International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 06 | June - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Exploring The Techniques and Applications of Hand Gesture Recognition for Human-Computer Interaction

Neelam Khada[1], Payal Wadkar[2], Kareena Mulla[3], Tejal Kurkure[4], Jithina Jose[5],

[1]Dr. D.Y. Patil Institute Of Technology, Pimpri
[2]Dr. D.Y. Patil Institute Of Technology, Pimpri
[3]Dr. D.Y. Patil Institute Of Technology, Pimpri
[4]Dr. D.Y. Patil Institute Of Technology, Pimpri
[5]Dr. D.Y. Patil Institute Of Technology, Pimpri

Abstract - Hand gesture controlled applications are software applications that are controlled by the user's hand movements. Hand gesture controlled applications have the potential to improve user experience by offering more intuitive and immersive ways to interact with system. Hand gesture controlled applications have a huge scope, including gaming, virtual reality, robotics, healthcare, and various other sectors.
In this paper, we have combined several disparate end results of subsystems such as virtual mouse, contactless keyboard, finger counting, taking selfies, sign language, game, handling installed applications and volume control using computer vision. For implementation, we used the open-source software library OpenCV, mediapipe and pyautogui. Various hand gestures are assigned to launch and control different applications. These hand gestures are recognized using machine learning trained models, Euclidean distance formula and pixel positions. Overall, it allows humans to interact with computers through hand gestures rather than traditional input/output mechanisms such as the keyboard and mouse, improving accessibility.

***Key Words*:** Computer vision, OpenCV, hand gesture, machine learning, TensorFlow, pyautogui, mediapipe, virtual mouse, virtual keyboard, sign language, gaming.

1.INTRODUCTION

Computer vision is an interdisciplinary field of study that seeks to enable computers to interpret and comprehend digital images and videos in a manner similar to human vision. It creates algorithms and models capable of accurately interpreting and analyzing visual data and extracting useful information from it. Image segmentation, object detection and tracking, facial recognition, and gesture recognition are examples of such tasks.

We have used OpenCV (Open Source Computer Vision) and Mediapipe, two well-known libraries that provide tools for implementing real-time hand gesture recognition.

OpenCV is an open-source computer vision library that provides various functions and algorithms for image and video processing. In the context of hand gesture recognition, OpenCV can be used to detect and track the position of the hand, extract features such as color and texture, and classify hand gestures using machine learning algorithms. Mediapipe includes a number of pre-built solutions for tasks like hand tracking and pose estimation, as well as a pipeline for developing custom solutions. Mediapipe also includes tools for visualization, logging, and performance monitoring.

Our software integrates several applications together and is handled using assigned hand gestures. These applications include contactless mouse, contactless keyboard, finger counting, taking selfies, sign language, game, handling installed applications and volume control. An interface is created using tkinter library which permits navigation to different applications as per requirement. The webcam captures the visual input once the application starts and processes the frames. The module named 'solutions.hands' from mediapipe is used to detect hands followed by 'process()' function which processes multi-hand-landmarks. The 'drawing_utils' module is used to draw the landmarks of the hand which moves along the hand in motion in frame. Certain gestures depending on fingers extended and relative position of each finger with respect to other fingers are assigned to operate as per respective application gestures which is a primary principle upon which computing is based on.

2. LITERATURE SURVEY

An optical mouse and keyboard are made using hand gestures and computer vision. A person's hand can make a variety of gestures that are read by the computer's camera, and the computer's mouse or cursor will move in response to those gestures, even making right and left clicks with different gestures. Similar to this, keyboard functions can be accessed using a variety of gestures. For example, you can swipe left and right by using four numbers and one finger to select an alphabet. Without the aid of a wire or any other extraneous hardware, it will act as a virtual mouse and keyboard. An algorithm is created by using defect calculations, mapping the mouse and keyboard, and generating the Convex hull defects first. The Convex Hull algorithm, however, may malfunction if another external

noise or flaw is found in the webcam's operational range. [1]

