

Ethan Owen

3/10/2024

EE113DW Status Report 8

Gesture Control for Home Devices Project - (solo project)

https://github.com/Ehanlion/HGR_project

Main Intention:

Week 9 was a culmination week; accumulating and implementing small and various additions and changes that had been worked on in prior weeks to create the most complete iteration of the project to date. By combining and adding these changes together, I created a program capable of controlling my home devices using gestures. This is an amazing and critical step, one of the last I will need to make in development, because it means that the project, save for complete integration on a raspberry pi and final testing, is completed. A nice achievement.

Conceptual Development:

I did no conceptual development this week. I had no need to. In prior weeks I coded and performed research on the side, but that was not necessary this week. Instead I focused on implementation and code design, trying to improve my current program to add robustness and flexibility.

Implementation and Development:

I worked on more advanced functionality with the device-interaction portion of my code. From prior weeks, my code can easily detect gestures within ~10 feet and with good accuracy. Now I needed to use this accuracy to create gesture 'sequences' that my program could understand, interpret, and use.

I sought to implement a color changing function, alongside more complex sequences to turn lights on and off. The color changing function was the main focus of my program. I wanted to allow a user to sweep their hand left and right through the air and essentially 'scroll' through color options on their lights. I used locality and hand position landmarks that my model extracts from each OpenCV frame to store where the hand was last detected within the frame. Then, in the next frame, I can back-compare the current position against a stored position and determine the direction that the hand was moving in.

This method is excellent apart from a few drawbacks. I needed to add a timeout after the program detects a swipe otherwise it freaks out. Additionally, it is a little unreliable as a solution, unlike the on and off functionality which works consistently. This is a problem I hope to address more in the upcoming weeks.

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.

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 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