2021 CSE 40626 Final Project Status Report I

Prepared by

Joseph Kuebler, Patrick Creaven, Matthew Kozak

University of Notre Dame,

Notre Dame, IN, 46556

**1. Project Goal and Milestone**

Our final project will be the default project, which is to detect and identify objects in the drone-captured image data set provided by Professor Yiyu Shi through DJI. The overall goal for our project is to be able to successfully train a neural network to differentiate between the objects presented to us in the training data and the testing data images fed to our model by the professor that we do not have access to. We intend to measure this goal with the accuracy measured by Intersection over Union (IoU) which evaluates the overlapping space between the ground truth bounding box that is a part of the training data, and our predicted bounding box that will be generated as a result of our model. An IoU close to 1.0 is excellent and an IoU above 0.5 is good, so for that reason, we aim to achieve an IoU within the range of 0.5-1.0, with a desire to eventually reside in the range of 0.7-1.0 as our model becomes more advanced.

Our goal for the first milestone was to have the data cleaned, understand the benefits and drawbacks of different models, use that understanding to decide on which model to pursue, and then have a basic starting version of that model working.

**2. Tasks Completed So Far**

So far we have been able to gather a better understanding of different image segmentation models, the Intersection over Union metric, instances of segmentation, and precision in relation to false positives and negatives. With this understanding, we have been able to research what types of models work best for this type of task. (<https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>)

As far as data preprocessing goes, the dataset provided to us already has a lot of this done. All images are of the same resolution and aspect ratios, so any resizing, cropping, rotating, and scaling is not necessary. Additionally, all images are correlated with an .xml file that contains existing labels and bounding boxes, so we do not need to label our data or create the bounding boxes. Finally, after inspecting the datasets visually, there are no images that would disrupt our data (i.e. all black images, repeated images, etc.). The only things we will want to do to our data, is normalize it and one hot encode the data. To do this we just divide by 255, because the input values are the pixels in the images that range from 0 to 255 and run a one-hot encoding method on the data to prepare it for binary classification which is essentially what we are doing by predicting what an image is.

As for modeling, we researched a few different methods, but because we have had experience working with Keras and Tensorflow in the past mini projects, we chose to use this tutorial as a reference for designing our model: <https://stackabuse.com/image-recognition-in-python-with-tensorflow-and-keras/> . Choosing a sequential model with relu activation and the same padding for each images. The optimal thing about this tutorial is that it accounts for overfitting using the general flow of convolutional, activation, dropout, and pooling. It also provides considerations for how many layers to use, but we will tweak these layers and experiment with our own to determine the optimal structure for our model on our data.

This is also a great tutorial on training object detection models from machinelearningmastery.com which is the website we used as a guide for Mini Projects 3 and 4. <https://machinelearningmastery.com/how-to-train-an-object-detection-model-with-keras/> .

We will synthesize these two tutorials in our starting model and begin to visualize our bounding boxes for the data to calculate the IoU for our model.

**3. Team contribution (40625 only)**

How each of the team members contributed to the project so far.