www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 1418-1424

e-ISSN : 2583-1062
Impact Factor : 7.001

GESTURE AND VOICE CONTROLLED VIRTUAL MOUSE

V. Sriram[1], Y. Sriya[2], M. Srujan Kumar[3], M. Sruthi[4], G. Suchithra[5], Prof. Md. Shafi[6]

[1,2,3,4,5]B. Tech School of Engineering Computer Science-(AI&ML) Malla Reddy University, India.

[6]Assistant Professor School of Engineering Computer Science-(AI&ML) Malla Reddy University, India.

ABSTRACT

Hand gesture recognition is a rapidly growing area of artificial intelligence that is being used to develop new and innovative ways for people to interact with technology. This paper suggests a system that allows users to control a virtual mouse using hand gestures, which are recognized and translated into mouse movements by artificial intelligence algorithms. This system is designed to provide an alternative interface for individuals who find traditional mice challenging to use. The suggested system employs a camera to capture images of the user's hand, which are subsequently analyzed by an artificial intelligence algorithm to identify the gestures being performed. The system is trained on a collection of hand movements to understand and identify various gestures. Once a gesture is identified, it is converted into a mouse movement, which is then carried out on the virtual display. The system is designed using a convolutional neural network (cnn) and the media pipe framework. It has the potential to be used in various scenarios, such as allowing devices to be operated without hands in dangerous situations and offering an alternative way for individuals with disabilities to interact with technology. In summary, the hand gesture-controlled virtual mouse system presents a promising method for enhancing user experience and promoting accessibility through the interaction between humans and computers. Finger tracking systems primarily concentrate on user interaction, with the goal of making gestures and hand movements more intuitive and creative. The primary purpose of finger tracking systems is to simplify computer usage and enhance user experience by enabling natural language and gesture interactions. The utilization of finger motion tracking and recognition systems facilitates typing in ways that enhance communication with computers through natural movements.

1. INTRODUCTION

This human civilization has been utilizing hand signals for communication since ancient times. Throughout history, various hand gestures like shaking hands, giving a thumbs up, and giving a thumbs down have been present in our surroundings. It is commonly thought that gestures are the simplest means of communication with others. Therefore, it would be beneficial to make use of the technology that we already have. This paper demonstrates real-world applications. The initial setup involves a budget-friendly usb web camera that can be utilized for providing input to the system. The entire process is divided into four steps, which include capturing videos, processing frames, extracting regions, and matching features. To an extreme extent, it can also be referred to as hardware since it employs a webcam for tracking hand movements. The interaction between humans and computers became a significant concern as technology continued to evolve rapidly. Conversely, individuals who are not experienced or older individuals often struggle to identify and press the precise letter they require. Achieving this challenge is the ultimate objective of our proposed system. Typing has evolved over time, starting with the use of keyboards, transitioning to touch screens, and now benefiting from advanced finger motion tracking systems. The potential of this innovative technology can provide a solution to the problem at hand. Using hand gestures to represent letters might not be as challenging for the elderly or amateur individuals, as long as they are familiar with the language. In a situation where the person has to type a short document or an email and typing using a keyboard or touch is the only option, they would have to stand behind others to assist them. To advance the field of human-computer interaction, the latest form of typing has evolved to include gesture recognition systems, which are more intuitive and user-friendly. Gesture recognition is a field that focuses on understanding human gestures using mathematical algorithms. Gesture recognition enables users to interact with computers effortlessly, eliminating the need for any mechanical devices. The effectiveness of these gesture recognition systems surpassed the use of touch pads, leading to more efficient communication between devices.

To recognize cars and other objects in the video frames, the project utilizes a convolutional neural network (cnn)-based model. To assist the system in focusing on regions where accidents occur, faster r-cnn is employed for accident localization. The system can be implemented in real-world settings because it is engineered to perform optimally in diverse lighting conditions, weather patterns, and camera perspectives.

The capability of gesture recognition systems to replace conventional input methods like touchpads underscores their potential to transform human-device interaction by providing faster, more intuitive, and touch-free control. Alongside gesture control, the project also incorporates advanced object detection techniques The model is engineered to function optimally in diverse lighting conditions, weather situations, and camera viewpoints, guaranteeing its effectiveness and dependability in practical scenarios. In summary, the combination of gesture recognition and computer vision-based accident detection creates a comprehensive system that broadens the possibilities of human-computer interaction. It not only provides an easy-to-use interface for people from all walks of life but also showcases potential applications in safety monitoring and automation.

