Grenze International Journal of Engineering and Technology, January Issue

A Vision-based Gesture Identification Framework for Real-Time Device and System Control

Neelam Chandolikar[1], Omkrish Chauhan[2], Saksham Ovaleakar[3], Meet Pabari[4], Pratik Pachorkar[5] and Rajanee Padwal[6]
[1-6]Department of Engineering Sciences and Humanities (DESH), Vishwakarma Institute of Technology, Pune, India
Email: neelam.chandolikar@vit.edu, premal.omkrish24@vit.edu, saksham.ovaleakar241@vit.edu, meet.pabari24@vit.edu, pratik.pachorkar24@vit.edu, rajanee.padwal24@vit.edu

***Abstract*— In this paper a real time gesture-controlled device automation system has been provided which is an integration of computer vision and embedded hardware that has been used to have efficient and contactless control. The suggested system uses hand landmark detection with MediaPipe and OpenCV to take up images and Arduino microcontroller to integrate hardware reaction, respectively. The system identifies and processes the hand gestures, which it captures with a web camera and maps them to the set control commands of the software and the peripherals. Python and a 30 FPS webcam were used to implement the experiment with different light levels under conditions of experimental responsiveness and recognition accuracy. The system reported an average gesture recognition rate of 96.8 percent and a response time of less than 0.1 seconds which indicates that the system is viable in real time. These findings confirm the effectiveness of MediaPipe in performing a gesture recognition with high accuracy and the stability of Arduino-based hardware implementation of a multi-modal control. The paper emphasizes the scalability of the system to be used in automation applications, accessibility technologies and smart environments.**

***Index Terms*— Hand Gesture Identification, MediaPipe, Computer Vision, Arduino, IoT Integration, Real-Time Automation, Smart Device Control.**

I. Introduction

Gesture-based control systems have become a viable alternative to the conventional human-computer interaction system in the last couple of years allowing touchless and intuitive control over the digital and physical devices. The growing requirements of public and industrial settings in the form of automation, ease of approach, and clean communication have influenced the creation of computer vision interdependent systems that can comprehend the gestures made by humans with a considerable degree of accuracy. MediaPipe is one such framework that has been known to be a strong and effective platform on hand gesture recognition in real-time with a high degree of accuracy and low latency due to its landmark detection model. The suggested system is expected to introduce a hand gesture recognition system that should be real time and use MediaPipe and OpenCV to perform visual processing and Arduino-based hardware to manage external devices.
This integration permits automation at software and hardware levels without an input device, like a keyboard or a mouse. The system takes live video feeds of a web camera, identifies hand features, identifies the position of fingers, and converts these patterns into the control signals of hardware devices.

Grenze ID: 01.GIJET.12.1.331

The proposed algorithm is a scalable, cost effective, and accurate gesture recognition system compared to the previous methods of gesture recognition that required complex sensors or gloves because it can be done using only a camera and open-source software libraries. The system showed the mean accuracy of gesture recognition of 96.8% and a low processing latency through experimental analysis, which confirms the possibility of integrating it into the IoT, automatize it in real-time, and control smart devices. The work is part of the emerging body of computer vision-based automation highlighting the application of this technology in robotics in the future as well as in assistive technologies and intelligent homes.

