Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA 2022)
IEEE Xplore Part Number: CFP22N67-ART; ISBN: 978-1-6654-9707-7

Virtual Mouse to Enhance User Experience and Increase Accessibility

1st S. Vasanthagokul
Dept. of Computer Science & Engineering
Sri Ramakrishna Engineering College
Coimbatore, India
https://orcid.org/0000-0003-0199-9189

2nd K. Vijaya Guru Kamakshi
Dept. of Computer Science &Engineering
Sri Ramakrishna Engineering College
Coimbatore, India
https://orcid.org/0000-0002-5488-5372

3rd Gaurab Mudbhari
Dept. of Computer Science &Engineering
Sri Ramakrishna Engineering College
Coimbatore, India
https://orcid.org/0000-0001-9961-7366

4th T. Chithrakumar
Dept. of Computer Science & Engineering
Sri Ramakrishna Engineering College
Coimbatore, India
https://orcid.org/0000-0001-5819-0984

Abstract— **A User Interface is a form of establishing a platform in which a human interacts with machines. There have been different types of user interfaces evolving over the period in line with the rapid growth of technology. Some of the most popular input devices used are mice, keyboards, touchscreens and styluses. A graphical interface is user friendly and, as a result, is widely used. Systems of contactless communication surfaces for interactions have been introduced to decrease the spread of germs and combat diseases like covid-19. This system also can be utilized by disabled people who still retain motor function in the hand and forearm. This paper put forward an AI-assisted virtual mouse system where these drawbacks are solved by utilizing a webcam/built-in camera for recording the motions of the hand and translating them into mouse actions via ML algorithms. The mouse actions are performed based on hand gestures, which are used to control the computer virtually. The hand detection algorithm is based on a deep learning model. So, the proposed system can reduce the spread of germs and help a computer be more accessible to people with special needs.**

Keywords—Hand Gesture, Graphical User Interaction, MediaPipe, Gesture Recognition

I. Introduction

There have been significant advances in Human-Computer Interaction in the past decade, especially in AR and VR. Emerging hardware is becoming more compact with each new version that comes out. It has also become essential to design a very natural system that is also easy to understand. This includes steps like designing layouts, widgets, menu systems, and windows, which all significantly impact the usability of an application [1]. This paper proposes a system that uses Artificial Intelligence to control the computer or device using image processing virtually. The focus of the proposed method is to be able to execute the basic actions of a mouse cursor virtually. With the help of ML algorithms, one can track the tip of the fingers, and the gesture made [2].

While using other devices like a wireless mouse, a person would have to worry about batteries and or rechargeable internal batteries or dongles that connect to the computer, but in this paper, the webcam captures the image and is processed by the computer, which is then given commands based on the gesture shown.

For developing the proposed system, OpenCV, a computer vision library, is used. This system was implemented using Python. In the proposed method, the package MediaPipe is used to track the hand and the tips of the fingers. Autopy and Pyautogui libraries were used for the motion of the cursor and performing functions like left-clicking and right-clicking Excellent results and high accuracy were observed. The proposed system can work very well in real-time.

The system put forward is used to solve the difficulties that one faces in real life when the usage of a mouse is problematic because of space constraints, where using a mouse would ruin the aesthetics of the surroundings and for people who have difficulties with their hands, which keeps them from using mice. Also, to reduce the transmission of germs via common surfaces. The proposed system can solve these problems since a webcam is utilized to recognize the hands and the tips of the fingers and interpret them into gestures.

II. Related Works

Using hand gestures to control the cursors is done in many ways. However, a few established pieces of research have been done previously on virtual mouse control.

In 2021, Surya Narayan Sharma and Dr A Rengarajan published a paper, “Hand Gesture Recognition using OpenCV and Python,” explaining real-time hand tracking and gesture recognition. Hand gestures are recognized even in low-intensity environments. First, depth images are obtained, and disparity maps and masking are generated. Then a polygon approximation is performed after nearest contour extraction. They used RCB and IR when lights were ON and only IR bitstream when lights were OFF [3].

