3 Evaluation And Comparison

In [1]:

```
from google.colab import files
files.upload()
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!kaggle datasets download -d mikaelstrauhs/airbus-ship-detection-train-set-70
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving kaggle.json to kaggle.json
Warning: Your Kaggle API key is readable by other users on this system! To fix this, you can run 'chmod 600 /r
oot/.kaggle/kaggle.json'
Downloading airbus-ship-detection-train-set-70.zip to /content
100% 18.5G/18.5G [05:11<00:00, 61.5MB/s]
100% 18.5G/18.5G [05:12<00:00, 63.8MB/s]
```

In [2]:

```
from zipfile import ZipFile
with ZipFile('airbus-ship-detection-train-set-70.zip', 'r') as Zip:

    print('Extracting all the files now...')
    Zip.extractall('data')

```

```
Extracting all the files now...
```

In [35]:

```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive",
force_remount=True).
```

In [36]:

```
root_path = './gdrive/My Drive/segmentation/'
```

3.1 Installing the Requred Libarary

In [37]:

```
!pip install -U segmentation-models
```

```
Requirement already up-to-date: segmentation-models in /usr/local/lib/python3.6/dist-packages (1.0.1)
Requirement already satisfied, skipping upgrade: efficientnet==1.0.0 in /usr/local/lib/python3.6/dist-packages
(from segmentation-models) (1.0.0)
Requirement already satisfied, skipping upgrade: keras-applications<=1.0.8,>=1.0.7 in /usr/local/lib/python3.
6/dist-packages (from segmentation-models) (1.0.8)
Requirement already satisfied, skipping upgrade: image-classifiers==1.0.0 in /usr/local/lib/python3.6/dist-pac
kages (from segmentation-models) (1.0.0)
Requirement already satisfied, skipping upgrade: scikit-image in /usr/local/lib/python3.6/dist-packages (from
efficientnet==1.0.0->segmentation-models) (0.16.2)
Requirement already satisfied, skipping upgrade: numpy>=1.9.1 in /usr/local/lib/python3.6/dist-packages (from
keras-applications<=1.0.8,>=1.0.7->segmentation-models) (1.19.5)
Requirement already satisfied, skipping upgrade: h5py in /usr/local/lib/python3.6/dist-packages (from keras-ap
plications<=1.0.8,>=1.0.7->segmentation-models) (2.10.0)
Requirement already satisfied, skipping upgrade: PyWavelets>=0.4.0 in /usr/local/lib/python3.6/dist-packages
(from scikit-image->efficientnet==1.0.0->segmentation-models) (1.1.1)
Requirement already satisfied, skipping upgrade: matplotlib!=3.0.0,>=2.0.0 in /usr/local/lib/python3.6/dist-pa
ckages (from scikit-image->efficientnet==1.0.0->segmentation-models) (3.2.2)
Requirement already satisfied, skipping upgrade: pillow>=4.3.0 in /usr/local/lib/python3.6/dist-packages (from
scikit-image->efficientnet==1.0.0->segmentation-models) (7.0.0)
Requirement already satisfied, skipping upgrade: imageio>=2.3.0 in /usr/local/lib/python3.6/dist-packages (fro
m scikit-image->efficientnet==1.0.0->segmentation-models) (2.4.1)
Requirement already satisfied, skipping upgrade: scipy>=0.19.0 in /usr/local/lib/python3.6/dist-packages (from
scikit-image->efficientnet==1.0.0->segmentation-models) (1.4.1)
Requirement already satisfied, skipping upgrade: networkx>=2.0 in /usr/local/lib/python3.6/dist-packages (from
scikit-image->efficientnet==1.0.0->segmentation-models) (2.5)
Requirement already satisfied, skipping upgrade: six in /usr/local/lib/python3.6/dist-packages (from h5py->ker
as-applications<=1.0.8,>=1.0.7->segmentation-models) (1.15.0)
Requirement already satisfied, skipping upgrade: cycler>=0.10 in /usr/local/lib/python3.6/dist-packages (from
matplotlib!=3.0.0,>=2.0.0->scikit-image->efficientnet==1.0.0->segmentation-models) (0.10.0)
Requirement already satisfied, skipping upgrade: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/py
thon3.6/dist-packages (from matplotlib!=3.0.0,>=2.0.0->scikit-image->efficientnet==1.0.0->segmentation-models)
(2.4.7)
Requirement already satisfied, skipping upgrade: kiwisolver>=1.0.1 in /usr/local/lib/python3.6/dist-packages
(from matplotlib!=3.0.0,>=2.0.0->scikit-image->efficientnet==1.0.0->segmentation-models) (1.3.1)
Requirement already satisfied, skipping upgrade: python-dateutil>=2.1 in /usr/local/lib/python3.6/dist-package
s (from matplotlib!=3.0.0,>=2.0.0->scikit-image->efficientnet==1.0.0->segmentation-models) (2.8.1)
Requirement already satisfied, skipping upgrade: decorator>=4.3.0 in /usr/local/lib/python3.6/dist-packages (f
rom networkx>=2.0->scikit-image->efficientnet==1.0.0->segmentation-models) (4.4.2)
```