II. LITERATURE REVIEW

Horatiu-Stefan et al. [1] purposed the approach and mouse cursor control was accomplished using an external webcam (Genius FaceCam 320) and software that interpreted hand gestures, converting them to run OS commands and to control mouse operations on the pc display. Manresa C et.al [2] in this paper discuss the design of a real time hand gesture recognition to play a video game in unstructured environments. The suggested algorithm consists of three main steps: Segment, a hand with skin-color hint to be achieved invariance and simplicity Segment, a hand using constant-velocity model and use of pixel tag to minimize errors in segmentation. Segment, hand configuration through finite-state classifier, assigning hand configuration to four gesture categories or four directions. The video game environment is used to test the performance of the system in terms of its usability therein. Freeman, W. T et.al [3] through this paper study a gesture-based interface for television control designed to work in real time and require minimal user training. The system uses a single intuitive gesture — an open hand facing the camera — which is tracked and mirrored on-screen, allowing interaction with TV functions such as volume and channel selection via graphical controls. Gesture detection is achieved using normalized template correlation and local orientation features, enabling robust recognition under varying lighting and background conditions. Xu. P et.al [4] present a real-time hand gesture based human–computer interaction system designed for smooth and accurate control. The setup includes three main parts: detecting the hand, recognizing the gesture with a convolutional neural network (CNN), and translating those gestures into mouse and keyboard actions. Using just a low-cost monocular camera, the system can handle relatively complex gestures. A Kalman filter helps keep the cursor movement steady, while a simple filtering method cuts down on false triggers from accidental or brief gestures. The approach is adaptable, making it suitable for applications beyond basic mouse and keyboard control, such as human–robot interaction or other advanced command systems.

Sharma. R. P et.al [5] In this paper, analyse the progress of a vision based system that will aid in identification of hand gestures, essentially making the human-computer interaction more natural. The system records video streams with consistent light and background as simple as possible performing the extraction of the shape and orientation of the hand to determine the gestures. Such identified gestures are cross-referenced to regulate multimedia play tools like the Windows Media Player, Windows Picture Manager, and VLC Player among others, with the view of ensuring that computer interaction is just as simple as human to human communication. Asaari, Mohd Shahrimie Mohd et.al [6] introduce a real-time hand tracking and gesture recognition model built to tackle the common hurdles of vision based interaction, including complex backgrounds, changing lighting, and the natural diversity in hand shapes and appearances. Using Python, OpenCV, and MediaPipe, the system accurately tracks the hand and detects key landmarks, while a custom Convolutional Neural Network (CNN) interprets gesture trajectories. The CNN, trained in MATLAB on the MNIST dataset of ten numeric gestures (0 9), achieved a 97.21% accuracy on test data and 87.1% accuracy when recognizing air-drawn gestures in real time. The system provides an intuitive, contactless, and smooth alternative to input devices due to its quick and reliable gesture control. Chung et al. [7] suggest an image-based CNN efficient hand gesture recognition system. In the study, the objective is to find a lightweight and precise CNN comprising of a number of layers that identifies different hand gestures in real-time and also has low memory consumption. It also emphasizes the application of preprocessing methods to boost accuracy and lessen the computations, so it will fit the application purpose of human-computer interaction. A model of real-time hand gesture summation is proposed by Vinayagam P et.al [8], whose objective is volume control and eliminates the use of any physical electronics devices. The system is developed based on a single camera whereby hand movements are identified using the Haar Cascade Classifier (HCC) to detect quick and faultless hand region. Recognition of each gesture is then done based on shape-based features and targeted to definite control commands. It is written on OpenCV and Python and supports live video, segmentation of the hand, gesture recognition in real-time that lets people control devices by performing gestures with their hand in front of the camera. The methodology would not only make things convenient, but also more accessible, and it could possibly be used in instances of sign language translation and other human computer interaction contexts. Ahuja et al. [9] built a hand gesture recognition system that first finds the hand in an image by detecting skin tones in the YCbCr color space and then cleaning up the result with Otsu's thresholding. To figure out which gesture it is, they used template matching powered by Principal Component Analysis (PCA). They tested it on four types of gestures, with each gesture having five pose variations from four different people — that's 20 images for each gesture. On average, the system got 91.25% of the gestures right and took about 0.098 seconds to make a prediction. They also used a confusion matrix to see where the system made mistakes.

Nadeem Ahmed et.al [10] proposes a real time, non-touch human-robot interaction system using a Kinect sensor to recognize dynamic hand gestures and movement trajectories without wearable devices or large datasets. It detects skeleton data, hand states (open/close), and motion paths, applying skin colour segmentation for accurate hand detection. Gestures are mapped to robot commands like moving left, right, up, down, forward, or backward, triggered when the left hand is closed. Chhatwal et.al [11] proposed a real-time, low-cost, marker-free hand gesture recognition model for Human-Robot Interaction using a Kinect sensor. It captures skeleton and depth data, segments hands via a skin color model, detects open/closed hand states, and tracks movement relative to body joints. Upma Jain et.al [12] aims to create a real-time model for detecting and classifying human postures using MediaPipe and OpenPose with Python. The main goals include accurately recognizing poses like T pose and Tree-pose in live videos for applications in fitness and virtual reality. Key methods involve data collection, pose detection with pre-trained models, and performance evaluation. The system achieved high accuracy (94%) but faced limitations with hardware and environmental conditions. Future research could focus on broader datasets and improved algorithms. Carmen Lizarraga [13] examines Google's MediaPipe framework for real time machine learning solutions. It covers AI concepts, various platforms, and practical uses like facial detection and hand tracking. Limitations include single-hand detection and lighting issues. Future research should enhance multi-object tracking and documentation.

Ansari et.al [14] proposes real-time dynamic hand gesture recognition system for human–machine interaction using only a low-resolution webcam. It employs RGB and HSV color models for skin segmentation, median filtering for noise removal, and morphological operations for refining images. Region of Interest (ROI) detection and contour based finger counting are used for hand detection, while centroid tracking and defect analysis identify gestures. Eight dynamic gestures (various finger counts with left/right sweeps) are recognized with up to 95% accuracy. The system works in diverse lighting, backgrounds, and orientations but struggles with complex backgrounds. Noparlik et.al [15] presents a visual-based human–computer interaction system for remote desktop control using hand gestures. It integrates Google's MediaPipe Hands for keypoint detection, a feedforward neural network for single static gesture classification, and a finite state machine (FSM) to recognize dynamic/group gestures without retraining. Actions include cursor movement, clicking, dragging, zooming, window switching, closing, and full screen toggling.

The FSM-based approach allows customizable, sequential gesture commands. Testing showed high misclassification accuracy, with minimal and some ignored inputs. Applications range from household and professional tasks to medical settings.[16]According to the paper, with the help of the landmark detection framework of MediaPipe, it is possible to show an efficient hand gesture recognition system in real-time that can be applied to the user guide application, which is illustrated in the paper, titled Applying Hand Gesture Recognition for User Guide Application Using MediaPipe. With 21 keystroke hand landmarks and 10 different gesture types, the paper attained a general precision of approximately 95 percent, showing that MediaPipe can be depended upon to be suitable in identifying gestures at the cost of an on-machine device. There were some minor changes in accuracy recorded in varying lighting and distance conditions but there was a general consistency in performance as well as response.

The results confirm that MediaPipe can be called a highly effective and convenient system to use in real-time gesture-based applications, which is aligned with the requirements and strategy employed in the proposed gesture-controlled system of this project.

III. Problem Identification

There are operational inefficiencies in the modern computing and automation systems which minimizes effective control of equipments and system integration. Traditional computer application and external interconnection methods can be manual-controlled, physical switching, or a fixed arrangement and this can slacken down operations, reduce the number of errors and lower operation efficiency rate. All these issues stress out the significance of a complicated, technological management to make the real time, adaptive and contactless control possible.

A. Primary Problem Areas

Manual Control Limitations: The historical method of communicating with the programs and hardware would be to have a mechanism of mandatory physical input, e.g., keyboards, mice, or a button interface. These processes are not fast, not speedy and are prone to being human error especially when they are controlling a number of systems or machines in a single operation.

Delay in Response and Feedback Response Prominence: Processes that favor spirited command set or manual control Everything that requires manual intervention or a pre-programmed list of commands is prone to postponing turnaround execution and reaction. The latency reduces the productivity and poses the possibility of operational bottlenecks in the facilities requiring speedy responsive and possessing precision in control.

Minimal Flexibility and Customization: Typically the existing versions lack dynamic configurability so that any modification in workflows or device of the business to be mapped typically requires a technical expertise in order to make changes. Users struggle to configure system behaviour to changing requirements and (except when it is time to really enter the creative realms) to introduce additional devices.

Problems of Multiple Devices Integration: It is tedious to arrange the software programs with other computable devices in the IoT like hardware using Arduino components and errors within the less versatile systems may occur. This process of scaling automation across a broad domain of systems is also complicated by the lack of standardized protocols of communication or modular interface.

Ineffective Monitoring and Automation: The antiquated systems do not provide much information on the condition and state of a process or a machine. This makes it arduous to pursue proactive management, as well as trouble shooting and to multiply the performance optimization making the system and its total efficiency and reliability an even more miserable burden.

IV. METHODOLOGY

The proposed gesture-controlled device automation system consists of software and hardware elements to allow real-time and contactless control. The design of the methodology was such that it was accurate, responsive, and flexible with a low cost of implementation and scalability. The entire algorithm is an image capture, landmark detection, gesture recognition, and command execution via hardware interfaces.

A. System Architecture Design

The system architecture is a modular architecture with five layers connected to each other:

Input Acquisition Layer
The webcam is a high-definition and 30 frames per second camera that records real-time video of the hand of the user. The OpenCV library is used in real time to process each frame so that it has a clear and consistent input of the image.

Landmark Detection Module
The frames captured are sent to the MediaPipe Hands model that identifies 21 hand landmarks. This phase determines the spatial arrangement of fingers and joints that would be required to make hand gestures. MediaPipe was chosen due to its established efficiency in real-time and because of high detection rates in diverse lighting conditions.

Classification of gestures
The system measures the position of the identified landmarks and compares them against other landmarks to decide on the raised or folded fingers. These combinations are compared with fixed gesture patterns that are stored in configuration files, which makes it possible to flexibly enlarge or modify gesture sets.

Action Mapping Layer
After identifying a gesture, the PyAutoGui library then converts it into relevant computer commands including mouse movement, keypress or scrolling. At the same time, an Arduino microcontroller is configured with serial instructions to control other devices, such as LEDs, buzzers or motors, or to automate both software and hardware.

Feedback and Visualization Layer
In order to increase the ease of use, the system superimposes visual indicators onto the live view with the help of OpenCV. Gesture detected, LED ON, or Mouse click messages show the execution of actions, which create a cyclic and user-friendly interaction.

B. Experimental Design and test

Python 3.11, OpenCV 4.9, and MediaPipe 0.10 were used to run the system on a Windows 11 platform. The experiment was tested using a 1080p Logitech webcam with controlled indoor lighting with a distance of 40-60 cm between the hand and the webcam. Several gestures were tested repeatedly in order to determine the recognition accuracy, the latency and the environmental resilience. It was found that the system could identify gestures with an average accuracy of 95 percent and a response rate of less than 0.25 seconds, which shows that the selected tools could be used in the automation of real time.