In 2017, D. K. Vishwakarma and V. Grover published a paper, “Hand gesture recognition in low-intensity environment using depth images,” which proposed a method of capturing input with a camera and preprocessing through which the Region of Interest is identified. Now the hand region is segmented, and the gesture is recognized using contour extraction, and the convex hull is found along with the defects in convexity, so depending on the defects, the gesture is recognized [4].

In 2020, Nikhil, Chinnam & Rao, Chukka & Prince, Brumancia & Kaliannan, Indira & Anandhi, T. & Ponnupillai, Ajitha proposed a study on “Finger recognition and gesture based virtual keyboard”, where they have used hand gestures to operate a virtual keyboard. In this, preprocessing like segmentation and morphological shifting is done. The feature extraction is made by finding the contours and then adjusting the curved frame and numerical operations on contours [5].

In 2018, Abhilash S S, Lisho Thomas, Naveen Wilson and Chaithanya C proposed a study on “Virtual mouse using hand gesture”, where they presented a system to drag and drop files on a computer screen by using computer vision. They utilized red tape on the tip of the hand and filtered the output from the camera to extract the hand's position on the screen [6].

In 2021, Dinh-Son Tran, Ngoc-Huynh Ho, Hyung-Jeong Yang, Soo-Hyung Kim and Guee Sang Lee proposed a study on "Real-time virtual mouse system using RGB-D images and fingertip detection," where they had used a Kinect sensor v2 and a webcam. First, they utilized a border tracing algorithm to extract the position of the centre of the hand. Then a K-cosine algorithm is utilized to identify finger-tips, and all of this is done at 30 frames per second [7].

In 2019, H. -Y. Chung, Y. -L. Chung and W. -F. Tsai proposed a study on "An Efficient Hand Gesture Recognition System Based on Deep CNN", in which they have developed two deep CNN architectures to recognize the hand gesture. Furthermore, they have done background subtraction to detect the ROI and used a kernelized correlation filter, and the images are resized to fit into the deep CNN model that is developed [8].

In summary, most of the existing systems use expensive IR sensors or RGB cameras or a combination of these two to arrive at the location of the hand in a given frame. Some publications implemented an image filter, which performs colour based filtering. Coloured tape on the tip of the finger, in combination with the previously mentioned filter, allows one to locate the position of a finger-tip in a given image. One publication has implemented a way to simultaneously perform training and predict the location of the hand in a frame. This type of prediction is very resource-intensive and cannot be deployed when limited hardware is present. The system aims to solve the problem of cost and ease of deployment.

III. Algorithms used for tracking hands and tips of fingers

A. *OpenCV*

OpenCV is a computer vision python library used to process images using complex machine learning algorithms. Object detection can also be done using OpenCV [9]. Computer vision applications that perform in real-time are developed using this library. It is also used to process images and video feeds. Some examples are face detection and object detection.

B. *MediaPipe*

It is a system which is utilized to pipeline ML ventures, and it is an open-source venture of Google [10]. This framework was built upon time series data which makes it applicable across many platforms of development. It is a multimodal framework and can be applied to images and videos [11]. A pipeline (Fig. 1) contains different parts that are named calculators, in which data flows as packets through each stream which is connected to each calculator. These streams flow from one calculator to another in a similar way to a directed graph [12]. The creation of custom applications by defining or replacing calculators is possible by the developer. They are reusable. Each calculator is a node and is connected by streams. These calculators can perform various tasks like transforming images into tensors or identifying hand landmarks, as shown in Fig. 1.