In [38]:
```
!pip install keras==2.4.0
```

```
Requirement already satisfied: keras==2.4.0 in /usr/local/lib/python3.6/dist-packages (2.4.0)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.6/dist-packages (from keras==2.4.0) (3.13)
Requirement already satisfied: scipy>=0.14 in /usr/local/lib/python3.6/dist-packages (from keras==2.4.0) (1.4.
1)
Requirement already satisfied: h5py in /usr/local/lib/python3.6/dist-packages (from keras==2.4.0) (2.10.0)
Requirement already satisfied: tensorflow>=2.2.0 in /usr/local/lib/python3.6/dist-packages (from keras==2.4.0)
(2.4.0)
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.6/dist-packages (from keras==2.4.0) (1.1
9.5)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from h5py->keras==2.4.0) (1.15.
0)
Requirement already satisfied: keras-preprocessing~=1.1.2 in /usr/local/lib/python3.6/dist-packages (from tens
orflow>=2.2.0->keras==2.4.0) (1.1.2)
Requirement already satisfied: wheel~=0.35 in /usr/local/lib/python3.6/dist-packages (from tensorflow>=2.2.0->
keras==2.4.0) (0.36.2)
Requirement already satisfied: wrapt~=1.12.1 in /usr/local/lib/python3.6/dist-packages (from tensorflow>=2.2.0
->keras==2.4.0) (1.12.1)
Requirement already satisfied: tensorflow-estimator<2.5.0,>=2.4.0rc0 in /usr/local/lib/python3.6/dist-packages
(from tensorflow>=2.2.0->keras==2.4.0) (2.4.0)
Requirement already satisfied: termcolor~=1.1.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow>=2.
2.0->keras==2.4.0) (1.1.0)
Requirement already satisfied: absl-py~=0.10 in /usr/local/lib/python3.6/dist-packages (from tensorflow>=2.2.0
->keras==2.4.0) (0.10.0)
Requirement already satisfied: opt-einsum~=3.3.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow>=
2.2.0->keras==2.4.0) (3.3.0)
Requirement already satisfied: grpcio~=1.32.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow>=2.2.
0->keras==2.4.0) (1.32.0)
Requirement already satisfied: google-pasta~=0.2 in /usr/local/lib/python3.6/dist-packages (from tensorflow>=
2.2.0->keras==2.4.0) (0.2.0)
Requirement already satisfied: gast==0.3.3 in /usr/local/lib/python3.6/dist-packages (from tensorflow>=2.2.0->
keras==2.4.0) (0.3.3)
Requirement already satisfied: tensorboard~=2.4 in /usr/local/lib/python3.6/dist-packages (from tensorflow>=2.
2.0->keras==2.4.0) (2.4.0)
Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.6/dist-packages (from tensorflow>=2.
2.0->keras==2.4.0) (3.12.4)
Requirement already satisfied: typing-extensions~=3.7.4 in /usr/local/lib/python3.6/dist-packages (from tensor
flow>=2.2.0->keras==2.4.0) (3.7.4.3)
Requirement already satisfied: flatbuffers~=1.12.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow>
=2.2.0->keras==2.4.0) (1.12)
Requirement already satisfied: astunparse~=1.6.3 in /usr/local/lib/python3.6/dist-packages (from tensorflow>=
2.2.0->keras==2.4.0) (1.6.3)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.6/dist-packages (from tensorboard~=2.
