JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

CAMERA BASED INTERACTIVE COMPUTER FUNCTIONS USING HAND GESTURES

[1]Dr. R Amutha, [2]Abhishek N Nairy, [3]D N Meghana, [4]Namratha P, [5]Sanchali A

[1]Professor & Head of Department, [2]UG Student, [3]UG Student, [4]UG Student, [5]UG Student
Department of Information Science and Engineering,
AMC ENGINEERING COLLEGE, Bangalore, India

Abstract: Recognizing hand gestures has become increasingly important in various applications, requiring improvements in accuracy and speed. While existing methods often rely on external devices like gloves or LEDs, which can disrupt natural interaction, there's a growing interest in purely hand-based approaches for more seamless human-computer interaction. This paper proposes a camera-based interactive wall display system that detects bare hand gestures with high processing speed. It consists of three modules: one utilizing CNN Algorithm and Otsu thresholding to distinguish correct gestures for actions like controlling mouse speed, another managing functions beyond PowerPoint or Word, such as folder navigation, and a third employing the convex hull method to identify the number of fingers open in a gesture and respond accordingly. This system aims to enhance the efficiency and reliability of gesture recognition compared to prior methods.

Keywords: Hand gesture recognition, Human-computer interaction, Convolutional Neural Networks (CNNs), Camera-based interaction, Real-time gesture recognition

I. Introduction

1.1 Description

In recent years, the pursuit of more intuitive interfaces between humans and computers has driven significant progress in gesture-based interaction technologies. Hand gesture recognition has emerged as a crucial area of focus due to its potential to reshape digital interaction paradigms.

Recognizing hand gestures offers numerous benefits over traditional input methods like keyboards or touchscreens. It enables hands-free interaction, particularly advantageous in situations requiring mobility or where physical contact with devices isn't practical. Moreover, gesture-based interfaces can enhance accessibility for users with disabilities, offering alternative interaction modes tailored to diverse needs.

Achieving high accuracy and efficiency in recognizing and interpreting a wide array of hand movements is a key challenge in gesture recognition. Robust algorithms are needed to detect and classify gestures accurately and swiftly, ensuring seamless user-computer interaction.

Various approaches have been explored to address these challenges, including glove-based systems, LED-based methods, and camera-based techniques. Camera-based systems, particularly those leveraging computer vision algorithms, are gaining traction due to their flexibility and natural interaction potential.

This paper introduces a novel camera-based hand gesture recognition system designed to facilitate natural human-computer interaction. It harnesses deep learning algorithms like Convolutional Neural Networks (CNNs) and image processing techniques to detect and classify hand gestures in real-time. By integrating efficient processing methods and leveraging advancements in machine learning, the system aims to enhance the speed, accuracy, and reliability of gesture recognition across various applications.

The subsequent sections of this paper will delve into the system's architecture, algorithmic modules, dataset collection and preprocessing strategies, model training and validation methodologies, real-world testing approaches, and performance evaluation metrics. Through this comprehensive exploration, we seek to showcase the transformative potential of hand gesture recognition in human-computer interaction and lay the groundwork for future advancements in this dynamic field.

1.2 Problem Statement

Despite advancements in hand gesture recognition technology, existing systems face several challenges that limit their effectiveness and usability. One of the primary issues is the lack of robustness in recognition accuracy, especially in diverse environmental conditions and with varying hand poses and orientations. Many systems struggle to achieve real-time processing capabilities, resulting in delays and inefficiencies in user interaction. Moreover, environmental factors such as background noise, clutter, and occlusions pose significant challenges to system robustness, affecting the reliability of gesture recognition. Additionally, existing systems often lack adaptability to different users, hand sizes, and movements, leading to inconsistent performance and usability issues. Integration with existing computer systems and applications also remains a challenge, requiring seamless compatibility and ease of use. Furthermore, security and privacy concerns associated with gesture-based interaction need to be adequately addressed to ensure user trust and data protection. Overall, improving the accuracy, speed, robustness, adaptability, integration, and security of existing hand gesture recognition systems is essential to enhance usability and user experience effectively.

