Ec601 Final Report

1. Open source components

For Data analysis and visualization, we mainly investigate the public kernels on Kaggle chanllenge, very useful, 5/5.

For ship-existence model, we refer to keras.application documents, https://keras.io/applications/ , very well documented, 5/5. We add dense layer and flatten layer after the pretrained model.

For U-net model, referring to https://github.com/qubvel/segmentation_models/tree/master/segmentation_models/unet , very well documented, 4/5. Based on their u-net model, we combine Inception v3 from keras.application (https://keras.io/applications/) as encoder.

For Flask and web, http://flask.pocoo.org/ this one is what we base on. The official tutorial documents of flask. Very useful, 4/5.

2. Data analysis

This is a Kaggle challenge, here is the link. https://www.kaggle.com/c/airbus-ship-detection. The data size is about 30GB. The training data are masked with a separate csv file. We need to submit the same mask csv file for the test data. For more details, you could find in our GitHub.
https://github.com/helibu/Airbus-ship-detection/tree/master/Data%20EDA
Sprint 2 slides in github give a pretty detailed analysis of the data.

3. Model used

We mainly have two models. One for ship-existence detection and the other for ship location model.

For ship-existence model, we refer to keras.application documents, https://keras.io/applications/ . We use VGG16 and ResNet50 pretrained with ImageNet, add dense layer and flatten layer after the pretrained model. And combine them to an ensemble model. Ship existence model details can be found here. https://github.com/helibu/Airbus-ship-detection/tree/master/Model/ship-existence%20model. Loss function is binary_crossentropy for we only need to detect if there is a ship or not. We tried simple CNN at first, but it doesn't work well. Then we used pretrained models trained with imagenet. These models have been well trained for classification purpose, which saved us a lot of time.

For U-net model, referring to https://github.com/qubvel/segmentation_models/tree/master/segmentation_models/unet . Based on their u-net model, we combine Inception v3 from keras.application (https://keras.io/applications/) as encoder. U-net model details could be found here. https://github.com/helibu/Airbus-ship-detection/tree/master/Model/U-net%20model. We use DICE/F1 score of the intersection of predictions and ground truth based on the uneven distribution of has-ship pixels and no ship pixels.

We choose U-net because most ships on images are quite small, and we need a model that can handle high resolution information. So, we choose U-net.