Gesture Recognition for Media Interaction: A Streamlit Implementation with OpenCV and MediaPipe

Research Proposal *in* International Journal for Research in Applied Science and Engineering Technology · September 2023

DOI: 10.22214/ijraset.2023.55775

4 authors, including:

Vaibhav Patil
Dr. Babasaheb Ambedkar Technological University
2 PUBLICATIONS **2** CITATIONS

SEE PROFILE

iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IX **Month of publication:** September 2023

DOI: https://doi.org/10.22214/ijraset.2023.55775

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IX Sep 2023- Available at www.ijraset.com

Gesture Recognition for Media Interaction: A Streamlit Implementation with OpenCV and MediaPipe.

Vaibhav Patil[1], Dr. Sanjay Sutar[2], Sanskruti Ghadage[3], Shubham Palkar[4]
[1]*B-Tech Student, Department of Information Technology, Dr. Babasaheb Ambedkar Technological University, Maharashtra, India*
[2]*HOD, Department of Information Technology, Dr. Babasaheb Ambedkar Technological University, Maharashtra, India*
[3, 4]*B-Tech Student, Department of Information Technology, Dr. Babasaheb Ambedkar Technological University, Maharashtra, India*

Abstract: ***This project aims to create a media player application that responds to hand gestures, using Python and the OpenCV library. The system taps into computer vision methods, like those used in depth-sensing cameras such as Kinect or Intel RealSense, to track and understand hand movements. It processes the depth data to extract hand features and employs machine learning (like CNNs or decision trees) to classify these into gestures. This lets the application accurately interpret user gestures and apply them to media commands—play, pause, volume, and more. All this works through a user-friendly interface that even lets users customize gestures for specific commands. The combo of OpenCV and Python enables an efficient and adaptable media control system. This fusion of computer vision and machine learning offers a seamless, natural way to navigate media playback, making it an immersive experience without needing physical controllers.***
Keywords: Python; OpenCV; Hand Gestures; Machine Learning; MediaPipe.

I. INTRODUCTION

The field of human-computer interaction (HCI) has witnessed significant advancements in recent years, particularly in the realm of natural user interfaces. One area of interest is the development of gesture-based control systems, which enable users to interact with digital devices using hand gestures instead of traditional input methods such as keyboards or mouse devices. These systems leverage computer vision and machine learning techniques to interpret and respond to hand movements accurately. Gesture-based interaction presents a more intuitive and engaging approach to media control compared to conventional methods like keyboards or remotes. This method not only mimics real-world actions but also holds the potential to enhance accessibility for individuals with physical limitations, granting them an inclusive and independent media control experience. Moreover, gesture-based interfaces bring novelty and captivation to applications, setting them apart in a competitive landscape. The hands-free nature of gesture control is particularly advantageous in scenarios where users have dirty or occupied hands. With technological advancements in computer vision, machine learning, and sensors, and gesture recognition systems have become more accurate and cost-effective, making them a practical choice for developing effective media controllers based on hand gestures. The research plan encompasses several crucial phases: Firstly, the development of a robust real-time gesture recognition system, employing diverse computer vision techniques for precise hand gesture interpretation. Subsequently, the integration of this system into a media player application facilitates media playback control, including play, pause, volume adjustment, and track navigation. The study entails an extensive evaluation of system performance, focusing on accuracy, responsiveness, and resilience across various conditions and scenarios. Furthermore, user experience assessment is pivotal, involving surveys and studies to gauge user satisfaction, ease of use, and intuitive interaction when utilizing hand gestures for media control. Valuable user feedback will inform refinements to enhance the overall interaction experience.

II. LITERATURE REVIEW

1) *Controlling Media Player with Hand Gestures using Convolutional Neural Network Stella Nadar, Simran Nazareth, Kevin Paulson, Nilambri Narkar (2021, IEEE)*

Improvement in technology, response time, and ease of operations are the concerns. Here is where human-computer interaction comes into play. This interaction is unrestricted and challenges the used devices such as the keyboard and mouse for input. Gesture recognition has been gaining much attention. Gestures are instinctive and are frequently used in day-to-day interactions. Therefore, communicating using gestures with computers creates a whole new standard of interaction. In this project, with the help of computer vision and deep learning techniques, user hand movements (gestures) are used in real time to control the media player.

