Oil Seepage Detection Report

Introduction

Studies of oil seepage or slicks in the ocean environment with synthetic aperture radar (SAR) have found that one of the most complex challenges to oil spill detection is the separation of mineral oil spills from slicks that are biogenic in origin. The possible occurrence of multiple scattering mechanisms beyond Bragg scattering for the sea surface, with or without biogenic or mineral oil slicks, and even under low to moderate wind conditions, has also been a subject of debate because the measured signals from these radar-dark surfaces can be contaminated easily by noise. Therefore, it's hard to fast and accurately recognize the spilled area through traditional methods if the number of SAR images is huge.

In this project, the objective is to segment the regions containing oil seepage through deep convolutional neural network (DCNN). There are two ways to build the CNN model. One is the direct CNN method. It means we build the convnets, Conv2D layers and MaxPooling2D layers, first to filter the original data and learn the local patterns. Then the dense layer is connected to the convolution layers. The other way is to use pretrained convnet in the model.

As for the problem we confront, the number of images is 790. It is a very small dataset. The direct CNN method needs much more data than this to train the model so that we can obtain good prediction. Then the second method should be used. There are many pretrained model to use. The corresponding accuracy results for ImageNet are presented in Table 1.

Table 1 Pretrained CNN models [1]. (Time per inference step is the average of 30 batches and 10 repetitions. - CPU: AMD EPYC Processor (with IBPB) (92 core) - Ram: 1.7T - GPU: Tesla A100 - Batch size: 32)

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	0.79	0.945	22,910,480	126	109.42	8.06
VGG16	528	0.713	0.901	138,357,544	23	69.5	4.16
VGG19	549	0.713	0.9	143,667,240	26	84.75	4.38
ResNet50	98	0.749	0.921	25,636,712	-	58.2	4.55
ResNet101	171	0.764	0.928	44,707,176	-	89.59	5.19
ResNet152	232	0.766	0.931	60,419,944	-	127.43	6.54
ResNet50V2	98	0.76	0.93	25,613,800	-	45.63	4.42
ResNet101V2	171	0.772	0.938	44,675,560	-	72.73	5.43

ResNet152V2	232	0.78	0.942	60,380,648	-	107.5	6.64
InceptionV3	92	0.779	0.937	23,851,784	159	42.25	6.86
InceptionResNetV2	215	0.803	0.953	55,873,736	572	130.19	10.02
MobileNet	16	0.704	0.895	4,253,864	88	22.6	3.44
MobileNetV2	14	0.713	0.901	3,538,984	88	25.9	3.83
DenseNet121	33	0.75	0.923	8,062,504	121	77.14	5.38
DenseNet169	57	0.762	0.932	14,307,880	169	96.4	6.28
DenseNet201	80	0.773	0.936	20,242,984	201	127.24	6.67
NASNetMobile	23	0.744	0.919	5,326,716	-	27.04	6.7
NASNetLarge	343	0.825	0.96	88,949,818	-	344.51	19.96
EfficientNetB0	29	-	-	5,330,571	-	46	4.91
EfficientNetB1	31	-	-	7,856,239	-	60.2	5.55
EfficientNetB2	36	-	-	9,177,569	-	80.79	6.5
EfficientNetB3	48	-	-	12,320,535	-	139.97	8.77
EfficientNetB4	75	-	-	19,466,823	-	308.33	15.12
EfficientNetB5	118	-	-	30,562,527	-	579.18	25.29
EfficientNetB6	166	-	-	43,265,143	-	958.12	40.45
EfficientNetB7	256	-	-	66,658,687	-	1578.9	61.62

It can be found in Table 1 that the Xception model has a great potential to deal with this oil seepage problem. The Xception model is the extension of inception architecture. The architecture of this model is shown in Figure 1 [2]. The detail of realizing this model can also be learned from that study [2]. In a nutshell, the model should include entry flow, middle flow and exit flow. The model built in my study is similar to this process.