The virtual mouse is operated by fingertip identification, and hand gesture recognition is suggested. In this study, two techniques are used to track the fingers: hand gesture detection and colored caps. There are three main steps in this process: (1) finger detection using color identification; (2) hand gesture tracking; and (3) implementation on the on-screen cursor. In this study, a convex hull is formed around the detected contour to create hand gesture tracking. To extract hand features, one uses the area ratio of the contour and hull formed. In-depth testing of this algorithm is conducted in real-world situations.[2]

By enabling hands-free interaction between people and computers, an algorithm is introduced to carry out mouse functions. It serves as an alternative to the conventional computer controlled by a mouse. by recognizing various facial expressions with computer vision, comparing them to previously stored expressions, and then taking actions in response to the move. People with physical disabilities will be helped by this algorithm to operate the mouse using their facial expressions and eye movements. They have the ability to scroll up and down, left and right click, move the cursor left and right, and up and down. Webcam, NumPy, dlib, and a few other crucial libraries are included in the system.[3]

It is suggested to use a single RGB camera in conjunction with an effective framework for recognizing hand typing motions and gestures to build a virtual keyboard. Virtual keyboards are discussed in a number of works in the Human Computer Interaction (HCI) field. Most of them use external equipment (depth sensor, leap motion, control glove, touch screen, etc.) as well as hand pose estimation, hand shape, and external equipment. This framework, however, works like a regular typing action in the air, similar to typing on a real QWERTY keyboard, and does not require any additional equipment or prior experience from users. Two hand typing gestures—touch and non-touch—are classified using convolutional neural networks (CNN). For each set of 10 fingers on two hands, we also train 11 non-touching and touching gestures. For the two gestures case, the proposed CNN model achieves a classification accuracy of 99.2%, and for the eleven gestures case, it achieves a classification accuracy of 91%.[4]

The brain's bioelectrical signals are collected by a computer-based system called a "Brian Computer Interface" (BCI), which then analyses and transforms those signals into commands that execute the user's intent. This study develops a BCI-based virtual keyboard with 36 keys, including 26 English alphabet keys (A-Z), 7 special characters, and 3 action keys, which can be operated by bioelectrical brain waves.

The study is broken up into two modules: hardware and software.Electroencephalogram (EEG) signals are analyzed for red, green, and blue (RGB) colors using an asynchronous mechanism. The results come from an experiment that was conducted over two sessions with the same participants (university students). Results of the experiment reveal an average spelling completion time of 2.3 minutes, a speed of 6.4 characters per minute (CPM), and an accuracy rate of 89.7%.[5]

Nowadays, a significant portion of people's personal lives are shared on social media. Automatically identifying individuals in personal photographs may greatly increase user convenience by making photo album organization easier. However, the traditional focus of computer vision for tasks involving human identification has been face recognition and pedestrian re-identification. Computer vision faces new difficulties when recognizing people in social media photos because of the uncooperative subjects (such as backward viewpoints and unusual poses) and significant changes in appearance. A straightforward framework for person recognition is developed in order to address this problem, utilizing convnet features from various image regions (such as the head, body, etc.). The proposed new recognition scenarios concentrate on the temporal and morphological difference between training and testing samples. The importance of various features in relation to time and viewpoint generalizability is thoroughly examined. On the PIPA benchmark, which is arguably the largest social media-based benchmark for person recognition to date, with a variety of poses, viewpoints, social groups, and events, it is shown throughout the process that the straightforward approach achieves cutting-edge results.[6]

Using the framework model of the gesture recognition system, gesture segmentation, gesture modelling and analysis, and gesture recognition, we methodically summarize the current research status of dynamic vision recognition technology in computer vision and analyze its drawbacks. The results suggest that the future development trends in this field will be deep vision sensor-based gesture recognition, multi-method cross-fusion gesture recognition, and gesture recognition based on straightforward wearable devices.[7]

