

# Software Controller using Hand Gestures

*Samruddhi Mumbare \**, Kunal Shivam, Priyanka Lokhande, Samruddhi Zaware, Varad Deshpande, Kuldeep Vayadande

Dept. of Information Technology, Vishwakarma Institute of Technology, Pune, Maharashtra, India

**Abstract.** New technologies emerge in response to the passage of time. Robotic hand gesture control is one of them. Collaboration frameworks based on gestures are becoming increasingly popular in the business and at home. The approach we suggested can greatly reduce the utilization of hardware components such as a keyboard and mouse. The goal of this research is to create a system that can recognize hand gestures and use them as an input command to connect with a computer or laptop. According to a recent study, the use of CNN technology is still lacking in Hand Gesture Recognition. Our research aims to leverage CNN technology to recognize gestures in both static and dynamic modes, and then deploy the trained model in real-time applications.

## 1. Introduction

In the real scenario of Human-Computer Interaction, gesture recognition is a hot topic. It may be used for virtual environment control, sign language translation, robot control, and music composition among other things[1]. Nowadays, most of the things are automated and physical contact between the system and work will also be ineffective. The concept of Action Vision through a variety of methods has been extensively researched. It works with the concepts of image processing like background subtraction and thresholding which helps in segmenting the object that is used via our hand for gesture recognition [2].

In paper [3], authors have implemented static and dynamic hand gesture recognition and demonstrated using a music player for static and a video game for dynamic gestures.

In static hand gesture recognition, we have used eight gestures that are static in nature and are used to trigger some of the operations in the music player made using python and pygame. The functionalities like playing the music, pausing the music, increasing volume, decreasing volume, stopping, playing music, etc. are handled by the hand gesture without actually doing it by a mouse or keyboard key.

The dynamic hand gesture system, which is the main contribution, demonstrates how to play a computer game with human motions. The goal of this research is to develop a way for a player to play a game without using a physical controller. The operations on the Game will be performed with the game's basic gaming controls based on the analysis performed by the program on identifying human Hand Gestures. A set of instructions is included in the program to identify human hand motions. The movements should be performed using the palms of the hands. The user interface module provides the user with all of the necessary

---

\* samruddhi.mumbare19@vit.edu

graphical user interfaces to register the user's arm positions for performing gestures. To recognize gestures, the gestures recognition module can be utilized. Finally, the analysis module will evaluate the human hand gesture and perform game controls depending on the estimated analysis. The technique is also beneficial to players who may be injured or have broken bones. Based on the analysis, the user may be able to do various functions with the same movements. This will allow players to have a greater engagement and impression of the game, making it more enjoyable to play. Hand gesture recognition systems that are totally reliable are currently being researched and developed. This system development can serve as a first step in laying a foundation that can be expanded in the future.

The later part of the paper follows the following sequence. Section 2 discusses the literature review; Section 3 covers the methodology and proposed system. In section 4 we are discussing the results and finally, section 5 covers the conclusion.

## **2. Literature Survey**

Gesture sensor systems are gaining momentum these days because of the way people and machines can easily interact. The goal of handwriting is to improve communication between people and computers for the purpose of transmitting the information.

Handwriting Recognition Sheet: A Review of Texts[1] provides a summary of current technology for detecting hand gestures, both vertical and dynamic. It shows all the methods for recognizing hand gestures that have been used in various research papers. Touch gestures are used in a variety of applications, including personal computer interactions, robots, sign language, digital and alphabetical values, and numbers. In the hand of touch detection and authors of [1] found that, compared to vision-based technologies and glove-based methods, the Kinect sensor is widely used. Compared to vertical hand gestures, the flexibility of hand gestures requires additional calculation. In various computer applications, hand gestures provide an enjoyable interactive environment.

