Airbus Ship Detection With UNet And Mask R-CNN

Ying Chi, Mike Li, Rebecca Chen, Ted Wang
University of Waterloo

Abstract

- This project is based on a Kaggle competition Airbus Ship Detection Challenge in 2018. The task is to detect if images contain ships first and then locate pixels including the ship. We use Resduial Neural Network **ResNet** to detect existence of ships. We implement and compare **UNet** and **Mask R-CNN**, which are two types CNN designed for segmenting images, to find all pixels including the ships.

Motivation

- More ships increase the chances of infractions at sea. This appeals more attention to ships on the open seas. More related analysis wanted to be done to satisfy raising concerns.
- The goal of **ship segmentation** is to simplify or change the representation of an image into something that is more meaningful and easier for following analysis.

Data Acquisition

- **Data introduction**: The dataset is from Kaggle,and it contains the Run-Length Encoded strings of ships in each image, and test images, size 768x768 pixels. Training set has 192,556 images and test set has 15,606 images.
- **Data Visualization**:

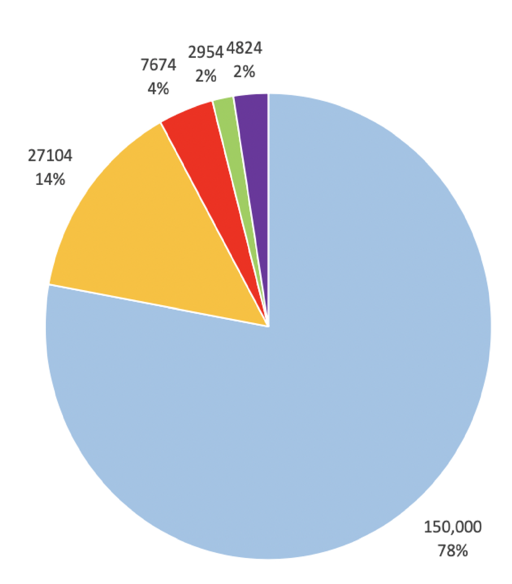

Figure 1: Training data composition

Model Architectures

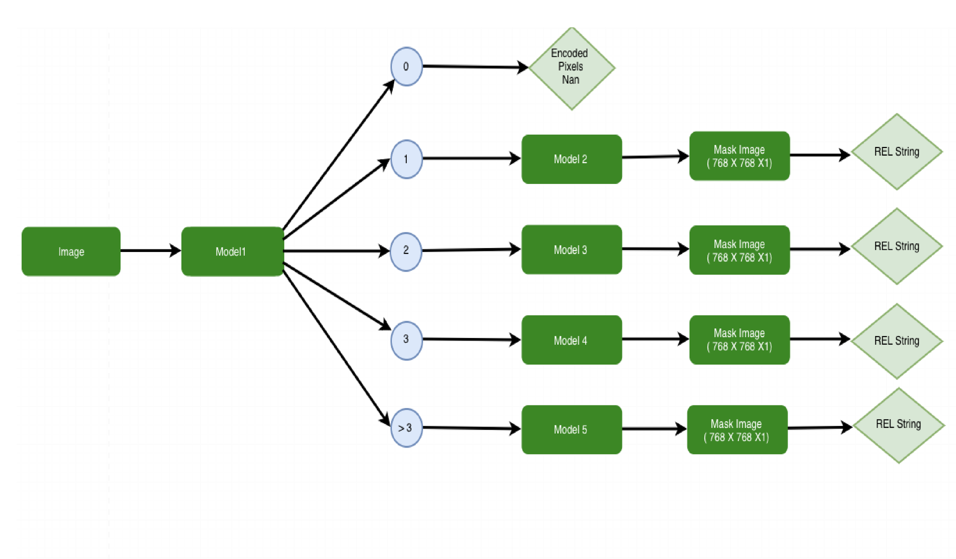

Figure 2: Designed Model Flowchat

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

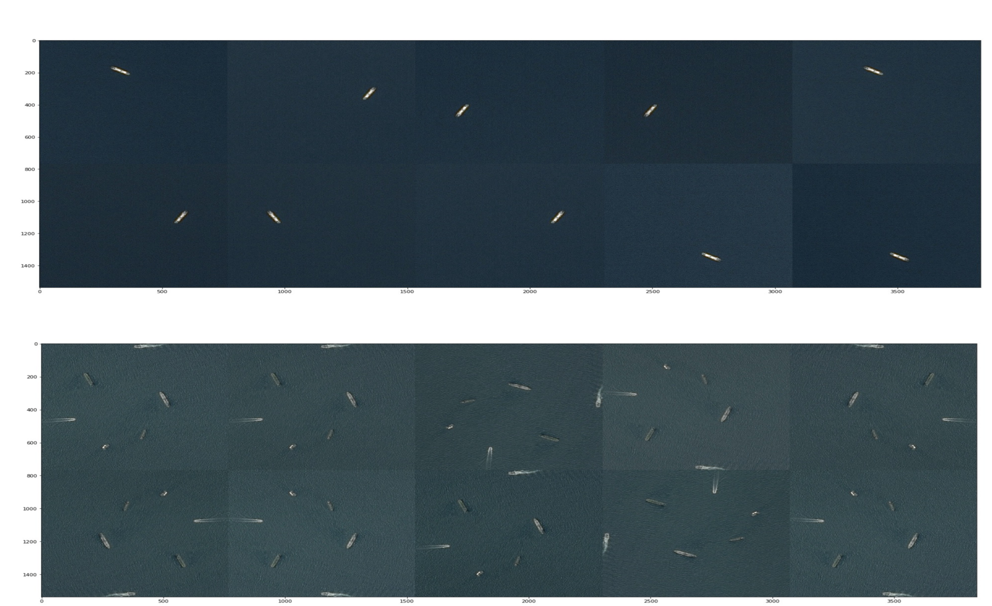

Figure 3: After Data Augmentation

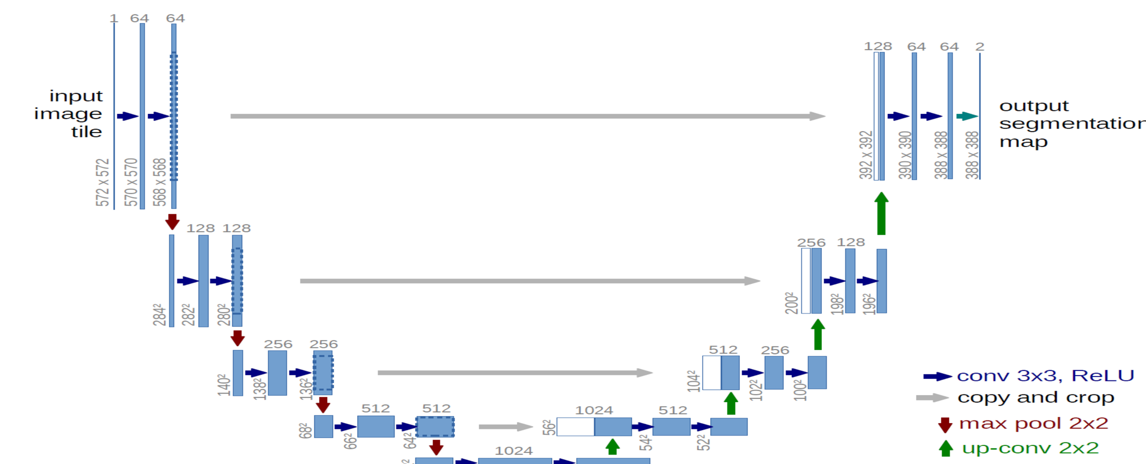

Figure 4: UNet

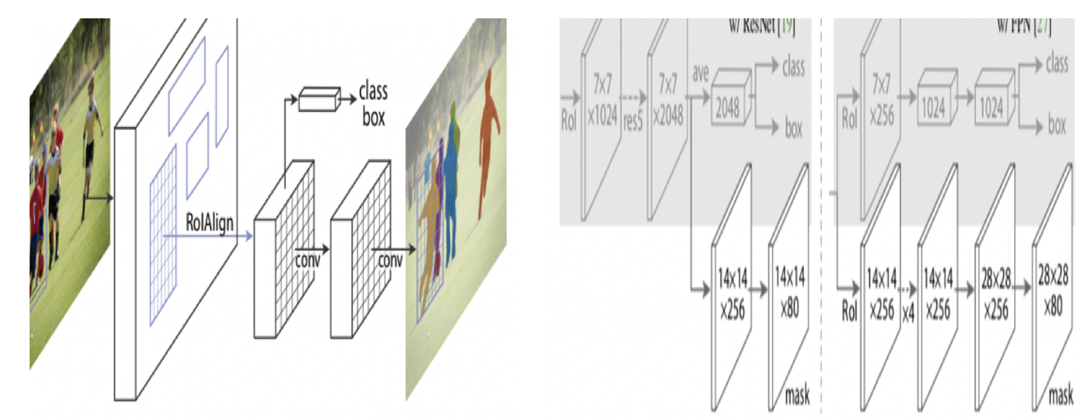

Figure 5: Mask R-CNN

Our Working Results

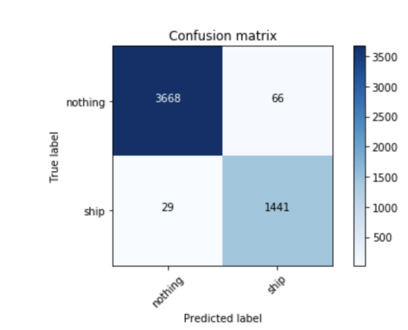

Figure 6: Confusion Matrix

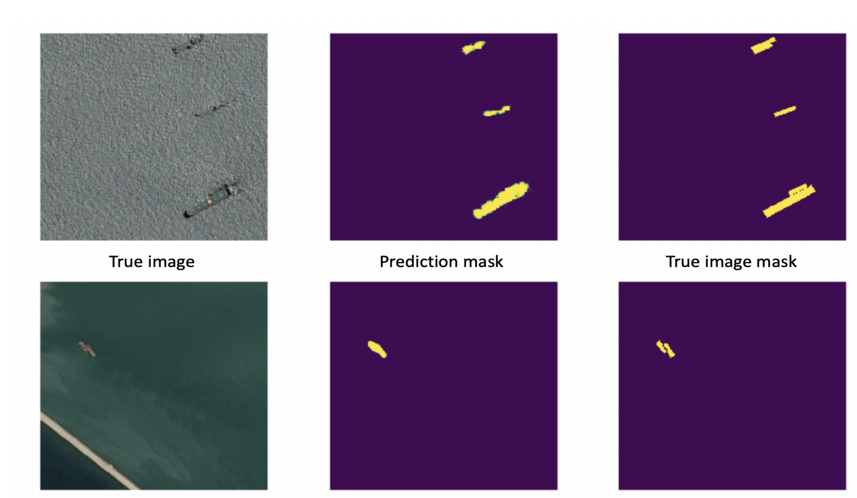

Figure 7: U-Net Training Result

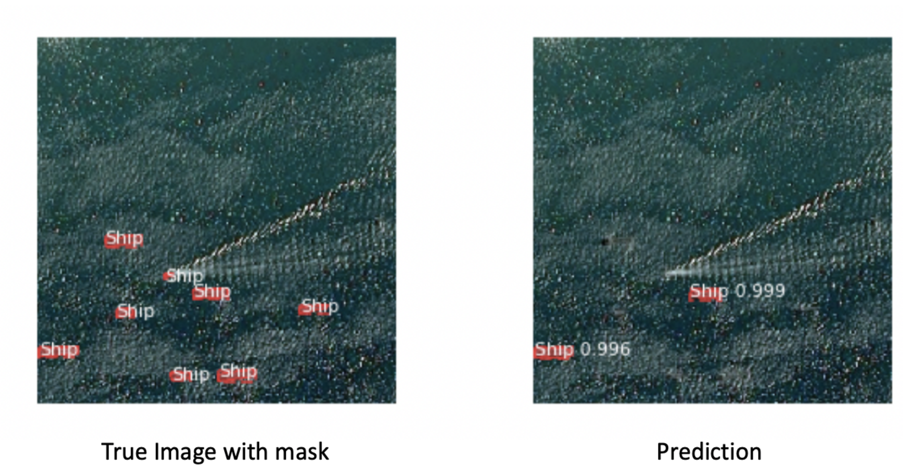

Figure 8: Mask-RCNN Training Result

Conclusion

Because of the computation and time limitation, we cannot achieve all parts of our method. We simplified method by using model1(ResNet34 model) to predict if images has ship or not(binary response). Then, use model2(U-net model) to do the segmentation prediction for all images that we predicted have ships.

Classification result: By training model1, we use batch size = 64 and 5 epoch, we can achieve an approximate converge validation accuracy. It already needs to take 3-4 hours of model1 training. Model1 gives us a validation accuracy around 98%, which means we can extract most of images into correct class.

Segmentation result:
We compared U-net model and Mask R-CNN model. The Kaggle score that UNet achieved higher than Mask R-CNN. And also, U-net is faster than Mask R-CNN. Therefore, we chose UNet34 as our model2 choice.
We only use the training images with ships to train our model2, it took around 6 hours to finished 7 epochs. Using the best model, our result score is 83.34% on Kaggle, ranking 233 out of 884.

Further works:
(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 consider.

Reference

- He, K., Gkioxari, G., DollÃąr, P. and Girshick, R. (2018). Mask R-CNN. [online] Arxiv.org. Available at: https://arxiv.org/abs/1703.06870 [Accessed 11 Dec. 2018].
- Ronneberger, O., Fischer, P. and Brox, T. (2018). U-Net: Convolutional Networks for Biomedical Image Segmentation. [online] Arxiv.org. Available at: https://arxiv.org/abs/1505.04597 [Accessed 11 Dec. 2018].