Detecting Road Overpasses from Satellite Images

Jared Able

1 Introduction

Roads and bridges are essential to civilian, commercial, and government transport across the world. One facet of roads that is often ill-captured by GPS navigation systems is that of road overpasses, and this can have severe consequences for drivers of large vehicles. We aim to rectify this lack by predicting the presence of road overpasses in a given satellite image. To do so, we apply a convolutional neural network to a dataset of satellite images that we labeled and assembled.

1.1 Stakeholders

GPS navigation companies, US government, road developers, advertising companies

2 Methods

To create our dataset, we randomly generated latitude/longitude coordinate pairs and thereby obtained satellite images in three different geographic areas: the United States, part of Asia, and part of Europe. However, randomly chosen satellite images are extremely likely to miss overpasses. We fixed this by calculating graph edge intersections using GeoPandas GeoDataFrame information obtained from OpenStreetMap via the Python package OSMnx. Afterwards, our random search was guided by presence of overpasses. As a nice bonus, this graph information also allowed us to automate our image labeling process.

Next, we had to harmonize the Google Maps satellite image with the OpenStreetMap graph data to ensure accurate labeling. Initially, overlaying the graph data on top of the satellite image yields incorrect shifting and scaling. To fix this, we performed a change of coordinates due to the curvature of the earth. After doing so, we were able to get a very good match.

We use the most popular model for image analysis: a convolutional neural network (CNN). A CNN consists of several layers that alter the image data as it passes through, and we implemented the common approach of using convolve and pool layer pairs. Adding dropout layers assists with overfitting, and randomflip and randombrightness layers add some artificial diversity to our training set. Additionally, we used model checkpoints to save our best model during training and batch loading to deal with the large size of our dataset.

Our image data is quite large at 640x640 pixels, and we have over 12,000 images, so working with this large dataset required heavy resources. We enlisted the help of Indiana University's high-powered computer BigRed200. This allowed us to run experiments in batches by scheduling through Slurm. After trying out a variety of layers and hyperparameters, the highest validation accuracy we achieved was 92%. The model was more likely to miss a true overpass than to detect a false overpass.

3 Future Goals

There's a lot of future work to consider. First, instead of detecting at least one overpass, we could count how many overpasses there are in a given image. We made some progress in this multi-classification counting problem, but the uneven distribution of our labels made training difficult (despite trying to counteract this uneven distribution with weights).

Next, we could increase the volume and diversity of our image dataset. With our automatic labeler and the wealth of global satellite image data, this would be relatively simple. One type of image we discovered that was omitted was satellite images near overpasses, but not containing overpasses themselves. Including this type of image would likely help models to determine the presence of overpasses by not valuing false cues from nearby data.

Finally, we could experiment more systematically with layers and hyperparameters such as the learning rate of our optimizer. Also, instead of using the commonly used loss function of cross entropy as we've done here, it would be interesting to create our own loss function tailored to our task. This would be most useful in counting the number of overpasses, where near-enough counts would be penalized less harshly than counts which were way off base.