4->tensorflow>=2.2.0->keras==2.4.0) (3.3.3)
Requirement already satisfied: google-auth<2,>=1.6.3 in /usr/local/lib/python3.6/dist-packages (from tensorboa
rd~=2.4->tensorflow>=2.2.0->keras==2.4.0) (1.17.2)
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.6/dist-packages (from tensorboard~=
2.4->tensorflow>=2.2.0->keras==2.4.0) (1.0.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.6/dist-packages (from t
ensorboard~=2.4->tensorflow>=2.2.0->keras==2.4.0) (1.7.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.6/dist-packages (fro
m tensorboard~=2.4->tensorflow>=2.2.0->keras==2.4.0) (0.4.2)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.6/dist-packages (from tensorboard
~=2.4->tensorflow>=2.2.0->keras==2.4.0) (2.23.0)
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.6/dist-packages (from tensorboard~
=2.4->tensorflow>=2.2.0->keras==2.4.0) (51.1.1)
Requirement already satisfied: importlib-metadata; python_version < "3.8" in /usr/local/lib/python3.6/dist-pac
kages (from markdown>=2.6.8->tensorboard~=2.4->tensorflow>=2.2.0->keras==2.4.0) (3.3.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.6/dist-packages (from google-au
th<2,>=1.6.3->tensorboard~=2.4->tensorflow>=2.2.0->keras==2.4.0) (0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4; python_version >= "3" in /usr/local/lib/python3.6/dist-packages
(from google-auth<2,>=1.6.3->tensorboard~=2.4->tensorflow>=2.2.0->keras==2.4.0) (4.6)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.6/dist-packages (from google-a
uth<2,>=1.6.3->tensorboard~=2.4->tensorflow>=2.2.0->keras==2.4.0) (4.2.0)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.6/dist-packages (from google
-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.4->tensorflow>=2.2.0->keras==2.4.0) (1.3.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.6/dist-packages (from requests<3,>=2.21.
0->tensorboard~=2.4->tensorflow>=2.2.0->keras==2.4.0) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.6/dist-packag
es (from requests<3,>=2.21.0->tensorboard~=2.4->tensorflow>=2.2.0->keras==2.4.0) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-packages (from requests<3,>
=2.21.0->tensorboard~=2.4->tensorflow>=2.2.0->keras==2.4.0) (2020.12.5)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.6/dist-packages (from requests<3,>=
2.21.0->tensorboard~=2.4->tensorflow>=2.2.0->keras==2.4.0) (3.0.4)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.6/dist-packages (from importlib-metadata; p
ython_version < "3.8"->markdown>=2.6.8->tensorboard~=2.4->tensorflow>=2.2.0->keras==2.4.0) (3.4.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.6/dist-packages (from pyasn1-mod
ules>=0.2.1->google-auth<2,>=1.6.3->tensorboard~=2.4->tensorflow>=2.2.0->keras==2.4.0) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.6/dist-packages (from requests-oauthl
ib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.4->tensorflow>=2.2.0->keras==2.4.0) (3.1.0)
```

