Ethan Owen
2/18/2024
EE113DW Status Report 5
Gesture Control for Home Devices Project - (solo project)
https://github.com/Ehanlion/HGR_project

Main Intention:
With midterm season closing, this week saw a ramping-up of development pace. My original plan has an important deadline to hit by the end of week 8 so it is crucial that, during this next week, development steams ahead with no interruptions. After my research last week, my board finally was shipped, however it is still in transit. In the meantime, I have been experimenting with another, less advanced raspberry pi board to become acclimatized to the ecosystem. This is an important step, and one I needed to take on development because integrating my code onto the RPI could be troublesome, so it is best that I understand how best to work with the RPI now instead of next week, where troubleshooting extensively could be a huge time-killer. Focusing this week on testing my RPI and testing integrating my code is a good foundational step that will set me up well for next week.

Conceptual Development:
This week, I researched how programming on an RPI works, how to integrate code into an RPI, how to download and install libraries, how to set up an RPI to be programmed via network, and how to have the RPI run just my program without any other interference.

Luckily the RPI I am using has network functionality, so I am able to connect it to whatever network I am working on. That makes integrating code and installing libraries is relatively straightforward. To do this step, all I need to do is wire the RPI to my desktop, connect to it, and manually download the libraries over the internet. The same general principle can be applied to downloading my program onto it, which can be done via a usb drive. This is the easier of the steps however.

The big part I needed to research was on how to program over the network once the RPI is configured and then how to have the RPI run my program and only my program. Tackling the first one, it is possible to set up an RPI to handle a remote SSH connection, allowing a remote user to execute command lines as a root user. This is the method I will use for my project. Next, there is a variant of the standard Raspberry PI OS (called Raspbian) that doesn't have a GUI. The OS is called 'Raspbian Lite' and it will be what I am using for the project. It is essentially the normal OS without any of the graphical overhead, meaning the RPI behaves more like a server than a user computer. There is a way with this OS to have a program be run immediately after startup as well and I can log the results back to my desktop via an automated SSH connection script.

Implementation and Development:
Like last week, there is not a lot of actual implementation nor development to show but the proof is in the pudding (as they say). I used what I researched to set up an RPI with the necessary libraries and connected to it via an SSH connection. I installed Raspbian Lite onto it and was able to run a simple hello world script. I was not able to test my full program on the board as of yet though.

Resources:
I continued to use online resources from both OpenCV and MediaPipe to advise me on the project. Now though, I have added resources directly from RaspberryPi's website to help me with the RPI design, interaction, and configuration. These sources together help me workout my programming and how to implement it on my actual board.

Individual Contribution:
This project is an individual project and as such, all development was undertaken by me.

References:

1. https://github.com/googlesamples/mediapipe/blob/main/examples/gesture_recognizer/python/gesture_recognizer.ipynb - Website for MediaPipe with an installation guide as well as a python example.
2. https://docs.python.org/3/library/asyncio.html - Website for Asyncio library
3. https://developers.google.com/mediapipe/solutions/vision/gesture_recognizer/python#live-stream - Website for MediaPipe with a python implementation of a hand tracking overlay.
4. https://docs.opencv.org/3.4/dd/d43/tutorial_py_video_display.html - Website for OpenCV's example on getting as video feed functioning
5. https://store-usa.arduino.cc/collections/iot-cloud-compatible?selectedStore=us - Website for Arduino's cloud computing boards
6. https://www.raspberrypi.com/news/introducing-raspberry-pi-5/ - Website for Raspberry Pi's introduction of the RPI5
7. https://forums.raspberrypi.com/viewtopic.php?t=229888 - Forums posting on Lite and non-gui variants of the RPI OS called Raspbian Lite.
8. https://www.raspberrypi.com/documentation/computers/remote-access.html - Website detailing how to setup and RPI over a network