Hand Gesture Controlled Virtual Mouse based on ML and Computer Vision.

Ms. Deepti Sachin Deshmukh[1]
[1]Assistant Professor, Department of Computer Science and Engineering.

[1]Aryamaan Bhardwaj, [2]Harsh Mourya,[3]Megha Rawat,
[4]Prarthna Verma

[2]Mahatma Gandhi Mission's College of Engineering & Technology,
Noida, India
[1]Aryamaan17@gmail.com ,[2]harshmaurya6666@gmail.com,
[3]lpmegha12@gmail.com , [4]pv1418soni@gmail.com

Abstract

Hand gesture recognition is a rapidly developing field of artificial intelligence that is being used to create innovative new ways for people to interact with devices. This paper proposes a hand gesture-controlled virtual mouse system that uses AI algorithms to recognize hand gestures and translate them into mouse movements. This system is intended to offer an alternative interface for users who struggle with traditional mice. The proposed system uses a camera to capture images of the user's hand, which are then processed by an AI algorithm to recognize the gestures being made. The system is trained on a dataset of hand gestures to learn to recognize different gestures. Once a gesture is recognized, it is translated into a corresponding mouse movement, which is then executed on the virtual screen.

This system is designed to scale and adapt to various environments and devices. All input operations can be virtually controlled using dynamic or static hand gestures, or in combination with a voice assistant. The system uses machine learning and computer vision algorithms to recognize hand gestures and voice commands, and does not require any additional hardware.

The system is implemented using a convolutional neural network (CNN) and the MediaPipe framework. It has potential applications such as enabling hands-free operation of devices in hazardous environments and providing an alternative interface for people with disabilities. Overall, the hand gesture-controlled virtual mouse system offers a promising approach to enhancing user experience and improving accessibility through human-computer interaction.

Keywords: *Virtual Mouse, Human Computer Interaction, Gesture Recognition, Computer Vision , Mediapipe , OpenCV , Machine Learning*

1. Introduction

Our daily lives are increasingly shaped by technology. There are many computer technologies in the world, and they are growing rapidly. AI systems are used to automate a wide range of tasks that would otherwise be difficult or impossible for humans to perform. They are becoming increasingly powerful and capable, and have the potential to revolutionize many aspects of our lives. Humans interact with computers using input devices such as mice. The mouse is a device used for interacting with a GUI which includes pointing, scrolling and moving etc. Using a hardware mouse or touchpadto perform complex tasks on a computer or laptop can be time-consuming, in case we are carrying hardware mouse wherever we go it would be damaged sometimes. [2]

Gesture controlled virtual mouse makes human -computer interaction simple by making use of hands Gestures and voice commands. The computer can be used with almost no direct contact. All input output operations can be virtually controlled by using static and dynamic hand gesture along with a voice assistant .This project leverages state-of-the-art machine learning and computer vision algorithms to recognize hand gestures and voice commands without requiring any additional hardware. It leverages convolutional neural networks (CNNs) implemented by MediaPipe and exposed to Python via pybind11.It consists of module which works directly on hands by making use of Mediapipe Hand detection.[3]

Hand gesture controlled virtual mouse using artificial intelligence is a technology that enables users to control their computer mouse cursor with hand gestures, without requiring a physical mouse. This technology uses a camera-based computer vision approach to track the user's hand movements and perform mouse functions on the computer screen. The system captures video input from a camera pointed at the user's hand and uses computer vision algorithms to identify the user's hand and track its movement. This information is given to machine learning models which have been trained to recognize specific hand gestures, such as pointing or swiping, and translate them into corresponding mouse movements, such as moving the cursor or clicking. This latest super cool technology has various advantages, including its potential to improve accessibility for people and its ability to provide a more natural and intuitive user experience. It can also be useful in situations where a physical mouse or touchpads not available or practical. The use of hand gestures as a control mechanism eliminates the need for a physical mouse and provides a more intuitive and natural way of interaction with computers. This technology has numerous applications in areas such as gaming, virtual reality and accessibility quite easy for people.

2. Related Works

There are some related works carried out on virtual mouse using hand gesture detection by wearing a glove in the hand and also using color tips in the hands for gesturerecognition, but they are no more accurate for in mouse functions. The recognition is not so accurate because of wearing gloves; also the gloves are also suited for some users and in few cases, the recognition is not so accurate because of the failure of detection of color tips. Some efforts have been made for camera-based detection of the hand gesture interface.

[1] Sneha U. Dudhane, Monika B. Gandhi, and Ashwini M. Patil in 2013 conceptualized a study on ―Cursor Control System Using Hand Gesture Recognition. In this system, the drawback is that the framed have to be stored first and then processed for detection which is much slower than what is required in real- time.

[2] Dung-HuaLiou, ChenChiung Hsieh, and David Lee in 2010 presumed a studyon ―A Real-Time Hand Gesture Recognition System Using Motion History Image. The primary drawback of the proposed system is the implementation ofcomplicated hand gestures.

[3] In 1990, Quam achieved a hardware-based system; in this model, the user is supposed to wear a data glove, although Quam's model gives highly accurate results, moreover many gestures are difficult to perform with a glove that restricts most of the free movement, speed and agility of the hand.

[4] Saurabh Singh, Vinay Pasi, and Pooja Kumari in 2016 proposed ―Cursor Control using Hand Gestures‖. The model offers the use of various bands of colors toperform a variety of mouse operations. Its limitation is owed to the fact of the requirement of different colors to perform required functions.

[5] Monika B. Gandhi, Sneha U. Dudhane, and Ashwini M. Patil in 2013presumed a study on "Cursor Control System Using Hand Gesture Recognition." In this work, the limitation is that the stored frames are needed to be processed for hand segmentation and skin pixel detection.

[6] V. K. Pasi, Saurabh Singh, and Pooja Kumari in 2016 conceptualized "Cursor Control Using Hand Gesture " in the IJCA journal. The system suggests the different bands to perform different functions of the mouse. The limitation is that this approach requires processing stored frames for hand segmentation and skin pixel detection.

[7] Chaihanya C, Lisho Thomas, Naveen Wilson, and Abhilash SS in 2018 proposed "Virtual Mouse Using Hand Gesture" where the model detection is based on colors. but, only few mouse function are performed.

3. Methodology

The overview of the hand gesture recognition the hand is detected using the background subtraction method and the result of this is transformed to a binary image. The fingers and palm are segmented to facilitate the finger recognition. The fingers are detected and recognized. Hand gestures are recognized using a

3.1. The major Modules in the system design are

3.1.1. The Camera Used in the AI Virtual Mouse System: The proposed system uses web camera for capturing images or video based on the frames. For capturing we are using CV library OpenCV which is belongs to python web camera will start capturing the video and OpenCV will create a object of video capture. To AI based virtual system the frames are passed from thecaptured web camera.

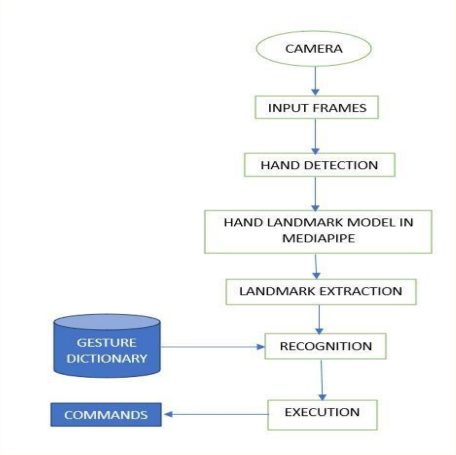

Figure 1 Flow Graph of Hand Gesture Recognition

3.1.2. Capturing the Video and Processing

The Artificial Intelligent virtual Mouse System uses the webcam where each frame is captured till the termination of the program. The video frames are converted from BGR to RGB color space and then processed to find the hands in each frame.

3.1.3. Rectangular Region for Moving through the Window

The system marks a rectangular region on the Windows display to capture hand gestures for mouse actions. When the hands are detected within the rectangular area, the system begins to detect the action and perform the corresponding mouse cursor functions. The rectangular region is drawn to capture hand gestures through the webcam for mouse cursor operations.

Mouse Functions Depending on the Hand Gestures and Hand Tip Detection using Computer Vision:

- For the Mouse Cursor moving around the Computer Window.

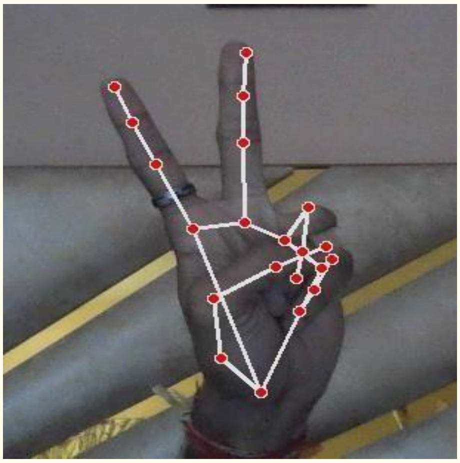

Figure 2: Computer Window with Mouse Cursor

- To Perform Left Button Click operation.

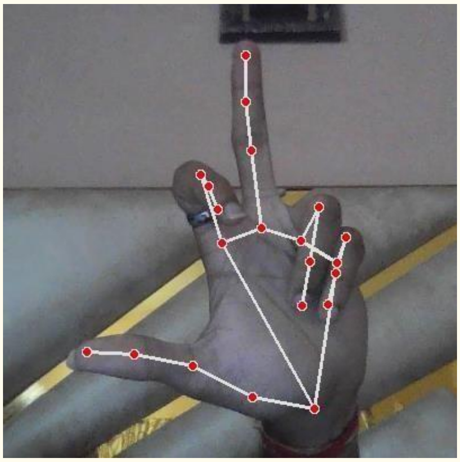

Figure 3: Mouse Operation- Left Click

- To Perform Right Button Click operation.

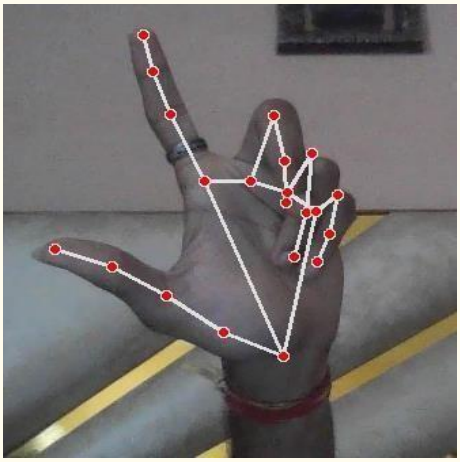

Figure 4: Mouse Operation- Right Click

- To perform a Double Click Operation.

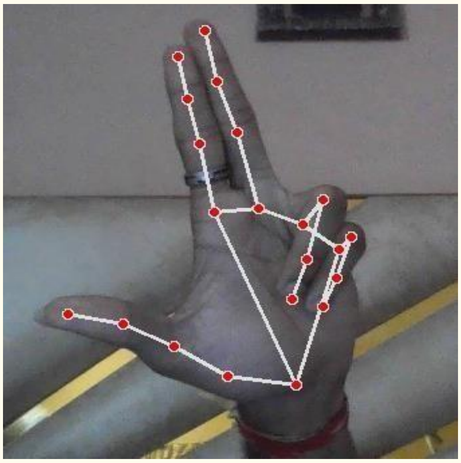

Figure 5: Mouse Operation-Double Click

To perform Scrolling Operation

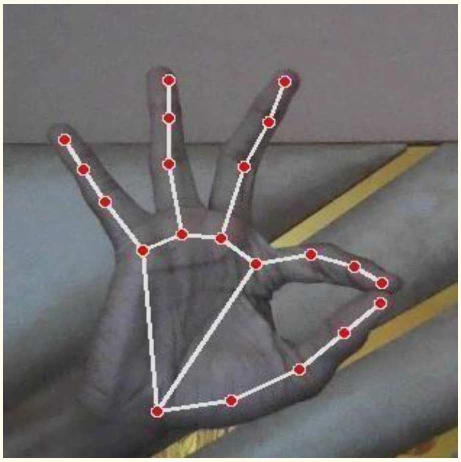

Figure 6: Mouse Operation-Scrolling

.

- To perform Drag & Drop Operation.

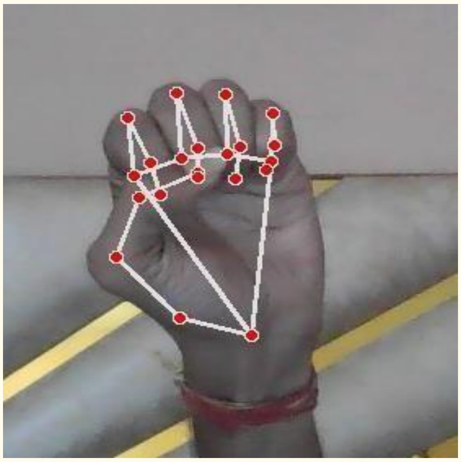

Figure 7: Mouse Operation-Drag & Drop

- To perform Multiple Item Selection.

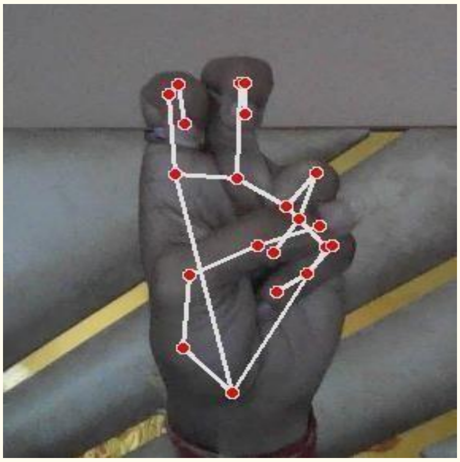

Figure 8: Mouse Operation- Multiple Item Selection

- To perform Volume Controlling.

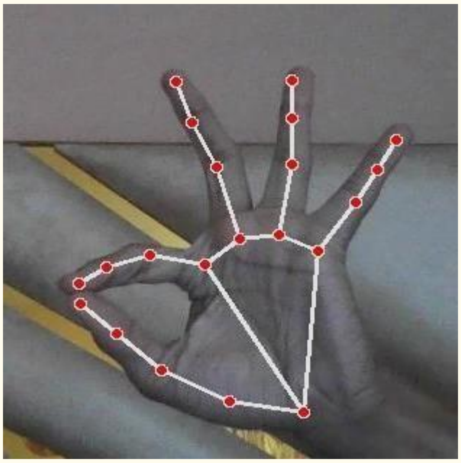

Figure 9: Mouse Operation- Volume Control

- To perform Brightness Controlling.

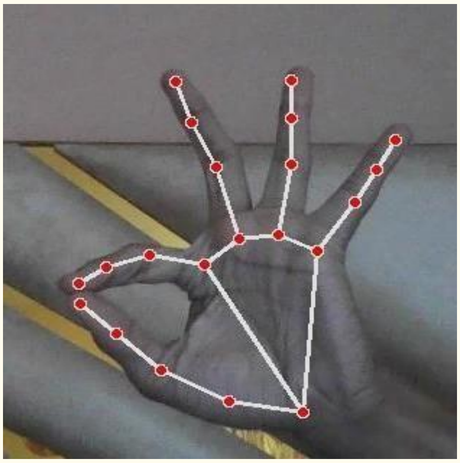

Figure 10: Mouse Operation- Brightness Control

- For No Action / Neutral Gesture to be performed on the Screen.

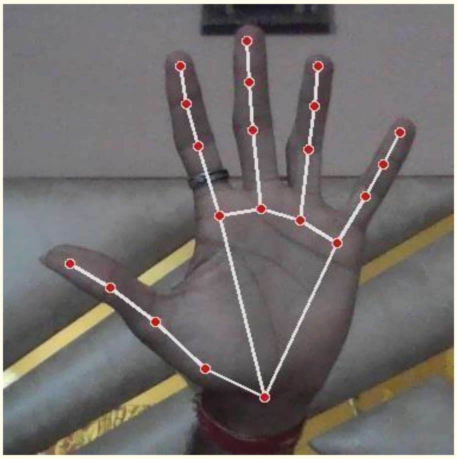

Figure 11: Neutral Gesture

3.1.4. Voice Assistant Features

The Voice Assistant feature has been included to launch gesture recognition through voice commands; and added certain features to improve the user engagement and they can assess whatever they need with less amount of effort and in hassle free manner. The Voice Assistant features which can be performed through the voice commands are:

- To Launch and End the Gesture Recognition.
- To Search for something over internet.
- To find a location what the user is looking for.
- To get an idea about Date and Time.

- To Copy and Paste contents.
- Sleep / Awake the Voice Assistant.
- To Exit the Voice Assistant.

4. Discussion and Analysis

4.1 Algorithms and Tools used :

For the purpose of hand and finger detection we are using the one of the effective open source library Mediapipe. It is a cross-platform framework developed by Google and OpenCV for performing machine learning tasks related to computer vision. This algorithm uses machine learning related concepts for detecting the hand gesture and to track their movements.

4.1.1 Mediapipe

Google created the open-source MediaPipe framework to enable the development of cross- platform, real-time computer vision applications. MediaPipe provides a variety of pre-built tools and components for processing and analyzing video and audio streams, such as object detection, pose estimation, and hand tracking.

MediaPipe allows developers to rapidly construct complex pipelines that combine multiple algorithms and processes, and execute in real time on a variety of h/w platforms, like CPUs, GPUs, and specialized accelerators like Google's Edge TPU.
MediaPipe also provides

interfaces for interacting with other popular machine learning libraries, such as TensorFlow and PyTorch, and supports multiple programming languages, including C++, Python, and Java.

MediaPipe is a comprehensive library for computer vision and machine learning tasks that offers a wide range of features. Here are a few of the library's main attributes and features:

- Video and Audio Processing: MediaPipe provides tools for real-time processing and analysis of video and audio streams. MediaPipe provides video and audio processing tools for real-time decoding, filtering, segmentation, and synchronization.
- Hand Tracking: MediaPipe can track hand movements in real-time, allowing for hand gesture recognition and interaction with virtual objects.

- Object Detection: MediaPipe provides real-time object detection and tracking using machine learning models. This functionality is useful for applications such as augmented reality, robotics, and surveillance.

- Pose Estimation: MediaPipe can estimate the poses of human bodies in real-time, enabling applications such as fitness tracking, sports analysis, and augmented reality.

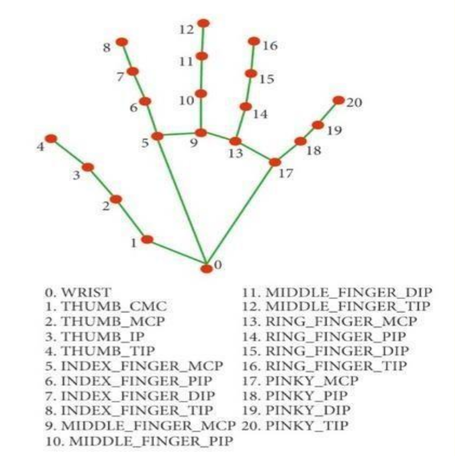

Figure 12: Hand Coordinates or Landmarks

For a variety of tasks, such as object detection, position estimation and more, Mediapipe offers tools for training and deploying machine learning models. All in all, Mediapipe is a potent tool kit that gives programmers the ability to easily create sophisticated real- time computer vision and ML applications.

4.1.2. OpenCV

A computer vision and ML software library called OpenCV is available for free download. Its objective is to aid programmers in the development of computer vision applications. Filtering, feature identification, object recognition, tracking, and other processing operations for images and videos are all available through OpenCV. Python, Java, and MATLAB are just a few of the numerous programming languages that it has bindings for. It is written in C++.Robotics, self-driving cars, AR, medical image analysis, and other fields are just a few of the fields where OpenCV can be employed. A wide range of algorithms and tools are included in the library, making it simple for programmers to build sophisticated computer vision applications.

The steps listed below can be used to broadly classify OpenCV's operation:

1. Loading and Preprocessing the Image/Video: OpenCV can load images or videos from a variety of sources such as files, cameras, or network streams. Oncethe image or video is loaded, it can be preprocessed by applying filters or transforming the image to a different color space, such as converting a color image to grayscale.

2. Feature Detection and Description: OpenCV can detect and extract features from an image or video, such as edges, corners, and blobs. These features can be used to identify objects or track their motion over time. OpenCV also provides algorithms for describing these features, which can be used to match them across multiple frames or images.

3. Object Detection and Recognition: OpenCV can be used to detect and recognize objects in an image or video. This can be done using a variety of techniques, such as template matching, Haar cascades, or deep learning-based methods.

4. Tracking: OpenCV can track objects in a video stream by estimating theirposition and motion over time. This can be done using a variety of algorithms, such as optical flow, mean-shift, or Kalman filtering.

5. Image and Video Output: Finally, OpenCV can be used to display or save the processed images or videos. This can be done by showing the images in a window, writing the video frames to a file, or streaming the video over a network.

In general, OpenCV offers a large variety of tools and techniques for working with image and video data, making it a potent library for computer vision applications.

4.1.3. Recurrent Neural Network (RNN) :

Recurrent Neural Networks (RNNs) are a type of artificial neural network that can process sequential data, such as text and speech, by maintaining an internal state that is updated at each time step. RNNs have a feedback loop that allows them to remember previous inputs, which makes them ideal for tasks such as language modeling, machine translation, and chatbot/ voice assistant development.

Chatbots and voice assistants are both computer programs that can simulate conversation with humans. RNNs can be used to create chatbots and voice assistants that are more natural and engaging than traditional chatbots and voice assistants, which often rely on rule-based systems or simple keyword matching.

RNNs can be used to train chatbots and voice assistants in a variety of ways. For example, RNNs can be used to train:

- A language model, which can then be used to generate text, translate languages, and answer questions in a comprehensive and informative way.
- A machine translation model, which can then be used to translate text in real time.
- A conversational AI Model, which can be used to create chatbots and voice assistants that can understand and respond to natural language commands andquestions.

Once a chatbot or voice assistant is trained using an RNN, it can be used to interact with user in a natural and engaging way. For example, a chatbot trained on customer service data can be used to answer customer question and resolve issues. A voice assistant trained on a variety of data can be used to control smart home devices, play music, and provide information to users.

Here is a simplified overview of how a chatbot or voice assistant using an RNN might

work:1.The user sends a message or speaks a command.

2. The chatbot or voice assistant’s RNN processes the message or command and predicts the next word, phrase, or action in the sequence.

3. The predicated word, phrase, or action is used to determine the appropriate response.

4. The chatbot or voice assistant generates a response or performs the requested action.

RNNs are a powerful tool for creating chatbots and voice assistants that are more natural and engaging than traditional chatbots and voice assistants. However, it is important to note that

RNNs can be complex and difficult to train. It is also important to have a large dataset of high-quality training data.

Here are some examples of chatbots and voice assistants that use RNN:

- Google Assistant
- Amazon Alexa
- Apple Siri
- Microsoft Cortana
- ChatGPT
- Bard

These chatbots and voice assistants are all able to understand and respond to natural language commands and questions, thanks to the power of RNNs.

4.1.4. Convolutional Neural Network (CNN) :

Convolutional Neural Networks (CNNs) are a type of artificial neural network that are well- suited for image recognition tasks. CNNs are able to learn spatial features in images, which makes them ideal for tasks such as classifying images, detecting objectsin images, and tracking objects in images.

CNNs can be used to create a hand gesture controlled virtual mouse by training a CNN on a dataset of images of hand gestures and the corresponding mouse cursor movements.The CNN will learn to identify the different hand gestures in the images and associate them with the corresponding mouse cursor movements.

Once the CNN is trained, it can be used to control a virtual mouse by capturing images of the user’s hand and using the CNN to predict the corresponding mouse cursor movement. This predicated mouse cursor movement can then be used to move the cursor on the screen.

Here is a simplified overview of how a hand gesture controlled virtual mouse using a CNN will work :

1. The webcam captures an image of the user’s hand.
2. The image is preprocessed to extract features such as edges and corners.
3. The preprocessed image is fed to the CNN.
4. The CNN predicts the corresponding mouse cursor movement.
5. The predicted mouse cursor movement is used to move the cursor on the screen.

CNNs are a powerful tool for creating hand gesture controlled virtual mouse that are accurate and robust. However, it is important to note that CNNs can be complex and difficult to train. It is also important to have a large dataset of high-quality training data.

Here are some advantages of using CNNs for hand gesture recognition:

- CNNs are able to learn spatial features in images, which is important for hand gesture recognition.
- CNNs are robust to noise and variations in the appearance of the hands.
- CNNs can be trained to recognize a wide variety of hand gestures.

Here are some challenges of using CNNs for hand gesture recognition:

- CNNs can be complex and difficult to train.
- CNNs require a large dataset of high- quality training data.
- CNNs can be computationally expensive to run.

Overall, CNNs are a promising approach for creating hand gesture controlled virtual mouse. CNNs are able to learn spatial features n images and are robust to noise and variations in the appearance of the hands. However, it is important to note that CNNs can be complex and difficult to train, and require a large dataset of high- quality training data.

4.1.5. Hidden Markov Model (HMM) :

Hidden Markov Models (HMMs) are a type of statistical model that can be used to model sequential data. HMMs are well-suited for hand gesture recognition because they can model the temporal dynamics of hand gestures.

To use an HMM in a hand gesture control virtual mouse system, the first step is to train the HMM on a set of labeled hand gesture data. The training data should include examples of all of the hand gestures that the system will be able to recognize.

Once the HMM has been trained, it can be used to classify new hand gestures by calculating the likelihood of each gesture given the observed hand landmarks. The gesture with the highest likelihood is then classified as the recognized gesture.

HMMs have been shown to be very effective for hand gesture recognition, and they have been used in a variety of hand gesture control applications, including virtual mouse control.

Here is an example of how an HMM could be used to implement a hand gesture control virtual mouse system:

- A computer vision algorithm is used to detect the user's hand and track its movement.
- The hand landmarks are extracted from the hand image.
- The HMM is used to classify the hand gesture.
- The corresponding mouse event is simulated using the pyautogui library.

For example, the following HMM could be used to implement a left-click gesture:

To use this HMM, the hand landmarks would be used to calculate the observation likelihoods for each state. The HMM would then be used to calculate the probability of each state given the observed hand landmarks. The state with the highest probability would then be classified as the recognized gesture.

If the recognized gesture is "click", then the pyautogui library would be used to simulate a left mouse click.

HMMs are a powerful tool for hand gesture recognition, and they can be used to implement hand gesture control virtual mouse systems with high accuracy.

5. Conclusion and Review Remarks

In conclusion, the paper proposes a hand gesture-controlled virtual mouse system that utilizes machine learning algorithms for recognizing and translating hand gestures into mouse movements. This innovative approach to human-computer interaction has the potential to enhance user experiences and improve accessibility, particularly for individuals who may have difficulty using traditional mice.

The system employs computer vision and machine learning algorithms, including Convolutional Neural Networks (CNNs), to recognize various hand gestures and perform corresponding mouse actions. Furthermore, it integrates a voice assistant feature to enable users to interact with the system using voice commands.

The methodology section of the paper outlines the use of tools and algorithms such as Media Pipe, OpenCV, RNNs, CNNs, and Hidden Markov Models (HMMs) to create an accurate and robust hand gesture recognition system. These technologies allow the system to process video input from a camera, identify and classify hand gestures, and perform mouse actions based on these gestures.

The discussion and analysis section highlights the advantages and challenges of using CNNs, RNNs, and HMMs for hand gesture recognition. It emphasizes the importance of having a large dataset of high-quality training data and the potential complexities involved in training these models.

Overall, the proposed hand gesture-controlled virtual mouse system is a promising development in the field of artificial intelligence and human-computer interaction. It offers potential applications in areas like accessibility, gaming, virtual reality, and hands- free operation in various environments. The combination of hand gesture recognition and voice commands provides a more intuitive and natural way of interacting with computers, reducing the reliance on physical input devices like mice and touchpads. However, the successful implementation of such a system would require addressing the complexities of training and fine-tuning the underlying AI models and ensuring a seamless user experience.

6. Future Scope

This paper proposes a hand gesture-controlled virtual mouse system that uses ML algorithms to recognize hand gestures and translate them into mouse movements. It offers a comprehensive overview of the methodology and tools used in developing this system. Here are some suggestions for the future scope of this project:

- **Enhanced Gesture Recognition:** Continuous research and development in machine learning can lead to more accurate and robust gesture recognition algorithms. Consider exploring advanced techniques, such as deep learning models (e.g., deep neural networks) or hybrid models that combine CNNs and RNNs to improve the recognition of complex gestures.

- **Accessibility:** One of the primary applications of ML- based virtual mouse software is to provide accessibility for individuals with physical disability or motor impairments. It allows them to interact with computers and devices without the need for traditional hardware mouse, making computing tasks more accessible and inclusive.

- **Healthcare:** In medical settings, AI based virtual mouse software can be usedto control computers and devices without physical contact, reducing the risk of cross-contamination and promoting a more hygienic environment.

- **Hardware Optimization:** explore the use of specialized hardware or sensors, such as dept cameras (e.g., Intel RealSense or Kinect) or wearable devices (e.g., Smart Gloves), to improve the accuracy and versatility of gesture recognition.

- **Security and Privacy:** Consider the security and Privacy implications of hand gesture control, implement safeguards to prevent unintended actions and ensure that the system respects user privacy.

7. Conflict of Interest

There is no conflict of interest in this work.

8. References

[1] Sneha U. Dudhane, Monika B. Gandhi, and Ashwini M. Patil in 2013 conceptualized a study on ―Cursor Control System Using Hand Gesture Recognition. In this system, the drawback is that the framed have to be stored first and then processed for detection which is much slower than what is required in real- time.

[2] Dung-HuaLiou, ChenChiung Hsieh, and David Lee in 2010 presumed a studyon ―A Real-Time Hand Gesture Recognition System Using Motion History Image. The primary drawback of the proposed system is the implementation ofcomplicated hand gestures.

[3] In 1990, Quam achieved a hardware-based system; in this model, the user is supposed to wear a data glove, although Quam's model gives highly accurate results, moreover many gestures are difficult to perform with a glove that restricts most of the free movement, speed and agility of the hand.

[4] Saurabh Singh, Vinay Pasi, and Pooja Kumari in 2016 proposed ―Cursor Control using Hand Gestures‖. The model offers the use of various bands of colors toperform a variety of mouse operations. Its limitation is owed to the fact of the requirement of different colors to perform required functions.

[5] Monika B. Gandhi, Sneha U. Dudhane, and Ashwini M. Patil in 2013presumed a study on "Cursor Control System Using Hand Gesture Recognition." In this work, the limitation is that the stored frames are needed to be processed for hand segmentation and skin pixel detection.

[6] V. K. Pasi, Saurabh Singh, and Pooja Kumari in 2016 conceptualized "Cursor Control Using Hand Gesture" in the IJCA journal. The system suggests the different bands to perform different functions of the mouse. The limitation is that this approach requires processing stored frames for hand segmentation and skin pixel detection.

[7] Chaihanya C, Lisho Thomas, Naveen Wilson, and Abhilash SS in 2018 proposed "Virtual Mouse Using Hand Gesture" where the model detection is based on colors. but only few mice function are performed.