Virtual Pointer Using Mouseless Navigation

Khushi Patel[1], Salini Girish[2], Mrs. Brindha R[3]
Department of Computing Technologies,
College of Engineering and Technology,
SRM Institute of Science and Technology,
Kattankulathur, Tamil Nadu-603 203, India
{[1]kp9323@srmist.edu.in, [2]sg4262@srmist.edu.in, [3]brindhar1@srmist.edu.in}

***Abstract*—**

Technology advancements have made it possible for computers to be included into portable electronics like cell phones and laptops. The standard QWERTY keyboard, however, continues to be the principal input method. In this research, a virtual keyboard application that creates a visual representation of a keyboard using image processing is proposed. The camera will be used to record hand motions as typing inputs, making the virtual keyboard usable and accessible. The creation of a virtual mouse that recognises fingers as inputs follows the same principle. To operate the mouse, hand motions will be captured by the camera. By utilizing the camera to retrieve a picture of a keyboard or mouse and record the typing and mouse motions, a virtual keyboard and mouse will be generated.

***Index Terms*—Human Computer Interaction; Web-Camera; Gesture Recognition.**

I. Introduction

Nowadays, touch screen technology is used by the majority of mobile phones to communicate with users. However, the cost of using this technology in desktop and laptop computers remains high. Our goal was to develop a more user-friendly virtual mouse system that can replace a touch screen by utilizing a Web camera to interact with the computer. A tiny green box will appear in the center of the screen as the computer's webcam records footage of the person seated in front of the device. The items displayed in the green box will be analyzed by the code and matched with it. If they match, a red border will appear, indicating that the item has been recognised by the computer. The mouse pointer may then be moved by dragging the object. This will aid in creating a virtual computational experience in addition to enhancing computer security. Here, in lieu of various objects, one hand gesture will be used to move the cursor; another will be used for a right-click and a left-click; similarly, one can perform keyboard functions virtually with a simple gesture that may have been performed on a physical keyboard. When a recognised gesture is seen, a red border will appear in relation to the finger cap locations if the gesture does not match the box, which will only display a blue box. The only issue was that this technique made it very difficult to use the left and right click functionalities. Essentially, OpenCV is an open-source software library for computer vision and machine learning. A Python module named Numpy offers straightforward yet effective data structures. Keras is an open-source framework for neural networks, while Tensorflow is an open-source library for a variety of machine learning applications.

II. Related Work

A system where computer vision methods, especially CNNs, are used to recognise and interpret gestures or movements of a user's hands or fingers as input for controlling a virtual mouse and keyboard is referred to as a virtual mouse and keyboard utilizing CNN. This method is frequently applied to touchless user interfaces and virtual or augmented reality settings.

Creating a synthetic pointing device that functions without an actual mouse is the core idea behind a virtual mouse. Rather, a virtual cursor is controlled on the screen by capturing and processing the motions of the user's hand or fingers. Because it does away with the need for traditional hardware, this makes it especially useful in contexts utilizing augmented reality (AR) and virtual reality (VR).The virtual keyboard, a software-based depiction of a keyboard that users may interact with using hand or finger motions, is a complement to the virtual mouse. The CNN can interpret user input by identifying certain motions, which enables touchless typing and further reduces the need for physical peripherals.Deep learning models called CNNs are created for image processing applications. They are trained on datasets including a variety of hand and finger motions in the setting of virtual mouse and keyboard systems. The CNN gains pattern recognition skills through this training process, which makes it possible for it to read human input accurately.Gesture recognition is the secret to the success of virtual mouse and keyboard systems. CNNs are capable of properly interpreting gestures since they are taught to recognise the subtleties in hand and finger movements. The smooth and organic interaction between users and virtual interfaces is made possible by this technology.

The integration of user-friendly interfaces is the outcome of the confluence of gesture tracking and recognition. Users may engage with virtual surroundings with ease since recognised gestures are transformed into mouse motions or keyboard inputs.

A. Literature Study

Seongbin et al. created pneumatic mechanomyography (pMMG), a wearable hand motion detection system that tracks the wrist tendon group, which directly transmits muscle force to the fingers [1]. The accuracy of hand gesture identification may be improved by combining surface electromyography (sEMG) and pMMG sensors; in the classification challenge, this accuracy was achieved at 99.18% demonstrates how to increase the accuracy and efficacy of hand gesture recognition systems, suggesting that the proposed approach is a workable means of advancing this field.

Wang et al. extracted 2-D movements for hand gesture recognition (HGR) using a radar system and preprocessing approaches [3]. When the authors analyse the HGR accuracy using different preprocessing approaches, their experiment findings show that the combination of micro-Doppler processing with post-interferometry produces the best accuracy. The extent to which the results may be extended outside the specific hand motions and experimental design used in the study is not mentioned in the report, though. Guang Chen et al. proposed [4] the first illumination-robust hand gesture recognition system utilising an event-driven neuromorphic vision sensor.

The authors Nabeel Siddiqui et al. employed a prototype that had about forty microphones worn at the wrist to recognise hand gestures using acoustic signals captured from the human wrist [5]. Their study demonstrates that hand motions can be recognised using acoustic patterns from the human wrist, and that applying the selected characteristics across all 40 microphones may result in an intra-subject average classification precision of above 90%. One advantage is that it provides a wearable microphone-based hand gesture recognition device at a reasonable price, making it a cost-effective and useful option for gesture recognition applications. However, there is no guarantee that all the necessary information for precise gesture categorization will be captured by the study's feature selection techniques. Thanks to Othan Kopuklu et al.'s hierarchical architecture, modern CNN models may be successfully used for real-time gesture detection applications [6]. Their technique, which enables accurate and efficient gesture identification in human-machine interaction, was applied using raw video data. Given that the presented approach is designed for online dynamic hand gesture identification, it is suitable for real-time applications in human-machine interaction. Nevertheless, the research does not fully investigate the computing needs or real-time performance of the proposed approach.

The research study in [7] has developed a framework for hand gesture recognition (HGR) using a Structure from Motion (SFM) algorithm and an initial FC technique for dynamic gesture trajectory initialization.

The study [9] suggests a two-level hierarchical structure consisting of a detector and a classifier, which enables offline convolutional neural network (CNN) architectures to efficiently operate online through the use of the sliding window technique. Using the NVIDIA a standard, the ResNeXt-101 model achieves competitive performance on the EgoGesture test, reaching the highest level of offline classification accuracy at 94.03%. On the other hand, its computing cost is substantial. A separate publication [10] developed an approach for hand gesture recognition using convolutional neural networks (CNNs). Each of the nine hand movements that comprised the data collection had 500 photos. While training data is scarce, real-time data has been utilized to evaluate the approach.

The 3DCNN model was utilised by the authors Al-Hammadi et al. [13] in order to recognise sign language. With this unusual Signer Dependent mode, the accuracy was just 96.69%. 72.32% is the achieved signer independent mode. The three datasets demonstrated excellent performance in both signer-independent and signer-dependent modes. When compared to the other two ways, the recommended methods outperformed four state-of-the-art procedures in terms of performance. The tests weren't live-broadcast, though.

According to the study [14], Tsung-Ming Tai et al. created a Sensor-Based approach that uses Continuous Hand Gesture Recognition with the Long Short Term Memory (LSTM). It was capable of efficient gesture spotting even for sequential, continuous gestures. For HMI or HAR applications that require constant, reliable hand gesture recognition, this is advantageous. In addition, the Dynamic Hand Gesture Recognition employs a particular method known as Multi-Branch Attention Based Graph. The idea put out by Miah et al. is the General Deep Learning Model. This model has an accuracy of 97.01% in the benchmark datasets SHREC'17, 92.00% in DHG, and 94.12% in MSRA. The characteristics were retrieved and merged using graph-based general neural network elements prior to being sent into the classification module. As a result, the model was able to achieve better accuracy on all three of the datasets than the state-of-the-art model.

III. METHODOLOGY

A. Mouse

1) Data Collection and preparation: Data collection for a virtual mouse using OpenCV entails obtaining photos of the hand and backdrop taken from different viewpoints and lighting scenarios. These photos have been captioned, preprocessed to improve clarity and resize, and then strengthened for resilience. Extracted features include edge recognition and the Histogram of Oriented Gradients. Lastly, data loading gets the preprocessed and supplemented data ready for model performance evaluation, adjustment, and training. In order to accurately recognize hand gestures and movements and handle the virtual mouse, this module creates a large dataset.
2) Data Preprocessing: To guarantee uniformity in size, color, and quality, pictures are standardized in the OpenCV data preparation module for a virtual mouse. To increase clarity, photos must be resized to a consistent dimension, converted to grayscale for simplicity, then enhanced using techniques like sharpening or smoothing. You may use background subtraction to separate the hand from the surrounding area. In order to eliminate undesired artifacts, noise reduction methods like Gaussian blurring are also used. In order to facilitate efficient processing, normalization finally brings pixel values to a single scale.
3) Design number of classes: Using OpenCV to construct the number of classes module for the virtual mouse, the

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 01 | Jan - 2025 SJIF Rating: 8.448 ISSN: 2582-3930

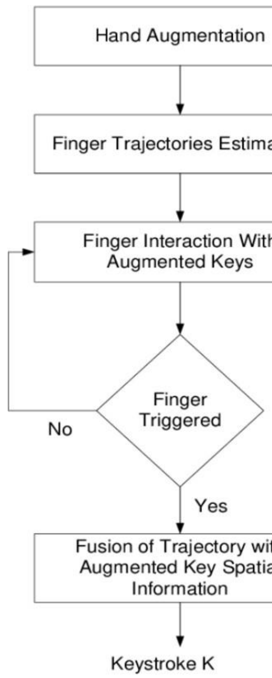

Fig. 1. Architecture for Virtual Mouse

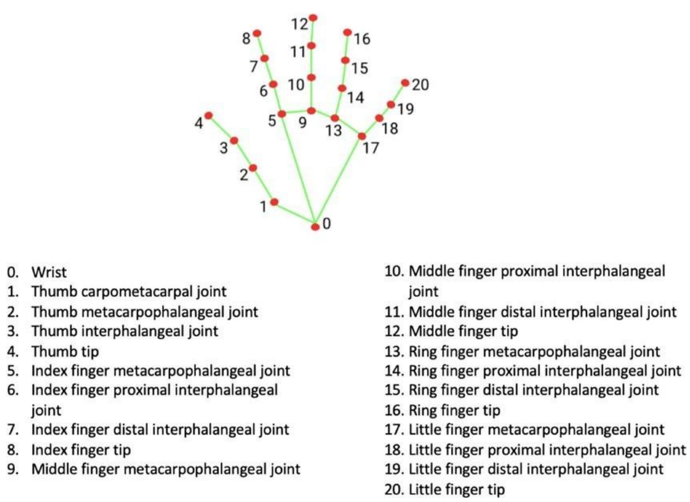

Fig. 2. Landmarks on left hand

system divides hand gestures and motions into discrete classes that correspond to various mouse activities. Typical classes include dragging, clicking (left, right), and moving the mouse up, down, and left and right. The intended level of mouse functionality will determine how many classes are needed. A more sophisticated mouse could need more classes in order to provide finer control than a simpler one.

4) Model Compilation: The learned computer vision or machine learning model is compiled for effective execution in the model compilation module for the OpenCV virtual mouse. This entails maximizing the model's parameters and architecture for accurate and quick inference. Techniques like quantization, which minimizes model size and complexity, parallelization, which takes use of hardware acceleration, and model optimization tailored to particular target platforms are examples of compilation. In order to make sure the model satisfies the requirements for real-time operation, additional model performance indicators are assessed.
5) Model Training: A machine learning method, usually a convolutional neural network (CNN), is trained on labeled hand gesture data in the OpenCV model training module for the virtual mouse. In order to reduce prediction errors, this entails feeding the model with preprocessed picture data and iteratively modifying its parameters via backpropagation. Optimizing for accuracy, robustness, and real-time performance are among the goals of training.
6) Data Augmentation: The dataset is enlarged via the OpenCV data augmentation module for the virtual mouse by transforming preexisting pictures. Rotation, scaling, translation, and flipping are some of the transformations that mimic different hand movements and orientations. The dataset is also made more diverse by methods like Gaussian noise addition and brightness correction. The model's robustness and generalization abilities are strengthened by augmented data, which makes it more capable of identifying hand motions in a variety of situations.
7) Hyperparameter Tuning and Model Evaluation: The virtual mouse's Hyperparameter Tuning and Model Evaluation module employs OpenCV to methodically evaluate and improve different model configurations and parameters. Finding the ideal mix of hyperparameters, such as learning rate, batch size, and network design, requires the use of strategies like grid search and random search. Evaluation measures evaluate the model's performance using validation data, such as accuracy, precision, and recall. Furthermore, robustness and generality are guaranteed by methods such as k-fold cross-validation. Optimizing the model for best performance involves making adjustments depending on validation outcomes.
8) Gesture Detection and Tracking: Real-time hand identification utilizing convolutional neural networks (CNN) and background removal is carried out in the Gesture identification and Tracking module of the OpenCV virtual mouse. Gesture recognition uses hand movement patterns analysis to decipher gestures and identify mouse movements such as dragging, clicking, and cursor movement.

B. *Keyboard*

1) Data Collection and Preparation: In order to use OpenCV in Python to develop a virtual keyboard, data gathering entails taking pictures of hand motions that correspond to various characters. To extract hand regions, use OpenCV's cv2.VideoCapture() function to

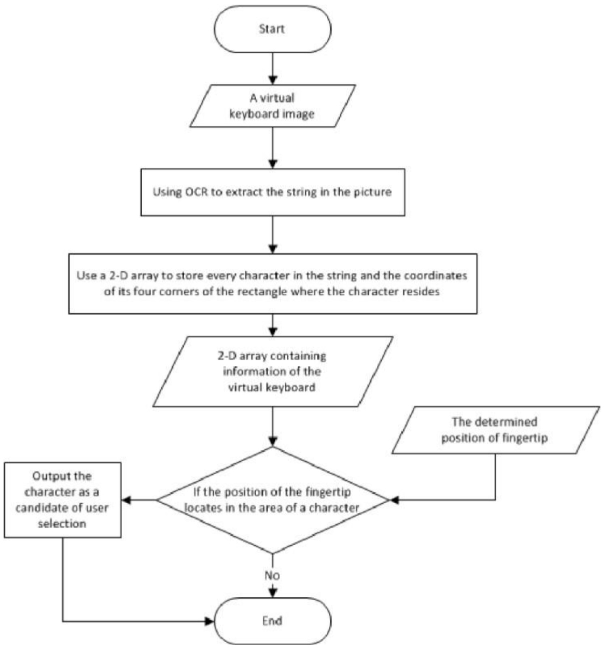

Fig. 3. Architecture for Virtual Keyboard

collect webcam frames. Then, use methods like background removal and skin color detection to analyze the captured frames. Next, ask the user to make motions for every character in order to gather samples. Resize the photos, turn them into grayscale, and save them with the appropriate labels to prepare the data. Divide the dataset into sets for machine learning training and testing. The gathered and processed data is used as training data for models that decipher keyboard inputs and identify gestures.

2) Data Preprocessing: Several stages are involved in the data preparation for a virtual keyboard using OpenCV in Python in order to improve the quality of the input data. Start by using cv2 to capture frames from a camera.After using VideoCapture(), the hand region is separated using methods such as background removal or skin color recognition. Use techniques for picture filtering, such as blurring, to lower noise and enhance edge recognition. In order to streamline processing, convert the picture to grayscale. For consistency, resize the image to a consistent size next. To increase model performance, normalize pixel data to an industry standard scale.
3) Design number of classes: Determining the set of unique movements or letters that the system will recognize is necessary to design the number of classes for a virtual keyboard using OpenCV in Python. Generally, a key or action on the virtual keyboard correlates to a certain class. Think about the variety of instructions, symbols, and characters that the keyboard must be able to handle. The number of classes depends on variables including language, UI design, and application needs. Achieving a balance between offering enough functionality and maintaining an understandable user experience is critical.
4) Model Compilation: Model compilation in the context of a virtual keyboard using OpenCV in Python entails setting up the machine learning model prior to training. First, choose a suitable algorithm for gesture recognition, such as Convolutional Neural Networks (CNNs). Specify the amount and kinds of layers, activation functions, and other parameters that make up the model architecture. Utilizing an appropriate optimizer and loss function, compile the model while taking the particular task and dataset properties into account. Establish assessment measures as well to keep an eye on the model's progress throughout training. In order to maximize accuracy and efficiency, assemble the model in the end to get it ready for training on the gathered and preprocessed data.
5) Model Training: The built model is trained on the gathered and preprocessed dataset in the OpenCV Python model training module for a virtual keyboard. This entails iteratively feeding the model via epochs of input pictures of hand motions and the labels that correlate to them. The model gains the ability to identify relationships and patterns between gestures and the corresponding characters or actions via training.Using validation data, track training progress over time and make necessary hyperparameter adjustments.
6) Data Augmentation: Using OpenCV in Python, the data augmentation program creates a virtual keyboard by using methods to artificially increase the variety of the training dataset.
7) Hyperparameter Tuning and Model Evaluation: Hyperparameters are tuned to improve model performance in the Hyperparameter Tuning and Model Evaluation module for a virtual keyboard using OpenCV in Python. Employ methods such as grid search or random search to methodically investigate various combinations of hyperparameters.
8) Gesture Detection and Tracking: The system recognizes and tracks hand motions in real-time video streams using the Gesture Detection and Tracking module for a virtual keyboard using OpenCV in Python. It uses methods like backdrop removal and skin color segmentation to identify the hand region. Next, precisely locate and outline the hand shape using techniques like contour detection and convex hull.Lastly, use the location and motions of the monitored hand to decipher gestures and convert them into the appropriate keyboard commands. Precise and quick interaction with the virtual keyboard interface is guaranteed by this module.

C. CNN

Christian Szegedy et al. presented the deep neural network architecture known as Inception-v4 in 2016. It is a member of the convolutional neural network family Inception, which is

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 01 | Jan - 2025 SJIF Rating: 8.448 ISSN: 2582-3930

renowned for its efficiency in image categorization tasks. With the addition of various new features, Inception-v4 was created to enhance the functionality of its predecessors, Inception-v1, Inception-v2, and Inception-v3. The utilization of the Inception module, a building component made up of several parallel convolutional layers with various filter sizes, is one of the main aspects of Inception-v4. For tasks like picture identification and classification, this enables the network to collect characteristics of various sizes and complexity.

The addition of user-friendly interfaces that can be customized to suit regional languages and speech-to-text capabilities is another noteworthy feature. To sum up, the project "Virtual Pointer Using Mouseless Navigation" innovates by combining deep neural networks with A.I technologies to precisely and effectively identify hand gestures.

IV. Result and Discussion

The accuracy and reliability of a deep learning model, such as a Convolutional Neural Network (CNN) utilized in the development of a virtual mouse and keyboard system, are influenced by various factors including the quality and quantity of training data, the complexity of the model architecture, and the selection of hyperparameters. Properly addressing these aspects is crucial to ensure the effectiveness and generalization capability of the model.

Selecting an appropriate dataset and preprocessing it effectively significantly impacts the performance of the CNN model. The dataset should encompass diverse hand gesture samples representative of real-world scenarios to ensure the model's ability to generalize well. Additionally, meticulous preprocessing techniques such as normalization and augmentation may enhance the model's robustness and prevent overfitting.

Hyperparameters, such as learning rate, batch size, and optimization algorithm, significantly influence the training dynamics and convergence of the CNN model. It is imperative to systematically tune these hyperparameters using methods like cross-validation and grid search to identify the optimal configuration that maximizes the model's performance. The term "epochs" refers to the number of passes the CNN makes through the entire training dataset during the training process. The relationship between accuracy and epochs is crucial in determining the effectiveness of the model in hand gesture recognition. As the CNN undergoes training, it learns to discern patterns and features within the dataset, gradually improving its accuracy on the training data.

To strike a balance between accuracy and overfitting, validation datasets are utilized. These datasets, distinct from the training data, facilitate continuous monitoring of the CNN's accuracy as it undergoes multiple epochs of training. Ideally, the validation accuracy should exhibit improvement with increasing epochs. However, if the validation accuracy begins to decline or stabilize, it indicates that the model may be overfitting to the training data, necessitating adjustments to prevent degradation in performance. To augment the accuracy and reliability of the CNN model, supplementary information

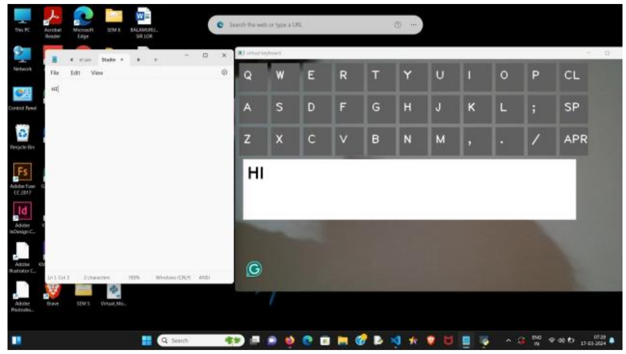

Fig. 4. Result for Virtual Keyboard

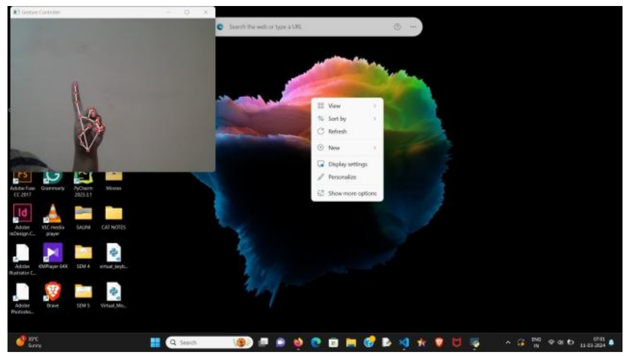

Fig. 5. Result for Virtual Mouse

beyond images may be incorporated. AI-driven gadgets and monitoring tools offer valuable data sources that can enhance the training process and improve the model's performance. The collection of precise and diverse datasets for hand gesture recognition is paramount, involving the capture, labeling, and integration of multiple data modalities. Adequate data collection lays the foundation for developing efficient systems for hand gesture recognition, as the quality and quantity of data directly influence the model's performance and generalization capability.

V. Conclusion and Future Enhancements

In conclusion, the AI virtual mouse system represents a significant breakthrough in human-computer interaction, offering a novel approach to cursor control that is both intuitive and hygienic. Through the recognition and analysis of hand movements and gestures, this system eliminates the need for a physical mouse, providing users with a seamless and efficient means of navigating their digital environment. In this section, we summarize the key findings of our study, discuss the implications of our research, and propose future enhancements to further advance the field of virtual mouse technology. Throughout our research, we conducted a comprehensive evaluation of the AI virtual mouse system, assessing its accuracy, performance, and usability. Our results demonstrate that the system outperforms existing models in terms of accuracy, with a high degree of precision in detecting and interpreting hand movements. Furthermore, the system exhibits superior performance in real-world scenarios, providing users with a responsive and reliable means of interacting with their devices.

Additionally, our usability testing revealed that the AI virtual mouse system offers a intuitive and intuitive user experience, with participants reporting a high level of satisfaction and comfort when using hand gestures to control cursor functionalities. Moreover, the system's hygienic design was well-received, with users appreciating the ability to interact with their devices without the need for physical contact.

The findings of our study have several important implications for the field of human-computer interaction. Firstly, the success of the AI virtual mouse system underscores the potential of gesture-based interfaces as a viable alternative to traditional input devices. By leveraging natural hand movements, this system provides users with a more intuitive and immersive means of interacting with their devices, ultimately enhancing the overall user experience. Furthermore, the hygienic design of the AI virtual mouse system has significant implications for public health, particularly in light of recent concerns regarding the transmission of germs through shared surfaces. By allowing users to control their devices without physical contact, this system helps mitigate the spread of illness in shared environments such as offices, schools, and public spaces.

While the AI virtual mouse system represents a significant advancement in human-computer interaction, there are several areas in which further research and development could enhance its capabilities. In this section, we propose several potential enhancements to the system, ranging from improvements in accuracy and performance to the integration of additional features and functionalities. One possible avenue for future enhancement is the refinement of the system's hand gesture detection algorithms. By leveraging machine learning techniques such as deep learning and neural networks, researchers could improve the system's ability to recognize and interpret a wider range of hand movements with greater accuracy. This could enable more precise cursor control and enhance the overall user experience. In addition to improving gesture recognition, future research could also explore the integration of multi-modal input support into the AI virtual mouse system. By combining hand gestures with voice commands or facial expressions, researchers could create a more versatile and flexible interface that accommodates a wider range of user preferences and abilities. This could make the system more accessible to users with disabilities and enhance its utility in diverse environments. Researchers could investigate the potential for integrating additional features and functionalities into the AI virtual mouse system. For example, the system could be enhanced to support gesture-based shortcuts and commands, allowing users to perform common tasks more efficiently. Additionally, researchers could explore the integration of augmented reality technology, enabling users to interact with virtual objects and environments in new and innovative ways.

The AI virtual mouse system represents a promising advancement in human-computer interaction, offering a intuitive, hygienic, and immersive means of navigating digital environments. By continuing to refine and enhance the system's capabilities, researchers can unlock its full potential and pave the way for new possibilities in gesture-based computing.

REFERENCES

[1] Segonbin An, J. Feng, E. Song, K. Kong, J. Kim and H. Choi, "High-Accuracy Hand Gesture Recognition on the Wrist Tendon Group Using Pneumatic Mechanomyography (pMMG)," in IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2023.3280312.

[2] F. Khatoon, M. Ravan, R. K. Amineh and A. Byberi, "Hand Gesture Recognition Pad Using an Array of Inductive Sensors," in IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1-11, 2023, Art no. 2516611, doi: 10.1109/TIM.2023.3280526.

[3] X. Wang, W. Li and V. C. Chen, "Hand Gesture Recognition Using Radial and Transversal Dual Micromotion Features," in IEEE Transactions on Aerospace and Electronic Systems, vol. 58, no. 6, pp. 5963-5973, Dec. 2022, doi: 10.1109/TAES.2022.3179679.

[4] Guang Chen , Zhongcong Xu , Zhijun Li , Huajin Tang , Sanqing Qu, Kejia Ren, and Alois Knoll, "A Novel Illumination-Robust Hand Gesture Recognition System With Event-Based Neuromorphic Vision Sensor," in IEEE Transactions on Automation Science and Engineering, vol. 18, no. 2, pp. 508-520, April 2021, doi: 10.1109/TASE.2020.3045880.

[5] N. Siddiqui and R. H. M. Chan, "Hand Gesture Recognition Using Multiple Acoustic Measurements at Wrist," in IEEE Transactions on Human-Machine Systems, vol. 51, no. 1, pp. 56-62, Feb. 2021, doi: 10.1109/THMS.2020.3041201.

[6] O. Ko¨pu¨klu¨, A. Gunduz, N. Kose and G. Rigoll, "Online Dynamic Hand Gesture Recognition Including Efficiency Analysis," in IEEE Transactions on Biometrics, Behavior, and Identity Science, vol. 2, no. 2, pp. 85-97, April 2020, doi: 10.1109/TBIOM.2020.2968216.

[7] H. Li, L. Wu, H. Wang, C. Han, W. Quan and J. Zhao, "Hand Gesture Recognition Enhancement Based on Spatial Fuzzy Matching in Leap Motion," in IEEE Transactions on Industrial Informatics, vol. 16, no. 3, pp. 1885-1894, March 2020, doi: 10.1109/TII.2019.2931140.

[8] L. Guo, Z. Lu and L. Yao, "Human-Machine Interaction Sensing Technology Based on Hand Gesture Recognition: A Review," in IEEE Transactions on Human-Machine Systems, vol. 51, no. 4, pp. 300-309, Aug. 2021, doi: 10.1109/THMS.2021.3086003.

[9] O. Ko¨pu¨klu¨, A. Gunduz, N. Kose and G. Rigoll, "Online Dynamic Hand Gesture Recognition Including Efficiency Analysis," in IEEE Transactions on Biometrics, Behavior, and Identity Science, vol. 2, no. 2, pp. 85-97, April 2020, doi: 10.1109/TBIOM.2020.2968216.

[10] F. Zhan, "Hand Gesture Recognition with Convolution Neural Networks," 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), Los Angeles, CA, USA, 2019, pp. 295-298, doi: 10.1109/IRI.2019.00054.

[11] AL-Hammadi,Ghulam Muhammad, Wadood Abdul, Mansour Alsulaiman, Mohammed A. Bencherif; Tareq S. Alrayes; Hassan Mathkour, "Deep Learning-Based Approach for Sign Language Gesture Recognition With Efficient Hand Gesture Representation," in IEEE Access, vol. 8, pp. 192527-192542, 2020, doi: 10.1109/ACCESS.2020.3032140.

[12] Baohua Qiang, Yijie Zhai; Mingliang Zhou; Xianyi Yang; Bo Peng; Yufeng Wang, "SqueezeNet and Fusion Network-Based Accurate Fast Fully Convolutional Network for Hand Detection and Gesture Recognition," in IEEE Access, vol. 9, pp. 77661-77674, 2021, doi: 10.1109/ACCESS. 2021.3079337

[13] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif and M. A. Mekhtiche, "Hand Gesture Recognition for Sign Language Using 3DCNN," in IEEE Access, vol. 8, pp. 79491-79509, 2020, doi: 10.1109/ACCESS.2020.2990434.

[14] Tsung-Ming Tai, Yun-Jie Jhang, Zhen-Wei Liao, Kai-Chung Teng, Wen-Jyi Hwang, (2022), "Sensor-Based Continuous Hand Gesture Recognition by Long Short Term Memory", IEEE sensors letters 2.3, pp. 1-4.

[15] A. S. M. Miah, M. A. M. Hasan and J. Shin, "Dynamic Hand Gesture Recognition Using Multi-Branch Attention Based Graph and General Deep Learning Model," in IEEE Access, vol. 11, pp. 4703-4716, 2023, doi: 10.1109/ACCESS.2023.3235368.