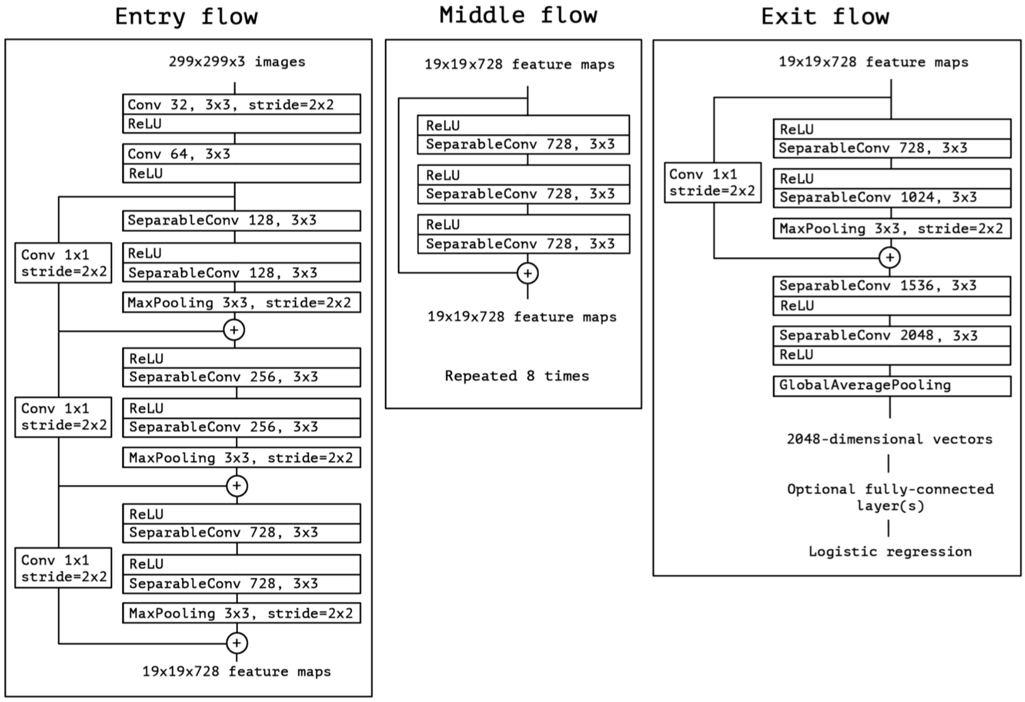


Figure 1. Xception architecture

Results

The SAR images are digitalized through **opencv-python** library. It transformed the images into the corresponding matrix: 790×256×256×3. The mask is transformed into matrix as: 790×256×256×1. The number “3” in image matrix is the color channel, and the number “1” in mask matrix is the oil seepage level. There are 8 levels (0-7). Because the main purpose of this study is to segment the oil seepage area, we can reduce this output labels into 0 and 1. The 0 represents no oil spill, and the 1 represents there is oil spill.

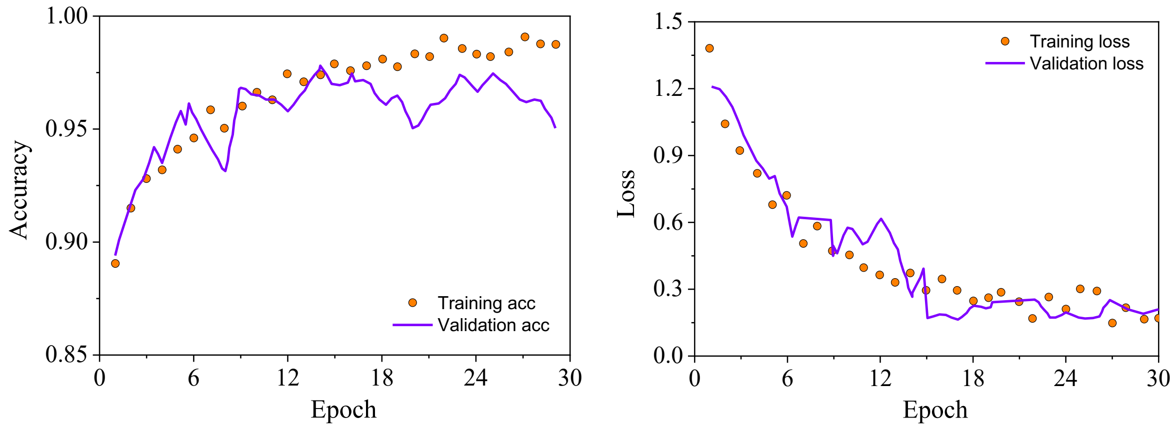
The metrics for loss function is based on the F_1Score function [3]. The F_1Score can be regarded as a harmonic mean. It is presented as:

$$F_1Score = \frac{2\sum(y_{true} \cdot y_{pred})}{\sum(y_{true} + y_{pred})} \tag{1}$$

A good loss function for Xception model based on F_1Score can be given as [4]:

$$Loss = 1 - F_1Score + binary\_crossentropy(y_{true}, y_{pred}) \tag{2}$$

The accuracy method is the default binary accuracy method in TensorFlow. The epoch is 30. The number of training data is 80% of the images. The validation data and test data are both 10% of the images respectively. It means the training image, validation image and test image are 632, 79 and 79 respectively. The results for the metrics are presented in Figure 2. It can be seen that the Xception model becomes better and better as the learning continues.



(a) (b)

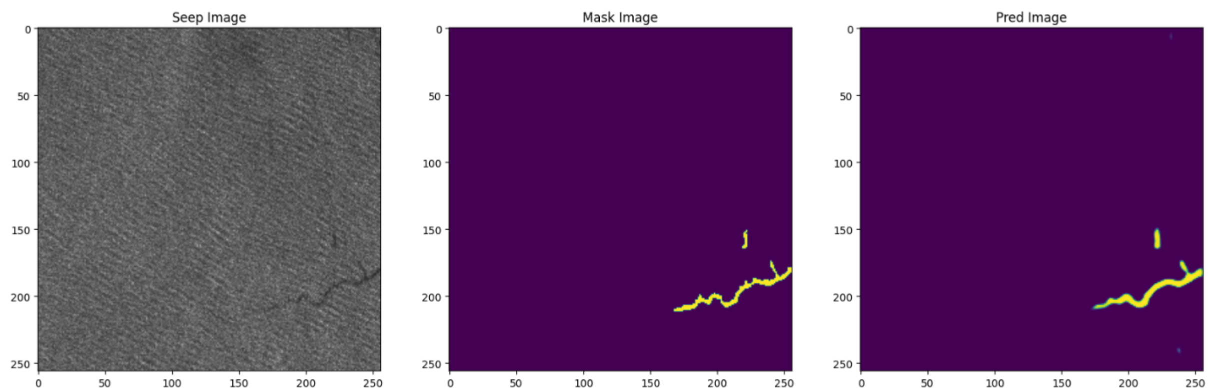
Figure 2. Training and validation results with 30 epochs. (a) accuracy; (b) loss

The segment results based on the pretrained Xception model are presented in Figure 3. It should be noted these are just some of samples to show the prediction results. More predicted images are presented in the folder attached. Using the code: model.evaluate(X_test, y_test), the results are presented below:

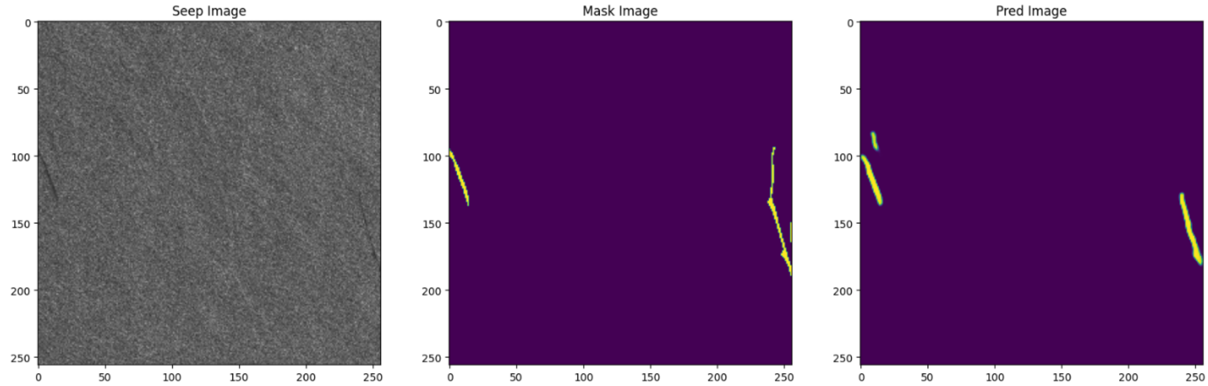
3/3 [==============================] - 5s 2s/step - Loss: 0.4117 – F_1Score: 0.6188 - acc: 0.9911