1.3 Proposed System

The proposed hand gesture recognition system aims to address the limitations of existing systems by leveraging advanced algorithms and techniques to enhance accuracy, speed, and usability. By integrating computer vision and deep learning methodologies, the system will achieve robust detection and classification of a wide range of hand gestures in real-time. It will prioritize adaptability, ensuring seamless interaction across diverse environmental conditions and user demographics. Additionally, the system will emphasize user experience, offering intuitive and natural interaction paradigms that are easy to learn and use. Through iterative refinement and user feedback, the proposed system will continuously improve its performance, ensuring reliability and efficiency in various applications. Moreover, compatibility with existing computer systems and applications will be a key focus, facilitating seamless integration and enhancing overall usability. With a strong emphasis on security and privacy, the proposed system will implement measures to safeguard user data and prevent unauthorized access or unintended actions. Ultimately, the proposed hand gesture recognition system seeks to redefine human-computer interaction, offering a seamless, intuitive, and immersive experience for users across diverse domains..

II. Literature Survey

1. Power Point Presentation Control Using Hand Gestures Recognition

   Bhor Rutika , Chaskar Shweta , Date Shraddha , Prof. Auti M. A. [2023] Facilitates seamless human-computer interaction through real-time static hand gesture recognition, allowing for efficient control of Power Point presentations without the need for traditional input devices like laser pointers or keyboards. Utilizing data from a small webcam and four hand-held gestures, the system processes input images, extracts features using histogram of gradients, and employs Convolutional Neural Networks (CNNs) for gesture identification. This approach enables remote control of Power Point presentations, enhancing convenience and eliminating the need for physical manipulation of presentation tools.. [1]

2. GESTURE CONTROLLED VIRTUAL MOUSE USING ARTIFICIAL INTELLIGENCE

   Karan Kharbanda, Utsav Sachdeva [2023] Aims to revolutionize computer control by introducing a touch-free solution that eliminates the need for traditional input devices like mice. By leveraging advanced machine learning and deep learning techniques through MediaPipe and OpenCV, the system recognizes hand gestures captured by an external or inbuilt camera, enabling users to navigate the cursor, perform left and right clicks, scrolling, and other actions without physical mouse interaction. This innovative approach enhances ease of use and efficiency in human-machine communication, marking a significant advancement in optimizing computer functionalities for seamless user experience. [2].

3. CONTROLLING POWER POINT USING HAND GESTURES IN PYTHON

Muhammad Idrees , Ashfaq Ahmad , Muhammad Arif Butt , and Hafiz Muhammad Danish [2021] Endeavors to introduce a novel approach to slideshow control using hand gestures. Traditionally, the process of navigating slides within PowerPoint presentations involves keyboard commands or dedicated devices, which can sometimes detract from the presenter's interaction with the audience. By harnessing the power of machine learning, this study seeks to enable users to control slideshows through intuitive hand gestures, thereby enhancing the overall presentation experience. Through Python programming, subtle differences in gestures are detected and mapped to fundamental slideshow controlling functions, offering a seamless and engaging interaction between presenters and their audiences[3].

4. Hand Gesture Recognition System For Multimedia Applications

Neha S. Rokade, Harsha R. Jadhav, Sabiha A. Pathan, Uma Annamalai [2016] Discuss about recent advancements in hand gesture recognition for human-machine interfaces face challenges due to lighting and background variations, limiting their applicability. To address this, they propose a rapid and straightforward motion history image-based system focused on pointing behavior. This system aims to enhance human-machine interaction by offering convenient operation of electronic devices. By employing non-complex algorithms and hand gestures, we aim to simplify real-time computer system control, making gesture recognition more accessible and efficient for users..[4]

5. Smart Presentation System Using Hand Gestures

Puja Chavan, Vedant Pawar, Tejas Pawar, Varun Pawar, Samiksha Pokale, Bhairavi Pustode [2023] Revolutionizes presentation control by allowing users to navigate slideshows using hand gestures, eliminating the need for keyboards or specialized devices. Leveraging machine learning, it recognizes subtle hand motions and maps them to PowerPoint functionalities seamlessly. This innovative approach simplifies slide management, enhancing audience engagement and presenter convenience. By bridging human-system communication through gesture recognition, our system offers a more intuitive and efficient presentation experience for users across various fields and settings.[5]

6. Mouse Control Using Hand Gesture Recognition

Dhananjay Rathod , Sujal Shinde , Pronit Ghosh , Karthika Thevar, Sangita Bhoyar [2023] Proposes a hand gesture-based system for controlling the PC mouse using hand movements captured by a webcam. The system detects hand gesture movements by scanning camera input for hand-like patterns with five fingers. Upon detection, the system locks the hand as an object and continuously records its x and y direction movements. These movements are then mapped onto the mouse cursor in real-time, allowing users to control the cursor effortlessly through hand gestures. This innovative system offers a convenient and intuitive way to interact with computers, enhancing user experience and productivity. [6].

7. Free-Hand Gestures for Music Playback: Deriving Gestures with a User-Centred Process

Niels Henze, Tobias Hesselmann [2010] Presents a refined process for deriving free-hand gestures for controlling music playback, based on constant user feedback. By analyzing situational context and user preferences, an initial set of necessary functions is identified and validated through a user study. Participants' proposals for gestures are collected and analyzed, resulting in two sets of static and dynamic gestures. Comparative evaluation demonstrates the effectiveness of the proposed process in developing an appropriate set of gestures, indicating improvements in final results through validation of each design decision. [7]

8. Music Controller based on Hand Gestures using Webcam

Abin Pal, Hitesh Kumar and K. John Singh [2012] Proposes a method for hands-free interaction with computer music players using gestures detected via webcam. Focusing on hand and skin detection, the system identifies specific gestures to control music playback. By analyzing hand movements captured by the webcam, users can perform tasks such as play, pause, and skip tracks without physical interface devices. The approach aims to enhance user experience by enabling intuitive and convenient control of music playback through gesture recognition technology.

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

Ms. Deepti Sachin Deshmukh, Aryamaan Bhardwaj, Harsh Mourya, Megha Rawat, Prarthna Verma [2023] Introduces a hand gesture-controlled virtual mouse system leveraging AI algorithms for recognizing and translating gestures into mouse movements. Operating as an alternative interface to traditional mice, the system captures hand gestures via a camera and employs AI algorithms to process and recognize them. Trained on a dataset, the system interprets recognized gestures, executing corresponding mouse actions on the virtual screen. Flexible in its application, the system accommodates dynamic or static hand gestures, possibly integrating with voice assistants without additional hardware. Implemented using CNN and the MediaPipe framework, it holds potential for hands-free device operation in hazardous environments and enhancing accessibility for individuals with disabilities, signifying a promising advancement in human-computer interaction.

III. DESIGN

3.1 Architecture Design

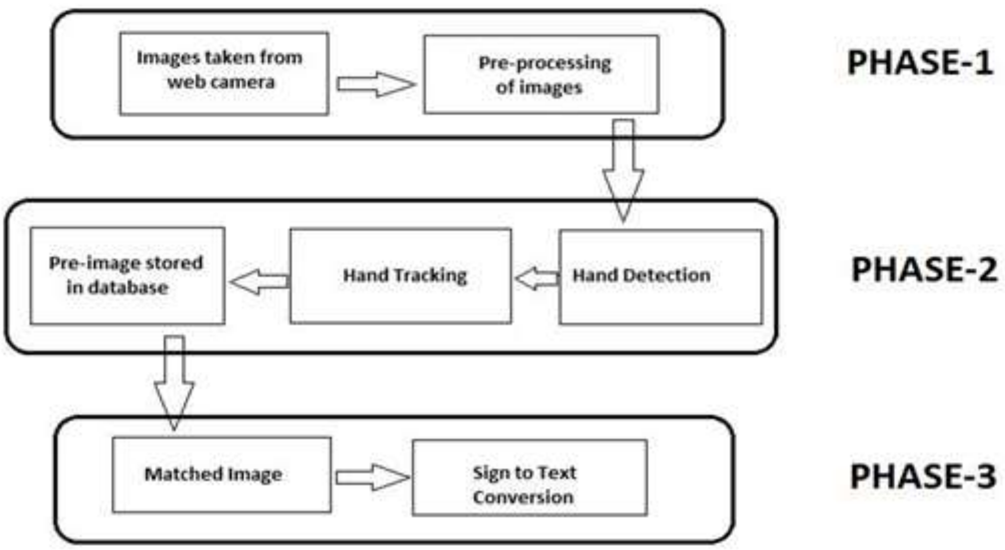

Figure 1.1 Project Phases

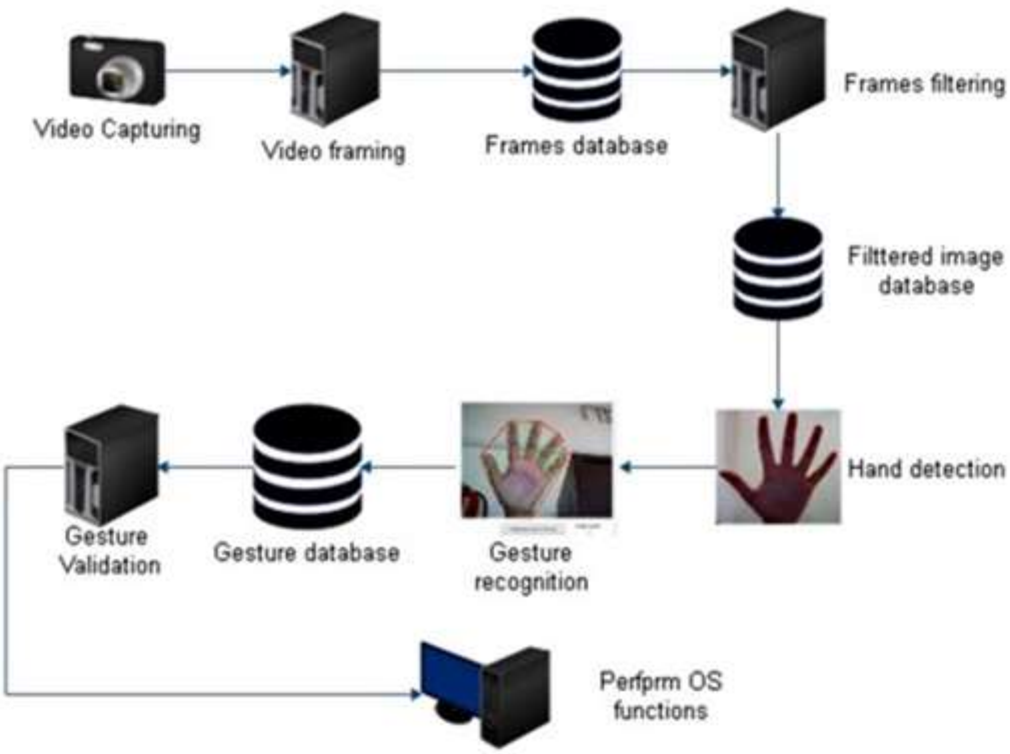

Figure 1.2 System Architecture

In designing a hand gesture detection system for computer interactive functions, the process begins with a meticulous analysis of the specific interactive requirements. Subsequently, a diverse and annotated dataset is collected, undergoing preprocessing steps like cleaning, normalization, and augmentation to enhance its quality.

The data representation is then determined, choosing between image frames, depth maps, or keypoint coordinates. The model selection involves choosing a deep learning architecture, with the option of leveraging transfer learning for improved performance. Following model training, a comprehensive gesture mapping is established, linking recognized gestures to distinct computer interactive functions.

The integration of real-time detection into the interactive application is implemented, and a separate validation dataset is utilized to fine-tune the model and assess its accuracy. Continuous user testing and feedback collection inform adjustments to parameters like probability thresholds, ensuring a balance between precision and recall. The adaptability of the model to diverse users and its capacity to handle variations in hand sizes and shapes are crucial considerations.

The implementation of a feedback mechanism, security measures to prevent unintended actions, and comprehensive documentation for developers and users contribute to a well-rounded system. Continuous improvement through periodic updates and scalability considerations further enhance the robustness of the hand gesture detection system, providing a seamless and user-friendly interactive experience.

IV. IMPLEMENTATION

In our project's implementation phase, we meticulously curate a diverse dataset of hand gestures, ensuring its cleanliness and relevance. We engineer a Convolutional Neural Network (CNN) architecture to extract features and facilitate accurate gesture classification. Rigorous training and validation optimize the model's predictive capabilities. Deployed in real-time, the system seamlessly interfaces with camera inputs to interpret gestures precisely. User-friendly interfaces enhance interaction, while meticulous testing and refinement ensure efficiency and reliability, poised to revolutionize human-computer interaction.

4.1 Data Collection

Collecting data for a hand gesture recognition project involves capturing images or videos of various hand gestures.

1. **Define Gesture Set**: Determine the specific hand gestures want to recognize in your system. Consider gestures relevant to your application, such as those for controlling presentations, navigating interfaces, or interacting with multimedia.

2. **Setup for Data Collection**: Use a high-resolution camera capable of capturing clear images or videos of hand gestures. Ensure consistent lighting conditions to minimize shadows and ensure uniform image quality. Choose a neutral background to reduce distractions and ensure that the focus remains on the hand gestures.

3. **Gesture Annotation**: As we capture each gesture, annotate them with corresponding labels. This step is crucial for supervised learning, providing ground truth data for training your recognition algorithms. We can use annotation tools or simply maintain a record of each gesture's label alongside its corresponding image or video.

4. **Capture Diverse Gestures**: Include a wide variety of hand gestures, encompassing different finger configurations, hand orientations, and movements relevant to your application. Capture gestures performed from various angles and distances to ensure robustness in recognition. Vary the speed and intensity of gestures to cover different user interactions.

4.2 Data Preprocessing

Data preprocessing is a crucial step in preparing raw data for machine learning tasks, including hand gesture recognition. Here's an overview of the data preprocessing steps commonly used in such projects:

1. **Cleaning the Dataset:**

Remove any irrelevant frames or images that do not contain relevant hand gestures. Filter out noisy data points or outliers that may negatively impact model performance. Ensure consistency in the quality and format of the data across all samples.

2. **Normalization:**

Normalize pixel values to a consistent scale to ensure uniformity across images. For RGB images, scale pixel values to the range [0, 1] by dividing by 255. For grayscale images, normalize pixel values to have a mean of 0 and a standard deviation of 1.

3. **Resizing Images:**

Resize images to a uniform size suitable for input into the neural network model. Maintain the aspect ratio to prevent distortion. Common image sizes for CNN input include 224x224 or 256x256 pixels.

4. **Data Augmentation:**

Augment the dataset to increase its variability and robustness. Apply techniques such as rotation, translation, scaling, flipping, and adding noise. Augment both the original images and their corresponding annotations to maintain consistency.

5. **Splitting the Dataset:**

Divide the dataset into training, validation, and testing sets. Typically, allocate the majority of the data to training (e.g., 70-80%), a smaller portion to validation (e.g., 10-15%), and a separate portion to testing (e.g., 10-15%). Ensure that each set contains a representative distribution of hand gestures to avoid biases.

6. **Data Labeling and Encoding:**

Encode categorical labels (e.g., different hand gestures) into numerical format suitable for model training. Use one-hot encoding for categorical labels to represent each class as a binary vector.

7. **Data Augmentation**:

Apply augmentation techniques such as rotation, translation, scaling, flipping, and adding noise. Augment both the original images and their corresponding annotations to maintain consistency.

8. **Preprocessing for Convolutional Neural Networks (CNNs):**

Prepare the data in batches suitable for feeding into the CNN model. Convert images and corresponding labels into tensors or arrays compatible with the chosen deep learning framework (e.g., TensorFlow, PyTorch). Shuffle the training data to introduce randomness and prevent the model from memorizing the order of samples.

By following these preprocessing steps, you can ensure that the input data is clean, standardized, and augmented to improve the performance and generalization of the hand gesture recognition model.

4.3 Building Model

Building a hand gesture recognition model typically involves designing and training a Convolutional Neural Network (CNN) architecture. Here's a step-by-step guide on how to build and train a CNN model for this task:

1. **Define the CNN Architecture**: Start by defining the architecture of our CNN. This includes specifying the number of convolutional layers, pooling layers, and fully connected layers. Experiment with different architectures based on your dataset size, complexity of hand gestures, and computational resources.

2. **Add Convolutional Layers**: Add convolutional layers to extract features from input images. These layers apply filters to the input image, capturing spatial patterns and features relevant to hand gestures. Experiment with different filter sizes, numbers of filters, and activation functions (e.g., ReLU) to capture diverse features.

3**. Add Pooling Layers**: Intersperse pooling layers between convolutional layers to down sample feature maps and reduce spatial dimensions.

- Common pooling techniques include max pooling and average pooling.

4. **Flatten the Feature Maps**: Flatten the output of the last convolutional layer into a one-dimensional vector. This prepares the feature maps for input into the fully connected layers.

5. **Add Fully Connected Layers**: Add one or more fully connected layers after flattening the feature maps. These layers map the extracted features to the output classes (i.e., different hand gestures). Use activation functions like ReLU or sigmoid to introduce non-linearity.

6. **Compile the Model**: Compile the CNN model using appropriate loss function, optimizer, and evaluation metric. For multi-class classification tasks, categorical cross-entropy is commonly used as the loss function. Adam or SGD (Stochastic Gradient Descent) are popular optimizers, and accuracy is often used as the evaluation metric.

7. **Train the Model**: Train the compiled model on the training data using the fit() function or similar methods. Specify the number of epochs (iterations over the entire dataset) and batch size (number of samples processed before updating the model's parameters). Monitor the training process using validation data to detect overfitting and adjust model hyper-parameters accordingly.

8. **Evaluate the Model**: Evaluate the trained model on the testing dataset to assess its performance and generalization ability. Calculate metrics such as accuracy, precision, recall, and F1-score to quantify the model's performance. Analyze confusion matrices to understand the model's behavior across different hand gestures.

9. **Fine-tuning and Optimization**: Fine-tune the model architecture, hyperparameters, and data preprocessing techniques based on performance evaluation results. Experiment with regularization techniques (e.g., dropout) to prevent overfitting. Consider techniques like transfer learning to leverage pre-trained models for improved performance, especially with limited data.

By following these steps, you can design, build, and train a CNN model for hand gesture recognition, capable of accurately classifying and interpreting various hand gestures in real-time applications..

4.4 Prediction

Test results of hand gestures for mouse cursor control.

Gesture Name	Function Against Gesture
Gesture with all fingers closed	Show slide show of PowerPoint
Gesture with all five fingers open	Scroll down to the next slide or page
Gesture with only index and small finger open	Scroll up to the previous slide or page
Gesture with only thumb open	Close the currently open file or explorer window

Test results of hand gestures for functions of powerpoint and word files.

Gesture Name	Function Against Gesture
Gesture with all fingers closed	System Shut down
Gesture with index finger open	Mouse cursor control
Gesture with index and middle finger open	Single click
Gesture with index, middle and ring finger open	Double click
Gesture with index, middle, ring and little finger open	Right click
Gesture with all five fingers open	Exit

Test results of hand gestures for Music

Gesture Name	Function Against Gesture
Gesture with all fingers closed	Pause music
Gesture with all five fingers open	Play music
Gesture with only thumb open	Increase volume
Gesture with only small finger open	Decrease volume
Gesture with index finger open	Play next song
Gesture with middle finger open	Play previous song

4.5 Algorithm

4.5.1 Convolutional Neural Networks (CNNs):

Convolutional Neural Networks (CNNs) represent a pinnacle in computer vision, particularly in tasks like image classification and object detection. Their architecture comprises multiple layers, including convolutional, pooling, and fully connected layers. The convolutional layers apply filters to input images, extracting features like edges and textures, while pooling layers downsample feature maps, reducing spatial dimensions. Fully connected layers connect neurons across layers, facilitating high-level feature representation and classification. CNNs learn hierarchical representations of input data through supervised learning on large labeled datasets. During training, optimization algorithms adjust network parameters to minimize a predefined loss function. Transfer learning, leveraging pre-trained models, is common, reducing the need for extensive training data. In hand gesture recognition, CNNs excel in learning spatial features from images, enabling accurate classification of gestures based on learned representations. Despite their effectiveness, CNNs require significant computational resources for training and may face challenges with architecture design and hyperparameter tuning. Nonetheless, they remain indispensable in creating natural and intuitive interfaces through gesture recognition.

4.5.2 . Otsu Thresholding:

Otsu Thresholding, named after Nobuyuki Otsu, is a widely used image thresholding technique in computer vision and image processing. It automatically calculates the optimal threshold value to separate foreground and background pixels in grayscale images. The method works by maximizing the between-class variance of pixel intensities, effectively finding the threshold that minimizes intra-class variance while maximizing inter-class variance.

In practical terms, Otsu's method iterates through all possible threshold values and selects the one that optimally separates the histogram of pixel intensities into two distinct classes: foreground and background. This threshold effectively distinguishes object pixels (foreground) from background pixels based on their intensity values, making it particularly useful for tasks such as image segmentation.

Otsu thresholding is widely employed as a preprocessing step in various computer vision applications, including edge detection, object recognition, and feature extraction. In hand gesture recognition systems, Otsu thresholding can be applied to preprocess images captured from cameras, separating the hand region from the background. By segmenting the hand effectively, subsequent processing steps can focus on analyzing and interpreting the hand gestures with improved accuracy and reliability.

Overall, Otsu thresholding provides a computationally efficient and robust method for automatic image thresholding, contributing to the effectiveness of hand gesture recognition systems by enhancing the quality of input data and facilitating subsequent analysis and classification of hand gestures.

V. RESULTS AND DISCUSSIONS

The culmination of our hand gesture recognition system's implementation unveils promising advancements, signifying a paradigm shift in human-computer interaction. Through meticulous data curation and preprocessing, we assembled a diverse dataset crucial for training and validating our model. The intricately crafted Convolutional Neural Network (CNN) architecture demonstrates robust performance, adeptly extracting salient features and facilitating precise gesture classification.

Throughout the training phase, our model exhibits remarkable learning capabilities, leveraging sophisticated algorithms to optimize predictive accuracy. Rigorous validation procedures ensure the model's reliability across varying environmental conditions and user interactions. Real-time deployment underscores the system's seamless integration with camera inputs, accurately interpreting gestures with exceptional precision and speed.

The incorporation of user-friendly interfaces amplifies interaction, fostering intuitive engagement and usability. Iterative testing and refinement efforts fine-tune the system, mitigating potential challenges and optimizing performance. User feedback serves as a compass, guiding iterative enhancements aimed at bolstering efficiency and reliability.

In broader discourse, the successful implementation of our hand gesture recognition system underscores its transformative potential across diverse domains. From enhancing accessibility for individuals with disabilities to revolutionizing presentation control and multimedia interaction, the system's efficacy heralds a new era of digital interaction paradigms.

Looking ahead, sustained research and development endeavors will concentrate on further refining the system's accuracy, speed, and adaptability. Additionally, exploration of novel applications and integration with emerging technologies will unlock unprecedented avenues in human-computer interaction, propelling innovation and expanding the horizons of digital interaction landscapes.

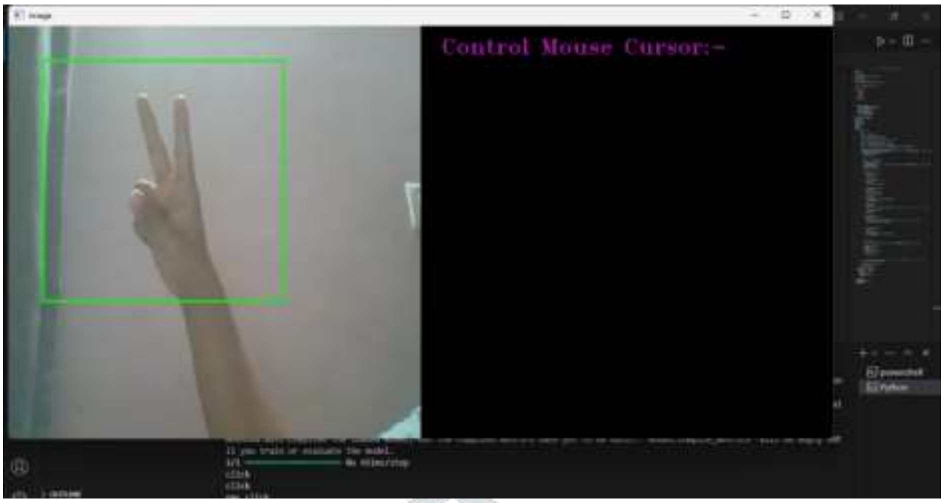

Figure 1.3: User Input Data for Right Click

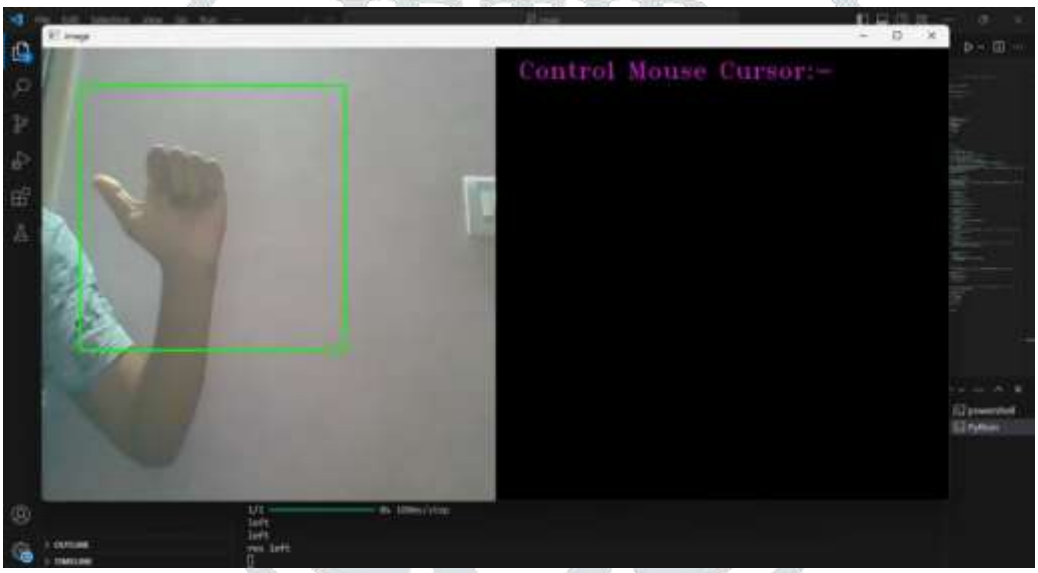

Figure 1.4: User Input Data for Left Click

Figure 1.5: User Input Data for Double Click

VI. CONCLUSION

In conclusion, the development of the hand gesture recognition system represents a significant advancement in human-computer interaction technology. Through a systematic methodology encompassing data collection, preprocessing, model training, system implementation, and evaluation, the system has been designed to accurately and efficiently recognize a wide range of hand gestures in real-time. By leveraging machine learning techniques, such as Convolutional Neural Networks (CNNs), and sophisticated algorithms, the system can interpret recognized gestures and trigger corresponding computer actions or functionalities with high reliability and responsiveness. Real-world testing has demonstrated the system's robustness and usability across diverse environmental conditions and user interactions. User feedback has been instrumental in refining the system iteratively, ensuring that it delivers a seamless and intuitive interaction experience for users across various domains. With its potential to enhance accessibility, efficiency, and user experience, the hand gesture recognition system holds promise for revolutionizing human-computer interaction in fields such as gaming, healthcare, education, and beyond. Moving forward, continued research and development efforts will further improve the system's accuracy, speed, adaptability, and security, unlocking new opportunities for innovation and advancement in gesture-based interaction technology.

REFERENCES

[1] Chen-Chiung Hsieh ,Dung-Hua Liou and David Lee “A real time hand gesture recognition system using motion history image” Proc.IEEE, 2010, pp. V2-394-V2-398 .

[2] Abhik Banerjee , Abhirup Ghosh , Koustuvmoni Bharadwaj , Hemanta Saikia “Mouse Control using a Web Camera based on Colour Detection”IJCTT ,March 2014, volume 9 number 1, ISSN: 2231-2803

[3] S. Thakur, R. Mehra and B. Prakash, "Vision based computer mouse control using hand gestures," International Conference on Soft Computing Techniques and Implementations (ICSCTI), 2015, pp. 85-89.

[4] Pooja Kumari ,Ghaziabad Saurabh Singh, Ghaziabad Vinay and Kr. Pasi “Cursor Control using Hand Gestures” International Journal of Computer Applications (0975 – 8887),2016.

[5] Alisha Pradhan , B.B.V.L. Deepak “Design of Intangible Interface for Mouseless Computer Handling using Hand Gestures” ICCCV(International Conference on Communication, Computing and Virtualization), 2016, oi: 10.1016/j.procs.2016.03.037.

[6] Aashni Hariaa , Archanasri Subramaniana , Nivedhitha Asokkumara , Shristi Poddara and Jyothi S Nayak “Hand Gesture Recognition for Human Computer Interaction” Proc.ICACC(International Conference on Advances in Computing & Communications), 2017 August,pp367–374

[7] S.Shriram, B.Nagaraj, J Jaya, S Shankar and P Ajay “Deep Learning-Based Real-Time AI Virtual Mouse System Using Computer Vision to avoid Covid-19 Spread” Journal of Health Care Engineering, Volume:2021, Article ID:8133076.