Controlling Media Player using Hand Gestures

*1 T Santosh, 2P. Ashwini, 3K. Nandini

*1 Associate Professor, Department of Information Technology, Bhoj Reddy Engineering College for Women, Hyderabad, India

2,3Department of Information Technology, Bhoj Reddy Engineering College for Women, Hyderabad, India

ARTICLE INFO

ABSTRACT

Gesture-based real-time gesture recognition systems received great attention in recent years because of their ability to interact with systems efficiently through human-computer interaction. Human-Computer Interaction can gain several advantages with the establishment of different natural forms of device-free communication. Gestures are a natural form of action that we often use in our daily lives to interact, so to use them as a way of communicating with computers generates a new paradigm of computing interaction. This project implements computer vision and gesture recognition techniques and develops a vision based low-cost input software for controlling the media player through gestures.

Keywords : Machine learning, OpenCV, Pycaw, Mediapipe, VLC Media Player, gesture, Human-Computer Interface, Webcam

I. INTRODUCTION

Nowadays, gesture recognition plays an important part in the interaction between humans and computers. To facilitate simple yet user-friendly communication between humans and computers hand Gestures can be used which enable us humans to interact with machines without having to use devices like keyboards, laser pens, etc. In the proposed system, users can use four simple gestures to control the Media Player without physically touching the PC. Gesture is a symbol of physical behavior or emotional expression. It includes body gesture and hand gesture. It falls into two categories: static gesture and dynamic gesture. For the former, the posture of the body or the gesture of the hand denotes a sign. For the latter, the movement of the body or the hand conveys some messages.

Gesture can be used as a tool of communication between computer and human. It is greatly different from the traditional hardware based methods and can accomplish human-computer interaction through gesture recognition. Gesture recognition determines the user intent through the recognition of the gesture or movement of the body or body parts. In the past decades, many researchers have strived to improve the hand gesture recognition technology. Hand gesture recognition has great value in many applications such as sign language recognition, augmented reality (virtual reality), sign language interpreters for the disabled, and robot control.

II. RELATED WORK

A. Title: Pixel-Based Hand Gesture Recognition with Kinect Depth Camera

Authors: Chong Wang

In 2015, Chong Wang, "Super Pixel-Based Hand Gesture Recognition with Kinect Depth Camera" proposed a system that uses the Kinect Deposit Camera. It is based on compact representation in the form of large pixels, which accurately capture shapes, textures, and deep touch features. As this program uses the Kinect camera for depth, system costs are higher.

B. Title: Handwriting The Recognition System

Authors: Swapnil D. Badgujar

In 2014, Swapnil D. Badgujar, "Handwriting The Recognition System "proposed a system that said see touching an unknown input by hand tracking and extraction method. This program is used in see one touch. There is a thought that is a fixed background so that the system is smaller search the tracking region. This program only controls the mouse finger using webcam.

C. Title: Motion MEMS Accelerator Based Non-Specific-User Hand Touch Recognition

Author: Ruize Xu, Shengli Zhou and Wen J.Li

In 2012, Ruize Xu, Shengli Zhou and Wen J. Li, "MEMS Accelerometer Based Non-Specific-User Hand Touch Recognition", was able to create a system that he could not identify various hand gestures such as up, down, right, and left, crossing and turning. Three different modules were developed that detect various hand gestures. MEMS (MicroElectromechanical System) Features 3- accelerometers axes are provided as inputs. Movement hand in three perpendicular direction was received by three accelerometers and sent to the system via Bluetooth. The segmentation algorithm was used and finally various hand gestures were recognized by the same touch is already saved in the system. People often prefer the internet to have a daily update weather, news etc. So, for this purpose they do keyboard and mouse functions. This program offers little accuracy in obtaining final touch points due the smallest size of the hand touch website.

D. Title: Robust The Vision based Hand Gestures Interface for Operating VLC Media Player

Authors: Anupam Agrawal , Siddharth Swarup Rautaray

In 2010, Anupam Agrawal and Siddharth Swarup Rautaray, "The Vision based Hand Gestures Interface for Operating VLC Media Player Application "program, in that the nearest K neighbor algorithm was used see various touches. Features of VLC media player which were driven by hand gestures including play, as well pause, Full screen, pause, increase volume, and decrease capacity. Lucas Kanade Pyramidical's Optical Flow The algorithm is used to detect hand input video. The algorithm mentioned above detects movement points in the image input. Then the methods of K find a hand centre. By using this facility, the hand is the same noticed. This program uses the database it contains various hand gestures and inputs compared with this image stored and appropriately VLC media player it was controlled. The current application is not very robust recognition phase.

E. Title: A-fast-sighted hand-based touch algorithm Recognition for Robot Control

Authors: Erol Ozgur, Asansarabi Malima

In 2006, it formed Erol Ozgur and Asansarabi Malima "A fast-sighted hand-based touch algorithm Recognition for Robot Control" which controlled the robot using hand gestures but with limited touch. First the division of the hand circuit was followed by pointing fingers and finally separating the gestures. The algorithm used is consistent in translation, rotation and hand scale. This program works on a robot control app with reliable performance.

III. PROPOSED SYSTEM

To run this project, we used a webcam to find user input. In transferring input such as images through the image processing process to process the image and to obtain information from the images. The model

provides basic types of VLC media player functions, that is, play, pause, volume up, volume down, mute, forward and backward the video. This is performed by hand gesture recognition. Hand detection is to play and pause the video. Hand gesture recognition is to perform other functions. We make use of different libraries and modules of Python, like OpenCV, Google Media pipe, Pyautogui, and sub process to build the application. The hand gesture recognition system on controlling media player works using real time gestures input from user using integrated webcam and provided gesture matches with a function to control the media player. The implementation of the project is as follows:

 First webcam will open and user provides gesture input through that.

 The VLC app and video to play must be opened with a sub process shell command.

 The hand landmarks should be done through Google media pipe.

 After reorganization of gesture then the video starts play- ing.

 It detects and detects hand gestures through the image processing process and corresponds to the keyboard keys provided by the Pyautogui library.
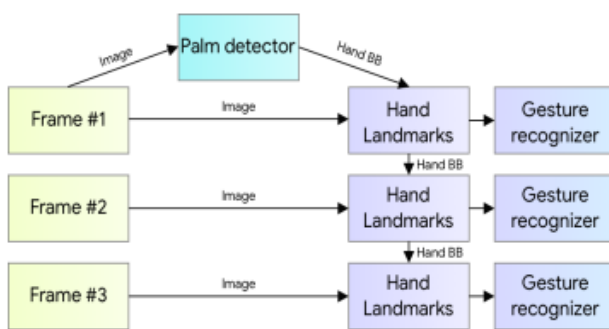
 Perform those operations

Fig 1: System Architecture

Following are the ways through which gesture controlled applications were created in the past:

A. Image Detection and Processing

Image Detection and Processing: OpenCV (Open Source Computer Vision), is one of the most widely used tools for computer vision and image detection and processing tasks. It is used in various applications such as face detection, video capture, motion detection, object disclosure, face mask detection, social distance, and much more. Steps For Image detection and processing which include capturing, resizing, converting, extracting, detecting then finally recognizing.

1. At first we capture the image in the form of frame from the video. img = cv2.imread('/image-path, cv2.IMREAD.COLOR)

2. Resize this image frame to 300 X 200.

3. Convert the BGR image into RGB image using the below code line: imgRGB = cv2.cvtColor(img, cv2.COLOR-BGR2RGB)

4. Detecting the boundary points.

5. Extract the Region of Intrest.

6. Recognize the Gesture

B. Gesture Recognition

Gesture Recognition: When the image frame is extracted, The Gesture in the image is recognised according to the calculations based on coordinates of the Media pipe Landmarks. Let us consider the figure 3 which shows the Hand Landmarks defined by MediaPipe over a palm. Here,the '0th' landmarks is at the wrist of the palm, Landmarks 1 to 4 are distributed all over the thumb where the '1st' landmark is at the base of the thumb and the '4th' landmark is at the tip of the thumb; Similarly landmarks 5 to 8 are distributed all over the index finger where '8th' landmark is at the tip and '5th' is at the base; Landmarks 9 to 12 are distributed all over the middle finger where the '12th' landmark is at the tip and '9th' is at the base; Landmarks 13 to 16 are distributed all over the ring finger where'13'is at the base and '16th' is at the tip, Landmarks 17 to 20 are distributed on the little finger where '17th' is at the base and '20th' is at the tip. Initially, the landmarks at the tip of the finger have ycoordinate greater than the y-coordinate of the landmark at the base of the finger (eg:The y-coordinate of the 12th landmark is greater than the ycoordinate of the 9th landmark). For the Pause and Play function spacebar is used and the Gesture assigned for that is a

closed wrist. Here In this case, when the wrist is closed the tips of the fingers get lowered than the base of the fingers, Thus the landmarks at the tips of the fingers gets lowered than the landmarks at the base of the fingers. So the new y-coordinates of the landmarks at the finger tips become smaller than the y-coordinates of the landmarks at the finger bases. Thus, the closed wrist is detected.

IV. RESULTS

The model is restricted to a plain background so that the model can detect the color of the skin based on the HSV values. After detecting the convexity defects, we will get to know the number of fingers shown by the user then we can bind the count of convexity defect with the key to control the media.

Video Capture:
A real time video is captured using OpenCV library and camera. The process involves selecting a rectangular portion of the captured image which is then filtered by a mask to extract the gray scaled binary image of the hand. For accurate tracking the user's hand the hue and saturation range should be selected appropriately according to the user. Tracking can be done by creating a filter of the hue and saturation range and applying it to the input image through bitwise-and operation.

Captured Image   *Masked Image*

Fig 2 : Output Images

V. CONCLUSION

In this project, we performed hand and gesture recognition in python using OpenCV. Using the recognized gestures, we controlled various UI elements in the screen, hence making it easier for users to interact with the system. Hand recognition is based on the lower and upper bound of color, resulting in a binary threshold image. Contours and convexity defects are calculated, based on which the number of fingers is found out, which becomes the gesture. This is then used to control UI elements. This implementation works well with a suitable background. In the future, we can improve upon many aspects of the project. Some of them are:

- Adaptive thresholding to properly segment hand and background
- Removing the dependency on background colour to identify the hand
- Adding more UI control with the recognized gestures specific to a particular application
- Improving the gesture recognition, possibly by using machine and deep learning

VI. REFERENCES

[1]. Oudah M, Al-Naji A, Chahl J. Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. J Imaging. 2020 Jul 23;6(8):73. doi: 10.3390/jimaging6080073. PMID: 34460688; PMCID: PMC8321080.

[2]. Karapinar Senturk, Z., & Bakay, M. S. (2021). Machine Learning Based Hand Gesture Recognition via EMG Data. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, 10(2). https://doi.org/10.14201/ADCAIJ202110212313 6

[3]. Bakheet, S., Al-Hamadi, A. Robust hand gesture recognition using multiple shape-oriented visual

cues. J Image Video Proc. 2021, 26 (2021). https://doi.org/10.1186/s13640-021-00567-1

[4]. Abhilash Dayanandan, Akshay Chakkungal, Anooj Kommeri, Deepak Koppuliparam bil, Dr. Prashant Nitnaware (May 2020). GestureControlled Media Player using TinyYoloV3. IRJET: International Research Journal of Engineering and Technology, 07(05). https://www.irjet.net/archives/V7/i5/IRJET-V7I5573.pdf

[5]. Raimundo F. Pinto, Carlos D. B. Borges, Antônio M. A. Almeida, IálisC. Paula, "Static Hand Gesture Recognition Based on Convolutional Neural Networks", Journal of Electrical and Computer Engineering, vol. 2019, Article ID 4167890, 12 pages, 2019. https://doi.org/10.1155/2019/4167890

[6]. Moin, A., Zhou, A., Rahimi, A., Menon, A., Benatti, S., Alexandrov, G., Tamakloe, S., Ting, J., Yamamoto, N., Khan, Y., Burghardt, F., Benini, L., Arias, A.C., & Rabaey, J.M. (2020). A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. Nature Electronics, 4, 54-63.

[7]. Rubin Bose, S. and Sathiesh Kumar, V. 'In-situ Identification and Recognition of Multi-hand Gestures Using Optimized Deep Residual Network'. 1 Jan. 2021 : 6983 – 6997.

[8]. Verdadero, M. S., Martinez-Ojeda, C. O., & Cruz, J. C. D. (2018). Hand Gesture Recognition System as an Alternative Interface for Remote Controlled Home Appliances. In 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM) (pp. 1-5). IEEE.

[9]. https://www.geeksforgeeks.org/find-and-draw-contours-using-opencvpython/

[10]. https://learnopencv.com/contour-detection-using-opencv-python-c/

[11]. https://learnopencv.com/convex-hull-using-opencv-in-python-and-c/

[12]. https://theailearner.com/2020/11/09/convexity-defects-opencv/

Cite this article as :