self.shuffle:
            self.indexes = np.random.permutation(self.indexes)
```

In [ ]:
```
# CLASSES = [0, 1, 2, 3, 4]
# train_dataset = Dataset(train, 256, 256, classes=CLASSES)
# test_dataset  = Dataset(test, 256, 256, classes=CLASSES)
```

In [42]:
```
train_dataset = Dataset(train)
test_dataset  = Dataset(test)
```

In [43]:
```
len(test_dataset)
```

Out[43]:
```
1490
```

In [44]:
```
len(train_dataset)
```

Out[44]:
```
4966
```

In [47]:
```
np.sum(train_dataset[1800][1])
```

Out[47]:
```
409600.0
```

In [ ]:
```
BATCH_SIZE=10
train_dataloader = Dataloder(train_dataset, batch_size=BATCH_SIZE, shuffle=True)
test_dataloader = Dataloder(test_dataset, batch_size=BATCH_SIZE, shuffle=True)

assert train_dataloader[0][0].shape == (BATCH_SIZE, 256, 1600, 3)
assert train_dataloader[0][1].shape == (BATCH_SIZE, 256, 1600, 5)

print(train_dataloader[0][0].shape)
print(train_dataloader[0][1].shape)
print(len(train_dataloader))
print(len(test_dataloader))
type(train_dataset[0])
```

```
(10, 256, 1600, 3)
(10, 256, 1600, 5)
496
149
```

Out[51]:
```
tuple
```

In [ ]:
```
# image_masks[4].shape
```

Out[88]:
```
(256, 1600, 3)
```

4.3 Applying Unet to segment the images

In [11]:
```
!pip install segmentation_models
```

```
Looking in indexes: https://pypi.org/simple, (https://pypi.org/simple,) https://us-python.pkg.dev/colab-wheels/public/simple/ (https://us-python.pkg.dev/colab-wheels/public/simple/)
Collecting segmentation_models
  Downloading segmentation_models-1.0.1-py3-none-any.whl (33 kB)
Collecting image-classifiers==1.0.0
  Downloading image_classifiers-1.0.0-py3-none-any.whl (19 kB)
Collecting keras-applications<=1.0.8,>=1.0.7
  Downloading Keras_Applications-1.0.8-py3-none-any.whl (50 kB)
     |████████████████████████████████| 50 kB 6.4 MB/s
Collecting efficientnet==1.0.0
  Downloading efficientnet-1.0.0-py3-none-any.whl (17 kB)
Requirement already satisfied: scikit-image in /usr/local/lib/python3.8/dist-packages (from efficientnet==1.0.0->segmentation_models) (0.18.3)
Requirement already satisfied: h5py in /usr/local/lib/python3.8/dist-packages (from keras-applications<=1.0.8,>=1.0.7->segmentation_models) (3.1.0)
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.8/dist-packages (from keras-applications<=1.0.8,>=1.0.7->segmentation_models) (1.21.6)
Requirement already satisfied: pillow!=7.1.0,!=7.1.1,>=4.3.0 in /usr/local/lib/python3.8/dist-packages (from scikit-image->efficientnet==1.0.0->segmentation_models) (7.1.2)
Requirement already satisfied: imageio>=2.3.0 in /usr/local/lib/python3.8/dist-packages (from scikit-image->efficientnet==1.0.0->segmentation_models) (2.9.0)
Requirement already satisfied: matplotlib!=3.0.0,>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from scikit-image->efficientnet==1.0.0->segmentation_models) (3.2.2)
Requirement already satisfied: scipy>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from scikit-image->efficientnet==1.0.0->segmentation_models) (1.7.3)
Requirement already satisfied: networkx>=2.0 in /usr/local/lib/python3.8/dist-packages (from scikit-image->efficientnet==1.0.0->segmentation_models) (2.8.8)
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.8/dist-packages (from scikit-image->efficientnet==1.0.0->segmentation_models) (2022.10.10)
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.8/dist-packages (from scikit-image->efficientnet==1.0.0->segmentation_models) (1.4.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib!=3.0.0,>=2.0.0->scikit-image->efficientnet==1.0.0->segmentation_models) (1.4.4)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib!=3.0.0,>=2.0.0->scikit-image->efficientnet==1.0.0->segmentation_models) (3.0.9)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-packages (from matplotlib!=3.0.0,>=2.0.0->scikit-image->efficientnet==1.0.0->segmentation_models) (0.11.0)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib!=3.0.0,>=2.0.0->scikit-image->efficientnet==1.0.0->segmentation_models) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from python-dateutil>=2.1->matplotlib!=3.0.0,>=2.0.0->scikit-image->efficientnet==1.0.0->segmentation_models) (1.15.0)
Installing collected packages: keras-applications, image-classifiers, efficientnet, segmentation-models
Successfully installed efficientnet-1.0.0 image-classifiers-1.0.0 keras-applications-1.0.8 segmentation-models-1.0.1
```

In [49]:
```
tf.keras.backend.clear_session()
```

In [12]:
```
# We are importing the pretrained unet from the segmentation models.
# https://github.com/qubvel/segmentation_models

tf.keras.backend.clear_session()
import segmentation_models as sm
from segmentation_models import Unet
from segmentation_models.metrics import iou_score
sm.set_framework('tf.keras')

tf.keras.backend.set_image_data_format('channels_last')
#from tensorflow.keras.applications.resnet import preprocess_input

# loading the unet model and using the resnet 50 and initialized weights with Imagenet weights.
# "classes" :different types of classes in the dataset.
backbone = 'resnet50'
preprocess_input = sm.get_preprocessing(backbone)
```

```
Segmentation Models: using `keras` framework.
```

In [ ]:
```
IMAGE_SHAPE = (256, 1600, 3)
```

In [ ]:
```
model = Unet(backbone_name = backbone, input_shape = IMAGE_SHAPE, classes = 5, activation = 'softmax', encoder_freeze = True,
             encoder_weights = 'imagenet', decoder_block_type = 'upsampling')
```

```
Downloading data from https://github.com/qubvel/classification_models/releases/download/0.0.1/resnet50_imagenet_1000_no_top.h5 (https://github.com/qubvel/classification_models/releases/download/0.0.1/resnet50_imagenet_1000_no_top.h5)
94592056/94592056 [==============================] - 5s 0us/step
```

In [ ]:
```
import tensorflow as tf
from keras.callbacks import Callback

class stop_at_iou(Callback):
        def __init__(self,model):
          self.model = model
          self.iou = .99
        def on_epoch_end(self,epoch,logs={}):
            if logs.get('val_iou_score') >= self.iou:
                 self.model.stop_training = True
            return
```

In [ ]:
```
# stopper = stop_at_iou(model)
```

In [ ]:
```
import tensorflow as tf
from tensorflow.keras import callbacks
optim = tf.keras.optimizers.Adam(learning_rate= 0.001)
focal_loss = sm.losses.cce_dice_loss
model.compile(optim, focal_loss, metrics=[iou_score])
callbacks = [callbacks.ModelCheckpoint('./best_model.h5', save_weights_only = True, save_best_only = True, \
                                       mode = 'max', monitor = 'val_iou_score', verbose = 1),
    callbacks.ReduceLROnPlateau(monitor = 'val_iou_score', patience = 3, mode = 'max', verbose = 1,min_lr=0.0001,factor=0.2)
]
```

In [ ]:
```
history = model.fit(train_dataloader, steps_per_epoch=(len(train_dataloader))//BATCH_SIZE, epochs=8,\
                    validation_data=test_dataloader, callbacks=callbacks)
```

```
Epoch 1/8
56/56 [==============================] - ETA: 0s - loss: 0.8481 - iou_score: 0.2125
Epoch 1: val_iou_score improved from -inf to 0.01705, saving model to ./best_model.h5
56/56 [==============================] - 1000s 18s/step - loss: 0.8481 - iou_score: 0.2125 - val_loss: 4.1532 - val_iou_score: 0.0171 - lr: 0.0090
Epoch 2/8
56/56 [==============================] - ETA: 0s - loss: 0.7983 - iou_score: 0.2298
Epoch 2: val_iou_score improved from 0.01705 to 0.14673, saving model to ./best_model.h5
56/56 [==============================] - 901s 16s/step - loss: 0.7983 - iou_score: 0.2298 - val_loss: 1.0848 - val_iou_score: 0.1467 - lr: 0.0090
Epoch 3/8
56/56 [==============================] - ETA: 0s - loss: 0.7798 - iou_score: 0.2443
Epoch 3: val_iou_score improved from 0.14673 to 0.18396, saving model to ./best_model.h5
56/56 [==============================] - 806s 15s/step - loss: 0.7798 - iou_score: 0.2443 - val_loss: 1.0159 - val_iou_score: 0.1840 - lr: 0.0090
Epoch 4/8
56/56 [==============================] - ETA: 0s - loss: 0.7610 - iou_score: 0.2548
Epoch 4: val_iou_score did not improve from 0.18396
56/56 [==============================] - 807s 15s/step - loss: 0.7610 - iou_score: 0.2548 - val_loss: 1.3220 - val_iou_score: 0.1533 - lr: 0.0090
Epoch 5/8
56/56 [==============================] - ETA: 0s - loss: 0.7530 - iou_score: 0.2603
Epoch 5: val_iou_score did not improve from 0.18396
56/56 [==============================] - 789s 14s/step - loss: 0.7530 - iou_score: 0.2603 - val_loss: 1.6251 - val_iou_score: 0.1021 - lr: 0.0090
Epoch 6/8
56/56 [==============================] - ETA: 0s - loss: 0.7566 - iou_score: 0.2546
Epoch 6: val_iou_score did not improve from 0.18396

Epoch 6: ReduceLROnPlateau reducing learning rate to 0.001799999922513962.
56/56 [==============================] - 802s 15s/step - loss: 0.7566 - iou_score: 0.2546 - val_loss: 1.1128 - val_iou_score: 0.1105 - lr: 0.0090
Epoch 7/8
56/56 [==============================] - ETA: 0s - loss: 0.7296 - iou_score: 0.2717
Epoch 7: val_iou_score improved from 0.18396 to 0.24066, saving model to ./best_model.h5
56/56 [==============================] - 789s 14s/step - loss: 0.7296 - iou_score: 0.2717 - val_loss: 0.7883 - val_iou_score: 0.2407 - lr: 0.0018
Epoch 8/8
56/56 [==============================] - ETA: 0s - loss: 0.7282 - iou_score: 0.2759
Epoch 8: val_iou_score improved from 0.24066 to 0.27711, saving model to ./best_model.h5
56/56 [==============================] - 802s 15s/step - loss: 0.7282 - iou_score: 0.2759 - val_loss: 0.7259 - val_iou_score: 0.2771 - lr: 0.0018
```

In [ ]:
```
history = model.fit(train_dataloader, steps_per_epoch=(len(train_dataloader))//BATCH_SIZE, epochs=50,\
                    validation_data=test_dataloader, callbacks=callbacks)
```

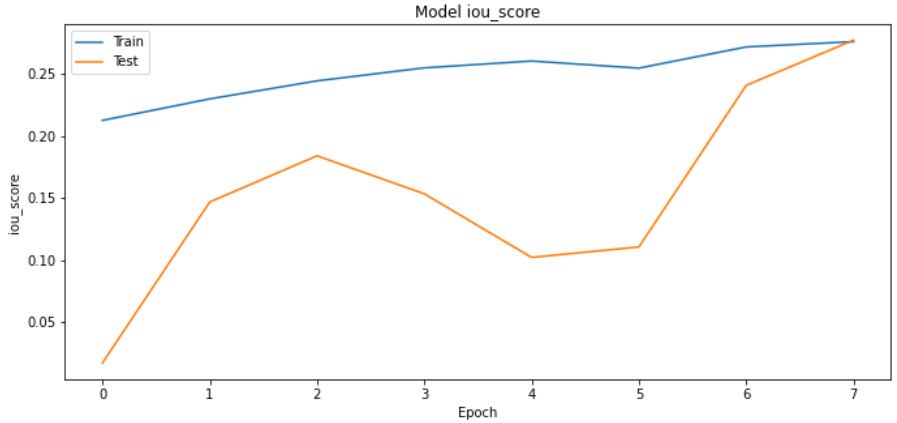
```
Epoch 1/50
56/56 [==============================] - ETA: 0s - loss: 0.9464 - iou_score: 0.1605
Epoch 1: val_iou_score improved from -inf to 0.03908, saving model to ./best_model.h5
56/56 [==============================] - 201s 3s/step - loss: 0.9464 - iou_score: 0.1605 - val_loss: 1.8377 - val_iou_score: 0.0391 - lr: 0.0010
Epoch 2/50
56/56 [==============================] - ETA: 0s - loss: 0.7054 - iou_score: 0.2933
Epoch 2: val_iou_score improved from 0.03908 to 0.22526, saving model to ./best_model.h5
56/56 [==============================] - 170s 3s/step - loss: 0.7054 - iou_score: 0.2933 - val_loss: 0.8251 - val_iou_score: 0.2253 - lr: 0.0010
Epoch 3/50
56/56 [==============================] - ETA: 0s - loss: 0.6817 - iou_score: 0.3164
Epoch 3: val_iou_score improved from 0.22526 to 0.23494, saving model to ./best_model.h5
56/56 [==============================] - 173s 3s/step - loss: 0.6817 - iou_score: 0.3164 - val_loss: 1.0161 - val_iou_score: 0.2349 - lr: 0.0010
Epoch 4/50
56/56 [==============================] - ETA: 0s - loss: 0.6766 - iou_score: 0.3222
Epoch 4: val_iou_score improved from 0.23494 to 0.29680, saving model to ./best_model.h5
56/56 [==============================] - 223s 4s/step - loss: 0.6766 - iou_score: 0.3222 - val_loss: 0.7382 - val_iou_score: 0.2968 - lr: 0.0010
Epoch 5/50
56/56 [==============================] - ETA: 0s - loss: 0.6449 - iou_score: 0.3479
Epoch 5: val_iou_score did not improve from 0.29680
56/56 [==============================] - 171s 3s/step - loss: 0.6449 - iou_score: 0.3479 - val_loss: 0.7446 - val_iou_score: 0.2870 - lr: 0.0010
Epoch 6/50
56/56 [==============================] - ETA: 0s - loss: 0.6449 - iou_score: 0.3505
Epoch 6: val_iou_score improved from 0.29680 to 0.29799, saving model to ./best_model.h5
56/56 [==============================] - 172s 3s/step - loss: 0.6449 - iou_score: 0.3505 - val_loss: 0.7216 - val_iou_score: 0.2980 - lr: 0.0010
Epoch 7/50
56/56 [==============================] - ETA: 0s - loss: 0.6397 - iou_score: 0.3527
Epoch 7: val_iou_score improved from 0.29799 to 0.30711, saving model to ./best_model.h5
56/56 [==============================] - 171s 3s/step - loss: 0.6397 - iou_score: 0.3527 - val_loss: 0.7112 - val_iou_score: 0.3071 - lr: 0.0010
Epoch 8/50
56/56 [==============================] - ETA: 0s - loss: 0.6333 - iou_score: 0.3592
Epoch 8: val_iou_score improved from 0.30711 to 0.32005, saving model to ./best_model.h5
56/56 [==============================] - 172s 3s/step - loss: 0.6333 - iou_score: 0.3592 - val_loss: 0.6996 - val_iou_score: 0.3200 - lr: 0.0010
Epoch 9/50
56/56 [==============================] - ETA: 0s - loss: 0.6241 - iou_score: 0.3670
Epoch 9: val_iou_score improved from 0.32005 to 0.32689, saving model to ./best_model.h5
56/56 [==============================] - 175s 3s/step - loss: 0.6241 - iou_score: 0.3670 - val_loss: 0.6863 - val_iou_score: 0.3269 - lr: 0.0010
Epoch 10/50
56/56 [==============================] - ETA: 0s - loss: 0.6453 - iou_score: 0.3478
Epoch 10: val_iou_score did not improve from 0.32689
56/56 [==============================] - 173s 3s/step - loss: 0.6453 - iou_score: 0.3478 - val_loss: 0.7184 - val_iou_score: 0.2987 - lr: 0.0010
Epoch 11/50
56/56 [==============================] - ETA: 0s - loss: 0.6283 - iou_score: 0.3620
Epoch 11: val_iou_score improved from 0.32689 to 0.33824, saving model to ./best_model.h5
56/56 [==============================] - 173s 3s/step - loss: 0.6283 - iou_score: 0.3620 - val_loss: 0.6641 - val_iou_score: 0.3382 - lr: 0.0010
Epoch 12/50
56/56 [==============================] - ETA: 0s - loss: 0.6097 - iou_score: 0.3814
Epoch 12: val_iou_score did not improve from 0.33824
56/56 [==============================] - 173s 3s/step - loss: 0.6097 - iou_score: 0.3814 - val_loss: 0.6659 - val_iou_score: 0.3357 - lr: 0.0010
Epoch 13/50
56/56 [==============================] - ETA: 0s - loss: 0.6430 - iou_score: 0.3492
Epoch 13: val_iou_score improved from 0.33824 to 0.34292, saving model to ./best_model.h5
56/56 [==============================] - 172s 3s/step - loss: 0.6430 - iou_score: 0.3492 - val_loss: 0.6616 - val_iou_score: 0.3429 - lr: 0.0010
```

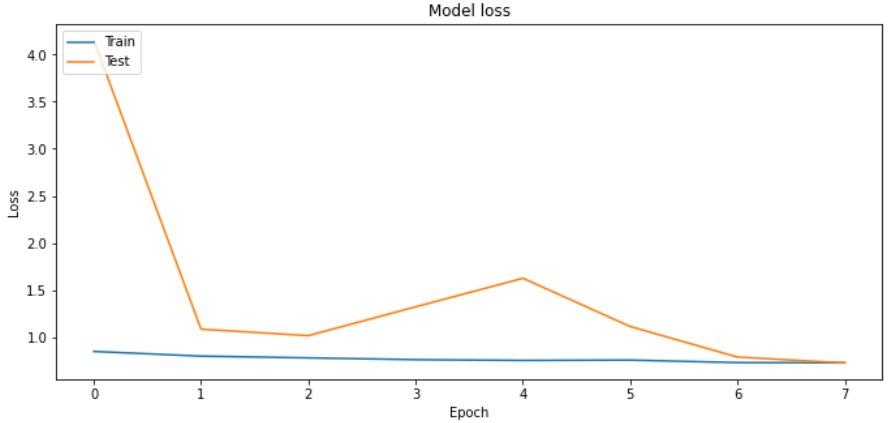
```
Epoch 14/50
56/56 [==============================] - ETA: 0s - loss: 0.6125 - iou_score: 0.3739
Epoch 14: val_iou_score improved from 0.34292 to 0.35121, saving model to ./best_model.h5
56/56 [==============================] - 173s 3s/step - loss: 0.6125 - iou_score: 0.3739 - val_loss: 0.6409 - val_iou_score: 0.3512 - lr: 0.0010
Epoch 15/50
56/56 [==============================] - ETA: 0s - loss: 0.6257 - iou_score: 0.3677
Epoch 15: val_iou_score did not improve from 0.35121
56/56 [==============================] - 172s 3s/step - loss: 0.6257 - iou_score: 0.3677 - val_loss: 0.6580 - val_iou_score: 0.3395 - lr: 0.0010
Epoch 16/50
56/56 [==============================] - ETA: 0s - loss: 0.6036 - iou_score: 0.3841
Epoch 16: val_iou_score did not improve from 0.35121
56/56 [==============================] - 170s 3s/step - loss: 0.6036 - iou_score: 0.3841 - val_loss: 0.6790 - val_iou_score: 0.3313 - lr: 0.0010
Epoch 17/50
56/56 [==============================] - ETA: 0s - loss: 0.6040 - iou_score: 0.3816
Epoch 17: val_iou_score improved from 0.35121 to 0.35233, saving model to ./best_model.h5
56/56 [==============================] - 172s 3s/step - loss: 0.6040 - iou_score: 0.3816 - val_loss: 0.6553 - val_iou_score: 0.3523 - lr: 0.0010
Epoch 18/50
56/56 [==============================] - ETA: 0s - loss: 0.6110 - iou_score: 0.3753
Epoch 18: val_iou_score did not improve from 0.35233
56/56 [==============================] - 170s 3s/step - loss: 0.6110 - iou_score: 0.3753 - val_loss: 0.6451 - val_iou_score: 0.3494 - lr: 0.0010
Epoch 19/50
56/56 [==============================] - ETA: 0s - loss: 0.6221 - iou_score: 0.3646
Epoch 19: val_iou_score improved from 0.35233 to 0.36218, saving model to ./best_model.h5
56/56 [==============================] - 170s 3s/step - loss: 0.6221 - iou_score: 0.3646 - val_loss: 0.6385 - val_iou_score: 0.3622 - lr: 0.0010
Epoch 20/50
56/56 [==============================] - ETA: 0s - loss: 0.6080 - iou_score: 0.3755
Epoch 20: val_iou_score did not improve from 0.36218
56/56 [==============================] - 171s 3s/step - loss: 0.6080 - iou_score: 0.3755 - val_loss: 0.6733 - val_iou_score: 0.3309 - lr: 0.0010
Epoch 21/50
56/56 [==============================] - ETA: 0s - loss: 0.5882 - iou_score: 0.3930
Epoch 21: val_iou_score improved from 0.36218 to 0.37297, saving model to ./best_model.h5
56/56 [==============================] - 169s 3s/step - loss: 0.5882 - iou_score: 0.3930 - val_loss: 0.6285 - val_iou_score: 0.3730 - lr: 0.0010
```

In [ ]:

4.4 Visualizing the result of Unet

In [ ]:
```
# Plot training & validation iou_score values
plt.figure(figsize=(25, 5))
plt.subplot(1, 2, 1)
plt.plot(history.history['iou_score'])
plt.plot(history.history['val_iou_score'])
plt.title('Model iou_score')
plt.ylabel('iou_score')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')

# Plot training & validation loss values
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```





In [ ]:
```
import matplotlib.pyplot as plt
plt.figure(figsize=(20, 5))
plt.subplot(1, 2, 1)
plt.show()
```

In [ ]:
```
images_ = test['image_path'].values
masks_ = test["mask_path"].values
lst = np.arange(len(images_))
len(lst)
```

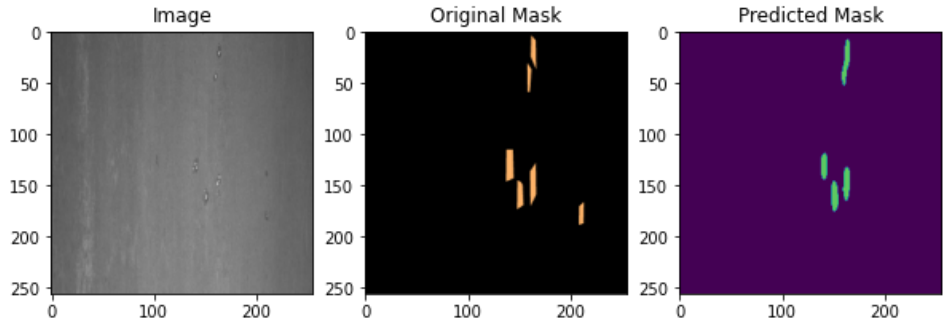
Out[72]:
```
1419
```

In [ ]:
```
np.random.choice(lst, size = 20, replace = False)
```

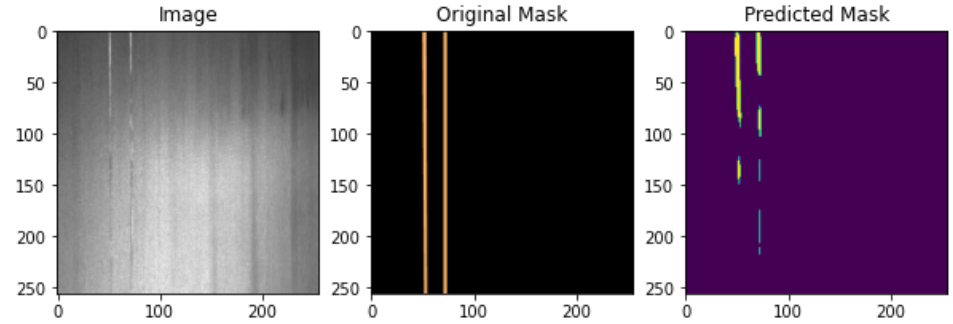
Out[73]:
```
array([  71,  296,  454, 1212,  400, 1012,  225,  421, 1341,   93,  935,
        625,  820,  175,  690,  277,  684,    8,  833, 1117])
```

In [ ]:
```
# Visualizing the predicted mask for test data.
import random
ids = np.random.choice(lst, size = 20, replace = False)
for i in ids:
  image = cv2.imread(images_[i], cv2.IMREAD_UNCHANGED)
  image = cv2.resize(image, (256, 256),  interpolation = cv2.INTER_AREA)
  image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
  image = np.expand_dims(image, axis=0)

  mask  = cv2.imread(masks_[i], cv2.IMREAD_UNCHANGED)
  mask = cv2.resize(mask, (256, 256),  interpolation = cv2.INTER_AREA)

  pred = model.predict(image,verbose=1)
  pred = tf.argmax(pred, axis=-1)

  fig = plt.figure(figsize=(10,6))

  ax1 = fig.add_subplot(1, 3, 1)
  ax1.imshow(image[0,:,:])
  ax2=fig.add_subplot(1, 3, 2)
  ax2.imshow(mask)
  ax3=fig.add_subplot(1, 3, 3)
  ax3.imshow(pred[0,:,:])
  ax1.title.set_text('Image')
  ax2.title.set_text('Original Mask')
  ax3.title.set_text('Predicted Mask')
  plt.show()
```

