Gesture controlled environment for a hands-free computational experience using Computer Vision

Prithwish Ganguly[1] and Dr. Biswajit Biswas[1]

[1]Ramakrishna Mission Vivekananda Centenary College, Rahara

Abstract

Gesture recognition is a rapidly growing field in the domain of human-computer interaction, with applications ranging from sign language recognition to virtual reality control. The system is designed to accurately detect and classify a variety of gestures performed by users in real-time. In this project, we develop a tool for mouse control using hand and eye gestures, and letters and numbers typing using hand gestures. Also, we develop a small voice assistant tool that can perform multiple tasks hands-free. In order to train the model for identifying letters and alphabets from hand gestures, we use a dataset of hand gesture images collected from various individuals and use Convolutional Neural Network (CNN) to find the most suitable model. The system is able to accurately classify a wide range of gestures with high precision, making it suitable for real-world applications. Overall, our project demonstrates the potential of machine learning in developing advanced gesture recognition systems that can enhance user interaction with digital devices and applications.

1 Introduction

The modern era is all about innovations. With the advancement of Artificial Intelligence, the way we interact with devices is constantly evolving. Gesture control is one such evolving area in the realm of Artificial Intelligence. It is a technology that interprets human gestures and movements using mathematical algorithms. It is a way for computers to understand human body language and reducing the need for other devices for computer interpretation. Nowadays, gesture control technology is used in a wide variety of fields, including, but not limited to, gaming, Virtual and Augmented Reality, touchless medical imaging in healthcare and robotics. Gesture control is also a very effective tool for people with physical disabilities and provides them with a method of interaction.

Interaction with a computer for any regular individual involves browsing the web, creating and writing documents etc. All activities involved with such interactions can be broadly classified into two major groups: using computer mouse for left click, right click, scroll; and using keyboard for typing. Besides, using voice commands to perform several activities is another area that is worth exploring. Our aim in this project is to combine all these areas and develop a tool that can translate hand gestures into corresponding mouse movements and typing actions, translate eye gestures into mouse movements, use voice for activity control. We envision this tool to be highly suited for people with certain disabilities where they can simply do certain gestures for computer interaction according to their convenience without any need to even touch the keyboard or mouse.

Mouse control with gestures is a very relevant area in modern day gesture recognition studies. There are a number of works in this area. We have seen advancements in VR ar AR technologies, Automated cars which track and learn its surroundings while moving. We make use of some of the existing works in our project as well, with some innovations.

As mentioned earlier, we also want to enable gesture-controlled typing. The idea is to make use of Convolutional Neural Networks (CNNs) to learn gesture images, and then type the corresponding English alphabets or numbers. The CNN is trained on a suitable image data for the purpose.

Voice assistant technology allows users to interact with devices using natural language, making tasks easier and more efficient. It can help users with a variety of tasks such as setting reminders, playing music, answering questions, and controlling smart home devices. By utilizing them, we can save time and increase productivity by completing tasks hands-free. Voice assistants can also provide personalized recommendations and information based on user preferences and behavior. Overall, this technology can improve the user experience and make daily life more convenient. In this project, we aim to develop a small voice-assistant for a hands-free experience.

The rest of the article is organised as follows. In Section 2, we discuss the methodologies we used to develop the gesture controlled mouse, gesture controlled typing and Voice Assistant. In Section 3, we show the performance of each of the proposed technologies in real applications. In Section 4, we discuss some possible future directions for our project.

2 Methodology

We now move into more details regarding the working of our proposed tool. The tool will be diverse and easy to use in most cases. It will be a system app and will really benefit many people. There is also an advancement in Augmented Reality and Virtual Reality apps in this current generation. The prices are at stake. Most people have a PC or a laptop. It has become a necessity in today's life. That same powerful device may have a camera attached or one may buy one externally. This project is all about a glimpse of what the future will hold for us.

Suppose we need to sign an important document. We can simply move our hands in the air but it will turn out to be the same, old signature. A simple camera will capture and process it. A camera is also a necessity nowadays. We can call and video chat with our friends and family easily and it is so much simpler to stay connected. Regarding computer usage, we can already envision a future where there would not be a mouse or a keyboard. For example, we already have gaming consoles that are more accurate and enjoyable than gaming using the traditional keyboard and mouse. Typing and scrolling are those actions which are thought to be possible nowadays using a camera, maybe with some gestures, which our highly capable sophisticated cameras will process and give us a result. The only thing to keep in mind is to assign a unique gesture to a unique activity. One might ask whether the innumerable things that a keyboard can be easily

implemented. Our answer is Yes. We do not know which key does what. It is just an innovation like any other. Keyboards and Mouse will not be obsolete but the majority will not require them. Slowly and steadily, people will adopt and catch up to the innovation, they will learn and will keep learning as long as we developers will keep developing.

In the tool we have developed, the user will be able to perform some basic functions like scrolling and typing using hand gestures. This will be a PC app and the user can move the mouse cursor, and can Single Click or Double Click, depending on the system they are using. If we need to type something, we can write in the air and the trained models will evaluate and get the result and write it in the required area. We also provide the user the flexibility of using their eyes as the mouse cursor with Single Click and Double Click features enabled. In our project, we have also made a voice assistant, which can play the music the user likes and can be used for knowing anything or for basic automations. We now look at the specific functions in more details.

2.1 Mouse control

This is the first part of our project where we had to use basic OpenCV functions for image processing, mediapipe functions for capturing the hand gesture. Mediapipe is extremely useful as it can be used to capture and detect our required movements and indices precisely. It uses its trained deep learning model to process frame by frame and is easy to implement. Mediapipe is a GPU demanding package and hardware specifications matter a lot here. We am using Intel i5-10500U processor with integrated GPU of 128mb. This is a CPU and GPU intensive task as CPU processes the frames and the GPU loads them, so in our setup it will likely be a bit slow. Hence, more optimisation with proper implementation is required. We have enabled two types of mouse control in the proposed application. We discuss about them in more details in the next two subsections.

2.1.1 Finger gesture

In the implementation of finger gesture, we can move the mouse cursor with the help of our index finger. We used the necessary libraries like OpenCV, Pyautogui and Mediapipe. It was a challenging project and we implemented Left click on our own and took help from existing projects from the internet for implementing the scroll and right click. Initially, the code has camera access once it is run and it captures the movement frame by frame. We used Mediapipe's solution package to detect the hand in every frame. We also had to flip in Y-axis to show our reflection image as we see in a mirror. By default Opencv uses BGR but we had to convert it to RGB for proper color manipulation. After passing the frame through processing, we had to use iteration for finding hand landmarks at each point. We did not use any depth measurement so there is nothing to worry about Z-axis. This provides us with the pixel values of the tip of the index finger as we move across

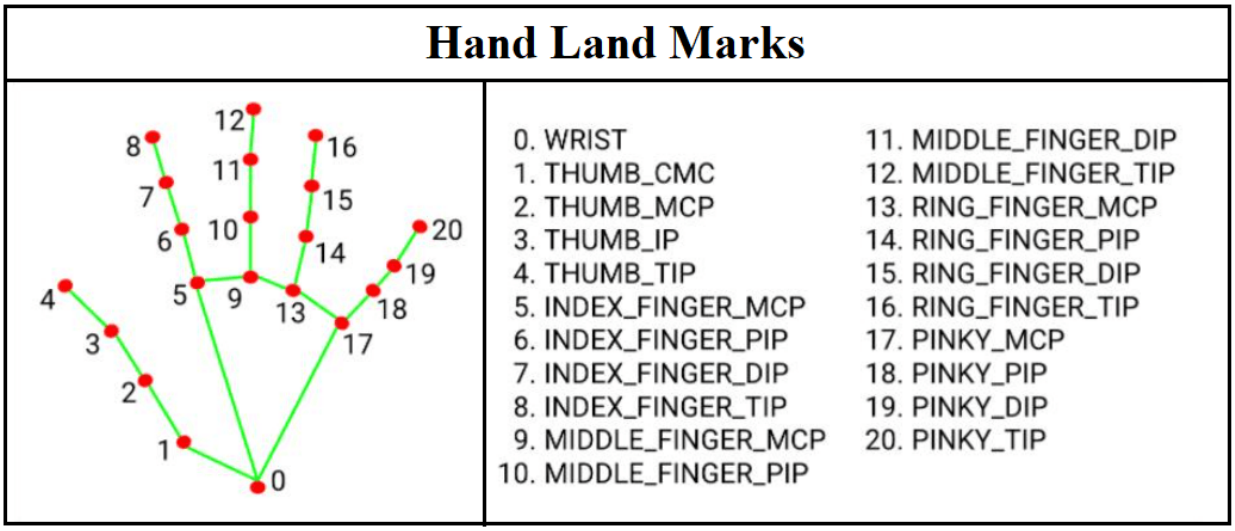

Figure 1: Illustration of hand gesture for mouse control

the screen. There is a limitation to this procedure as it only moves in the specified frame. So we need to move with respect to the ratio of our actual screen. Hence we had to calculate and multiply that as a scaling factor to find the actual X and Y coordinates.

As we can see the tip of the index finger is marked as '8' so mediapipe will track that position frame by frame and track it. If we want to use a single click we can bring out our thumb which has an id of '4' close to the tip of the index finger i.e if the distance between them reaches below a significant value across y axis, we can perform a single click. Pyautogui uses the click function. Then consecutively we had to provide a sleep function to stop it after single click for a significant amount of time. On pressing Escape the frame window will close and it's execution will be complete.

2.1.2 Eye gesture

Here we discuss the alternative implementation of mouse control, which is by eye gestures. We provide the user the flexibility of using either of hand or eye gestures according to their convenience or limitations. In this implementation, as in finger gesture, we have shown single click only. For this, we need OpenCV, Pyautogui and Mediapipe libraries for Camera control and image processing, for automation and for movement detection respectively. This program needs a function of mediapipe called $face\_mesh$. It captures every face landmarks, which we need to process then.

As we can see in Figure 2, the right eye of the person has landmarks 474 to 478 at 4 corners of the iris, which we have marked and they get captured in every frame and we look for their sudden movement. This solely requires a good camera which is hard to get on Work Laptops. This technology is widely used in Virtual Reality Headsets, for example, in Apple Vision Pro. This

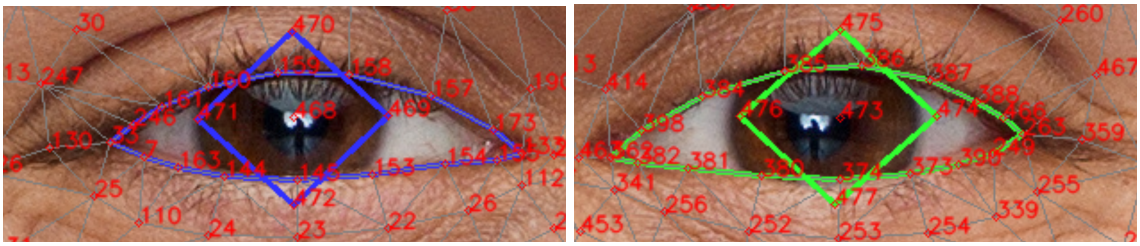

Figure 2: Illustration of eye gesture for mouse control

technology is easy to implement without much cost and can give a competitive edge in shooting games. With a little bit more optimisation this will be good to go. As for the left eye which works on winking as the landmarks 145 and 159 come close, we had to calculate the absolute difference between them and on reaching a minimum limit, the click is supposed to work followed by a significant gap. This is not a recommended to use at the moment because it can cause eye strain and rapid winking. As usual, if one must terminate the program, one must press escape key in windows. A point to remember is that, the key match for MacOS and Windows may not match most of the time.

2.2 Alphabets and numbers typing using hand gesture

We have already discussed about the controlling of mouse. However, the first thing that comes to mind when we think about creating a hands-free environment is ways to provide hands-free typing. While accessing a computer, when the pointed cursor changes to 'I', we know that is when we must type. Our application will also have the ability to understand that and shift the functionality from mouse control to typing when the cursor changes to 'I'. Suppose we want to type any of the 26 alphabets (not case-sensitive) and 10 numbers. We can represent them by 36 digits (0-9 for the numbers, 10-35 for the 26 alphabets). Similar to the mouse control, camera is essential for gesture-controlled typing. Our program requires use of Python modules such as OpenCV, Mediapipe and Numpy, and the captured gesture is first drawn on a black canvas. We draw in front of our camera and then it gets merged with the black canvas as they are on the same frame. Mediapipe detects the tip of the index finger when we need to type. So then we can draw the alphabet or the number we need. Our tool will then process the drawing to an actual English letter or digit.

The machine must learn automatically from the drawn picture and recognise which letter is it. So, we had to use some existing dataset to have the machine learn the patterns. For that, we used a dataset from Kaggle consisting of 442,450 28x28 images and their classification into one of the 36 classes. We used 90% of the data as the training data. As pointed out by many, CNN models are extremely useful in image classification, and that was our primary goal here as well. So, we created

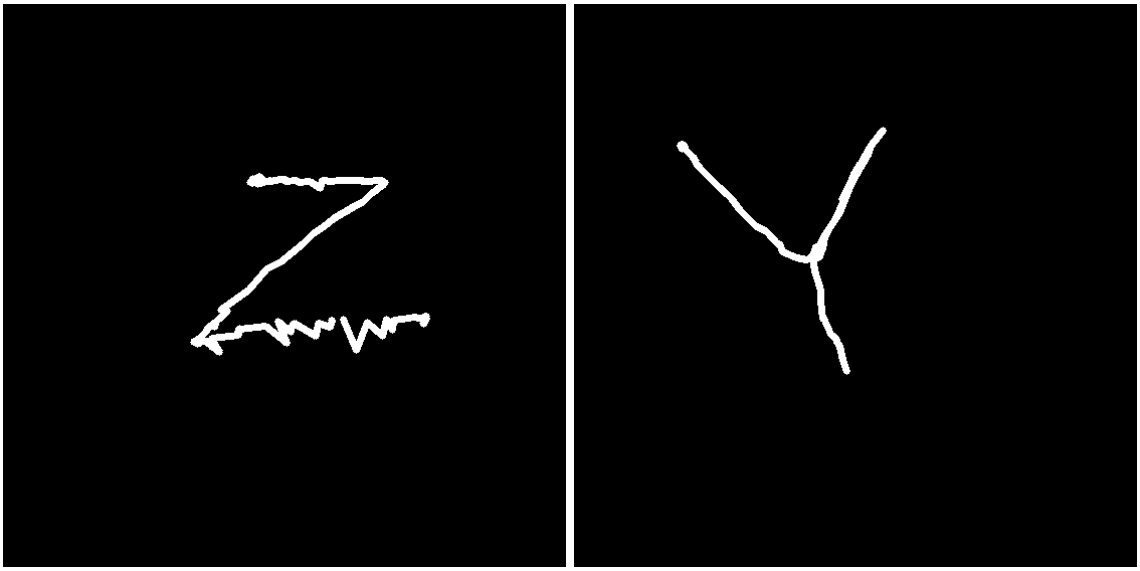

Figure 3: Captured images of the letters Z and Y

a Convolutional Neural Network (CNN) model which can predict one of the digits or alphabets given any captured image similar to Figure 3. Our model achieved around 98% accuracy on the training data. We used 'keras' as the deep learning framework and 'matplotlib' for the purpose of plotting an image from 1D array. Next, we provide more details about the data used and the choice of the CNN model. Note that for the CNN model, we did some trial and error with the choices of the hyperparamters before coming up with the proposed choices.

1. Data description and pre-processing: As was previously mentioned, the dataset consists of 4,42,450 images each of resolution 28x28. It iss divided into 36 classes with 0-9 denoting the corresponding numbers and 10-35 denoting the alphabets A-Z. We performed one-hot encoding on the response classes to nullify numeric value prioritization. The pixel data were scaled in the range of 0-1 by min-max scaling for proper evaluation.

2. Choice of CNN Model: We conducted a trial-and-error with the number of dense hidden layers, the number of neurons in each hidden layer, the number of convolution filters and pooling layers, and dropout proportion. The final choices of parameters are as follows:

    - Number of filter layers: 3 of size 3×3
    - Number of filters in each layer: 32
    - Pooling type: Max pooling
    - Dropout: 20 % dropout after first pooling layer

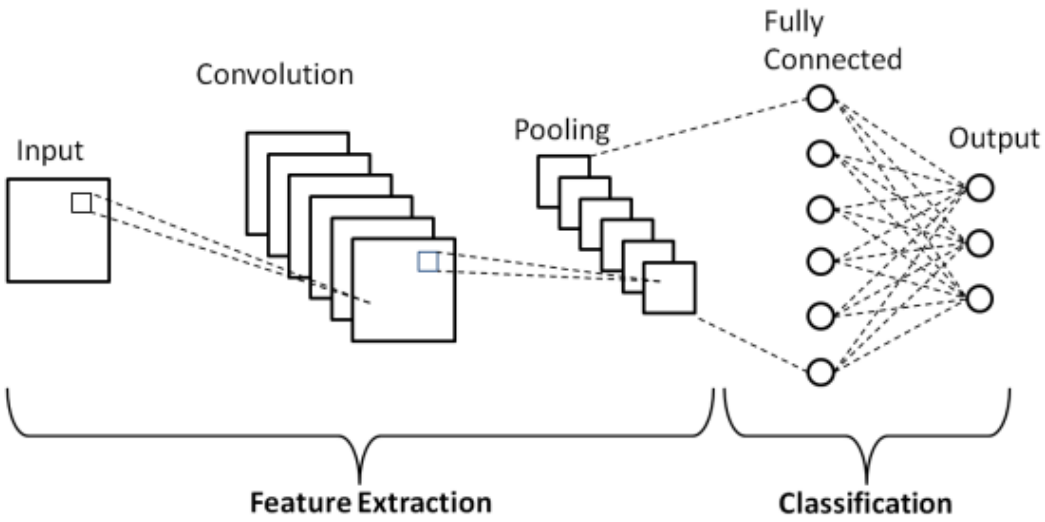

Figure 4: Representation of a typical CNN

- Number of hidden dense layers: 1 with 70 neurons.
- Optimizer: Adam
- Loss: Categorical Crossentropy
- Hidden layer activation function: ReLU
- Output layer activation function: Softmax
- Padding: Same

2.3 Voice Assistant

So far, we have discussed typing and mouse control using hand or eye gesture. However, such gestures can be a bit restrictive regarding what they can perform. A more flexible way of dealing with a computer without hands is by voice command. Voice command can help perform several activities, and is well suited for different functionalities For example, in a scenario when the user wants to hear some music, it should be straightforward with a Voice Assistant. In this project, we have implemented a small voice assistant which recognises the voice and converts it to a string which can then be manipulated, and based on communications with the database, it will act accordingly. This compact, multifunctional voice assistant is designed to simplify the user's daily routine and keep him entertained.

With the ability to search Wikipedia, this voice assistant provides instant access to a wealth of knowledge on a wide array of topics. Whether one is looking to learn about historical events, scientific concepts, or famous personalities, the voice assistant can quickly retrieve the information

one needs, making it an invaluable tool for both work and leisure. Not only does this device cater to one's thirst for knowledge, but it also serves as a source of entertainment. With the capability to play songs, the user can easily enjoy favorite tunes with a simple voice command, creating a seamless and enjoyable listening experience. Additionally, the voice assistant is programmed to deliver a lighthearted touch to the day by sharing jokes, adding a dash of humor to your daily interactions.

In addition to its educational and entertainment features, the voice assistant offers practical utility by providing the current time upon request. Whether one is in the midst of a busy schedule or simply want to stay on top of his day, this function ensures that the person is always aware of the time, allowing for better time management and organization.

Compact, versatile, and user-friendly, this voice assistant is the ultimate companion. Its diverse range of capabilities makes it an indispensable tool for anyone looking to streamline their daily tasks.

3 Results

In this section, we illustrate how our proposed methodologies perform. In Section 3.1, we can see how the hand or eye gesture image is first captured by the computer from the user before further processing. Section 3.2 illustrates a scenario where a gesture of a letter using hand is successfully captured by the computer, transformed into an image, and then identified as the letter 'T' with the developed CNN model running in the background. Finally, in Section 3.3, we give a few examples of how the developed Voice assistant works and how it is able to successfully carry out all the instructions.

3.1 Mouse control

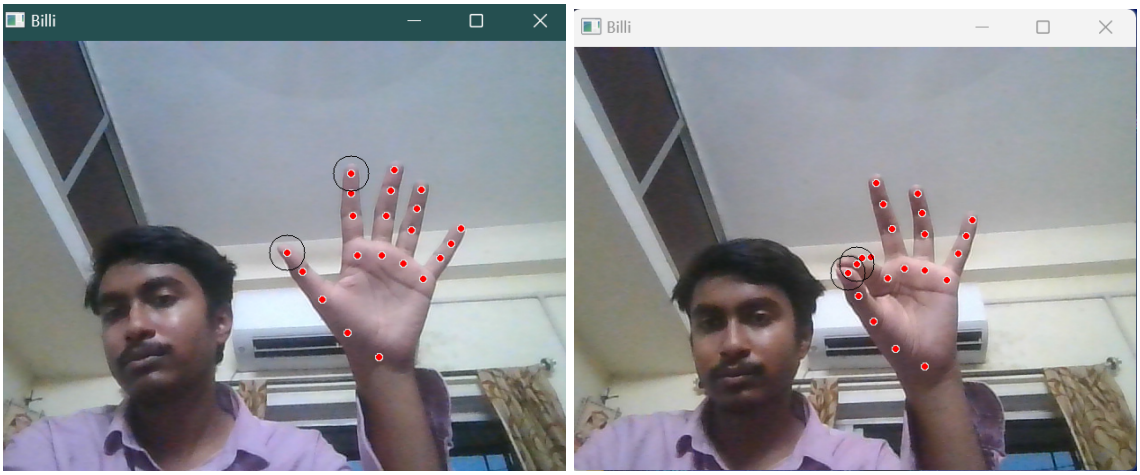

Figure 5: Hand gesture

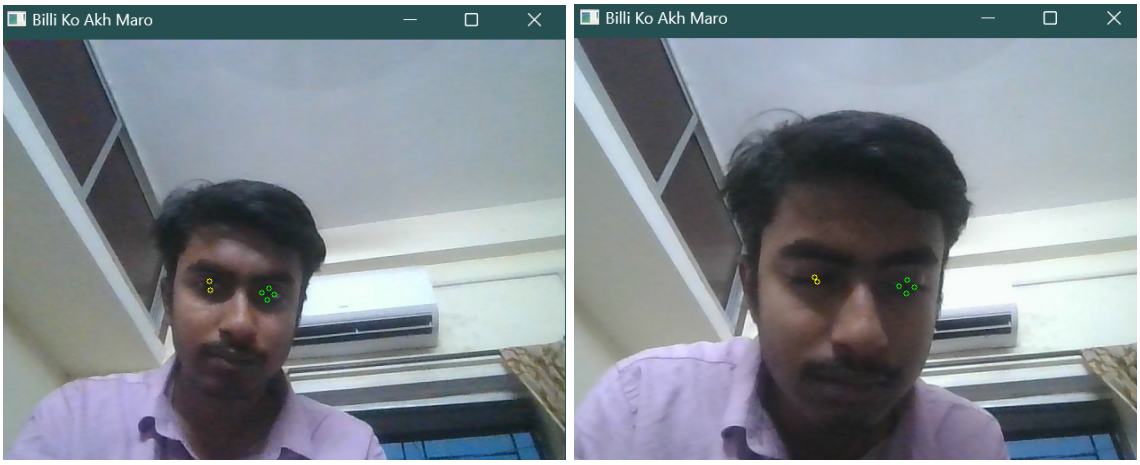

Figure 6: Eye gesture

3.2 Typing

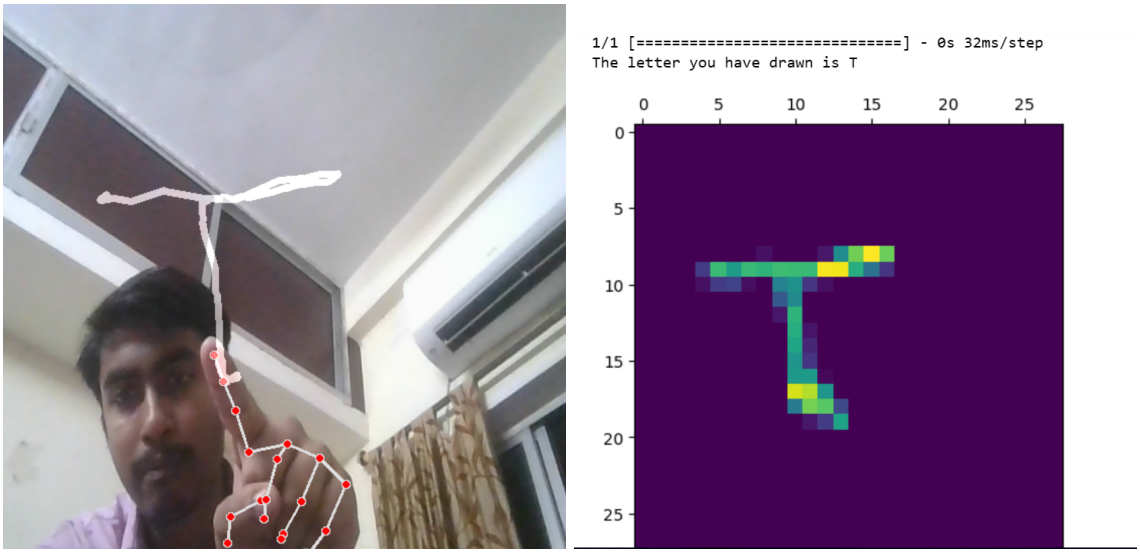

Figure 7: Typing using hand gestures

3.3 Voice Assistant

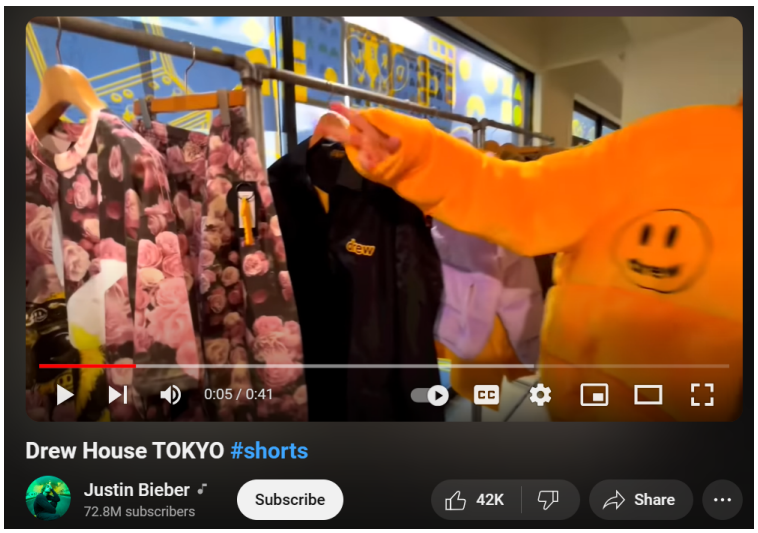

Figure 8: Search result

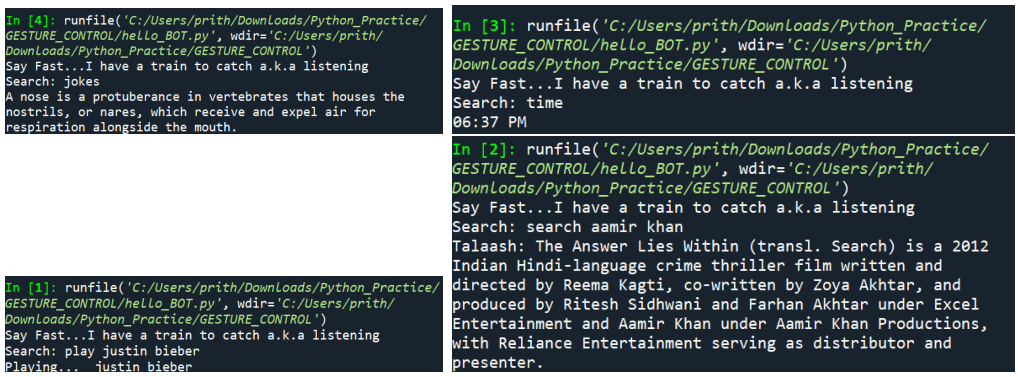

Figure 9: Search result

4 Discussion

Gesture-controlled functionality is a broad direction in the field of Computer Vision. Gesture control brings new light to the field of innovation and technology. With the current rise in fields like AR and VR and when their technology costs a fortune, it will always be economic to implement them in a diverse way.

There can be quite a few future directions of research based on the current work. We need to bring more optimization for low-end devices. Other possible future work could involve including more diverse gestures and improving the system's performance in challenging lighting and background conditions. Moreover, we aim to integrate all the functionalities into a simplified computer software, so that it becomes easily accessible to users.

References

- freecodingcamp.org/
- pypi.org/project/mediapipe/
- developers.google.com/mediapipe/
- Unity Documentation
- Keras Documentation
- github.com/siddharthlh24/OpenCV-tennis-game-control/
- pypi.org/project/wikipedia/
- pypi.org/project/pyjokes/