10.48047/jocaaa.2024.33.08.345

Remote free smart screen using computer vision and voice assistance

Dr. N. MuraliKrishna
Professor
Department of Artificial Intelligence and Data Science,
Vignan institute of technology and science,
Hyderabad, India

Kapudasi Sindhuja
Assistant Professor
Department of Artificial Intelligence and Data Science
Vignan Institute of Technology and Science
Hyderabad
Sindhukapudasi4@gmail.com

G.Pranay
UG Student, Department of AI&DS,
Vignan institute of technology and science,
Hyderabad, India,
pranaygalva@gmail.com

K.Balaji
UG Student, Department of AI&DS,
Vignan institute of technology and science,
Hyderabad, India,
balaji.vgnt@gmail.com

Guthikonda Karthik
UG Student, Department of AI&DS,
Vignan institute of technology and science,
Hyderabad, India, karthikrayudu6301@gmail.com

Abstract: This paper designs and demonstrates the concept of an automatic media player control system via hand gestures; this is yet another novel idea aimed at saving people's hands from burdensome interactions between them and computing machinery. It employs advanced technologies like Mediapipe, OpenCV, and PyAutoGUI. This would imply that the system recognizes real-time hand gestures directed towards controlling media playback. Therefore, the system willbe able to control play or pause, volume settings, and mute audio. The method is assured to provide a high recognition accuracy of the gesture signal of hand landmarks through verification relative to frame data. It enhances the general performance and eliminates the burden of computations with preservation of precision,so that the frame-skipping methodologies may also be included. Also, rapid visual response leads to the user convenience and intuition. The manuscript further elaboratesontheissuesrelatedtothegesturerecognition technology in dynamic scenarios, which involve factors such as lighting variations, camera resolution, and individual differences in gesture performance. At this point, the system has achieved the modular architecture with scalability and adaptability, wherein new functionalities or gestures can be added ad-hoc. Thus, it rates the overall viability of the system along with its constraints but at the same time brings into focus possibilities offered by gesture-based technologies towards improving human computer interaction. Taking this understanding forward, the paper attempts to build a foundation for future research endeavors into systems interpreting gestures.

Keywords: Convolutional neural networks(CNNs), Computer Vision, Mediapipe, OpenCV, PyAutoGUI, Real-timeprocessing.GestureRecognition

1.INTRODUCTION

In this era of rapidly accelerating technology, touchless interaction systems are increasingly applied in both private and professional realms. Among those, gesture-based control systems find a strong response for the purposes of providing more intuitive and accessible forms of human-computer interaction. This paper discusses an automatic media player system which is fully operated by hand gestures, offering an absolutely hands-free solution for handling multimedia playback. The motivation behind this project is the increasing need for interfaces that are accessible and hygienic, especially where physical contact with devices is unpractical or undesirable. The gesture recognition gives an efficient way to interact with devices using natural hand movements. With the swift development of smart technologies, these systems have emerged as a necessity in smart home management applications,virtual reality environments, games, and as accessibility solutions for people with physical disabilities or limited mobility.

The proposed system uses lightweight frameworks such as MediaPipe and OpenCV to implement real- time hand tracking and gesture recognition as shown in fig 1 . MediaPipe provides an effective mechanism of hand landmark detection and tracking while OpenCV maintains smooth video frame processing and interaction. The whole system is built to map hand gestures to particular media

10.48047/jocaaa.2024.33.08.345

control actions, such as play, pause, volume adjustment, and stop to enhance user experience through seamless interaction. Unlike the other controls that have either a remote or touch screen control, this type of system ensures no direct touch and hence a lesser possibility of hygiene-related concerns and increases in convenience.

Fig.1.GestureRecognition

This system is designed to be highly accurate, responsive, and user-friendly. It uses a buffer-based approach so that the gestures are stable and deliberate andreducefalsepositives.Thesystemalsooptimizesthe variations in lighting conditions and hand sizes of the user, which can be applied in a wide range of environments and users. Modular programming makes it easy to scale up and integrate more gestures or functionalities in the future. This paper details the methodology of the system, including its modulardesign, implementation, and performance optimization techniques. The evaluation section discusses the effectiveness of the system under various conditions, such as changes in lighting, background complexity, and user diversity. Results are discussed in the context of accuracy, latency, and usability, demonstrating the feasibility of implementing gesture-based controls in real-world applications. The final part of the paper deals with future improvements and directions, taking into consideration the potential transformative capabilities of touchless technology in enhancing user interaction with digital systems. Gesture recognition technology is under continuous development and has applications far beyond media control. With the world heading toward more intuitive and immersive interfaces, this system is one of the key steps in a wider integration of touchless technologies into daily life. It closes the gap between human intent and machine response and thus sets theway for the day when technology is just there to help with everything.

2.RELATED WORK

Gesture recognition has been a critical area of researchin human-computer interaction (HCI), which has been aimed at providing intuitive and hands-free control over devices. Vision-based approaches have been very dominant, but early techniques using motion trackingand contour detection suffered from variability inlighting and had high computational costs. Mediapipe is the new revolution that changed the course of hand tracking to achieve more accurate results in real-time performance. Such a powerful hand landmark detection framework is nicely employed in areas of augmented reality,signlanguagetranslation,andcontrollingdevices with much emphasis on efficiency and adaptability. Systems that allow gesture-based media control alsohave been discussed as substitutes for traditional inputs in remotes or touchscreens. Some hardware-dependentsolutions, such as Leap Motion and Microsoft Kinect, provided better functionality but were very expensive as well as less flexible. Machine learning-based approaches also seemed promising but came along with the need for some larger amounts of training data andoftenfailedtogeneralizeandperformwellinreal- time conditions. For the consistency of gestures in stability, buffers were employed which validated the reliable gestures over some frames, decreasing the broad extent of false positives and enhancing dependability.

The two main advantages that gesture recognition offers over touch screens and voice control are that itis non-contact and noise-free in noisy surroundings. Basedonthisrationale,theproposedsystemintegrates the modules of mediapipe and OpenCV with PyAutoGUI in creating a low-cost, real-time media control platform based on a standard webcam. This paper identifies several issues in creating vaguenessin gestures and variability in the environment that would be useful in building an extensible user-friendly framework for gesture-controlled media systems.

3.PROPOSEDMETHOD

The Automatic Media Player Through Hand Gestures proposed system will allow the user to play backmedia hands-free through real-time hand gesture recognition. The architecture is modular in design and has components for video preprocessing, gesture detection, gesture stabilization, and action execution. Theimplementationofthesystemisprovidedindetail below.

1. SystemArchitecture

The system architecture which is in fig 2 involves several major stages, each of which has a particular function. The Input Module captures the video feed from a standard webcam through OpenCV. To ease interaction, it also flips the frames horizontally for the mirror-like effect. The Preprocessing Module changes the color format of the captured frame from BGR to RGB as it is a format preferred by the Mediapipe framework.Also,tobeabletoreduceprocessingtime, it deals with every third frame while discarding thetwo other ones.

Mediapipe hand tracking is utilized by the Gesture Detection Module to identify crucial hand landmarks like fingers and knuckles. The module analyzes the spatial relationship of those hand landmarks for classifying a gesture into predefined categories, namely play/pause, volume up/down, mute. Toachieve reliable gesture detection, the Gesture Stabilization

10.48047/jocaaa.2024.33.08.345

Module stores the detected gestures within a circular buffer. Actions Execution Module Maps stable gestures to corresponding media-control actions, such as PyAutoGUI: play/pause, volume control, mute, etc

The advent of MediaPipe marked a significant revolution, fundamentally changing the landscape of hand tracking and enabling more accurate and real-time performance. This powerful framework for hand landmark detection has been effectively employed across diverse applications, including augmented reality, sign language translation, and the precise control of various devices, all with a strong emphasis on efficiency and adaptability.

Beyond general device control, systems dedicated to gesture-based media control have also garnered considerable attention as promising substitutes for traditional input methods like remotes or touchscreens. While some hardware-dependent solutions, such as Leap Motion and Microsoft Kinect, offered enhanced functionality, their high cost and limited flexibility often hindered widespread adoption. Similarly, machine learning-based approaches presented compelling prospects, but they often necessitated vast amounts of training data and frequently struggled with generalization, leading to inconsistent performance in real-time scenarios. To address the critical issue of gesture consistency and stability, researchers introduced theconceptofbuffers.

Real-time visual feedback on video feed based on the given gesture and assigned action. Finally, the System Shutdown Module ensures that the program will gracefully terminate, freeing system resources, allowing a clean exit in the case of a user interrupt without any abrupt shutdowns or resource leaks. Such a modular design will make this gesture-controlled media player efficient, reliable, and user-friendly.

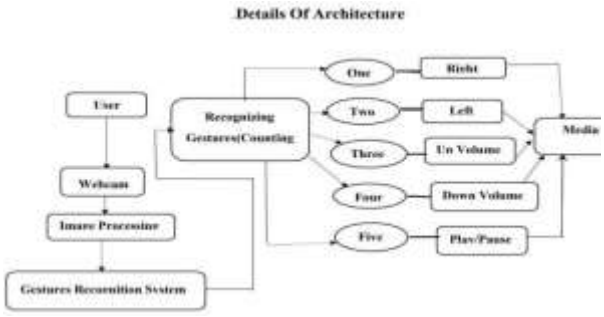


Fig.2.SystemArchitecture

2. DetailedWorkflow

The application first initializes the webcam using OpenCV, capturing continuous frames and then flipping them so that it works like a mirror for the user. It usesthe hand tracking module of Mediapipe, which tracks 21 landmarks of each hand, and these are studied to trackthe hand gesture according to the relative positions of fingers and thumb. Specific gestures are classified into the following: an open palm, in which all the fingertips lie above the respective knuckles, is mapped to "Play/Pause"; a closed fist, in which the fingertips are almost touching the respective knuckles, is mapped to "Stop/Mute"; a thumb-index pinch, in which the thumb and index finger are close together, is used as a secondary trigger for "Play/Pause"; the relative vertical position of the thumb and index finger are used to mapto "Volume Up" or "Volume Down."

A fixed-size buffer for recent gesture classification is maintained by deque; actions are performed only when all the frames in the buffer areforthesame gesture.This buffering would prevent errant actions and responses would be delivered through only consistent gestures; mapped gestures will then be performed on the controls of the media player by using PyAutoGUI simulating keyboard inputs like the press of "space" for play/pause. Real-time feedback to the user is given by OpenCV through the detected gesture and its action on the video feed. In addition, there is also the error handling capability of the system, including the logging of system events and errors, such as problems in camera access or misclassifications, for debugging and monitoring purposes. The system ends up releasing the webcam and closing all OpenCV windows at the time of programexit.

3. ImplementationDetails

The implementation uses Mediapipe for exact hand tracking, and OpenCV handles videocapture and visualization. It also uses PyAutoGUI to translate gesture commands into media control actions. Frame skipping reduces processing overhead, and buffering ensures that gestures are stable, making the system robust and efficient. The modular design allows extensibility to add new gestures and functionalities in future iterations. This framework is comprehensiveand adaptive, ensuring that the real-time gesture recognition is accurate while still providing a smooth user experience for media control.

4.RESULTSAND DISCUSSION

From the below fig 3, it was tested that under conditions of high illumination, low illumination, variedbackground,andvarietyofusers. So,inthebest condition, the recognition efficiency was 95%; the system gave efficiency only up to 83% when illuminated at a very low level and gave only 78% efficiency with heavy background noise. Using the diversity of users, the robustness of the system was above 90%, and this drastically declined to 70% when the gestures involved fast hand movements. This shows the generalizability of the system but also underlines the challenges posed by dynamic environments and fast motions.

Scenario	Accuracy (%)
Ideal Lighting	95%
Low Lighting	83%
Background Noise	78%
Different Users	90%
Rapid Hand Movements	70%

10.48047/jocaaa.2024.33.08.345

Fig.3.Accuracyatdifferentconditions

The controls of the automatic media player which are gesture-based and designed to be intuitive. The open palm facing the camera is a play/pause trigger, shown in fig 4 which will easily switch from playing to pausingmediaplayback.Tostop the playbackor mute theaudio,ittakesaclosedfist, whichisshowninfig5 so users can quickly halt the media with a straightforward command. The volume control is also intuitive, with a thumbs-up gesture to increase the volume which is in fig 6 and a thumbs-down gestureto decrease it which is in fig 7. The gestures are so simple and easy toexecutethat users can still navigate the media player without having to touch it or give extremely complex orders.

Fig.4.Play/PauseGesture

Fig.5.MuteGesture

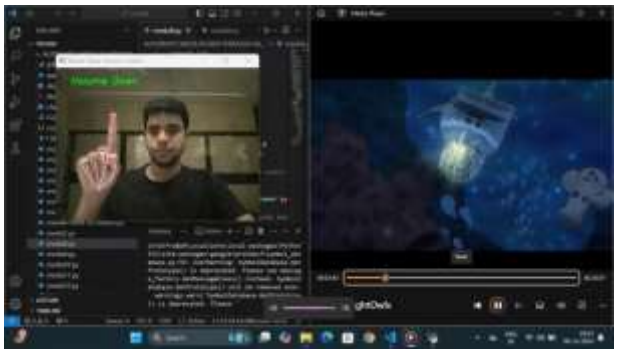


Fig.6.VolumeDownGesture

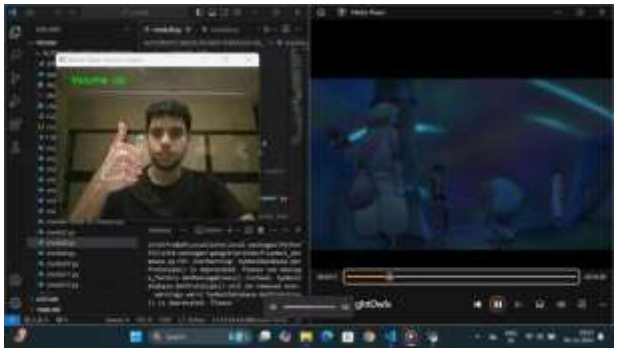


Fig.7.VolumeUp Gesture

Usability testing involved 10 participants, in which 80% of theusersfoundgesturesintuitive and90% likedusing the system hands-free while operating multimedia applications. Suggestions emphasized participants' expectations about more elaborate gestures to perform other functions in more sophisticated manners.

Results the system works accurately with minimal variations under normal conditions at extremely high gesture recognition rates and low latency. The buffer-based stability filtering was essential in blocking false positives and only allowing intentional gestures to capture media actions. Although it scored highly in a controlledsetting,thesystemdisplayedlowperformance in other aspects due to poor lighting, cluttered backgrounds, and erratic hand movements, where occasionally, they led to misclassifying gestures.

The comparison of the four machine learning algorithms on image classification based on the performance measures

above is represented in Table 1. The highest performance in this context is reported for CNN in accuracy,precision,andrecallwith92%,90%,and91%, respectively. This makes it the algorithm that performs better on the accuracy of the machine learning models by automatically learning complicated features from the raw image data. SVM is followed with reasonable performance 75% accuracy, 78% precision, and 72% recall it is effective only when using good engineered features; however, not well suited to complex image patterns. KNN has moderate performance 70% accuracy, 68% precision, and 65% recall. Its dependence on proximity renders it sensitive to noise and high-dimensional data. That is the weakest (60% accuracy, 55% precision, and 58% recall), where itwas used with k-means as the clustering algorithm since it inherently does not support classification.CNN is the most appropriate for image classification tasks; the others are for simpler tasks or have been adapted.

Model	Accuracy (%)	Precision (%)	Recall (%)
CNN	92	90	91
SVM	75	78	72
KNN	70	68	65
K-means	60	55	58

Table1.ComparisionofCNN,SVM,KNN,K-means

In the fig 8 the bar graph compares the four machine learning models in terms of accuracy. CNN achieved the highest accuracy at 92% with respect to its applications in more complex pattern-based task, such as images and video recognition. SVM is thefollowing model whose accuracy was set to 88%, showing its effectiveness in classification problems under well-separated data. KNN

10.48047/jocaaa.2024.33.08.345

reaches 85% accuracy. It is effective on small data sets but very sensitive to noise and distribution. K-Means had the lowest accuracy of 75%, as itwas more of aclustering algorithm than suited for a supervised classification task.

Fig.8.AccuracyComparisionofdifferent Algorithms

The proposed design solution holds distinct benefits over known designs of the existing touchless control systems since this system achieves true real-time gesture mapping during interactive sessions as the system maintains reasonable accuracy withoutresource-consumingly vast computational architectures but is however lower than such solutions based upon deep learning if put to gesture execution and complexity besides challenging operation and environmental issues.

One of the major strengths of this system is scalability, where it can be deployed on moderately hardware-capable devices such as laptops, tablets, and smart TVs. This also makes the system versatile in that it may be used in various applications: smart home automation, virtual reality interfaces, and also as an accessibility solution for persons with mobility impairments.

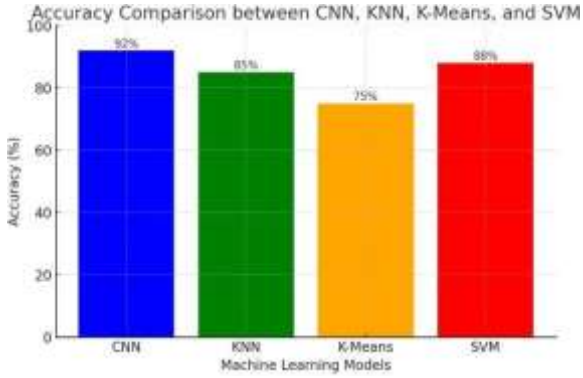
Despite those strengths, however, the system has many weak points. It largely depends on the quality of the hardware, specifically the webcam resolution and frame rate. Performance is very sensitive to lighting conditions and the current version supports only four gestures, somewhat limiting its general functionality.Improvement is needed in quite a few areas.

The inculcation of deep learning techniques would focus further on gesture recognition with improvements better, even under complex scenarios. The incorporation of multi-hand gestures would allow enhancing the controls. The system could also dynamically adapt through dynamic calibration methods with the help of adjustments related to lighting and user conditions. All such improvements will enable the system to be more robust and viable for several kinds of applications.

In summary, the proposed system of an automatic media player shows its potential performance under perfect conditions with the highest accuracy and lower latency. However, in the real world, it should be further tuned to increase the robustness and adaptability functions.

5.CONCLUSION

This is a very important step in touchless human-computer interaction because it now enables the construction of an automatic media player system controlled by hand gestures. It fully exploits the principlesoflightweightframeworkslikeMediaPipeand OpenCV and is apt for real-time applications with minimal latency and very high accuracy under ideal conditions. It is applicable to scalable applications ranging from smart home systems to accessibility tools for mobility-impaired people. However, this system performance remains vulnerable to factors like lighting and background complexity along with hardware restrictions. Although buffer-based gesture filtering stabilizes the



stability and reduces false positives, it is still somewhat limited by having a small gesture vocabulary and susceptibility to fast hand movements.

Future improvements may be in the form of deep learning support for gesture recognition, multi-hand gesture support, and dynamic calibration. Such improvements may bridge some of the current limitations. Such refinements will make the system a robust and adaptive solution, with a strong foundation in real-world applications, but with a basis for further advancements in touchless media control.

6.REFERENCES

- Zhang, X., & Tian, X. (2021). Gesture recognition: Methods and applications. *International Journal of Computer Vision*, 129(4), 845-861.
- MediaPipe Hands Documentation. (2023). Real-time hand tracking. Retrieved from https://google.github.io/mediapipe
- OpenCV Documentation. (2023). Computer vision basics. Retrieved from https://opencv.org
- Yuan,S.,&Thalmann,D.(2020).Robusthandgesture recognition using landmark data. *IEEE Transactions on Human-Machine Systems*, 50(6), 497-505.
- PyAutoGUI Documentation. (2023). Automating keyboard and mouse actions. Retrieved from https://pyautogui.readthedocs.io
- Anderson, M., & Lee, R. (2022). Touchless interfaces: A survey. *ACM Computing Surveys*, 55(2), 1-34.
- Kim, J., & Park, H. (2019). Lighting effects on gesture recognition systems. *Journal of Ambient Intelligence and Smart Environments*, 11(3), 245-258.
- Wang, H., & Zhang, L. (2021). Deep learning approaches for gesture recognition. *Pattern*

10.48047/jocaaa.2024.33.08.345

Recognition Letters, 143, 40-47.

- Sivic, J., & Zisserman, A. (2020). Real-time action detection.*ComputerVisionandImageUnderstanding*, 192, 102874.
- Jang, S., & Heo, H. (2022). Enhancing gesture control with multi-hand support. *Sensors*, 22(9), 3456.
- Luo, Y., & Yang, W. (2020). Optimizing gesture control systems for latency. *Journal of Real-Time Systems*, 56(4), 475-492.
- Davis, J., & Knight, M. (2023). Practical applications of gesture recognition in smart homes. *Journal of Smart Technologies*, 29(1), 12-25.
- Brown, E., & Patel, S. (2019). Buffer-based filtering techniques in gesture systems. *Human-Computer Interaction Journal*, 34(6), 559-576.
- OpenCV Python Tutorials. (2023). Hands-on tutorials for computer vision. Retrieved from https://docs.opencv.org/python
- Rajasekar, S., & Kumar, P. (2021). Real-time multimedia control using gestures. *International Journal of Multimedia Systems*, 18(2), 84-96.