Thomas Keeley Individual Report
Group 8

The concepts learned in this course have culminated to prompt a final project. My background and continued interests in Geospatial analysis were met with the other group members and it was decided to pursue an object detection project focused on satellite imagery. There are several use cases of object detection in satellite imagery with robust training datasets that target a large collection of object classes. One particular class of object that when viewed as a target of object detection in satellite imagery could have many analytical implications is small vehicles. Small vehicles appear consistently in satellite imagery of urban and populated regions and the ability to detect and classify these objects would prove to be useful in many applications. This project therefore focuses on developing an object detection application aimed at small vehicles in satellite imagery.

The first step was to evaluate the available datasets that provide annotations for small vehicles. There were a handful to choose from, but the group decided to move forward with DOTA which contains oriented annotations for over 10 object classes including small vehicles. I documented all of the steps to follow in the downloading, unzipping and collecting all of the required data from the command line in a series of commands. These efforts set up the code to be run in annotate.py. This initial effort in programmatically ingesting and processing the annotations to be used in the subsequent modelling process was largely an individual effort on my part in developing the data into a usable format for the model architecture that I intended on using. The entire annotate.py file was written by me in an effort to kickstart this project and set up the rest of the group to conduct successfully the modelling process of their choosing.

The steps involved in this processing module include reading all of the annotations from the formatted text, recognizing and extracting annotations for small vehicles, reformatting the annotations from oriented bounding boxes to horizontal bounding boxes, and compiling the annotations in a data frame. An additional functionality was later added that would crop and slice the images into smaller images. The annotations are then appropriately refactored to reflect the locations of objects in the newly developed image slices. Finally, the annotations are split into train, test, and validation sets and saved as csv files. The csv files are formatted in the required fashion for implementation of Retinanet which is the model effort that I chose to explore in this project.

Retinanet is a single stage detector that leverages a concept of focal loss which increases the weights for harder to classify regions and down weights the background areas of the image. This essentially maintains the speed of single stage detectors but enhances the accuracy of the operation. The implementation of Retinanet in this project follows the implementation by GitHub users yhenon. This pytorch implementation serves as the basis of conducting the object detection in satellite imagery using DOTA in the particular exploration of this project. The implementation by yhenon is formatted and optimized to run on the COCO dataset which varies in types of objects and general aspect of the images in relation to satellite imagery. The input data and a few individual features were altered to function using the target of this project, but the modules contained the yhenon repository were largely preserved and utilized in conducting the training process as well as validation.

The steps involved in leveraging the modules in the yhenon repository were documented in the repository to provide further opportunity for the group members to conduct similar analysis and potentially enhance the results of the Retinanet modelling. Next, a post-training module was written to instantiate the model to run evaluation metrics and produce annotations. There is a module in the yhenon that provides the capability to both produce the mAP metric and calculate the annotations. The code written in evaluate.py leverage this module and is used for initial processing of the trained model. The function I developed following producing the predicted annotations provide a way of visually analyzing the results by drawing a rectangle for the predicted bounding boxes on the images. This function was crucial in evaluating the validity of the model throughout the testing phase of this project.

Next, a quick function was developed to compare the performance of the various models trained in Retinanet by calculating the inference time of each and displaying the mAP. This function is used in comparing these models for the final project report. There was also an additional effort contained in evaluate.py that aimed at evaluating the trained model on a satellite image from outside of the DOTA dataset. The intention of this was to test the full functionality of this application in the context of Geospatial analysis by evaluating on a large, very-high resolution satellite image containing several thousand vehicles. There would not be a way to measure the performance of this evaluation as there are not any provided annotations for this imagery. This function has the ability to ingest a collection of sliced images from the larger satellite image and conduct the object detection on the image. Next, these annotations are converted to a format that relates back to the geographic context of the original image. This allows the user to utilize this train model and create geographically attributed detections that can then be further analyzed in the geospatial context.

A significant individual effort relied upon the code developed in annotate.py and evaluate.py as a large majority of time was devoted to conducting modelling experiments and troubleshooting using the Retinanet implementation. This was an incredibly arduous task of trial and error as the training process often took several hours and refactoring of input data. Aside from developing the files that provide the data processing capability, this modelling and comparison process was the most consuming task of my individual contribution. Several factors went into analyzing the limitations and performance of the trained models in this phase of the project. One of the biggest contributions to making this implementation work for the use cases of small vehicles in satellite images was slicing the images into smaller dimensions. Additionally, testing the different architecture depths by using different ResNet versions in constructing the training architectures proved to have a positive impact on the performance. This led to the comparative analysis conducted in assessing the application of satellite imagery in the object detection domain using Retinanet by further understanding the requirements to analyze smaller objects and larger dimensional imagery datasets.

The analysis conducted in this project has left a lot be desired in that the results were less than expected. Further exploration and improvements will be made beyond the scope of this course in assessing the limitations in the repository code and overall network architecture. This field of research within the Geospatial domain will be a continued interest and further exploration will be conducted to fine tune the results of these efforts put forth. Further evaluation of other methods and implementations will also be considered in producing a more custom solution to object detection in satellite imagery.