2. LITERATURE REVIEW

Road accidents remain a significant contributor to property damage, loss of life, and severe injuries on a global scale. Traditional methods of accident detection that require physical intervention and rely on eyewitness accounts often lead to delays in emergency responses, which can have fatal outcomes. By analyzing live CCTV footage in real-time, advanced accident detection systems that leverage artificial intelligence and computer vision have emerged to enhance response times. Early attempts to tackle environmental challenges such as weather and camera angles relied on motion detection techniques like background subtraction and optical flow. Modern systems that employ deep learning techniques, such as convolutional neural networks (cnns) and recurrent neural networks (r-cnn), significantly enhance accuracy. Although reducing false positives and achieving real-time performance are challenging, they are not impossible. In recent years, writing in the air has emerged as a captivating and demanding area of research in the field of image processing and pattern recognition. It can enhance the interaction between humans and machines in various applications. Object tracking is regarded as a crucial task within the field of computer vision. The system explains how computer vision can be used to track the movement of the finger. The generated text can be utilized for a wide range of applications, including sending messages, emails, etc. The project aims to create a motion-to-text converter that can be integrated into intelligent wearable devices, enabling users to write from the air. The project capitalizes on this opportunity and concentrates on creating a motion-to-text converter that can be utilized as software for intelligent wearable devices, enabling users to write from the air.

3. METHODOLOGY

3.1Existing System

The system employs an external device known as the leap motion controller to capture users' hand and finger movements, facilitating intuitive interaction with digital content. This approach achieves an accuracy of 90.125%, and to create a virtual cursor and keyboard, the complex convex hull algorithm is utilized. Nevertheless, the system encounters challenges like the inability to accurately identify multiple hand gestures simultaneously, and the intricate color segmentation process employed in the creation of hand gesture-based mouse technology is also intricate. Furthermore, the speech-to-text feature has a restricted language translation capability, which presents another limitation. Demonstrating multi-touch and mid-air gestures is more difficult than single-touch gestures because the latter only requires accurate path-following without worrying about hand posture. In contrast, multi-touch and mid-air gestures depend on the location and motion of multiple fingers or the entire hand. Most teaching systems prioritize guiding users through the required hand movements and path for gestures, rather than emphasizing hand posture and contact form. They also prioritize commands that can be executed with single-touch input devices, such as a mouse.

3.2 Proposed System

A new system has been created that can recognize hand movements in real-time, without requiring any additional devices, solely relying on a camera for input. The system's precision is contingent upon the quality of the camera employed. This solution eliminates the need for additional hardware devices, providing a contact-free environment that enhances convenience and reduces expenses. Alongside gesture recognition, the system integrates a speech-to-text feature, enhancing its usability and enabling more diverse forms of user interaction. Utilizing Python's built-in libraries streamlines the system's development, making it cost-effective and efficient while minimizing complexity. Because the system doesn't require specialized hardware like a leap motion controller, it helps reduce costs. One notable advantage is that the system requires minimal training time, allowing users to operate it effortlessly. Its dependence on easily accessible resources, coupled with its user-friendly design, makes this system a practical and accessible choice for a wide range of applications.

3.3 Modules

1. Palm Detection Module:

The palm detection module is in charge of identifying the initial position of the hand in an image, just like how face detection works in media pipe face mesh. This module is specifically designed to accommodate a wide range of hand sizes and diverse high-contrast palm patterns, ensuring its versatility and adaptability in various conditions. The module employs square bounding boxes as anchors in machine learning to identify the palm. Due to their smaller size, the module utilizes a non-maximum suppression algorithm, which eliminates redundant bounding boxes and enhances the accuracy of detection. By implementing these strategies, the palm detection module can attain an average precision of 95.7%, guaranteeing accurate identification of palm regions even in challenging or cluttered surroundings.

2 Hand landmark Module:

Once the palm detection module analyzes the image, the hand landmark module steps in, concentrating on accurately pinpointing specific areas on the hand. This module identifies 21 critical 3d coordinates that indicate the position of the hand's knuckles, fingers, and wrist. The model employs regression techniques to acquire consistent hand pose representations, enabling it to accurately track hand movements and gestures. The hand landmark module is highly resilient to partial occlusions, allowing it to maintain its accuracy even when the hand is partially obscured in the image. This module is crucial for comprehending the precise locations of each hand joint, which is vital for applications like gesture recognition or hand tracking.

3. Cursor Control Module:

The cursor module is responsible for converting the detected hand movements into cursor movements on the screen. Once the palm detection module identifies the hand region, and the hand landmark module provides the precise coordinates of key hand landmarks, the cursor module utilizes this information to track the hand's movement. These coordinates are initially converted from the camera's resolution to the actual screen resolution. The system employs these translated coordinates to mimic cursor movement, enabling the user to interact with the screen in a manner similar to using a conventional mouse. The module flawlessly tracks the cursor's movement, mirroring the position and motions of the hand or finger, providing a seamless and intuitive method of cursor control through hand gestures.

4. KEYBOARD MODULE

The keyboard module allows the system to mimic keyboard input by analyzing the hand movements detected by the palm detection and hand landmark modules. It collaborates with the cursor module, concentrating on the precise positioning of interactive buttons on the screen. The system determines the position of each virtual button on the screen and uses the cursor's location to identify when the hand or finger arrives at the button's position. By aligning the cursor's position with the button's position on the screen, the system mimics a button press. To achieve this functionality, the module employs opencv's button module, which enables the creation of buttons with different attributes, such as their position. When the cursor moves over a button, it can activate the associated action or mimic keyboard input, enabling users to type or interact with the system through gestures.

5. Speech Recognition:

The speech recognition module employs the speech recognition library to transform audio input received from the system's microphone into written text. This module is created to identify voice commands or spoken input, allowing for hands-free interaction with the system. It captures sound through the microphone and analyzes it to recognize speech patterns. The acknowledged speech can be utilized to regulate different components of the system or input text, making it a potent tool for accessibility and voice-controlled applications. This module enhances user interaction by enabling them to control the system through natural language commands, in addition to visual or gesture-based inputs.

5. ARCHITECTURE

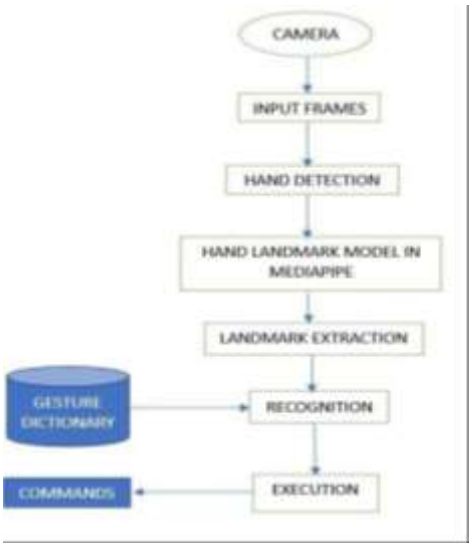
The architecture of the gesture-based voice-controlled virtual mouse system combines computer vision, machine learning, and voice recognition technologies. The system is structured in a modular manner, consisting of three main components: image capture, gesture recognition, and command execution. The image acquisition module captures live video footage using a webcam. The gesture recognition module utilizes computer vision techniques to identify and categorize hand movements. Simultaneously, the voice recognition module processes audio commands. These inputs are then sent to a control logic unit that analyzes the data and decides on the appropriate mouse action, such as moving the cursor, clicking, or scrolling. The design prioritizes smooth data movement, utilizing streamlined pipelines to ensure optimal real-time performance. By combining both hand and voice commands, the system provides improved accuracy and user-friendliness.

5.1 Methods & Algorithms:

The system integrates various techniques for reliable performance. The system utilizes the mediapipe library to detect hand gestures, which efficiently tracks 21 key points on the hand. This model guarantees quick and precise identification of landmarks, even in intricate backgrounds. To classify gestures, a custom convolutional neural network (cnn) is utilized to identify predefined gestures. The CNN is trained on a diverse dataset of hand gestures, guaranteeing its ability to perform well under different lighting conditions and hand orientations. To enable voice commands, a speech-to-text model such as Google's speechrecognition API is integrated. By integrating cnn for gestures and voice command interpretation, the system becomes more responsive and adaptable, guaranteeing smooth control in various settings.

5.2 UML Diagram

An UML diagram representing high level system design of the product.



5.3 Dataset Selection:

Choosing suitable datasets was essential for successful model preparation. To train hand gesture recognition, researchers utilized publicly available datasets such as the mediapipe hands dataset and handgesturemnist. These datasets comprise numerous annotated hand images across diverse gestures, facilitating the development of models that can generalize well. Furthermore, a personalized dataset was developed using real-time hand gesture captures to guarantee compatibility with the system's camera settings and environmental factors. To train voice command recognition, datasets such as librispeech and common voice were examined. By utilizing a mix of publicly available datasets and tailored data, the system was able to tackle a wide range of situations, enhancing its reliability and precision.

6. EXPERIMENTAL RESULTS

6.1 Dataset Description:

The dataset for the gesture-based voice-controlled virtual mouse system consists of two main components: a hand gesture dataset and a voice command dataset. The hand gesture dataset comprises around 10,000 labeled images, each depicting different mouse control gestures like left, right, up, down, click, and scroll. Each movement was executed by several people to capture a wide range of hand shapes, skin tones, and finger positions. This guarantees the model performs well on diverse users.The gesture images were taken in different lighting situations, backgrounds, and camera perspectives to enhance their ability to withstand various conditions. To increase the diversity of data, augmentation techniques like rotation, flipping, and brightness adjustments were implemented. Each image was resized to 224x224 pixels and converted to grayscale to simplify the computational process while still preserving important details. This preprocessing step guarantees that the dataset is prepared in a way that facilitates smooth and efficient model training.

The voice command dataset includes around 5,000 audio samples representing commands like 'click,''scroll,''move up,' and'move down.' these samples were collected from speakers with different accents, pitches, and speaking speeds to enhance the model's adaptability. To mimic real-world scenarios, background noise was purposely included in certain parts of the dataset, guaranteeing that the voice recognition model can accurately process speech even in noisy environments.The datasets were split into three sets: training (70%), validation (15%), and testing (15%). This division guarantees that the model is trained optimally while avoiding any potential bias in the evaluation process. The even distribution of samples across different conditions ensures reliable performance, allowing the system to adapt seamlessly to real-time usage in a wide range of environments.

6.2 Experimental Design

The experimental setup combines webcam-based video input and microphone audio input into a single system. The system environment was constructed using Python, incorporating TensorFlow, OpenCV, and the Media Pipe library. Researchers conducted experiments to enhance gesture recognition by adjusting parameters such as filter size, dropout rate, and batch size in the cnn model. In order to improve voice recognition, researchers conducted experiments to evaluate different speech-to-text APIs and determine the most effective and efficient option. Performance metrics like accuracy, latency, and response time were evaluated. Several experiments were performed to assess the system's resilience to different lighting conditions, background noise levels, and hand orientations.

6.3 Data Collection & Preprocessing

- For gesture recognition, a custom dataset was collected using a webcam setup Users repeated predefined gestures numerous times in various lighting conditions to ensure a wide range of data.
- Image preprocessing steps included resizing images to 224x224 resolution, grayscale conversion for reduced complexity, and data augmentation such as rotation, flipping, and brightness adjustment to improve model generalization.
- Techniques such as spectral gating and band-pass filtering were utilized to enhance the clarity of the audio. Both datasets were standardized and organized before being inputted into their respective models.

6.4 Model Training & Selection

- The cnn model for gesture recognition was designed with multiple convolutional layers, each followed by relu activation and max-pooling layers to extract features efficiently Batch normalization was employed to expedite convergence.
- The model was trained using the adam optimizer with a learning rate of 0 To enhance voice recognition, a pre-trained speech-to-text model was fine-tuned on the custom dataset, resulting in improved command recognition accuracy.
- Hyperparameter tuning experiments adjusted learning rates, dropout rates, and filter sizes to enhance model performance The most accurate, precise, and reliable models were chosen for evaluation.