For the purpose of recognizing hand gestures in Japanese Sign Language (JSL), a sensor-based data acquisition glove is being developed. Five flex sensors, an Inertial Measurement Unit (IMU), and three Force Sensing

Resistors (FSRs) are used to measure the amount of finger bending and hand movement data. The computer receives the detected data via an Arduino Micro. Using the Support Vector Machine (SVM) and Dynamic Time Wrapping (DTW) algorithms, the average hand gesture recognition accuracy for a single subject is 96.9% and 94.5%, respectively. Our proposed system achieves an average recognition accuracy of about 82.5% for cross-recognition among three subjects.[8]

We present a hand gesture recognition system that uses a sensor and a microcontroller to convert a gesture movement into a signal. The 1D Convolutional Neural Network (1D-CNN) used by this hand gesture recognition system is able to directly extract features from the temporal signals that were captured in their natural state. On a disabled person's mobile phone, an audio voice and text-based notification conveys the resulting word or phrase to a normal person. Additionally, to decrease latency in output prediction, the trained 1D-CNN model is implemented directly in the Android phone as opposed to running on the server. The trained model achieves recognition accuracy of 97.96% on test data made up of numerous samples of ten different patterns.[9]

New technologies have opened up a world of possibilities for mankind that were previously either impossible or miraculous. This project is just one example of the myriad possibilities in the field of computer vision. Through the use of a speech-based feedback device, this project aims to help the blind community experience the world on their own. This project suggests that future work should include identifying walkable areas, text recognition and text-to-speech, identifying and locating particular kinds of objects, and walking navigation. The COCO dataset and the YOLO object detection algorithm are used to achieve this. Our project will help a blind person walk more easily by identifying and avoiding obstacles in front of them. Additionally, it will help them read texts because OCR, which uses Python and an API for text recognition, makes this possible. The final output for the users is then produced using gTTS to convert the text to speech.[10]

3.RESEARCH GAP

Table-1: Survey of paper

References	Hand Gesture Representation															
	Parameters					Feature extraction Techniques		Classifiers								
	Hand Shape	Hand Orientation	Hand Location	Hand Motion	Others	Convolutional Neural network(CNN)	Convex Hull Algorithm	Others	Support Vector Machine(SVM)	Dynamic Time Wrapping(DTW)	Hidden Markov Model(HMM)	Distance metrices	Neural Networks	Deep Learning	You Only Look Once(YOLO)	K-Means	Others
[1]	✓	✓	✓	✓			✓	✓				✓	✓	✓			
[2]	✓	✓	✓	✓			✓					✓	✓	✓			
[3]					✓	✓						✓					
[4]			✓	✓		✓						✓	✓	✓			
[5]					✓	✓		✓	✓			✓	✓	✓			
[6]					✓				✓			✓		✓			
[7]				✓	✓			✓								✓	
[8]	✓	✓	✓	✓				✓	✓	✓							
[9]	✓	✓	✓	✓		✓							✓	✓			
[10]					✓			✓				✓			✓		

4. TECHNOLOGY USED

- Python
- MediaPipe
- TensorFlow
- PyCharm
- OpenCv
- PyAutoGUI
- Kera
- tkinter

5.METHODOLOGY

In existing systems there are several standalone applications which run via hand gestures. These standalone applications

serve only one functionality. But our software integrates several standalone applications together which exists as one coherent system. Our software aims to make use of hand gestures to provide ease of communication as they are the most natural way of communication. It minimizes the dependency on input/output devices like keyboard and mouse. It provides high compatibility, portability as the only requirement is camera. Software starts by presenting a User Interface comprising of icons for navigating to several applications. Clicking on the icon starts a particular application. Computer vision libraries like OpenCV, MediaPipe, TensorFlow are used to develop the software. The camera takes live video input and detects hand gestures. These gestures are then used for performing various tasks in the application. After closing the current application user returns to the main User Interface.