Out[37]: [0.41173055768013, 0.6188395619392395, 0.9911010265350342]

The accuracy of the model is as high as 0.99. It can also be seen that the predicted results in Figure 3 are great. The high accuracy doesn't mean that every mask image and predicted image are almost the same. The reason is that the total compared data are 790×10%×256×256×2 = 10,354,688. That means the 1% difference of data may also lead to some difference between mask image and predicted image

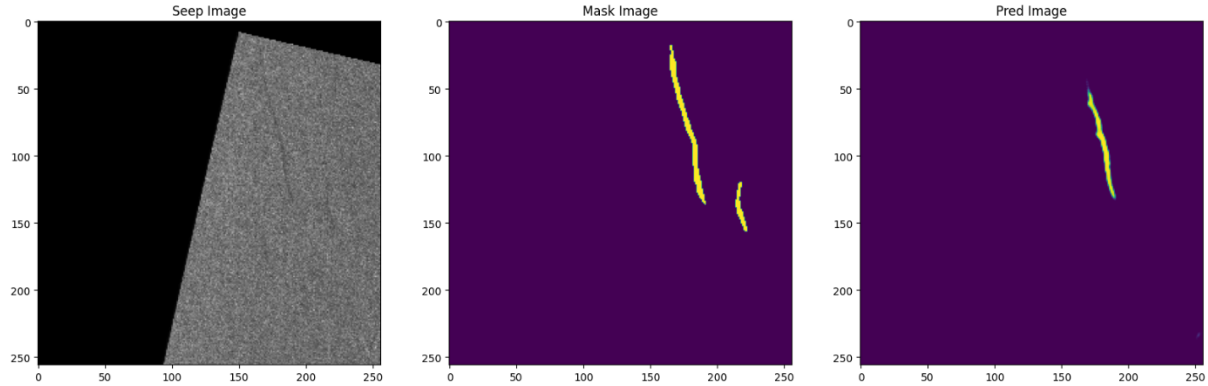


(a)



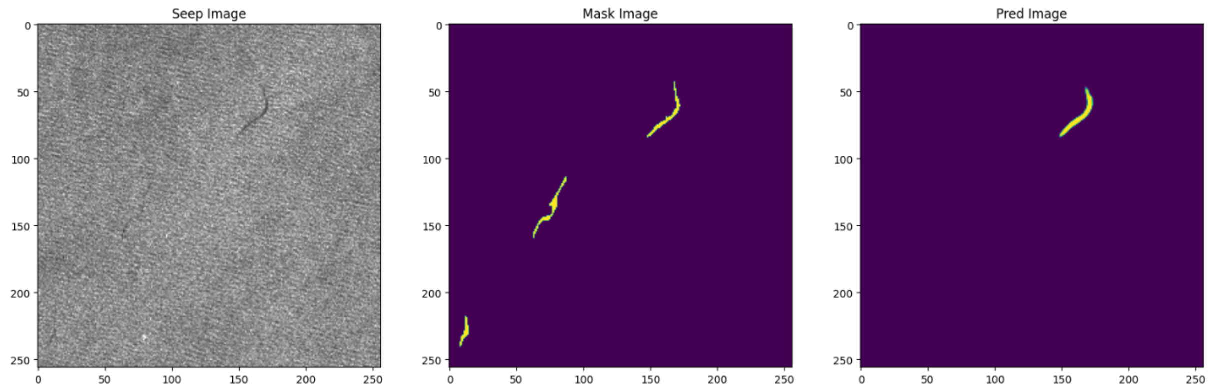
Seep Image
Mask Image
Pred Image

(b)



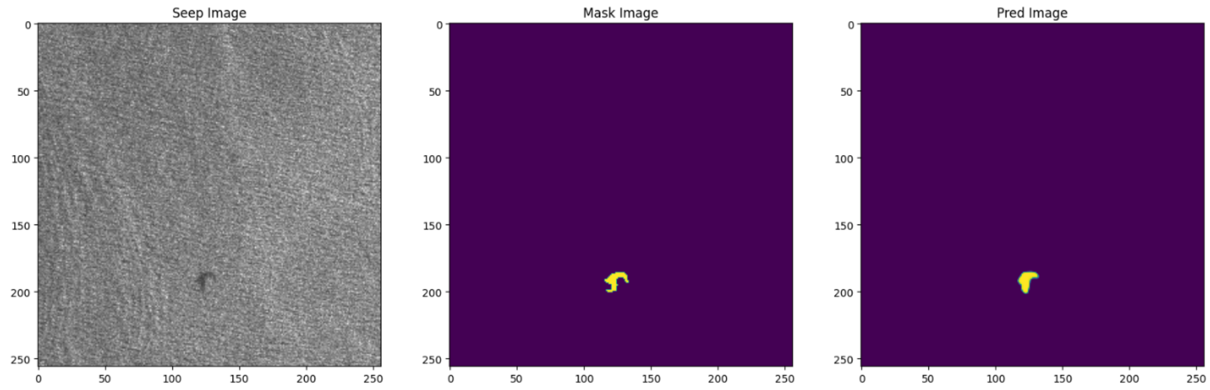
Seep Image
Mask Image
Pred Image

(c)

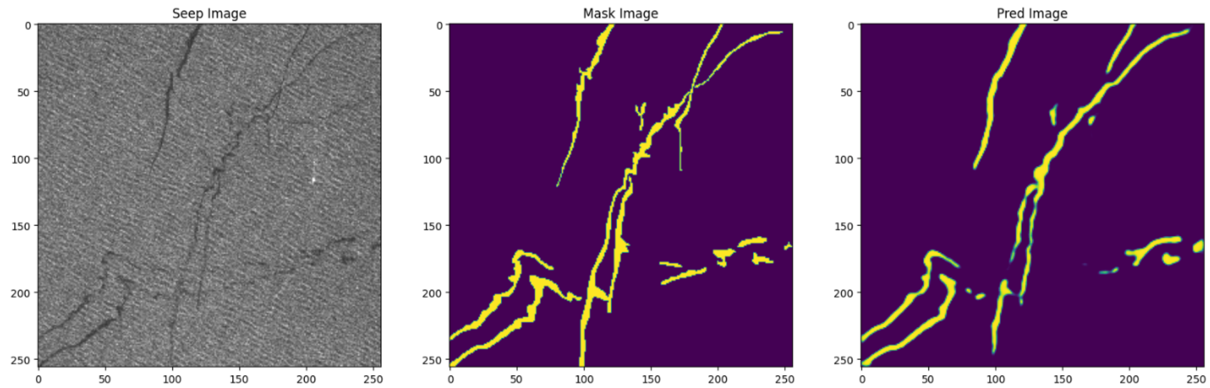


Seep Image
Mask Image
Pred Image

(d)



(e)



(f)

Figure 3. Comparison of original seep images, mask images and predicted images.

Module specification

The modules include three parts. The first one is the main module. It includes the image digitalization, metrics and the model training. The second one is the Xception model based on previous study [2]. The third part is output of the comparison results in Figure 3. TensorFlow is used to build the DCNN model. The trained model for this problem is also saved as: model-hxt(xception)-oil_seepage.

Others

There is an optional task, which requires the model to classify the seepage levels. The level is from 0 to 7. The problem changes from binary classification or image segmentation to multiclass classification. The loss function should be replaced by CategoricalCrossentropy in TensorFlow. Considering the time, this study won't involve this part. It can be presented in the future.

Reference:

[1] Simon, M., Rodner, E., & Denzler, J. (2016). Imagenet pre-trained models with batch normalization. *arXiv preprint arXiv:1612.01452*.

[2] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

[3] Tharwat, A. (2020). Classification assessment methods. Applied Computing and Informatics.

[4] De Laurentiis, L., Jones, C. E., Holt, B., Schiavon, G., & Del Frate, F. (2020). Deep Learning for Mineral and Biogenic Oil Slick Classification with Airborne Synthetic Aperture Radar Data. IEEE Transactions on Geoscience and Remote Sensing.