MediaPipe uses a Single Shot Detector (SSD) and a Hand Landmark Model [13]. For identification and interpretation, the SSD neural network is utilized [14]. The first step is to train the model for palm recognition by utilizing non-maximum curtailment as it works greater on smaller targets like the palm and the fist. The training is done by using anchor boxes in the training set. This model works by identifying the regions where there is a high probability of an object being found, in this case it is a palm, and it suppresses most of the predicted locations of objects which have high overlap with the prediction with highest probability. The cropped image of the identified palm is fed to the Hand Landmark Model. The pre-trained landmark model is made up of 21 knuckle/joint locations as shown in Fig. 2. The model was previously trained on approximately 30,000 images. These images consisted of real images and synthetic images. Landmark lines and points were annotated manually or high-quality synthetic renders of a hand model were used. Depth maps were used when they were available. This mix of images was used to combat poor generalization if only synthetic images were used.

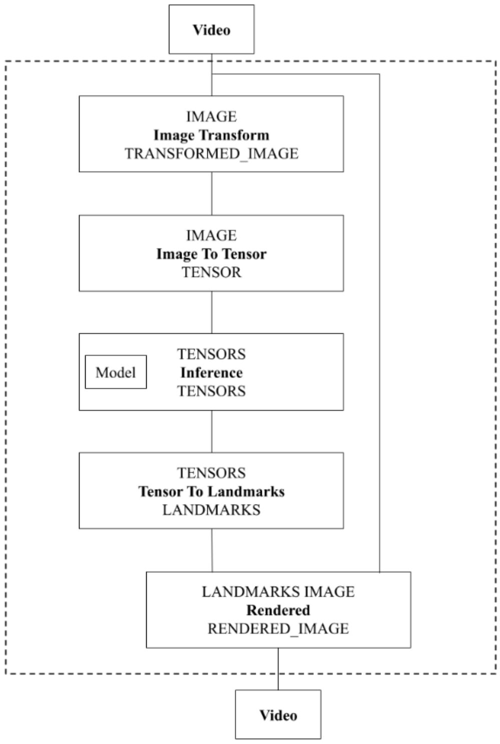

Fig. 1. MediaPipe Framework

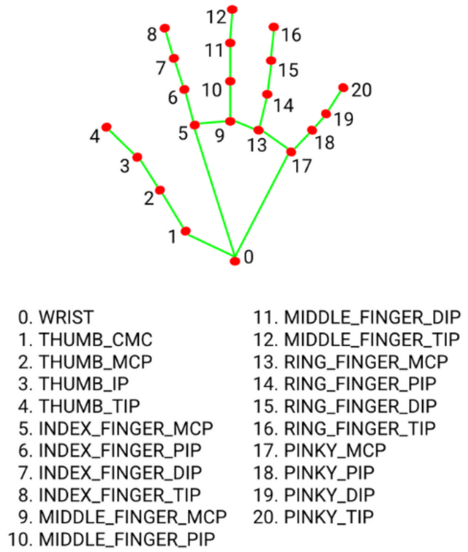

Fig. 2. Hand Landmarks

Source: Adapted from [15]

IV. Methodology

The flow diagram explains the functions performed using the system in various conditions, which are defined clearly in Fig. 3.

A. Hardware and Software Used

A laptop possessing an intel i7 7th generation processor with 8 GB of ram and running on Ubuntu 20.04 LTS OS was used in the deployment of the system put forward. A 720p webcam is utilized. Python version 3.7.2 was used to run the system put forward. In reference to the section on related works, the system utilized here is far simpler to set up when compared to existing solutions.

B. Webcam utilized by the implemented system

A webcam captures images frame by frame in the implemented system. One of the image processing libraries of Python named OpenCV is used in this project where it creates the video capture object. Firstly, the webcam captures the frame and passes it to the AI assisted virtual mouse system. This is done by invoking the webcam capture API present in OpenCV. This video from the webcam is captured in a series of images, i.e., frames. This is cheaper alternative when compared to IR cameras.

C. The Process of Capturing and Processing the Video Stream

Frames are captured until the program stops executing. After that, the frames proceed from the BGR colour space to the RGB colour space. This helps to find the palms captured in the frame, as depicted in Fig. 2.

