Computer Control Using Vision-Based Hand Motion Recognition System

Anshal Varma[1], *Sanyukta Pawaskar*[2], *Sumedh More*[3], *and Ashwini Raorane*[4]

[1]Department of Electronics Engineering, Ramrao Adik Institute of Technology, D. Y. Patil Deemed to be University, Navi Mumbai, ans.var.rt18@rait.ac.in

[2]Department of Electronics Engineering, Ramrao Adik Institute of Technology, D. Y. Patil Deemed to be University, Navi Mumbai,san.paw.rt18@rait.ac.in

[3]Department of Electronics Engineering, Ramrao Adik Institute of Technology, D. Y. Patil Deemed to be University, Navi Mumbai, sum.mor.rt18@rait.ac.in

[4]Department of Electronics Engineering, Ramrao Adik Institute of Technology, D. Y. Patil Deemed to be University, Navi Mumbai, ashwini.raorane@rait.ac.in

Abstract. In our day-to-day communication and expression, gestures play a crucial role. As a result, using them to interact with technical equipment requires small cognitive data processing on our part. Because it creates a large barrier between the user and the machine, using a physical device for human-computer interaction, such as a mouse or keyboard, obstructs the natural interface. In this study, we created a sophisticated marker-free hand gesture detection structure that can monitor both dynamic and static hand gestures. Our system turns motion detection into actions such as opening web pages and launching programs. This system will bring a revolution in various industries, which has the potential to replace traditional devices and time-consuming computer handling methods.

1 Introduction

Today's world demands Human Control Interaction. The easiest and most natural way of communication is Hand gestures. Gesture recognition is a fast and accurate means of interacting with the computer. It is used in human-computer interaction for text input. The user's hand gestures are captured and decoded to determine the desired outcome coded with that gesture. Researchers in the IT industry have been looking for many intuitive ways to interact with computers and other devices. They are looking for an easy way to interact with these devices through measly communications between the user and the device. Besides the conventional way (i.e., keyboard and mouse), nowadays there are other means such as voice commands (Alexa, Google Assistant), and gestures (virtual reality) that are getting popular in particular sectors, such as remotely controlling equipment at dangerous places.

Gesture recognition is a subset of pattern recognition technology that identifies specific human gestures via sensors in our computer devices. The idea is to provide a more natural and intuitive way for people to interact with their devices. No mouse, no trackpad, no keyboard. Modern gesture recognition technologies are made up of two main types. A sensor-based (or haptic-based) and camera-based (or image-based) system. Wearable equipment, positioned on the arms or hands or both, involves techniques involving the use of additional devices, which should be worn by the user. Through devices capable of estimating position and movement, such as accelerometers and gyroscopes, these devices can map the positioning of the user's members in a way to extract information about their movement.

On the other hand, vision data-based groups employ cameras to capture the user's movement without the need for extra equipment. The data captured by the camera may include depth pictures, infrared signals, and other means if different types of cameras are used in addition to the more standard RGB frame capture. The data captured should be by the means of a camera and not with any sensor or device attached to the user. Vision-based approaches are deemed less invasive to the user, allowing for broader use.

Machine learning approaches gained prominence in academia and business as a result of advancements in the computer sector to handle natural language processing issues, e.g., voice recognition and vision computational challenges. The complexity of

computing is one of the major obstacles to employing these video classification approaches to accomplish the issue of gesture recognition. This is related not just to the volume of data contained in a video, but also to the complicated techniques employed in data processing employing neural networks, such as 3D convolution networks (3DCNN), and LSTM (Long-Short Term Memory) networks, Optical Flow processing, and so on.

This article proposes a convolutional neural network model gesture recognition tool designed for use in systems shipped. More specifically, gesture recognition, that is, making predictions for each new frame of image obtained using only video capture in RGB in order to not require a special camera for use. This technique was developed based on high-accuracy models aimed at generic applications, noting which structures would be feasible to implement as their own versions for systems embedded, with a view to minimizing the complexity of computation and keeping accuracy compatible with these techniques.

2 Literature Survey

Research in the fields of human-computer interaction (HCI) and its use in virtual environments are done a lot lately. Researchers have attempted to use video devices for HCI to identify virtual objects in order to control the system environment. Various natural motions can be identified, tracked, and analyzed by utilizing web cameras as the input device. Sarita Gavale et al. [7] worked on a system that uses the ultrasonic sensor HCSR04 to detect the motion of the hand with certainty. This sensor's range is around 1 to 13 feet. A hand gesture controller, also known as human activity gesture, detects hand movement. During the sensor testing for this project, the Sensor Kit UNO Arduino kit was used, which is required for processing raw data with microcontrollers.