3.2 Importing the libaray

In [39]:

```
import os, sys, gc

import pandas as pd
import numpy  as np
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.pyplot as plt
from pathlib import Path
from tqdm.auto       import tqdm
from multiprocessing import Pool, cpu_count

from cv2        import resize
from skimage.io import imread     as skiImgRead
# from imgaug     import augmenters as iaa, HooksImages

from skimage.morphology import label

from sklearn.model_selection import train_test_split
import cv2
from segmentation_models import Unet
from segmentation_models import get_preprocessing
from segmentation_models.utils import set_trainable
from segmentation_models.losses import bce_jaccard_loss, bce_dice_loss
from segmentation_models.metrics import iou_score, f2_score
import tensorflow as tf
```

In [40]:

```
IMG_HW    = 768
ZOOM_HW   = 384
DIR = 'data/train_v3/train_v3/Images'

BACKBONE  = 'resnet34'

preprocess_input = get_preprocessing(BACKBONE)
```

3.3 Helper Functions

3.3.1 Helper function to visualize mask on the top of the image

In [41]:

```
def mask_overlay(image, mask, color=(0,1,0)):
    mask = np.dstack((mask, mask, mask)) * np.array(color)
    weighted_sum = cv2.addWeighted(mask, 0.5, image, 0.5, 0.,dtype=cv2.CV_32F)
    img = image.copy()
    ind = mask[:, :, 1] > 0
    img[ind] = weighted_sum[ind]
    return img
```

3.3.2 Encoding RLE(run-length-encoding)

In [42]:

```
def multi_rle_encode(img):
    labels = label(img[:, :, 0])
    return [rle_encode(labels==k) for k in np.unique(labels[labels>0])]

def rle_encode(img):
    pixels = img.T.flatten()
    pixels = np.concatenate([[0], pixels, [0]])
    runs = np.where(pixels[1:] != pixels[:-1])[0] + 1
    runs[1::2] -= runs[::2]
    return ' '.join(str(x) for x in runs)
```

3.3.3 Decoding RLE(run-length-encoding)

Explantion of deconding rle total number of pixel = 768 * 768 = 589,824 EncodedPixels = [264661 17 265429 33 266197 33...] first two element is 264661 and 17 means that from 264661 to 264661+17(264678) pixel value is 1 and same for 265429 and 33 means that from 265429 to 265429+33(265462) pixel value is 1

In [43]:

```python

def rle_decode(mask_rle, shape=(768, 768)):

    s = mask_rle.split()
    starts, lengths = [np.asarray(x, dtype=int) for x in (s[0:][::2], s[1:][::2])]
    starts -= 1
    ends = starts + lengths
    img = np.zeros(shape[0]*shape[1], dtype=np.uint8)
    for lo, hi in zip(starts, ends):
        img[lo:hi] = 1
    return img.reshape(shape).T

def masks_as_image(in_mask_list):
    all_masks = np.zeros((768, 768), dtype = np.int16)
    for mask in in_mask_list:
        if isinstance(mask, str):
            all_masks += rle_decode(mask)
    return np.expand_dims(all_masks, -1)
```

In [44]:

```python
def load_paired_test_data(df, dir_prefix, augmentation=None):
    img_id = df.index.unique()[0]

    try:
        image = preprocess_input( skiImgRead( os.path.join(dir_prefix, img_id) ) )
    except:
        image = preprocess_input( np.zeros((IMG_HW, IMG_HW, 3), dtype=np.uint8) )

    image = resize(image, (ZOOM_HW,ZOOM_HW))
    return image
```

3.4 Loading data

In [45]:

```
test_csv = pd.read_csv(os.path.join(root_path,'validation.csv'))
test_csv.head()
```

Out[45]:

	Unnamed: 0	ImageId	EncodedPixels	withShip	npixel
0	0	395e9eb84.jpg	228948 2 229716 4 230483 7 231251 9 232018 12 ...	True	1252.0
1	1	395e9eb84.jpg	268840 1 269607 3 270374 5 271141 6 271907 9 2...	True	95.0
2	2	395e9eb84.jpg	212833 1 213599 4 214365 6 215131 9 215897 11 ...	True	90.0
3	3	395e9eb84.jpg	236647 1 237414 2 238182 2 238949 3 239717 3 2...	True	49.0
4	4	395e9eb84.jpg	234272 2 235036 6 235800 10 236566 12 237334 1...	True	59.0

In [46]:

```
#define array which keep track of score
score = []
```

3.5 Prediction On Train Models

3.5.1 Load Base Line model

In [47]:

```

from keras.models import model_from_json
file_path = os.path.join(root_path,'Models/model(1).json')
json_file = open(file_path, 'r')
loaded_model_json = json_file.read()
json_file.close()
Base_line_model = model_from_json(loaded_model_json)
weight_file = os.path.join(root_path,'data/best_model(1).h5')
Base_line_model.load_weights(weight_file)
print("Loaded model from disk")
```

```
Loaded model from disk
```

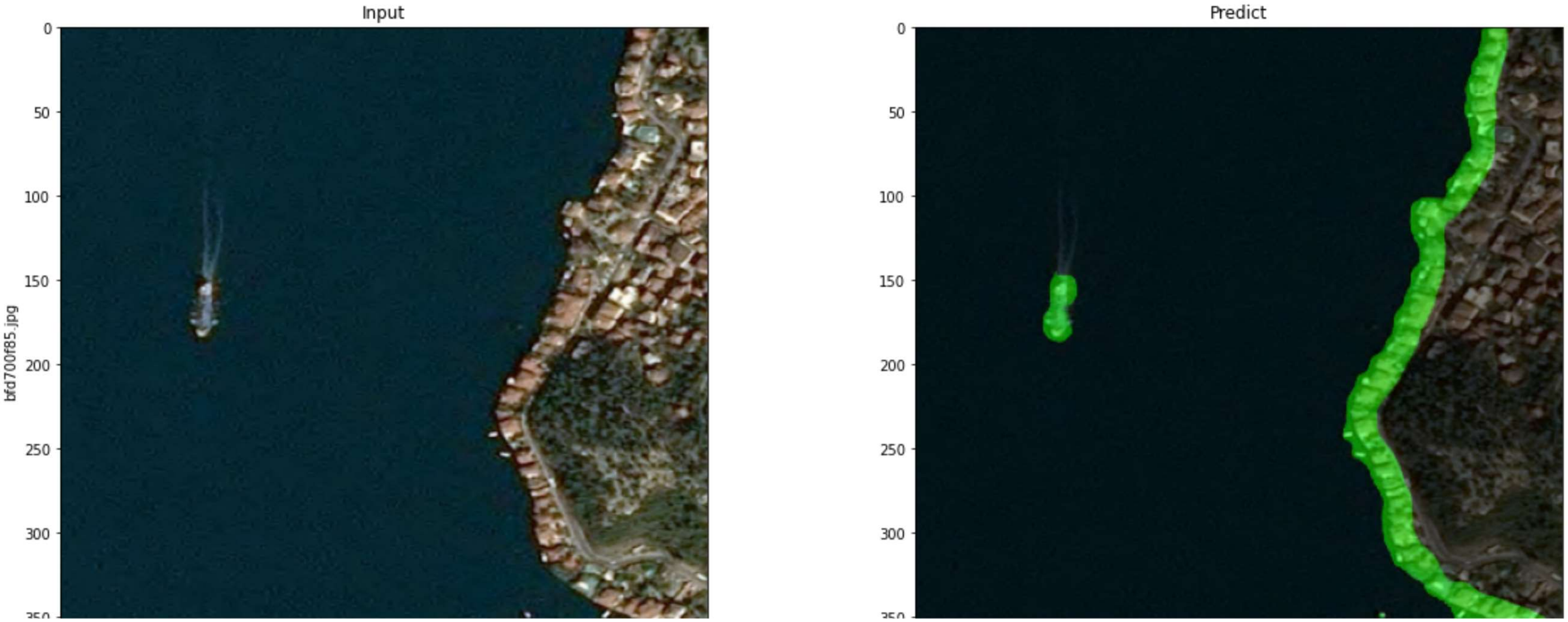
In [48]:
```
fig, axs = plt.subplots(ncols=2, nrows=10, figsize=(20,100))
axs[0,0].set_title('Input')
axs[0,1].set_title('Predict')

for i, img_id in enumerate( np.random.choice(test_csv['ImageId'], 10) ):
    img_df = test_csv.set_index('ImageId').loc[[img_id]]
    x = load_paired_test_data(img_df, DIR)

    yp = Base_line_model.predict(np.expand_dims(preprocess_input(x), axis=0) )

    axs[i,0].set_ylabel(img_id)
    axs[i,0].imshow(x)
    axs[i,1].imshow(mask_overlay(x/255 , yp[0]))

plt.show()
```



3.5.2 Unet-Resnet34 model

In [49]:

```
from keras.models import model_from_json
# load json and create model
file_path = os.path.join(root_path,'Models/model_resnet.json')
json_file = open(file_path, 'r')
loaded_model_json = json_file.read()
json_file.close()
model_resnet = model_from_json(loaded_model_json)
# load weights into new model
weight_file = os.path.join(root_path,'data/best_model_resnet.h5')
model_resnet.load_weights(weight_file)
print("Loaded model from disk")
```

```
Loaded model from disk
```

In [50]:

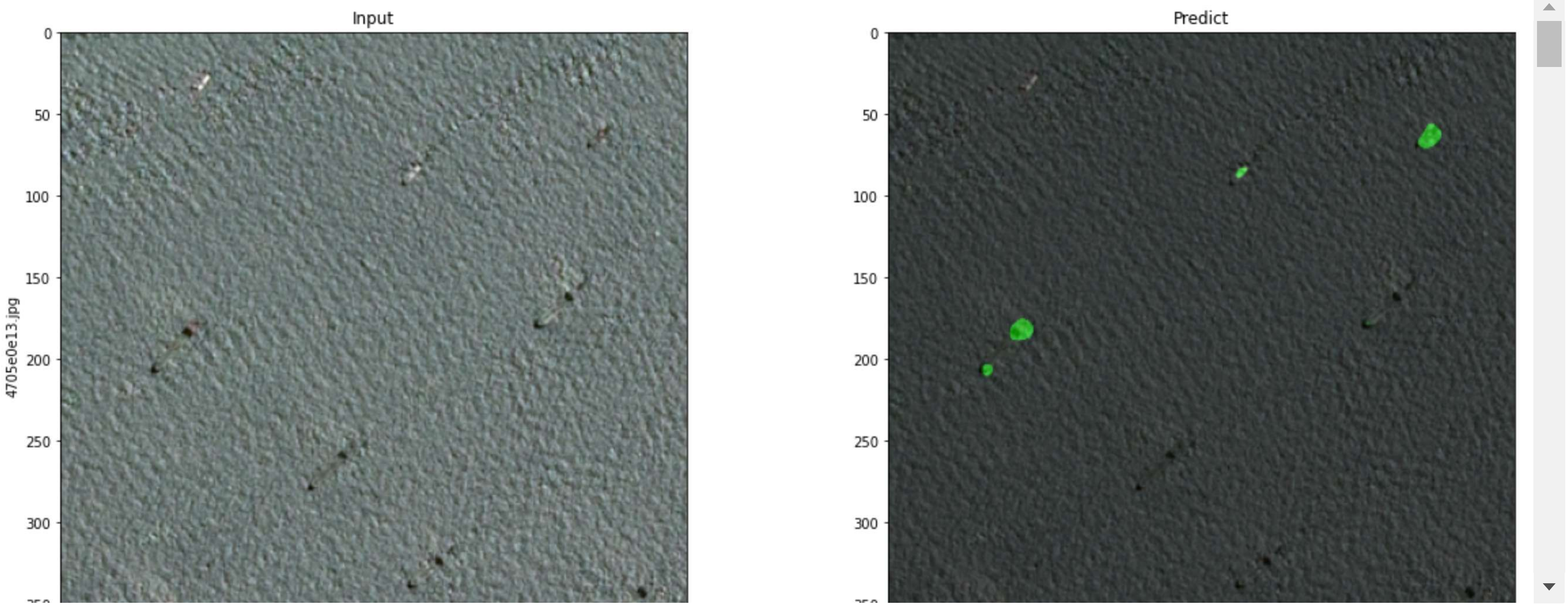
```
fig, axs = plt.subplots(ncols=2, nrows=10, figsize=(20,100))
axs[0,0].set_title('Input')
axs[0,1].set_title('Predict')

for i, img_id in enumerate( np.random.choice(test_csv['ImageId'], 10) ):

    img_df = test_csv.set_index('ImageId').loc[[img_id]]
    x = load_paired_test_data(img_df, DIR)
    yp = model_resnet.predict(np.expand_dims(preprocess_input(x), axis=0) )
    axs[i,0].set_ylabel(img_id)
    axs[i,0].imshow(x)
    axs[i,1].imshow(mask_overlay(x/255 , yp[0]))

plt.xticks([])
plt.yticks([])

plt.show()
```



3.5.3 Unet-Vgg model

In [51]:

```
from keras.models import model_from_json
# load json and create model
file_path = os.path.join(root_path,'Models/model_vgg(1).json')
json_file = open(file_path, 'r')
loaded_model_json = json_file.read()
json_file.close()
model_vgg = model_from_json(loaded_model_json)
# load weights into new model
weight_file = os.path.join(root_path,'data/best_model_vgg.h5')
model_vgg.load_weights(weight_file)
print("Loaded model from disk")
```

```
Loaded model from disk
```

