Bi-Weekly Report Number 5

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

December 14, 2018

Department of Computer Science

University College London

Week Overview

Having already decided on using the Google TensorFlow object detection API. We still wanted to further understand the basic behind object detection and recognition algorithms. So, we researched more into the technologies we were using and other similar APIs to further our knowledge in the area.

The first step in developing was to make sure that we had all the dependencies installed and that the sample program given with the API worked. Although we may need TensorFlow GPU in the future for our system, we first installed TensorFlow CPU to test our environment and get used to using the library. After getting the sample program to work we then moved on to object detection from a live feed.

As we were not supplied a camera for this project, we tried to setup python with our phone cameras. To do this we used DroidCam (Android app) to stream the video to a Computer acting as a webcam. Then we used the OpenCV library to read the video feed to a viewable window. To add object detection, we looped frames from the video into the detection algorithm from the sample code. The end result was successful, however the frames per second of the video was very low (about 1 frame per 5 seconds). However, this was to be expected as we were only using the CPU version of TensorFlow.

After the last few weeks of research, we can finally start designing a system with confidence in our understanding of the APIs and Frameworks we are using. For our project we will be using Microsoft Azure services to handle the ‘detection’ of objects. To do this we will need to upload a video stream to the cloud service. This separates the system into 2 parts, which we have broken down what tasks need to be done for each component.

Front-End (Local PC)

- Video input device connected here
- Shows the live video feed
- Uploads video stream to the backend
- Shows live object status retrieved from backend
- Highlights objects in video feed that have been detected

BackEnd (Azure)

- Retrieves video stream from Front-End
- Runs video stream through object detection API
- API uses the medical instrument models we have trained ourselves
- Sends the detection data back to FrontEnd

List of tasks Completed

- Research into object detection and recognition algorithms
- Setup environment with TensorFlow CPU
- Use TensorFlow object detection API with COCO (Common objects in context) on their test images
- Created simple application using TensorFlow object detection API with live video feed from phone camera
- Create a plan for development and lay out the tasks needed to be completed

Plan for Christmas

- Finish website
- Make training pipeline
- Take preliminary photos for generating models
- Setup a simple program that connected to Azure, establishing the front and backend of our system.
- Prepare for elevator pitch
    - Slides
    - Content

Individual tasks completed

Benedict

In the last 2 weeks I tried to setup the environment on the computers in the labs, but they did not have protobuf compiler installed. Therefore, I had to setup the environment to work with the sample code on my own personal computer. Then I adapted the sample code to work with the video feed from my android phone.

Shirin

Continued working on the website by setting up the design as well as writing content.