

Ethan Owen

1/21/2024

EE113DW Status Report 1

Gesture Control for Home Devices Project - (solo project)

Main Intention:

This week was spent improving the original development undertaken during the first week of the project. Within the first week, I installed packages for OpenCV as well as configured my Python environment that I would be using for development. The second week was spent using those installed features to code a basic video capture and hand tracking program. This subtle progress contributes in great amounts toward the goal of the project since the entirety of the project is built upon the backs of a program that can perform video capture and hand tracking. I need a solid base program to build upon and develop which is why this simple work up until now is so crucial.

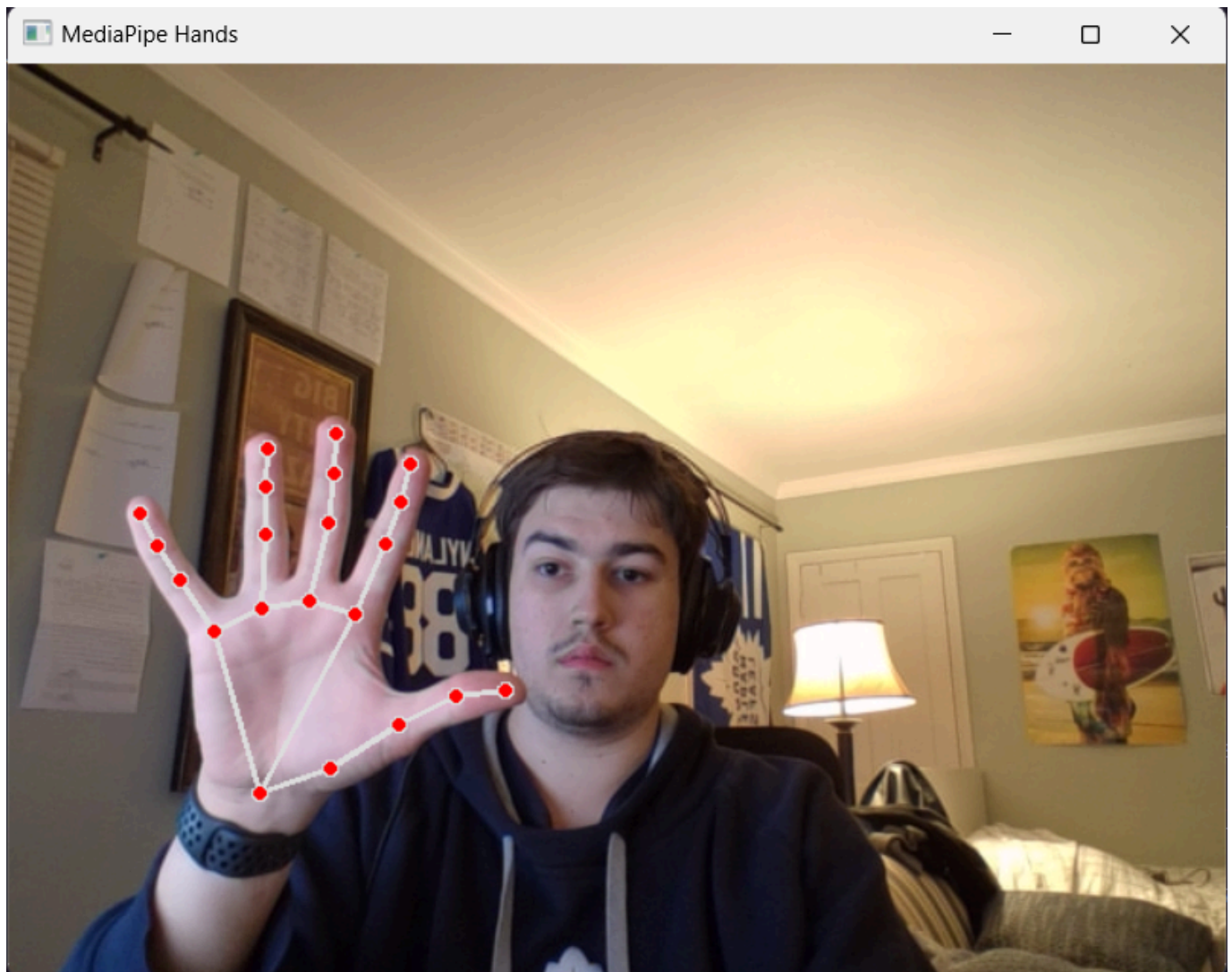
Conceptual Development:

This project uses gesture recognition with the library of OpenCV. While I knew about the library and what its core function was, I have never used it. This week I researched how to use OpenCV to access a webcam and generate a live video feed. I then took my research a step further and began to look at packages that encapsulate the gesture recognition that my project hopes to implement into an easy to use and understand package. Initially, my plan was to hard code the hand tracking for the project which I thought would be tedious, unreliable, and inaccurate. However, I found a package developed by Google called 'Media-Pipe' that has functionality already in-built to perform gesture recognition. It uses a pre-trained artificial intelligence model (by Google) that can recognize seven pre-programmed hand gestures: unrecognized, closed fist, thumbs-up, thumbs-down, pointing up, victory, and rock-on. It also maps the hand into defined regions with tracked points of interest that can be used without the model to examine hand placement and feature placement within frames from a video input.

MediaPipe is an incredibly useful library because the most unreliable component of this project would have been poor gesture recognition and the inability to train a comprehensive model on my own. The introduction of the MediaPipe library will be a massive boost to both the accuracy of my project as well as the development pace since now I don't need to program the hand-tracking and gesture recognition from the ground up and can diversify my project instead.

Implementation and Development:

This week I used OpenCV as well as a new library called MediaPipe to code a program that performs video capture and hand tracking. OpenCV is used to open a video feed by extracting frames from the web camera. These frames are then fed to the MediaPipe framework which handles the hand tracking. We then overlay the MediaPipe tracking onto the frame from OpenCV and display it to the user to visualize what the program is detecting.



This is a freeze-framed image of my program. There are 20 red-dots, tracking components of the hand. It is capable of tracking two hands at the same time although only one is shown. The image was taken by me.

Resources:

To code my program that I developed this week I used example code programs from OpenCV's website as well as tutorials on implementation MediaPipe's hand tracking and gesture recognition framework for a live video feed. OpenCV has widespread documentation and examples on implementing basic functions such as web camera video feeds. MediaPipe has tutorials for gesture recognition and I used the ones for python live video-feeds. These are both incredibly useful because the program I created is not specialized now, it is just a basic building block of the overall project. This means that well regarded examples like the ones I used are incredibly useful in helping start the project development on the right foot.

Individual Contribution:

My team only has one person on it, so everything I detailed and worked on encompasses the entirety of development up to this point.

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://developers.google.com/mediapipe/solutions/vision/gesture_recognizer/python#live-stream - Website for MediaPipe with a python implementation of a hand tracking overlay.
3. https://docs.opencv.org/3.4/dd/d43/tutorial_py_video_display.html - Website for OpenCV's example on getting a video feed functioning
4. <https://github.com/Ehanlion> - My personal GitHub where I am storing all my files for this project in a repository named 'HGR_project.'