Bi-Weekly Report Number 8

Computer Vision for Object Detection in Medicine

Team Number 6

Author	*Email*
Benedict Chan	benedict.chan.17@ucl.ac.uk
Shirin Harandi	shirin.harandi.17@ucl.ac.uk

COMP0016 System Engineering

February 21, 2019

Department of Computer Science

University College London

Week Overview

During the week we were finally able to connect the VM to the local machine by using Python Flask. After sectioning off specific ports for the program to use we then went on to installing all the required libraries in order for the system to run (such as virtualenv and flask).

While this allows the machines to communicate with one another we wanted to run the system continuously. In order to have this functionality we wanted to use Nginx as a load balancer and reverse proxy in order to balance the multiple connection requests that the server will be receiving. After that we decided on using Gunicorn in order to setup our continuous deployment. After this we were able to access send receive information between the local machine and VM.

After setting up a simple flask application to send data between 2 applications, we then went on to split our application into parts. To communicate between the applications, we decided to use a GET request that sent JSON data along with it. The client application reads a frame from the camera. It then sends the image data to the web service as an array through JSON. The web service then runs the TensorFlow algorithm over the frame and sends back the detection details as the return for the GET request.

List of tasks Completed

- Connect local machine to VM
- Enable Python script on VM to run continuously
- Split application into 2 parts
- Setup communication between 2 applications with test data

Plans for Next Two Weeks

- Start user interface
- Look into moving TensorFlow on the VM
- Connect Local machine to VM (With GPU)
- Make a test API that runs on the cheaper VM to allow for client-side testing

Individual tasks completed

Benedict

I helped set up a simple flask application. I mainly focused on testing and experimenting with splitting our current application into one that would work in 2 parts. I moved the TensorFlow detection and setup algorithm onto the web service and the rest on the client side.

Shirin

After a long process I was finally successful in connecting the local machine to the virtual machine and sending messages between the two. It is now also possible for VM python script to run constantly meaning we do not need to log into the VM at any time.