Ibraam Melika # Road Detection

A. So for this project we are tasked to train models to detect routable road networks over entire cities. The repo provided by cresi does this and it is also able to provide speed limits and travel time estimates for each roadway. Cresi is able to successfully do this with spacenet imagery and labels and google satellite imagery with OSM labels. Some major issues that we ran into. I for one have not had any experience with Docker. I did have access to a GPU so the best route to tackling this project was to go down the docker route instead of the Google colab route. This ended up leading to a few complications with Docker. One issue was when cloning the repo and creating the image, due to my hardware and network limitations it took some time to download around 20 GBs of data from the repo and make that image. After several hours and after completing the image, I made a docker container. However I end up with another problem, the jupyter notebook won't launch. Docker would give a link and produce an html for the jupyter notebook but it wasn't accessible. However after many hours of troubleshooting I was able to make docker show my jupyter notebook.

B. Now how this works, we input the image into the model, then in return the model sends us the segmentation mask of a lot of post processing tasks. With this we get a road network out of these images or road data. To explain it further into words, the detector works by first taking the image or map data, then extracts the segmentation mask from the cutout. After that it would result in some sort of build of a total stitched map which is then divided by a coverage map and that outputs or returns the normalized map data.

Ibraam Melika

Road Detection

```
map data  <-->  Segmentation mask
```

Build total stitched map

Divided by the coverage map

Equals

The normalized map