D. Drawing Rectangle and Matching with Screen for Mapping

Finger tips are captured through the device's web camera; the computer screen is being controlled using the transformational algorithm in the MediaPipe library. After that, the system detects the hand and assigns a function to the finger. Finally, a rectangular box will be created in the screen of the webcam region, which will be movable throughout the screen by the use of a cursor.

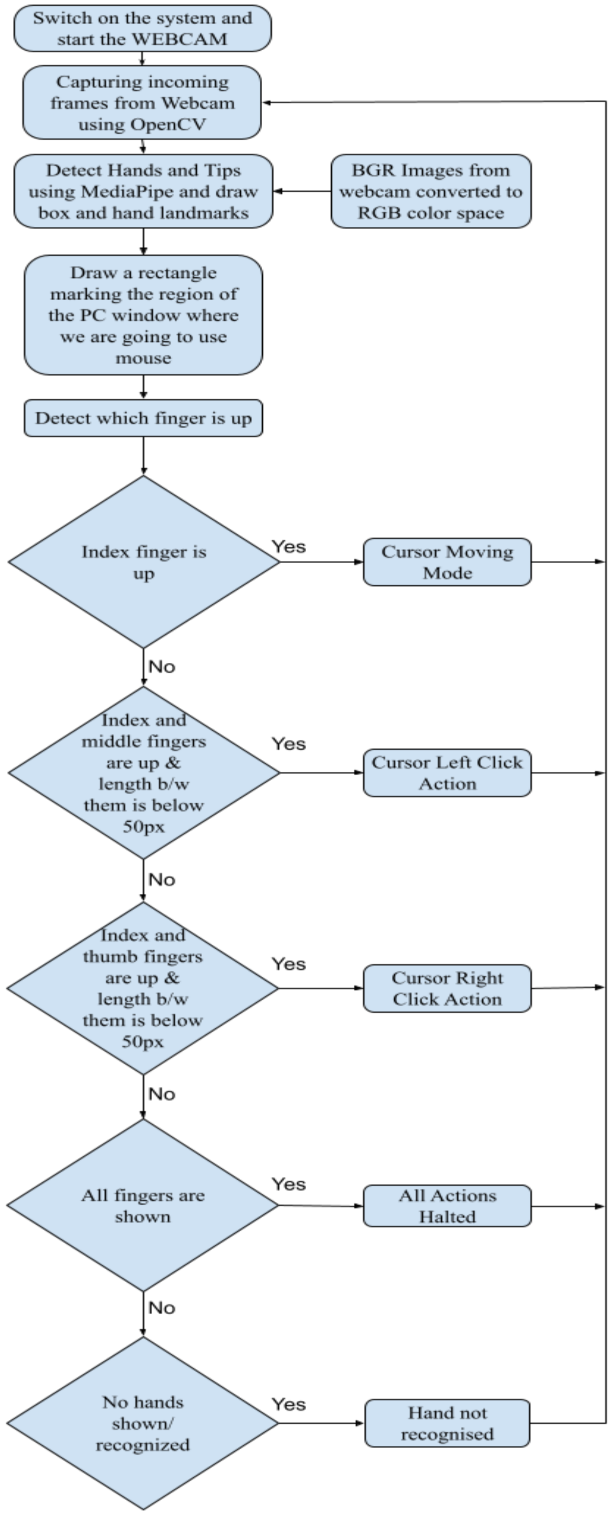

Fig. 3. Flow diagram of the implemented system

E. The detection of which fingers are raised and performing the corresponding mouse function

At the current level, the tip IDs that are bound with the raised fingers are detected using the MediaPipe framework, as shown in Fig. 1 and Fig. 2, and based on it, the specific function is performed. The tip IDs correspond to different joint locations in the hand. Each ID has a corresponding X and Y coordinate relative to the screen. The relative positions to one another are retrieved using this X and Y coordinate. It can be found which joint location is above another joint location in the X-Y plane. The information regarding which finger is raised is identified using this.

