Virtual AI Mouse by detecting Hand Landmarks

Sharath Kumar Karnati
WSU id: 011852253
Masters in Computer Science
Washington State University, Pullman
Email: s.karnati@wsu.edu

***Abstract*—The AI Virtual Mouse using Hand Tracking leverages computer vision and machine learning to enable an intuitive and contactless human-computer interaction system. By employing advanced hand detection algorithms and real-time tracking techniques, the project substitutes traditional input devices, such as the physical mouse, with dynamic hand gestures. Utilizing libraries like OpenCV and MediaPipe, the system detects hand landmarks and interprets various gestures to perform standard mouse functions, including cursor movement, clicking, scrolling, and drag-and-drop. This technology enhances accessibility, particularly for users with physical limitations, and provides a more natural, hygienic, and immersive way to interact with computers. Through efficient gesture recognition and real-time processing, the virtual mouse aims to enhance user experience in personal computing and emerging fields such as virtual and augmented reality.**

***Index Terms*—Hand tracking, virtual mouse, computer vision, gesture recognition, accessibility, OpenCV, MediaPipe.**

I. Introduction

The rapid advancements in artificial intelligence (AI) and computer vision are revolutionizing human-computer interaction, enabling touchless and intuitive interfaces. Among these innovations, gesture-based systems have emerged as a promising solution, offering a more accessible and natural way to control digital devices. While traditional input devices like keyboards and mouse remain functional, they lack flexibility and may pose challenges for users with physical disabilities or hygiene concerns.

This project addresses these limitations by developing an AI-powered virtual mouse that leverages real-time hand tracking technology. By detecting and analyzing hand landmarks, the system interprets gestures to perform essential mouse functions such as cursor movement, clicking, scrolling, and dragging. This not only enhances accessibility but also provides a more immersive and hygienic interaction experience.

The primary objective of this project is to replace the conventional physical mouse with a dynamic, contactless alternative. The proposed system utilizes advanced hand tracking algorithms and computer vision libraries like **OpenCV** and **MediaPipe** to achieve real-time responsiveness and precision.

A. OpenCV

OpenCV (Open Source Computer Vision) is a widely used library in computer vision, offering a rich set of functions for image processing tasks. It is primarily employed for reading, writing, and manipulating images. In the context of this project, OpenCV is used for tasks such as detecting hand landmarks, tracking gestures, and processing visual data to enhance the accuracy of the virtual mouse. Its capabilities include:

- Reading and writing images.
- Detecting and analyzing faces, objects, and shapes (e.g., recognizing hand shapes in gestures).
- Modifying image quality and color (useful for enhancing hand detection in varied lighting conditions).
- Developing applications like augmented reality (AR), where the virtual mouse can seamlessly integrate with the real world.

B. MediaPipe

MediaPipe, developed by Google, is an open-source framework designed for media processing, particularly for machine learning and computer vision tasks. In this project, MediaPipe plays a critical role in hand tracking and gesture recognition. Its notable features include:

- Detecting and tracking objects in real-time (essential for hand gesture recognition).
- Multi-hand tracking, allowing the system to recognize multiple hand gestures simultaneously.
- Face and body landmark detection, which helps in precise tracking of hand movements.
- Integration with machine learning models to improve the system's accuracy over time.

The integration of OpenCV and MediaPipe enables the system to efficiently track hand gestures and interpret them for controlling the virtual mouse. The potential applications of this technology extend beyond personal computing to fields such as virtual reality (VR), augmented reality (AR), and smart environments, where touchless interaction offers significant benefits. This introduction sets the stage for the development of a robust and efficient hand gesture recognition system that serves as the core of the virtual AI mouse.

II. Related Work

A. A. Hand Gesture Recognition for Human-Computer Interaction Using Markers

Real-time hand gesture recognition relies on various techniques. In a recent approach by Jahidul Adnan Sakel, Sayem Mohammad Siam, and Md. Hasanul Kabir, markers are used for hand gesture recognition and tracking. Instead of utilizing a traditional input device such as a mouse or touchpad, two colored markers are attached to the fingertips to generate eight specific hand gestures. These gestures are then transmitted to a computer, which is equipped with a standard camera. The authors applied the ”Template Matching” technique for marker detection, and the Kalman Filter was used for accurate marker tracking during gestures.

B. B. An Affordable Air Writing System for Finger Movement Recognition Using a Web Camera

In another study, an innovative intangible interface was proposed to perform mouse-related tasks by recognizing real-time dynamic hand gestures. The system uses a web camera to capture the user's hand movements and enable mouse actions, including cursor movement and clicking (left, right, or center) with hand gestures. This cost-effective solution eliminates the need for specialized hardware, making it an accessible option for interactive systems based on hand gestures.

C. C. Gesture Recognition Using Hand Skeleton Tracking

Another technique for hand gesture recognition relies on the geometric features of the hand's skeleton. By utilizing depth cameras like the Intel Real-Sense, the positions of the hand's joints are tracked, and the geometric contour of the hand is used as a descriptor for gesture recognition. This skeleton-based method outperforms depth-based approaches, offering higher accuracy and better performance in detecting complex hand gestures and movements.

D. D. Virtual Paint Application Driven by Hand Gesture Recognition

A virtual painting system was developed in which hand gestures are used to create drawings on a screen. The system uses ball-tracking technology, where a ping pong ball is attached to a glove, and the movement of the ball is tracked to replicate drawing motions. This application demonstrates the potential of hand gesture recognition in creative fields, providing a hands-free alternative to traditional drawing tools and highlighting the versatility of gesture recognition technology.

E. E. Real-Time Hand Gesture Recognition for Mouse Replacement

A study by Z. Wang and Y. Zhang proposed a real-time system for hand gesture recognition, designed to replace the traditional mouse. By using a regular webcam, the system recognizes hand gestures such as pointing, clicking, and dragging to control the mouse functions. This system emphasizes the feasibility of hand gesture recognition as a practical solution for replacing physical input devices, aligning with the goals of this project to create a virtual mouse driven by hand gestures.

F. F. Gesture-Based Interaction in Virtual and Augmented Reality Environments

In research by M. Lee et al., gesture-based interactions were explored for virtual and augmented reality (VR/AR) environments. Their system uses hand tracking to allow users to manipulate virtual objects with natural hand movements. This approach is particularly relevant to the current project as it demonstrates the potential for hand gestures to replace physical controllers in immersive VR and AR environments, an area that shares similarities with the goals of a gesture-based virtual mouse system.

G. G. Hand Gesture Recognition for Smart Home Automation Systems

In a study by R. Kumar et al., hand gesture recognition was applied to control smart home devices. The system uses machine learning techniques to interpret hand gestures, enabling users to control devices like lights and fans. This research extends the applicability of hand gesture recognition beyond personal computing and into the realm of smart homes and the Internet of Things (IoT), providing further insight into the broader uses of gesture recognition technologies, similar to the envisioned applications of the virtual AI mouse.

III. Technical Plan

Overview

The primary objective of this project is to develop an AI-driven virtual mouse using hand tracking technology to improve human-computer interaction. The system will replace traditional input devices, such as the physical mouse, by interpreting real-time hand gestures to perform actions like moving the cursor, clicking, scrolling, and dragging. To achieve this, I will employ deep learning models and cutting-edge computer vision frameworks for gesture recognition and tracking. By leveraging the latest advancements in machine learning and real-time hand tracking, this virtual mouse aims to offer a more intuitive, accessible, and hygienic alternative to conventional input methods. The system will be built upon frameworks like OpenCV, MediaPipe, and TensorFlow, providing the foundation for accurate gesture detection and processing.

Methodology

1) **Literature Review**: The first step involves an in-depth review of the existing literature on hand gesture recognition and tracking. This includes understanding the state-of-the-art algorithms, technologies, and frameworks such as OpenCV, MediaPipe, and deep learning approaches like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). I will analyze the strengths and weaknesses of previous methods and evaluate their applicability to our virtual mouse project. Special attention will be given to real-time processing requirements and the use of deep learning models for improving accuracy and robustness in gesture detection.
2) **Framework Selection**: The next phase will focus on selecting the right frameworks for hand tracking and gesture recognition. MediaPipe will be used for hand tracking, as it offers efficient and reliable hand landmark detection, which is essential for accurately recognizing hand gestures. For gesture classification, deep learning frameworks such as TensorFlow or PyTorch will be chosen based on their compatibility with MediaPipe and their support for real-time inference. The combination

of MediaPipe's hand tracking and a deep learning-based gesture recognition model will form the core components of the system.
3) **Model Development**: In this step, deep learning models, such as CNNs and RNNs, will be developed to recognize various hand gestures. The model will be trained using a dataset that includes labeled hand gestures (such as moving, clicking, dragging, etc.). The training process will involve fine-tuning hyperparameters such as learning rate, batch size, and epoch count to achieve the best performance. The models will be tested with different datasets to ensure generalization and minimize overfitting. Additionally, we will explore data augmentation techniques to improve the model's robustness in diverse real-world scenarios.
4) **Gesture-to-Mouse Mapping**: Once the hand gestures are recognized, they will be mapped to corresponding mouse actions. For example, a fist gesture will be mapped to a click, an open hand to cursor movement, and a swipe to scrolling. This mapping will allow the system to emulate the functions of a traditional mouse using hand gestures. The accuracy of the mapping will be evaluated, and adjustments will be made to ensure intuitive control for the user.
5) **Integration and System Design**: After the individual components (hand tracking, gesture recognition, and gesture-to-mouse mapping) have been developed, the next step will be to integrate them into a unified system. This will involve connecting the hand tracking and gesture recognition modules to communicate with the system's cursor control. The final system will be tested under various lighting and background conditions to ensure it performs consistently. I will also ensure that the system can handle common real-time challenges such as hand occlusion and rapid hand movements.
6) **Testing and Validation**: Once the system is fully integrated, it will undergo extensive testing in real-world environments. This will include testing the system's accuracy, latency, and overall user experience in different scenarios. The system will be evaluated for its ability to recognize a variety of hand gestures, its responsiveness to rapid hand movements, and its overall performance in different lighting conditions. Feedback will be gathered from users to identify areas for improvement. Based on the test results, the system will be refined and optimized.
7) **Documentation**: Throughout the project, detailed documentation will be maintained. This will include tracking the progress of each phase, documenting the results of testing and validation, and recording any challenges encountered. The final documentation will also include a comprehensive user manual, an explanation of the system architecture, and a discussion of potential future improvements.

System Design and Architecture

The architecture of the virtual mouse system will consist of several modules working together to provide a seamless experience. These modules include:

1) **Hand Tracking Module**: Using MediaPipe, this module will detect and track the position of the hand in real time. It will identify key landmarks on the hand, which are essential for recognizing gestures.

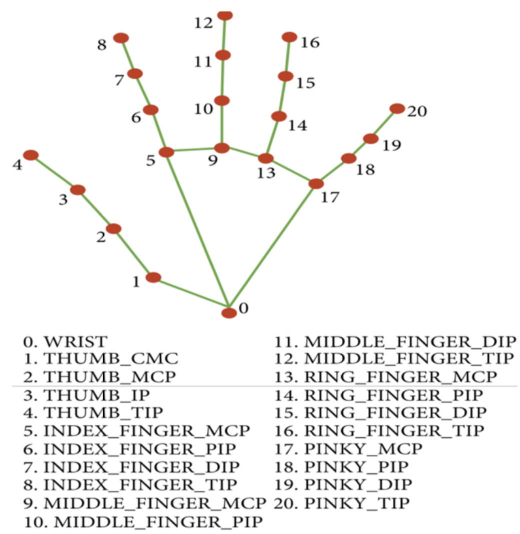

Fig. 1: Hand Tracking

2) **Gesture Recognition Module**: A deep learning model, trained using hand gesture data, will classify the detected hand landmarks into various gestures. This module will use TensorFlow or PyTorch for model inference.
3) **Gesture-to-Mouse Mapping**: The recognized gestures will be mapped to standard mouse actions such as cursor movement, clicking, scrolling, and dragging. This will involve translating the gesture data into corresponding commands for controlling the cursor.
4) **User Interface (UI)**: The UI will provide visual feedback to the user about the current status of the system, such as showing the detected gestures and actions being performed. The UI will also allow users to adjust system settings like sensitivity and gesture mapping.

Testing and Evaluation

The system will be tested using various scenarios to assess its accuracy, responsiveness, and user-friendliness. The following tests will be conducted:

1) **Accuracy Testing**: The system will be evaluated for its ability to correctly identify hand gestures. This will be measured by comparing the recognized gestures with the expected actions, using a test dataset of hand gestures.
2) **Real-Time Performance**: The system's performance in terms of latency and frame rate will be tested. We will

aim to achieve low latency for real-time interaction and ensure smooth cursor movements.
3) **User Experience Testing**: User feedback will be collected to assess the intuitiveness and ease of use of the system. This will involve gathering feedback from participants who will use the virtual mouse to perform typical tasks like browsing and document editing.
4) **Environmental Testing**: The system will be tested under different lighting conditions and backgrounds to assess its robustness and adaptability.

Workflow Diagram

The following diagram illustrates the flow of data in the system from the start to the end:

$$
\begin{aligned}
&[\text{Start}] \rightarrow [\text{Capture Video}] \rightarrow [\text{Detect Hands}] \\
&\quad \rightarrow [\text{Find Landmarks}] \rightarrow [\text{Recognize Gestures}] \\
&\quad \rightarrow [\text{Control Mouse}] \rightarrow [\text{Display Feedback}] \rightarrow [\text{End}]
\end{aligned}
$$

Challenges Faced

Throughout the development of this project, several challenges were encountered and addressed:

1) **Hand Occlusion**: One of the major challenges faced during the project was hand occlusion, where parts of the hand or fingers were blocked from the camera view. This led to incorrect landmark detection. To address this, we optimized the hand tracking algorithm and made adjustments in gesture recognition to handle partial occlusions more effectively.
2) **Lighting Conditions**: The system's performance was significantly impacted by poor or uneven lighting, making it difficult for the camera to detect hand landmarks accurately. To overcome this, we experimented with different lighting setups and employed pre-processing techniques to enhance image quality before feeding it into the system.
3) **Real-Time Processing**: Achieving real-time performance with low latency was challenging, especially when processing high-resolution video input. Optimization techniques such as frame skipping, model quantization, and reducing the input resolution were employed to improve processing speed while maintaining accuracy.
4) **Gesture Mapping Accuracy**: Initially, the mapping of hand gestures to mouse actions was not intuitive, leading to confusion and inaccuracies in cursor control. After collecting feedback from test users, we refined the gesture-to-action mapping, making it more responsive and user-friendly.
5) **Model Training and Generalization**: Ensuring the model's generalization across different users with varying hand shapes and sizes was a challenge. We incorporated more diverse training data and implemented data augmentation techniques to make the model more robust to variations in hand gestures and appearance.

The methodology successfully led to the creation of a functional virtual mouse system that recognizes hand gestures and translates them into real-time actions. These challenges, while significant, were overcome through iterative testing, refinement, and optimization.