6. IMPLEMENTATION

1)Import OpenCV, MediaPipe, TensorFlow, PyAutoGUI, Tkinter

2)Create interface comprising of list of applications using Tkinter. Import PIL(Python Imaging Library) to load image and 'label()' function to display image. Create buttons using 'createButton()' function and use 'command' argument to call back function to open particular application.

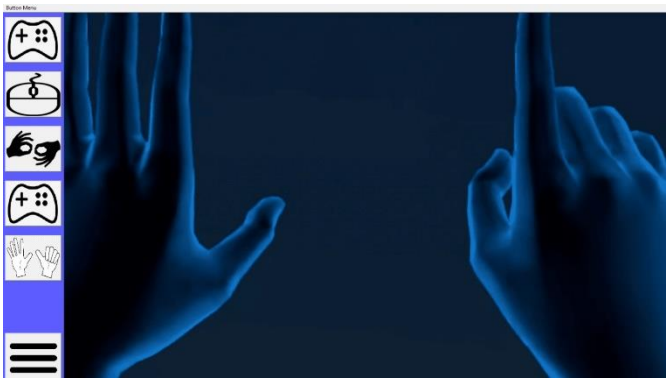

Fig -1: User Interface

3)Capture video using 'VidoeCapture()' . Create an object to read image using 'read()' function. Convert image from BGR to RGB using 'cvtColor()' function as mediapipe cannot process BGR image.

3)Create an object to detect hands using 'mediapipe.solutions.hands.hand()' . Call 'process()' function to process hand . Call 'multi_hand_landmarks' to detect multi hand landmarks. (img of hand landmarks)

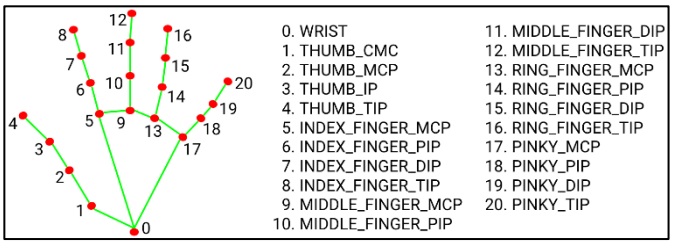

Fig -2: Hand Landmark Model

4)Call 'draw_landmarks' function from 'drawing_utils' class to draw landmark points on hand and use 'HAND_CONNECTIONS' class variable to show connection between landmark points.

(img hand corrdinate screen hand landmarks)

5)If application does not require trained model, use Euclidean distance between hand landmark coordinates to identify gesture and perform functionality. For example to perform mouse click , calculate Euclidean distance between tip of index finger(landmark index 8) and middle finger(landmark index 12) and if distance if less than certain threshold, perform left click using 'pyautogui.click()'.

6)Use teachable machine powered by Google to train the model. Create classes for classification of hand gesture. For Example for sign language recognition application, create classes from 'A' to 'Z' by uploading approximately 300 images of each class and henceforth, model is trained. Use 'classifier()' function from cvzone.ClassificationModule which takes real time images and .h5 file of model as input for classification of gesture.