In this project, seven gestures are defined to control the media players using hand gestures. The proposed web application enables the user to use their local device camera to identify their gesture and execute the control over the media player and similar applications (without any additional hardware). It increases efficiency and makes interaction effortless by letting the user control his/her laptop/desktop from a distance.

2) *Human-Computer Interface Using Hand Gesture Recognition Based on Neural Network H. Jalab, H. K. Omer (2015, IEEE)*

Gestures are one of the most vivid and dramatic ways of communication between humans and computers. Hence, there has been a growing interest in creating easy-to-use interfaces by directly utilizing the natural communication and management skills of humans. This paper uses a neural network to present a hand gesture interface for controlling a media player. The proposed algorithm recognizes a set of four specific hand gestures, namely: Play, Stop, Forward, and Reverse. Our algorithm is based on four phases, Image acquisition, Hand segmentation, Feature extraction, and Classification. A frame from the webcam camera is captured, and then skin detection is used to segment skin regions from background pixels. A new image is created containing the hand boundary. Hand shape feature extraction is used to describe the hand gesture. An artificial neural network has also been utilized as a gesture classifier. 120 gesture images have been used for training. The obtained average classification rate is 95\%. The proposed algorithm develops an alternative input device to control the media player and also offers different gesture commands, which can be useful in real-time applications. Comparisons with other hand gesture recognition systems have revealed that our system shows better performance in terms of accuracy. The automatic vision-based recognition of hand gestures for sign language and control of electronic devices, like digital TV, and play stations was considered a hot research topic recently. However, the general problems of these works arise due to many issues, such as the complex backgrounds, the skin color, and the nature of static and dynamic hand gestures.

3) *System Application Control Based on Hand Gesture Using Deep Learning V Niranjani, R Keerthana, B Mohana Priya, K Nekalya, A K Padmanabhan (2021, IEEE)*

The Human-Computer Interaction progresses toward interfaces that seem to be natural and intuitive to use rather than the customary usage of keyboard and mouse. A hand gesture recognition system is one of the crucial techniques to build user-friendly interfaces, because of its diversified application and the potential of interacting with machines proficiently. Hand gestures including the movement of hands, fingers, or arms are considerable for interaction. The proof levels of the hand gestures are perceived from the level of static gestures to the dynamic gestures or intricate foundation through which the communication of human feeling with computers succeed. The proposed solution is framed by the identification of hand gestures as it possesses the perk of being used effortlessly and does not require an intervening medium. The existing system for application access is inflexible and arduous for people with blindness and hand deformities regarding human-computer interaction. A deep convolutional neural network (DCNN) is put forward in this paper, to use hand gesture recognition and immediately classify them by preserving even the not-hand area without any detection or segmentation process. Hence the proposed objective is to use different hand gestures via an integrated webcam with the aid of deep learning concepts beneficial for the visually impaired and people with a hand disability. The two approaches are static hand gestures and dynamic hand gestures. The predetermined gesture is entirely recognized by the static hand gesture method. While on the contrary in the dynamic method of gesture recognition, the meaning of the gesture is unclogged via its movement. The static gesture is contrary to the dynamic gesture and is less practical, though it possesses the perk of being a method with fewer difficulties.

III. PROPOSED SYSTEM

The system's objective is to create a media player control application that enables users to manage media playback through hand gestures. By leveraging computer vision methods, the system identifies and understands these gestures, eliminating the necessity for conventional input devices like keyboards and mice.

A. System Components and Technologies Used

1) *Webcam:* A webcam is used to capture the live video feed of the user's hand gestures.
2) *Gesture Recognition Algorithm:* The system employs a gesture recognition algorithm to analyze the video frames and identify specific hand gestures.
3) *Media Player:* The media player is responsible for playing, pausing, stopping, and adjusting the volume of media content.
4) *Gesture Mapping:* The system maps recognized gestures to corresponding media player commands.
5) Streamlit: A web application framework used for creating interactive user interfaces.

6) MediaPipe: A cross-platform framework for building multimodal applied machine learning pipelines
7) *PyAutoGUI:* A Python module that enables programmatically controlling the mouse and keyboard.
8) *NumPy*

Project Flow:

A fundamental library for scientific computing with Python, used for numerical operations.