F. Computer vision is used to execute mouse functions based on the gestures performed by the hand and being interpreted by utilizing the bound tip IDs in conjunction with machine learning algorithms

1) Cursor moving mode: If the index finger with tip ID 8 is raised and all other fingers are not raised. The Python autopy package moves the mouse pointer around the screen. It is depicted in Fig. 4.

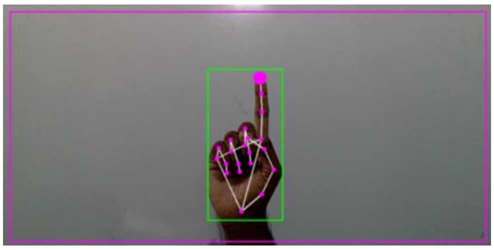

Fig. 4. Cursor moving mode

2) Cursor Left Click mode: If fingers with tip IDs 8 and 12 are raised, if the space between the tips of the finger is less than 50 pixels, the autopy package is made used to execute the function of the left click of a mouse. It is depicted in Fig. 5 and Fig. 6.

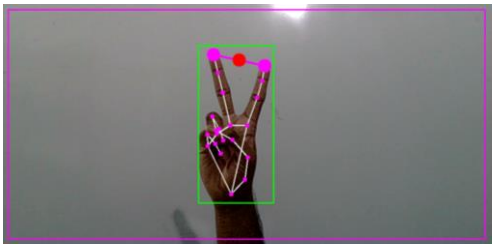

Fig. 5. Left-click mode – unclicked

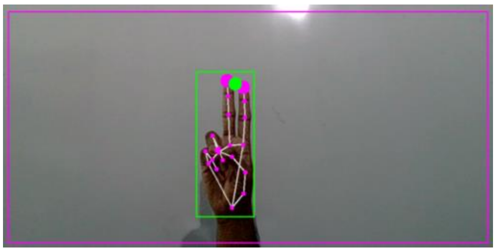

Fig. 6. Left-click mode – clicked

3) Cursor Right Click mode: If fingers with tip IDs 4 and 8 are raised and the space between the tips is less than 50 pixels, the autopy package is made to execute the function of the right click of a mouse. It is depicted in Fig. 7 and Fig. 8.

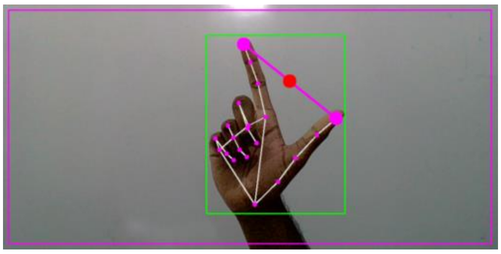

Fig. 7. Right-click mode – unclicked

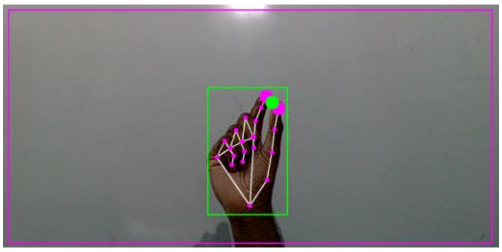

Fig. 8. Right-click mode – clicked

4) Stop Executing All Actions Mode: If fingers with tip IDs 4, 8, 12, 16 and 20 are raised, the system is made to stop executing any action on the screen, as shown in Fig. 9.

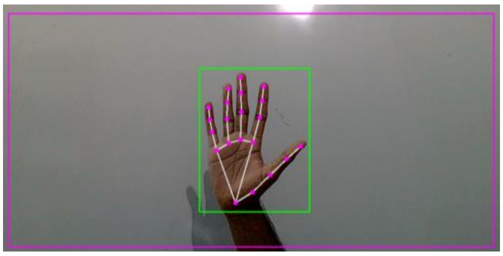

Fig. 9. No action performed

V. RESULTS

A system for human-machine interaction using computer vision has been delivered. Various lighting conditions and background conditions at different distances have been used in the testing process. Three people each had 25 tries to test the system. This resulted in 300 different gestures, which were labelled manually. The results are depicted in Table 1. The model used here has an overall accuracy of 99%. The existing models have accuracies lower than this as shown in Fig. 10 and Table 2.