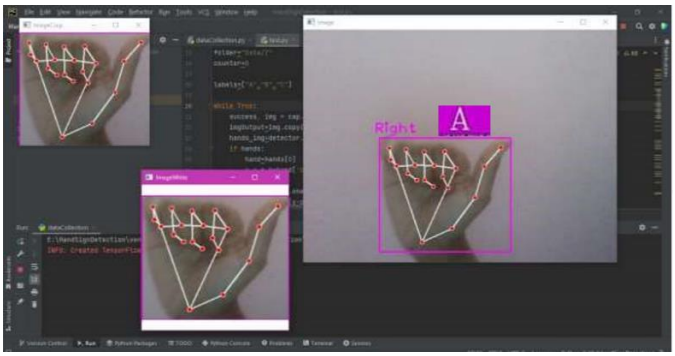

Fig -3: Hand Sign Detection

7)Close application after performing the task

8)Return to main interface

7. ALOGRITHM

- Import respective libraries
- Create interface comprising of menu for navigating/opening applications
- While true:
    - Take video input and convert it into frames.
    - Convert bgr frame to rgb for mediapipe processing
    - Detect and track hand in motion
    - Draw the hand landmarks
    - Identify the multi hand landmark coordinates and scale the coordinates to pixel position of screen
    - Control application via:
        - If it requires trained model: use tensorflow model
        - Else: calculate euclidean distance between landmarks to identify particular gesture
    - Perform task
    - Close application once done
    - Break while loop
- Return to main interface

8. MATHEMATICAL MODEL

8.1 Euclidean distance:

It is used to calculate the distance between landmark coordinates either along x-axis, y-axis. It can be calculated from the Cartesian coordinates of the points using the Pythagorean theorem, therefore occasionally being called the Pythagorean distance.

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

.

8.2 Scaling:

Initially the hand landmark coordinates are float points in range of 0 to 1 with respect to x, y and z-axis of frame.

These coordinates are converted into integer values to represent the pixels of frame using the following formula:

cx = int(lm.x * frame-width)

cy = int(lm.y*frame-height)

where,

lm.x = x-coordinate of landmark

lm.y = y-coordinate of landmark

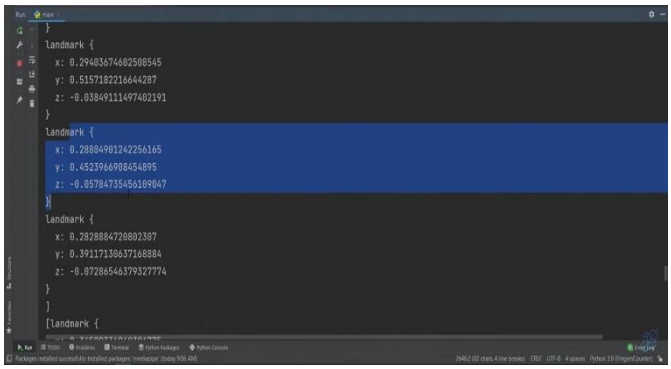

Fig -4: Hand Landmark Co-ordinates

As pixel coordinates calculated above are with respect to frame's dimensions we need to scale them to screen's

dimensions by taking into consideration the layout of screen's pixel coordinates .

