Sean Yang

Pd 4

2/13/19

Journal report 17

I found a webrtc demo with Tensorflow object detection with flask on the backend a couple of weeks ago. This week, I looked at the source code of the webrtchacks demo and gained a solid understanding of what it tries to do. With this knowledge, I will be able to modify the demo and save lot of work. In the demo, it uses Tensorflow to handle the data. My project is quite similar, but it uses other Python libraries instead. Many of the functions in the demo remain useful. For example, I can still use the way the demo reads in data and loads images into numpy array.

Based on the instructions from the github repo, I’m supposed to use Docker to make it work properly. That is really annoying because Docker requires windows 10 pro. I was able to use my other linux boot on my laptop to use Docker. Docker still had issue. It turns out, that wasn’t necessary. The point of Docker was to solve dependency issues. Since I’m using different libraries and methods, Docker wasn’t actually necessary. In the future, I should probably only stop and setup for dependencies when I run into problems.

I got the demo to run successfully in a flask server locally. I will do everything locally first and put it on the server when it’s ready. This shouldn’t be an issue since Evan was able to do it. I have already started modifying the demo and got the camera capture to work.