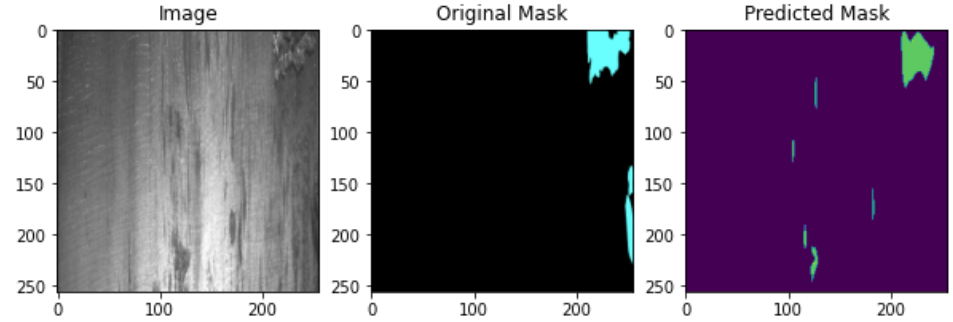
```
1/1 [==============================] - 0s 23ms/step
```



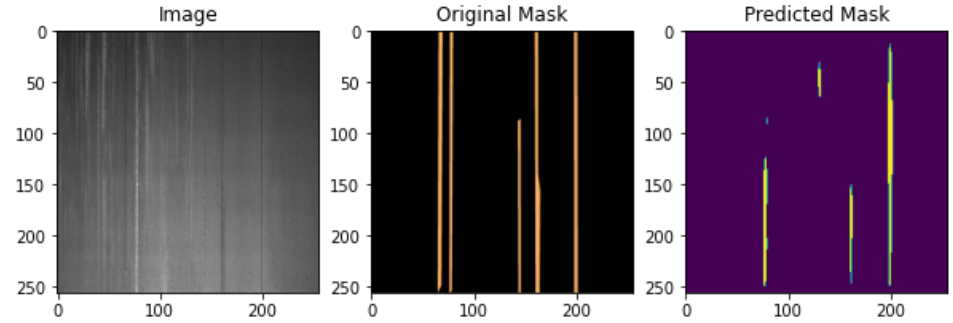

```
1/1 [==============================] - 0s 22ms/step
```



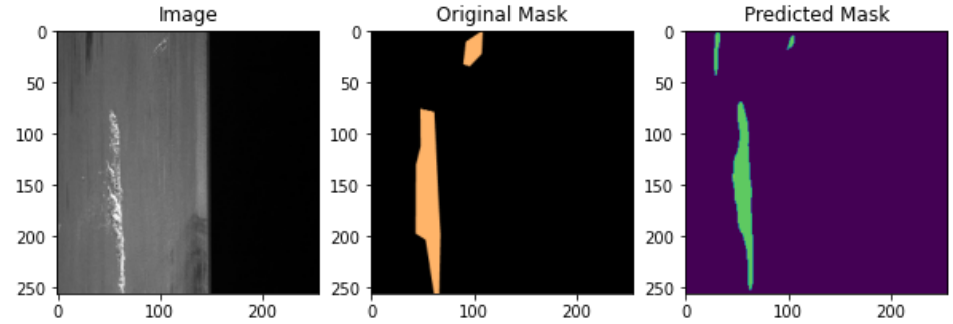

```
1/1 [==============================] - 0s 24ms/step
```



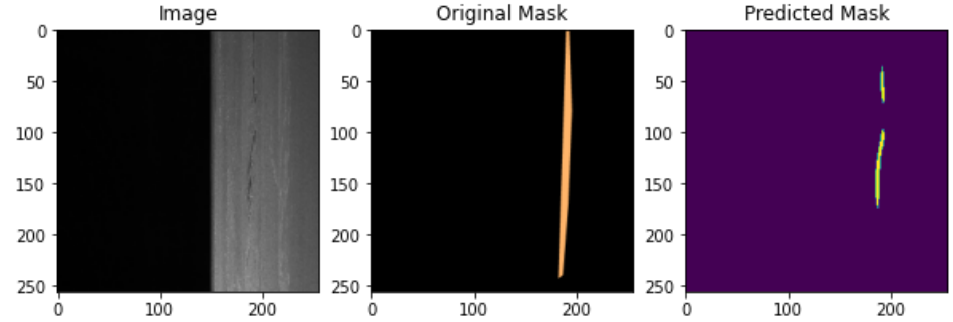

```
1/1 [==============================] - 0s 48ms/step
```



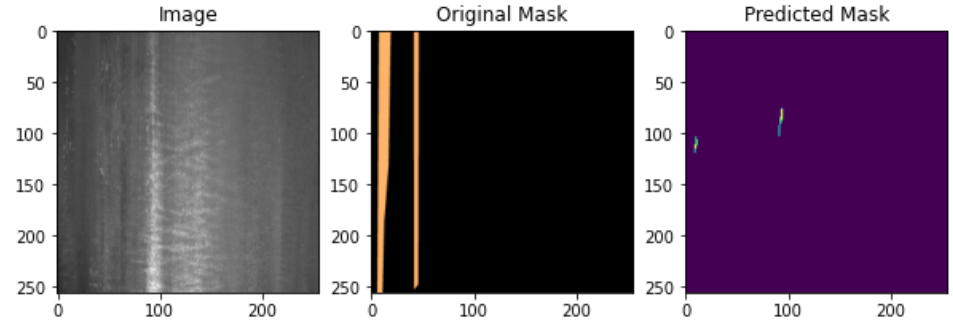

```
1/1 [==============================] - 0s 21ms/step
```



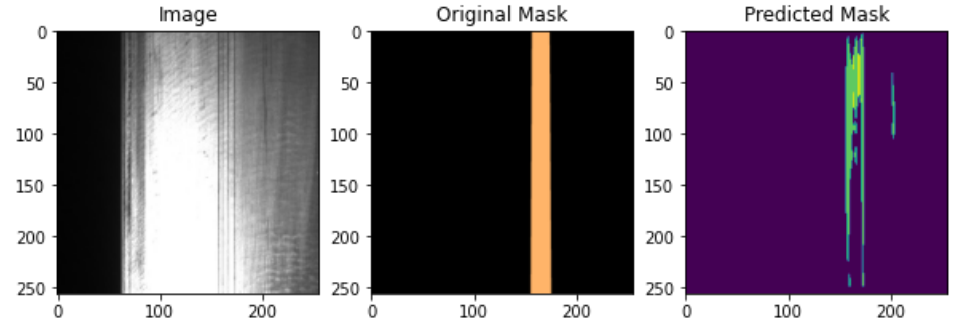

```
1/1 [==============================] - 0s 24ms/step
```



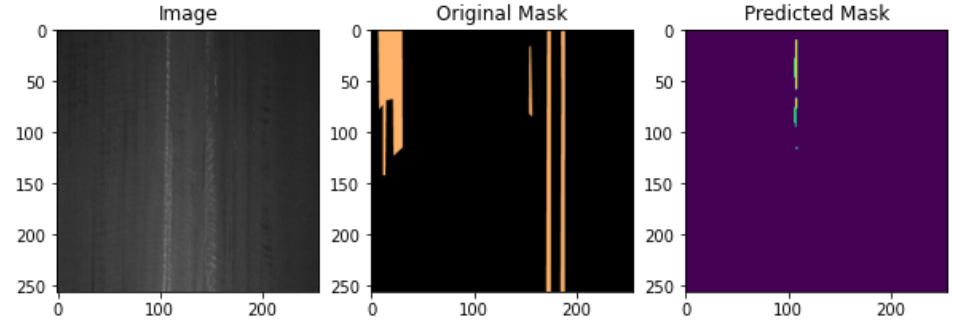

```
1/1 [==============================] - 0s 23ms/step
```



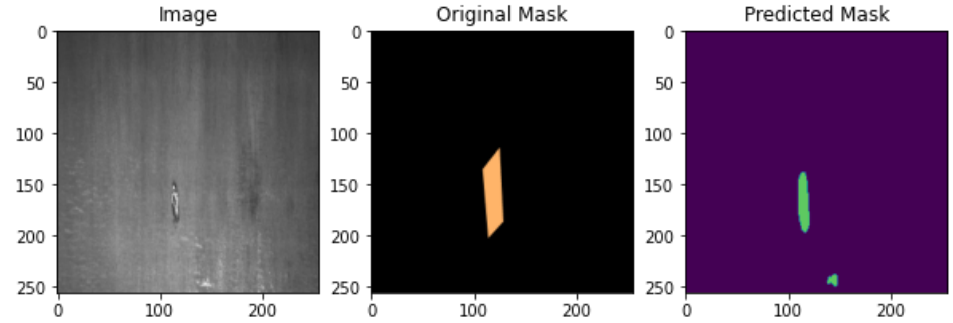

```
1/1 [==============================] - 0s 35ms/step
```



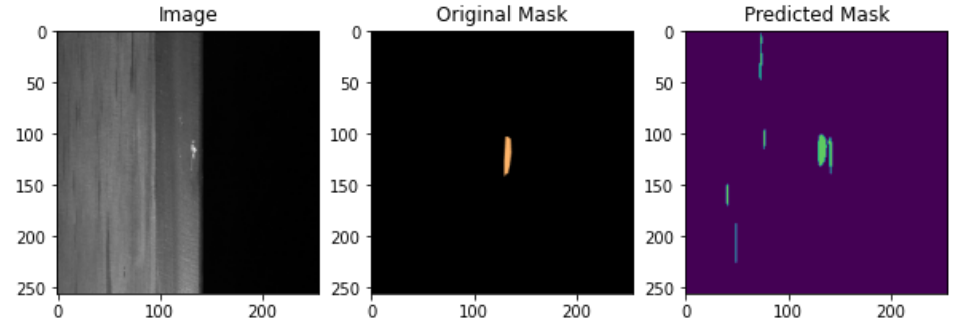

```
1/1 [==============================] - 0s 22ms/step
```



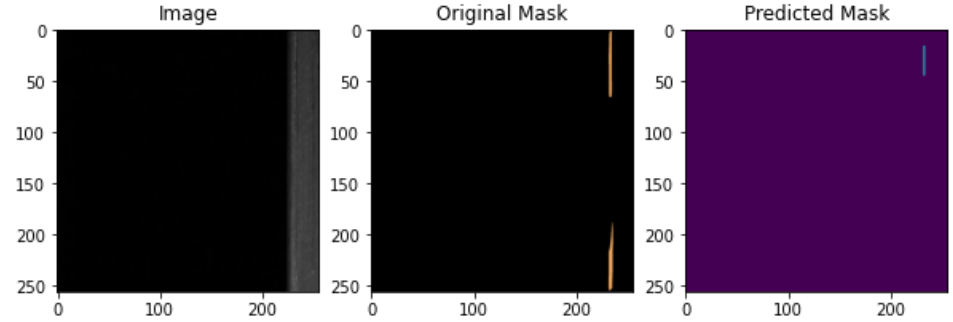

```
1/1 [==============================] - 0s 33ms/step
```



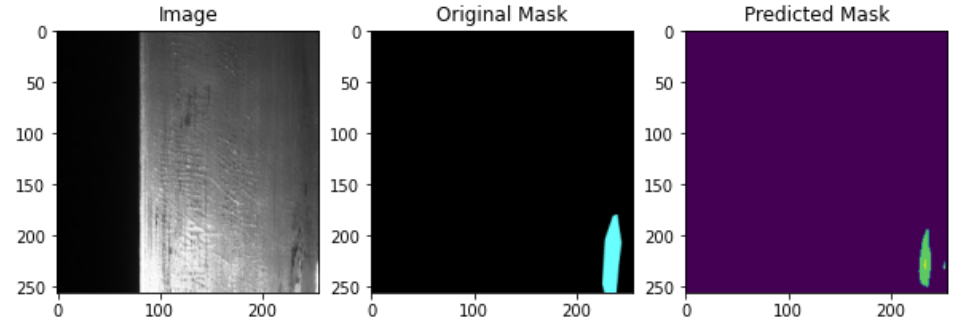

```
1/1 [==============================] - 0s 23ms/step
```



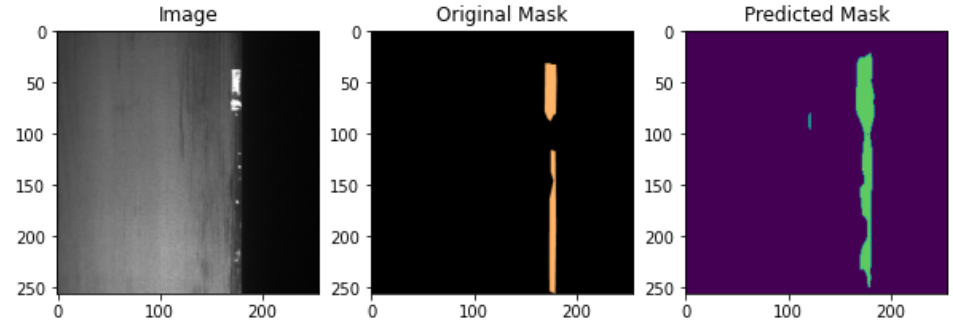

```
1/1 [==============================] - 0s 21ms/step
```



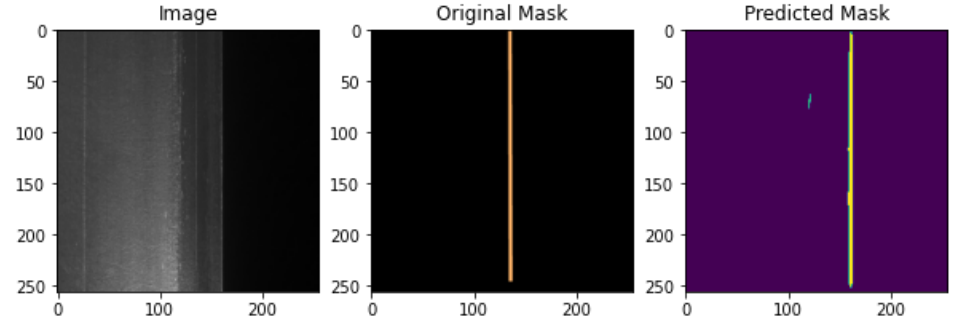

```
1/1 [==============================] - 0s 24ms/step
```



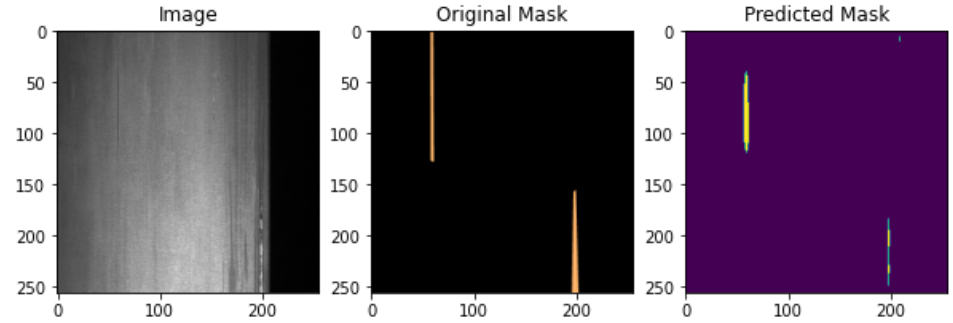

```
1/1 [==============================] - 0s 29ms/step
```



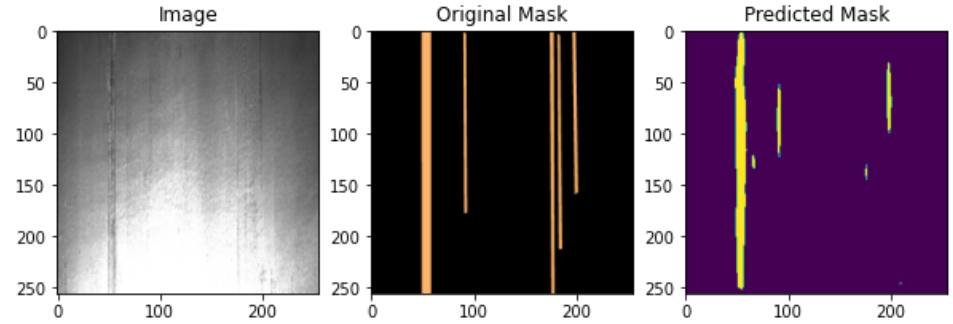

```
1/1 [==============================] - 0s 22ms/step
```



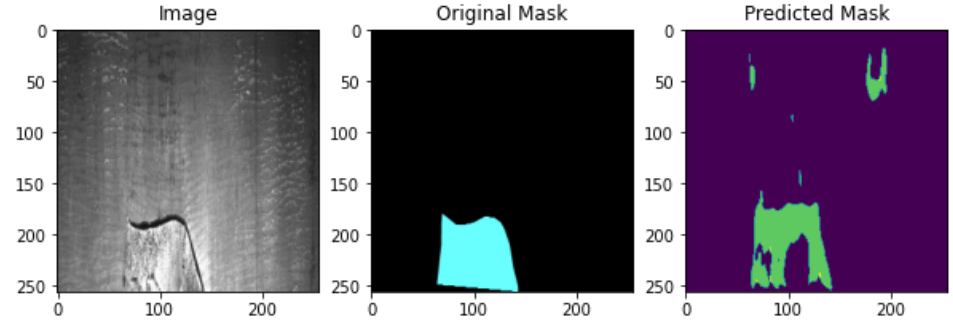

```
1/1 [==============================] - 0s 22ms/step
```



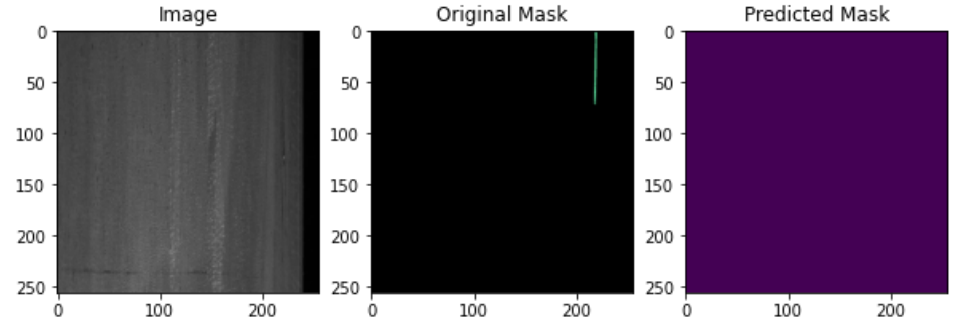

```
1/1 [==============================] - 0s 23ms/step
```




```
1/1 [==============================] - 0s 22ms/step
```




```
1/1 [==============================] - 0s 23ms/step
```




```
1/1 [==============================] - 0s 26ms/step
```

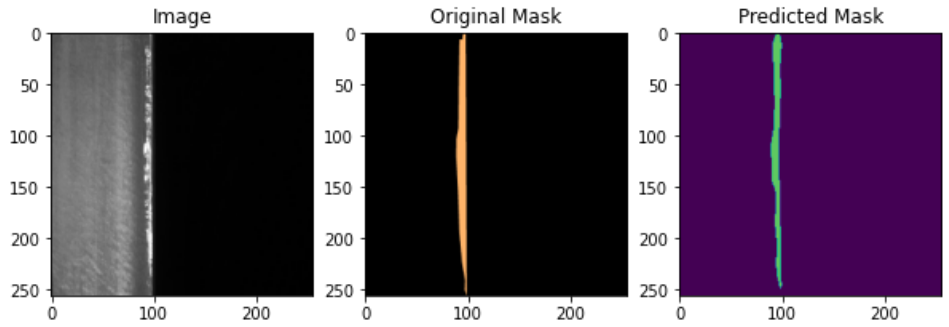


Image
Original Mask
Predicted Mask
0
50
100
150
200
250
0
100
200

In [ ]:

In [ ]:
```
history = model.fit(train_dataloader, steps_per_epoch=(len(train_dataloader))//BATCH_SIZE, epochs=30,\
                    validation_data=test_dataloader, callbacks=callbacks)
```

```
Epoch 1/30
49/49 [==============================] - ETA: 0s - loss: 0.8418 - iou_score: 0.2096
Epoch 1: val_iou_score improved from -inf to 0.06799, saving model to ./best_model.h5
49/49 [==============================] - 200s 4s/step - loss: 0.8418 - iou_score: 0.2096 - val_loss: 1.6162 - val_iou_score: 0.0680 - lr: 0.0010
Epoch 2/30
49/49 [==============================] - ETA: 0s - loss: 0.6954 - iou_score: 0.3042
Epoch 2: val_iou_score improved from 0.06799 to 0.15711, saving model to ./best_model.h5
49/49 [==============================] - 168s 3s/step - loss: 0.6954 - iou_score: 0.3042 - val_loss: 1.0744 - val_iou_score: 0.1571 - lr: 0.0010
Epoch 3/30
49/49 [==============================] - ETA: 0s - loss: 0.6541 - iou_score: 0.3348
Epoch 3: val_iou_score improved from 0.15711 to 0.24199, saving model to ./best_model.h5
49/49 [==============================] - 167s 3s/step - loss: 0.6541 - iou_score: 0.3348 - val_loss: 0.8804 - val_iou_score: 0.2420 - lr: 0.0010
Epoch 4/30
49/49 [==============================] - ETA: 0s - loss: 0.6709 - iou_score: 0.3277
Epoch 4: val_iou_score improved from 0.24199 to 0.25260, saving model to ./best_model.h5
49/49 [==============================] - 169s 3s/step - loss: 0.6709 - iou_score: 0.3277 - val_loss: 0.8154 - val_iou_score: 0.2526 - lr: 0.0010
Epoch 5/30
49/49 [==============================] - ETA: 0s - loss: 0.6581 - iou_score: 0.3366
Epoch 5: val_iou_score improved from 0.25260 to 0.28638, saving model to ./best_model.h5
49/49 [==============================] - 167s 3s/step - loss: 0.6581 - iou_score: 0.3366 - val_loss: 0.7416 - val_iou_score: 0.2864 - lr: 0.0010
Epoch 6/30
49/49 [==============================] - ETA: 0s - loss: 0.6376 - iou_score: 0.3523
Epoch 6: val_iou_score improved from 0.28638 to 0.31449, saving model to ./best_model.h5
49/49 [==============================] - 167s 3s/step - loss: 0.6376 - iou_score: 0.3523 - val_loss: 0.6913 - val_iou_score: 0.3145 - lr: 0.0010
Epoch 7/30
49/49 [==============================] - ETA: 0s - loss: 0.6572 - iou_score: 0.3394
Epoch 7: val_iou_score did not improve from 0.31449
49/49 [==============================] - 166s 3s/step - loss: 0.6572 - iou_score: 0.3394 - val_loss: 0.6987 - val_iou_score: 0.3114 - lr: 0.0010
Epoch 8/30
49/49 [==============================] - ETA: 0s - loss: 0.6388 - iou_score: 0.3543
Epoch 8: val_iou_score improved from 0.31449 to 0.31566, saving model to ./best_model.h5
49/49 [==============================] - 168s 3s/step - loss: 0.6388 - iou_score: 0.3543 - val_loss: 0.6844 - val_iou_score: 0.3157 - lr: 0.0010
Epoch 9/30
49/49 [==============================] - ETA: 0s - loss: 0.6396 - iou_score: 0.3548
Epoch 9: val_iou_score did not improve from 0.31566
49/49 [==============================] - 168s 3s/step - loss: 0.6396 - iou_score: 0.3548 - val_loss: 0.6993 - val_iou_score: 0.3098 - lr: 0.0010
Epoch 10/30
49/49 [==============================] - ETA: 0s - loss: 0.6250 - iou_score: 0.3658
Epoch 10: val_iou_score improved from 0.31566 to 0.32982, saving model to ./best_model.h5
49/49 [==============================] - 168s 3s/step - loss: 0.6250 - iou_score: 0.3658 - val_loss: 0.6755 - val_iou_score: 0.3298 - lr: 0.0010
Epoch 11/30
49/49 [==============================] - ETA: 0s - loss: 0.6295 - iou_score: 0.3552
Epoch 11: val_iou_score did not improve from 0.32982
49/49 [==============================] - 167s 3s/step - loss: 0.6295 - iou_score: 0.3552 - val_loss: 0.7084 - val_iou_score: 0.3092 - lr: 0.0010
Epoch 12/30
49/49 [==============================] - ETA: 0s - loss: 0.6051 - iou_score: 0.3800
Epoch 12: val_iou_score improved from 0.32982 to 0.33747, saving model to ./best_model.h5
49/49 [==============================] - 167s 3s/step - loss: 0.6051 - iou_score: 0.3800 - val_loss: 0.6713 - val_iou_score: 0.3375 - lr: 0.0010
Epoch 13/30
49/49 [==============================] - ETA: 0s - loss: 0.6116 - iou_score: 0.3729
Epoch 13: val_iou_score did not improve from 0.33747
49/49 [==============================] - 169s 3s/step - loss: 0.6116 - iou_score: 0.3729 - val_loss: 0.7057 - val_iou_score: 0.3226 - lr: 0.0010
```