x-coordinate of pixel=(screen-width)/(frame-width)*cx

y-coordinate of pixel=(screen-height)/(frame-height)*cy

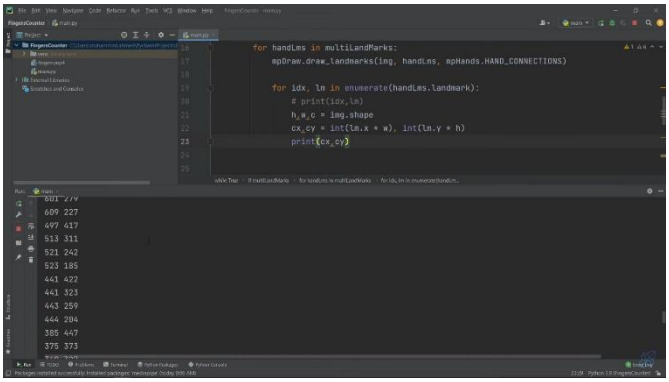

Fig -5: Scaled Hand Landmark Co-ordinates

9. ARCHITECTURE

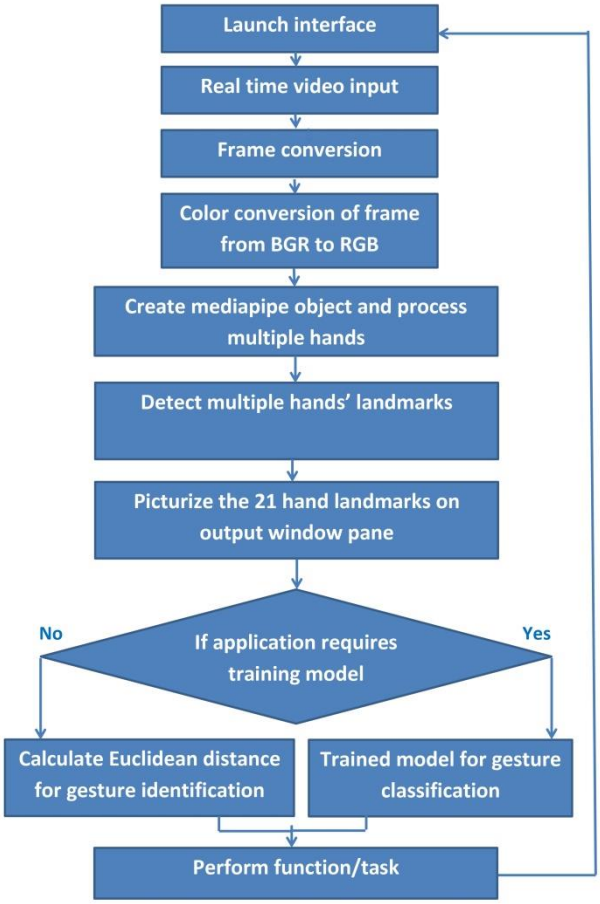

Fig -6: Architecture

10. ACCURACY

We are first defining the gestures that our system is intended to recognize. After which we collected data by recording videos of us performing pre-defined gestures under variety of conditions like ideal conditions, uneven lighting conditions, and with different camera angels.

Under ideal conditions:

For a specific alphabet, out of 10 predictions 8 were TP, 2 were FP so,

Precision=TP/(TP+FP) = 8/10=80%

Where,

TP=True Positive

FP=False Positive

It contains an inherent error as there is a limitation to the measurement of distance between tip of index finger and thumb due to which volume can be reduced to zero but incremented to only 95% of total volume.

Accuracy= (expected output-actual output)/expected output*100% = (100-95)/100*100 = 5%.

Application can achieve expected output with 95% accuracy and relative error is 5%

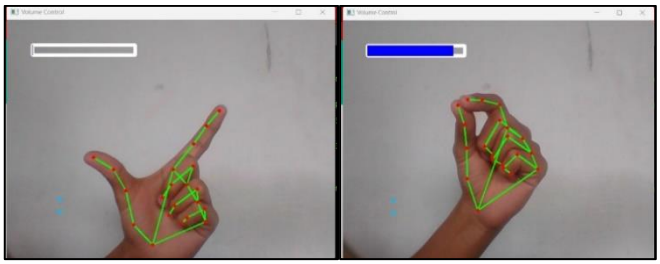

Fig -7: Left-Volume reduced to 0%, Right-Volume increased to 95%

Under ideal conditions the precision is 100% and under non-ideal conditions like uneven lighting and camera angle the precision reduces to 60% as for 10 predictions 6 were correct.