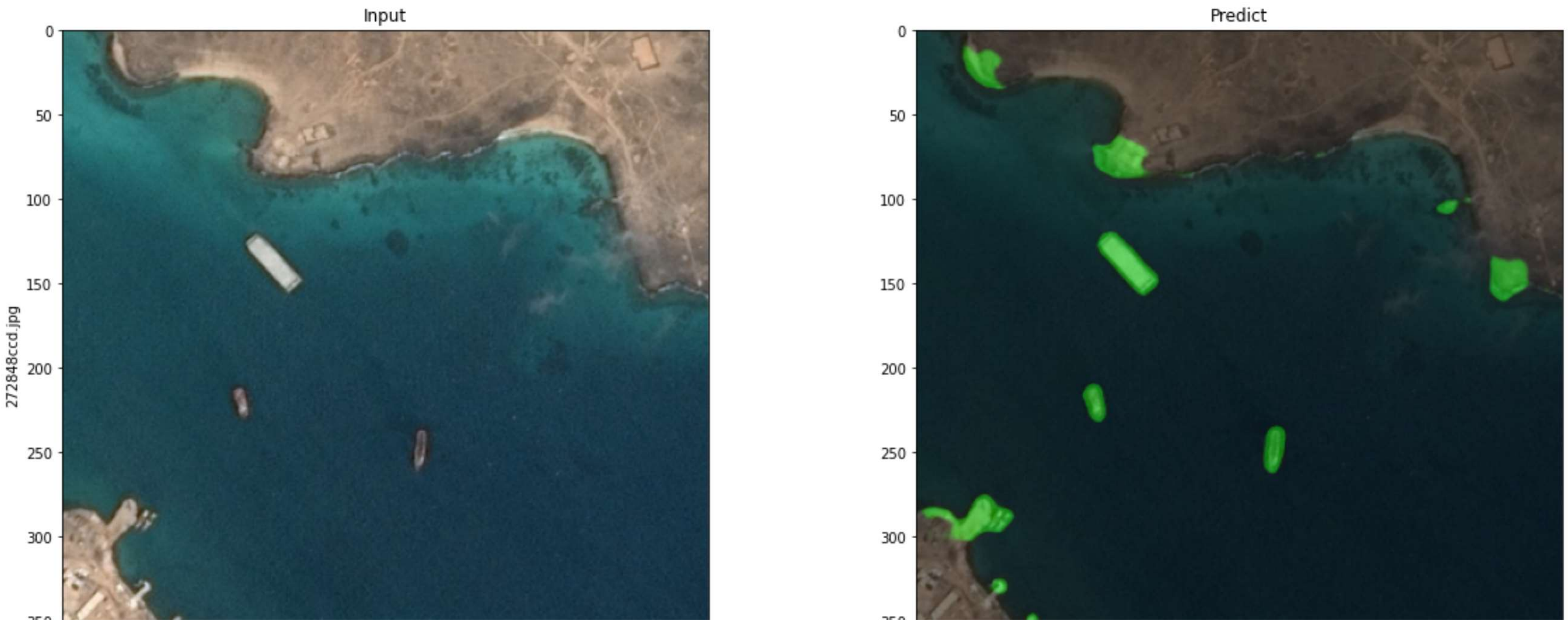
In [52]:
```
fig, axs = plt.subplots(ncols=2, nrows=10, figsize=(20,100))
axs[0,0].set_title('Input')
axs[0,1].set_title('Predict')

for i, img_id in enumerate( np.random.choice(test_csv['ImageId'], 10) ):

    img_df = test_csv.set_index('ImageId').loc[[img_id]]
    x = load_paired_test_data(img_df, DIR)
    yp = model_vgg.predict(np.expand_dims(preprocess_input(x), axis=0) )
    axs[i,0].set_ylabel(img_id)
    axs[i,0].imshow(x)
    axs[i,1].imshow(mask_overlay(x/255 , yp[0]))

plt.xticks([])
plt.yticks([])

plt.show()
```



3.5.4 FPN (Feature Pyramid Network)

In [53]:

```

from keras.models import model_from_json
# Load json and create model
file_path = os.path.join(root_path,'Models/FPN.json')
json_file = open(file_path, 'r')
loaded_model_json = json_file.read()
json_file.close()
fpn = model_from_json(loaded_model_json)
# Load weights into new model

weight_file = os.path.join(root_path,'data/best_model_fpn.h5')
fpn.load_weights(weight_file)
print("Loaded model from disk")
```

```
Loaded model from disk
```

In [54]:

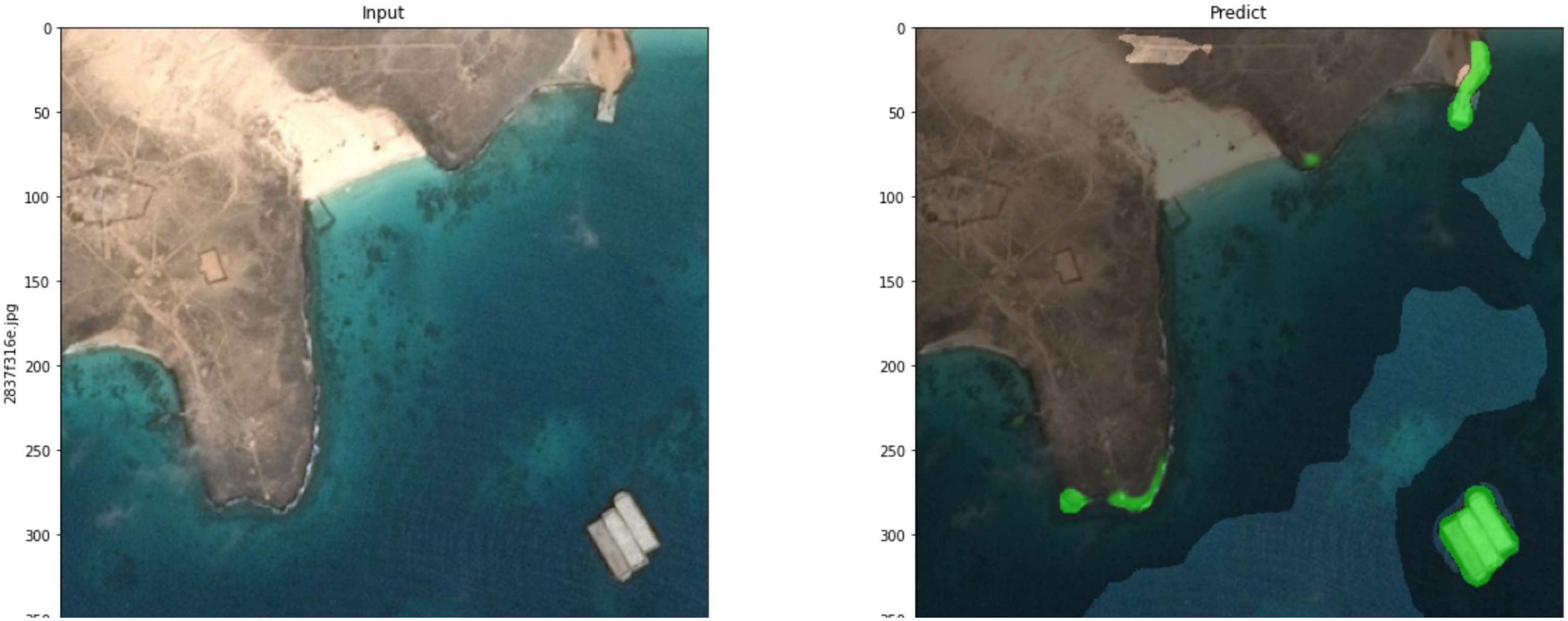
```
fig, axs = plt.subplots(ncols=2, nrows=10, figsize=(20,100))
axs[0,0].set_title('Input')
axs[0,1].set_title('Predict')

for i, img_id in enumerate( np.random.choice(test_csv['ImageId'], 10) ):

    img_df = test_csv.set_index('ImageId').loc[[img_id]]
    x = load_paired_test_data(img_df, DIR)
    yp = fpn.predict(np.expand_dims(preprocess_input(x), axis=0) )
    axs[i,0].set_ylabel(img_id)
    axs[i,0].imshow(x)
    axs[i,1].imshow(mask_overlay(x/255 , yp[0]))

plt.xticks([])
plt.yticks([])

plt.show()
```



In [ ]: