1 Problem Address

Shipping traffic is growing fast. More ships increase the chances of infractions at sea like environmentally devastating ship accidents, piracy, illegal fishing, drug trafficking, and illegal cargo movement. This has compelled many organizations, from environmental protection agencies to insurance companies and national government authorities, to have a closer watch over the open seas. Airbus offers comprehensive maritime monitoring services by building a meaningful solution for wide coverage, fine details, intensive monitoring, premium reactivity and interpretation response. Combining its proprietary-data with highly-trained analysts, they help to support the maritime industry to increase knowledge, anticipate threats, trigger alerts, and improve efficiency at sea.

Our project is to classify collected images whether contain any ships with a efficient model. We break down the problem by find any pixels of a image containing the ships. However, most images contain no ships and only a few of images capture ships. To handle this unbalanced dataset efficiently, we divide the task into one big binary classification task about identifying if there any ship inside of the image and another smaller segmentation task to figure out how many ships contains in a certain image.

2 Dataset Description

Training set contains 192,556 images(each has 768*768 pixels with 3 channels). Some images only contain the sea background, some images contain different number of ships. The number of ships varies from 1 to 15. Test set contains 15,606 images. The size of test set is far more less than the training set.

Folder	Type	Dimension	Size
train_v2.zip	jpg	192,556	26.3GB
test_v2.zip	jpg	15,606	2.12GB
train_ship_segmentation.csv	csv table	231725*2	43 MB

The csv file contains encoded information about location of pixels that contain ships in each pictures in the trainging set.

3 Data Pre-processing and Visulization

We decoded csv location information first by converting into image-mask matrix(768*768 pixel, 0 for no ship and 1 for pixel that contains ship). Then plotting ships count pie chart in the training set and get ideas how ships/no ships images distribute.

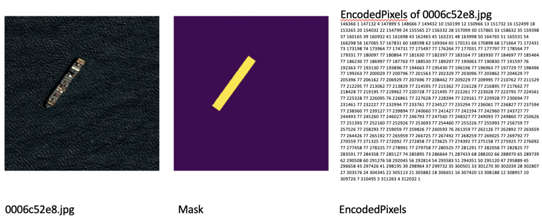

Figure 1: Images,Mask and RLE

4 Dataset Description and Data Preparation

4.1 Dataset

Training set: Training set contains 192,556 images(each has 768*768 pixels with 3 channels). Some images only contain the sea background, some images contain different number of ships. The number of ships varies from 1 to 15. From Figure 1, we can found that 78% of training images are no-ship images(which is relative

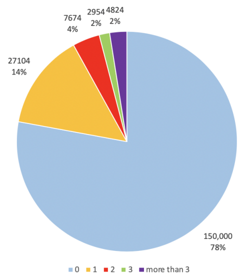

Figure 2: Training data composition

non-important for segmentation task). And only 2% of images have more than 3 ships.

Test set: Test set contains 15,606 images. The size of test set is far more less than the training set.

Segmentation: Segmentation provides the Run-Length Encoding(RLE) strings format for the training images. This csv is 231725*2, frist column is ImageId and second column is EncodedPixels. If there is no ship in one satellite image, the corresponding EncodedPixels is empty; if there are ships in one satellite image, there should be EncodedPixels coded by Run-Length Encoding format.

4.2 Run-Length Encode and Decode

Encode: convert image-mask matrix(768*768 pixels,0 or 1) to string format(RLE).
Decode: convert RLE string to image-mask matrix(768*768 pixels, 0 or 1).
Encode and decode are convertible process, and Figure 2 shows the relationship between original image, mask image and RLE string.

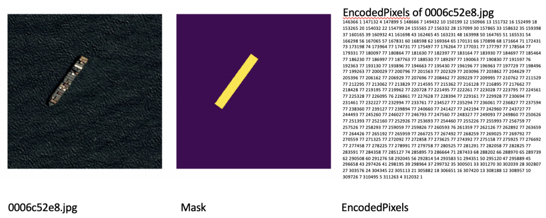

Figure 3: Images,Mask and RLE

5 Method

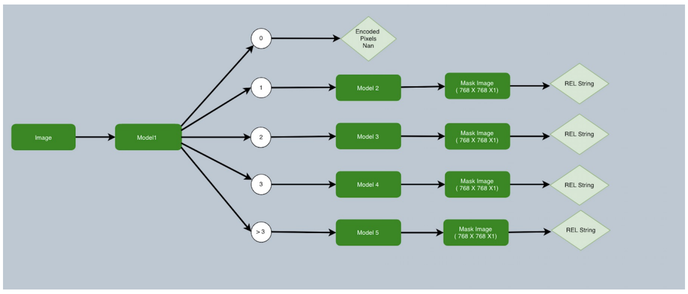

Figure 4: Stratified Modeling structure

We proposed a combined method called stratified modeling to solving this task, showed in Figure 3. We can divide the segmentation task into 2 problems:
(1) Classification problem: find how many ships in each image(5 classes).
(2) Segmentation problem: if there is any ship in the image, find the position and shape of the ships.

The reason why we want to choose this method to solve the main task is that: (a) The portion of non-ship images is too large(according to training data, around 80% of images have no ship). If we use the same segmentation model to predict all images, it will waste too much time on training and predict no-ships images. Also there will be many images has no ship, but we may predict some noisy pixels that has ship(which will increasing type II error). (b) Classification problem is much more easier than segmentation problem, after we got a higher accuracy on classification, we actually already do a great segmentation task on no-ship's test images(If the class prediction is correct, the segmentation should say all pixels in this images is 0) (c) When there are ships in an image, for model 2 to model 5, each model will be more sensitive and are more powerful to segment the corresponding number of ships.

5.1 Images Classification

From Figure 3, we used ResNet34 model as model 1 to deal with the images classification problem(5 classes responses: 0-no ship,1-one ship,2-two ships,3-three ships,4-more than three ships). The structure for ResNet34 we used is shown as Figure 4:[2]

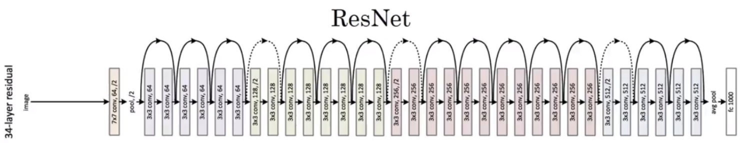

Figure 5: Images,Mask and RLE

5.2 Ship Segmentation

5.2.1 Data Augmentation

Most images are overshoot images, the orientation of the ships are multi-directional. To make our models have more generality, data augmentation is necessary for the training set. We accomplish the data segmentation through the following operationsRotation with angle between 0 and 360; Flipping; Adjust brightness of images.

5.2.2 Ship Segmentation

To solve segmentation task, we actually need to predict a class label(0 or 1, represent if this pixels is a part of a ship) for each pixels in the image. There are two popular models to solving this segmentation task, U-net and Mask R-Cnn [3][4]. By comparing this two models, we found that Mask R-cnn spent more time on training and also the score was not higer than U-net model. Therefore, we decided to use U-Net model as our model 2 to model 5(Figure 3). One attractive reason for us is that U-net model is fast, it only need to take less than one second to finish a segmentation task for a 512*512 image[3]. Figure 5 shows the architecture for U-net.[3]

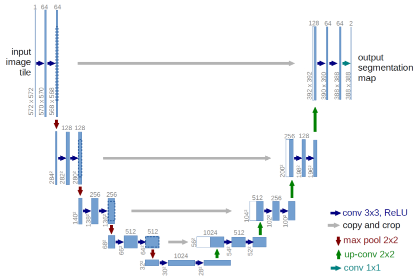

Figure 6: Images,Mask and RLE

Overall, our training and test algorithm is showed as Figure 6.

Training Steps:

Input:
All Training picture(192556 images with768*768*3)
***Preprocessing:**
We can directly get the number of ships in each training picture by summary train_ship_segmentation.csv.
***Main Model:**

Model 1: ResNet34 Model
Training data: {original Image, number of ships}

Model 2 to 5: Mask- RCNN or Unet-CNN

Partitioning the training data into 5 sets(0 ship, 1 ship, 2 ships, 3 ships, >3 ships). Using corresponding partition data to train model 2 to model5.
e.g: use {original image, mask image} in 1 ship group to training model 2 and so on.
***Argumentation:**
To improve generality of model2 to 5, we add image argumentation for images (rotation, flipping, Bighting ..)

Prediction Steps:

Input:
A new image(768*768*3)
Model 1:
Input: {768*768*3 matrix}
Output: predict the number of ships in the new image.(Integer)

If equal to 0, then return empty, **END**;

Otherwise:

If equal to 1, then this image enter model 2.

If equal to 2, then this image enter model3 and so on.

Model 2 to 5:
Input: {768*768*3 matrix}
Output: A mask image (768*768*1 pixels, each one is 0 or 1).
Encoding:
Converting mask image into required RLE format string,
Return the RLE string, **END**

Figure 7: Images,Mask and RLE

6 Results

Because of the computation and time limitation, we cannot achieve all parts of our method. We simplified our method by using model 1(ResNet model) to predict if images has ship or not(binary response). Then, use model 2(U-net model) to do the segmentation prediction for all images that we predicted have ships. We used Kaggle server with our working environment, however each time of running is limited in 6 hours. Also, we separated training data into training and validation set to test our result for both models.

First, we used complete training images(192,556 images) to train model 1 with 5 epochs, it took around 3-4 hours. Model 1 has validation accuracy around 98%. After use model 1 to predict test images, we predicted 3318 images have ships over 15,606 images. Then, for those images that we predict no ship exist, the corresponding mask just an empty mask. Next, we only need to do ship segmentation on 3,318 images.

Then we used partition training images that having ships(42,556 images) to train model 2 with data argumentation, it took around 6 hours to finished 7 epochs. Finally, we used this model to predict the ship segmentation of 3,318 test images we extract from model 1.

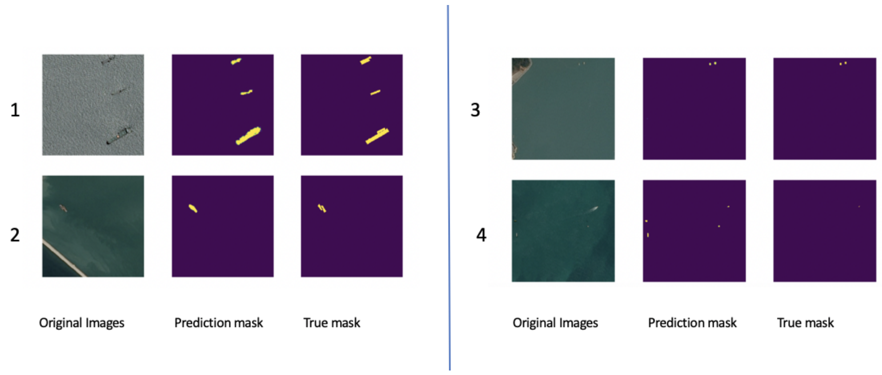

Figure 8: Image, Prediction mask and True mask

Figure 7 are four predict results we got based on model 2. In general, we can find that this model works well, however it still shows some problems. In the second example, there are 2 ships in the image which are close to each other. But our model predicted them in the same group of pixels, which will be encoded as only one ship exists. And in the fourth example, there is only one ship, but there are four ships in our predicted masks.

We believe the prediction could be better if we applied our complete method. It will be more sensitive to segment the corresponding number of ships when we actually predict the number of ships in the image correctly.

Finally, we transferred the mask images to RLE strings for each images for submission. Based on the final training model we have, we got a result of intersection over union(IoU) score with 0.833 in Kaggle, the rank is 224 in the public leader board. The 1st place in the leader board got 85.448 IoU score.

7 Discussion and Future work

The results of our model did not achieve our expectation. The main reason may be the lack of training time and we did not accomplish our complete method either.

In advance, we should working on some aspects:
(1) Implement complete method we proposed.
(2) Increase training epochs for each model.
(3) Some faster and more advanced image recognition models like Fast R-CNN, YOLO may be considered.

8 Reference

[**1**] Kaggle Website. Airbus Ship Detection. Retrived from https://www.kaggle.com/c/airbus-ship-detection
[**2**] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
[**3**] Ronneberger, O., Fischer, P., Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.
[**4**] He, K., Gkioxari, G., Dollár, P., Girshick, R. (2017, October). Mask r-cnn. In Computer Vision (ICCV), 2017 IEEE International Conference on (pp. 2980-2988). IEEE.