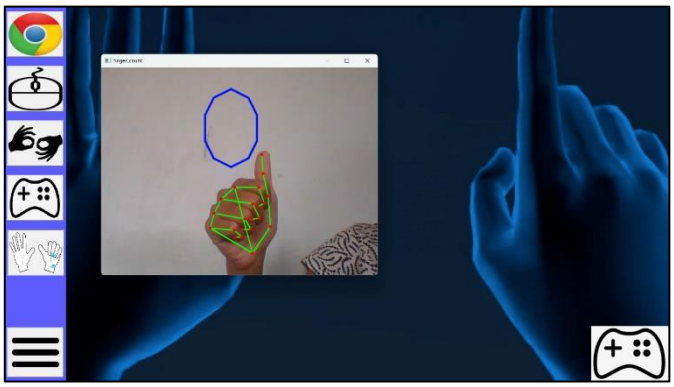

Fig -8: FP due to camera angle

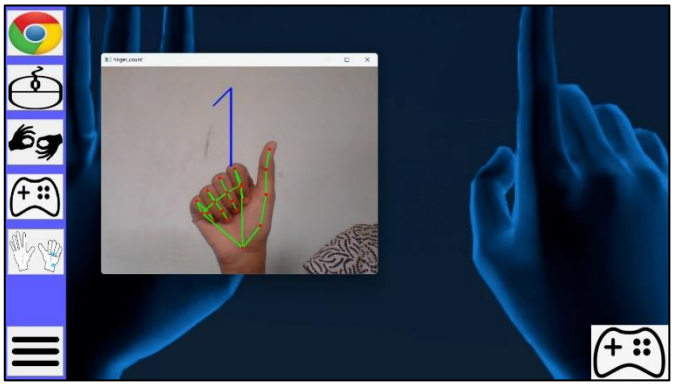

Fig -9: *TP under ideal conditions*

11. CONCLUSIONS

A system containing virtual mouse to navigate, virtual keyboard to type, hand sign detection for verbally challenged people, text to speech converter, gaming, volume control and zoom-in/zoom-out, handling files, taking selfies, drawing is developed and is controlled using hand gestures and computer vision. It enables users to perform basic computing activities with ease. It minimizes the dependency on input/output devices like keyboard and mouse. It provides security against keyloggers, helps verbally challenged and visually impaired people. It is user friendly and has a huge scope in sectors like healthcare, industries, teaching.

ACKNOWLEDGEMENT

Without the outstanding assistance of our project guide, Prof. Jithina Jose, and other project organizers, this publication and the research supporting it would not have been possible. From our initial meeting through the final draught of this paper, their passion, expertise, and meticulous attention to detail have inspired us and kept our work on track.

REFERENCES

1. S. R. Chowdhury, S. Pathak and M. D. A. Praveena, "Gesture Recognition Based Virtual Mouse and Keyboard," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), Tirunelveli, India, 2020, pp. 585-589.
2. 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.
3. K. Meena, M. Kumar and M. Jangra, "Controlling Mouse Motions Using Eye Tracking Using Computer Vision," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2020, pp. 1001-1005.
4. A. Enkhbat, T. K. Shih, T. Thaipisutikul, N. L. Hakim and W. Aditya, "HandKey: An Efficient Hand Typing Recognition using CNN for Virtual Keyboard," 2020 - 5th International Conference on Information Technology (InCIT), Chonburi, Thailand, 2020, pp. 315-319.
5. N. Naseeb, M. Alam, O. B. Samin, M. Omar, S. S. Khushbakht and S. A. Shah, "RGB based EEG Controlled Virtual Keyboard for Physically Challenged People," 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2020, pp. 1-5.
6. S. J. Oh, R. Benenson, M. Fritz and B. Schiele, "Person Recognition in Personal Photo Collections," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 1, pp. 203-220, 1 Jan. 2020.
7. H. Cui and Y. Wang, "Research on Gesture Recognition Method Based on Computer Vision Technology," 2020 International Conference on Computer Information and Big Data Applications (CIBDA), Guiyang, China, 2020, pp. 358-362.
8. X. Chu, J. Liu and S. Shimamoto, "A Sensor-Based Hand Gesture Recognition System for Japanese Sign Language," 2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech), Nara, Japan, 2021, pp. 311-312.
9. A. Zanzarukiya, B. Jethwa, M. Panchasara and R. Parekh, "Assistive Hand Gesture Glove for Hearing and Speech Impaired," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), Tirunelveli, India, 2020, pp. 837-841.
10. L. Abraham, N. S. Mathew, L. George and S. S. Sajan, "VISION-Wearable Speech Based Feedback System for the Visually Impaired using Computer Vision," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), Tirunelveli, India, 2020, pp. 972-976.