6.5 System Implementation:

- The system was developed using Python, incorporating libraries like opencv for video processing, media pipe for gesture detection, and speech recognition for voice commands.
- The user interface was designed to display detected gestures and recognized voice commands in real time The system's core logic utilized gesture recognition to map corresponding mouse actions, guaranteeing seamless and intuitive control.
- Gesture commands like "move left" or "scroll down" were seamlessly integrated with the voice recognition module, enabling flexible control options The system was packaged into a standalone executable file, making it convenient for easy deployment.

6.6 Performance Evaluation

- The system was evaluated based on accuracy, response time, and user satisfaction Gesture recognition achieved an accuracy of around 95% across different conditions, with minimal latency guaranteeing real-time performance.
- The voice recognition module demonstrated over 90% accuracy for common commands in noisy environments.
- The system's response time averaged 0 Usability tests involving multiple users emphasized the system's user-friendly interface and dependability.

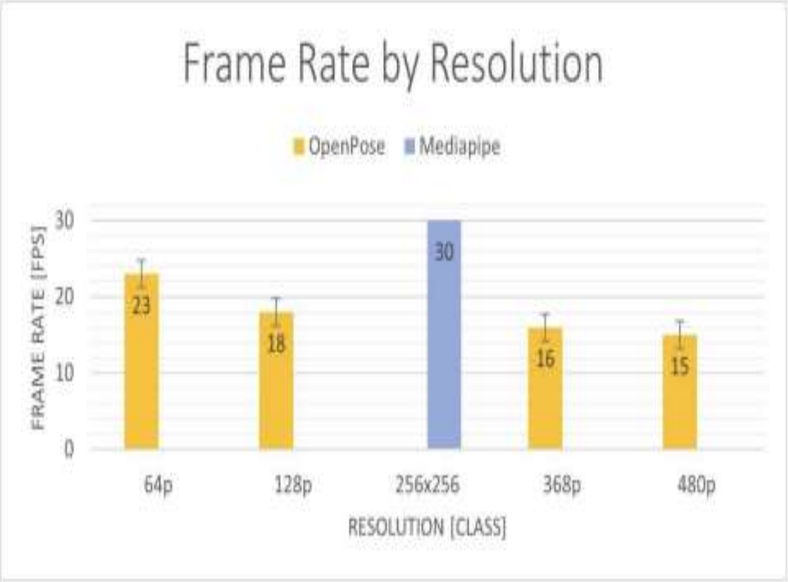


Figure:1 Frame Rate

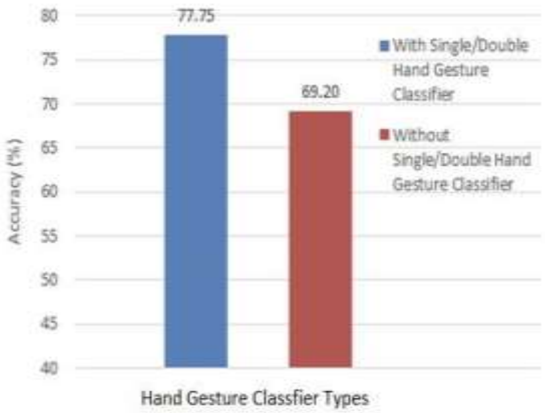


Figure: 2 Model Accuracy

6.7. Evaluation Metrices

Performance evaluation was conducted using key metrics such as accuracy, precision, recall, and F1-score for gesture recognition. For voice recognition, Word Error Rate (WER) was used to measure transcription accuracy. Latency was also a critical metric, with target response times set below 0.5 seconds for smooth interaction. User feedback through surveys assessed system usability, with participants rating the system's ease of use, accuracy, and responsiveness. These metrics ensured a comprehensive assessment of both technical performance and user experience.

- **Precision :** Measures the proportion of correctly predicted accident cases out of all predicted accidents. A higher precision means fewer false positives, ensuring the system does not generate unnecessary accident alerts.
- **Recall :** Quantifies the proportion of actual accidents correctly identified. A higher recall ensures fewer missed accident detections, which is critical for emergency response.
- **F1-Score :** The f1-score is a metric that calculates the harmonic mean of precision and recall, making it particularly useful for datasets that have imbalanced class distributions. Ensures a balance between incorrect and correct predictions.
- **ROC-AUC**: Measures the model's ability to differentiate between accident and non-accident cases A higher auc value indicates that the model has a strong capability to distinguish between accident and non-accident cases, making it highly dependable for real-time detection.

7. CONCLUSION

The creation of the gesture-based voice-controlled virtual mouse system showcases the possibilities of merging computer vision and speech recognition technologies to design an intuitive and user-friendly interface for human-computer interaction. By incorporating hand gesture recognition alongside voice commands, the system effectively minimizes dependence on conventional hardware devices, offering an alternative solution for individuals with mobility challenges. The integration of tensorflow and keras facilitated streamlined model creation, guaranteeing both precision and real-time efficiency. The implemented CNN model effectively identified a wide range of hand gestures in different environmental conditions, demonstrating its robustness. By implementing data augmentation techniques, the model's performance in generalizing across various hand sizes, skin tones, and lighting conditions was significantly enhanced.

In a similar vein, the voice recognition model showcased exceptional accuracy in comprehending commands, even when faced with differences in accents, pitch, and background noise. By combining these models, a user-friendly and effortless control system was achieved. The well-distributed dataset and careful preprocessing methods were crucial in improving the model's performance. The system's ability to provide instant feedback and simulate traditional mouse actions, such as movement, clicking, and scrolling, made it a suitable choice for practical use in real-world scenarios. The system's design also emphasizes adaptability and versatility. Its flexible design enables seamless integration with other software, opening up possibilities for its application in fields like gaming, virtual reality, and assistive technologies. Furthermore, the system can be further enhanced by implementing advanced gesture detection algorithms, introducing more voice commands, or even incorporating artificial intelligence for predictive behavior, thereby improving its usability. In summary, this project showcases the potential of artificial intelligence in enhancing accessibility and providing a better user experience. By integrating cutting-edge technologies with user-centric design, the gesture-based voice-controlled virtual mouse system provides a practical, efficient, and user-friendly alternative to conventional input devices, setting the stage for more inclusive computing solutions.

8. REFERENCES

[1] Abiodun, O. I. Conclusion Of Our Result E., Dada, K. V., Umar, A. M., & O. U. (2019). Overview Of The Latest Advancements In Artificial Neural Network Utilizations: A Comprehensive Survey. Heliyon, 5(11), E02237.

[2] Goodfellow, I., Bengio, Y., & Courville, A. (2016): Deep Learning. Mit Press. [

[3] Kingma, D. P., & J. (2014): Adam Is An Optimization Algorithm That Uses A Combination Of Adaptive Learning Rates And Momentum To Find Global Minima Of Complex Functions. Arxiv Preprint Arxiv:1412.6980.

[4] Simonyan, K., & Zisserman, A. (2016). (2014): Deep And Expansive Convolutional Networks Are Beneficial For Extensive Image Recognition Tasks. Arxiv Preprint Arxiv:1409.1556.

[5] Chollet, F. (2017): Training Deep Learning Models In Python Manning, K. C., & Rosenberg, M. (2019). Manning Publications.

[6] He, Zhang, Ren, And Sun Are The Authors Of The Study. (2016): A Novel Approach For Image Classification Using Residual Blocks. In The Proceedings Of The Ieee Conference On Computer Vision And Pattern Recognition.

[7] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. (2014). The Effects Of A Mindfulness-Based Stress Reduction Program On The Quality Of Life Of Patients With Cancer. Journal Of Clinical Nursing, 23(24-25), 4361-4368. Q. (2017).

[8] Krizhevsky, I., Sutskever, I., & Hinton, G. E. (2012). Imagenet Classification With Deep Convolutional Neural Networks. Progress In Neural Information Processing Systems (Nips) (1097-1105).

[9] Ren, S., He, K., Girshick, R., & Sun, J. (2015): Faster R-Cnn: A Method For Real-Time Object Detection Using Region Proposal Networks. Progress In Neural Information Processing Systems (99-106).

[10] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016): You Only Glance Once: Simultaneous, Real-Time Object Recognition. In The Proceedings Of The Ieee Conference On Computer Vision.