- The project begins by setting up the development environment and installing the required libraries.
- The Streamlit application is created with the necessary user interface elements, such as buttons and video displays.
- The Mediapipe library is used to access the webcam and perform hand gesture recognition.
- The hand landmarks are extracted using the Mediapipe library, and the relevant gestures are recognized based on the position of the fingers.
- Once a gesture is recognized, appropriate actions are triggered using PyAutoGUI to control media playback.
- The NumPy library is utilized for efficient numerical operations and data manipulation if required.
- The application is tested extensively to ensure proper functionality and usability.

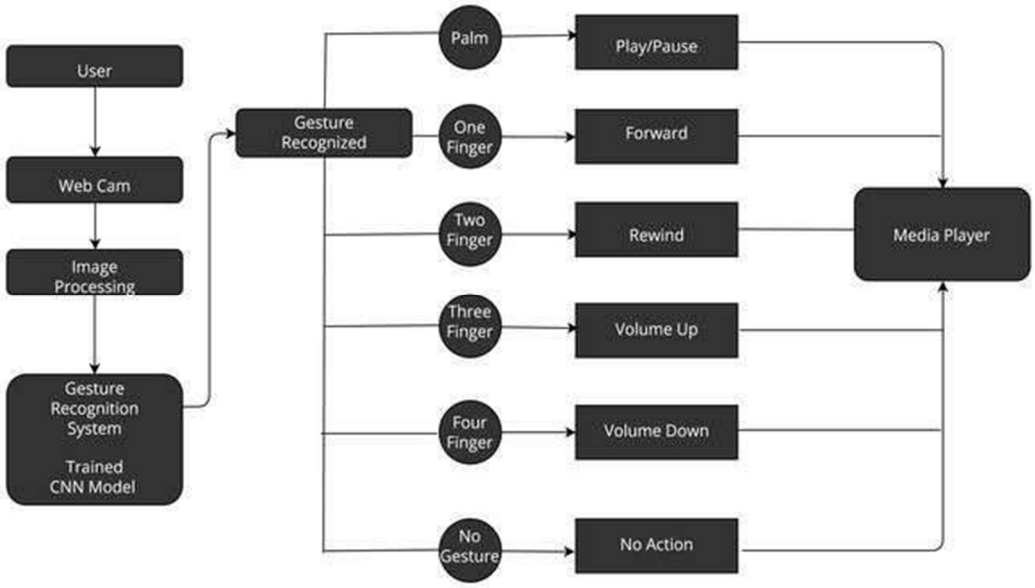


Figure 1Use- Case Diagram

IV. ALGORITHM

A. *Fingers Recognition*

1) *Co-ordinate axes for Computer Screen*

The diagram below shows the ax-wielding interface axes used when working with Computer Vision. Using these links, we can track the area of interest and arrange various activities according to the object's layout on the screen**.**

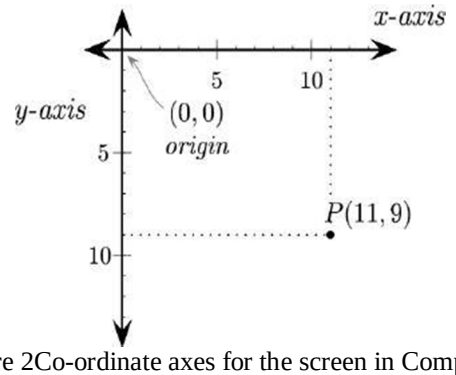


Figure 2Co-ordinate axes for the screen in Computer Vision

2) Hand Landmarks defined by MediaPipe

Using these hand landmarks, we can define various gestures and link them to their corresponding intended functionalities to control the media player. Coordinates on the screen and the landmarks on the hand are mapped together to generate the desired output.

a) In the segmentation image of fingers, the labeling algorithm is applied to mark the regions of the fingers. In the result of the labeling method, the detected regions in which the number of pixels is too small are regarded as noisy regions and discarded. Only the regions of enough sizes are regarded as fingers and remain.

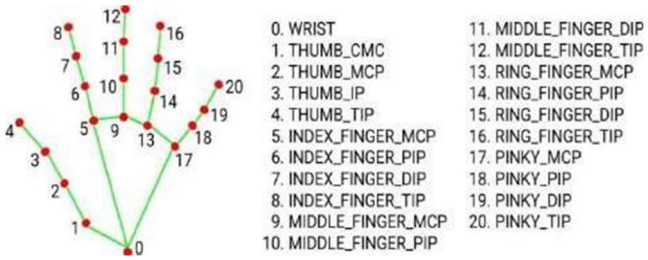


Figure 3Hand Landmarks

b) With the help of the palm mask, fingers, and the palm can be segmented easily. The part of the hand that is covered by the palm mask is the palm, while the other parts of the hand are the fingers. Based on the count the features of the media are controlled e.g.:

a) If the count of the finger is 1 the media will be forward.

b) If the count of the finger is 2 the media will be Rewind.

c) Similarly, we find the distance between the tip of the thumb and the index finger using a hypotenuse. We achieve this by calling the math hypo function and then passing the difference between x2 and x1 and the difference between y2 and y1. And based on the length between the index finger and thumb finger the volume is controlled dynamically. If the length between fingers is increased the volume increases and vice versa.

3) Contours

Figure 4Hand Contours with OpenCV

Contours are defined as the line joining all the points along the boundary of an image that has the same intensity. Contours come in handy in shape analysis, finding the size of the object of interest, and object detection.

4) Convexity Defects

Convexity defects are found from the convexity hull, which is the farthest point from the convex points, i.e., if the fingertips are convex points, then the trough between fingers is the convexity defect. and these defects are counted. The angle between the fingers is found using the cosine rule, so we understand the difference between index, middle, ring, little, and thumb fingers. The contour of the hand in the region of interest is the set of points that correspond to the extremities of the human hand, which in turn define the hand's boundaries. The contour is then analyzed for the gesture and also approximated into a polygon. The canny edge detection method opts to calculate the contour.

Figure 5Convexity Defects

V. METHODOLOGY

The project introduces a novel means of media player control via hand gestures, aligning with users' real-world interactions. This intuitive approach offers a seamless and interruption-free experience, eliminating the need for extra devices. Moreover, it expands interaction possibilities by allowing diverse forms of engagement, rather than confining users to a single input point.

The process begins with capturing an image, which is subsequently converted into RGB format. The code then proceeds to verify the presence of multiple hands within the image. An empty list serves as a repository for elements representing the detected hand's characteristics. These elements encompass the number of points comprising the hand, derived through the utilization of media pipe technology.

A. Play/Pause

The system has succeeded in getting a hand gesture to pause and detect the action to be performed, so the corresponding video play action is active.

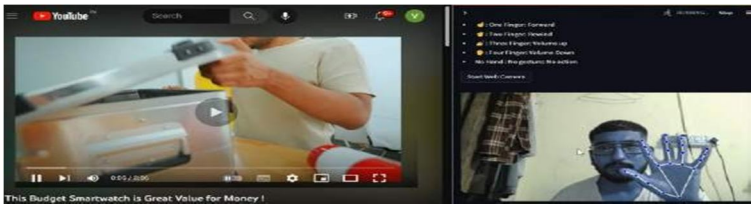


Figure 6Hand Gesture for play

B. Forward

The system has succeeded in detecting the hand gesture and detecting the action to be performed, so the corresponding video transfer action (forward seeking) is active.

C. Rewind

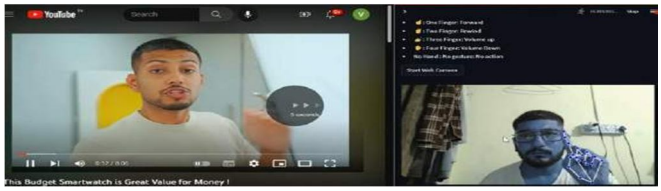


Figure 7Hand Gesture for forward

The system has succeeded in detecting the hand gesture and detecting the action to be performed, so the corresponding video transfer action (backward seeking) is active.