The paper[2] explains how to build an electrical system that can recognize twelve manual gestures made by an interlocutor with one of their hands in real-time in a scenario with controlled lighting and background. The proposed system supports hand rotations, translations, and scale changes in the camera plane. Analog devices ADSP BF-533 Ez-Kit Lite evaluation card is required for the system. Displaying a letter connected with a recognized gesture as a concluding stage to develop the process is recommended. A visual depiction of the recommended algorithm, on the other hand, can be obtained in a personal computer's visual toolbox. Deaf and hard-of-hearing people will be able to speak with the general public thanks to recent technology that links them to computers.

Touch gestures are one of the most common ways of communicating with computers, and the precise real-time interpretation of moving hand gestures. The research [3] developed a system that uses animation history (MHI) and feedforward neural networks to identify movement in front of a web camera in real-time. To capture motion pictures in the image, the background of the framed frames is first removed using the Gaussian-based background front partition method, and then the sound removal to extract the random sound from the frame has been done. Then, using Otsu binarization, a binary threshold is used to determine the optimal threshold value. These processed frames are assembled, and a compiling moving picture is created using a design approach based on the similarity of the structure. This [3] method also calculates and uses the structural similarities between the collected image and the first frame. The author proposes a system of a real-time dynamic touch detection system based on computer perception and sensory networks in this study. The suggested system in [3] creates animated historical images based on the deviation of the structure from the

collected frame, which gives a sense of local and temporary movement. When a moving historical image deviates below a certain limit, it is categorized using a neural network to determine whether the action was detected or not.

The probability of each phase is determined using the neural network, and when the high probability class exceeds 0.8 chances, it is defined as the appropriate effect. Real-time experimental findings suggest that combining dynamic history images with neural networks produces more satisfying results.

CNN + RNN Depth and Scissors based on Power Hand Detection in ambient research[4], authors done monitoring based on human sense sensitivity and touch detection. This paper specifically focuses on the function of detecting a flexible hand, using both skeletal and depth data obtained from deep-RGB sensors, and powerful deep learning modes like CNN and RNN. Computer-Based Handwriting Recognition focuses on reviewing the literature on hand-touch techniques and introduces their relevance and limitations under a variety of contexts[4]. In addition, it has developed the functionality of these methods, focusing on computer-aided detection techniques that address similarity of points, method of hand distribution, division algorithms and constraints, number and types of touch, set of data used and acquisition distance and camera type used.

The discussed technique in [5], focuses on how a computer vision system can detect, recognize, and interpret hand gesture identification with challenging elements such as pose, orientation, position, and scale. Different forms of gestures, such as numerals and sign languages, have been built into this system to perform efficiently for creating. Before the image processing, the image from the real-time video is examined using a Haar-cascade Classifier to detect the hand gesture. The detection of the hand is performed in this work using the theories of Region of Interest (ROI) and Python programming. As a consequence, the system was able to detect, recognize, and interpret hand gestures using computer vision utilizing Python and OpenCV, as well as generate hand gesture numbers and sign languages.

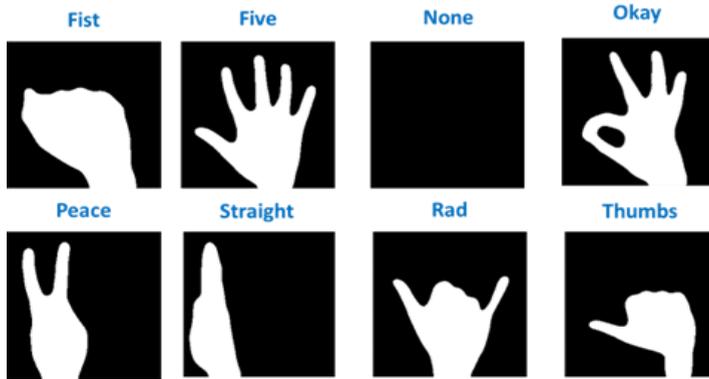
### **3. Methodology**

Gestures can range from any body movement or condition but usually from the face or hand. It is a research field that works on human-computer interactions and has many applications such as creating robotic music controls etc[7]. In our research, an actual application of the same is implemented which functions on various hand gestures. Diving deep into it an understanding was made that the main part or the heart of this research will be making use of a real-time hand gestures recognition. This approach can be used in developing many applications like a virtual mouse, a video player controller or virtual painters, and many more things. Gestures classify as static or dynamic in nature, which split the system into two.

