Group #13 Airbus Ship Detection Challenge

Xintong Hao, Yuchen Nie, Chunxi Wang, Zulin Liu
https://github.com/XintongHao/Machine-Learning-Project
Department of Electrical & Computer Engineering, College of Engineering, Boston University

ABSTRACT

In this project, we use multiple machine learning models to locate ships in images, and create functions to put an aligned bounding box segment around the ships we located. Many images do not contain ships, and those that do may contain multiple ships. Ships within and across images may differ in size and be located in open sea, at docks, marinas, etc.
We tried three U-net models with different hyperparameter, and a transfer learning model of pre-trained Mask R-CNN model which is for object detection. There are two metrics for performance evaluation: IoU for loss function, Binary accuracy for confidence of having ships.

PROJECT DIAGRAM

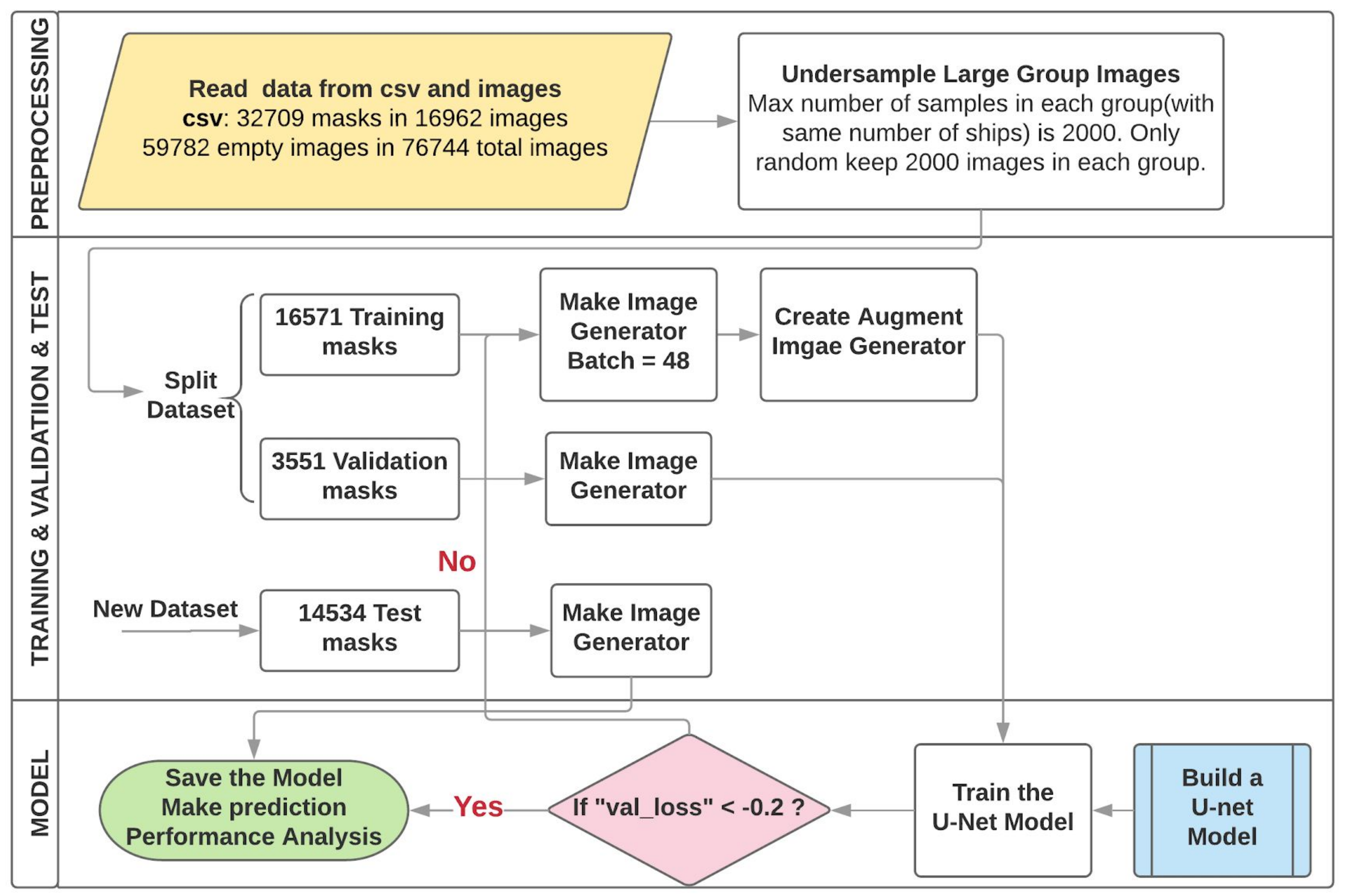

Fig 1. Project Diagram

a) Data preprocessing: read data and balance training dataset.
b) Split training dataset into 80% training set and 20% validation set, and use 14534 new images as test set.
c) Feature X: Generate input vectors with training batch $\in \{48, 96, 128\}$ and generate data augmentations.
d) Train the models using feature X and target y from training set.
e) Check validation loss using validation set. Once the loss (IoU) is lower than -0.2, finish training and save the model.

DATA PREPROCESSING

The EDA process shows that the original dataset was imbalanced, where 77.9% images were empty images (didn't contain any ships). Combined with other data features that we observed {num_ships, file_size_kb}, we undersample the majority class where we randomly select at each iteration 2000 samples from the majority class and use only those as training data, combined with all samples from the minority class.

METHODOLOGIES

I. U-net

The U-net model we used is based on standard U-net[2] shown as shown on the right, which consists of a contracting path (left side) and a expansive path (right side.) The contracting path follows the typical architecture of CNN

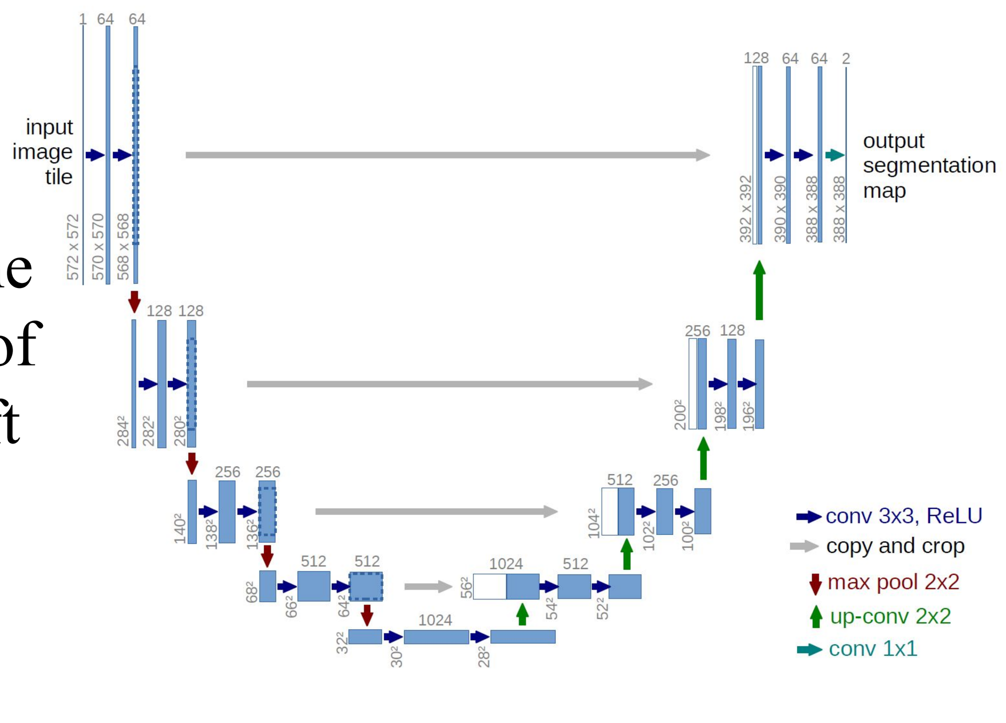

Fig 2. Architecture of standard U-net

a) Before convolutional layers, we add Gaussian noise and batch normalization on the input image.
b) Contracting path: it follows by 5 times repeated two 3x3 convolutions, and one 2x2 max-pooling operation.
c) Expansive path: it consist of 4 times repeated an upsampling of the feature map followed by 2x2 convolution, a concatenation with the correspondingly cropped feature map from the contracting path, and two 2x2 convolutions.
d) Final layer to map output vector to our desired number of class.

II. Mask R-CNN transfer learning model

Mask R-CNN (regional convolutional neural network) is a two stage framework: the first stage scans the image and generates proposals(areas likely to contain an object). And the second stage classifies the proposals and generates bounding boxes and masks. We apply this model to our data and then do the prediction and validation.

RESULTS

	U-net batch=48	U-net batch=96	U-net batch-128
IoU Loss	-0.536	-0.70	-0.670
Accuracy	99.70%	99.9%	99.74%

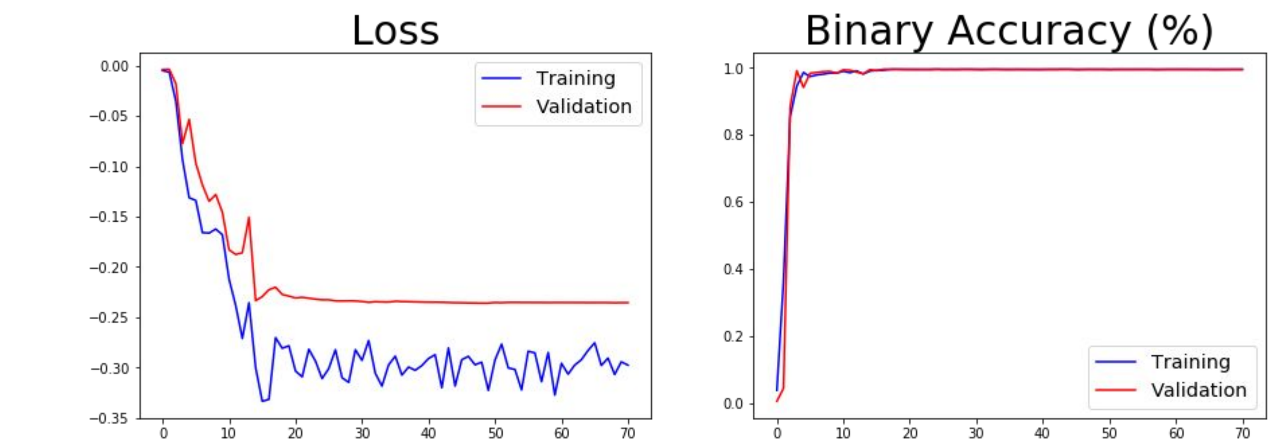

Fig 3. Loss History of U-net batch=96

Mask_RCNN	loss	MRCNN box loss	Mask loss
Train	0.9603	0.1834	0.3290
Validation	1.0662	0.1988	0.3466

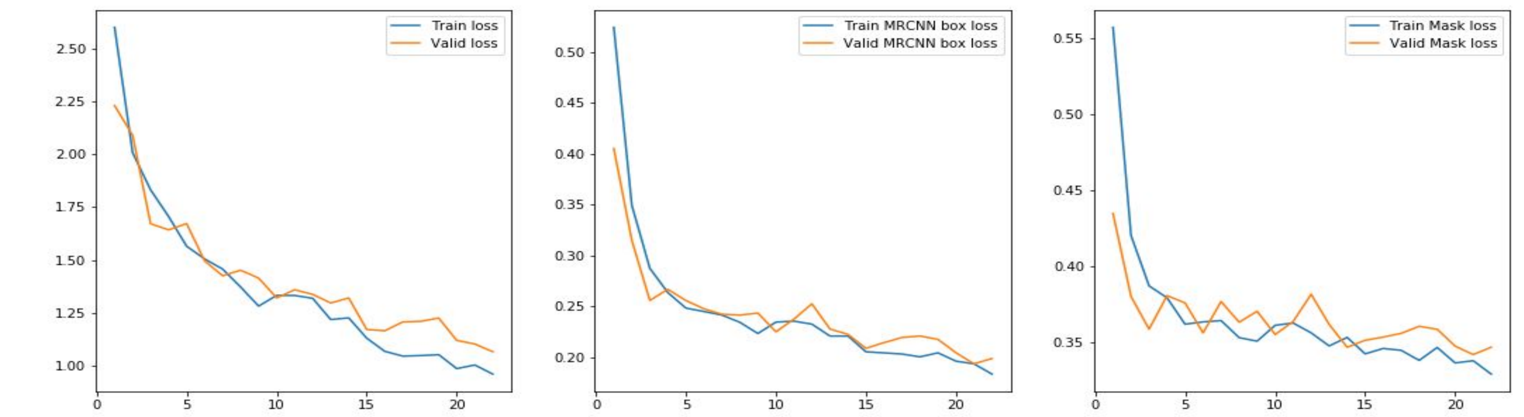

Fig 4. Loss History of Mask R-CNN model

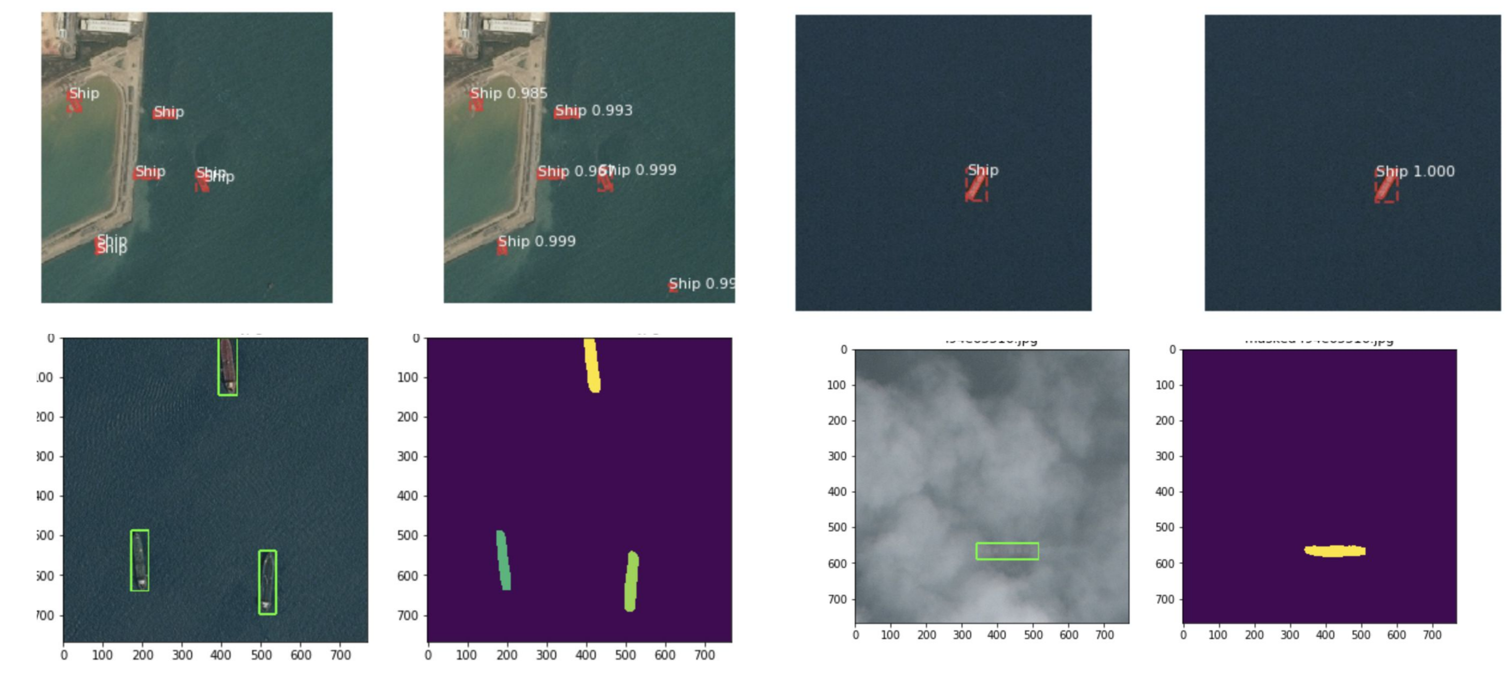

Fig 5. Prediction visualization

REFERENCES

[1] Kaggle competition: *https://www.kaggle.com/c/airbus-ship-detection*
[2] U-net: O. Ronneberger, P. Fischer, T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015
[3] Kaggle kernel:
https://www.kaggle.com/hmendonca/u-net-model-with-submission