D. Volume up

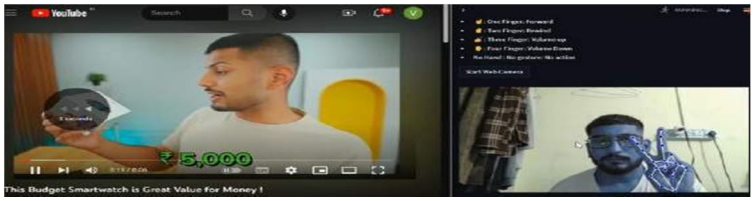


Figure 8Hand Gesture for rewind

The system has succeeded in detecting the Volume Up hand touch and detecting the action to be performed, so the corresponding action to increase the video volume is effective

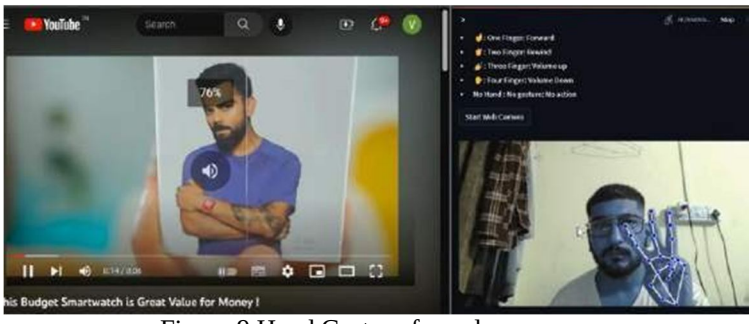


Figure 9 Hand Gesture for volume up

E. Volume Down

The system has succeeded in detecting the Volume Down hand touch and detecting the action to be performed, so the corresponding action to decrease the video volume is effective.

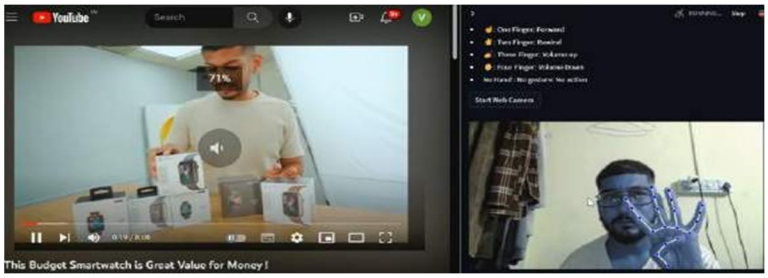


Figure 10Hand Gesture for volume down

VI. RESULTS

A. Hand Gesture Recognition

The system successfully recognized and classified a variety of hand gestures performed by users. These gestures included play, pause, stop, volume up, volume down, and skip/seek gestures.

B. Media Player Control

The system effectively translated the recognized hand gestures into corresponding commands for the media player. Users were able to control media playback operations, such as play, pause, stop, and adjust volume, by performing predefined gestures.

C. Real-time Interaction

The system achieved real-time performance, providing instantaneous feedback and responsiveness to the user's hand gestures. Media playback operations closely followed the recognized gestures without noticeable delays.

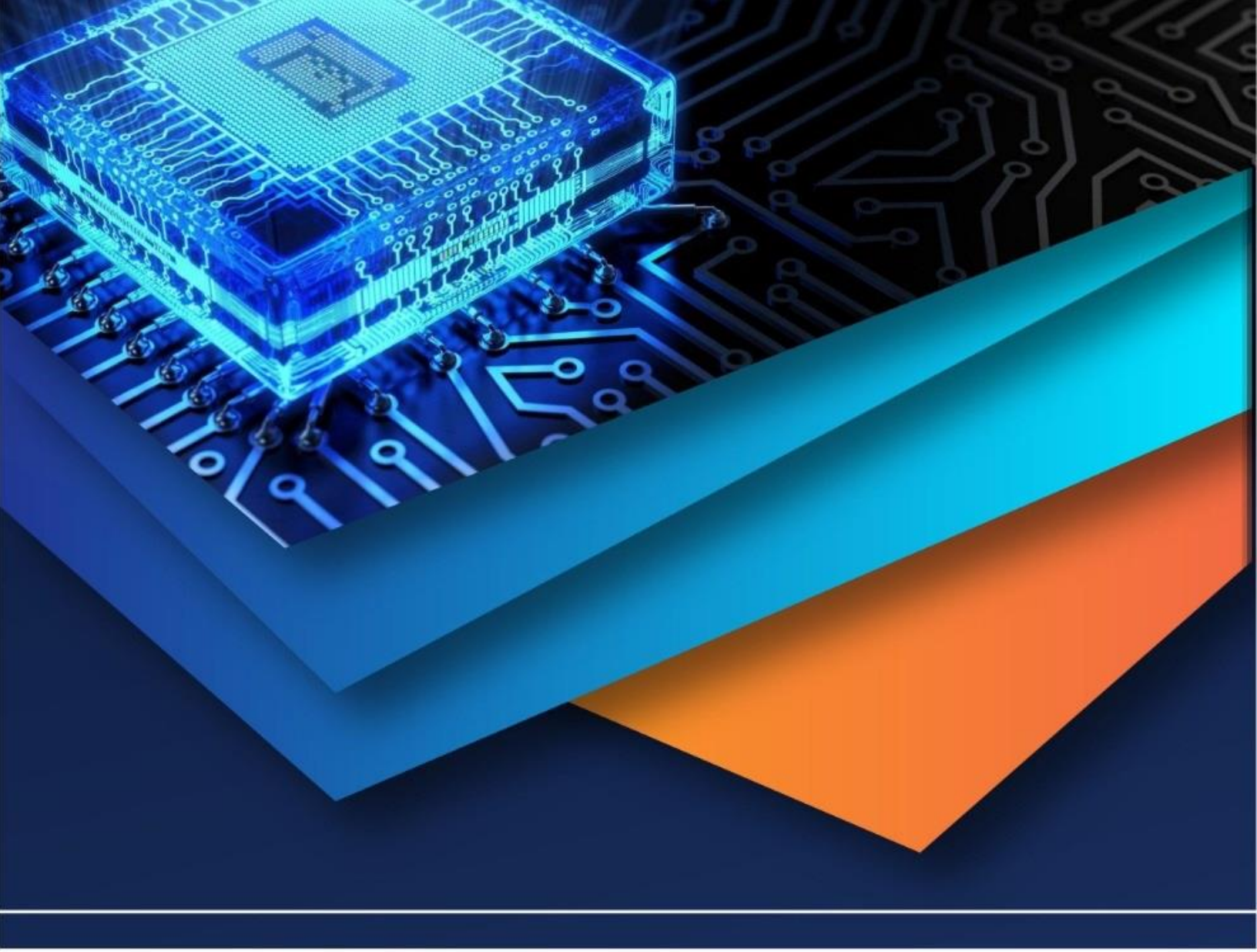
VII. CONCLUSION

In the current world, many resources are available to provide input to any application some require physical touch, and some without the use of physical touch (speech, hand touch, etc.), the user can manage the system remotely without using the keyboard and mouse. This application provides a novel human-computer interface where the user can control the media player (VLC) using hand gestures. The system-specific touch to control the VLC player functions. The user will provide a touch as inserted depending on the activity you are interested in. The app provides the flexibility to define a user's touch of interest with a specific command that makes the app more useful for people with physical disabilities, as they can define touch according to their ability. The system managed to detect the volume down of the Volume Down and detect the action to be performed, so the corresponding action to lower the video volume is active. The program has successfully detected the rewind touch and detected the action to be performed, so the corresponding video rewind action is active.

REFERENCES

[1] M. M. Kobylanski and A. Borylo, "Media Player Control Using Hand Gestures," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2016.
[2] S. Manogaran and K. S. Murugan, "Hand Gesture Recognition Techniques for Human-Computer Interaction," International Journal of Computer Applications, 2017.
[3] P. Lertkittiporn, "A Review of Hand Gesture Recognition Techniques," Proceedings of the International MultiConference of Engineers and Computer Scientists, 2018.
[4] Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing," Pearson, 2017.
[5] Simon Haykin and Michael Moher, "Introduction to Analog and Digital Communications," Wiley, 2017.
[6] Iain Matthews and Simon Baker, "Active Appearance Models Revisited," International Journal of Computer Vision, 2004.
[7] OpenCV Documentation: https://docs.opencv.org/

1. MediaPipe Documentation: https://mediapipe.dev/
2. TensorFlow Tutorials on Image Classification: https://www.tensorflow.org/tutorials/images/classification



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY