Gesture Control Automation: A Webcam-Based Assistive System for Communication and Computer Control

[1]Snehal Nihate, [2]Sanket Umredkar, [3]Sejal Nandankar, [4] Revati Satpute, [5] Prof. Komal Tiwari, [6] Prof. Dr. Sushama Telrandhe

[1234] ,Department of Computer Science and Engineering,GNIET, RTMNU, Nagpur, India
[5] Assistant Professor, Department of Computer Science and Engineering,GNIET, RTMNU,Nagpur,India
[6] , Associate Professor, Department of Computer Science and Engineering,GNIET, RTMNU,Nagpur,India

Corresponding Author Email: snehalnihate2004@gmail.com

How to Cite this Article:

Nihate, S., Umredkar, S., Nandankar, S., Satpute, R. & Tiwari, K. (2026). Gesture Control Automation: A Webcam-Based Assistive System for Communication and Computer Control. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(8), 1-9. https://doi.org/10.55041/ijcope.v2i8.301

https://doi.org/10.55041/ijcope.v2i8.301

Abstract—

Gesture recognition is becoming a useful way of interacting with computers without using traditional devices such as keyboards, mice, and touchscreens. This type of interaction can be especially helpful for people who have physical or communication difficulties. In this paper, we propose **Gesture Control Automation**, a webcam-based system that recognizes custom hand gestures and uses them for communication and computer control. The system captures live video through a webcam and uses OpenCV for image processing and MediaPipe for hand landmark detection. The detected landmarks are used to identify predefined hand gestures.

The main idea of the proposed system is to use the recognized gesture for more than one purpose. A gesture can be converted into text, the text can be converted into speech, and selected gestures can also be used to perform predefined computer operations. To reduce accidental commands, the system can use confidence checking and multiple-frame confirmation before executing an automation action. The proposed approach is simple, low-cost, and does not require special wearable sensors. The system is mainly intended to improve accessibility and provide a common interface for communication and computer control. The work also identifies a gap in existing systems, where gesture recognition, communication, and automation are often developed separately. The proposed system attempts to combine these

Keywords— Gesture Recognition; Computer Vision; MediaPipe; OpenCV; Assistive Technology; Automation

I. INTRODUCTION

Computers have become an important part of our daily life. We use them for education, communication, work, entertainment, and many other activities. In most cases, computers are operated using a keyboard, mouse, touchscreen, or other physical devices. These methods are convenient, but they may not be equally easy for everyone. People with physical disabilities or communication difficulties can face problems while using such devices.

Hand gestures provide another way of interacting with a computer. A person can perform a specific hand gesture and the computer can identify it and perform the corresponding action. With the development of Computer Vision and Artificial Intelligence, it has become possible to recognize hand gestures using an ordinary camera. Earlier work in gesture recognition used image-processing techniques and manually designed features, while later research introduced machine learning and deep learning for better recognition [1]–[4].

One important development in this area is hand landmark detection. Instead of directly classifying the complete camera image, the positions of important points on the hand can be detected and used for gesture recognition. MediaPipe Hands is one such framework that provides real-time hand landmark detection from camera input [5].

Gesture recognition has also been explored for assistive communication. Different research works have attempted to convert hand or sign gestures into text or speech. At the same time, gesture-based systems have been developed for controlling home appliances, presentations, media players, and other computer functions.

Although there has been a lot of research in these areas, most systems mainly concentrate on one particular task. A communication system may focus on gesture-to-text or gesture-to-speech, while an automation system may focus only on controlling a device. This creates an opportunity to combine these functions into one application.

The proposed **Gesture Control Automation** system is designed with this idea. It uses a webcam to recognize predefined custom gestures and converts the recognized gesture into a common intent. The intent can then be used to display text, generate speech, or perform an automation command. The main goal is to provide a simple and accessible interaction method, particularly for users who have difficulty with conventional input methods.

The main objectives of this work are to develop a real-time gesture recognition system, create a custom gesture vocabulary, provide text and speech output, support computer automation, and reduce accidental actions during real-time use.

II. LITERATURE REVIEW

Gesture recognition has been studied for several years and has developed from simple image-processing methods into more advanced machine-learning and deep-learning systems.

Freeman and Roth presented an early vision-based approach for hand gesture recognition using orientation histograms [1]. Their work showed that hand shape and orientation could be used to recognize different gestures from visual input. Pavlovic, Sharma, and Huang later studied visual interpretation of hand gestures for Human-Computer Interaction and discussed different approaches for using gestures as an interface between people and computers.

For dynamic gestures, the movement of the hand over time is also important. Starner and Pentland used Hidden Markov Models for real-time American Sign Language recognition and showed that temporal information can help in understanding gesture sequences . Later, Molchanov et al. applied 3D Convolutional Neural Networks for dynamic hand gesture recognition . Deep-learning methods improved the ability to learn complex patterns, although these methods generally require larger datasets and greater computational resources.

Another important development has been real-time hand landmark detection. MediaPipe Hands provides a practical way of finding important hand points such as fingertips and finger joints using camera input . This approach is useful for systems that need real-time performance and low hardware requirements.

Recent work has also explored gesture recognition for assistive communication. Researchers have developed systems that recognize gestures and convert them into text or speech. Other studies have used hand gestures to operate home appliances and computer applications. These studies show that gesture recognition can be used for both communication and automation.

However, the reviewed work also shows some common problems. Recognition performance can change with lighting, background, hand orientation, camera distance, and differences between users. Systems trained on one user may also perform differently when another person uses them. Reviews of hand gesture recognition have identified dataset size, continuous gesture recognition, segmentation, and generalization as important challenges .

The main research gap identified from the literature is not the absence of gesture recognition itself. Gesture recognition is already a well-developed field. The gap is in combining the available technologies into one simple system. Communication and automation are often treated separately. In our proposed work, a recognized gesture is converted into a common intent and can then be sent to text, speech, or automation output.

This approach is intended to make the system more useful without requiring a separate recognition system for every function.

III. METHODOLOGY

The proposed system follows a simple sequence from camera input to the final output. The main stages are data collection, image processing, hand landmark detection, gesture recognition, intent mapping, and output generation.

A. Data Collection

Since the project uses custom gestures, a project-specific dataset is required. A number of gestures are selected according to the communication and automation requirements of the system.

Possible gestures include -
HELP, YES, NO, THANK YOU, STOP, NEXT, PREVIOUS, PLAY/PAUSE, VOLUME UP, and VOLUME DOWN.

Multiple samples of each gesture are collected using a webcam. The dataset should include small changes in hand position, distance, orientation, and lighting. Normal hand movements without an intentional gesture are also included as a no-gesture class.This helps the system distinguish between an actual command and ordinary hand movement.

B. Image Acquisition

The webcam captures live video from the user. OpenCV is used to access the camera and process individual video frames.

The frames are continuously passed to the hand-detection stage. Since the system is intended to work in real time, unnecessary processing should be avoided.

C. Hand Landmark Detection

MediaPipe Hands is used to detect the hand and identify its landmarks [5]. These landmarks represent important points on the hand, including fingertips and finger joints.

The landmark coordinates provide information about the hand structure and make it possible to recognize different hand positions.

D. Feature Extraction

The detected landmarks are converted into useful features.

The features can include:

- Relative position of fingertips
- Distance between hand landmarks
- Number of extended fingers
- Finger direction
- Hand orientation
- Movement between consecutive frames

The coordinates are normalized so that the system is less dependent on the exact location of the hand in the camera frame.

E. Gesture Recognition

The extracted features are used to identify the gesture.

For a small gesture vocabulary, rule-based recognition can be used. For a larger vocabulary, machine-learning models such as Random Forest or Support Vector Machine can be trained using the extracted landmark features.

The recognition stage should provide both a gesture label and an estimated confidence value.

F. Gesture Intent Mapping

The proposed system introduces a simple intermediate stage called the **Gesture Intent Layer**.I
nstead of directly attaching each gesture to one output, the recognized gesture is first mapped to an intent.
For example:

Open Palm → HELP
Thumb Up → YES
Closed Fist → STOP
Two-Finger Gesture → NEXT

The same intent can then be used by different parts of the application.

G. Text Output

For communication-related gestures, the recognized intent is displayed as text.

For example:

Gesture → HELP
Text Output → "I need help."

This gives the user a visual method for communicating a message.

H. Speech Output

The generated text can be converted into speech using a Text-to-Speech module.

For example:
Gesture → HELP → Text → "I need help." → Voice

This can help users communicate a message without speaking directly.

I. Automation Control

Some gestures are assigned to computer operations.
For example:

NEXT → Next slide
PLAY/PAUSE → Play or pause media
VOLUME UP → Increase volume

The commands can be modified according to the user's needs.

J. Confidence and Temporal Checking

A problem in real-time recognition is that one incorrect frame can sometimes result in a wrong gesture prediction. If this prediction is directly connected to automation, an unwanted action may occur.

To reduce this problem, the proposed system can use a confidence threshold and multiple-frame confirmation.

For example, the gesture should remain stable for a few consecutive frames before an automation command is executed.This makes the system less sensitive to temporary incorrect detections.

IV. RESULTS AND DISCUSSION

The proposed system is currently intended to be evaluated after implementation using the custom gesture dataset. Since final testing determines the actual performance, numerical values should only be added after experiments are completed.

The first measurement will be **gesture recognition accuracy**. This will show how many test gestures are correctly identified by the system.

Precision, recall, and F1-score can also be calculated to understand the quality of individual gesture classes. These measurements are useful when some gestures are easier to recognize than others.Another important measurement is **response time**.
Since the system is designed for real-time use, the delay between performing a gesture and receiving the output should be kept low.

For the automation part, the **false-trigger rate** is especially important. A system can have good recognition accuracy but still be inconvenient if it performs commands when the user has not intended to do so. This is why confidence and temporal checking are included in the proposed design.
The complete system should also be evaluated from the beginning to the end of the process.

For communication:
Gesture → Recognition → Text → Speech

For automation:

Gesture → Recognition → Intent → Command

The system should be tested under different conditions such as changes in lighting, background, hand position, and distance from the camera. It should also be tested with different users whenever possible.

Table I: Proposed Performance Evaluation

Parameter	Measurement
Recognition Accuracy	To be measured
Precision	To be measured
Recall	To be measured
F1-Score	To be measured

Parameter	Measurement
Response Time	To be measured
False Trigger Rate	To be measured
Communication Success Rate	To be measured
Automation Success Rate	To be measured

The expected result is a system that can recognize the selected custom gestures in real time and correctly perform the associated communication or automation function.

An important point from our study is that the quality of the complete interaction matters more than recognition accuracy alone. For example, if the gesture is correctly recognized but the speech output is delayed or the wrong automation command is executed, the system will still not be useful. Therefore, evaluation should focus on the complete user task.

V. CONCLUSION

Gesture recognition has already been widely studied using image processing, machine learning, deep learning, and hand landmark detection. The existing research shows that gestures can be used successfully for communication and computer control.

The proposed **Gesture Control Automation** system builds on these existing technologies and combines them in one application. A webcam is used to capture the user's hand, OpenCV handles video processing, and MediaPipe provides hand landmarks. The recognized gesture is then converted into an intent that can generate text, speech, or a computer command.

The main contribution of the proposed work is therefore the **integration of different functions rather than creating a completely new gesture-recognition algorithm**. This makes the project different from a basic gesture-control system. The use of a common intent layer also makes it easier to add new gestures and new output functions later.

The system is mainly designed to provide a simple and accessible interaction method for people who may find traditional computer controls difficult. Its low-cost webcam-based design also makes it easier to develop and use.

Future work can include a larger dataset, more users, dynamic gestures, multilingual speech, smart-home integration, and testing with differently-abled users. These improvements can help determine how well the system performs in practical situations.

ACKNOWLEDGMENT

We would like to thank the Department of Computer Science and Engineering, Guru Nanak Institute of Engineering and Technology, Nagpur, for providing the necessary support and facilities for this project.
We are also thankful to our guide, faculty members, and everyone who provided suggestions and support during the development of the proposed system.

REFERENCES

1] W. T. Freeman and M. Roth, “Orientation Histograms for Hand Gesture Recognition,” in *Proc. IEEE Int. Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, 1995.

[2] V. I. Pavlovic, R. Sharma, and T. S. Huang, “Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 677–695, 1997.

[3] T. Starner and A. Pentland, "Real-Time American Sign Language Recognition from Video Using Hidden Markov Models," in *Proc. IEEE Int. Symp. Computer Vision*, 1995, pp. 265–270.

[4] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand Gesture Recognition with 3D Convolutional Neural Networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, 2015.

[5] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "MediaPipe Hands: On-Device Real-Time Hand Tracking," arXiv:2006.10214, 2020.

[6] M. Oudah, A. Al-Naji, and J. Chahl, "Hand Gesture Recognition Based on Computer Vision: A Review of Techniques," *Journal of Imaging*, vol. 6, no. 8, p. 73, 2020.

[7] T. Vuletic, A. Duffy, L. Hay, C. McTeague, G. Campbell, and M. Grealy, "Systematic Literature Review of Hand Gestures Used in Human Computer Interaction Interfaces," *International Journal of Human-Computer Studies*, 2019.

[8] J. Qi, L. Ma, Z. Cui, *et al.*, "Computer Vision-Based Hand Gesture Recognition for Human-Robot Interaction: A Review," *Complex & Intelligent Systems*, 2024.

[9] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed. Cham, Switzerland: Springer, 2022.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.