#### **3.1 Static Hand Gesture Recognition**

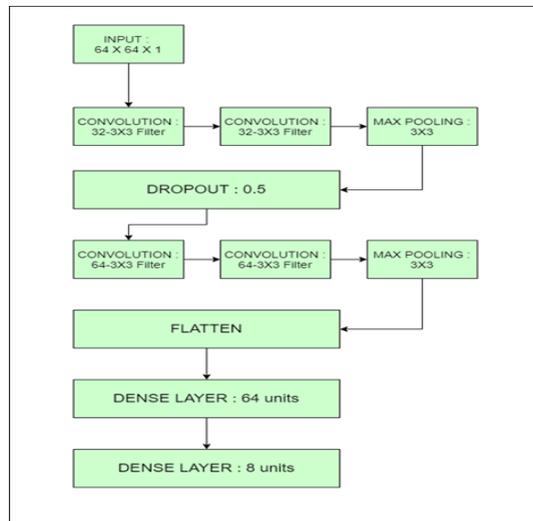
In static hand gestures, we have done multiclass classification based on eight different hand gestures using convolutional neural networks. For eight different classes of hand gestures, namely fist, five, none, okay, peace, rad, straight, and thumb, we took 1000 images per class, i.e., 8000 images, to train the model. Similarly, we have taken 125 images per class i.e., 1000 images for testing, and 125 images per class i.e., 1000 images for validation purposes.

The image attached here shows the eight gestures taken for model building and further development of the music player.



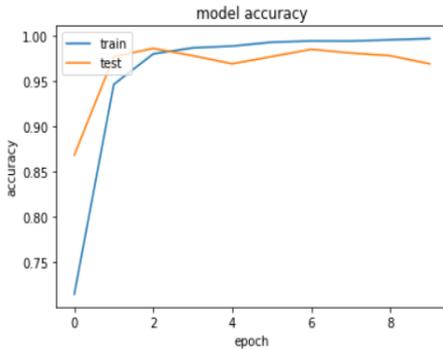
**Figure 1.** Eight gestures on which the model is trained.

The CNN model has 99.55% training accuracy and 98.73% validation accuracy. In order to get these results, we have built our model using the proper number of convolutions, max pooling, dense, dropout, and flattened layers.

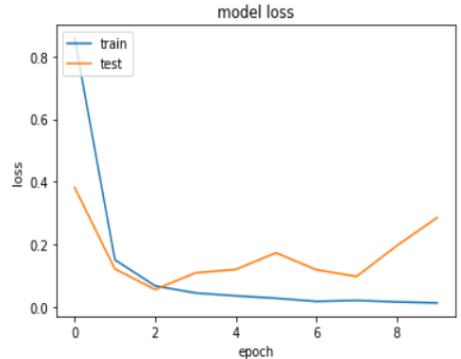


**Figure 2.** CNN Architecture for the model.

The above-mentioned architecture is used in a real-time application of a music player which is coded using python with the help of Keras and pygame libraries.



**Figure 3.** Training and Validation accuracy.



**Figure 4.** Training and Validation Loss.

The music player system has three main working modes.

**3.1.1. Test Model.** This mode deals with model testing. In this, we take the input of eight gestures namely .st, .ve, none, okay, peace, rad, straight, and thumb, and check whether the system is detecting it properly or not. As soon as the system detects all the gestures properly, the user is taken to the next mode by pressing a keyboard key where the user is able to listen to the music using all the gesture controls.

**3.1.2. Gesture Mode.** This mode deals with the overall working of the music player using static gesture detection. In this, every gesture gives its functionality. Once the particular gesture is detected, the activities linked with it are invoked and music is played accordingly. In our system, rad is used to load the song, fist is used to play the song, five is used to pause the song, peace is used to decrease the volume, okay is used to increase the volume, straight is used to stop the song and none retains the previously working action.