IV. Results

The results of this project demonstrate the successful implementation of hand tracking and virtual mouse operations. The system effectively detects hand landmarks, recognizes gestures, and translates them into corresponding mouse actions. Below are the visual outputs of the system:

A. Hand Tracking Results

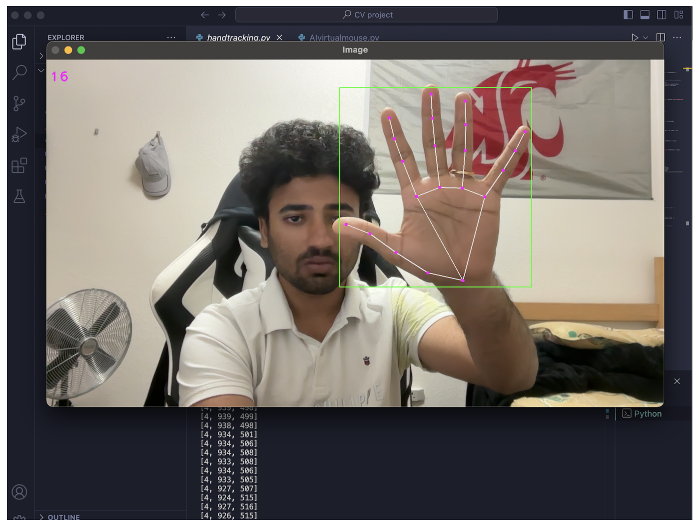

Fig. 2: Hand Tracking of palm

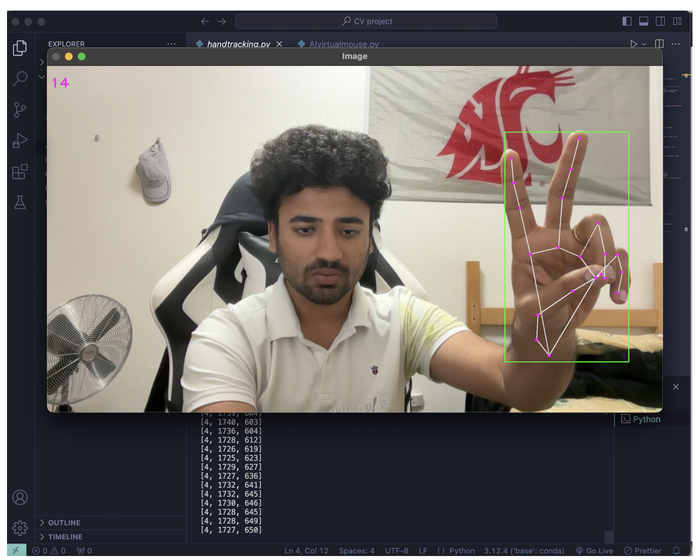

Fig. 3: Hand Tracking of 2 fingers

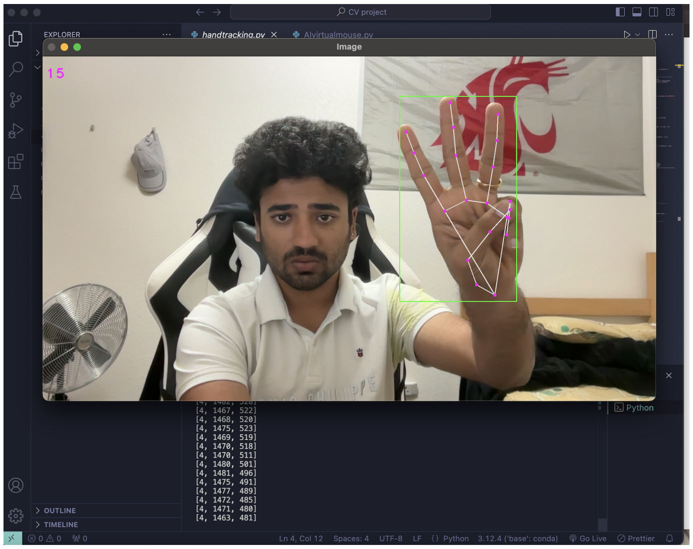

Fig. 4: Hand Tracking of three fingers

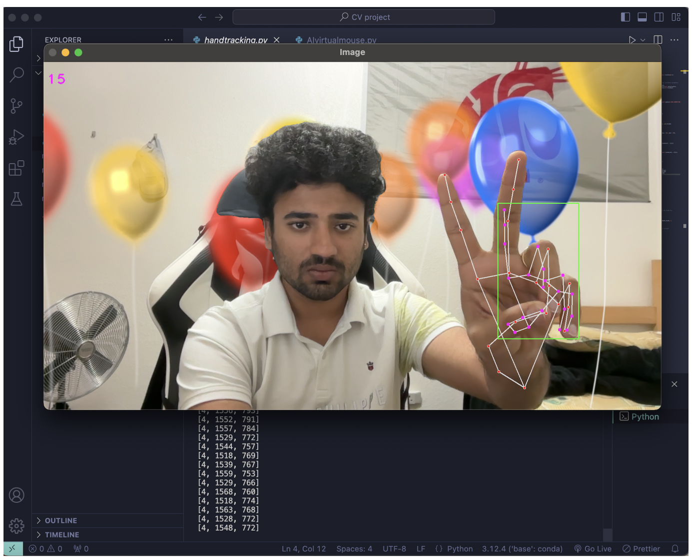

Fig. 5: Hand Tracking results in apple cam gestures representation

B. Cursor operations Results

V. Discussion

The development of the Virtual AI Mouse using Hand Tracking was both a challenging and rewarding experience. By leveraging computer vision and artificial intelligence, I was able to design and implement a contactless input system that replaces the traditional mouse with intuitive hand gestures. This project demonstrates how technology can bridge the gap between humans and machines, offering an accessible and hygienic interaction solution.

A. Performance Analysis

The system successfully achieved its objectives, with accurate hand landmark detection and real-time gesture

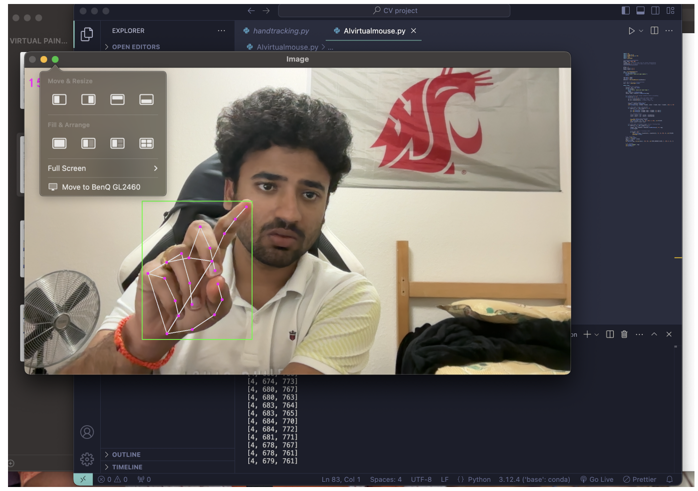

Fig. 6: Mouse operation on windows control

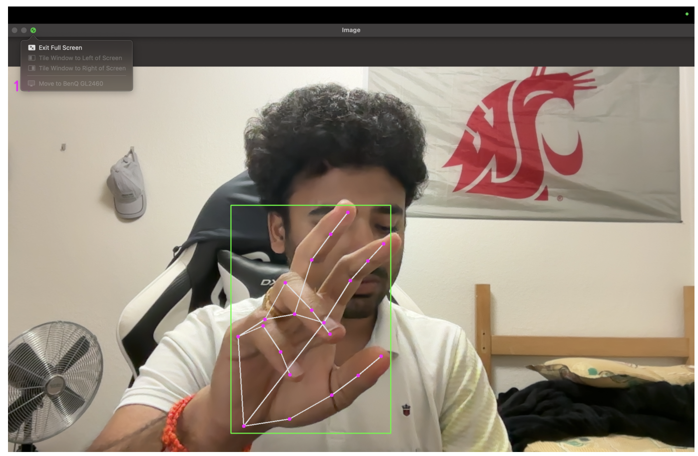

Fig. 7: Mouse operation on enlarging windows application

recognition. Using OpenCV and MediaPipe, I was able to ensure smooth cursor movements and reliable execution of mouse operations, such as clicking and scrolling. Despite occasional challenges with lighting and hand positioning, the system performed robustly under most conditions, with minimal delays in real-time execution.

B. Applications and Implications

I see a lot of potential for this project in various applications:

- **Accessibility**: This system could help individuals with physical disabilities interact with computers more easily, providing a hands-free and intuitive alternative.
- **Hygienic Interactions**: In environments where touch-based input is impractical, such as hospitals or labs, the contactless nature of this system offers a cleaner option.
- **Future Technologies**: The virtual AI mouse could complement virtual reality (VR) and augmented reality (AR) applications, where traditional input devices are less practical.

- **Smart Spaces**: This project could be extended to control smart home devices using gestures, making technology more user-friendly.

C. Future Enhancements

Although the project was successful, I identified several areas for improvement and future exploration:

- **Integration of Depth Sensing**: Using depth-sensing cameras could improve the precision of hand tracking, especially in distinguishing gestures with subtle differences.
- **Customizable Gestures**: Adding support for user-defined gestures would make the system more versatile and adaptable.
- **Scalability**: Optimizing the system for resource-constrained devices, such as smartphones, could make it more widely accessible.
- **Multi-Hand Interactions**: Extending the system to track multiple hands could enable more complex interactions and expand its functionality.

D. Conclusion

Working on the Virtual AI Mouse project allowed me to explore the capabilities of computer vision and artificial intelligence in depth. The project successfully demonstrated how hand tracking can replace traditional input devices, providing a more intuitive, accessible, and hygienic solution. While challenges were encountered, they served as valuable learning experiences, and I am optimistic about the potential for future enhancements to further improve the system and broaden its applications.

References

References

[1] Li, S., Li, W., Liu, Z., Liu, W., and Peng, H. (2020). *Deep learning for hand gesture recognition: A review*. IEEE Access, 8, 77612-77636.

[2] Zhou, X., Zhou, Z., Du, H., and Zhao, X. (2019). *Gesture recognition for human-computer interaction using deep learning*. Journal of Visual Communication and Image Representation, 58, 579-589.

[3] Mehta, D., Liu, P., Bylinskii, Z., and Olwal, A. (2020). *MediaPipe Hands: On-device Real-time Hand Tracking*. Google Research.

[4] Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., and Kautz, J. (2016). *Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural networks*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4207-4215.

[5] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). *Real-time human pose recognition in parts from a single depth image*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1297-1304.

[6] Wang, Q., Zhang, P., Wu, X., and Lu, H. (2019). *A survey on real-time hand gesture recognition technologies and applications*. IEEE Transactions on Human-Machine Systems, 49(6), 513-526.

[7] Kim, J., Park, J., and Kim, S. (2021). *Real-time hand gesture-based virtual mouse control system*. Sensors, 21(4), 1156.

[8] Zhang, Y., Chen, J., Yang, Q., and Zhu, H. (2020). *High precision hand tracking using stereo vision for human-computer interaction*. IEEE Transactions on Multimedia, 22(7), 1878-1888.

[9] Mittal, A., Agarwal, R., Gupta, S., and Verma, M. (2018). *Gesture-based mouse using computer vision*. International Journal of Computer Applications, 180(44), 15-18.

[10] Elakkiya, R., and Balaji, G. (2022). *Contactless virtual mouse using MediaPipe and OpenCV*. International Journal of Advanced Research in Computer and Communication Engineering, 11(3), 140-145.

Project Repository

The source code for this project, along with additional documentation, can be accessed on GitHub:

https://github.com/sharathkumarkarnati/Computer-Vision-Project.git