# Ship Detection in Satellite Images

Deep Neural Networks with Tensorflow

Prof. Michael Grossberg

Sujoy Debnath

The document details the development of a ship detection system which detects ships in satellite images using convolutional neural network and tensorflow. It further describes the future work that can be included as part building the project for more accurate classification of ships.

# Introduction

Object detection is one of the highly researched areas in computer science. There are various kinds of object detection. Classification and localization classifies an object in an image. Object detection identifies objects in an image. Semantic segmentation identifies objects of different classes in an image. Where instance segmentation identifies objects at pixel level segmentation by applying a different color to each object in the image.

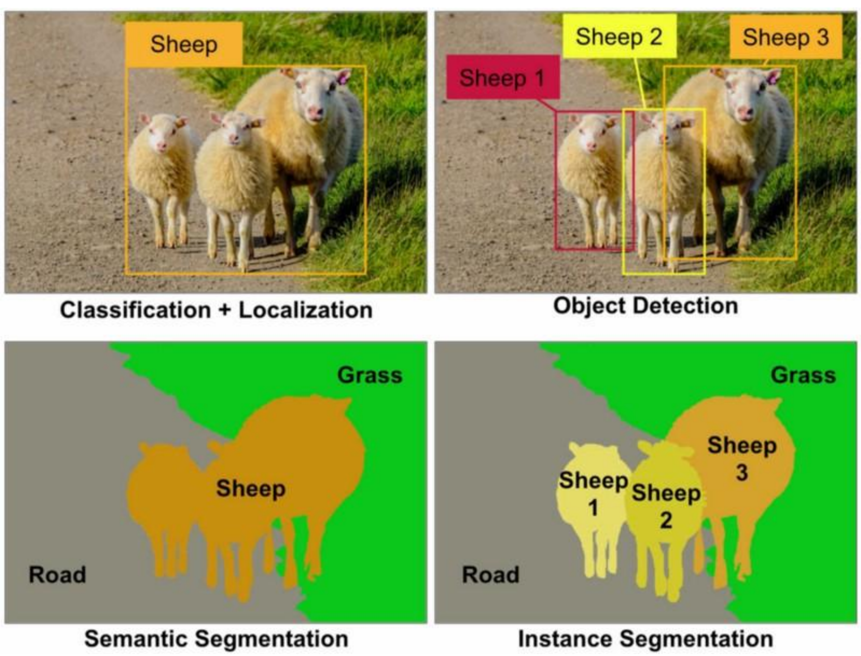


Figure: Different types of object detection

In the project I have used CNN (Convolutional neural network) to detect a total of 26 classes of ships from satellite images.

# Dataset Description

I have used the [HRSC2016]() dataset from kaggle. HRSC (High Resolution Ship Collection) dataset images were collected from Google Earth. The dataset contains images from two scenarios including ships on sea and ships close to inshore docks. The image size ranges from 300*300 to 1500*900 and most of the images are larger that 1000*600 in resolution. Although the dataset is supposed to have 31 classes, 5 classes have no images associated with them. So total number of classes are 26.


Figure 1: Sample Dataset

The dataset annotation contains information regarding the bounding box of the train image and the class of the image. Here is a sample from the annotation file of the relevant information,

| | |
|---|---|
| <Img_FileName> is the name of the file.<br><Img_ID> is the id of the image.<br><Imge_FileFmt> is the image format. Which is bmp.<br><Img_Location> contains the lat and long of the geolocation of the image.<br><br><Img_SizeWidth> and <Img_SizeDepth> are width and height of the image.<br><br>The image segmentation information is contained in the <HRSC_Object> tag.<br><br>Here, the <Class_ID> represents the class of the ship.<br><br><box_xmin> , <box_ymin>, <box_xmax>, <box_ymax> contains the bounding box the object or segment.<br><br><mbox_cx>,<mbox_cy> is the center coordinates of the area denoted by the bounding box.<br><br><mbox_w>, <mbox_h> is the width and height of the bounding box segment.<br><br><mbox_ang> is the angle of the segment relative to XY axis. | <HRSC_Image><br><Img_ID>100000001</Img_ID><br><Place_ID>100000001</Place_ID><br><Source_ID>100000001</Source_ID><br><Img_NO>100000001</Img_NO><br><Img_FileName>100000001</Img_FileName><br><Img_FileFmt>bmp</Img_FileFmt><br><Img_Date>2014-07-01</Img_Date><br><Img_CusType>sealand</Img_CusType><br><Img_Location>69.040297,33.070036</Img_Location><br><Img_SizeWidth>1166</Img_SizeWidth><br><Img_SizeHeight>753</Img_SizeHeight><br><Img_SizeDepth>3</Img_SizeDepth><br><Img_Resolution>1.07</Img_Resolution><br><br><Img_Rotation>090d</Img_Rotation><br><Annotated>1</Annotated><br><HRSC_Objects><br><HRSC_Object><br><Object_ID>100000001</Object_ID><br><Class_ID>100000013</Class_ID><br><Object_NO>100000001</Object_NO><br><truncated>0</truncated><br><difficult>0</difficult><br><box_xmin>194</box_xmin><br><box_ymin>243</box_ymin><br><box_xmax>972</box_xmax><br><box_ymax>507</box_ymax><br><mbox_cx>582.9349</mbox_cx><br><mbox_cy>353.2006</mbox_cy><br><mbox_w>778.1303</mbox_w><br><mbox_h>174.2541</mbox_h><br><mbox_ang>-0.2144308</mbox_ang><br><segmented>0</segmented><br><seg_color><br></seg_color><br><header_x>964</header_x><br><header_y>290</header_y><br></HRSC_Object><br></HRSC_Objects><br></HRSC_Image> |

In Figure 1, the green rectangle represents the bounding box information of the ship object. The yellow rectangle represents the bounding box information when we consider the angle of the bounding box. It is clear that, when the rotation is included the bounding box gives a more precise location of the object.

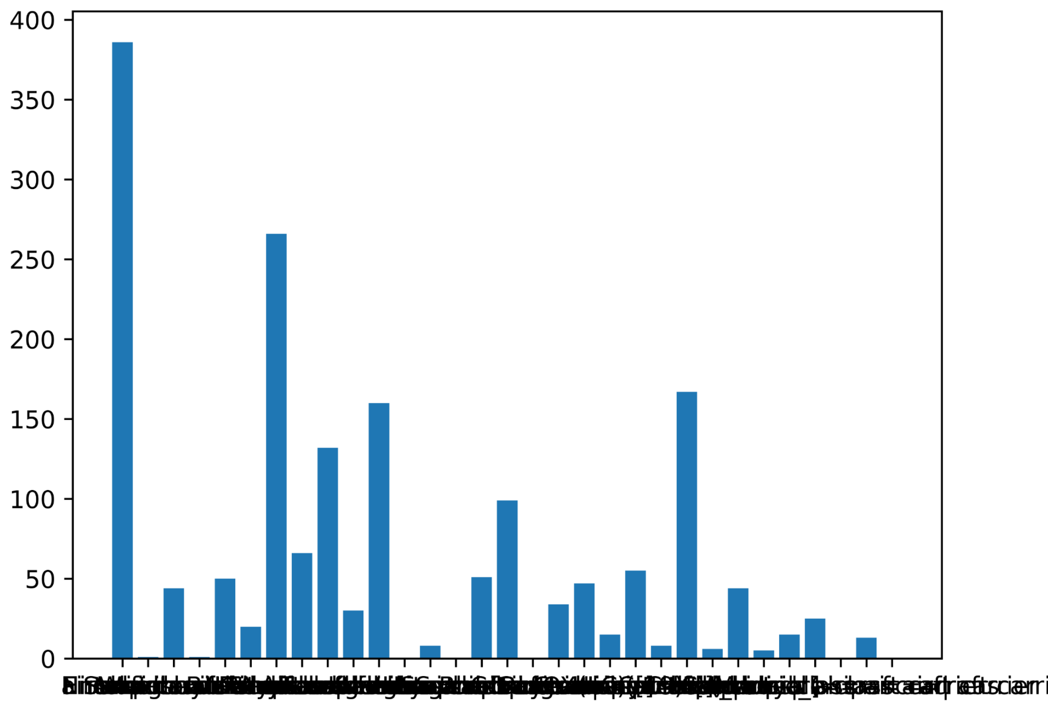


Figure 2 : Per class Data distribution.

From the per class data distribution, we can see that one class of dataset dominates the most, whereas some classes have very little data.

# Similar Works

In the paper, **[“A Novel CNN-based Method for Accurate Ship Detection in HR Optical Remote Sensing Images via Rotated Bounding Box”]** [by Linhao Li, Zhiqiang Zhou, Bo Wang, Lingjuan Miao and Hua Zong] the authors detected the various classes of the ships using Rotated Region Proposal network that uses Orientation agnostic regression to better predict the rotation invariant feature modules. They used a convolutional network with multilevel adaptive pooling and a Region proposal network with Acting rotating filters( ARF) to produce feature maps with orientation agnostic IoUs.

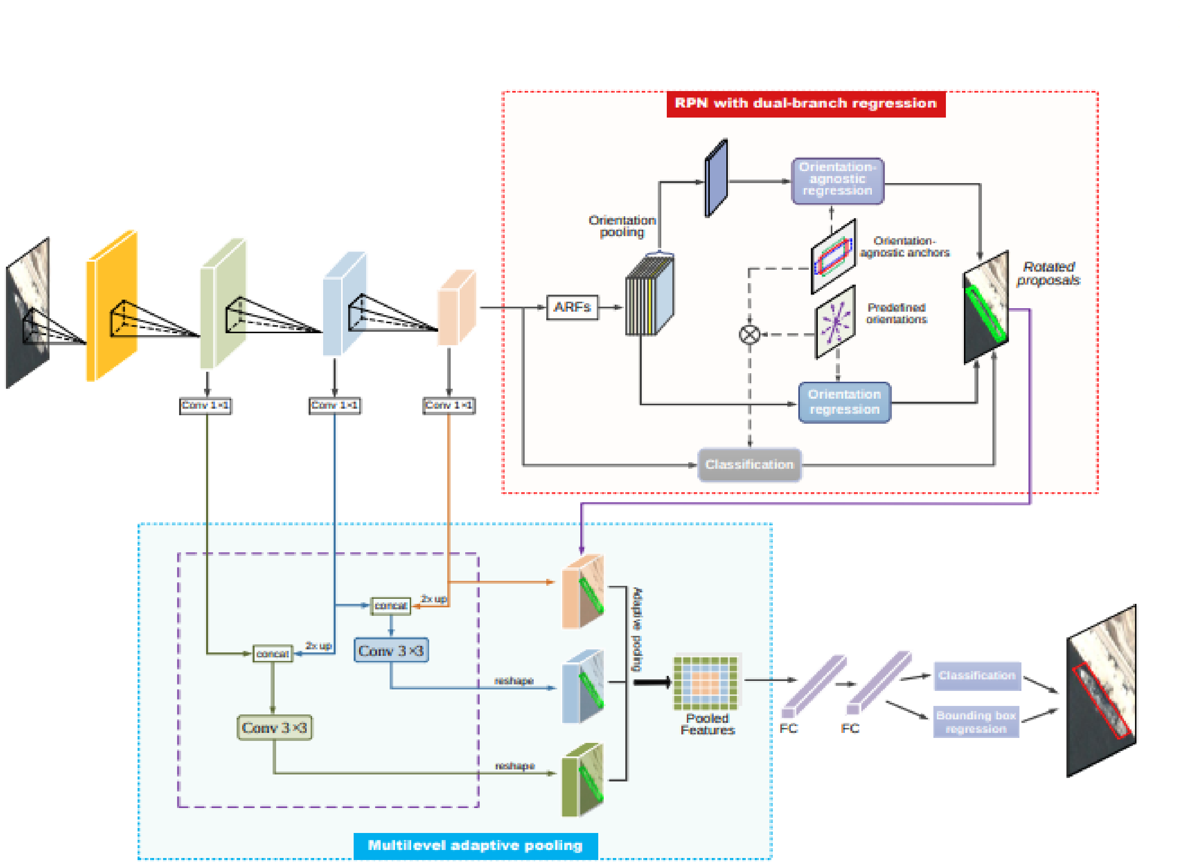


Fig. 1. An overview of the proposed method.

Figure 3: Similar Works

# Data Preprocessing

Image data is normalized in 3 channels by dividing by 255. Class data converted to categorical data and label data converted to one hot encoding. Bounding box normalized by dividing by width and height. Image path contains the image data.

# Basic Regression Model

- Used VGG16 with 'imagenet' weights as the input as features for our model.
- Fully Connected layer as Dense layer for Bbox regression with sigmoid activation.
- Fully Connected layer as Dense layer for class classification with softmax activation.
- Optimizer : Adam with 0.0001 Learning rate.
- Losses:
  - Bbox: Mean Squared Error.
  - Classes : Categorical Cross entropy.

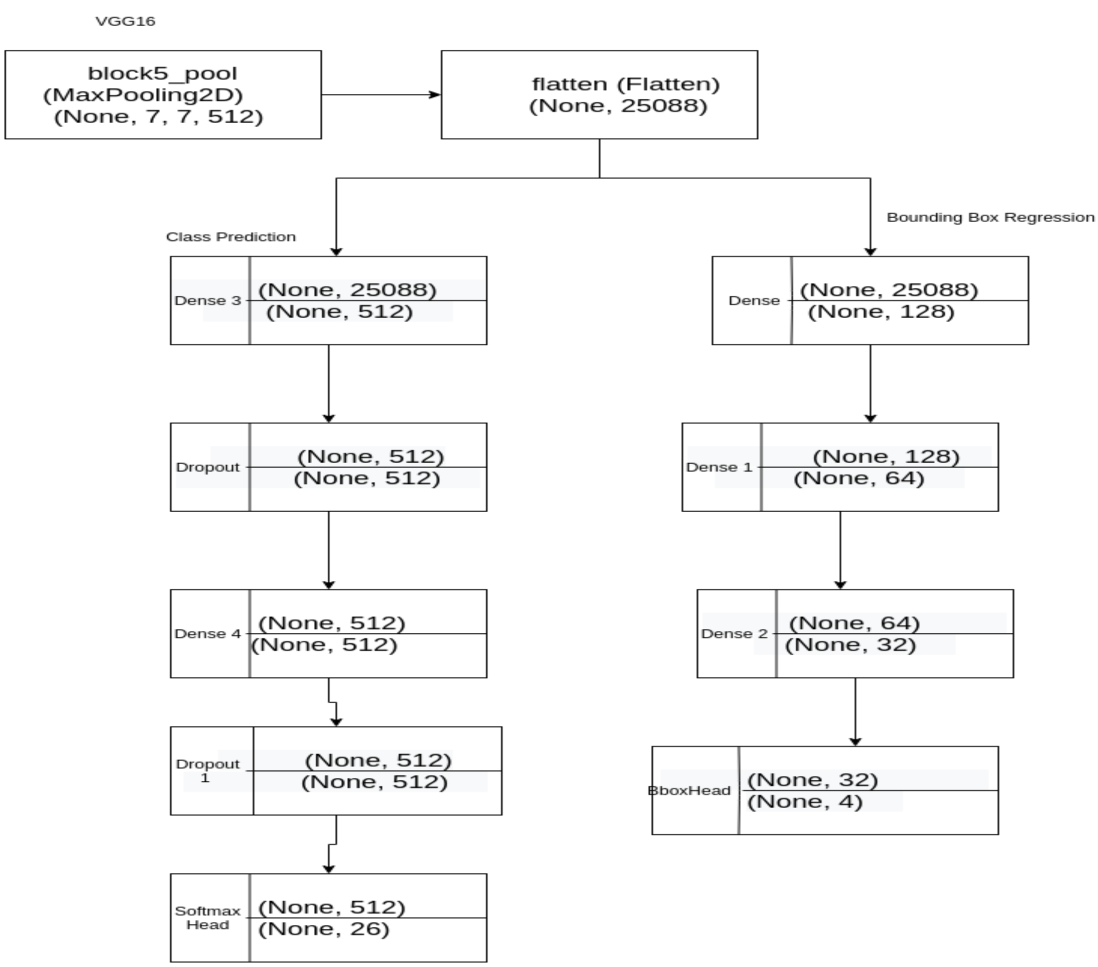


Figure 4: Model Summary

In the bounding box regression I did not use any dropout. In the classification branch I used Dropout rate of 0.2.

# Basic Regression Model Output

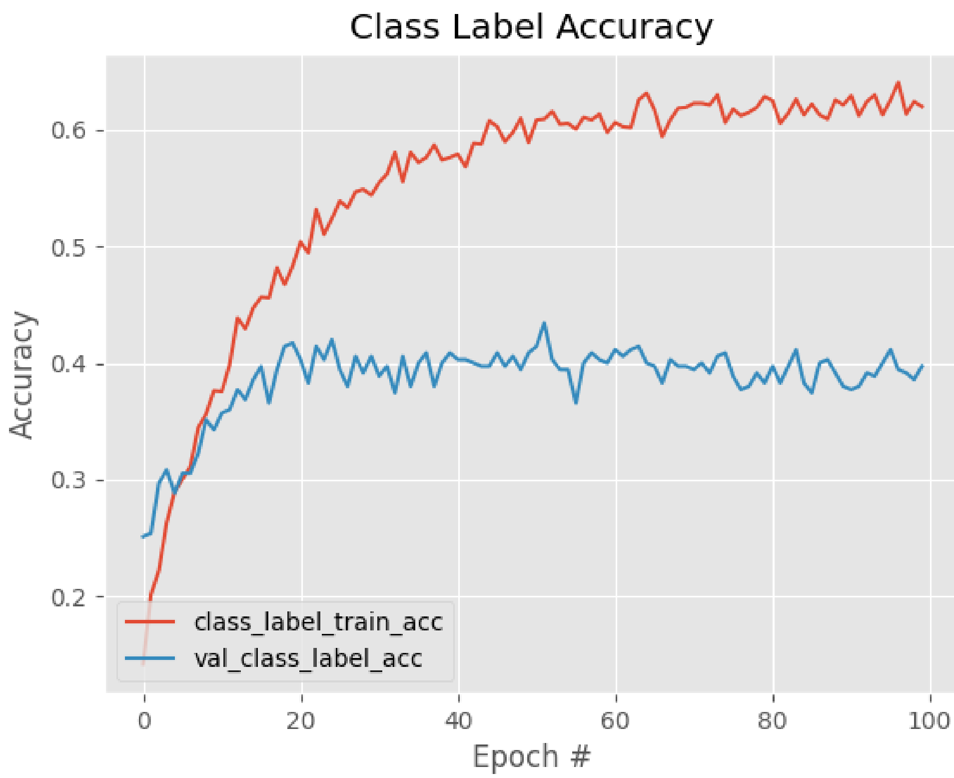


Figure 5: Class Label Accuracy

After 100 epochs, it is clear that the basic regression model training accuracy is improving but validation accuracy is not improving. This is a clear sign of overfitting.

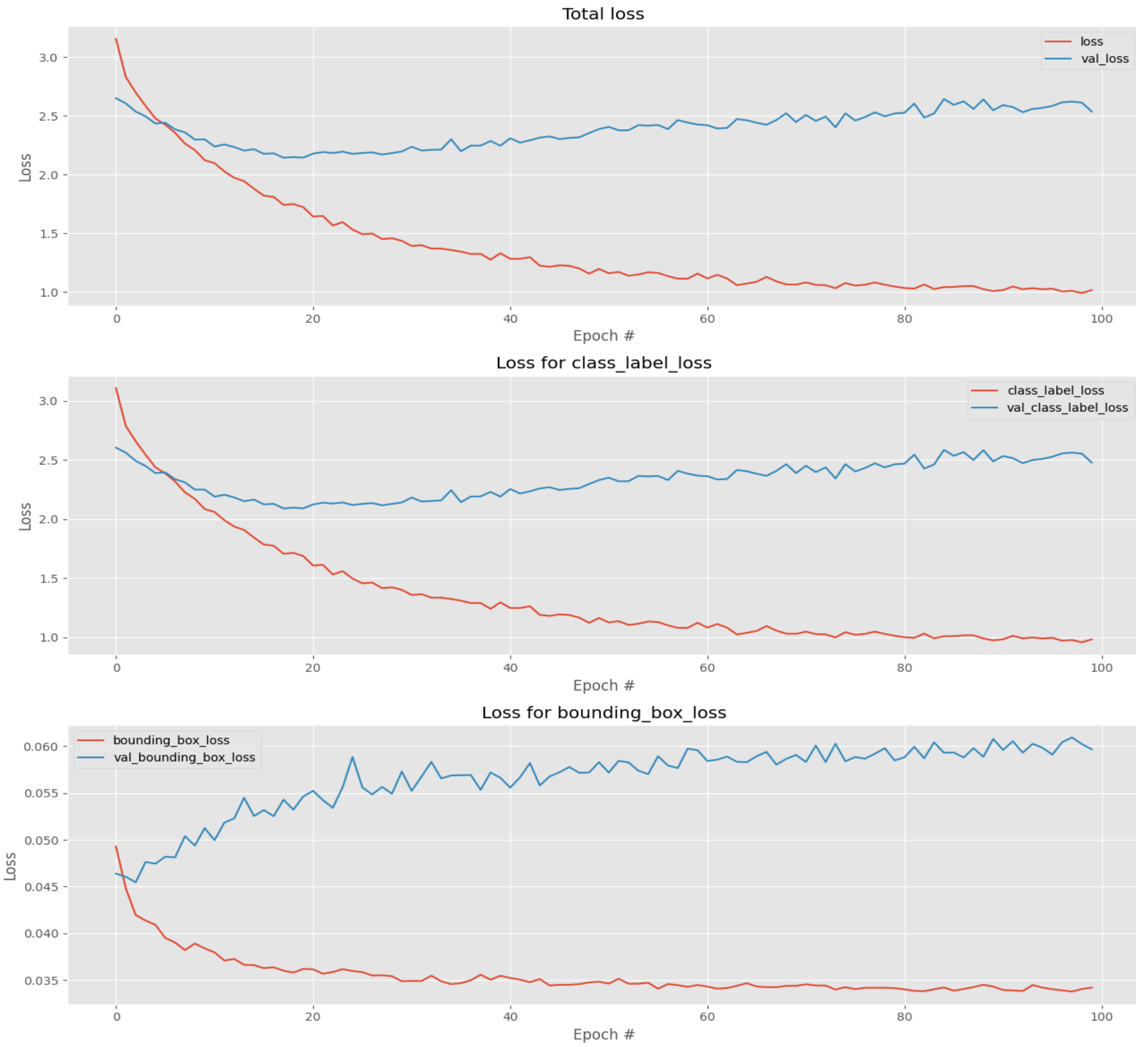


Figure 6: Total Loss

Same as in the loss we can see that, Total training loss decreased but validation loss did not decrease. We can see the worst case in bounding box loss. Because I did not use any dropouts, Loss for bounding validation increased although training loss decreased over time in epochs.

# Basic Regression Model Analysis

- Clear sign of overfitting.
- Steps to take
  - Stratify the data
  - Data augmentation
  - Scikitlearn classification report analysis

Apply other models like Fast R CNN which uses feature pyramid network to compute features.

# Data Augmentation

As we can see in figure 2, per class data is imbalanced and skewed towards one class only. It can seriously affect the validation rate of the training. To minimize the imbalanced data set, I used data augmentation. Data augmentation is a process where new image data can be generated from the existing data as a preprocessing step or as a runtime feed to the model. I used data augmentation to generate datasets as a preprocessing step.

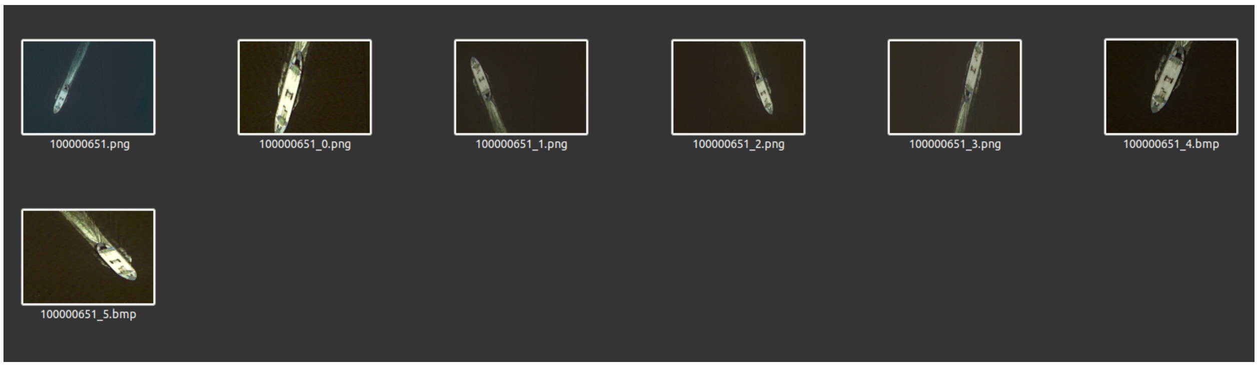


Figure 7:  Original Image( Top Left)- Augmented Image( Rest of the images)

- ❖ Used Parameters for augmentations:
  - ➢ Scaling
  - ➢ Vertical and horizontal flip
  - ➢ Brightness and contrast
  - ➢ Zoom and crop
- ❖ Output:
  - ➢ Augmented Image
  - ➢ Json file with bounding box and class label data.

Figure 8 shows the data distribution per class after image augmentation is applied. As we can see some classes still have relatively few images because those classes originally had few images.

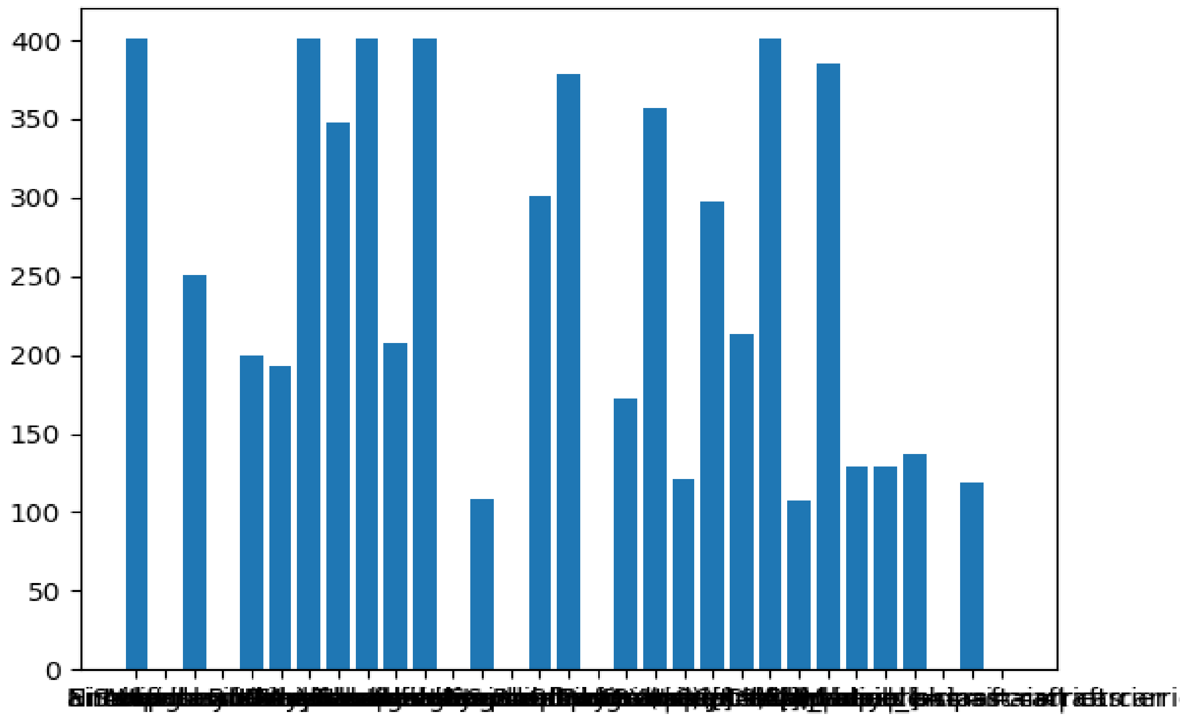


Figure 8: Updated data distribution per class after image augmentation.

# Updated CNN model

Now that image augmentation is done. I tried some modification for the CNN model. I applied dropout to both class prediction branch and bounding box regression branch.

The updated CNN model structure can be seen in Figure 9.

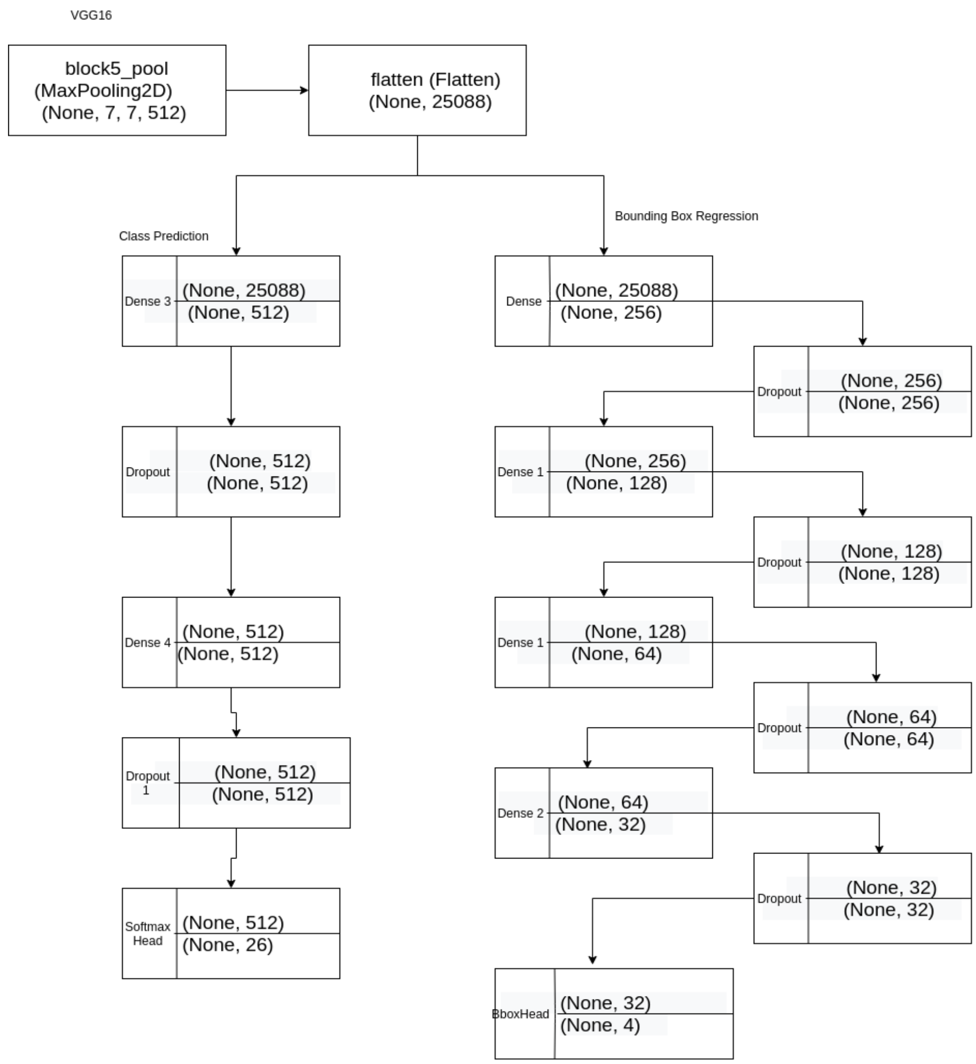


Figure : 9 updated CNN model

# Updated CNN Model Output

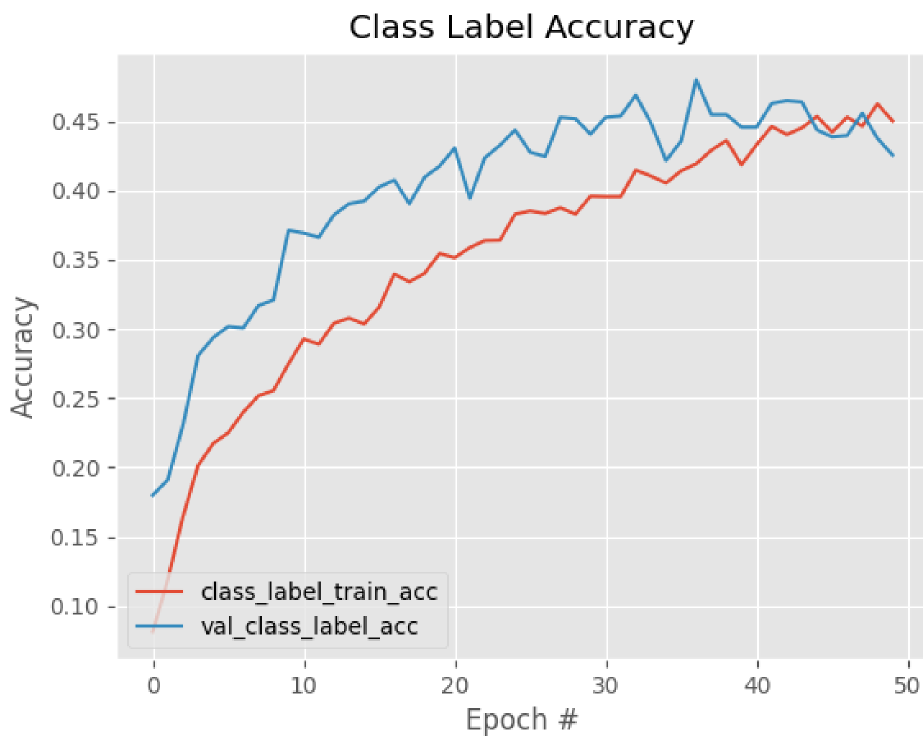


Figure 10:  class label accuracy

The updated CNN model provides a better result than the basic regression model. Although it does not give a better class label accuracy, it significantly performed better in training and validation. Also it minimized the overfitting problem. In figure 10, we can see that after 50 epochs, validation and class label accuracy kept improving. In Figure 11, we can see that, the loss for both training and validation set kept decreasing. As of compared to figure 6, where we saw a high bias between training and validation loss. Here we can see that in the updated model, the bias  has significantly decreased.

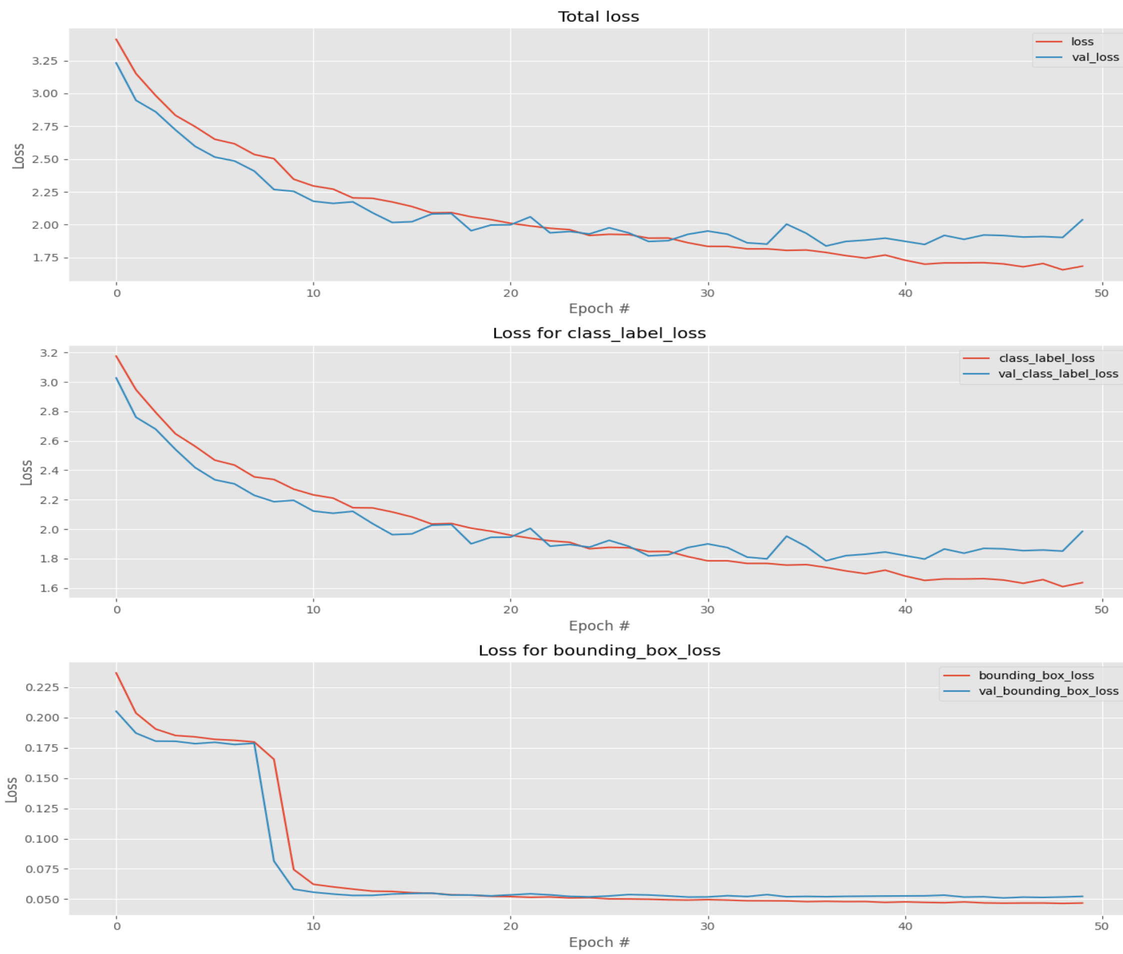


Figure 11: Loss for updated CNN model

# Updated CNN Model Predictions

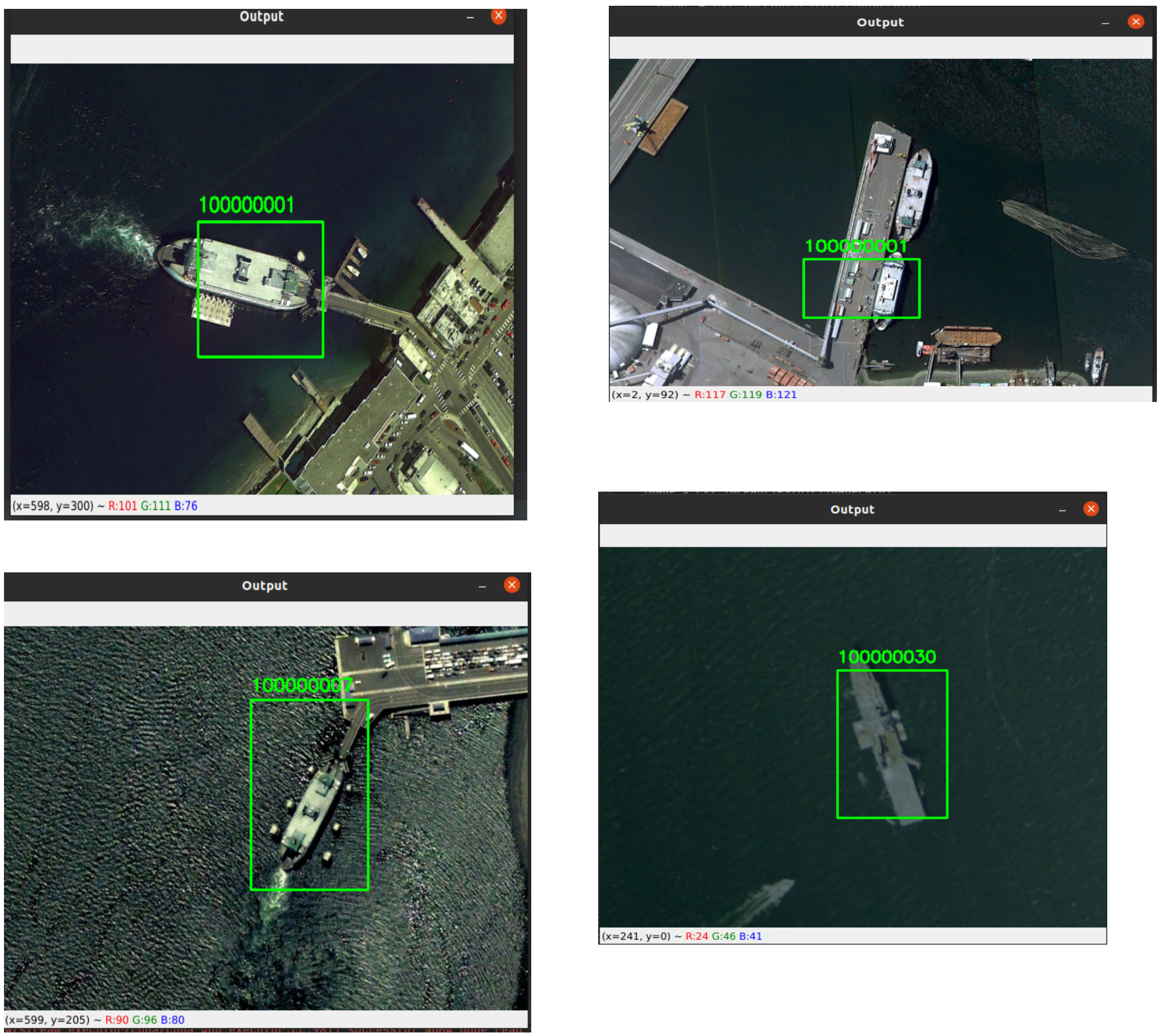


Figure 12: Updated CNN Model Predictions

In figure 12, we can see some predictions of the test images.
It has some limitations mainly, it can not detect multiple objects in the same image. To detect multiple objects and get better class prediction accuracy I would need to use advanced models for detecting images.

# Fast RCNN Model

There are various state of the art models that can detect multiple objects in the same image. Mainly YoloV4,  Mask R CNN. As I am more familiar  with how Mask RCNN works, I wanted to use the Fast R CNN model for my dataset.

Fast RCNN uses a Feature Pyramid Network( FPN) to generate bounding box proposals from a given image. It then calculates the IoU between the ground truth and the generated features or anchors and applies Region Proposal Network to identify areas with foreground and background. The foregrounds have a higher IoU threshold and backgrounds have lower IoU thresholds. Then it applies  ROI Head with ROI Polling the detect and predict the segment of the image.

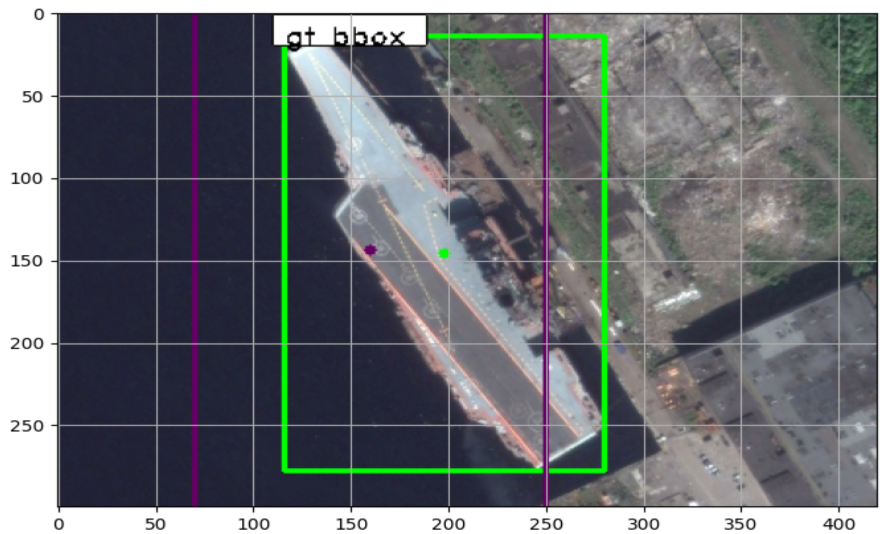


Figure 13 : Anchor(purple) and ground truth(green) rectangle of Fast R CNN

# Fast R CNN Model Output

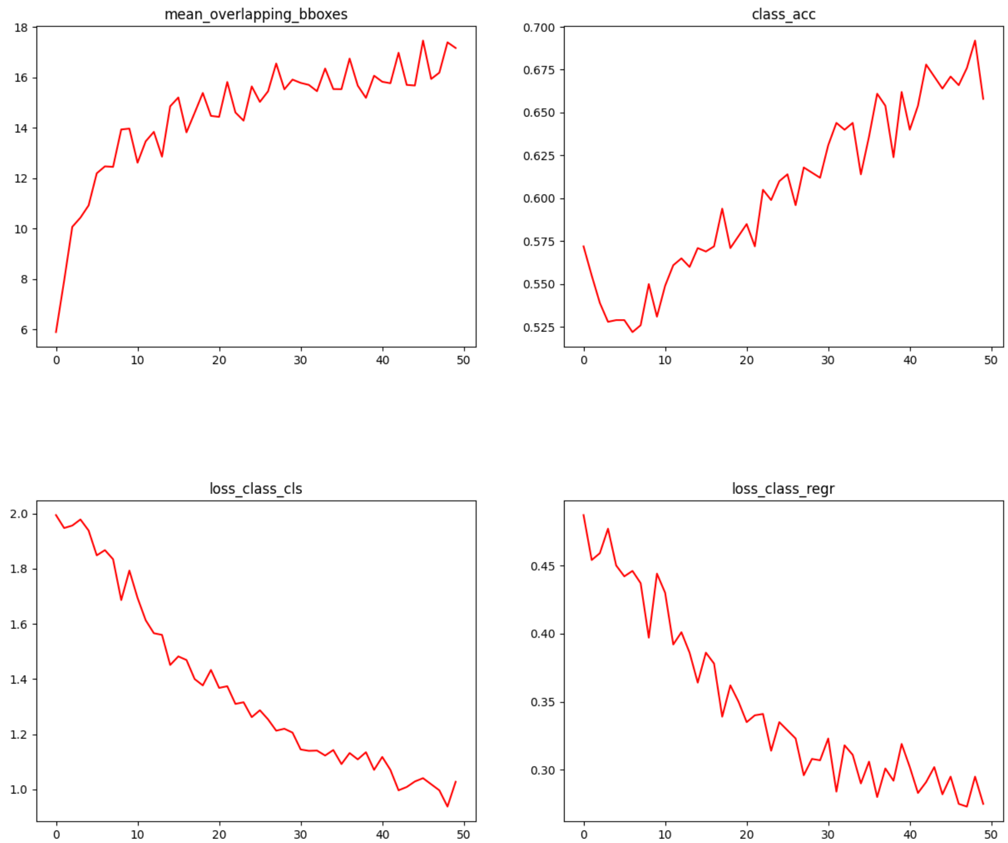


Figure 14: Class accuracy and loss

With only 50 iterations we can see a massive improvement in the performance of class accuracy. With 1000-2000 iterations It should be able to perform better. Model can be further improved by including the rotation angle in the feature extraction process.