[2] Aashni Haria et al. built a robust and engaging gesture-based system. These motions are then utilized to handle and control various desktop elements. To detect fingertip, this system use the Graham scan technique and the contour detection algorithm. Chua, S.N.D., et. al. [4] For hand gesture identification and localization, their project uses a single RGB camera. Due to the lack of standard gestures intended for human–computer interaction, the authors created their own dataset comprising all gestures used for training and their accompanying commands. Bedregal, Benjamin, et al. [13] introduced a fuzzy rule-based method for hand gesture identification. The method is extremely reliant on the preceding extensive examination of the properties of the gestures to be detected, as well as the manual transfer of those results to the recognition system. Experiments with users were conducted by Rios-Soria et al. [9] to assess the suggested gesture-detection algorithm's functionality. We asked participants to make a set of motions and actions in front of the camera and then measured whether or not the movements were detected. An observer kept track of whether the algorithm's output matched the actual gesture being performed.

Hojoon Park [10] used the index finger to move the cursor and the angle between the index and thumb to click events. Chu-Feng Lien [11] controlled the mouse cursor with only his fingertips, and his clicking approach was based on image density, requiring the user to hold the mouse cursor at the target place for a brief time. Fingertip tracking was used by A.Erdem et al. [12] to control mouse motion. A mouse button click was implemented by designing a screen such that when a user's hand passed over the region, a click occurred. Another way of clicking was utilized by Robertson et al. [13]. To mark a clicking event, they employed the thumb's motion (from a "thumbs-up" posture to a fist). The mouse cursor was moved by moving the hand while making a certain hand sign. Without the use of special color objects or gloves, Shahzad Malik devised a real-time system that can track the 3D position and 2D orientation of each hand's Index finger and thumb. Nasser H. Dardas et al. created a 3D game control system based on finger gesture detection.

3 Problem Formulation

Gesture-based engagement is becoming more common in our daily lives, yet we are still not taking advantage of its full potential. Gestures provide the user with a new form of interaction that closely resembles their participation in the real world. This seems natural and does not necessitate the use of an additional device or interruption. Furthermore, instead of a single input, they provide the user with multiple options. This project helps to overcome the disadvantages of standard input techniques by allowing the user to intuitively give commands. Devices have become an increasingly crucial element of our daily lives as the information society develops. In recent decades, the most common form of human-computer interaction (HCI) has been based on basic input devices such as the keyboard and mouse. This is effective, but the technique's underlying flaw is that there is an inherent lack of naturalness and physical connection with the computer.

With the introduction of new display technology such as virtual reality, this constraint becomes even more critical. As a result, some novel solutions for overcoming the HCI bottleneck have recently been devised. This situation has been thoroughly investigated, and numerous viable options have been identified. People frequently rely on their hand movements to engage with the environment, hence gesture interaction is believed to be superior to other ways. As a result, it gives users a natural and three-dimensional experience. As a result, the system that must be designed must meet the following requirements. i.e.

- The system will look for human behavior

- Process the action and convert it to input
- Lastly, perform a user-defined action

4 Design Methodology

4.1 3D CNN

This review article takes a look at the fundamentals of CNN and how it may be used to a variety of radiological jobs, as well as its challenges and future directions in the field of radiology..This paper also covers two difficult conditions in using CNN for radiological tasks: short datasets and overfitting, as well as techniques to avoid them. Knowing the standards and benefits of CNN, as well as its limits, is essential for maximising its potential in diagnostic radiology, with the objective of improving radiologists' overall performance and patient care.

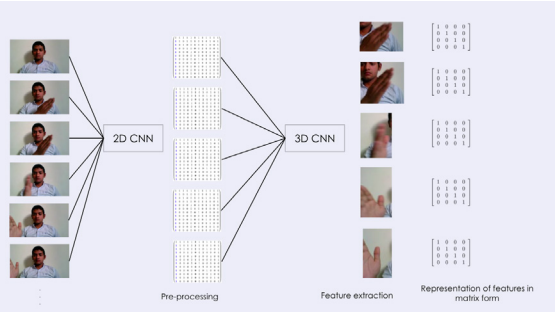

Fig 1. Work flow in CNN

In 2D CNNs, convolutions are implemented at the 2D characteristic maps to compute functions from the simplest spatial dimensions. When implemented to video evaluation problems, it's suitable to seize the movement statistics encoded in a couple of contiguous frames. To this end, we advise to carry out 3-d convolutions withinside the convolution levels of CNNs to compute functions from each spatial and temporal dimensions. The 3-d convolution is executed through convolving a 3-d kernel to the dice shaped through stacking a couple of contiguous frames together. By this construction, the characteristic maps withinside the convolution layer is hooked up to a couple of contiguous frames withinside the preceding layer, thereby shooting movement statistics a 3-d convolutional kernel can simplest extract one form of functions from the body dice, for the reason that kernel weights are replicated throughout the complete dice. A standard layout precept of CNNs is that the quantity of characteristic maps need to be elevated in past due layers through producing a couple of varieties of functions from the identical set of lower-degree characteristic maps.

4.2 RT3D 16F

For the development of the proposed network, the structures of the network that compose the state of the art were analyzed. Most of these networks do not have the purpose of executing in embedded systems, and certain structures that make up these networks end up not being good candidates for this purpose. One example of this is the processing of flow frames using algorithms of Optical Flow, which is widely present.

In preliminary tests, the use of this type of processing has proven computationally expensive (high computation time), even in simplified versions of the algorithm. The 3D convolution structure is present in most state-of-the-art gesture recognition techniques. It is on this that the models proposed in this article are based. The level of computational complexity of a layer of a 3DCNN network depends on the dimensionality of the input tensor. A tensor, in this context, serves as a generalisation of dimensional numerical representations such as vectors (1D) and matrices (2D), however, being able to represent any representation containing N dimensions. In order to reduce the complexity of video frames directly on a 3DCNN network, the model recommends having a 2DCNN network that receives the frames. preprocesses these, thus reducing the dimensionality of the data, in addition to aiding in the extraction of spatial features (being able to identify elements present in the frames, such as hands, arms, etc.)

.

Layer	Input Channels	Output Channels	Filter	Pooling
2D Conv	3	16	[3,3]	No
2D Conv	16	32	[3,3]	No
2D Conv	32	64	[3,3]	[2,2]
2D Conv	64	128	[3,3]	[2,2]
2D Conv	128	256	[3,3]	[2,2]
3D Conv	256	256	[3,3,3]	No
3D Conv	256	256	[3,3,3]	[2,2,2]
3D Conv	256	256	[3,3,3]	[2,1,1]
3D Conv	256	256	[3,3,3]	[2,2,2]

Table 1. 2D and 3D Convolution Layers for the proposed model

The layers that make up the blocks of 2DCNN and 3DCNN are listed in Table I. These are the main network parameters. In addition, there are activation layers using ReLU [14] (an activation function that maintains the signal identical for positive values and zeroes for negative values) and batch normalization layers, which were omitted for simplicity. The linear block of the network consists of 3 layers. This block has the function of reducing the output tensor of the 3DCNN network to progressively transform the 6144 nodes into 3072 nodes, then 1024 nodes, and finally into the number of classes existing in the model, depending on the data set.

5 System Implementation

This project proposes a convolutional neural network model gesture recognition tool designed for use in modern systems. The adaptation of 2D convolution networks to operate in one more dimensionality is the 3D convolutional network (3DCNN). Video frames are images with two-dimensional representations. When several frames of video are combined, the information is now represented by three dimensions: the height of the frames, the width of the frames, and the different frames themselves.

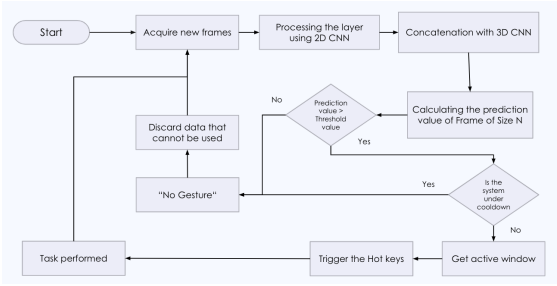

Fig 4. Architecture Diagram

The model will be trained on a dataset consisting of 148,092 sample images containing gestures divided into 26 classes. Here, the input stream of webcam video is set at 12 to 16 frames per second, as the greater the number of frames per second, the greater the amount of movement information contained in the images and, therefore, the greater the expectation of the highest accuracy for the technique. The size of frame will have a height of 100 px and a width of 200px for processing the image.

Webcam-based analysis is based on the user's distinguish data about their surroundings. The complete gesture interactive mechanisms that act as a building block for vision-based hand gesture recognition systems are composed of three fundamental phases. Hand detection and segmentation of related picture areas are the first steps in hand gesture recognition systems. A vast number of strategies have been presented that makes use of a variety of visual cues, as well as their combination in many circumstances. It is a quick approach to detect and operate at an image acquisition frame rate, but it may also be used for tracking if the detection method is fast enough to work at an image acquisition frame rate. Tracking hands, on the other hand, is notoriously difficult since they move quickly and their look might vary dramatically within a few frames.

The frame-to-frame connection of the segmented hand regions or characteristics towards interpreting the observed hand movements is known as tracking. Gesture recognition is a technique that allows users to conduct essential orders with their hands and other actions by providing devices with real-time data via motion sensors. As the major source of data input, the device's motion sensors recognise and analyse human motions.

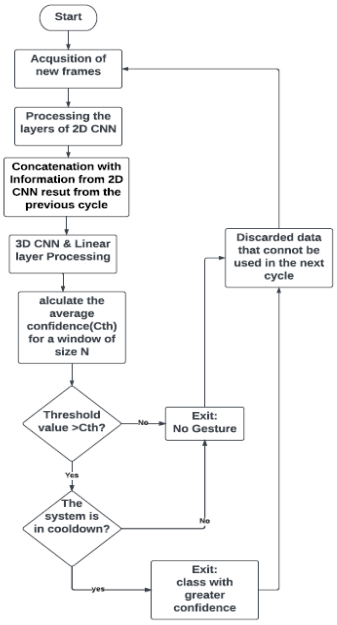

Fig 5. Flow of the Project

As a result, the identification of human hand movements by artificial intelligence (AI) systems has been a significant advancement in the previous decade, with applications in high-precision surgical robots, health monitoring devices, game systems, and other applications.Despite the fact that gesture recognition is still in its infancy, the creative market environment that has seen Virtual Reality become a global phenomenon forces us to recognise that its application domains may continue to expand. Finally, the entire undertaking proved to be a beneficial learning experience.

6 Result and Discussion

The developed project is a gesture detection based computer operating system which is expected to take gesture as the data from the user and take action given to that gesture. The image is trained using RT3D algorithm and a ckp file is created out of these trained images. The trained data is compared with the tracked images and the algorithm recognises the gesture and executes the action.

6.1 Initialization

The main file and the learned model are retained for initialization. We type "conda activate dante" to activate the conda environment for the machine, then "python main.py" to access the main file where the application is written, after launching cmd.

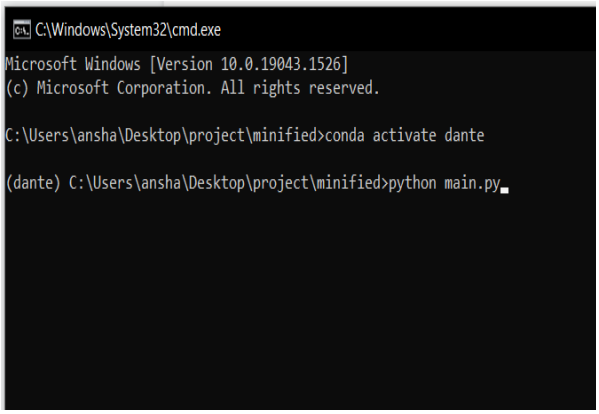

Fig. 6(a). Initialization

6.2 Gesture Detection

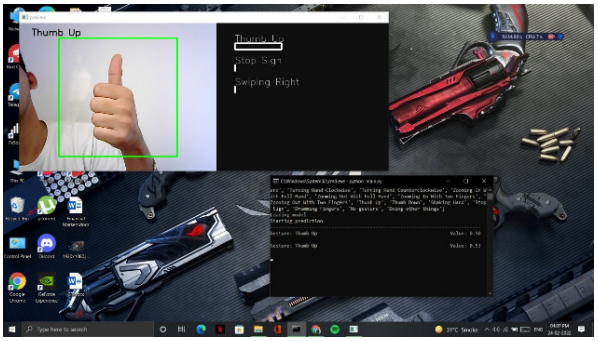

Fig. 6(b). Gesture gets Recognised

Following the program's initialization, all of the models are loaded, and the system begins predicting motions using a webcam. As soon as the user makes the motion, the trained model begins to forecast and displays the top three predictions.

6.3 Gesture Application

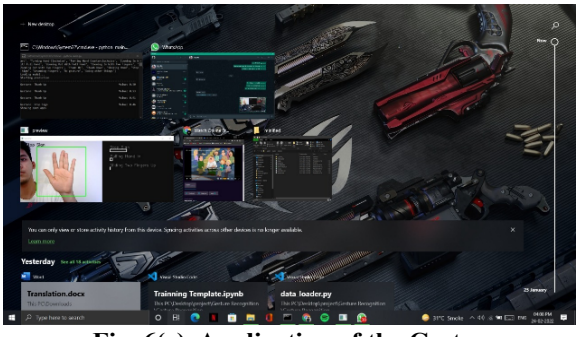

Fig. 6(c). Application of the Gesture

The model reacts to the gesture when it has been identified by the model. Depending on the active window in which the gesture is done, each gesture has a distinct action. For example, in the photo, the gesture was a stop sign, and the action key attached to it was win + tab, which displayed all open tabs.

6.4 Threshold Value

The value of the gesture made by the user is displayed in the cmd. The value for the threshold is 0.45. If the gesture value is greater than 0.45, the action associated with the gesture for the given active window will be performed. If the value falls below the threshold, it will pause for 0.1 seconds before predicting again.

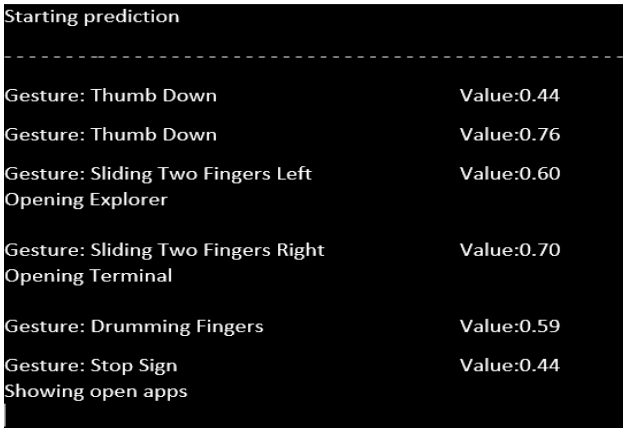

Fig. 6(d). Threshold Value

Given below (fig) is a list of the 20 actions performed by the system. You can find all the gestures and its action assigned to the system in the figure. In a similar way, various gestures can be added for performing different actions.

Implemented Gestures

Gesture	Desktop	Chrome	File Explorer	VLC
Stop Sign	Recent Tabs	Switch to another tab	Opening Chrome	Play/pause
Sliding two fingers Right	Opening Terminal	Open new tab	Taking Screenshot	Forward 10 seconds
Sliding two Fingers Left	Opening Explorer	Closing Tab	Closing Explorer	Rewinding 10 seconds
Sliding two fingers up	Virtual Keyboard	Maximize Window	Navigate to previous folder	Increase volume
Sliding two fingers down	Open YouTube	Minimize Window	Opens selected folder	Decrease Volume
Swiping up	-	Open Chrome	-	-
Swiping down	-	Minimize all	-	-
Zooming in	-	Zoom In	-	-
Zooming out	-	Zoom out	-	-

Fig 7. Gestures and their respective action

Accuracy & Analysis

To analyze the accuracy of the gesture, we tried to evaluate the prototype 20 users. Each of them performed six gestures with right and left hand, resulting in 240 observations. We calculated an average of observations for each gesture. Given below is the study.

Hand Used	Gesture Performed					
	Swiping Left	Swiping Right	Two Fingers To The Right	Two Fingers To The Left	Drumming Fingers	Shaking Hands
Right	98%	97%	95%	99%	100%	91%
Left	95%	99%	91%	95%	96%	92%
Total	96.5%	98%	93%	97%	98%	91.5%

Table 2. Analysis of the gesture accuracy

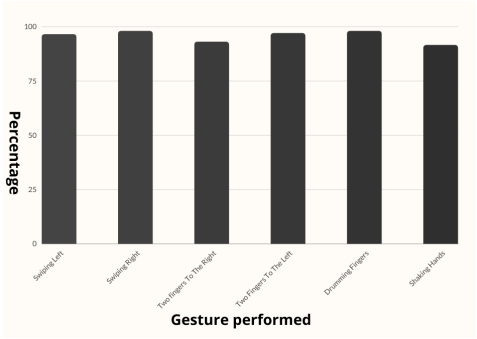

Fig 8. Analysis of the gesture accuracy

7 Conclusion

Thus, the project aims to provide control over the computer without using traditional interfaces like mouse, and keyboard which somewhere becomes a bottleneck that depends on heavy machine user interaction. from reading lots of related articles we have learned that we can easily reduce this bottleneck with computer vision and make it scalable and be used at a large scale in various fields. By using a 2DCNN network to preprocess the frames, this study builds a hand gesture detection system for engaging with various applications such as picture browsers, games, and so on and gives a successful solution towards a user-friendly interface between human and machine. Using a low frame rate minimizes the quantity of data to be processed and allows us to execute our model with more time between frames. Finally, the flickering of the gesture-controlled mouse is decreased using the smoothening approach. Moreover, the proposed system gives more performance and uses a system of cooldown to avoid multiple computations of the same gesture.

8 Future Scope

Gesture recognition makes it easy to use a variety of devices, including personal navigation devices, computers, laptops, and mobile phones.The growth trend of touchless gesture recognition in niche applications, including games, could boost the growth of the touchless sensor market during the forecast period.Medical applications have additional gesture recognition capabilities that nurses and doctors may not be able to touch the display or trackpad for health and safety reasons, but still need to control the system.Proper gestures, such as swiping by hand or using your finger as a virtual mouse, are a safer and faster way to control your device.

References

1. S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.
2. Aashni Haria, Archanasri Subramanian, Nivedhitha Asokkumar, Shristi Poddar, Jyothi S Nayak, Hand Gesture Recognition for Human Computer Interaction, Procedia Computer Science, Volume 115, 2017, Pages 367-374, ISSN 1877-0509
3. S. M. A. Hoque, M. S. Haq and M. Hasanuzzaman, "Computer Vision Based Gesture Recognition for Desktop Object Manipulation," 2018 International Conference on Innovation in Engineering and Technology (ICIET), 2018, pp. 1-6, doi: 10.1109/CIET.2018.8660916.
4. Chua, S.N.D., Chin, K.Y.R., Lim, S.F. et al. Hand Gesture Control for Human–Computer Interaction with Deep Learning. J. Electr. Eng. Technol. (2022).
5. S. Song, D. Yan and Y. Xie, "Design of control system based on hand gesture recognition," 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), 2018, pp. 1-4, doi: 10.1109/ICNSC.2018.8361351.
6. F. Brandolt Baldissera and F. L. Vargas, "A Light Implementation of a 3D Convolutional Network for Online Gesture Recognition," in IEEE Latin America Transactions, vol. 18, no. 02, pp. 319-326, February 2020, doi: 10.1109/TLA.2020.9085286.
7. D. Xu, Y. Chen, C. Lin, X. Kong and X. Wu, "Real-time dynamic gesture recognition system based on depth perception for robot navigation," 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2012, pp. 689-694, doi: 10.1109/ROBIO.2012.6491047.
8. Gavale, Sarita & Jadhav, Yogesh. (2020). HAND GESTURE DETECTION USING ARDUINO AND PYTHON FOR SCREEN CONTROL. International Journal of Engineering Applied Sciences and Technology. 5.271-276. 10.33564/IJEAST.2020.v05i03.043.
9. Chaudhary, Ankit & Raheja, Jagdish & Das, Karen & Sonia, Raheja. (2011). Intelligent Approaches to interact with Machines using Hand Gesture Recognition in a Natural way: A Survey. International Journal of Computer Science and Engineering Survey. 2. 10.5121/ijcses.2011.2109.
10. Rios-Soria, D.J. & Schaeffer, S.E. & Garza-Villarreal, S.E.. (2013). Hand-gesture recognition using computer-vision techniques. 1-8.
11. H. A. Jalab and H. K. Omer, "Human computer interface using hand gesture recognition based on neural network," 2015 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW), 2015, pp. 1-6, doi: 10.1109/NSITNSW.2015.7176391.
12. Gupta, Aviral & Sharma, Neeta & Scholar, M. (2020). A REAL TIME CONTROLLING COMPUTER THROUGH COLOR VISION BASED TOUCHLESS MOUSE. 9. 5077.
13. Lenman, Sören & Bretzner, Lars & Thuresson, Björn. (2012). Computer Vision Based Hand Gesture Interfaces for Human-Computer Interaction.