**3.1.3. Reset.** This mode is used to reset the background which results in starting the new session of the music player window so to invoke the music player window, we need to set the background first.

## 3.2. Dynamic Hand Gesture Recognition

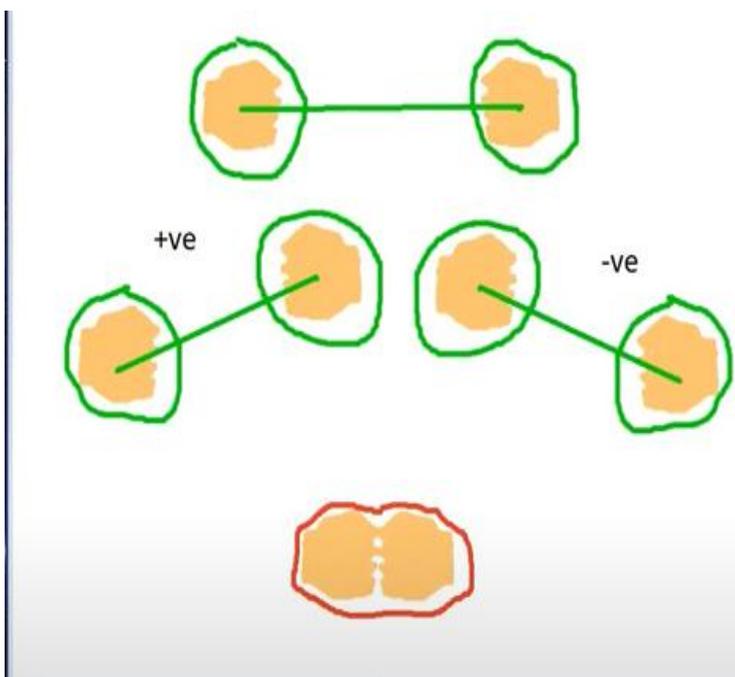
The system has three main components,

- Hand Tracking & Gesture Recognition
- Hand Position Detection
- Keyboard Control

**3.2.1. Hand Tracking and Gesture Recognition.** This is the region that will be marked for both of the user's hands to mark the pivot positions using the hand gloves colour, with the goal of creating a Centre point for the hands' movement region. The region designated for both hands will be used for the entire task of

tracking the hand movement gesture and performing front-end application control. This is a critical task that must be completed correctly across the entire system. The colour of the hand gloves determines the hand's Centre. This glove is used to track the centre of the hand for gesture movement. The pivoting of the Centre positions will be used in this process. The distance between the two hands is calculated after detecting the pivot region.

The user's hand movement gesture is the system's input. A Webcam is used to provide input to the system, which captures continuous frames to create a video that tracks the Centre of the hand as it moves away making a gesture. The application's UI also shows the same video frame. The webcam continuously captures frames, which are then processed to map the motion of the user's hand gesture to the keyboard cursor input for the running game application.



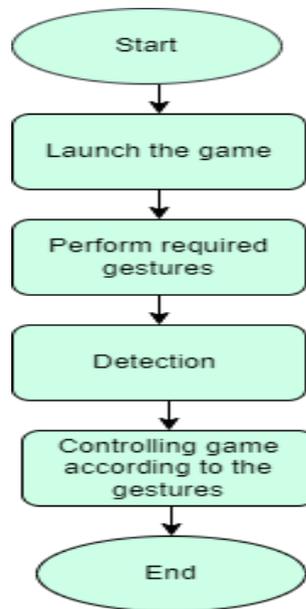
**Figure 5.** Generation of Contours[2]

**3.2.2. Hand Position detection.** The hand's position is tracked by the pivot region's centre and the hand itself. The direction in which the hand is moved will be determined by dividing the location of the hand centre by the location of the pivot region's centre, i.e., locating the centres of both hands and drawing an imaginary line connecting them and calculating the angle difference using horizontal as a base. The track of the hand movement gesture will be identified using this position detection.

**3.2.3. Keyboard Control.** It will be a simple matter of mapping different hand gestures to specific functions once the hand gestures have been recognized. The next step would be to see if there was a hand gesture made. If a hand gesture is made, the function is used to control the front-end application again by sending the input to it.

The loop is exited if no change in hand position is detected, and it will be restarted when a change in hand position is detected.

The user's hand movement gesture is the system's input. A Webcam is used to provide input to the system, which captures continuous frames to create a video that tracks the centre of the hand as it moves away making a gesture. The application's UI also shows the same video frame. The webcam continuously captures frames, which are then processed to map the motion of the user's hand gesture to the keyboard cursor input for the running game application.



**Figure 6.** The proposed system for the dynamic game controller using hand gestures.

Libraries Used:

1. The OpenCV- The package is used for image processing and performing computer vision tasks.
2. Pynut- This is a package for controlling and monitoring the keyboard.

The Python OpenCV API includes a wide range of tracking algorithms, each of which has been developed to be better than the one before it. When it comes to an effective service of Real-Time object tracking, the CSRT Algorithm is the appropriate algorithm in the OpenCV Library. This is why the app achieves the most efficient, approximate, and quickly recognized gesture translations. Following that, the Gesture Tracking Input Conversion Logic receives these tracked inputs.

The logic of the Gesture Tracking Input Conversion Logic estimates the precise position of the monitored Hand Gesture input. Hand Gloves of a certain hue, designated in the program for both hands, track this position. The distance between the hands is then determined. This is accomplished by utilizing graph ideas to trace the distance between both centres and determining the exact position of both hands. After finding the locations of both hands, an imaginary line connecting their centres

is drawn, and the angle is determined using the horizontal as a reference. This block now supports angle values and creates the keyboard letter value for the Hand Gesture input's suitable location. If the imaginary line is parallel to the horizontal, it activates the 'W' keychar and sends a keyboard cursor control signal to the Game application operating alongside the system, according to the settings. Similarly, if the imaginary line is oriented perpendicular and the distance is zero, the 'S' keychar is pressed and the keyboard cursor control signal is transmitted. Simultaneously, it looks for the Right Turn Gesture Track and sends the 'D' keychar keyboard cursor control signal to the Game Application if it's on the right side of the plot. The 'D' keychar is activated in opposition to this gesture, and the keyboard cursor control signal is provided to the gaming program at the same time.

## 4. Results

### 4.1. Static Hand gesture recognition

The images obtained from gestures-controlled music players are as follows: The accuracy obtained by the trained model is approximately 97%, whereas the validation accuracy of the trained model is 98%.

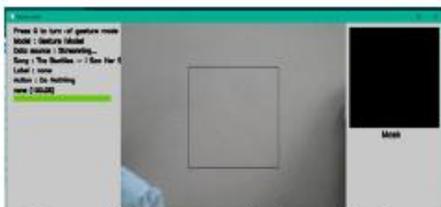


Figure 5. None gesture in the music player

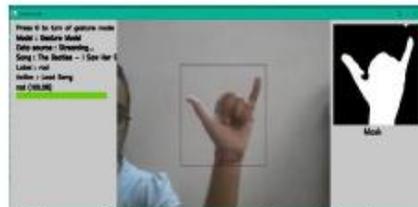


Figure 6. Rad gesture in the music player

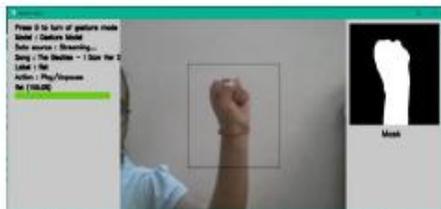


Figure 7. Fist gesture in the music player



Figure 8. Straight gesture in the music player

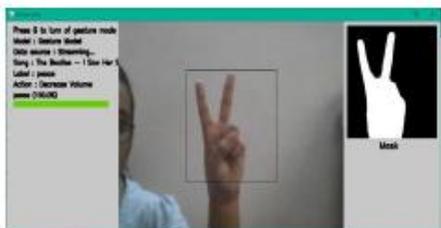


Figure 9. Peace gesture in the music player

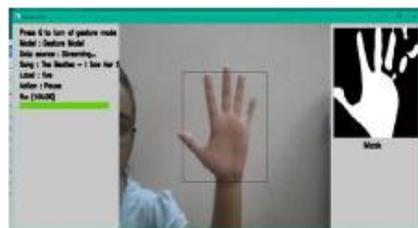


Figure 10. Five gesture in the music player

## 4.2. Dynamic Hand Gesture Recognition

Below are the results of the Dynamic game controller using Hand-Gestures.



Figure 11. Accelerating



Figure 12. Turning Right



Figure 13. Reverse



Figure 11. Turning Left

## 5. Conclusion and Future scope

The OpenCV package is used to construct the machine vision-based keyboard cursor control utilizing the hand gesture system in Python. The technology can follow the user's hand and control the movement of a keyboard cursor while playing a game. Different hand movements will be used to control the keyboard cursor. The device has the potential to be a viable computer keyboard replacement, but owing to the limits faced, it will not be able to totally replace the computer keyboard. When the template matching hand gesture identification approach is combined with a machine learning classifier, the accuracy of hand gesture recognition may be enhanced.

This application may be regarded as a beginning in the field of hand gesture applications, therefore it can be greatly improved. Using some more hand movements principles and Open CV Algorithms, this program may be developed to control the mouse cursor. With the use of Neural Networks-based logic, far higher precision may be achieved. Performance tracking may be enhanced to achieve better outcomes. When the template matching hand gesture identification approach is combined with a machine learning classifier, the accuracy of hand gesture recognition may be enhanced.

This will take a lot longer to build, but it will enhance the accuracy of gesture detection.

## References

- [1] B Nandwana, S Tazi, S Trivedi, D Kumar and S K Vipparthi A survey paper on hand gesture recognition 2017 7th International Conference on Communication Systems and Network Technologies (CSNT) pp. 147-152
- [2] C A Octavian, H Mihaela and I J Catalin Gesture recognition using python 2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD) 139-144
- [3] Munasinghe, Nuwan Dynamic Hand Gesture Recognition using Computer vision and Neural Networks 2018 10.1109/I2CT.2018.8529335
- [4] Kajan, Slavomír, Jozef Goga, and Ondrej Zsíros Comparison of Algorithms for Dynamic Hand Gesture Recognition 2020 Cybernetics and Informatics(K&I) IEEE.
- [5] Ismail, Ahmad Puad, et al Hand gesture recognition on python and OpenCV 2021 IOP Conference Series: Materials Science and Engineering Vol. 1045, No. 1. IOP Publishing
- [6] Suresh, Malavika, Avigyan Sinha, and R P Aneesh Real-Time Hand Gesture Recognition using Deep Learning 2019 International Journal of Innovations and Implementations in Engineering
- [7] Zhang and Z. Shi Deformable deep convolutional generative adversarial network in the microwave-Base hand gesture recognition system 2017 International Conference on Wireless Communications and Signal Processing (Nanjing)
- [8] K.-Y. Lian, C.-C. Chiu, Y.-J. Hong and W.-T. Sung Wearable armband for real-time hand gesture Recognition 2017 International Conference on Systems, Man, and Cybernetics (Banff)
- [9] N. Siddiqui and R. H. M. Chan A wearable hand gesture-recognition device based on acoustic Measurements at the wrist 2017 39th Annual International Conference of the IEEE Engineering Medicine and Biology Society (Seowipo)
- [10] V. Sombandith, A. Walairacht, and S. Walairacht Hand gesture recognition for Lao alphabet sign language using HOG and correlation 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information technology (Phuket)