Hand Gesture controls for YouTube Media Player

Natnael Bereda

Stony Brook University

CSE/ISE/EST 323: Human Computer Interaction

Dr. Xiaojun Bi

May 10, 2023

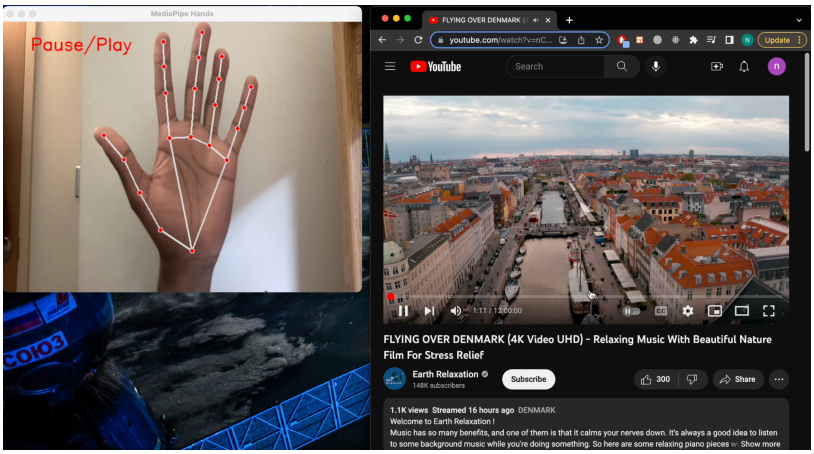
Abstract

Using YouTube media controls on our personal computers can be a problem when snacking, as dirt and grime on the hand can make it difficult to accurately control a touch screen, keyboard or trackpad. This can be a frustrating experience that can negatively impact their user experience which can cause accidental taps or swipes and lead to unintended actions on YouTube and can lead to our device being unsanitary. Voice controls are built into YouTube for searching, but YouTube currently does not provide gesture control functionalities. Approach to this problem would be implemented mainly on personal computers with webcams. It would be to use image processing algorithms and deep learning to identify the different shapes of the user's hand, and perform the specific action based on each individual shape like Play, Pause, Skip, Full screen, Mute and Volume up/down on a video. The concept, although not always accurate, was successful in adding gesture controls to YouTube's already existing controls, and can be applied to other media players with the same build.

Keywords: Hand Gesture controls, YouTube, image processing, deep learning

Introduction and motivation

YouTube provides two main ways we can interact with the media player: we can either use our keyboard shortcuts or mouse/touchpad. Hand Gesture controls add on to these controls and give users more options. Gesture control technology is used in a wide variety of applications, including gaming, virtual reality, and robotics. Gesture control technology has rapidly evolved from primitive input devices to fine detail recognition. It has been used in a wide range of applications, from research experiments to day-to-day commercial products. The development of gesture control technology can be traced back to the 1980s and has since been used to help disabled individuals and has been tested in various prototypes, including hands-free intelligent wheelchair control systems(Gesture Control Technology: An investigation on the potential use in Higher Education.) Today, the development of deep learning and computer vision algorithms has made it possible to recognize complex hand gestures and movements, opening up new possibilities for the use of this technology. There are multiple reasons and motivations to build a hand gesture system like Convenience, Accessibility, Novelty, and Future-proofing. In terms of Convenience, we can easily navigate through their media player without having to physically interact with the device, making it more convenient. For example, while snacking, Hand gesture controls provide an alternative way to keep our devices sanitary from greasy or unsanitary hands.



For individuals with physical disabilities or limitations, a hand gesture control provides an alternative way of controlling media playback. When it comes to Novelty, Hand gesture controls are a unique and innovative way to interact with media, which can make using a media player more engaging and exciting. Hand gesture controls can also be seen as an attempt of Future-proofing, as technology continues to evolve, hand gesture controls are becoming more common and may become a standard way of interacting with devices in the future. Building a hand gesture control for a media player now can help future-proof the device and ensure that it remains relevant and usable in the years to come.

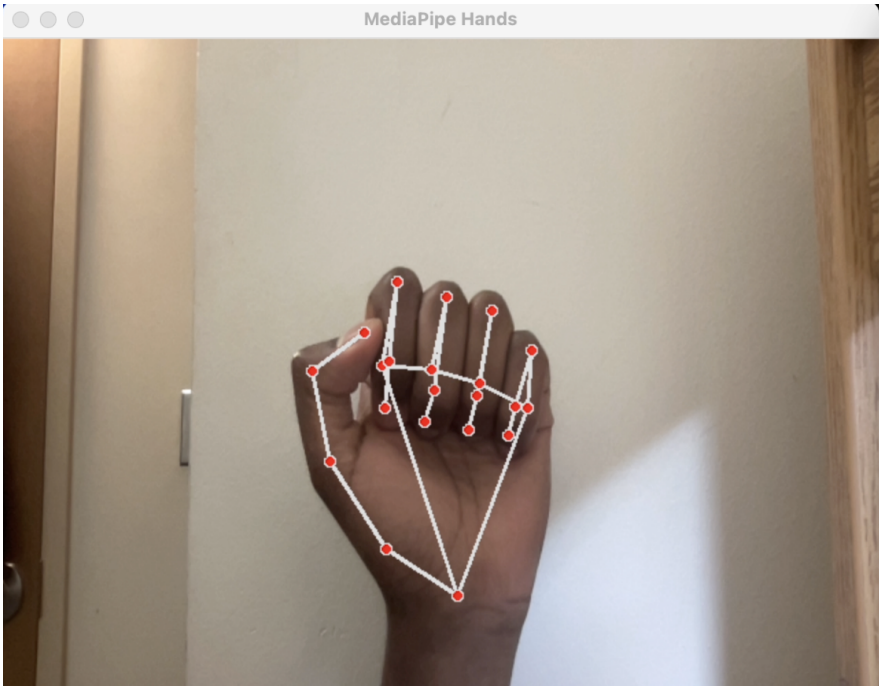
Related works

Multiple hand gesture control technologies exist that are not specifically built for YouTube media player, but perform the same operations for different technologies. Some relevant hand gesture control technologies include Kinect for Xbox, Leap Motion, Myo Armband, and Google's Soli radar technology. Although Discontinued, Kinect for Xbox is a motion sensing input device by Microsoft for Xbox 360 and Xbox One video game consoles. It uses cameras, microphones, and software to track body movements and voice commands, enabling users to interact with their console without the need for a physical controller. The device is widely used not only for gaming but also in other areas such as fitness, education, and healthcare. Leap Motion is a small device that uses cameras and infrared sensors for hand tracking and to enable gesture controls for media players and other applications. It has been used in a variety of applications, including gaming, virtual and augmented reality, and healthcare for fine motor skill rehabilitation. Myo wristband is a wearable gesture control device that uses electromyography (EMG) sensors to detect muscle movements and translate them into digital

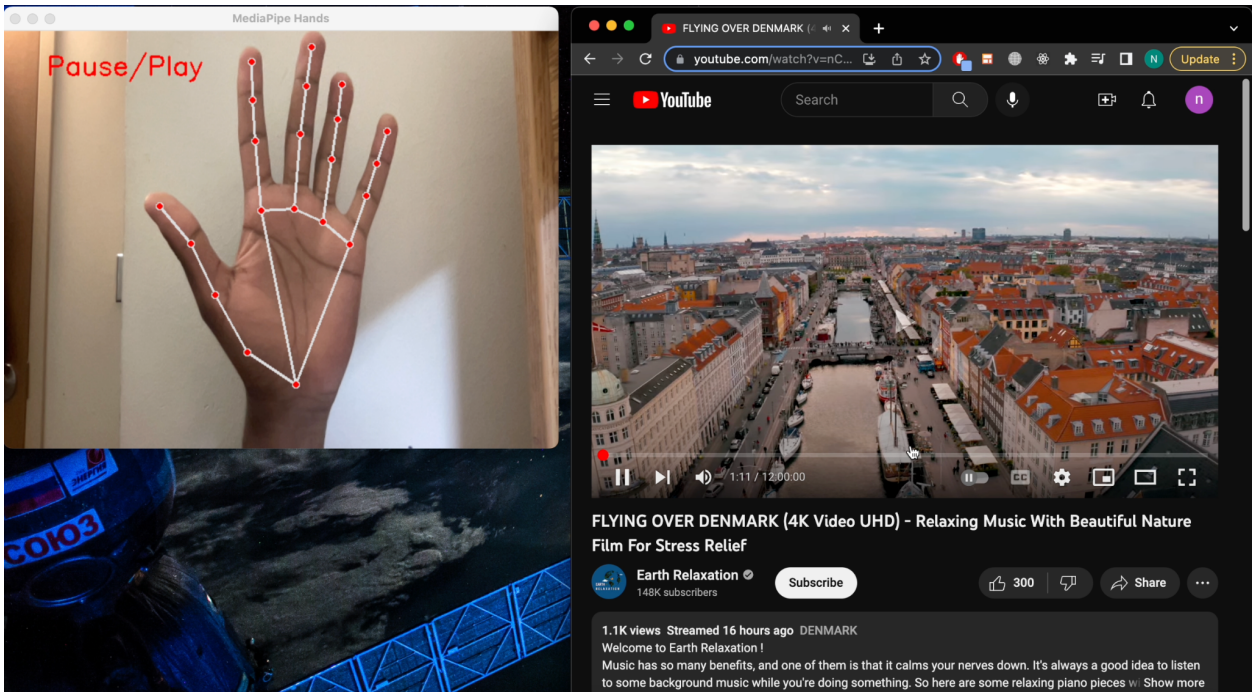
commands. It can be used for a variety of applications, including controlling computers, mobile devices, and smart home devices, as well as in virtual and augmented reality environments. Google's Soli radar technology, more specifically, Nest Hub, integrates Google Assistant with a touchscreen display and can be used to control various smart home devices, play music and videos, display photos, and provide helpful information such as weather updates and recipes. It also supports voice and gesture control. Overall, while there are similarities between this project and existing technologies, each has its own unique approach and strengths. This hand gesture control project may offer a more accessible and affordable option for users who want to control media players with hand gestures, as it can be built using common hardware and open-source software(Python) but it still provides less quality, capability and reliability as the other technologies.

Technical “meat”

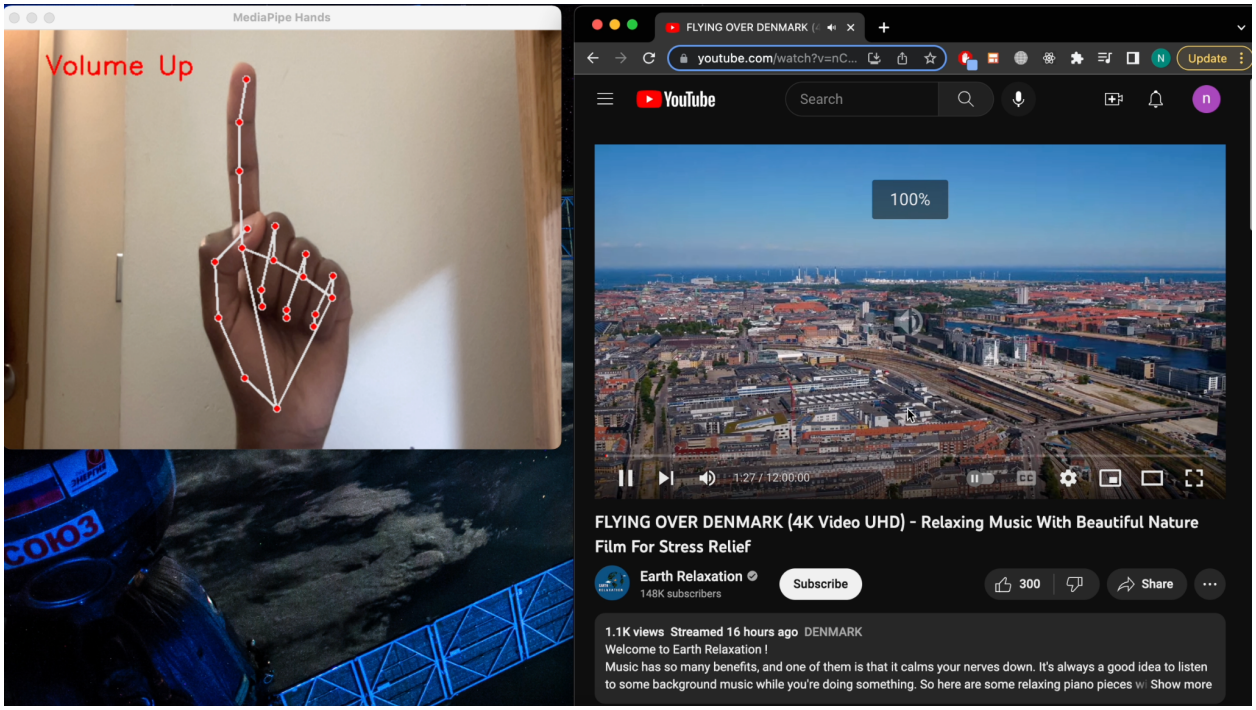
The program is developed using Jupyter Notebook and implemented in Python 3. Required libraries for the project are Numpy, OpenCV, Mediapipe, Pyautogui, Time, and Sys. The hand detection algorithm is written for the right hand, with the palm facing the camera. After successfully running the program, make sure the browser window is focused in the YouTube media player since the program automates keyboard presses based on the different gestures. The media player controls include Pause/Play, Skip video, Volume up/Volume down, Mute, Seek left/right, and Full screen control. The gestures associated with each control are shown as follows:



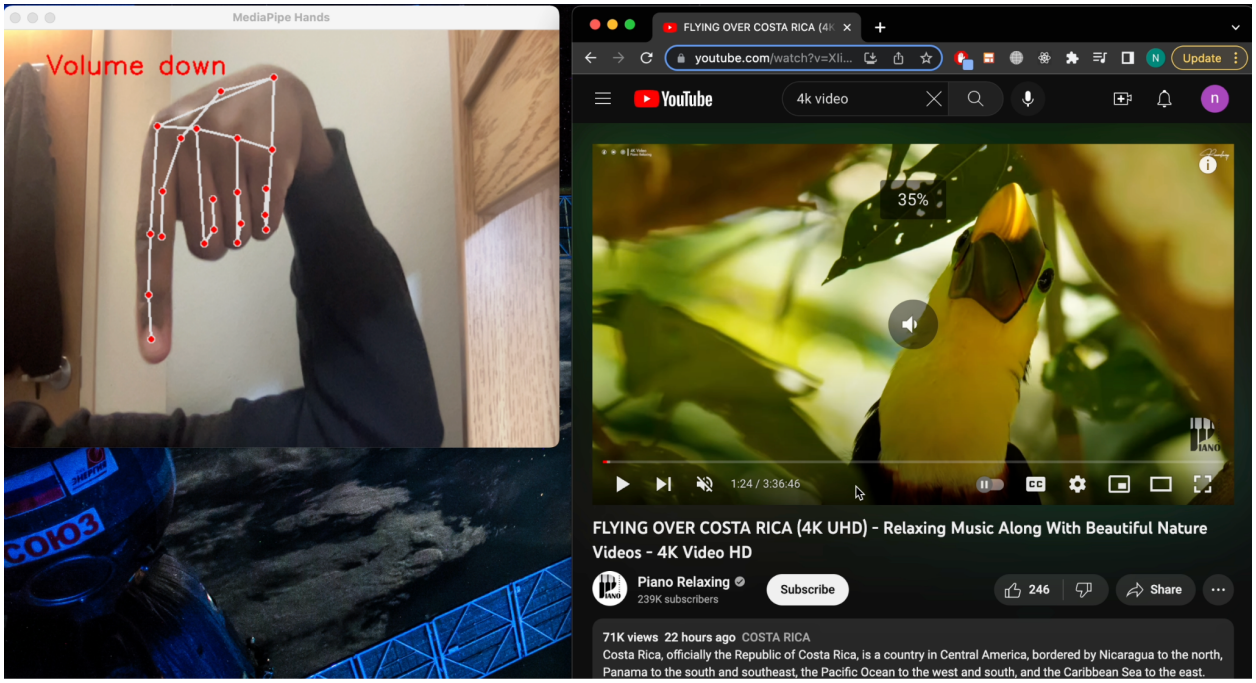
Closed palm position (Default position/ No action)



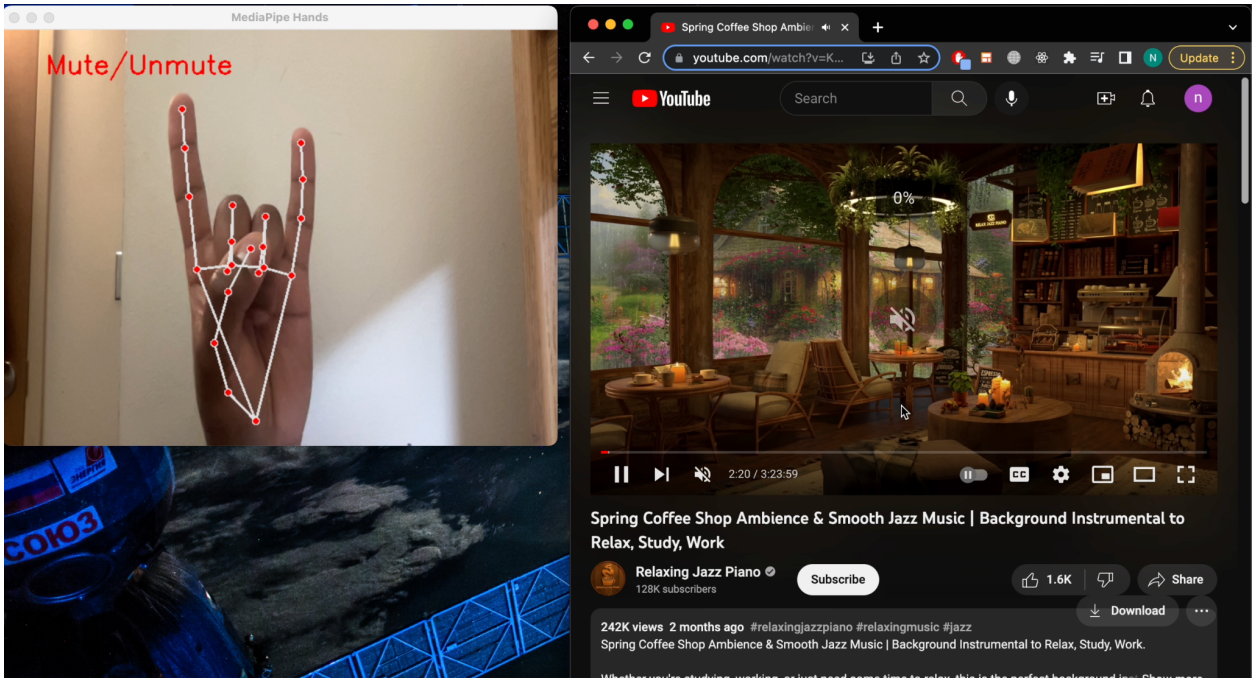
Open palm position (Pause/Play controls)



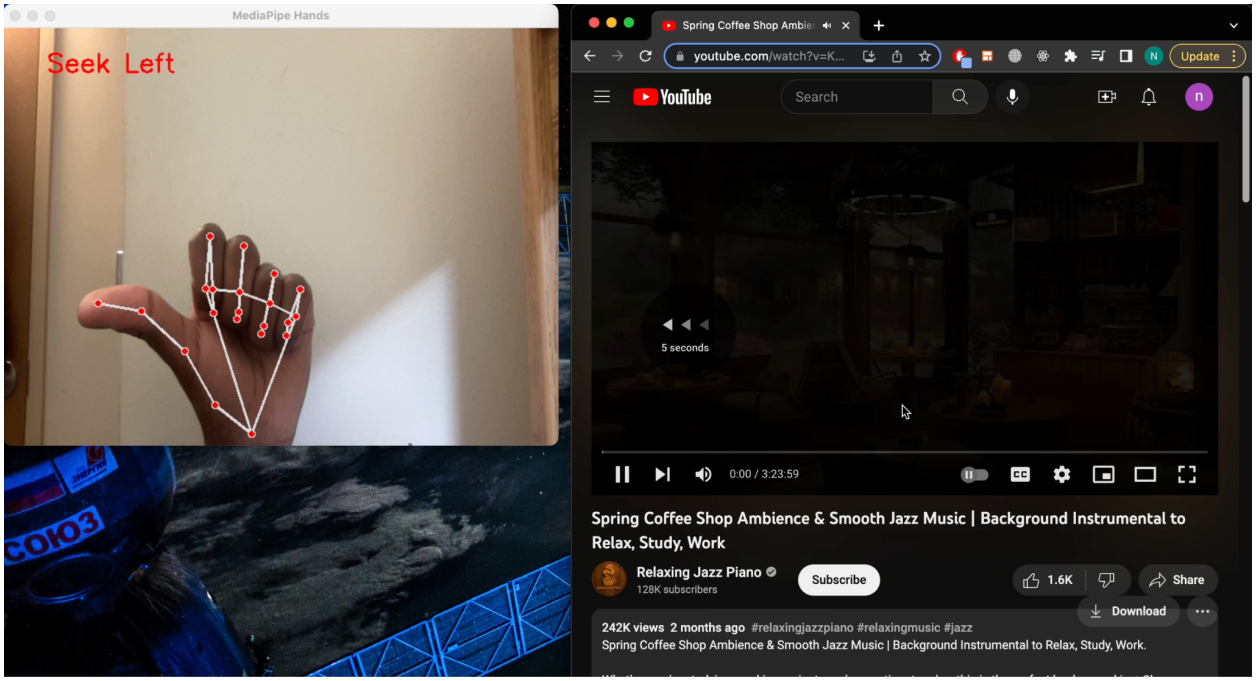
Index finger extended (Volume Up control)



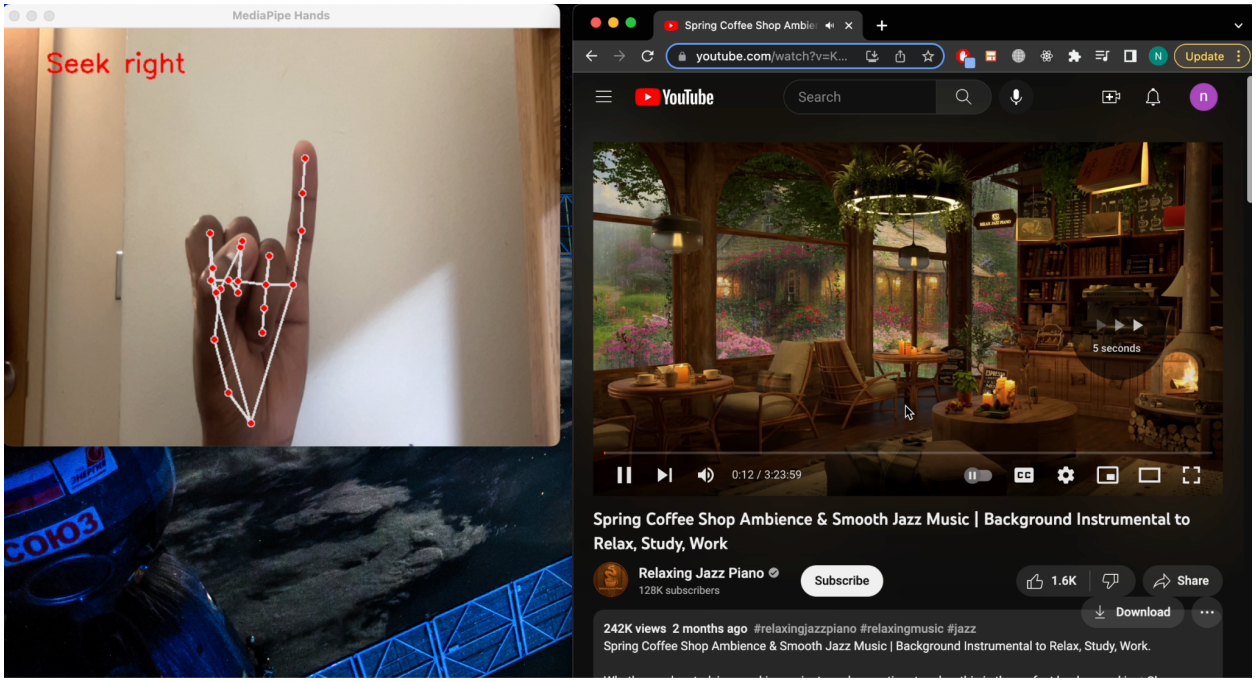
Index finger pointing down (Volume down control)



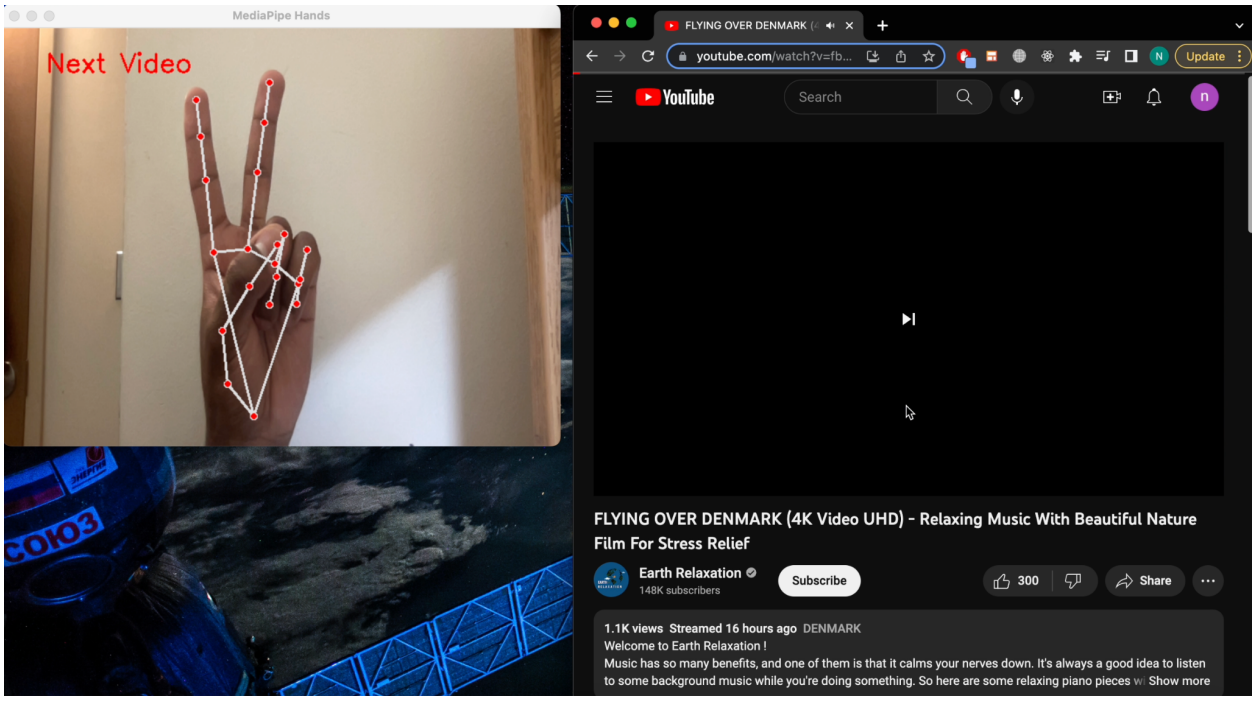
Love you gesture (Mute/Unmute control)



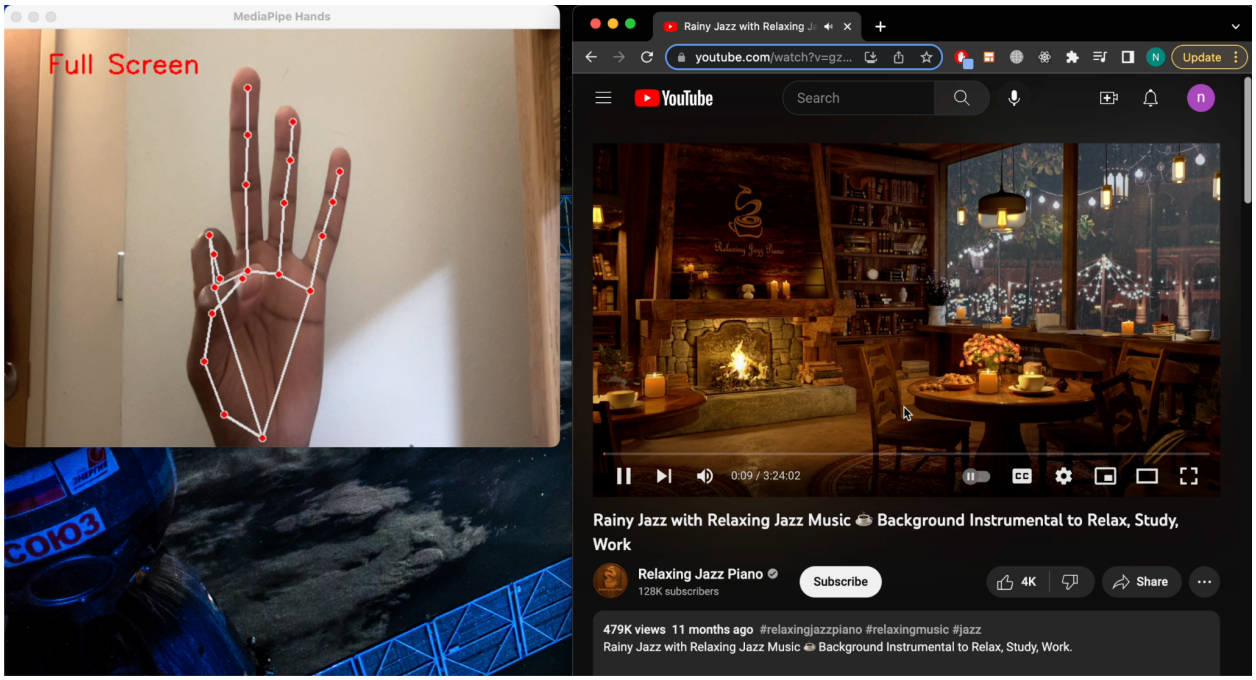
Thumb pointing to the left (Seek Left control)



Pinky finger extended (Seek Right control)



Peace sign (Skip to next video control)



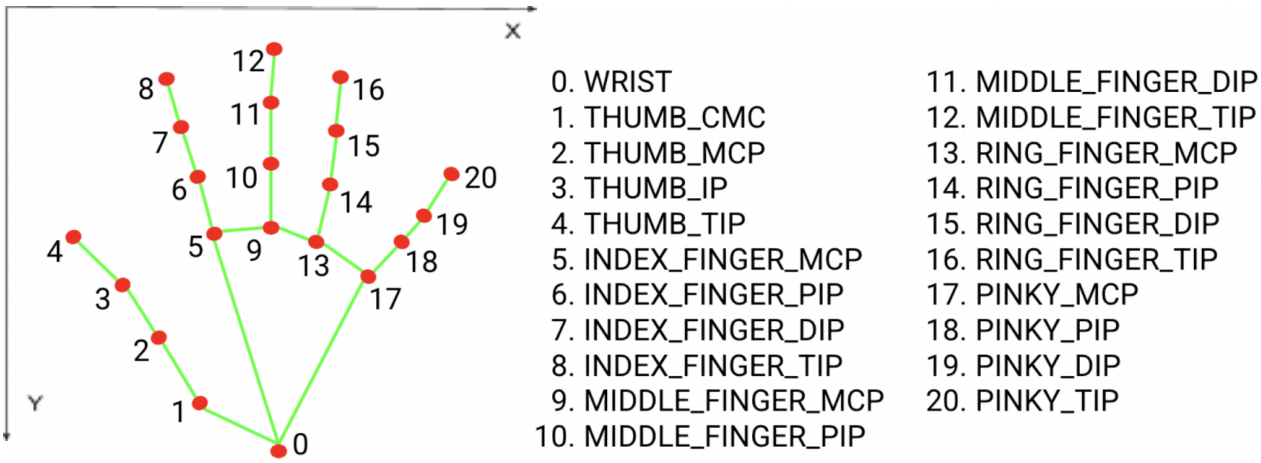
Middle, Ring and Pinky fingers extended (Full screen control)

Implementation

OpenCV library used for real-time computer vision applications. OpenCV is used in the program to read video frames from the camera, flip the image horizontally, and convert the image color space from BGR to RGB for compatibility with Mediapipe. OpenCV will continue to take inputs from the webcam till the program is interrupted by the user.

MediaPipe is the python library that will be the heart of the hand gesture control program. Mediapipe is a framework developed by Google that provides a pipeline for building computer vision applications. It includes pre-built models and processing modules for various tasks such as hand tracking, pose estimation, face detection, and more. In this program, MediaPipe is used to detect and track the hand landmarks in real-time video frames. This is done by using the hand landmark model bundle which can detect the keypoint localization of 21 hand-knuckle coordinates within the detected hand regions. The palm detection model localizes the region of

hands from the whole input image, and the hand landmarks detection model finds the landmarks on the cropped hand image defined by the palm detection model.



Once we have the hand landmarks, We can use the hand landmark coordinates to distinguish between the different gestures. For example, to check if the index finger is raised, since the Y-axis is inverted we check if the INDEX_FINGER_TIP is less than the INDEX_FINGER_MCP joint, and check if the Ring, Middle and Pinky fingers tips are greater than their respective MCP joints; if it is the case that means only the index finger is raised and we can associate that gesture with an action.

Once we can identify the different Gestures using MediaPipe, the next step is to control the YouTube Media player based on those gestures. PyautoGUI is used to Send keystrokes based on the current Gesture. PyAutoGUI is a library that lets our Python scripts control the mouse and keyboard to automate interactions with other applications; This is why it's important that the browser window with the YouTube video we are currently viewing is focused on, since we will use PyAutoGUI to simulate keyboard presses. Numpy library is used to store the coordinates of the finger joints, and the Time library introduces delay between recognitions to give the user and PC enough time to switch between gestures.

Validation

The hand gesture control system demonstrated promising outcomes with gestures accurately recognized under varying lighting and background conditions. Limited Testing done with users of diverse arm shapes and sizes, a female friend with a smaller arm and a male friend with a medium-sized arm, also yielded satisfactory results. However, enhancements are required to improve the accuracy of recognizing certain gestures that utilize the same fingers, which may result in misinterpretation.

It is worth noting that the system assumes users possess functional fingers, and hence, may not be fully accessible to individuals with disabilities. Therefore, future work could focus on expanding the system's accessibility to users with disabilities, such as exploring alternative gesture recognition models that rely on different types of input or expanding the system to recognize gestures made with prosthetic devices.

Technical prerequisites for using the system include the installation of Python and availability of a webcam. Furthermore, compatibility issues may arise while working with the current version of Python and integrating mediapipe and opencv libraries, thus presenting further challenges. Future work could explore ways to improve the compatibility and ease of installation of these libraries, potentially by creating a user-friendly installer or providing detailed instructions for troubleshooting common issues.

Discussion and/or future work

The project successfully demonstrated the capability of using hand gestures to control media players. It is worth noting that this system can be extended to other media players that use the same keyboard shortcuts for media control. Although I wanted to develop a chrome

extension for wider accessibility, technical difficulties were encountered while working with the YouTube API and Chrome manifest V3 extension builder, leading me to abandon the idea. The TensorFlow JavaScript library also presented issues, as it was not working properly, and the chrome extension script I wrote for the YouTube page was not functioning correctly, despite extensive troubleshooting. However, further work can be done to improve the program's inclusivity and accuracy by detecting gestures for both arms. Additionally, pre-trained hand gesture models can be further incorporated in the future to increase the accuracy of the gesture predictions. These enhancements will improve the system's accessibility and make it more user-friendly for individuals with disabilities.

Conclusions

In conclusion, the implementation of hand gesture controls for YouTube using image processing algorithms and deep learning is a feasible solution for improving the user experience and accessibility of YouTube. The project successfully added gesture control functionalities to YouTube's existing controls, allowing users to perform specific actions based on different hand shapes, such as play, pause, skip, full screen, mute, and volume up/down. While this project offers a more accessible and affordable option for controlling media players with hand gestures using common hardware and open-source software, it still provides less quality, capability, and reliability compared to other technologies like Kinect for Xbox, Leap Motion, Myo Armband, and Google's Soli radar technology. Nonetheless, this project demonstrates the potential for hand gesture controls in media players and other applications, and with the continuous advancement in deep learning and computer vision algorithms, hand gesture control may become a standard way of interacting with devices in the future.

Acknowledgements & References

https://intranet.birmingham.ac.uk/it/innovation/documents/public/Gesture-Control-Technology.pdf

https://developers.google.com/mediapipe/solutions/vision/hand_landmarker

https://www.ultraleap.com/product/leap-motion-controller/

https://time.com/4173507/myo-armband-review/

https://atap.google.com/soli/#nest-hub

https://opencv.org/