Desktop Control Using Dynamic Hand Gesture

Suraj Datir[1], Kanishk Suryawanshi[2], Sujal Mehra[3], Ashwajit Surwade[4], Parth Samgir[5], Dr.Manisha Mali[6]

[1,2,3,4,5,6]Department of Computer engineering, Vishwakarma institute of information Technology

ABSTRACT
Hand gesture recognition is crucial in human-computer communication. As we can see, a lot more new technology advancements are occurring. Smartphones typically employ biometric authentication as one of them. Another modern type of human-computer interaction is hand gesture recognition, which enables us to operate our computers by shaking our palms in the direction of a webcam.
Robot control, home automation, human-machine interface (HCI), and other areas have all used gesture recognition. It is possible to classify hand motions as a type of nonverbal communication. The interplay between the hands of users and computer programs is the paper's main topic. The system engineered enables a user to control any computer feature or application by using only single hand gestures.

KEYWORD: Hand Landmark, Gesture Recognition, Mediapipe, Opencv

INTRODUCTION

A mouse is an input device that simplifies pointing with interaction. It includes mechanical and digital versions, with the latter replacing the former. The mechanical mouse uses a latex ball for cursor movement, while digital mice are now common.
Optical mice use LED sensors to track pointer movement. Laser-based mice, introduced for better precision and overcoming optical mouse limitations, followed. With significant technological advancements, wireless mice were developed to enhance accuracy and simplify mouse movement. There will always be restrictions because the mouse is a piece of hardware, and there might be problems like mouse clicks not functioning correctly and other things, no matter how much the precision of the mouse increases. Because the mouse is a hardware device, it too has a lifespan during which it is functional and must be replaced after that. Everything becomes virtualized as technology advances, including voice recognition. The spoken language is recognised and translated into text using speech recognition technology Thus, speech recognition and eye tracking, which uses our eyes to operate the mouse pointer, might eventually replace keyboards. The mouse may one day be replaced with eye tracking.
Gestures can take any shape, such as a hand picture, pixel image, or a human-given stance, as long as it doesn't require a lot of processing resources or difficulties to make the devices needed for recognition function. The virtual mouse that is the subject of this study is built using Pyautogui and OpenCV. Webcam pictures are captured using the computer vision Python module known as OpenCV. A Python package called Pyautogui is used to define keyboard and mouse actions. OpenCv techniques are used for different gesture detection.

LITERATURE REVIEW

Recording is done using a regular RGB web camera at 30 frames per second (fps) in the direction of a desktop computer with a 19-inch monitor [1]. They changed the camera's capture resolution to 1280 960 and downscaled it to 227 227, which is the CNN's input size. Restricting the training database would be the primary challenge. A real-world system is unable to hold an ever-growing volume of data.

[2] The results of the tests indicate that the proposed method can accurately track hands in real-time at 30 frames per second. There are still several serious limitations to this study. Consequently, the subsequent research endeavor aims to surmount these limitations and optimize the fingertip tracking algorithm.

Using the Mediapipe library to capture hand motions, Gestop is a novel framework for controlling the desktop using hand gestures [3]. The end user might request the addition of new functionalities. Not only does their framework provide a fully functional alternative to the mouse, but it's also easy to add more unique gestures and actions, enabling users to use gestures for a wider variety of desktop apps.

[4] provides a model that can identify hand gestures and use that information to forecast the English alphabet. CNN is used to extract characteristics and detect hand gestures. It may be used in the future for virtual reality concepts, human-machine interface, and other things. Additionally, computers (i.e., those without a mouse or keyboard), robots, and automobiles may be controlled using this gesture.

In [5], deep CNN is employed. Ten males and ten females are among the sources from which the dataset was compiled. After receiving an input image, a Deep Convolutional Neural Network prioritizes different parts of the image. The accuracy of the created model is 92%. The hand gesture categorization process requires 2.44 milliseconds.

The model developed in this paper not only refers to human controls but also detects colors. Main aim is to develop mouse movement without using the system mouse. OpenCV helps to capture the user video in 1920*1080 resolution and with fps of 40 [6]. The model is made to recognise gestures, and its subsequent use depends on the commands that users provide the system and how they want gestures to be recognised. The mouse pointer movement will also be recorded and made to function automatically at the same time.

[7] It comprises three key phases: hand gesture monitoring, finger detection through color identification, and cursor execution. A convex hull is created around the detected contour for hand gesture tracking, and hand features are obtained from the hull-to-contour size ratio. The study extracts mouse controls like pointer control, left click, right click, and scrolling capabilities.

The system obtains the original picture using a USB camera and OpenCV.Using the Otsu adjustable threshold segmentation in the Cr channels of the YCbCr color space, the acquired gesture image's color image sequences are segment [8]. Clustering is the foundation of the Otsu approach.The system may control the mouse pointer's movement and execute a left-click operation during the control phase by analyzing the gesture's motion direction.

[9]. The KCF method is used for hand tracking In order to recognise the hand characteristics of the ROI for hand gesture identification, deep CNNs are used. Test data sets recognition rate is 95.61%, and the training data sets recognition rate is 99.90%.

This solution enables remote management of a PowerPoint presentation through the recognition of four hand gestures captured by a standard USB 2 PC webcam with a resolution of 160 x 120 pixels [10]. The system captures input motion images, processes them, and compares the extracted attributes with a database of 100 images for each gesture using K-Nearest Neighbour classification.

This system provides a simple interface for those with disabilities as well as others. The characteristics of a picture are extracted using MATLAB's HOG feature. Version R2016a of MATLAB has a built-in HOG feature [11]. The k-nearest neighbour algorithm is used as a classifier to categorize these extracted pictures.With an accuracy of 82.71%, the gesture recognition system can identify the gesture and control the appliance.

Various reference colors are employed for tracking in the color matching-based tracking approach. A frame is partitioned into a pattern of squares cells for further analysis after being searched for pixels that match the reference colors [12]. After being labeled as active, cells with a high enough density are arranged into groups of cells based on proximity. Every time the system loses track of the hand for a long time, it must first look for an initialization gesture, which must then be repeated.

Real-time video input is captured using a webcam; the picture frames are retrieved, and then every image is processed. Each element of the mxn matrix with a specified resolution and a 1x3 matrix of red, green, and blue channels per element serves as the representation of a single frame [13].Prior to separating the blue and red components from the video, it is horizontally inverted. Subtraction involves initially converting a rotated image into grayscale, and then subtracting it from the separate red and blue band images. Centroid detection is done after the binary image has been created. By Using hand movements we can click left and right in our screen. [14] A single camera is used for video input and the frame resolution is set to 360 x 240 pixels. A color is selected from the camera frame that is used as the tracker for mouse movement. Color thresholding, centroid detection is done on the image.

This strategy makes advantage of Deep CNN. Several sources, including 10 men and 10 women, contributed to the dataset [15]. Assigning weights to various features of the input picture using a convolutional neural network with deep layers. The testing procedure begins with the identification of an image pixel in the movie, after which individual frames are selected for image processing. For the purpose of aiding error correction, the frame is transformed to gray scale in MATLAB. The accuracy of the created model is 92%. Hand movements are categorized in roughly 2.44 milliseconds.

Through a USB connection, the microcontroller sends the computer the processed, computed distance data that the sensors have supplied [16]. The computer's processing software categorizes the computed motions based on their mathematical representation and filters and analyses the distance data.

This research introduces a convolution neural network-based improved automated model for hand gesture identification (CNN) [17] The proposed approach detects the hand shape using both the depth and color from the sensor, ensuring its robustness in congested settings. The two main model are hand detection and gesture recognition.

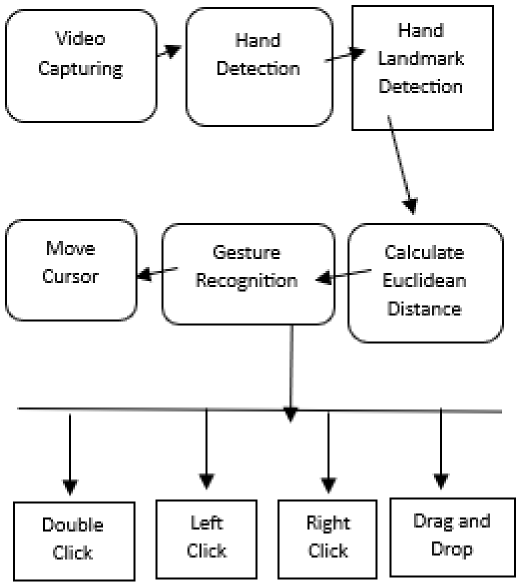
Image capture, manual the process of segmentation feature extraction, and categorization are the four steps of our method. Skin detection is used to identify skin areas from background pixels while taking a picture with the camera. Hand boundaries are added to a new image [18]. The hand gesture is described by extracting hand form elements. As a gesture classifier, an artificial neural network has also been used. For training, 120 gesture photos were employed. The average categorization rate that was attained was 95%.

The identification of hand movements is employed to frame the suggested solution since it has the advantage of being used naturally and does not call for an intermediary medium. According to this study, an DCNN, can rapidly detect hand gestures and categorize them while still keeping even the non-hand area [19]. For people who are visually challenged and/or have hand problems, the advised objective is to employ various hand movements using an integrated camera with deep learning.

Researchers were able to develop a reliable gesture detection system without the usage of any markers, making it more accessible and affordable. They sought to create gestures in this gesture recognition system that would cover nearly all facets of human-computer interaction, including system functionality, application launch, and opening of some well-known websites [20]

RESEARCH METHODOLOGY

Block Diagram



- **MediaPipe**

Mediapipe, an open source, multi-platform framework for building a system to analyze perceptual data across multiple formats, including audio and video, has been made available by Google. Numerous solutions are incorporated into MediaPipe, such as facial recognition and posture estimation. A single picture captured by a camera with a monocular lens can also yield 21 3D positions in X, Y, and Z. The surrounding box's height and width are used to normalize the X and Y coordinates of the produced output 3D coordinates. Using the wrist location as the origin, the Z coordinate, which represents the depth information, has a value that decreases with increasing distance from the camera and increases with increasing distance from the camera.

Hand Landmark

The hand landmarks model employs regression, or direct position prediction, to offer precise point placement of 21 3D palm-knuckle positions in the observed hand regions. One picture captured by a monocular camera may yield and be used to obtain 21 3D positions in X, Y, and Z. The enclosing box's width and height are used to normalize the output 3D coordinates' X and Y axes.The depth information is represented by a Z coordinate whose origin is the finger position.The Z coordinate's value changes depending on how near something is to the camera and how far away it is.

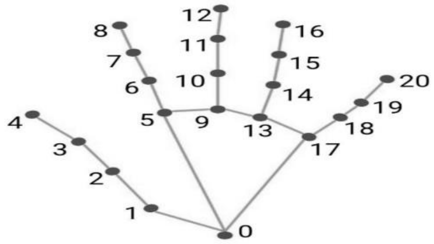


Fig.2. 21 Hand Landmark Points

$\tan(\Theta) = (m2 – m2) / (1\text{-}m1\ m2)$ (1)

where m1 is a horizontal line ie. 0 and m2 is the downward line drawn by Landmark 0 and Landmark 9

i.e., . $\tan(\alpha) = m2 = abs(y9 – y0 / x9 – x0)$ (2)

Since m1 = 0, therefore
tan(Θ) α m2 (3)
Since tan is a horizontal function, the angle will be directly proportional to the slope m2:
Θ α m2

- **Hand Classification**

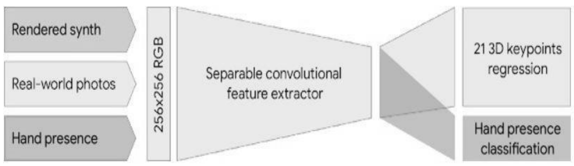
The recognition of left hand, right hand, major hand, minor hand, and dominant hand is based on the output of the MediaPipe hand tracking module and the classification information obtained from it. The classification labels for each detected hand is obtained from the classification information provided by the MediaPipe module. If the label for a hand is 'Right', it is considered as the right hand. Otherwise, it is considered as the left hand. Dominant hand is used to select the major and minor hand used by the system. If the right hand is the dominant hand, then the major hand will be set as the right hand and the minor hand will be set as the left hand and vice versa.

- **Image Preprocessing**

The image frames are read from an input video stream. As gestures need to be recognised, the images are flipped vertically around the horizontal axis to generate the mirror image corresponding to the input image. The images are changes from BlueGreenRed to RedGreenBlue. BGR (Blue-Green-Red) is the default color space for representing pictures in OpenCV, but RGB (Red-Green-Blue) is more often used in many other libraries and applications.Therefore, it's customary to convert the BGR pictures to RGB when using OpenCV and wanting to communicate with other libraries or display images in a typical RGB format.The images are converted to ‘read only’ mode until they are inserted into the results array for further gesture detection of the image to prevent accidental modifications.

BLAZEPALM

BlazePalm is a neural network-based algorithm used in Mediapipe for hand recognition and tracking. It is designed to accurately and efficiently detect hand regions in real time video or image data. It utilizes a lightweight CNN model to identify and track hand regions in real-time. BlazePalm is particularly useful for applications that require hand gesture recognition, hand tracking, or interaction with virtual objects using hand movements.



GESTURE CONTROL

Euclidean Distance: Signed and unsigned Euclidean distance is calculated between two points. Signed Euclidean Distance: Compares the y-coordinates of the two landmarks specified by the indices in point[0] and point[1] in the hand result object. If the y-coordinate of the first landmark is less than the y-coordinate of the second landmark, the sign is changed to 1.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (xi - yi)^2} \quad (5)$$

The Signed Euclidean distance between two landmark points in the hand is calculated considering the sign based on the comparison of their y-coordinates. The signed distance can be useful for various hand gesture recognition or hand motion analysis algorithms that rely on the relative positions of the landmarks. The system assigns different fingers based on the calculated ratios of distances between landmarks representing finger tips, middle knuckles, and base knuckles. Each inner list contains three elements representing the indices of landmarks corresponding to the fingertip, middle knuckle, and base knuckle. Binary representations of finger states are stored which defines whether a finger is in active state or inactive state. Further the distances are calculated accordingly and ratio between these are obtained for recognising gestures.

RESULT

The aforementioned technique provides an outline of the next evolution of human-computer interaction, including how to handle a system without putting the user in direct contact with it.It can be used without wireless technology.The system is evaluated with hand gesture recognition and fingertip detection in a variety of light and dark environments with varying distance from the camera for tracking the hand and fingertip detection.

Table1. Gestures and Actions

Gesture Used	Action Performed
Whole Palm	Hand Detection
V shaped fingers	Move Mouse
Mid finger	Left Click
Index Finger	Right Click
Palm and Fist	Drag and Drop
Two Fingers Crossed	Double click
Left Pinch Horizontally	Scroll

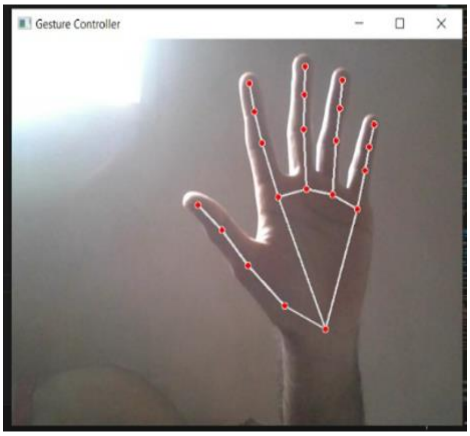


Fig.4. Palm Detection right hand (Hand Landmarks)

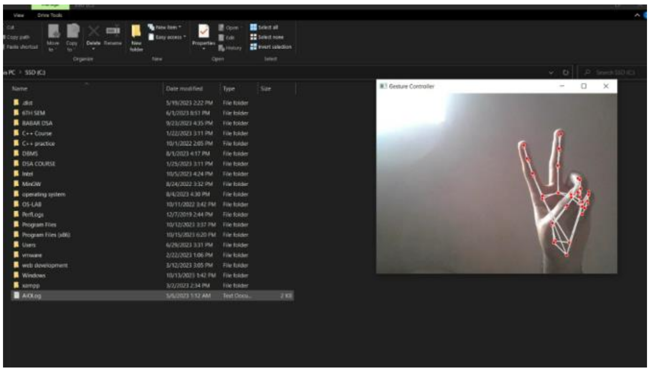


Fig.5. Cursor movement using Right hand V-gesture

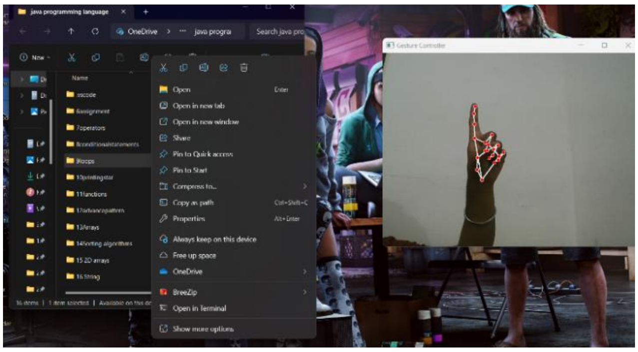


Fig.6. Right click using right hands index finger gesture

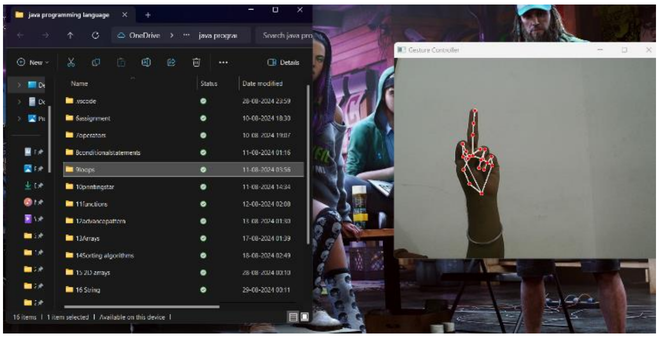


Fig.7. Left Click using right hands middle finger gesture

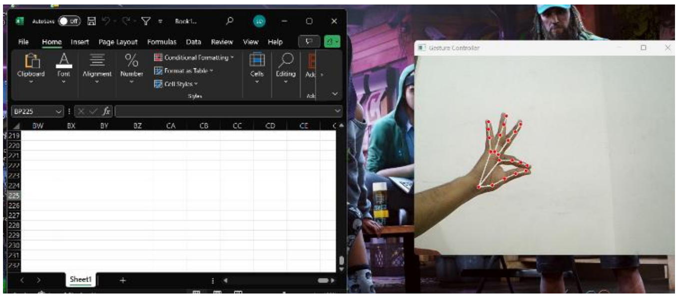


Fig.8. left hand pinch for scroll function

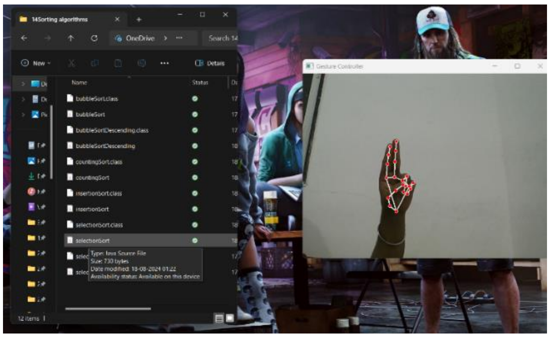


Fig.9. Right hand cross finger for double click

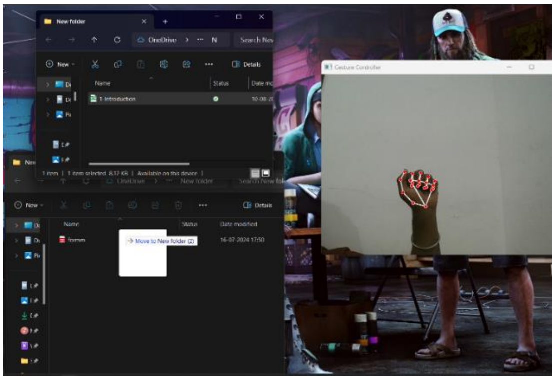


Fig.10. Palm and Fist for Drag and Drop

CONCLUSION

The evolution of human-computer interaction has accelerated over time. Future development of this gesture movement detection system has a lot of potential since the usage of gesture recognition in conjunction with controlling the mouse will make people's lives simpler and more intelligent. Employing hand gestures to operate a computer offers a viable and natural method to work with technology. It makes it unnecessary to use physical input devices like mouse or keyboards, enabling more realistic and immersive experiences.

The use of hand gestures to operate computers has the potential to revolutionize human-computer interaction and open the door to a more seamless and interactive computing experience with additional study and development. Hand landmarks and Euclidean distances are used during the whole process. Despite actually touching any hardware, it enables the user to manage every aspect of mouse operation, including movement, clicks, and drag and drop. To operate the mouse, it exclusively makes use of the system's built-in camera. Python, OpenCV, Mediapipe, and PyAutoGui utilities are all used in the program.To make it easier to illustrate how the system works, all motions appear on the screen of the camera with the relevant indication.

REFERENCES

1. Fuhl, W. (2020). “From perception to action using observed actions to learn gestures”. User Modeling and User-Adapted Interaction, 31(1), 105–120. doi:10.1007/s11257-020-09275-3.
2. Tran, D.-S., Ho, N.-H., Yang, H.-J., Kim, S.-H., & Lee, G. S. (2020). Real-time virtual mouse system using RGB-D images and fingertip detection. Multimedia Tools and Applications. doi:10.1007/s11042-020-10156-5

3. S K, S., & Sinha, N. (2020). Gestop: Customizable Gesture Control of Computer Systems. 8th ACM IKDD CODS and 26th COMAD. doi:10.1145/3430984.3430993
4. Yang, Z., & Zheng, X. (2021). Hand Gesture Recognition Based on Trajectories Features and Computation-Efficient Reused LSTM Network. IEEE Sensors Journal, 21(15), 16945–16960. doi:10.1109/jsen.2021.3079564
5. V. Niranjani, R. Keerthana, B. Mohana Priya, K. Nekalya and A. K. Padmanabhan, "System application control based on Hand gesture using Deep learning," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 1644-1649, doi: 10.1109/ICACCS51430.2021.9441732.
6. K. S. Varun, I. Puneeth and T. P. Jacob, "Virtual Mouse Implementation using Open CV," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 435-438, doi: 10.1109/ICOEI.2019.8862764.
7. V. V. Reddy, T. Dhyanchand, G. V. Krishna and S. Maheshwaram, "Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition," 2020 IEEE-HYDCON, Hyderabad,India,2020,pp.1-5,doi:10.1109/HYDCON48903.2020.9242677.
8. 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), Zhuhai, China, 2018, pp. 1-4, doi: 10.1109/ICNSC.2018.8361351.
9. H. -Y. Chung, Y. -L. Chung and W. -F. Tsai, "An Efficient Hand Gesture Recognition System Based on Deep CNN," 2019 IEEE International Conference on Industrial Technology (ICIT), Melbourne, VIC, Australia, 2019, pp. 853-858, doi: 10.1109/ICIT.2019.8755038.
10. T. P. Salunke and S. D. Bharkad, "Power point control using hand gesture recognition based on hog feature extraction and k-nn classification," 2017 International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2017, pp. 1151-1155, doi: 10.1109/ICCMC.2017.8282654.
11. S. Kaur, Anuranjana and N. Nair, "Electronic Device Control Using Hand Gesture Recognition System for Differently Abled," 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2018, pp. 371-375, doi: 10.1109/CONFLUENCE.2018.8442982.
12. C. J. Cohen, G. Beach and G. Foulk, "A basic hand gesture control system for PC applications," Proceedings 30th Applied Imagery Pattern Recognition Workshop (AIPR 2001). Analysis and Understanding of Time Varying Imagery, Washington, DC, USA, 2001, pp. 74-79, doi: 10.1109/AIPR.2001.991206.
13. S. Thakur, R. Mehra and B. Prakash, "Vision based computer mouse control using hand gestures," 2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI),Faridabad,India,2015,pp.85-89,doi: 10.1109/ICSCTI.2015.7489570.
14. Y. V. Parkale, "Gesture Based Operating System Control," 2012 Second International Conference on Advanced Computing & Communication Technologies, Rohtak,India,2012,pp. 318-323, doi: 10.1109/ACCT.2012.58.
15. V. Niranjani, R. Keerthana, B. Mohana Priya, K. Nekalya and A. K. Padmanabhan, "System application control based on Hand gesture using Deep learning," 2021 7th International Conference on Advanced Computing and Communication Systems(ICACCS),Coimbatore, India, 2021,pp.1644-1649,doi: 10.1109/ICACCS51430.2021.9441732.

16. G. Sziládi, T. Ujbányi and J. Katona, "Cost-effective hand gesture computer control interface," 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Wroclaw,Poland,2016,pp.000239-000244, doi: 10.1109/CogInfoCom.2016.7804555.
17. M. A. Rady, S. M. Youssef and S. F. Fayed, "Smart Gesture-based Control in Human Computer Interaction Applications for Special-needPeople,"2019NovelIntelligent and Leading Emerging Sciences Conference (NILES),Giza,Egypt,2019,pp.244-248, doi:10.1109/NILES.2019.8909324.
18. 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 NewSmart World(NSITNSW), Riyadh,SaudiArabia,2015,pp.1-6, doi: 10.1109/NSITNSW.2015.7176405.
19. V. Niranjani, R. Keerthana, B. Mohana Priya, K. Nekalya and A. K. Padmanabhan, "System application control based on Hand gesture using Deep learning," 2021 7th International Conference on Advanced Computing and CommunicationSystems(ICACCS),Coimbatore,India, 2021,pp.1644-1649,doi:10.1109/ICACCS51430.2021.9441732.
20. 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, https://doi.org/10.1016/j.procs.2017.09.092.