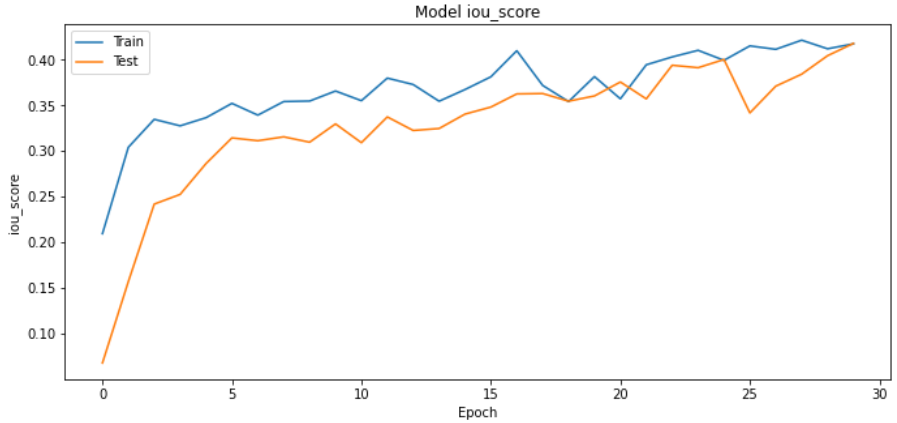
```
Epoch 14/30
49/49 [==============================] - ETA: 0s - loss: 0.6385 - iou_score: 0.3547
Epoch 14: val_iou_score did not improve from 0.33747
49/49 [==============================] - 168s 3s/step - loss: 0.6385 - iou_score: 0.3547 - val_loss: 0.6786 - val_iou_score: 0.3248 - lr: 0.0010
Epoch 15/30
49/49 [==============================] - ETA: 0s - loss: 0.6178 - iou_score: 0.3675
Epoch 15: val_iou_score improved from 0.33747 to 0.34055, saving model to ./best_model.h5
49/49 [==============================] - 168s 3s/step - loss: 0.6178 - iou_score: 0.3675 - val_loss: 0.6565 - val_iou_score: 0.3405 - lr: 0.0010
Epoch 16/30
49/49 [==============================] - ETA: 0s - loss: 0.6002 - iou_score: 0.3814
Epoch 16: val_iou_score improved from 0.34055 to 0.34827, saving model to ./best_model.h5
49/49 [==============================] - 169s 3s/step - loss: 0.6002 - iou_score: 0.3814 - val_loss: 0.6472 - val_iou_score: 0.3483 - lr: 0.0010
Epoch 17/30
49/49 [==============================] - ETA: 0s - loss: 0.5661 - iou_score: 0.4100
Epoch 17: val_iou_score improved from 0.34827 to 0.36267, saving model to ./best_model.h5
49/49 [==============================] - 212s 4s/step - loss: 0.5661 - iou_score: 0.4100 - val_loss: 0.6400 - val_iou_score: 0.3627 - lr: 0.0010
Epoch 18/30
49/49 [==============================] - ETA: 0s - loss: 0.6155 - iou_score: 0.3719
Epoch 18: val_iou_score improved from 0.36267 to 0.36308, saving model to ./best_model.h5
49/49 [==============================] - 169s 3s/step - loss: 0.6155 - iou_score: 0.3719 - val_loss: 0.6376 - val_iou_score: 0.3631 - lr: 0.0010
Epoch 19/30
49/49 [==============================] - ETA: 0s - loss: 0.6388 - iou_score: 0.3543
Epoch 19: val_iou_score did not improve from 0.36308
49/49 [==============================] - 167s 3s/step - loss: 0.6388 - iou_score: 0.3543 - val_loss: 0.6342 - val_iou_score: 0.3547 - lr: 0.0010
Epoch 20/30
49/49 [==============================] - ETA: 0s - loss: 0.6083 - iou_score: 0.3816
Epoch 20: val_iou_score did not improve from 0.36308
49/49 [==============================] - 165s 3s/step - loss: 0.6083 - iou_score: 0.3816 - val_loss: 0.6316 - val_iou_score: 0.3604 - lr: 0.0010
Epoch 21/30
49/49 [==============================] - ETA: 0s - loss: 0.6360 - iou_score: 0.3573
Epoch 21: val_iou_score improved from 0.36308 to 0.37573, saving model to ./best_model.h5
49/49 [==============================] - 167s 3s/step - loss: 0.6360 - iou_score: 0.3573 - val_loss: 0.6181 - val_iou_score: 0.3757 - lr: 0.0010
Epoch 22/30
49/49 [==============================] - ETA: 0s - loss: 0.5870 - iou_score: 0.3946
Epoch 22: val_iou_score did not improve from 0.37573
49/49 [==============================] - 169s 3s/step - loss: 0.5870 - iou_score: 0.3946 - val_loss: 0.6314 - val_iou_score: 0.3572 - lr: 0.0010
Epoch 23/30
49/49 [==============================] - ETA: 0s - loss: 0.5830 - iou_score: 0.4032
Epoch 23: val_iou_score improved from 0.37573 to 0.39407, saving model to ./best_model.h5
49/49 [==============================] - 169s 3s/step - loss: 0.5830 - iou_score: 0.4032 - val_loss: 0.5851 - val_iou_score: 0.3941 - lr: 0.0010
Epoch 24/30
49/49 [==============================] - ETA: 0s - loss: 0.5658 - iou_score: 0.4104
Epoch 24: val_iou_score did not improve from 0.39407
49/49 [==============================] - 169s 3s/step - loss: 0.5658 - iou_score: 0.4104 - val_loss: 0.5943 - val_iou_score: 0.3914 - lr: 0.0010
Epoch 25/30
49/49 [==============================] - ETA: 0s - loss: 0.5701 - iou_score: 0.3995
Epoch 25: val_iou_score improved from 0.39407 to 0.40036, saving model to ./best_model.h5
49/49 [==============================] - 169s 3s/step - loss: 0.5701 - iou_score: 0.3995 - val_loss: 0.5826 - val_iou_score: 0.4004 - lr: 0.0010
Epoch 26/30
49/49 [==============================] - ETA: 0s - loss: 0.5605 - iou_score: 0.4153
Epoch 26: val_iou_score did not improve from 0.40036
49/49 [==============================] - 166s 3s/step - loss: 0.5605 - iou_score: 0.4153 - val_loss: 0.6502 - val_iou_score: 0.3419 - lr: 0.0010
```

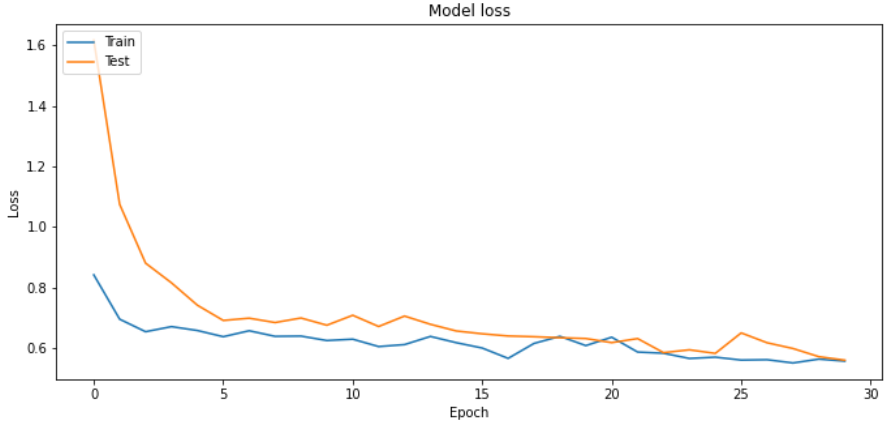
```
Epoch 27/30
49/49 [==============================] - ETA: 0s - loss: 0.5617 - iou_score: 0.4115
Epoch 27: val_iou_score did not improve from 0.40036
49/49 [==============================] - 168s 3s/step - loss: 0.5617 - iou_score: 0.4115 - val_loss: 0.6176 - val_iou_score: 0.3711 - lr: 0.0010
Epoch 28/30
49/49 [==============================] - ETA: 0s - loss: 0.5510 - iou_score: 0.4215
Epoch 28: val_iou_score did not improve from 0.40036

Epoch 28: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
49/49 [==============================] - 169s 3s/step - loss: 0.5510 - iou_score: 0.4215 - val_loss: 0.5987 - val_iou_score: 0.3843 - lr: 0.0010
Epoch 29/30
49/49 [==============================] - ETA: 0s - loss: 0.5637 - iou_score: 0.4121
Epoch 29: val_iou_score improved from 0.40036 to 0.40452, saving model to ./best_model.h5
49/49 [==============================] - 169s 3s/step - loss: 0.5637 - iou_score: 0.4121 - val_loss: 0.5717 - val_iou_score: 0.4045 - lr: 2.0000e-04
Epoch 30/30
49/49 [==============================] - ETA: 0s - loss: 0.5570 - iou_score: 0.4175
Epoch 30: val_iou_score improved from 0.40452 to 0.41797, saving model to ./best_model.h5
49/49 [==============================] - 170s 4s/step - loss: 0.5570 - iou_score: 0.4175 - val_loss: 0.5602 - val_iou_score: 0.4180 - lr: 2.0000e-04
```

In [ ]:
```
%matplotlib inline
import matplotlib.pyplot as plt
```

Visualizing result of Unet

In [ ]:
```
# Plot training & validation iou_score values
plt.figure(figsize=(25, 5))
plt.subplot(1, 2, 1)
plt.plot(history.history['iou_score'])
plt.plot(history.history['val_iou_score'])
plt.title('Model iou_score')
plt.ylabel('iou_score')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')

# Plot training & validation loss values
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```





In [ ]:
```
images_ = validation['image_path'].values
masks_ = validation["mask_path"].values
lst = np.arange(len(images_))
len(lst)
np.random.choice(lst, size = 20, replace = False)
```

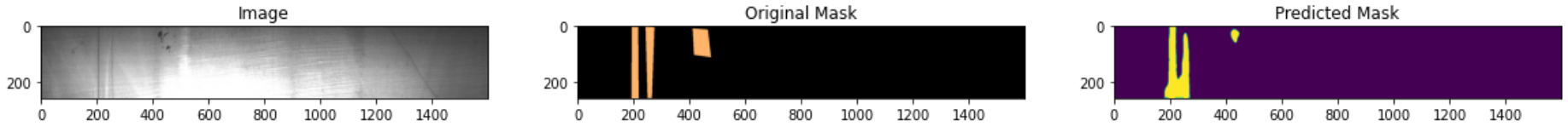
Out[74]:
```
639
```

In [ ]:
```
np.random.choice(lst, size = 20, replace = False)
```

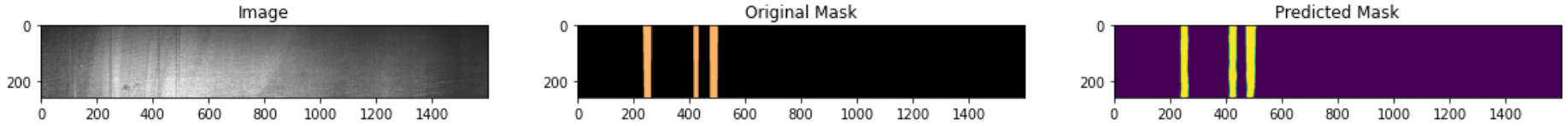
Out[75]:
```
array([145,   6, 196, 365, 104, 405, 185, 336,  48, 366, 211,  78, 485,
       175, 245, 430, 557, 607, 530, 421])
```

In [ ]:
```
# Visualizing the predicted mask for test data.
import random
ids = np.random.choice(lst, size = 20, replace = False)
for i in ids:
  image = cv2.imread(images_[i], cv2.IMREAD_UNCHANGED)
  # image = cv2.resize(image, (256, 256),  interpolation = cv2.INTER_AREA)
  image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
  image = np.expand_dims(image, axis=0)

  mask  = cv2.imread(masks_[i], cv2.IMREAD_UNCHANGED)
  # mask = cv2.resize(mask, (256, 256),  interpolation = cv2.INTER_AREA)

  pred = model.predict(image,verbose=1)
  pred = tf.argmax(pred, axis=-1)

  fig = plt.figure(figsize=(20,14))

  ax1 = fig.add_subplot(1, 3, 1)
  ax1.imshow(image[0,:,:])
  ax2=fig.add_subplot(1, 3, 2)
  ax2.imshow(mask)
  ax3=fig.add_subplot(1, 3, 3)
  ax3.imshow(pred[0,:,:])
  ax1.title.set_text('Image')
  ax2.title.set_text('Original Mask')
  ax3.title.set_text('Predicted Mask')
  plt.show()
```

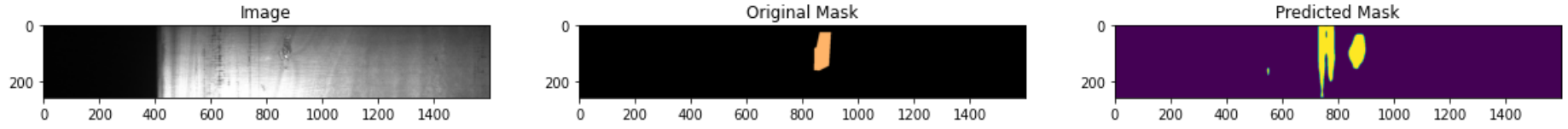
```
1/1 [==============================] - 0s 25ms/step
```



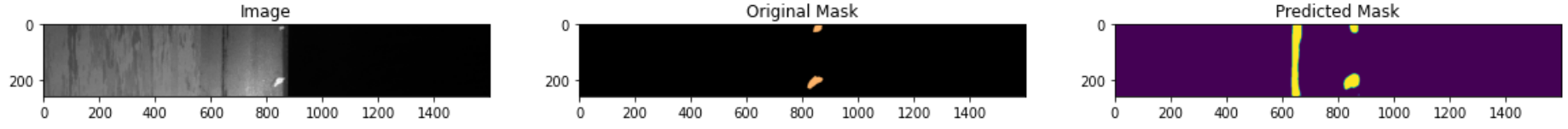

```
1/1 [==============================] - 0s 23ms/step
```



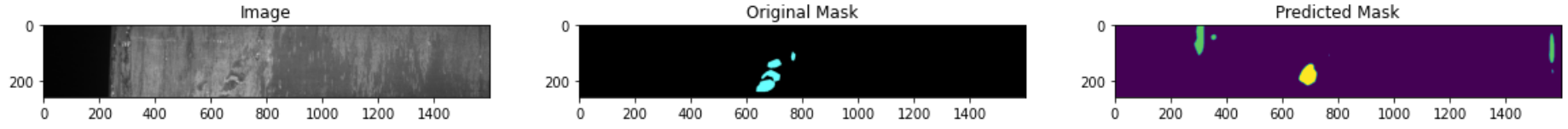

```
1/1 [==============================] - 0s 25ms/step
```



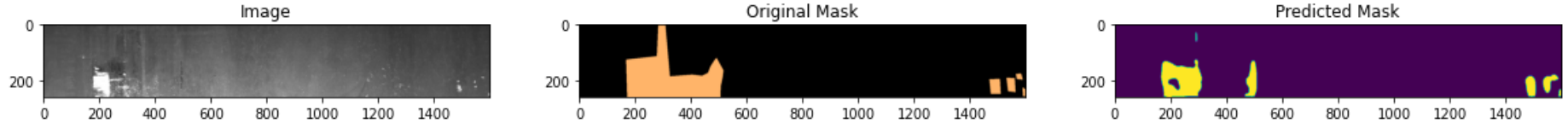

```
1/1 [==============================] - 0s 27ms/step
```



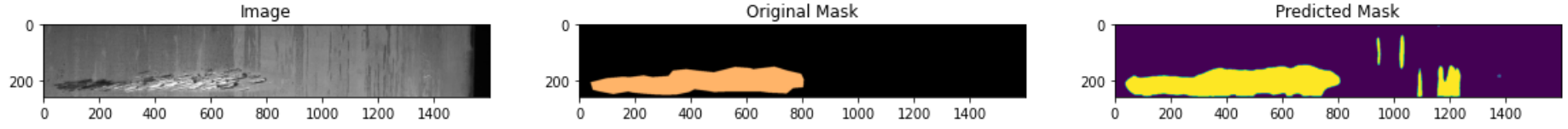

```
1/1 [==============================] - 0s 26ms/step
```



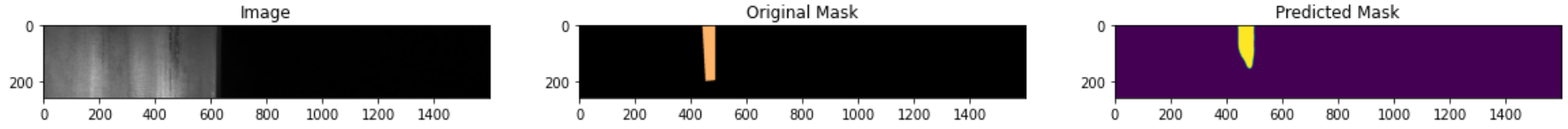

```
1/1 [==============================] - 0s 24ms/step
```



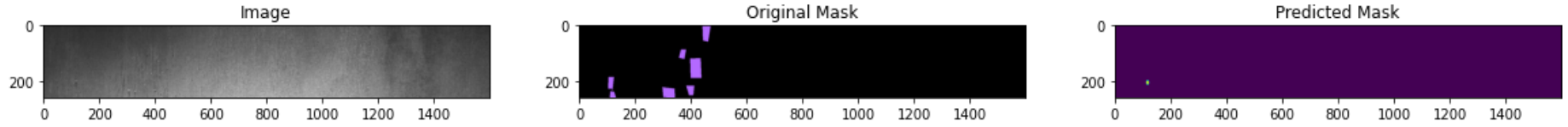

```
1/1 [==============================] - 0s 27ms/step
```



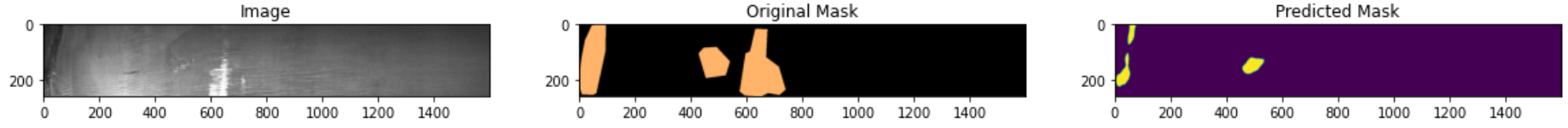

```
1/1 [==============================] - 0s 24ms/step
```



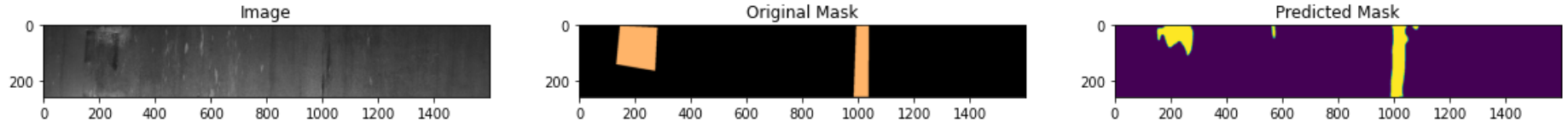

```
1/1 [==============================] - 0s 23ms/step
```



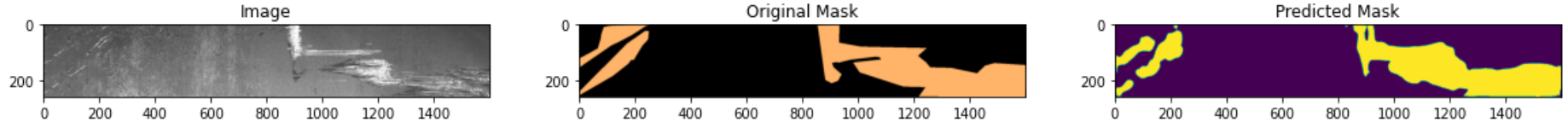

```
1/1 [==============================] - 0s 31ms/step
```



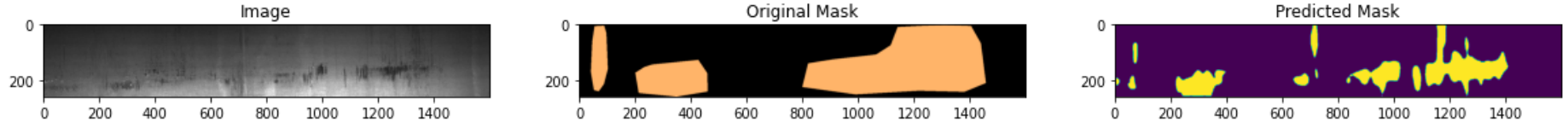

```
1/1 [==============================] - 0s 31ms/step
```



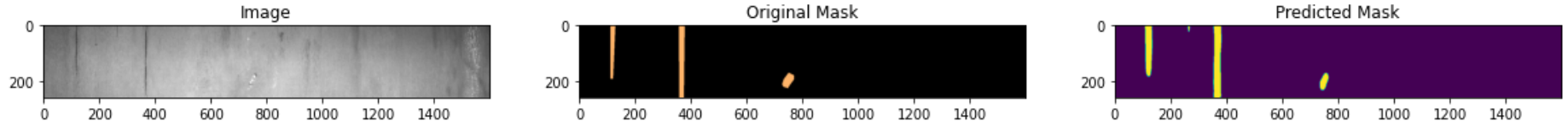

```
1/1 [==============================] - 0s 23ms/step
```



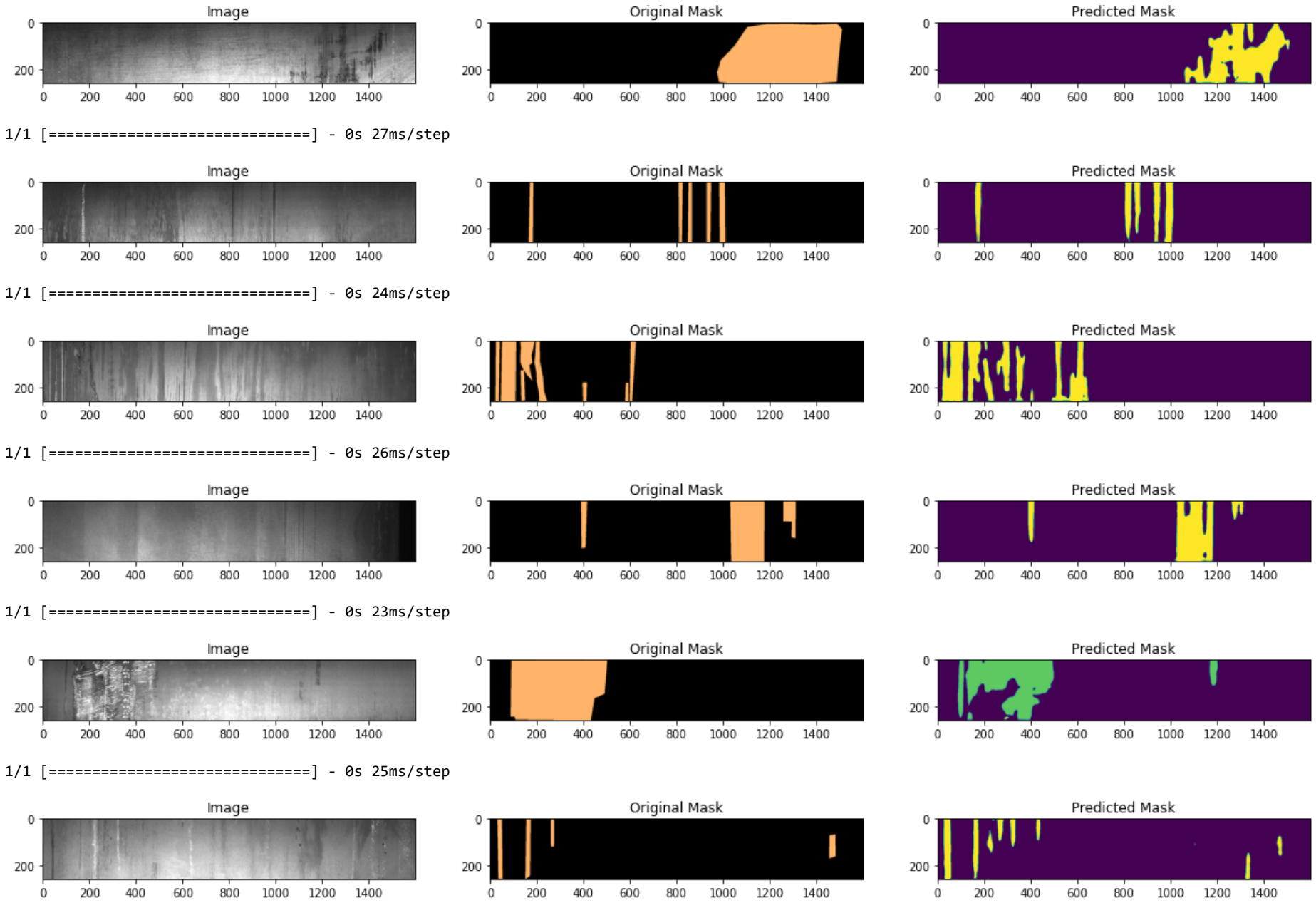

```
1/1 [==============================] - 0s 25ms/step
```




```
1/1 [==============================] - 0s 24ms/step
```




```
1/1 [==============================] - 0s 25ms/step
```




```
1/1 [==============================] - 0s 27ms/step
```



```
1/1 [==============================] - 0s 24ms/step
```



```
1/1 [==============================] - 0s 26ms/step
```



```
1/1 [==============================] - 0s 23ms/step
```



```
1/1 [==============================] - 0s 25ms/step
```

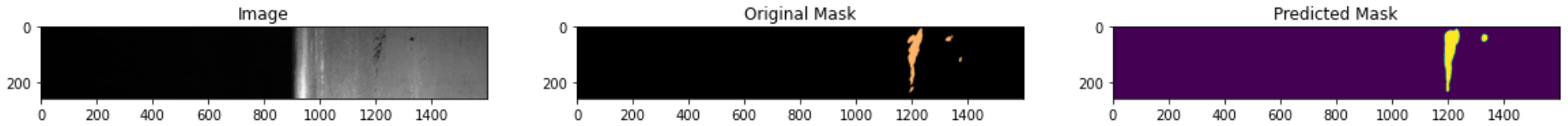
In [ ]:
```
history = model.fit(train_dataloader, steps_per_epoch=(len(train_dataloader))//BATCH_SIZE, epochs=20,\
                    validation_data=test_dataloader, callbacks=callbacks)
```

```
Epoch 1/20
49/49 [==============================] - ETA: 0s - loss: 0.5503 - iou_score: 0.4203
Epoch 1: val_iou_score improved from 0.41797 to 0.42461, saving model to ./best_model.h5
49/49 [==============================] - 171s 4s/step - loss: 0.5503 - iou_score: 0.4203 - val_loss: 0.5477 - val_iou_score: 0.4246 - lr: 2.0000e-04
Epoch 2/20
49/49 [==============================] - ETA: 0s - loss: 0.5373 - iou_score: 0.4306
Epoch 2: val_iou_score improved from 0.42461 to 0.42679, saving model to ./best_model.h5
49/49 [==============================] - 170s 3s/step - loss: 0.5373 - iou_score: 0.4306 - val_loss: 0.5438 - val_iou_score: 0.4268 - lr: 2.0000e-04
Epoch 3/20
49/49 [==============================] - ETA: 0s - loss: 0.5467 - iou_score: 0.4232
Epoch 3: val_iou_score did not improve from 0.42679
49/49 [==============================] - 169s 3s/step - loss: 0.5467 - iou_score: 0.4232 - val_loss: 0.5491 - val_iou_score: 0.4216 - lr: 2.0000e-04
Epoch 4/20
49/49 [==============================] - ETA: 0s - loss: 0.5531 - iou_score: 0.4227
Epoch 4: val_iou_score improved from 0.42679 to 0.43502, saving model to ./best_model.h5
49/49 [==============================] - 168s 3s/step - loss: 0.5531 - iou_score: 0.4227 - val_loss: 0.5347 - val_iou_score: 0.4350 - lr: 2.0000e-04
Epoch 5/20
49/49 [==============================] - ETA: 0s - loss: 0.5262 - iou_score: 0.4423
Epoch 5: val_iou_score did not improve from 0.43502
```

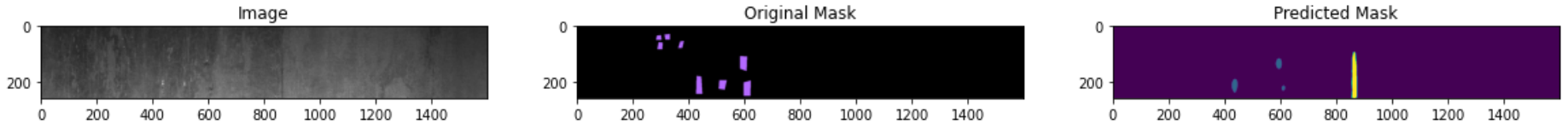
Visualizing the result of Unet

In [ ]:
```
# Visualizing the predicted mask for test data.
import random
ids = np.random.choice(lst, size = 20, replace = False)
for i in ids:
  image = cv2.imread(images_[i], cv2.IMREAD_UNCHANGED)
  # image = cv2.resize(image, (256, 256),  interpolation = cv2.INTER_AREA)
  image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
  image = np.expand_dims(image, axis=0)

  mask  = cv2.imread(masks_[i], cv2.IMREAD_UNCHANGED)
  # mask = cv2.resize(mask, (256, 256),  interpolation = cv2.INTER_AREA)

  pred = model.predict(image,verbose=1)
  pred = tf.argmax(pred, axis=-1)

  fig = plt.figure(figsize=(20,14))

  ax1 = fig.add_subplot(1, 3, 1)
  ax1.imshow(image[0,:,:])
  ax2=fig.add_subplot(1, 3, 2)
  ax2.imshow(mask)
  ax3=fig.add_subplot(1, 3, 3)
  ax3.imshow(pred[0,:,:])
  ax1.title.set_text('Image')
  ax2.title.set_text('Original Mask')
  ax3.title.set_text('Predicted Mask')
  plt.show()
```

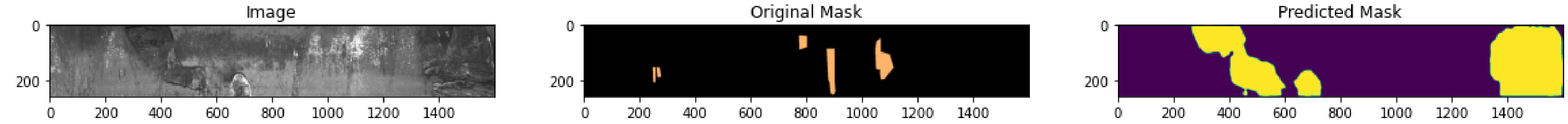
```
1/1 [==============================] - 0s 25ms/step
```



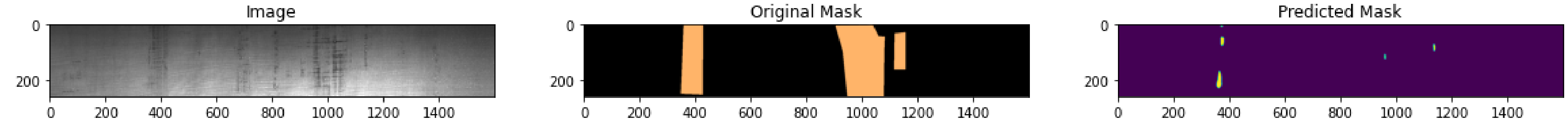

```
1/1 [==============================] - 0s 23ms/step
```



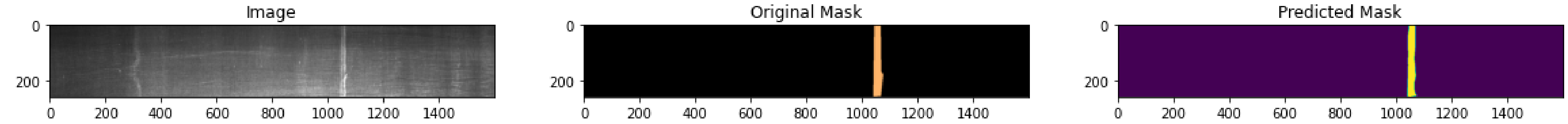

```
1/1 [==============================] - 0s 25ms/step
```




```
1/1 [==============================] - 0s 24ms/step
```

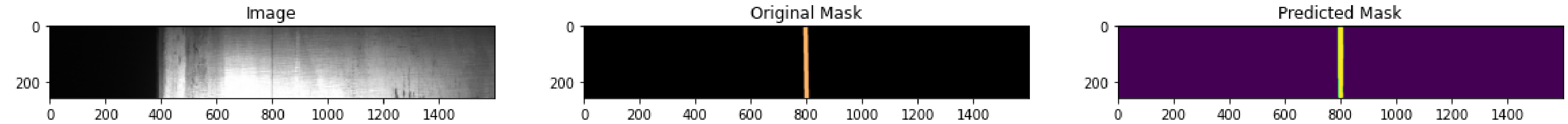



```
1/1 [==============================] - 0s 25ms/step
```

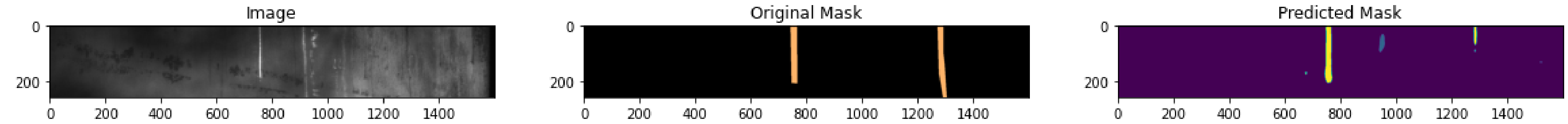



```
1/1 [==============================] - 0s 23ms/step
```

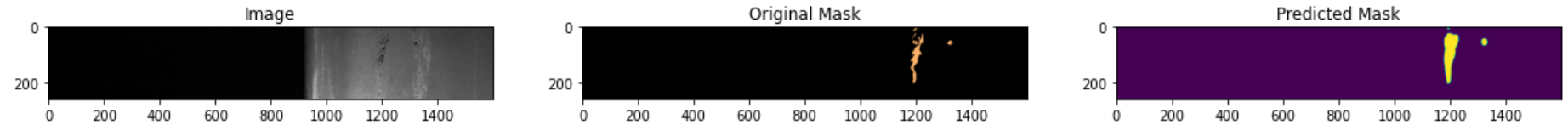

```
1/1 [==============================] - 0s 23ms/step
```



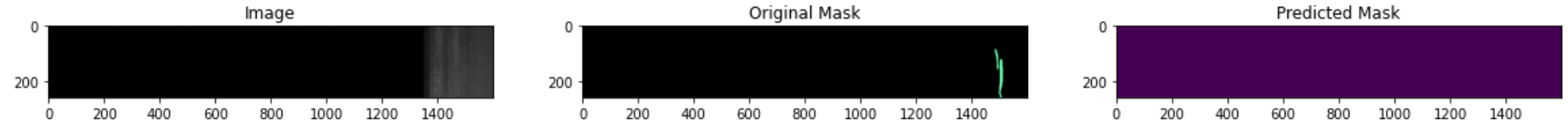

```
1/1 [==============================] - 0s 23ms/step
```



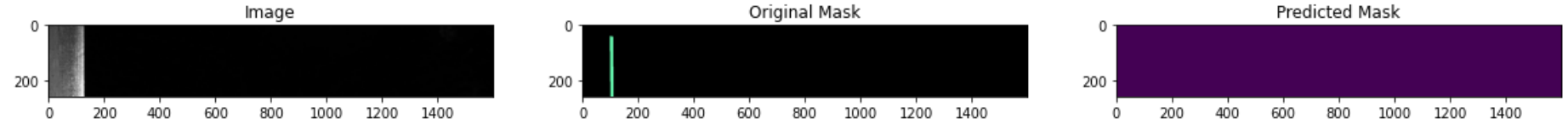

```
1/1 [==============================] - 0s 23ms/step
```



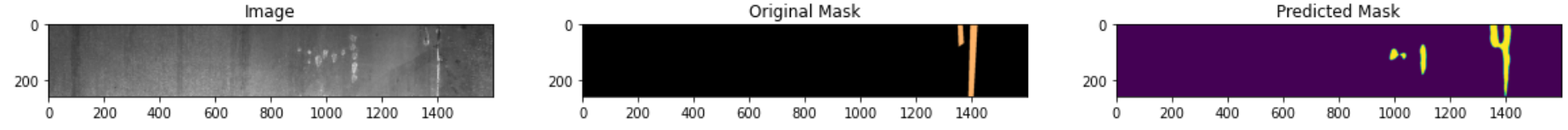

```
1/1 [==============================] - 0s 25ms/step
```



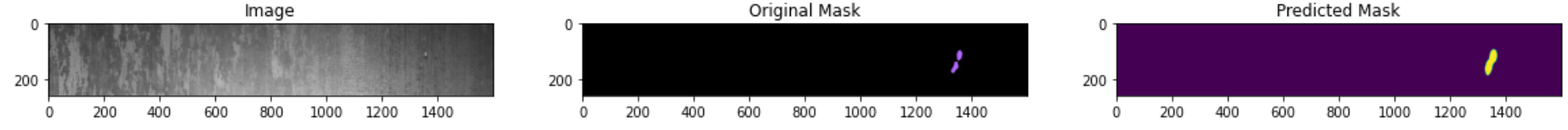

```
1/1 [==============================] - 0s 30ms/step
```



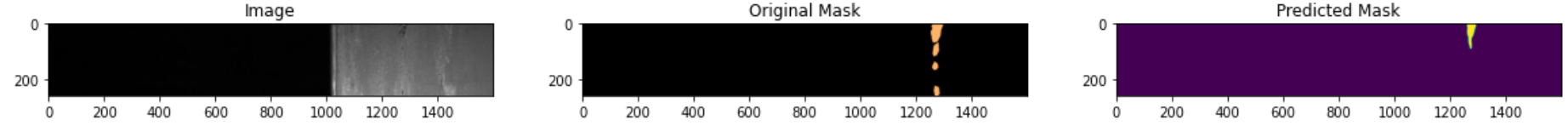

```
1/1 [==============================] - 0s 24ms/step
```



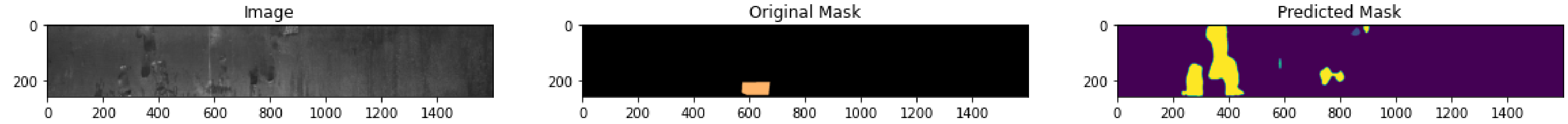

```
1/1 [==============================] - 0s 24ms/step
```



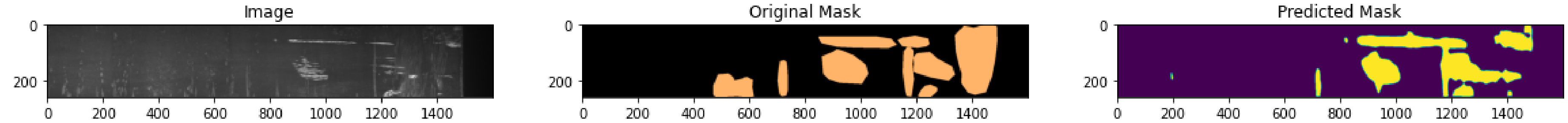

```
1/1 [==============================] - 0s 24ms/step
```



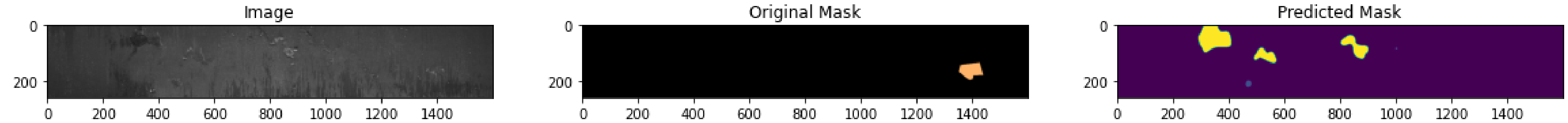

```
1/1 [==============================] - 0s 26ms/step
```



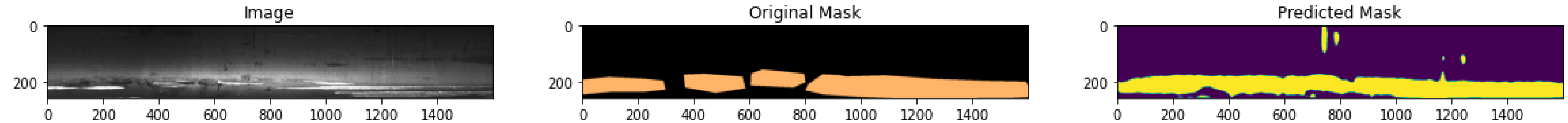

```
1/1 [==============================] - 0s 26ms/step
```




```
1/1 [==============================] - 0s 23ms/step
```

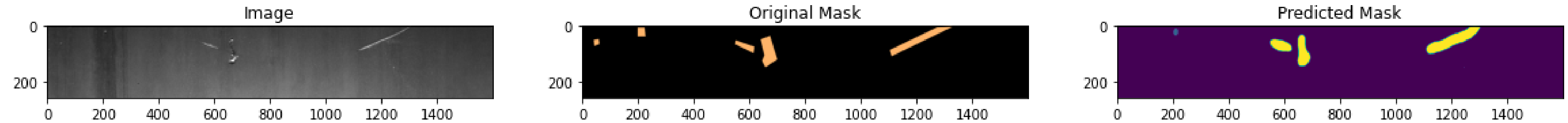



```
1/1 [==============================] - 0s 23ms/step
```




```
1/1 [==============================] - 0s 23ms/step
```


```
1/1 [==============================] - 0s 23ms/step
```



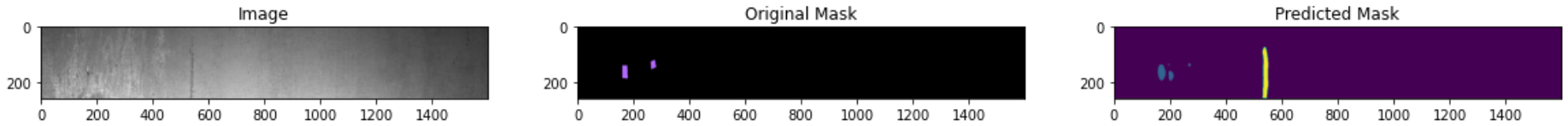
In [ ]:
```
history = model.fit(train_dataloader, steps_per_epoch=(len(train_dataloader))//BATCH_SIZE, epochs=20,\
                    validation_data=test_dataloader, callbacks=callbacks)
```

```
Epoch 1/20
49/49 [==============================] - ETA: 0s - loss: 0.5298 - iou_score: 0.4455
Epoch 1: val_iou_score did not improve from 0.44580
49/49 [==============================] - 212s 4s/step - loss: 0.5298 - iou_score: 0.4455 - val_loss: 0.5265 - val_iou_score: 0.4425 - lr: 1.0000e-04
Epoch 2/20
49/49 [==============================] - ETA: 0s - loss: 0.5190 - iou_score: 0.4489
Epoch 2: val_iou_score did not improve from 0.44580
49/49 [==============================] - 167s 3s/step - loss: 0.5190 - iou_score: 0.4489 - val_loss: 0.5284 - val_iou_score: 0.4393 - lr: 1.0000e-04
Epoch 3/20
49/49 [==============================] - ETA: 0s - loss: 0.5291 - iou_score: 0.4344
Epoch 3: val_iou_score did not improve from 0.44580
49/49 [==============================] - 167s 3s/step - loss: 0.5291 - iou_score: 0.4344 - val_loss: 0.5244 - val_iou_score: 0.4428 - lr: 1.0000e-04
Epoch 4/20
49/49 [==============================] - ETA: 0s - loss: 0.5149 - iou_score: 0.4511
Epoch 4: val_iou_score did not improve from 0.44580
49/49 [==============================] - 169s 3s/step - loss: 0.5149 - iou_score: 0.4511 - val_loss: 0.5306 - val_iou_score: 0.4367 - lr: 1.0000e-04
Epoch 5/20
49/49 [==============================] - ETA: 0s - loss: 0.5294 - iou_score: 0.4433
Epoch 5: val_iou_score did not improve from 0.44580
```

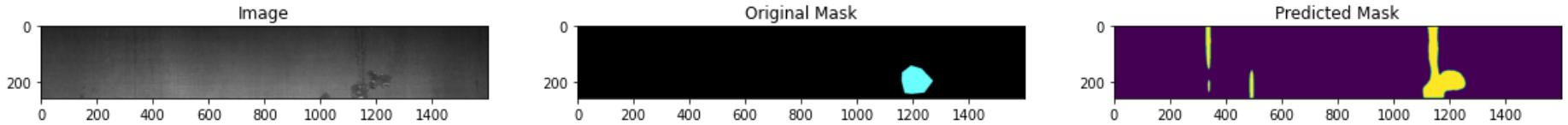
Visualizing the result of Unet

In [ ]:
```
# Visualizing the predicted mask for test data.
import random
ids = np.random.choice(lst, size = 20, replace = False)
for i in ids:
  image = cv2.imread(images_[i], cv2.IMREAD_UNCHANGED)
  # image = cv2.resize(image, (256, 256),  interpolation = cv2.INTER_AREA)
  image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
  image = np.expand_dims(image, axis=0)

  mask  = cv2.imread(masks_[i], cv2.IMREAD_UNCHANGED)
  # mask = cv2.resize(mask, (256, 256),  interpolation = cv2.INTER_AREA)

  pred = model.predict(image,verbose=1)
  pred = tf.argmax(pred, axis=-1)

  fig = plt.figure(figsize=(20,14))

  ax1 = fig.add_subplot(1, 3, 1)
  ax1.imshow(image[0,:,:])
  ax2=fig.add_subplot(1, 3, 2)
  ax2.imshow(mask)
  ax3=fig.add_subplot(1, 3, 3)
  ax3.imshow(pred[0,:,:])
  ax1.title.set_text('Image')
  ax2.title.set_text('Original Mask')
  ax3.title.set_text('Predicted Mask')
  plt.show()
```

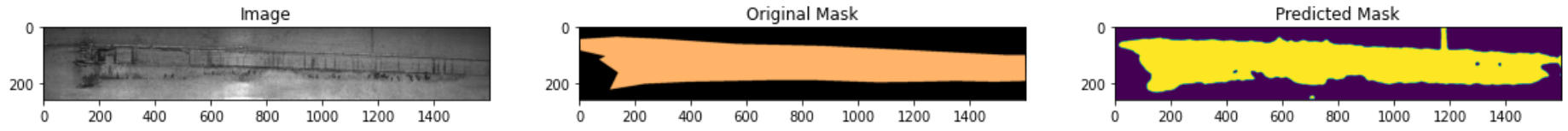
```
1/1 [==============================] - 0s 208ms/step
```



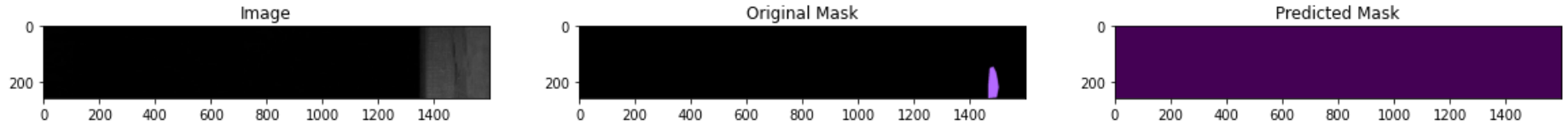

```
1/1 [==============================] - 0s 23ms/step
```



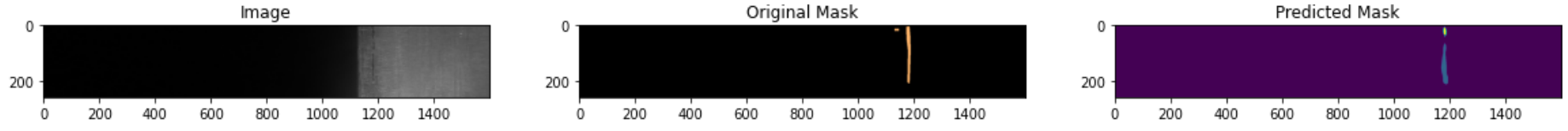

```
1/1 [==============================] - 0s 28ms/step
```



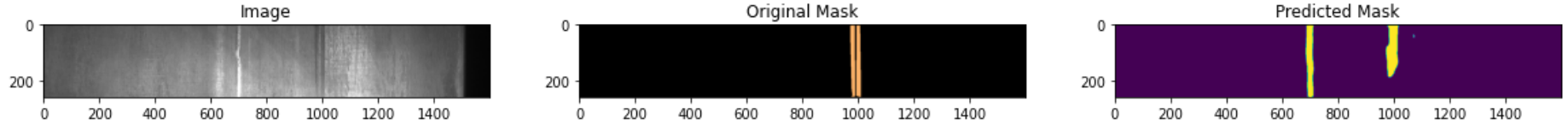

```
1/1 [==============================] - 0s 24ms/step
```



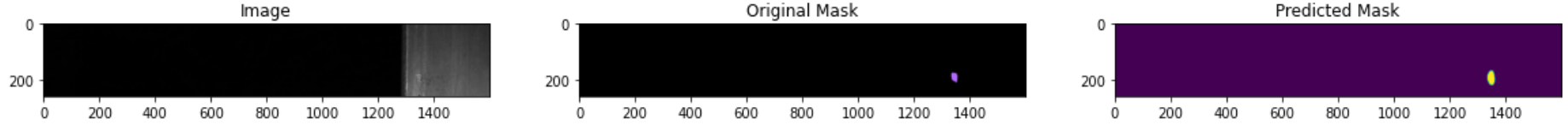

```
1/1 [==============================] - 0s 23ms/step
```



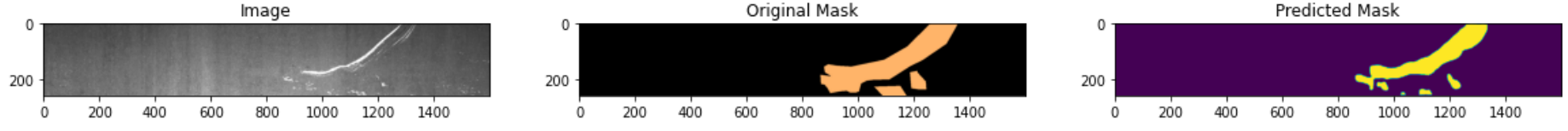

```
1/1 [==============================] - 0s 23ms/step
```



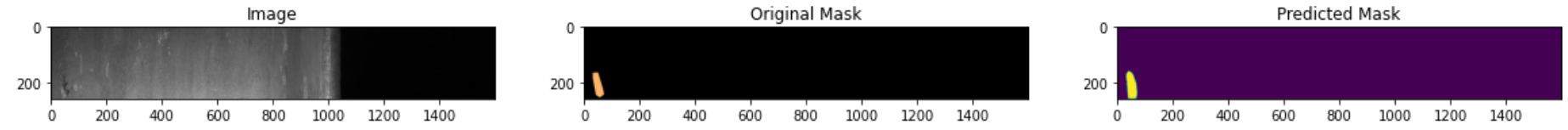

```
1/1 [==============================] - 0s 32ms/step
```



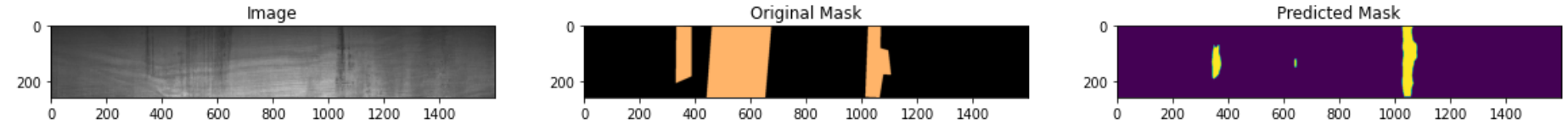

```
1/1 [==============================] - 0s 29ms/step
```



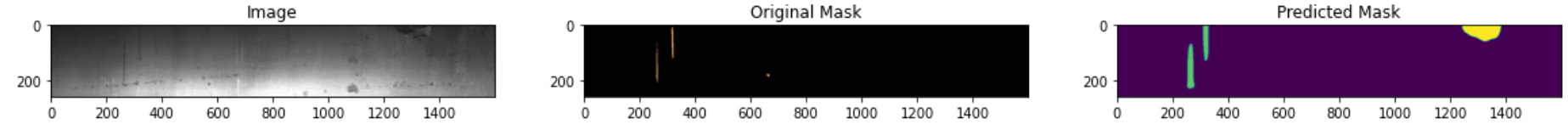

```
1/1 [==============================] - 0s 23ms/step
```



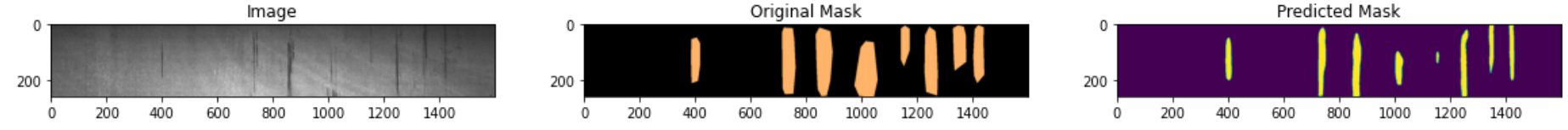

```
1/1 [==============================] - 0s 24ms/step
```



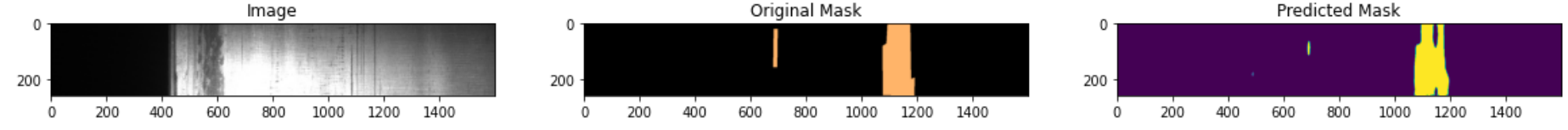

```
1/1 [==============================] - 0s 22ms/step
```



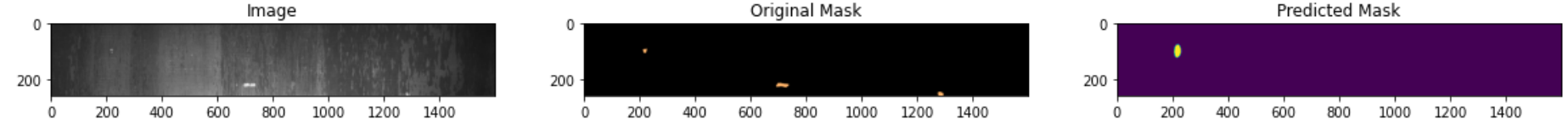

```
1/1 [==============================] - 0s 24ms/step
```



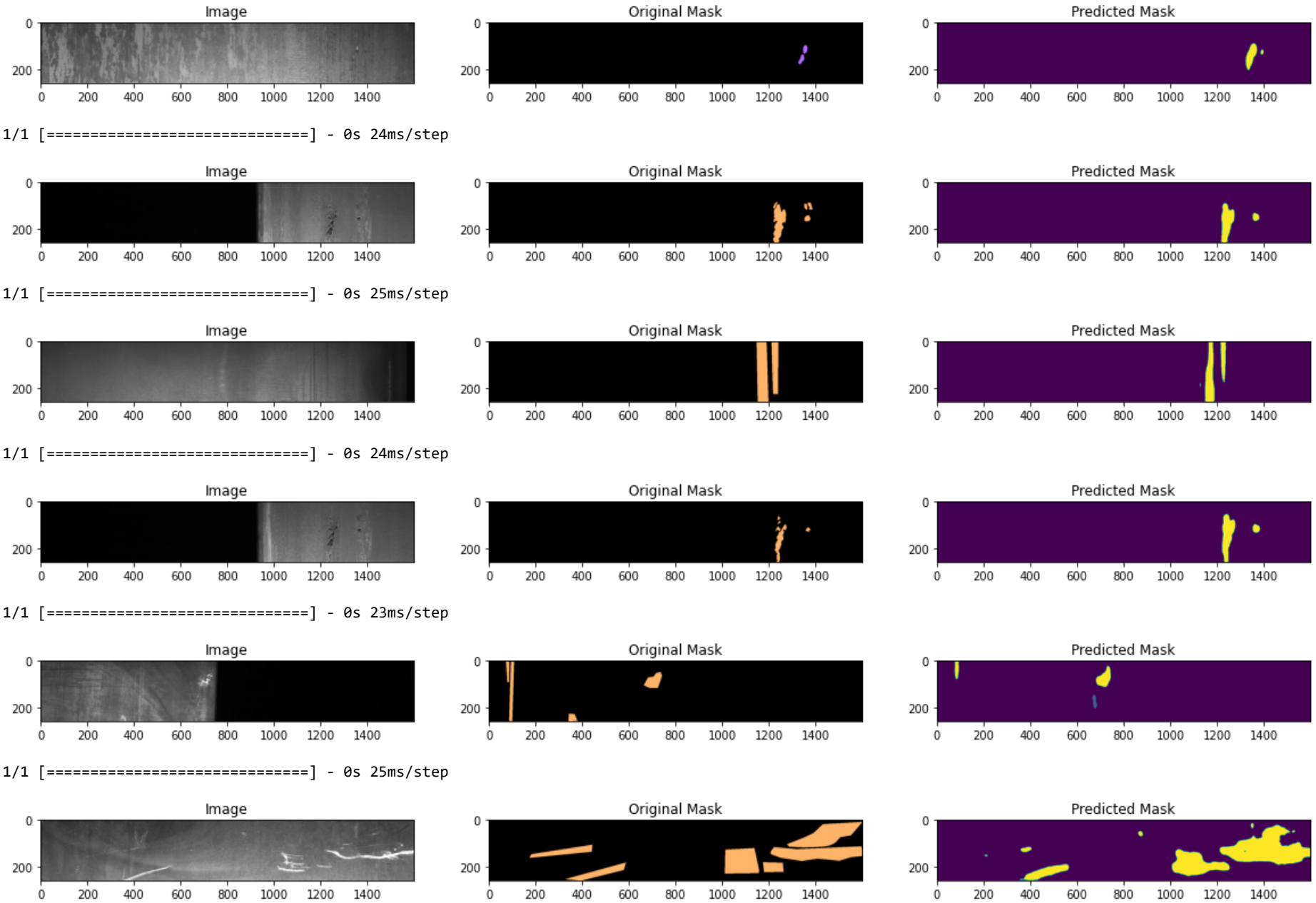

```
1/1 [==============================] - 0s 23ms/step
```




```
1/1 [==============================] - 0s 26ms/step
```




```
1/1 [==============================] - 0s 24ms/step
```



4.5 Applying VGG16

In [ ]:
```
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Conv2D, BatchNormalization, Activation, MaxPool2D, Conv2DTranspose, Concatenate, Input
from tensorflow.keras.models import Model
from tensorflow.keras.layers import GlobalAveragePooling2D, Dense, Conv2D, BatchNormalization, Dropout, Input, MaxPool2D , Flatten


tf.keras.backend.clear_session()
import segmentation_models as sm
from segmentation_models import Unet
from segmentation_models.metrics import iou_score
sm.set_framework('tf.keras')
```

```
Segmentation Models: using `keras` framework.
```

In [ ]:
```
from keras.applications.vgg16 import VGG16
from keras.applications.vgg16 import preprocess_input
model = VGG16(input_shape=[256, 1600, 3], weights='imagenet', include_top=False)
model.summary()
```

```
Model: "vgg16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 256, 1600, 3)]    0

 block1_conv1 (Conv2D)       (None, 256, 1600, 64)     1792

 block1_conv2 (Conv2D)       (None, 256, 1600, 64)     36928

 block1_pool (MaxPooling2D)  (None, 128, 800, 64)      0

 block2_conv1 (Conv2D)       (None, 128, 800, 128)     73856

 block2_conv2 (Conv2D)       (None, 128, 800, 128)     147584

 block2_pool (MaxPooling2D)  (None, 64, 400, 128)      0

 block3_conv1 (Conv2D)       (None, 64, 400, 256)      295168

 block3_conv2 (Conv2D)       (None, 64, 400, 256)      590080

 block3_conv3 (Conv2D)       (None, 64, 400, 256)      590080

 block3_pool (MaxPooling2D)  (None, 32, 200, 256)      0

 block4_conv1 (Conv2D)       (None, 32, 200, 512)      1180160

 block4_conv2 (Conv2D)       (None, 32, 200, 512)      2359808

 block4_conv3 (Conv2D)       (None, 32, 200, 512)      2359808

 block4_pool (MaxPooling2D)  (None, 16, 100, 512)      0

 block5_conv1 (Conv2D)       (None, 16, 100, 512)      2359808

 block5_conv2 (Conv2D)       (None, 16, 100, 512)      2359808

 block5_conv3 (Conv2D)       (None, 16, 100, 512)      2359808

 block5_pool (MaxPooling2D)  (None, 8, 50, 512)        0

=================================================================
Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:
```
# for layer in model.layers:
#   layer.trainable = True
# input1 = model.output
# conv1 = Conv2D(filters=512, kernel_size=(4,4), padding="same", activation="softmax")(input1)
# con2d_tr1 = Conv2DTranspose(5, (2, 2), strides=2, padding="same")(conv1)
# con2d_tr2 = Conv2DTranspose(5, (2, 2), strides=16, padding="same")(con2d_tr1)
# model1 = Model(inputs = model.input, outputs = con2d_tr2)
# model1.summary()
```

In [ ]:
```
for layer in model.layers:
  layer.trainable = False
input1 = model.output
u1 = Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same') (input1)
c1 = Conv2D(64, (3, 3), activation='elu', padding='same') (u1)
c1 = Conv2D(64, (3, 3), activation='elu', padding='same') (c1)

u2 = Conv2DTranspose(32, (2, 2), strides=(2, 2), padding='same') (c1)
c2 = Conv2D(32, (3, 3), activation='elu', padding='same') (u2)
c2 = Conv2D(32, (3, 3), activation='elu', padding='same') (c2)

u3 = Conv2DTranspose(32, (2, 2), strides=(2, 2), padding='same') (c2)
c3 = Conv2D(32, (3, 3), activation='elu', padding='same') (u3)
c3 = Conv2D(32, (3, 3), activation='elu', padding='same') (c3)

u4 = Conv2DTranspose(16, (2, 2), strides=(2, 2), padding='same') (c3)
c4 = Conv2D(16, (3, 3), activation='elu', padding='same') (u4)
c4 = Conv2D(16, (3, 3), activation='elu', padding='same') (c4)

u5 = Conv2DTranspose(8, (2, 2), strides=(2, 2), padding='same') (c4)
c5 = Conv2D(8, (3, 3), activation='elu', padding='same') (u5)
c5 = Conv2D(8, (3, 3), activation='elu', padding='same') (c5)

outputs = Conv2D(5, (1, 1), activation='softmax') (c5)

model1 = Model(inputs=model.input, outputs=[outputs])
# adam = Adam(learning_rate=0.001)
# model.compile(optimizer=adam, loss=bce_dice_loss, metrics=[dice_coef])
```

In [ ]:
```
model1.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 256, 1600, 3)]    0

 block1_conv1 (Conv2D)       (None, 256, 1600, 64)     1792

 block1_conv2 (Conv2D)       (None, 256, 1600, 64)     36928

 block1_pool (MaxPooling2D)  (None, 128, 800, 64)      0

 block2_conv1 (Conv2D)       (None, 128, 800, 128)     73856

 block2_conv2 (Conv2D)       (None, 128, 800, 128)     147584

 block2_pool (MaxPooling2D)  (None, 64, 400, 128)      0

 block3_conv1 (Conv2D)       (None, 64, 400, 256)      295168

 block3_conv2 (Conv2D)       (None, 64, 400, 256)      590080

 block3_conv3 (Conv2D)       (None, 64, 400, 256)      590080

 block3_pool (MaxPooling2D)  (None, 32, 200, 256)      0

 block4_conv1 (Conv2D)       (None, 32, 200, 512)      1180160

 block4_conv2 (Conv2D)       (None, 32, 200, 512)      2359808

 block4_conv3 (Conv2D)       (None, 32, 200, 512)      2359808

 block4_pool (MaxPooling2D)  (None, 16, 100, 512)      0

 block5_conv1 (Conv2D)       (None, 16, 100, 512)      2359808

 block5_conv2 (Conv2D)       (None, 16, 100, 512)      2359808

 block5_conv3 (Conv2D)       (None, 16, 100, 512)      2359808

 block5_pool (MaxPooling2D)  (None, 8, 50, 512)        0

 conv2d_transpose (Conv2DTra  (None, 16, 100, 64)      131136
 nspose)

 conv2d (Conv2D)             (None, 16, 100, 64)       36928

 conv2d_1 (Conv2D)           (None, 16, 100, 64)       36928

 conv2d_transpose_1 (Conv2DT  (None, 32, 200, 32)      8224
 ranspose)
```

```
conv2d_2 (Conv2D)           (None, 32, 200, 32)       9248

conv2d_3 (Conv2D)           (None, 32, 200, 32)       9248

conv2d_transpose_2 (Conv2DT  (None, 64, 400, 32)      4128
ranspose)

conv2d_4 (Conv2D)           (None, 64, 400, 32)       9248

conv2d_5 (Conv2D)           (None, 64, 400, 32)       9248

conv2d_transpose_3 (Conv2DT  (None, 128, 800, 16)     2064
ranspose)

conv2d_6 (Conv2D)           (None, 128, 800, 16)      2320

conv2d_7 (Conv2D)           (None, 128, 800, 16)      2320

conv2d_transpose_4 (Conv2DT  (None, 256, 1600, 8)     520
ranspose)

conv2d_8 (Conv2D)           (None, 256, 1600, 8)      584

conv2d_9 (Conv2D)           (None, 256, 1600, 8)      584

conv2d_10 (Conv2D)          (None, 256, 1600, 5)      45

=================================================================
Total params: 14,977,461
Trainable params: 262,773
Non-trainable params: 14,714,688
_________________________________________________________________
```

In [ ]:

In [ ]:
```
BATCH_SIZE=16
train_dataloader = Dataloder(train_dataset, batch_size=BATCH_SIZE, shuffle=True)
test_dataloader = Dataloder(test_dataset, batch_size=BATCH_SIZE, shuffle=True)

assert train_dataloader[0][0].shape == (BATCH_SIZE, 256, 1600, 3)
assert train_dataloader[0][1].shape == (BATCH_SIZE, 256, 1600, 5)

print(train_dataloader[0][0].shape)
print(train_dataloader[0][1].shape)
print(len(train_dataloader))
print(len(test_dataloader))
type(train_dataset[0])
```

```
(16, 256, 1600, 3)
(16, 256, 1600, 5)
310
93
```

Out[77]:
```
tuple
```

In [ ]:
```
import tensorflow as tf
from tensorflow.keras import callbacks
optim = tf.keras.optimizers.Adam(learning_rate= 0.0009)
focal_loss = sm.losses.cce_dice_loss
model1.compile(optim, focal_loss, metrics=[iou_score])
callbacks = [callbacks.ModelCheckpoint('./best_model.h5', save_weights_only = True, save_best_only = True, \
                                       mode = 'max', monitor = 'val_iou_score', verbose = 1),
    callbacks.ReduceLROnPlateau(monitor = 'val_iou_score', patience = 3, mode = 'max', verbose = 1,min_lr=0.0001)
]
```

In [ ]:
```
history = model1.fit(train_dataloader, steps_per_epoch=(len(train_dataloader))//BATCH_SIZE, epochs=3,\
                     validation_data=test_dataloader, callbacks=callbacks)
```

```
Epoch 1/3
19/19 [==============================] - ETA: 0s - loss: 1.0917 - iou_score: 0.0965
Epoch 1: val_iou_score improved from -inf to 0.18266, saving model to ./best_model.h5
19/19 [==============================] - 118s 6s/step - loss: 1.0917 - iou_score: 0.0965 - val_loss: 0.8787 - val_iou_score: 0.1827 - lr: 9.0000e-04
Epoch 2/3
19/19 [==============================] - ETA: 0s - loss: 0.8165 - iou_score: 0.2218
Epoch 2: val_iou_score improved from 0.18266 to 0.23409, saving model to ./best_model.h5
19/19 [==============================] - 115s 6s/step - loss: 0.8165 - iou_score: 0.2218 - val_loss: 0.7964 - val_iou_score: 0.2341 - lr: 9.0000e-04
Epoch 3/3
19/19 [==============================] - ETA: 0s - loss: 0.7801 - iou_score: 0.2389
Epoch 3: val_iou_score improved from 0.23409 to 0.25021, saving model to ./best_model.h5
19/19 [==============================] - 114s 6s/step - loss: 0.7801 - iou_score: 0.2389 - val_loss: 0.7779 - val_iou_score: 0.2502 - lr: 9.0000e-04
```

In [ ]:
```
history = model1.fit(train_dataloader, steps_per_epoch=(len(train_dataloader))//BATCH_SIZE, epochs=10,\
                     validation_data=test_dataloader, callbacks=callbacks)
```

```
Epoch 1/10
19/19 [==============================] - ETA: 0s - loss: 0.7582 - iou_score: 0.2633
Epoch 1: val_iou_score improved from 0.25021 to 0.27136, saving model to ./best_model.h5
19/19 [==============================] - 115s 6s/step - loss: 0.7582 - iou_score: 0.2633 - val_loss: 0.7421 - val_iou_score: 0.2714 - lr: 9.0000e-04
Epoch 2/10
19/19 [==============================] - ETA: 0s - loss: 0.7549 - iou_score: 0.2607
Epoch 2: val_iou_score improved from 0.27136 to 0.27514, saving model to ./best_model.h5
19/19 [==============================] - 114s 6s/step - loss: 0.7549 - iou_score: 0.2607 - val_loss: 0.7384 - val_iou_score: 0.2751 - lr: 9.0000e-04
Epoch 3/10
19/19 [==============================] - ETA: 0s - loss: 0.7355 - iou_score: 0.2701
Epoch 3: val_iou_score did not improve from 0.27514
19/19 [==============================] - 112s 6s/step - loss: 0.7355 - iou_score: 0.2701 - val_loss: 0.7291 - val_iou_score: 0.2737 - lr: 9.0000e-04
Epoch 4/10
19/19 [==============================] - ETA: 0s - loss: 0.7141 - iou_score: 0.2884
Epoch 4: val_iou_score improved from 0.27514 to 0.28375, saving model to ./best_model.h5
19/19 [==============================] - 114s 6s/step - loss: 0.7141 - iou_score: 0.2884 - val_loss: 0.7160 - val_iou_score: 0.2838 - lr: 9.0000e-04
Epoch 5/10
19/19 [==============================] - ETA: 0s - loss: 0.6970 - iou_score: 0.3027
Epoch 5: val_iou_score improved from 0.28375 to 0.29815, saving model to ./best_model.h5
19/19 [==============================] - 116s 6s/step - loss: 0.6970 - iou_score: 0.3027 - val_loss: 0.7025 - val_iou_score: 0.2982 - lr: 9.0000e-04
Epoch 6/10
19/19 [==============================] - ETA: 0s - loss: 0.6995 - iou_score: 0.3000
Epoch 6: val_iou_score improved from 0.29815 to 0.29858, saving model to ./best_model.h5
19/19 [==============================] - 115s 6s/step - loss: 0.6995 - iou_score: 0.3000 - val_loss: 0.7011 - val_iou_score: 0.2986 - lr: 9.0000e-04
Epoch 7/10
19/19 [==============================] - ETA: 0s - loss: 0.6686 - iou_score: 0.3260
Epoch 7: val_iou_score improved from 0.29858 to 0.33162, saving model to ./best_model.h5
19/19 [==============================] - 114s 6s/step - loss: 0.6686 - iou_score: 0.3260 - val_loss: 0.6635 - val_iou_score: 0.3316 - lr: 9.0000e-04
Epoch 8/10
19/19 [==============================] - ETA: 0s - loss: 0.6402 - iou_score: 0.3505
Epoch 8: val_iou_score improved from 0.33162 to 0.33512, saving model to ./best_model.h5
19/19 [==============================] - 114s 6s/step - loss: 0.6402 - iou_score: 0.3505 - val_loss: 0.6567 - val_iou_score: 0.3351 - lr: 9.0000e-04
Epoch 9/10
19/19 [==============================] - ETA: 0s - loss: 0.6492 - iou_score: 0.3506
Epoch 9: val_iou_score improved from 0.33512 to 0.33789, saving model to ./best_model.h5
19/19 [==============================] - 115s 6s/step - loss: 0.6492 - iou_score: 0.3506 - val_loss: 0.6494 - val_iou_score: 0.3379 - lr: 9.0000e-04
Epoch 10/10
19/19 [==============================] - ETA: 0s - loss: 0.6501 - iou_score: 0.3419
Epoch 10: val_iou_score did not improve from 0.33789
19/19 [==============================] - 116s 6s/step - loss: 0.6501 - iou_score: 0.3419 - val_loss: 0.6498 - val_iou_score: 0.3371 - lr: 9.0000e-04
```

In [ ]:
```
# del model1
```

In [ ]:
```
history = model1.fit(train_dataloader, steps_per_epoch=(len(train_dataloader))//BATCH_SIZE, epochs=10,\
                     validation_data=test_dataloader, callbacks=callbacks)
```

```
Epoch 1/10
19/19 [==============================] - ETA: 0s - loss: 0.6358 - iou_score: 0.3519
Epoch 1: val_iou_score improved from 0.33789 to 0.34003, saving model to ./best_model.h5
19/19 [==============================] - 115s 6s/step - loss: 0.6358 - iou_score: 0.3519 - val_loss: 0.6539 - val_iou_score: 0.3400 - lr: 9.0000e-04
Epoch 2/10
19/19 [==============================] - ETA: 0s - loss: 0.6575 - iou_score: 0.3333
Epoch 2: val_iou_score improved from 0.34003 to 0.34605, saving model to ./best_model.h5
19/19 [==============================] - 114s 6s/step - loss: 0.6575 - iou_score: 0.3333 - val_loss: 0.6448 - val_iou_score: 0.3460 - lr: 9.0000e-04
Epoch 3/10
19/19 [==============================] - ETA: 0s - loss: 0.6476 - iou_score: 0.3445
Epoch 3: val_iou_score improved from 0.34605 to 0.35284, saving model to ./best_model.h5
19/19 [==============================] - 111s 6s/step - loss: 0.6476 - iou_score: 0.3445 - val_loss: 0.6381 - val_iou_score: 0.3528 - lr: 9.0000e-04
Epoch 4/10
19/19 [==============================] - ETA: 0s - loss: 0.6381 - iou_score: 0.3525
Epoch 4: val_iou_score did not improve from 0.35284
19/19 [==============================] - 111s 6s/step - loss: 0.6381 - iou_score: 0.3525 - val_loss: 0.6425 - val_iou_score: 0.3435 - lr: 9.0000e-04
Epoch 5/10
19/19 [==============================] - ETA: 0s - loss: 0.6264 - iou_score: 0.3606
Epoch 5: val_iou_score improved from 0.35284 to 0.35344, saving model to ./best_model.h5
19/19 [==============================] - 115s 6s/step - loss: 0.6264 - iou_score: 0.3606 - val_loss: 0.6362 - val_iou_score: 0.3534 - lr: 9.0000e-04
Epoch 6/10
19/19 [==============================] - ETA: 0s - loss: 0.6391 - iou_score: 0.3498
Epoch 6: val_iou_score improved from 0.35344 to 0.35408, saving model to ./best_model.h5
19/19 [==============================] - 111s 6s/step - loss: 0.6391 - iou_score: 0.3498 - val_loss: 0.6305 - val_iou_score: 0.3541 - lr: 9.0000e-04
Epoch 7/10
19/19 [==============================] - ETA: 0s - loss: 0.6385 - iou_score: 0.3527
Epoch 7: val_iou_score did not improve from 0.35408
19/19 [==============================] - 111s 6s/step - loss: 0.6385 - iou_score: 0.3527 - val_loss: 0.6338 - val_iou_score: 0.3511 - lr: 9.0000e-04
Epoch 8/10
19/19 [==============================] - ETA: 0s - loss: 0.6338 - iou_score: 0.3523
Epoch 8: val_iou_score improved from 0.35408 to 0.35520, saving model to ./best_model.h5
19/19 [==============================] - 113s 6s/step - loss: 0.6338 - iou_score: 0.3523 - val_loss: 0.6341 - val_iou_score: 0.3552 - lr: 9.0000e-04
Epoch 9/10
19/19 [==============================] - ETA: 0s - loss: 0.6354 - iou_score: 0.3568
Epoch 9: val_iou_score improved from 0.35520 to 0.35784, saving model to ./best_model.h5
19/19 [==============================] - 111s 6s/step - loss: 0.6354 - iou_score: 0.3568 - val_loss: 0.6273 - val_iou_score: 0.3578 - lr: 9.0000e-04
Epoch 10/10
19/19 [==============================] - ETA: 0s - loss: 0.6194 - iou_score: 0.3640
Epoch 10: val_iou_score did not improve from 0.35784
19/19 [==============================] - 114s 6s/step - loss: 0.6194 - iou_score: 0.3640 - val_loss: 0.6417 - val_iou_score: 0.3416 - lr: 9.0000e-04
```

In [ ]:
```
history = model1.fit(train_dataloader, steps_per_epoch=(len(train_dataloader))//BATCH_SIZE, epochs=10,\
                     validation_data=test_dataloader, callbacks=callbacks)
```

```
Epoch 1/10
19/19 [==============================] - ETA: 0s - loss: 0.6460 - iou_score: 0.3476
Epoch 1: val_iou_score did not improve from 0.35784
19/19 [==============================] - 114s 6s/step - loss: 0.6460 - iou_score: 0.3476 - val_loss: 0.6424 - val_iou_score: 0.3387 - lr: 9.0000e-04
Epoch 2/10
19/19 [==============================] - ETA: 0s - loss: 0.6119 - iou_score: 0.3673
Epoch 2: val_iou_score improved from 0.35784 to 0.35855, saving model to ./best_model.h5
19/19 [==============================] - 112s 6s/step - loss: 0.6119 - iou_score: 0.3673 - val_loss: 0.6281 - val_iou_score: 0.3586 - lr: 9.0000e-04
Epoch 3/10
19/19 [==============================] - ETA: 0s - loss: 0.6032 - iou_score: 0.3742
Epoch 3: val_iou_score did not improve from 0.35855
19/19 [==============================] - 113s 6s/step - loss: 0.6032 - iou_score: 0.3742 - val_loss: 0.6259 - val_iou_score: 0.3581 - lr: 9.0000e-04
Epoch 4/10
19/19 [==============================] - ETA: 0s - loss: 0.6024 - iou_score: 0.3841
Epoch 4: val_iou_score improved from 0.35855 to 0.36267, saving model to ./best_model.h5
19/19 [==============================] - 110s 6s/step - loss: 0.6024 - iou_score: 0.3841 - val_loss: 0.6133 - val_iou_score: 0.3627 - lr: 9.0000e-04
Epoch 5/10
19/19 [==============================] - ETA: 0s - loss: 0.6447 - iou_score: 0.3396
Epoch 5: val_iou_score improved from 0.36267 to 0.37547, saving model to ./best_model.h5
19/19 [==============================] - 110s 6s/step - loss: 0.6447 - iou_score: 0.3396 - val_loss: 0.6070 - val_iou_score: 0.3755 - lr: 9.0000e-04
Epoch 6/10
19/19 [==============================] - ETA: 0s - loss: 0.6025 - iou_score: 0.3739
Epoch 6: val_iou_score did not improve from 0.37547
19/19 [==============================] - 111s 6s/step - loss: 0.6025 - iou_score: 0.3739 - val_loss: 0.6180 - val_iou_score: 0.3685 - lr: 9.0000e-04
Epoch 7/10
19/19 [==============================] - ETA: 0s - loss: 0.6379 - iou_score: 0.3587
Epoch 7: val_iou_score did not improve from 0.37547
19/19 [==============================] - 110s 6s/step - loss: 0.6379 - iou_score: 0.3587 - val_loss: 0.6354 - val_iou_score: 0.3527 - lr: 9.0000e-04
Epoch 8/10
19/19 [==============================] - ETA: 0s - loss: 0.6408 - iou_score: 0.3491
Epoch 8: val_iou_score did not improve from 0.37547

Epoch 8: ReduceLROnPlateau reducing learning rate to 0.0001.
19/19 [==============================] - 111s 6s/step - loss: 0.6408 - iou_score: 0.3491 - val_loss: 0.6520 - val_iou_score: 0.3394 - lr: 9.0000e-04
Epoch 9/10
19/19 [==============================] - ETA: 0s - loss: 0.6230 - iou_score: 0.3649
Epoch 9: val_iou_score did not improve from 0.37547
19/19 [==============================] - 108s 6s/step - loss: 0.6230 - iou_score: 0.3649 - val_loss: 0.6317 - val_iou_score: 0.3572 - lr: 1.0000e-04
Epoch 10/10
19/19 [==============================] - ETA: 0s - loss: 0.6270 - iou_score: 0.3533
Epoch 10: val_iou_score did not improve from 0.37547
19/19 [==============================] - 107s 6s/step - loss: 0.6270 - iou_score: 0.3533 - val_loss: 0.6278 - val_iou_score: 0.3617 - lr: 1.0000e-04
```

In [ ]:
```
import tensorflow as tf
from tensorflow.keras import callbacks
optim = tf.keras.optimizers.Adam(learning_rate= 0.001)
focal_loss = sm.losses.cce_dice_loss
model1.compile(optim, focal_loss, metrics=[iou_score])
callbacks = [callbacks.ModelCheckpoint('./best_model.h5', save_weights_only = True, save_best_only = True, \
                                       mode = 'max', monitor = 'val_iou_score', verbose = 1),
    callbacks.ReduceLROnPlateau(monitor = 'val_iou_score', patience = 3, mode = 'max', verbose = 1,min_lr=0.0001)
]
```

In [ ]:
```
model1.load_weights("/content/best_model.h5")
```

In [ ]:
```
history = model1.fit(train_dataloader, steps_per_epoch=(len(train_dataloader))//BATCH_SIZE, epochs=5,\
                     validation_data=test_dataloader, callbacks=callbacks)
```

```
Epoch 1/5
19/19 [==============================] - ETA: 0s - loss: 0.6312 - iou_score: 0.3596
Epoch 1: val_iou_score improved from -inf to 0.36423, saving model to ./best_model.h5
19/19 [==============================] - 166s 9s/step - loss: 0.6312 - iou_score: 0.3596 - val_loss: 0.6252 - val_iou_score: 0.3642 - lr: 0.0010
Epoch 2/5
19/19 [==============================] - ETA: 0s - loss: 0.6125 - iou_score: 0.3704
Epoch 2: val_iou_score improved from 0.36423 to 0.36843, saving model to ./best_model.h5
19/19 [==============================] - 110s 6s/step - loss: 0.6125 - iou_score: 0.3704 - val_loss: 0.6094 - val_iou_score: 0.3684 - lr: 0.0010
Epoch 3/5
19/19 [==============================] - ETA: 0s - loss: 0.6013 - iou_score: 0.3747
Epoch 3: val_iou_score improved from 0.36843 to 0.37341, saving model to ./best_model.h5
19/19 [==============================] - 163s 9s/step - loss: 0.6013 - iou_score: 0.3747 - val_loss: 0.6035 - val_iou_score: 0.3734 - lr: 0.0010
Epoch 4/5
19/19 [==============================] - ETA: 0s - loss: 0.6070 - iou_score: 0.3776
Epoch 4: val_iou_score did not improve from 0.37341
19/19 [==============================] - 111s 6s/step - loss: 0.6070 - iou_score: 0.3776 - val_loss: 0.6071 - val_iou_score: 0.3721 - lr: 0.0010
Epoch 5/5
19/19 [==============================] - ETA: 0s - loss: 0.6311 - iou_score: 0.3562
Epoch 5: val_iou_score did not improve from 0.37341
19/19 [==============================] - 110s 6s/step - loss: 0.6311 - iou_score: 0.3562 - val_loss: 0.6448 - val_iou_score: 0.3522 - lr: 0.0010
```

In [ ]:
```
import tensorflow as tf
from tensorflow.keras import callbacks
optim = tf.keras.optimizers.Adam(learning_rate= 0.001)
focal_loss = sm.losses.cce_dice_loss
model1.compile(optim, focal_loss, metrics=["accuracy"])
callbacks = [callbacks.ModelCheckpoint('./best_model.h5', save_weights_only = True, save_best_only = True, \
                                       mode = 'max', monitor = 'val_iou_score', verbose = 1),
    callbacks.ReduceLROnPlateau(monitor = 'val_iou_score', patience = 3, mode = 'max', verbose = 1,min_lr=0.0001)
]
```

In [ ]:
```
model1.load_weights("/content/best_model.h5")
```

In [ ]:
```
history = model1.fit(train_dataloader, steps_per_epoch=(len(train_dataloader))//BATCH_SIZE, epochs=5,\
                     validation_data=test_dataloader, callbacks=callbacks)
```

```
Epoch 1/5
19/19 [==============================] - ETA: 0s - loss: 0.6178 - accuracy: 0.9540

WARNING:tensorflow:Can save best model only with val_iou_score available, skipping.
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_iou_score` which is not available. Available metrics are: loss,accuracy,val_loss,
val_accuracy,lr

19/19 [==============================] - 112s 6s/step - loss: 0.6178 - accuracy: 0.9540 - val_loss: 0.6120 - val_accuracy: 0.9522 - lr: 0.0010
Epoch 2/5
19/19 [==============================] - ETA: 0s - loss: 0.6126 - accuracy: 0.9537

WARNING:tensorflow:Can save best model only with val_iou_score available, skipping.
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_iou_score` which is not available. Available metrics are: loss,accuracy,val_loss,
val_accuracy,lr

19/19 [==============================] - 107s 6s/step - loss: 0.6126 - accuracy: 0.9537 - val_loss: 0.6181 - val_accuracy: 0.9488 - lr: 0.0010
Epoch 3/5
19/19 [==============================] - ETA: 0s - loss: 0.6102 - accuracy: 0.9541

WARNING:tensorflow:Can save best model only with val_iou_score available, skipping.
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_iou_score` which is not available. Available metrics are: loss,accuracy,val_loss,
val_accuracy,lr

19/19 [==============================] - 110s 6s/step - loss: 0.6102 - accuracy: 0.9541 - val_loss: 0.6052 - val_accuracy: 0.9515 - lr: 0.0010
Epoch 4/5
19/19 [==============================] - ETA: 0s - loss: 0.5801 - accuracy: 0.9508

WARNING:tensorflow:Can save best model only with val_iou_score available, skipping.
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_iou_score` which is not available. Available metrics are: loss,accuracy,val_loss,
val_accuracy,lr

19/19 [==============================] - 112s 6s/step - loss: 0.5801 - accuracy: 0.9508 - val_loss: 0.6303 - val_accuracy: 0.9564 - lr: 0.0010
Epoch 5/5
19/19 [==============================] - ETA: 0s - loss: 0.6295 - accuracy: 0.9535

WARNING:tensorflow:Can save best model only with val_iou_score available, skipping.
WARNING:tensorflow:Learning rate reduction is conditioned on metric `val_iou_score` which is not available. Available metrics are: loss,accuracy,val_loss,
val_accuracy,lr

19/19 [==============================] - 113s 6s/step - loss: 0.6295 - accuracy: 0.9535 - val_loss: 0.6086 - val_accuracy: 0.9518 - lr: 0.0010
```

In [ ]:
```
import tensorflow as tf
from tensorflow.keras import callbacks
optim = tf.keras.optimizers.Adam(learning_rate= 0.001)
focal_loss = sm.losses.cce_dice_loss
model1.compile(optim, focal_loss, metrics=[iou_score, "accuracy"])
callbacks = [callbacks.ModelCheckpoint('./best_model.h5', save_weights_only = True, save_best_only = True, \
                                       mode = 'max', monitor = 'val_iou_score', verbose = 1),
    callbacks.ReduceLROnPlateau(monitor = 'val_iou_score', patience = 3, mode = 'max', verbose = 1,min_lr=0.0001)
]
```

In [ ]:
```
model1.load_weights("/content/best_model.h5")
```

In [ ]:
```
history = model1.fit(train_dataloader, steps_per_epoch=(len(train_dataloader))//BATCH_SIZE, epochs=10,\
                     validation_data=test_dataloader, callbacks=callbacks)
```

```
Epoch 1/10
19/19 [==============================] - ETA: 0s - loss: 0.6376 - iou_score: 0.3512 - accuracy: 0.9494
Epoch 1: val_iou_score improved from -inf to 0.37633, saving model to ./best_model.h5
19/19 [==============================] - 116s 6s/step - loss: 0.6376 - iou_score: 0.3512 - accuracy: 0.9494 - val_loss: 0.6135 - val_iou_score: 0.3763 - v
al_accuracy: 0.9492 - lr: 0.0010
Epoch 2/10
19/19 [==============================] - ETA: 0s - loss: 0.6119 - iou_score: 0.3673 - accuracy: 0.9545
Epoch 2: val_iou_score did not improve from 0.37633
19/19 [==============================] - 109s 6s/step - loss: 0.6119 - iou_score: 0.3673 - accuracy: 0.9545 - val_loss: 0.6221 - val_iou_score: 0.3552 - v
al_accuracy: 0.9469 - lr: 0.0010
Epoch 3/10
19/19 [==============================] - ETA: 0s - loss: 0.6112 - iou_score: 0.3704 - accuracy: 0.9502
Epoch 3: val_iou_score did not improve from 0.37633
19/19 [==============================] - 113s 6s/step - loss: 0.6112 - iou_score: 0.3704 - accuracy: 0.9502 - val_loss: 0.6132 - val_iou_score: 0.3637 - v
al_accuracy: 0.9475 - lr: 0.0010
Epoch 4/10
19/19 [==============================] - ETA: 0s - loss: 0.6095 - iou_score: 0.3677 - accuracy: 0.9514
Epoch 4: val_iou_score did not improve from 0.37633

Epoch 4: ReduceLROnPlateau reducing learning rate to 0.00010000000474974513.
19/19 [==============================] - 109s 6s/step - loss: 0.6095 - iou_score: 0.3677 - accuracy: 0.9514 - val_loss: 0.6089 - val_iou_score: 0.3691 - v
al_accuracy: 0.9550 - lr: 0.0010
Epoch 5/10
19/19 [==============================] - ETA: 0s - loss: 0.5567 - iou_score: 0.4144 - accuracy: 0.9545
Epoch 5: val_iou_score did not improve from 0.37633
19/19 [==============================] - 109s 6s/step - loss: 0.5567 - iou_score: 0.4144 - accuracy: 0.9545 - val_loss: 0.5998 - val_iou_score: 0.3759 - v
al_accuracy: 0.9517 - lr: 1.0000e-04
Epoch 6/10
19/19 [==============================] - ETA: 0s - loss: 0.6043 - iou_score: 0.3712 - accuracy: 0.9519
Epoch 6: val_iou_score did not improve from 0.37633
19/19 [==============================] - 110s 6s/step - loss: 0.6043 - iou_score: 0.3712 - accuracy: 0.9519 - val_loss: 0.5996 - val_iou_score: 0.3745 - v
al_accuracy: 0.9525 - lr: 1.0000e-04
Epoch 7/10
19/19 [==============================] - ETA: 0s - loss: 0.5900 - iou_score: 0.3847 - accuracy: 0.9526
Epoch 7: val_iou_score improved from 0.37633 to 0.38026, saving model to ./best_model.h5
19/19 [==============================] - 110s 6s/step - loss: 0.5900 - iou_score: 0.3847 - accuracy: 0.9526 - val_loss: 0.5922 - val_iou_score: 0.3803 - v
al_accuracy: 0.9531 - lr: 1.0000e-04
Epoch 8/10
19/19 [==============================] - ETA: 0s - loss: 0.5883 - iou_score: 0.3855 - accuracy: 0.9491
Epoch 8: val_iou_score did not improve from 0.38026
19/19 [==============================] - 112s 6s/step - loss: 0.5883 - iou_score: 0.3855 - accuracy: 0.9491 - val_loss: 0.5918 - val_iou_score: 0.3798 - v
al_accuracy: 0.9532 - lr: 1.0000e-04
Epoch 9/10
19/19 [==============================] - ETA: 0s - loss: 0.5798 - iou_score: 0.3959 - accuracy: 0.9514
Epoch 9: val_iou_score improved from 0.38026 to 0.38409, saving model to ./best_model.h5
19/19 [==============================] - 114s 6s/step - loss: 0.5798 - iou_score: 0.3959 - accuracy: 0.9514 - val_loss: 0.5894 - val_iou_score: 0.3841 - v
al_accuracy: 0.9535 - lr: 1.0000e-04
Epoch 10/10
19/19 [==============================] - ETA: 0s - loss: 0.5981 - iou_score: 0.3761 - accuracy: 0.9501
Epoch 10: val_iou_score did not improve from 0.38409
19/19 [==============================] - 111s 6s/step - loss: 0.5981 - iou_score: 0.3761 - accuracy: 0.9501 - val_loss: 0.5949 - val_iou_score: 0.3790 - v
al_accuracy: 0.9547 - lr: 1.0000e-04
```

In [ ]:
```
history = model1.fit(train_dataloader, steps_per_epoch=(len(train_dataloader))//BATCH_SIZE, epochs=10,\
                     validation_data=test_dataloader, callbacks=callbacks)
```

```
Epoch 1/10
19/19 [==============================] - ETA: 0s - loss: 0.5861 - iou_score: 0.3816 - accuracy: 0.9563
Epoch 1: val_iou_score did not improve from 0.38409
19/19 [==============================] - 117s 6s/step - loss: 0.5861 - iou_score: 0.3816 - accuracy: 0.9563 - val_loss: 0.5880 - val_iou_score: 0.3831 - v
al_accuracy: 0.9536 - lr: 1.0000e-04
Epoch 2/10
19/19 [==============================] - ETA: 0s - loss: 0.5766 - iou_score: 0.3921 - accuracy: 0.9545
Epoch 2: val_iou_score did not improve from 0.38409
19/19 [==============================] - 110s 6s/step - loss: 0.5766 - iou_score: 0.3921 - accuracy: 0.9545 - val_loss: 0.5946 - val_iou_score: 0.3780 - v
al_accuracy: 0.9539 - lr: 1.0000e-04
Epoch 3/10
19/19 [==============================] - ETA: 0s - loss: 0.5921 - iou_score: 0.3762 - accuracy: 0.9552
Epoch 3: val_iou_score did not improve from 0.38409
19/19 [==============================] - 112s 6s/step - loss: 0.5921 - iou_score: 0.3762 - accuracy: 0.9552 - val_loss: 0.5932 - val_iou_score: 0.3809 - v
al_accuracy: 0.9543 - lr: 1.0000e-04
Epoch 4/10
19/19 [==============================] - ETA: 0s - loss: 0.5787 - iou_score: 0.3957 - accuracy: 0.9561
Epoch 4: val_iou_score did not improve from 0.38409

Epoch 4: ReduceLROnPlateau reducing learning rate to 0.0001.
19/19 [==============================] - 112s 6s/step - loss: 0.5787 - iou_score: 0.3957 - accuracy: 0.9561 - val_loss: 0.5967 - val_iou_score: 0.3757 - v
al_accuracy: 0.9551 - lr: 1.0000e-04
Epoch 5/10
19/19 [==============================] - ETA: 0s - loss: 0.5759 - iou_score: 0.3945 - accuracy: 0.9521
Epoch 5: val_iou_score did not improve from 0.38409
19/19 [==============================] - 109s 6s/step - loss: 0.5759 - iou_score: 0.3945 - accuracy: 0.9521 - val_loss: 0.5914 - val_iou_score: 0.3794 - v
al_accuracy: 0.9545 - lr: 1.0000e-04
Epoch 6/10
19/19 [==============================] - ETA: 0s - loss: 0.5832 - iou_score: 0.3909 - accuracy: 0.9547
Epoch 6: val_iou_score did not improve from 0.38409
19/19 [==============================] - 111s 6s/step - loss: 0.5832 - iou_score: 0.3909 - accuracy: 0.9547 - val_loss: 0.5887 - val_iou_score: 0.3828 - v
al_accuracy: 0.9551 - lr: 1.0000e-04
Epoch 7/10
19/19 [==============================] - ETA: 0s - loss: 0.5851 - iou_score: 0.3878 - accuracy: 0.9513
Epoch 7: val_iou_score did not improve from 0.38409
19/19 [==============================] - 111s 6s/step - loss: 0.5851 - iou_score: 0.3878 - accuracy: 0.9513 - val_loss: 0.5859 - val_iou_score: 0.3839 - v
al_accuracy: 0.9552 - lr: 1.0000e-04
Epoch 8/10
19/19 [==============================] - ETA: 0s - loss: 0.5806 - iou_score: 0.3884 - accuracy: 0.9557
Epoch 8: val_iou_score improved from 0.38409 to 0.38706, saving model to ./best_model.h5
19/19 [==============================] - 111s 6s/step - loss: 0.5806 - iou_score: 0.3884 - accuracy: 0.9557 - val_loss: 0.5838 - val_iou_score: 0.3871 - v
al_accuracy: 0.9549 - lr: 1.0000e-04
Epoch 9/10
19/19 [==============================] - ETA: 0s - loss: 0.5871 - iou_score: 0.3824 - accuracy: 0.9562
Epoch 9: val_iou_score did not improve from 0.38706
19/19 [==============================] - 111s 6s/step - loss: 0.5871 - iou_score: 0.3824 - accuracy: 0.9562 - val_loss: 0.5861 - val_iou_score: 0.3839 - v
al_accuracy: 0.9552 - lr: 1.0000e-04
Epoch 10/10
19/19 [==============================] - ETA: 0s - loss: 0.5862 - iou_score: 0.3807 - accuracy: 0.9555
Epoch 10: val_iou_score did not improve from 0.38706
19/19 [==============================] - 113s 6s/step - loss: 0.5862 - iou_score: 0.3807 - accuracy: 0.9555 - val_loss: 0.5866 - val_iou_score: 0.3854 - v
al_accuracy: 0.9547 - lr: 1.0000e-04
```

In [ ]:
```
history = model1.fit(train_dataloader, steps_per_epoch=(len(train_dataloader))//BATCH_SIZE, epochs=10,\
                     validation_data=test_dataloader, callbacks=callbacks)
```

```
Epoch 1/10
19/19 [==============================] - ETA: 0s - loss: 0.5969 - iou_score: 0.3787 - accuracy: 0.9544
Epoch 1: val_iou_score improved from 0.38706 to 0.38755, saving model to ./best_model.h5
19/19 [==============================] - 112s 6s/step - loss: 0.5969 - iou_score: 0.3787 - accuracy: 0.9544 - val_loss: 0.5818 - val_iou_score: 0.3876 - v
al_accuracy: 0.9550 - lr: 1.0000e-04
Epoch 2/10
19/19 [==============================] - ETA: 0s - loss: 0.5745 - iou_score: 0.3912 - accuracy: 0.9537
Epoch 2: val_iou_score did not improve from 0.38755
19/19 [==============================] - 112s 6s/step - loss: 0.5745 - iou_score: 0.3912 - accuracy: 0.9537 - val_loss: 0.5868 - val_iou_score: 0.3855 - v
al_accuracy: 0.9546 - lr: 1.0000e-04
Epoch 3/10
19/19 [==============================] - ETA: 0s - loss: 0.5860 - iou_score: 0.3863 - accuracy: 0.9541
Epoch 3: val_iou_score did not improve from 0.38755
19/19 [==============================] - 112s 6s/step - loss: 0.5860 - iou_score: 0.3863 - accuracy: 0.9541 - val_loss: 0.5828 - val_iou_score: 0.3868 - v
al_accuracy: 0.9560 - lr: 1.0000e-04
Epoch 4/10
19/19 [==============================] - ETA: 0s - loss: 0.5765 - iou_score: 0.3944 - accuracy: 0.9535
Epoch 4: val_iou_score did not improve from 0.38755
19/19 [==============================] - 114s 6s/step - loss: 0.5765 - iou_score: 0.3944 - accuracy: 0.9535 - val_loss: 0.5878 - val_iou_score: 0.3845 - v
al_accuracy: 0.9554 - lr: 1.0000e-04
Epoch 5/10
19/19 [==============================] - ETA: 0s - loss: 0.5832 - iou_score: 0.3910 - accuracy: 0.9524
Epoch 5: val_iou_score did not improve from 0.38755
19/19 [==============================] - 116s 6s/step - loss: 0.5832 - iou_score: 0.3910 - accuracy: 0.9524 - val_loss: 0.5842 - val_iou_score: 0.3865 - v
al_accuracy: 0.9553 - lr: 1.0000e-04
Epoch 6/10
19/19 [==============================] - ETA: 0s - loss: 0.5977 - iou_score: 0.3750 - accuracy: 0.9527
Epoch 6: val_iou_score did not improve from 0.38755
19/19 [==============================] - 110s 6s/step - loss: 0.5977 - iou_score: 0.3750 - accuracy: 0.9527 - val_loss: 0.5871 - val_iou_score: 0.3849 - v
al_accuracy: 0.9547 - lr: 1.0000e-04
Epoch 7/10
19/19 [==============================] - ETA: 0s - loss: 0.5682 - iou_score: 0.4058 - accuracy: 0.9529
Epoch 7: val_iou_score did not improve from 0.38755
19/19 [==============================] - 163s 9s/step - loss: 0.5682 - iou_score: 0.4058 - accuracy: 0.9529 - val_loss: 0.5824 - val_iou_score: 0.3874 - v
al_accuracy: 0.9544 - lr: 1.0000e-04
Epoch 8/10
19/19 [==============================] - ETA: 0s - loss: 0.5804 - iou_score: 0.3895 - accuracy: 0.9578
Epoch 8: val_iou_score improved from 0.38755 to 0.38920, saving model to ./best_model.h5
19/19 [==============================] - 117s 6s/step - loss: 0.5804 - iou_score: 0.3895 - accuracy: 0.9578 - val_loss: 0.5804 - val_iou_score: 0.3892 - v
al_accuracy: 0.9540 - lr: 1.0000e-04
Epoch 9/10
19/19 [==============================] - ETA: 0s - loss: 0.5703 - iou_score: 0.3987 - accuracy: 0.9558
Epoch 9: val_iou_score did not improve from 0.38920
19/19 [==============================] - 117s 6s/step - loss: 0.5703 - iou_score: 0.3987 - accuracy: 0.9558 - val_loss: 0.5865 - val_iou_score: 0.3837 - v
al_accuracy: 0.9547 - lr: 1.0000e-04
Epoch 10/10
19/19 [==============================] - ETA: 0s - loss: 0.5733 - iou_score: 0.4019 - accuracy: 0.9572
Epoch 10: val_iou_score did not improve from 0.38920
19/19 [==============================] - 118s 6s/step - loss: 0.5733 - iou_score: 0.4019 - accuracy: 0.9572 - val_loss: 0.5884 - val_iou_score: 0.3831 - v
al_accuracy: 0.9553 - lr: 1.0000e-04
```

In [ ]:
```
model1.load_weights("/content/drive/MyDrive/Case Study-2/best_model_vgg16_18-12-22.h5")
```

4.6 Visualizing the result of VGG16

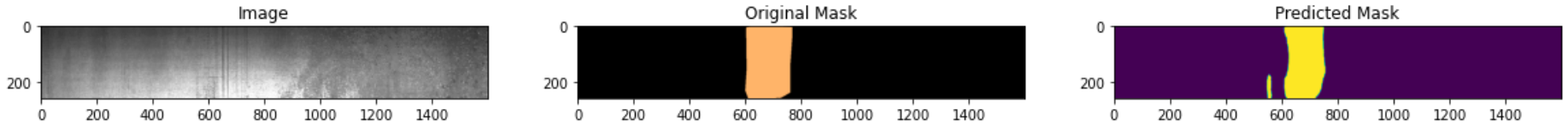
In [ ]:
```
images_ = validation['image_path'].values
masks_ = validation["mask_path"].values
lst = np.arange(len(images_))
len(lst)
np.random.choice(lst, size = 20, replace = False)
```

Out[61]:
```
array([ 97, 167, 321,  92, 376, 380, 252,  75, 455,  58,  29, 312, 557,
       149, 600, 407, 605, 546, 354, 400])
```

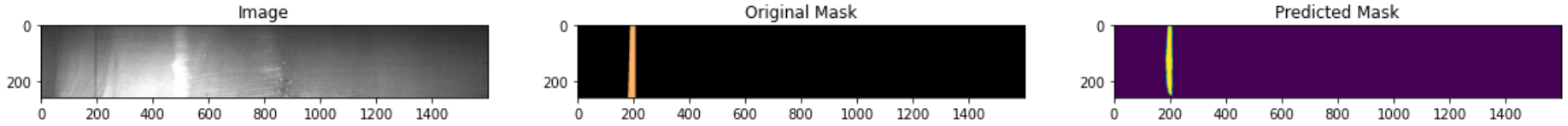
In [ ]:
```
%matplotlib inline
import matplotlib.pyplot as plt
```

In [ ]:
```
# Visualizing the predicted mask for test data.
import random
ids = np.random.choice(lst, size = 20, replace = False)
for i in ids:
  image = cv2.imread(images_[i], cv2.IMREAD_UNCHANGED)
  # image = cv2.resize(image, (256, 256),  interpolation = cv2.INTER_AREA)
  image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
  image = np.expand_dims(image, axis=0)

  mask  = cv2.imread(masks_[i], cv2.IMREAD_UNCHANGED)
  # mask = cv2.resize(mask, (256, 256),  interpolation = cv2.INTER_AREA)

  pred = model1.predict(image,verbose=1)
  pred = tf.argmax(pred, axis=-1)

  fig = plt.figure(figsize=(20,14))

  ax1 = fig.add_subplot(1, 3, 1)
  ax1.imshow(image[0,:,:])
  ax2=fig.add_subplot(1, 3, 2)
  ax2.imshow(mask)
  ax3=fig.add_subplot(1, 3, 3)
  ax3.imshow(pred[0,:,:])
  ax1.title.set_text('Image')
  ax2.title.set_text('Original Mask')
  ax3.title.set_text('Predicted Mask')
  plt.show()
```

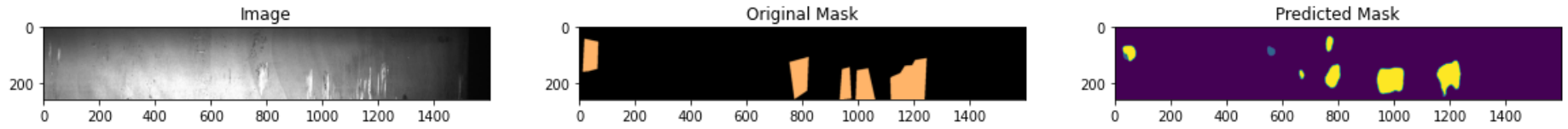
```
1/1 [==============================] - 0s 19ms/step
```



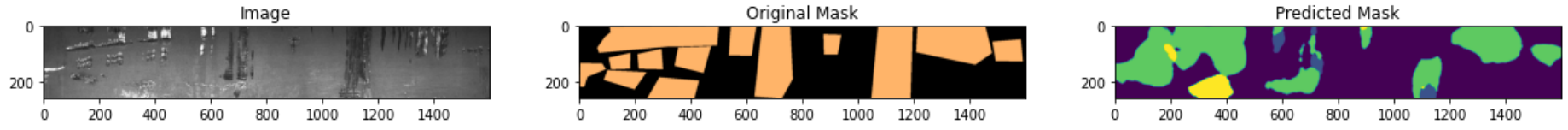

```
1/1 [==============================] - 0s 17ms/step
```



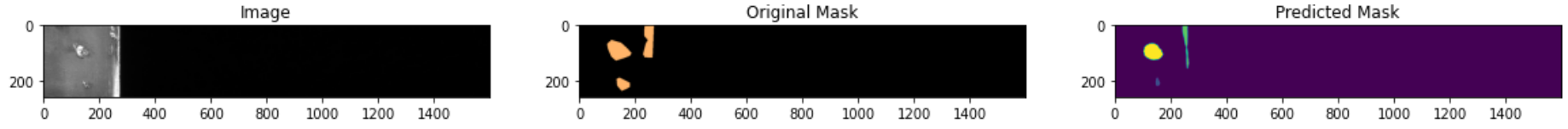

```
1/1 [==============================] - 0s 17ms/step
```



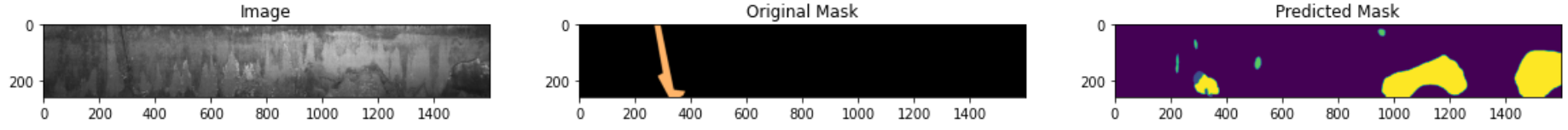

```
1/1 [==============================] - 0s 18ms/step
```



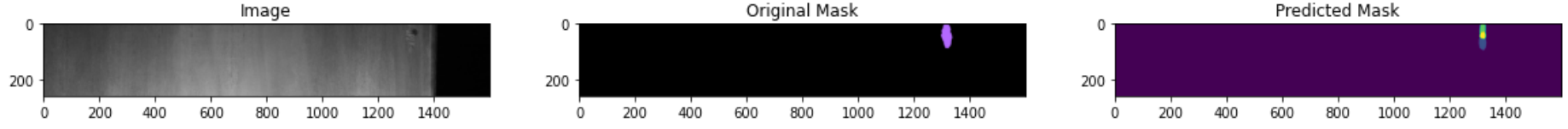

```
1/1 [==============================] - 0s 17ms/step
```



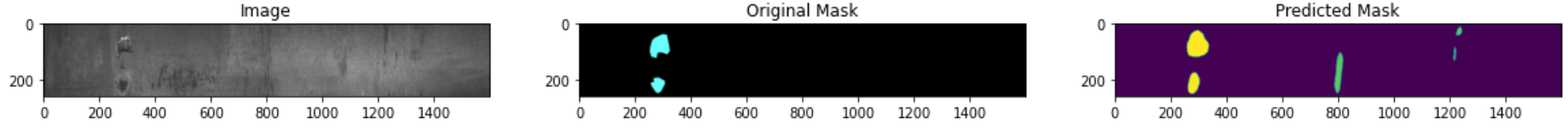

```
1/1 [==============================] - 0s 18ms/step
```



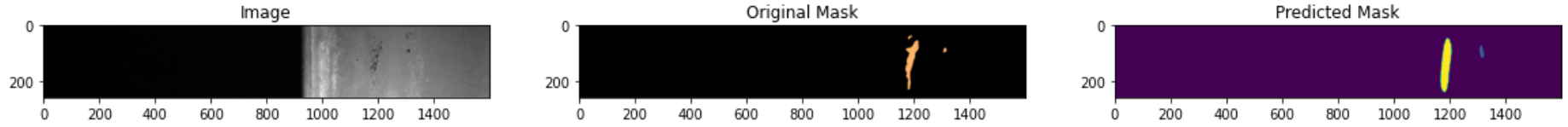

```
1/1 [==============================] - 0s 18ms/step
```



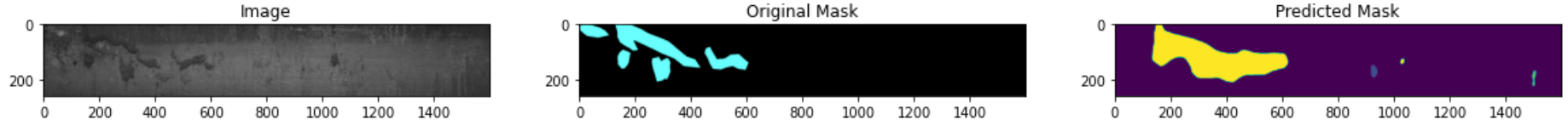

```
1/1 [==============================] - 0s 17ms/step
```



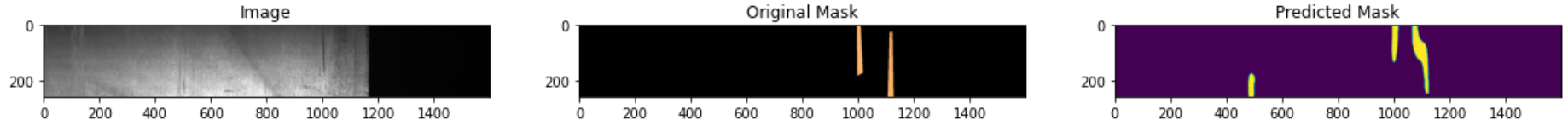

```
1/1 [==============================] - 0s 19ms/step
```



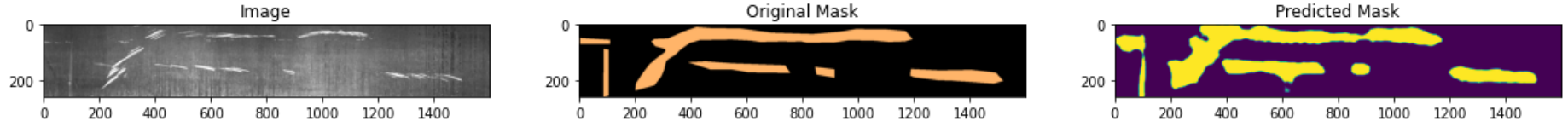

```
1/1 [==============================] - 0s 25ms/step
```



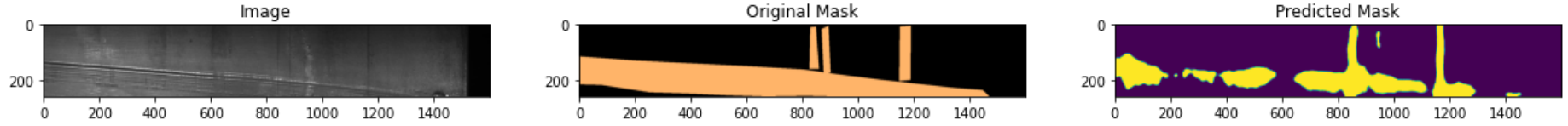

```
1/1 [==============================] - 0s 18ms/step
```



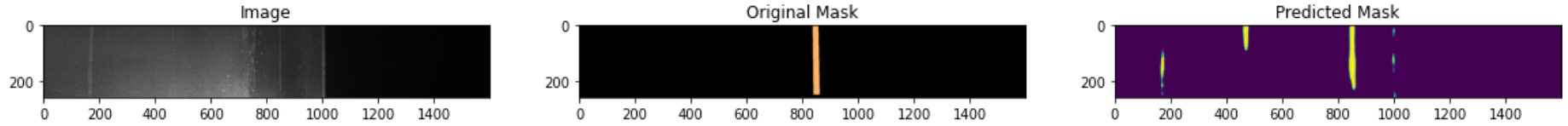

```
1/1 [==============================] - 0s 17ms/step
```



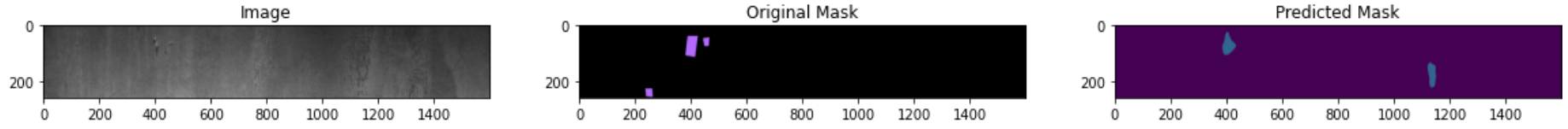

```
1/1 [==============================] - 0s 19ms/step
```



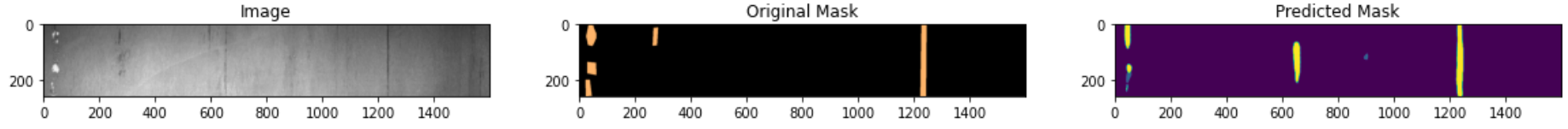

```
1/1 [==============================] - 0s 19ms/step
```



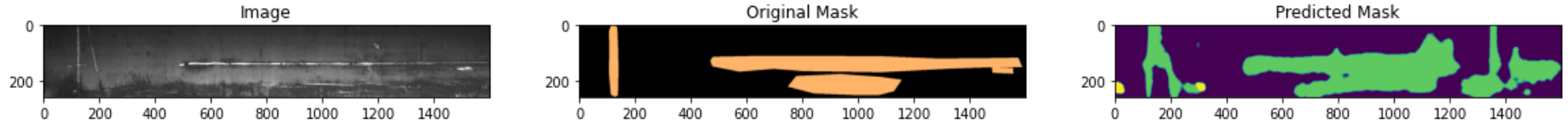

```
1/1 [==============================] - 0s 20ms/step
```



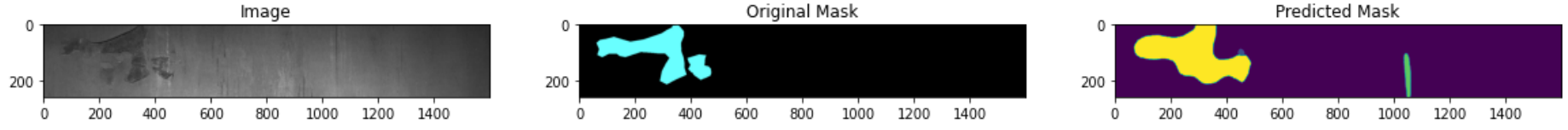

```
1/1 [==============================] - 0s 18ms/step
```



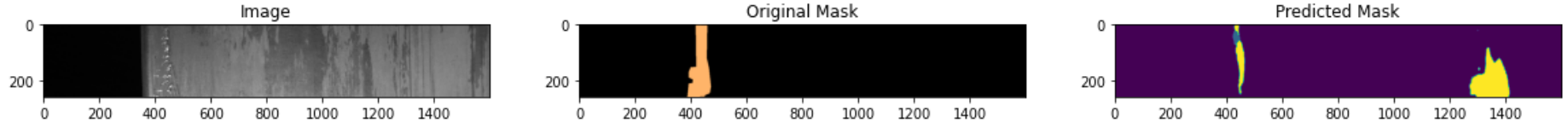

```
1/1 [==============================] - 0s 17ms/step
```



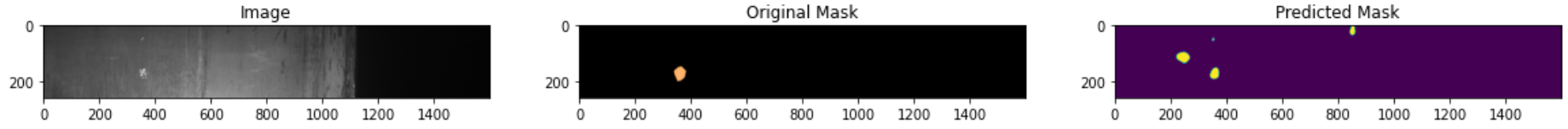

```
1/1 [==============================] - 0s 17ms/step
```




```
1/1 [==============================] - 0s 18ms/step
```




```
1/1 [==============================] - 0s 19ms/step
```



In [ ]:

4.7 Applying Xception

In [85]:
```
model_X = tf.keras.applications.Xception(
    include_top=False,
    weights="imagenet",
    input_shape=(256, 1600, 3),
    classes=1000,
    classifier_activation="softmax",
)
```

In [86]:
```
model_X.summary()
```

```
Model: "xception"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_3 (InputLayer)           [(None, 256, 1600,   0           []
                                3)]

 block1_conv1 (Conv2D)          (None, 127, 799, 32  864         ['input_3[0][0]']
                                )

 block1_conv1_bn (BatchNormaliz  (None, 127, 799, 32  128        ['block1_conv1[0][0]']
 ation)                         )

 block1_conv1_act (Activation)  (None, 127, 799, 32  0           ['block1_conv1_bn[0][0]']
                                )

 block1_conv2 (Conv2D)          (None, 125, 797, 64  18432       ['block1_conv1_act[0][0]']
                                )
```

In [87]:
```
for layer in model_X.layers:
  layer.trainable = False
input1 = model_X.output

u1 = Conv2DTranspose(512, (2, 2), strides=(2, 2), padding='same') (input1)
c1 = Conv2D(512, (3, 3), activation='elu', padding='same') (u1)
c1 = Conv2D(512, (3, 3), activation='elu', padding='same') (c1)

u2 = Conv2DTranspose(128, (2, 2), strides=(2, 2), padding='same') (c1)
c2 = Conv2D(128, (3, 3), activation='elu', padding='same') (u2)
c2 = Conv2D(128, (3, 3), activation='elu', padding='same') (c2)

u3 = Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same') (c2)
c3 = Conv2D(64, (3, 3), activation='elu', padding='same') (u3)
c3 = Conv2D(64, (3, 3), activation='elu', padding='same') (c3)

u4 = Conv2DTranspose(32, (2, 2), strides=(2, 2), padding='same') (c3)
c4 = Conv2D(32, (3, 3), activation='elu', padding='same') (u4)
c4 = Conv2D(32, (3, 3), activation='elu', padding='same') (c4)

u5 = Conv2DTranspose(16, (2, 2), strides=(2, 2), padding='same') (c4)
c5 = Conv2D(16, (3, 3), activation='elu', padding='same') (u5)
c5 = Conv2D(16, (3, 3), activation='elu', padding='same') (c5)

# u6 = Conv2DTranspose(8, (2, 2), strides=(2, 2), padding='same') (c5)
# c6 = Conv2D(8, (3, 3), activation='elu', padding='same') (u6)
# c6 = Conv2D(8, (3, 3), activation='elu', padding='same') (c6)

outputs = Conv2D(5, (1, 1), activation='softmax') (c5)

model_xcep = Model(inputs=model_X.input, outputs=[outputs])
```

In [88]:
```
model_xcep.summary()
```

```
Model: "model_2"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_3 (InputLayer)           [(None, 256, 1600,   0           []
                                3)]

 block1_conv1 (Conv2D)          (None, 127, 799, 32  864         ['input_3[0][0]']
                                )

 block1_conv1_bn (BatchNormaliz  (None, 127, 799, 32  128        ['block1_conv1[0][0]']
 ation)                         )

 block1_conv1_act (Activation)  (None, 127, 799, 32  0           ['block1_conv1_bn[0][0]']
                                )

 block1_conv2 (Conv2D)          (None, 125, 797, 64  18432       ['block1_conv1_act[0][0]']
                                )

 block1_conv2_bn (BatchNormaliz  (None, 125, 797, 64  256        ['block1_conv2[0][0]']
 ation)                         )

 block1_conv2_act (Activation)  (None, 125, 797, 64  0           ['block1_conv2_bn[0][0]']
                                )

 block2_sepconv1 (SeparableConv  (None, 125, 797, 12  8768       ['block1_conv2_act[0][0]']
 2D)                            8)

 block2_sepconv1_bn (BatchNorma  (None, 125, 797, 12  512        ['block2_sepconv1[0][0]']
 lization)                      8)

 block2_sepconv2_act (Activatio  (None, 125, 797, 12  0          ['block2_sepconv1_bn[0][0]']
 n)                             8)

 block2_sepconv2 (SeparableConv  (None, 125, 797, 12  17536      ['block2_sepconv2_act[0][0]']
 2D)                            8)

 block2_sepconv2_bn (BatchNorma  (None, 125, 797, 12  512        ['block2_sepconv2[0][0]']
 lization)                      8)

 conv2d_30 (Conv2D)             (None, 63, 399, 128  8192        ['block1_conv2_act[0][0]']
                                )

 block2_pool (MaxPooling2D)     (None, 63, 399, 128  0           ['block2_sepconv2_bn[0][0]']
                                )

 batch_normalization_8 (BatchNo  (None, 63, 399, 128  512        ['conv2d_30[0][0]']
 rmalization)                   )

 add_24 (Add)                   (None, 63, 399, 128  0           ['block2_pool[0][0]',
                                )                                 'batch_normalization_8[0][0]']
```

```
block3_sepconv1_act (Activatio  (None, 63, 399, 128  0          ['add_24[0][0]']
n)                             )

block3_sepconv1 (SeparableConv  (None, 63, 399, 256  33920      ['block3_sepconv1_act[0][0]']
2D)                            )

block3_sepconv1_bn (BatchNorma  (None, 63, 399, 256  1024       ['block3_sepconv1[0][0]']
lization)                      )

block3_sepconv2_act (Activatio  (None, 63, 399, 256  0          ['block3_sepconv1_bn[0][0]']
n)                             )

block3_sepconv2 (SeparableConv  (None, 63, 399, 256  67840      ['block3_sepconv2_act[0][0]']
2D)                            )

block3_sepconv2_bn (BatchNorma  (None, 63, 399, 256  1024       ['block3_sepconv2[0][0]']
lization)                      )

conv2d_31 (Conv2D)             (None, 32, 200, 256  32768       ['add_24[0][0]']
                               )

block3_pool (MaxPooling2D)     (None, 32, 200, 256  0           ['block3_sepconv2_bn[0][0]']
                               )

batch_normalization_9 (BatchNo  (None, 32, 200, 256  1024       ['conv2d_31[0][0]']
rmalization)                   )

add_25 (Add)                   (None, 32, 200, 256  0           ['block3_pool[0][0]',
                               )                                 'batch_normalization_9[0][0]']

block4_sepconv1_act (Activatio  (None, 32, 200, 256  0          ['add_25[0][0]']
n)                             )

block4_sepconv1 (SeparableConv  (None, 32, 200, 728  188672     ['block4_sepconv1_act[0][0]']
2D)                            )

block4_sepconv1_bn (BatchNorma  (None, 32, 200, 728  2912       ['block4_sepconv1[0][0]']
lization)                      )

block4_sepconv2_act (Activatio  (None, 32, 200, 728  0          ['block4_sepconv1_bn[0][0]']
n)                             )

block4_sepconv2 (SeparableConv  (None, 32, 200, 728  536536     ['block4_sepconv2_act[0][0]']
2D)                            )

block4_sepconv2_bn (BatchNorma  (None, 32, 200, 728  2912       ['block4_sepconv2[0][0]']
lization)                      )

conv2d_32 (Conv2D)             (None, 16, 100, 728  186368      ['add_25[0][0]']
                               )

block4_pool (MaxPooling2D)     (None, 16, 100, 728  0           ['block4_sepconv2_bn[0][0]']
```

```
                                )

batch_normalization_10 (BatchN  (None, 16, 100, 728  2912       ['conv2d_32[0][0]']
ormalization)                   )

add_26 (Add)                    (None, 16, 100, 728  0          ['block4_pool[0][0]',
                                )                                'batch_normalization_10[0][0]']

block5_sepconv1_act (Activatio  (None, 16, 100, 728  0          ['add_26[0][0]']
n)                              )

block5_sepconv1 (SeparableConv  (None, 16, 100, 728  536536     ['block5_sepconv1_act[0][0]']
2D)                             )

block5_sepconv1_bn (BatchNorma  (None, 16, 100, 728  2912       ['block5_sepconv1[0][0]']
lization)                       )

block5_sepconv2_act (Activatio  (None, 16, 100, 728  0          ['block5_sepconv1_bn[0][0]']
n)                              )

block5_sepconv2 (SeparableConv  (None, 16, 100, 728  536536     ['block5_sepconv2_act[0][0]']
2D)                             )

block5_sepconv2_bn (BatchNorma  (None, 16, 100, 728  2912       ['block5_sepconv2[0][0]']
lization)                       )

block5_sepconv3_act (Activatio  (None, 16, 100, 728  0          ['block5_sepconv2_bn[0][0]']
n)                              )

block5_sepconv3 (SeparableConv  (None, 16, 100, 728  536536     ['block5_sepconv3_act[0][0]']
2D)                             )

block5_sepconv3_bn (BatchNorma  (None, 16, 100, 728  2912       ['block5_sepconv3[0][0]']
lization)                       )

add_27 (Add)                    (None, 16, 100, 728  0          ['block5_sepconv3_bn[0][0]',
                                )                                'add_26[0][0]']

block6_sepconv1_act (Activatio  (None, 16, 100, 728  0          ['add_27[0][0]']
n)                              )

block6_sepconv1 (SeparableConv  (None, 16, 100, 728  536536     ['block6_sepconv1_act[0][0]']
2D)                             )

block6_sepconv1_bn (BatchNorma  (None, 16, 100, 728  2912       ['block6_sepconv1[0][0]']
lization)                       )

block6_sepconv2_act (Activatio  (None, 16, 100, 728  0          ['block6_sepconv1_bn[0][0]']
n)                              )

block6_sepconv2 (SeparableConv  (None, 16, 100, 728  536536     ['block6_sepconv2_act[0][0]']
2D)                             )
```

```
block6_sepconv2_bn (BatchNorma  (None, 16, 100, 728  2912       ['block6_sepconv2[0][0]']
lization)                       )

block6_sepconv3_act (Activatio  (None, 16, 100, 728  0          ['block6_sepconv2_bn[0][0]']
n)                              )

block6_sepconv3 (SeparableConv  (None, 16, 100, 728  536536     ['block6_sepconv3_act[0][0]']
2D)                             )

block6_sepconv3_bn (BatchNorma  (None, 16, 100, 728  2912       ['block6_sepconv3[0][0]']
lization)                       )

add_28 (Add)                    (None, 16, 100, 728  0          ['block6_sepconv3_bn[0][0]',
                                )                                'add_27[0][0]']

block7_sepconv1_act (Activatio  (None, 16, 100, 728  0          ['add_28[0][0]']
n)                              )

block7_sepconv1 (SeparableConv  (None, 16, 100, 728  536536     ['block7_sepconv1_act[0][0]']
2D)                             )

block7_sepconv1_bn (BatchNorma  (None, 16, 100, 728  2912       ['block7_sepconv1[0][0]']
lization)                       )

block7_sepconv2_act (Activatio  (None, 16, 100, 728  0          ['block7_sepconv1_bn[0][0]']
n)                              )

block7_sepconv2 (SeparableConv  (None, 16, 100, 728  536536     ['block7_sepconv2_act[0][0]']
2D)                             )

block7_sepconv2_bn (BatchNorma  (None, 16, 100, 728  2912       ['block7_sepconv2[0][0]']
lization)                       )

block7_sepconv3_act (Activatio  (None, 16, 100, 728  0          ['block7_sepconv2_bn[0][0]']
n)                              )

block7_sepconv3 (SeparableConv  (None, 16, 100, 728  536536     ['block7_sepconv3_act[0][0]']
2D)                             )

block7_sepconv3_bn (BatchNorma  (None, 16, 100, 728  2912       ['block7_sepconv3[0][0]']
lization)                       )

add_29 (Add)                    (None, 16, 100, 728  0          ['block7_sepconv3_bn[0][0]',
                                )                                'add_28[0][0]']

block8_sepconv1_act (Activatio  (None, 16, 100, 728  0          ['add_29[0][0]']
n)                              )

block8_sepconv1 (SeparableConv  (None, 16, 100, 728  536536     ['block8_sepconv1_act[0][0]']
2D)                             )
```

```
block8_sepconv1_bn (BatchNorma  (None, 16, 100, 728  2912       ['block8_sepconv1[0][0]']
lization)                       )

block8_sepconv2_act (Activatio  (None, 16, 100, 728  0          ['block8_sepconv1_bn[0][0]']
n)                              )

block8_sepconv2 (SeparableConv  (None, 16, 100, 728  536536     ['block8_sepconv2_act[0][0]']
2D)                             )

block8_sepconv2_bn (BatchNorma  (None, 16, 100, 728  2912       ['block8_sepconv2[0][0]']
lization)                       )

block8_sepconv3_act (Activatio  (None, 16, 100, 728  0          ['block8_sepconv2_bn[0][0]']
n)                              )

block8_sepconv3 (SeparableConv  (None, 16, 100, 728  536536     ['block8_sepconv3_act[0][0]']
2D)                             )

block8_sepconv3_bn (BatchNorma  (None, 16, 100, 728  2912       ['block8_sepconv3[0][0]']
lization)                       )

add_30 (Add)                    (None, 16, 100, 728  0          ['block8_sepconv3_bn[0][0]',
                                )                                'add_29[0][0]']

block9_sepconv1_act (Activatio  (None, 16, 100, 728  0          ['add_30[0][0]']
n)                              )

block9_sepconv1 (SeparableConv  (None, 16, 100, 728  536536     ['block9_sepconv1_act[0][0]']
2D)                             )

block9_sepconv1_bn (BatchNorma  (None, 16, 100, 728  2912       ['block9_sepconv1[0][0]']
lization)                       )

block9_sepconv2_act (Activatio  (None, 16, 100, 728  0          ['block9_sepconv1_bn[0][0]']
n)                              )

block9_sepconv2 (SeparableConv  (None, 16, 100, 728  536536     ['block9_sepconv2_act[0][0]']
2D)                             )

block9_sepconv2_bn (BatchNorma  (None, 16, 100, 728  2912       ['block9_sepconv2[0][0]']
lization)                       )

block9_sepconv3_act (Activatio  (None, 16, 100, 728  0          ['block9_sepconv2_bn[0][0]']
n)                              )

block9_sepconv3 (SeparableConv  (None, 16, 100, 728  536536     ['block9_sepconv3_act[0][0]']
2D)                             )

block9_sepconv3_bn (BatchNorma  (None, 16, 100, 728  2912       ['block9_sepconv3[0][0]']
lization)                       )

add_31 (Add)                    (None, 16, 100, 728  0          ['block9_sepconv3_bn[0][0]',
```

```
                                )                                 'add_30[0][0]']

block10_sepconv1_act (Activati  (None, 16, 100, 728  0          ['add_31[0][0]']
on)                             )

block10_sepconv1 (SeparableCon  (None, 16, 100, 728  536536     ['block10_sepconv1_act[0][0]']
v2D)                            )

block10_sepconv1_bn (BatchNorm  (None, 16, 100, 728  2912       ['block10_sepconv1[0][0]']
alization)                      )

block10_sepconv2_act (Activati  (None, 16, 100, 728  0          ['block10_sepconv1_bn[0][0]']
on)                             )

block10_sepconv2 (SeparableCon  (None, 16, 100, 728  536536     ['block10_sepconv2_act[0][0]']
v2D)                            )

block10_sepconv2_bn (BatchNorm  (None, 16, 100, 728  2912       ['block10_sepconv2[0][0]']
alization)                      )

block10_sepconv3_act (Activati  (None, 16, 100, 728  0          ['block10_sepconv2_bn[0][0]']
on)                             )

block10_sepconv3 (SeparableCon  (None, 16, 100, 728  536536     ['block10_sepconv3_act[0][0]']
v2D)                            )

block10_sepconv3_bn (BatchNorm  (None, 16, 100, 728  2912       ['block10_sepconv3[0][0]']
alization)                      )

add_32 (Add)                    (None, 16, 100, 728  0          ['block10_sepconv3_bn[0][0]',
                                )                                 'add_31[0][0]']

block11_sepconv1_act (Activati  (None, 16, 100, 728  0          ['add_32[0][0]']
on)                             )

block11_sepconv1 (SeparableCon  (None, 16, 100, 728  536536     ['block11_sepconv1_act[0][0]']
v2D)                            )

block11_sepconv1_bn (BatchNorm  (None, 16, 100, 728  2912       ['block11_sepconv1[0][0]']
alization)                      )

block11_sepconv2_act (Activati  (None, 16, 100, 728  0          ['block11_sepconv1_bn[0][0]']
on)                             )

block11_sepconv2 (SeparableCon  (None, 16, 100, 728  536536     ['block11_sepconv2_act[0][0]']
v2D)                            )

block11_sepconv2_bn (BatchNorm  (None, 16, 100, 728  2912       ['block11_sepconv2[0][0]']
alization)                      )

block11_sepconv3_act (Activati  (None, 16, 100, 728  0          ['block11_sepconv2_bn[0][0]']
on)                             )
```

```
block11_sepconv3 (SeparableCon  (None, 16, 100, 728  536536     ['block11_sepconv3_act[0][0]']
v2D)                           )

block11_sepconv3_bn (BatchNorm  (None, 16, 100, 728  2912       ['block11_sepconv3[0][0]']
alization)                     )

add_33 (Add)                   (None, 16, 100, 728  0           ['block11_sepconv3_bn[0][0]',
                               )                                 'add_32[0][0]']

block12_sepconv1_act (Activati  (None, 16, 100, 728  0          ['add_33[0][0]']
on)                            )

block12_sepconv1 (SeparableCon  (None, 16, 100, 728  536536     ['block12_sepconv1_act[0][0]']
v2D)                           )

block12_sepconv1_bn (BatchNorm  (None, 16, 100, 728  2912       ['block12_sepconv1[0][0]']
alization)                     )

block12_sepconv2_act (Activati  (None, 16, 100, 728  0          ['block12_sepconv1_bn[0][0]']
on)                            )

block12_sepconv2 (SeparableCon  (None, 16, 100, 728  536536     ['block12_sepconv2_act[0][0]']
v2D)                           )

block12_sepconv2_bn (BatchNorm  (None, 16, 100, 728  2912       ['block12_sepconv2[0][0]']
alization)                     )

block12_sepconv3_act (Activati  (None, 16, 100, 728  0          ['block12_sepconv2_bn[0][0]']
on)                            )

block12_sepconv3 (SeparableCon  (None, 16, 100, 728  536536     ['block12_sepconv3_act[0][0]']
v2D)                           )

block12_sepconv3_bn (BatchNorm  (None, 16, 100, 728  2912       ['block12_sepconv3[0][0]']
alization)                     )

add_34 (Add)                   (None, 16, 100, 728  0           ['block12_sepconv3_bn[0][0]',
                               )                                 'add_33[0][0]']

block13_sepconv1_act (Activati  (None, 16, 100, 728  0          ['add_34[0][0]']
on)                            )

block13_sepconv1 (SeparableCon  (None, 16, 100, 728  536536     ['block13_sepconv1_act[0][0]']
v2D)                           )

block13_sepconv1_bn (BatchNorm  (None, 16, 100, 728  2912       ['block13_sepconv1[0][0]']
alization)                     )

block13_sepconv2_act (Activati  (None, 16, 100, 728  0          ['block13_sepconv1_bn[0][0]']
on)                            )
```

```
block13_sepconv2 (SeparableCon  (None, 16, 100, 102  752024     ['block13_sepconv2_act[0][0]']
v2D)                           4)

block13_sepconv2_bn (BatchNorm  (None, 16, 100, 102  4096       ['block13_sepconv2[0][0]']
alization)                     4)

conv2d_33 (Conv2D)             (None, 8, 50, 1024)  745472     ['add_34[0][0]']

block13_pool (MaxPooling2D)    (None, 8, 50, 1024)  0          ['block13_sepconv2_bn[0][0]']

batch_normalization_11 (BatchN  (None, 8, 50, 1024)  4096       ['conv2d_33[0][0]']
ormalization)

add_35 (Add)                   (None, 8, 50, 1024)  0          ['block13_pool[0][0]',
                                                                'batch_normalization_11[0][0]']

block14_sepconv1 (SeparableCon  (None, 8, 50, 1536)  1582080    ['add_35[0][0]']
v2D)

block14_sepconv1_bn (BatchNorm  (None, 8, 50, 1536)  6144       ['block14_sepconv1[0][0]']
alization)

block14_sepconv1_act (Activati  (None, 8, 50, 1536)  0          ['block14_sepconv1_bn[0][0]']
on)

block14_sepconv2 (SeparableCon  (None, 8, 50, 2048)  3159552    ['block14_sepconv1_act[0][0]']
v2D)

block14_sepconv2_bn (BatchNorm  (None, 8, 50, 2048)  8192       ['block14_sepconv2[0][0]']
alization)

block14_sepconv2_act (Activati  (None, 8, 50, 2048)  0          ['block14_sepconv2_bn[0][0]']
on)

conv2d_transpose_10 (Conv2DTra  (None, 16, 100, 512  4194816    ['block14_sepconv2_act[0][0]']
nspose)                        )

conv2d_34 (Conv2D)             (None, 16, 100, 512  2359808    ['conv2d_transpose_10[0][0]']
                               )

conv2d_35 (Conv2D)             (None, 16, 100, 512  2359808    ['conv2d_34[0][0]']
                               )

conv2d_transpose_11 (Conv2DTra  (None, 32, 200, 128  262272     ['conv2d_35[0][0]']
nspose)                        )

conv2d_36 (Conv2D)             (None, 32, 200, 128  147584     ['conv2d_transpose_11[0][0]']
                               )

conv2d_37 (Conv2D)             (None, 32, 200, 128  147584     ['conv2d_36[0][0]']
                               )
```

```
conv2d_transpose_12 (Conv2DTra  (None, 64, 400, 64)  32832      ['conv2d_37[0][0]']
nspose)

conv2d_38 (Conv2D)             (None, 64, 400, 64)  36928      ['conv2d_transpose_12[0][0]']

conv2d_39 (Conv2D)             (None, 64, 400, 64)  36928      ['conv2d_38[0][0]']

conv2d_transpose_13 (Conv2DTra  (None, 128, 800, 32  8224       ['conv2d_39[0][0]']
nspose)                        )

conv2d_40 (Conv2D)             (None, 128, 800, 32  9248       ['conv2d_transpose_13[0][0]']
                               )

conv2d_41 (Conv2D)             (None, 128, 800, 32  9248       ['conv2d_40[0][0]']
                               )

conv2d_transpose_14 (Conv2DTra  (None, 256, 1600, 1  2064       ['conv2d_41[0][0]']
nspose)                        6)

conv2d_42 (Conv2D)             (None, 256, 1600, 1  2320       ['conv2d_transpose_14[0][0]']
                               6)

conv2d_43 (Conv2D)             (None, 256, 1600, 1  2320       ['conv2d_42[0][0]']
                               6)

conv2d_44 (Conv2D)             (None, 256, 1600, 5  85         ['conv2d_43[0][0]']
                               )

==================================================================================================
Total params: 30,473,549
Trainable params: 9,612,069
Non-trainable params: 20,861,480
__________________________________________________________________________________________________
```

In [89]:

```
import tensorflow as tf
from tensorflow.keras import callbacks
optim = tf.keras.optimizers.Adam(learning_rate= 0.001)
focal_loss = sm.losses.cce_dice_loss
model_xcep.compile(optim, focal_loss, metrics=[iou_score, "accuracy"])
callbacks = [callbacks.ModelCheckpoint('./best_model_xcep.h5', save_weights_only = True, save_best_only = True, \
                                       mode = 'max', monitor = 'val_iou_score', verbose = 1),
    callbacks.ReduceLROnPlateau(monitor = 'val_iou_score', patience = 3, mode = 'max', verbose = 1,min_lr=0.0001)
]
```

In [62]:
```
BATCH_SIZE=10
train_dataloader = Dataloder(train_dataset, batch_size=BATCH_SIZE, shuffle=True)
test_dataloader = Dataloder(test_dataset, batch_size=BATCH_SIZE, shuffle=True)

assert train_dataloader[0][0].shape == (BATCH_SIZE, 256, 1600, 3)
assert train_dataloader[0][1].shape == (BATCH_SIZE, 256, 1600, 5)

print(train_dataloader[0][0].shape)
print(train_dataloader[0][1].shape)
print(len(train_dataloader))
print(len(test_dataloader))
type(train_dataset[0])
```

```
(10, 256, 1600, 3)
(10, 256, 1600, 5)
496
149
```

Out[62]:
```
tuple
```

In [90]:
```
history = model_xcep.fit(train_dataloader, steps_per_epoch=(len(train_dataloader))//BATCH_SIZE, epochs=20,\
                    validation_data=test_dataloader, callbacks=callbacks)
```

```
Epoch 1/20
49/49 [==============================] - ETA: 0s - loss: 0.8858 - iou_score: 0.1885 - accuracy: 0.9108
Epoch 1: val_iou_score improved from -inf to 0.20605, saving model to ./best_model_xcep.h5
49/49 [==============================] - 146s 3s/step - loss: 0.8858 - iou_score: 0.1885 - accuracy: 0.9108 - val_loss: 0.8447 - val_iou_score: 0.2060 - v
al_accuracy: 0.9434 - lr: 0.0010
Epoch 2/20
49/49 [==============================] - ETA: 0s - loss: 0.8309 - iou_score: 0.2114 - accuracy: 0.9235
Epoch 2: val_iou_score improved from 0.20605 to 0.21173, saving model to ./best_model_xcep.h5
49/49 [==============================] - 137s 3s/step - loss: 0.8309 - iou_score: 0.2114 - accuracy: 0.9235 - val_loss: 0.8370 - val_iou_score: 0.2117 - v
al_accuracy: 0.9434 - lr: 0.0010
Epoch 3/20
49/49 [==============================] - ETA: 0s - loss: 0.8198 - iou_score: 0.2201 - accuracy: 0.9261
Epoch 3: val_iou_score improved from 0.21173 to 0.22335, saving model to ./best_model_xcep.h5
49/49 [==============================] - 138s 3s/step - loss: 0.8198 - iou_score: 0.2201 - accuracy: 0.9261 - val_loss: 0.8080 - val_iou_score: 0.2234 - v
al_accuracy: 0.9257 - lr: 0.0010
Epoch 4/20
49/49 [==============================] - ETA: 0s - loss: 0.8151 - iou_score: 0.2221 - accuracy: 0.9294
Epoch 4: val_iou_score improved from 0.22335 to 0.22934, saving model to ./best_model_xcep.h5
49/49 [==============================] - 136s 3s/step - loss: 0.8151 - iou_score: 0.2221 - accuracy: 0.9294 - val_loss: 0.8038 - val_iou_score: 0.2293 - v
al_accuracy: 0.9277 - lr: 0.0010
Epoch 5/20
49/49 [==============================] - ETA: 0s - loss: 0.8212 - iou_score: 0.2162 - accuracy: 0.9322
Epoch 5: val_iou_score did not improve from 0.22934
49/49 [==============================] - 136s 3s/step - loss: 0.8212 - iou_score: 0.2162 - accuracy: 0.9322 - val_loss: 0.8174 - val_iou_score: 0.2178 - v
al_accuracy: 0.9110 - lr: 0.0010
Epoch 6/20
49/49 [==============================] - ETA: 0s - loss: 0.8116 - iou_score: 0.2237 - accuracy: 0.9251
Epoch 6: val_iou_score improved from 0.22934 to 0.23409, saving model to ./best_model_xcep.h5
49/49 [==============================] - 137s 3s/step - loss: 0.8116 - iou_score: 0.2237 - accuracy: 0.9251 - val_loss: 0.8110 - val_iou_score: 0.2341 - v
al_accuracy: 0.9133 - lr: 0.0010
Epoch 7/20
49/49 [==============================] - ETA: 0s - loss: 0.8079 - iou_score: 0.2282 - accuracy: 0.9240
Epoch 7: val_iou_score did not improve from 0.23409
49/49 [==============================] - 136s 3s/step - loss: 0.8079 - iou_score: 0.2282 - accuracy: 0.9240 - val_loss: 0.8041 - val_iou_score: 0.2271 - v
al_accuracy: 0.9414 - lr: 0.0010
Epoch 8/20
49/49 [==============================] - ETA: 0s - loss: 0.7972 - iou_score: 0.2312 - accuracy: 0.9343
Epoch 8: val_iou_score did not improve from 0.23409
49/49 [==============================] - 136s 3s/step - loss: 0.7972 - iou_score: 0.2312 - accuracy: 0.9343 - val_loss: 0.8040 - val_iou_score: 0.2283 - v
al_accuracy: 0.9249 - lr: 0.0010
Epoch 9/20
49/49 [==============================] - ETA: 0s - loss: 0.8126 - iou_score: 0.2236 - accuracy: 0.9213
Epoch 9: val_iou_score did not improve from 0.23409

Epoch 9: ReduceLROnPlateau reducing learning rate to 0.00010000000474974513.
49/49 [==============================] - 134s 3s/step - loss: 0.8126 - iou_score: 0.2236 - accuracy: 0.9213 - val_loss: 0.8106 - val_iou_score: 0.2261 - v
al_accuracy: 0.9247 - lr: 0.0010
Epoch 10/20
49/49 [==============================] - ETA: 0s - loss: 0.8012 - iou_score: 0.2292 - accuracy: 0.9225
Epoch 10: val_iou_score did not improve from 0.23409
49/49 [==============================] - 137s 3s/step - loss: 0.8012 - iou_score: 0.2292 - accuracy: 0.9225 - val_loss: 0.8005 - val_iou_score: 0.2303 - v
al_accuracy: 0.9331 - lr: 1.0000e-04
```

```
Epoch 11/20
49/49 [==============================] - ETA: 0s - loss: 0.7992 - iou_score: 0.2309 - accuracy: 0.9285
Epoch 11: val_iou_score did not improve from 0.23409
49/49 [==============================] - 139s 3s/step - loss: 0.7992 - iou_score: 0.2309 - accuracy: 0.9285 - val_loss: 0.7984 - val_iou_score: 0.2312 - v
al_accuracy: 0.9353 - lr: 1.0000e-04
Epoch 12/20
49/49 [==============================] - ETA: 0s - loss: 0.7969 - iou_score: 0.2336 - accuracy: 0.9259
Epoch 12: val_iou_score did not improve from 0.23409

Epoch 12: ReduceLROnPlateau reducing learning rate to 0.0001.
49/49 [==============================] - 140s 3s/step - loss: 0.7969 - iou_score: 0.2336 - accuracy: 0.9259 - val_loss: 0.7957 - val_iou_score: 0.2325 - v
al_accuracy: 0.9342 - lr: 1.0000e-04
Epoch 13/20
49/49 [==============================] - ETA: 0s - loss: 0.7883 - iou_score: 0.2374 - accuracy: 0.9320
Epoch 13: val_iou_score did not improve from 0.23409
49/49 [==============================] - 140s 3s/step - loss: 0.7883 - iou_score: 0.2374 - accuracy: 0.9320 - val_loss: 0.7951 - val_iou_score: 0.2325 - v
al_accuracy: 0.9295 - lr: 1.0000e-04
Epoch 14/20
49/49 [==============================] - ETA: 0s - loss: 0.7907 - iou_score: 0.2355 - accuracy: 0.9327
Epoch 14: val_iou_score did not improve from 0.23409
49/49 [==============================] - 141s 3s/step - loss: 0.7907 - iou_score: 0.2355 - accuracy: 0.9327 - val_loss: 0.7966 - val_iou_score: 0.2326 - v
al_accuracy: 0.9278 - lr: 1.0000e-04
Epoch 15/20
49/49 [==============================] - ETA: 0s - loss: 0.7890 - iou_score: 0.2376 - accuracy: 0.9343
Epoch 15: val_iou_score improved from 0.23409 to 0.23642, saving model to ./best_model_xcep.h5
49/49 [==============================] - 141s 3s/step - loss: 0.7890 - iou_score: 0.2376 - accuracy: 0.9343 - val_loss: 0.7923 - val_iou_score: 0.2364 - v
al_accuracy: 0.9375 - lr: 1.0000e-04
Epoch 16/20
49/49 [==============================] - ETA: 0s - loss: 0.7892 - iou_score: 0.2363 - accuracy: 0.9352
Epoch 16: val_iou_score did not improve from 0.23642
49/49 [==============================] - 140s 3s/step - loss: 0.7892 - iou_score: 0.2363 - accuracy: 0.9352 - val_loss: 0.7943 - val_iou_score: 0.2317 - v
al_accuracy: 0.9367 - lr: 1.0000e-04
Epoch 17/20
49/49 [==============================] - ETA: 0s - loss: 0.7930 - iou_score: 0.2348 - accuracy: 0.9395
Epoch 17: val_iou_score did not improve from 0.23642
49/49 [==============================] - 142s 3s/step - loss: 0.7930 - iou_score: 0.2348 - accuracy: 0.9395 - val_loss: 0.7938 - val_iou_score: 0.2361 - v
al_accuracy: 0.9358 - lr: 1.0000e-04
Epoch 18/20
49/49 [==============================] - ETA: 0s - loss: 0.7828 - iou_score: 0.2433 - accuracy: 0.9331
Epoch 18: val_iou_score did not improve from 0.23642
49/49 [==============================] - 137s 3s/step - loss: 0.7828 - iou_score: 0.2433 - accuracy: 0.9331 - val_loss: 0.7930 - val_iou_score: 0.2339 - v
al_accuracy: 0.9379 - lr: 1.0000e-04
Epoch 19/20
49/49 [==============================] - ETA: 0s - loss: 0.7947 - iou_score: 0.2343 - accuracy: 0.9308
Epoch 19: val_iou_score did not improve from 0.23642
49/49 [==============================] - 148s 3s/step - loss: 0.7947 - iou_score: 0.2343 - accuracy: 0.9308 - val_loss: 0.7924 - val_iou_score: 0.2349 - v
al_accuracy: 0.9386 - lr: 1.0000e-04
Epoch 20/20
49/49 [==============================] - ETA: 0s - loss: 0.7963 - iou_score: 0.2340 - accuracy: 0.9326
Epoch 20: val_iou_score improved from 0.23642 to 0.23653, saving model to ./best_model_xcep.h5
49/49 [==============================] - 147s 3s/step - loss: 0.7963 - iou_score: 0.2340 - accuracy: 0.9326 - val_loss: 0.7909 - val_iou_score: 0.2365 - v
al_accuracy: 0.9405 - lr: 1.0000e-04
```

Summary

In [4]:
```
from prettytable import PrettyTable
table = PrettyTable()
table.title = " Summary "
table.field_names = ['Model','No. of epochs', 'Validation loss', 'Validation IOU Score']

table.add_row(["Unet_with_resnet50","70",'0.52',"0.45"])
table.add_row(["VGG16","60", '0.58' ,"0.39"])
table.add_row(["Xception","20",'0.79',"0.24"])

print(table)
```

```
+-------------------------------------------------------------------------+
|                                 Summary                                 |
+--------------------+--------------+-----------------+----------------------+
|       Model        | No. of epochs | Validation loss | Validation IOU Score |
+--------------------+--------------+-----------------+----------------------+
| Unet_with_resnet50 |      70      |       0.52      |         0.45         |
|       VGG16        |      60      |       0.58      |         0.39         |
|      Xception      |      20      |       0.79      |         0.24         |
+--------------------+--------------+-----------------+----------------------+
```

In [ ]: