

Ethan Owen

3/3/2024

EE113DW Status Report 7

Gesture Control for Home Devices Project - (solo project)

[https://github.com/Ehanlion/HGR\\_project](https://github.com/Ehanlion/HGR_project)

### **Main Intention:**

Week 8 was an excellent week in development. Coming off prior weeks, primarily focused on research and learning, week 8 saw key implementation and steps taken to actually develop new code. I wrote code that connected my gesture recognition code to functions provided by the huesdk package (used to control Phillip's Hue lights), allowing my code to talk directly to the networked lights in my room. In testing, I was able to power on and off the lights remotely in my room, from a gesture-to-camera distance of about 10 feet. This is a critical development step because integrating gesture recognition with the actual device control was the final hurdle I needed to overcome to ensure this project is completed on time. Thankfully, with this week's work, I am completely confident that the project will be fully completed on time, if not even a little early.

### **Conceptual Development:**

This week, I worked with the huesdk package as well as the Phillips Hue Hardware to create a test program, allowing me to remotely control my lights from a program on my computer. This test program was an updated version of what I had built last week and required me to further explore the functions of the huesdk package.

The package has the ability to search for a Hue bridge, the controller for the lights, via its username and an ipv4 address. Once the bridge is found, there are functions available to connect to the bridge. We can then list the information about connected devices, in this case lights, and create instances of those lights within our code. This gives us the ability to control the lights individually. Researching this process gave me confidence to integrate my solution in code.

### **Implementation and Development:**

This week, I re-created my original gesture recognition code but with significant modifications. Primarily, I added the code used in my test program for huesdk to allow connection to the hue bridge and control of the lights. I removed the GUI output of the program as well since, when integrated into the RPI, there is no need for a GUI since it will be running without monitoring. This speeds up the program process time as well since the main hindrance is the rendering of the frames for output. Testing this program saw success. A naive approach, ruling that a closed fist turns off lights and an open palm turns on lights, was a quick way to test the overall integration for functionality. In the next week I will refine the thresholds for controlling the lights and add more functions, like changing color and brightness.

### **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. Furthermore, I have added Huesdk to the list of packages being used by the project. The documentation for huesdk has great tutorials on setting up test programs and code to begin integrating the packages with a project so I have been using those as guidance.

**Satisfaction:**

By the end of week 7, my original project guidelines had established that I would train a new AI model using a website called Kaggle to perform the gesture recognition. However this step was originally added assuming that my first implementation of the project *would not use an AI model*. Under actual development though, I *did* use an AI model so this step is a little bit nonsensical. Also by this point my project would have been integrated and tested on the RPI5 that I planned on using. Since it is still shipping, I have not completed this step, but I have performed testing with an older and less powerful board to a successful degree. The gesture recognizing function already works great and now that I have a library to control device connections (huesdk), I am very confident my project is on track to finish on time or even ahead of time.

**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](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](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](https://docs.opencv.org/3.4/dd/d43/tutorial_py_video_display.html) - Website for OpenCV's example on getting a 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 an RPI over a network
9. <https://pypi.org/project/huesdk/> - Website for Huesdk with tutorials