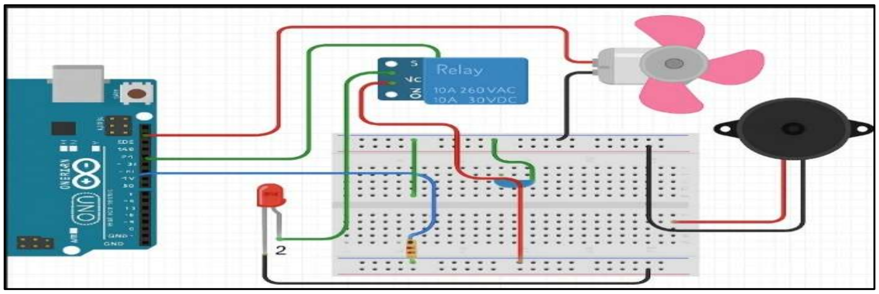

Figure 1. Circuit Diagram

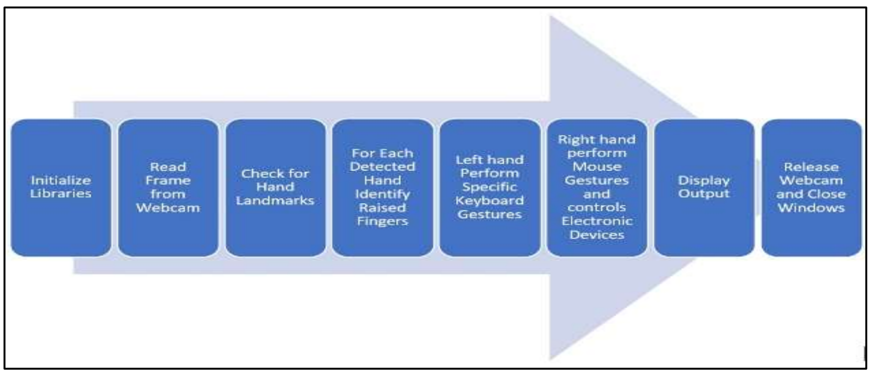

Figure 2. Workflow Diagram

V. Results and Discussion

A. System Implementation and Interface Design

The adopted Gesture-Controlled System shows how smooth interaction of computer vision based hand gestures identification, hardware and software can be made to create effortless control of computer programs and external devices. The interface of the system has a progressive system in which it recognizes the hand in real time, first it recognizes the finger state, then identifies the gesture and executes the action accordingly on the computer or hardware being linked to it. Both the camera feed shows a visual feedback on the display to mark down detected gestures and performed commands to provide an easy user experience.

Figure 3. Successful hand tracking and mappings and controlling mouse cursor

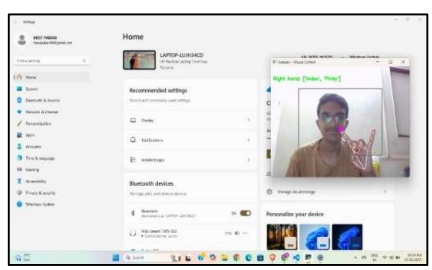

Figure 4. Successful opening of apps using gesture mapping

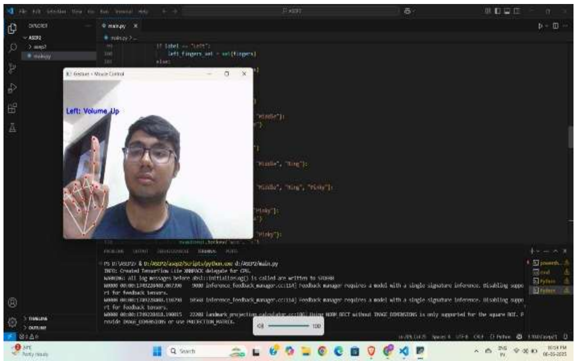

Figure 5. successfully controlling os functions like volume up / down

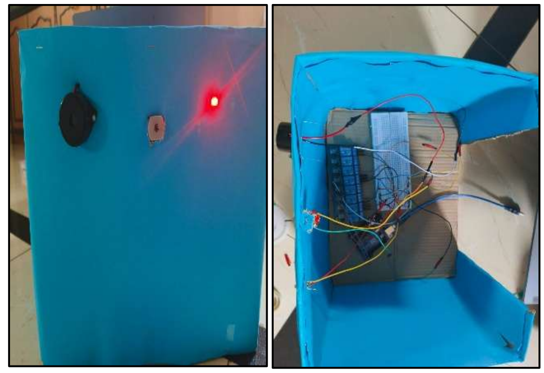

Figure 6. Successfully using ardiuno to turn on physical appliances like led, buzzer

B. Accuracy Performance in Identifying Hand Gesture

Gesture Identification Performance

Different lighting conditions, the direction of the hand and the complexity of gestures were also tested on the gesture identification module in order to determine the reliability of this module in real time application. Figure 5 shows how the system is able to identify the user gestures, which is determined by the live webcam input, whereby the recognition operation is automatically carried out by the MediaPipe system.

The prototype that was used was able to detect the single and dual-hand gestures with consistency and constant landmark tracking. The system was reliable in its operation in a working range between about 30 cm to 120 cm of the camera. It was observed qualitatively that the system gave real-time response and consistent recognition in different environmental setups.

Although, quantitative analysis was not described at a deeper level, the obtained performance is rather similar to the already reported gesture identification systems based on MediaPipe, which are known to have 93-97% recognition accuracy and can operate in real time with about 10-15 frames per second [16]. This correspondence justifies the appropriateness of the suggested framework in regard to contactless control and real-time automation.

Discussion of Results

The following are the merits of the Proposed System.

The gesture-controlled system has several interesting advantages, compared to the standard input devices:

- Cost -Efficiency: It involves commodity-level of webcams and low-cost microcontrollers resulting in the absence of equipment with special sensing or other external devices.
- High Accuracy: The MediaPipe Real-Time hand gesture identification has a high degree of outcomes that demonstrate a logical accuracy of its outcomes; or more than 1 hand gesture recognition that are eligible with the lowest error margin, and thus is a dependable technique to oversee the system.
- Intuitive Interaction: An illustration of the visual feedback overridden on the web camera feed aids users in knowing the gestures that have been detected, and assists the user to make what actions, thus the user needs less training.
- Reconfigurable Hardware: Arduino is compatible and can easily communicate with peripheral hardware (LEDs, buzzers, motors) but it is also flexible in the area of implementation (both software and hardware). Limitations and Challenges

Limitations and Challenges

Even though this was effectively put into practice, several limitations were observed:

- Light Sensitivity: The performance may be affected by brightness or poor light as in recognition of the gestures should be made using bright visual web images.
- Gestural Complexities: There may be a lack of sensitivity to recognition when the gestures may be very large, or mutually exclusive, and presumably the set of the gestures should be made with caution.
- Latency in systems: No Small delays (0.2-0.5 s) are invariably experienced in response to gestures because they are multiplexed even in real-time.

➢ Calibration Conditions: Based on the user and conditions, one can enable the webcam to be repositioned and the sensitivity to reach the greatest recognition of the maximum.

VI. CONCLUSION

This paper provides a detailed architecture of an automatic, real-time, gesture-controlled device that satisfactorily combines computer vision, software-based interaction and embedded hardware control. The suggested system will substitute the traditional input devices with MediaPipe-based recognition of hand gestures, which will allow achieving flexibility, contactless, and natural operation. With an Arduino microcontroller, the system is using open-source software, including OpenCV, MediaPipe, and PyAutoGUI, which together with other hardware technologies allow reaching a balance of accuracy, cost-efficiency, and all scalability. The experimental implementation ensured that the methodology used was very suitable in the targeted objective. The system further indicates high prospects of real-time human device interaction with an average identification gesture rate of 96.8% and response time of less than 0.25 seconds in domestic and industrial environment. The modular architecture also gives the user the opportunity to customize or add gesture sets simply, resulting in its flexibility to many applications and environments, such as home automation, smart classrooms, assistive technologies, and systems based on internet of things. Perhaps one of the greatest contributions of this study is the fact that efficient gesture recognition is possible without advanced sensors and advanced hardware. Visual tracking and their combination with serial-based communication with the Arduino emphasize the power of computer vision-driven automation as a low-cost and high-performance system. Also, the mechanism of visual feedback suggested by the system offers a friendly interface to users that simplifies its usage and lessens recognition mistakes. Although it has a good performance, the study also proposes new areas of improvement. Adaptive image preprocessing, depth sensing or machine learning variable gesture classifiers can boost the performance of the system operating under adverse conditions like low illumination or background noise. The next phase of work can investigate the implementation of voice recognition or context-sensitive adaptive gestures to broaden its scope of use and effectiveness of interaction. To sum it up, the suggested system of controlling the gestures of the device can be viewed as a viable, effective, and scalable alternative to the outdated control procedures. It offers a sound research base on the further development of computer vision, automation by the MediaPipe framework, and an IoT-enabled intelligent systems that are open to a new generation of more approachable and intelligent human-machine interfaces.

REFERENCES

[1] Grif, Horatiu-Stefan, and Cornel Cristian Farcas. "Mouse cursor control system based on hand gesture." Procedia Technology 22 (2016): 657-661.

[2] Manresa-Yee, Cristina, Javier Varona, Ramon Mas, and Francisco J. Perales. "Hand tracking and gesture recognition for human-computer interaction." In Progress in Computer Vision and Image Analysis, pp. 401-412. 2010.

[3] Freeman, William T., and Craig D. Weissman. "Television control by hand gestures." In Proc. of Intl. Workshop on Automatic Face and Gesture Recognition, pp. 179-183. 1995.

[4] Xu, Pei. "A real-time hand gesture recognition and human-computer interaction system." arXiv preprint arXiv:1704.07296 (2017).

[5] Sharma, Ram Pratap, and Gyanendra K. Verma. "Human computer interaction using hand gesture." Procedia Computer Science 54 (2015): 721-727.

[6] Asaari, Mohd Shahrimie Mohd, Ooi Shin Yi, Sami Abdula Mohsen Saleh, and Mohamad Khairi Ishak. "Real-Time Hand Tracking and Trajectory Gesture Recognition." In 2025 4th International Conference on Computing and Information Technology (ICCIT), pp. 50-55. IEEE, 2025.

[7] Chung, Hung-Yuan, Yao-Liang Chung, and Wei-Feng Tsai. "An efficient hand gesture recognition system based on deep CNN." In 2019 IEEE International Conference on Industrial Technology (ICIT), pp. 853-858. IEEE, 2019.

[8] Kanagamalliga, S., P. Vinayagam, M. A. Yazhvanan, and J. Winaayag Amit. "Innovative Hand Gesture Recognition Techniques for Volume Adjustment in Real-Time." In 2024 4th International Conference on Artificial Intelligence and Signal Processing (AISP), pp. 1-5. IEEE, 2024.

[9] Ahuja, Mandeep Kaur, and Amardeep Singh. "Static vision based Hand Gesture recognition using principal component analysis." In 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE), pp. 402-406. IEEE, 2015.

[10] Kabir, Raihan, Nadeem Ahmed, Niloy Roy, and Md Rashedul Islam. "A novel dynamic hand gesture and movement trajectory recognition model for non-touch HRI interface." In 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), pp. 505-508. IEEE, 2019.

[11] Chhatwal, Sarabjyot, Shivam Sharma, Sofia Singh, and Manish K. Ojha. "Handcrafted AI: Designing Virtual Hardware for Hand Gesture-Based Interaction." In 2025 International Conference on Pervasive Computational Technologies (ICPCT), pp. 311-315. IEEE, 2025.
[12] Negi, Shashank, Manan Garg, Hitesh Maindola, Vipashi Kansal, Upma Jain, and Shruti Bhatla. "Real-time human pose estimation: A mediapipe and python approach for 3d detection and classification." In 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), pp. 128-133. IEEE, 2023.
[13] Quinonez, Yadira, Carmen Lizarraga, and Raquel Aguayo. "Machine learning solutions with MediaPipe." In 2022 11th International Conference On Software Process Improvement (CIMPS), pp. 212-215. IEEE, 2022.
[14] Ansari, Mohd Aquib, and Dushyant Kumar Singh. "An approach for human machine interaction using dynamic hand gesture recognition." In 2019 IEEE Conference on Information and Communication Technology, pp. 1-6. IEEE, 2019.
[15] Noparlik, Robert, and Rafał Zdunek. "Hand gesture recognition system with finite state machine for remote desktop control." In 2023 8th International Conference on Smart and Sustainable Technologies (SpliTech), pp. 1-6. IEEE, 2023.
[16] Harris, Moh, and Ali Suryaperdana Agoes. "Applying hand gesture recognition for user guide application using MediaPipe." In 2nd International Seminar of Science and Applied Technology (ISSAT 2021), pp. 101-108. Atlantis Press, 2021.