TABLE I. MODEL ACCURACY

Hand Gesture	Mouse Action Executed	Evaluation		
		Success	***Failure***	***Accuracy (%)***
Only when the index finger is raised	Cursor movement	75	0	100
Only when the thumb and index finger are raised	Right Click	72	3	96
Only when the index and middle finger are raised.	Left Click	74	1	99
All the fingers are raised	No action performed	75	0	100
Results		296	4	99

TABLE II. ACCURACY COMPARISON

Existing Model	***Accuracy***
Hand gesture using OpenCV & Python	68.80 %
Real time system usinig RGB-D Images	96.13 %
Hand Gesture based on CNN	84.99 %
Model used in this paper	99.0 %

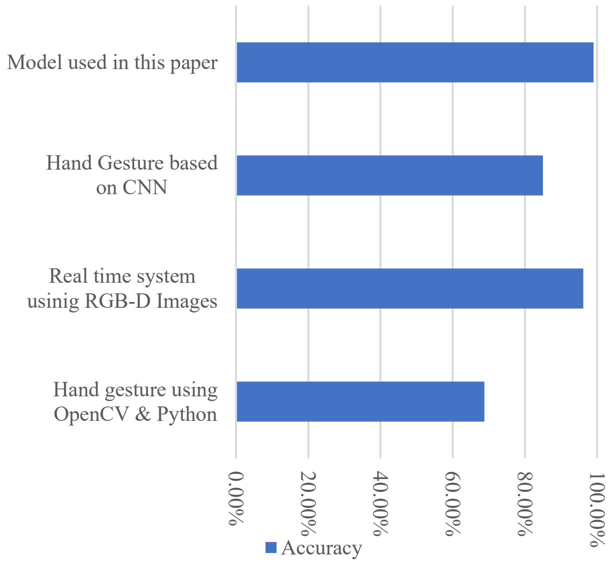

Fig. 10. Comparison of Accuracies

VI. APPLICATIONS

The project hand tracking and controlling computer cursor using image processing has a wide range of applications. Since the project is controlling the cursor with the user's hand, wherever there is the use of the cursor in the system, there is an application for the project; apart from that, some of the major application of the project is listed below.

A. Interesting and sophisticated user interface

Generally, people use a mouse for their computers and other systems. The project will enable users to interact with the system in a contactless and unique way. A virtual mouse is a concept that will take some time to get used to as people have years of muscle memory from using a physical mouse. In places where people want a change in the way they use a mouse in their daily lives, this system can be used in such places

B. Contactless interface

Since COVID-19, many people are avoiding contact with things that are common in use with other people, and mouse is one of them. Not only COVID, but there are also other communicable viruses which transmit from one person to another due to human interactions with the object and the system since it is contactless, that will reduce the probability of getting those kinds of diseases to almost zero. This is possible as the common physical interfaces for transferring harmful bacteria and viruses are removed.

C. For people having strained injury

Almost 40% of strain injury is caused due to the high use of mice daily by the people. Since people have to sit with their arms straight for a longer duration to use a mouse, which causes strain injury, the system is based on the elimination of mice, which will eventually reduce the strain injury due to a mouse.

D. Interactive gaming purpose

There are many interactive games nowadays where there is the use of a mouse which is popular among people, and the system can make these gaming experiences much more interactive and entertaining. Not only on PC gaming but also in virtual reality gaming. If a player doesn't have to use any physical devices and can play with just their hands, it will make their gaming experience far better than the usual one.

VII. CONCLUSION

This AI-Assisted Virtual Mouse System aims to be able to use the machine in a more user-friendly way by using hand-based gestures to interact with a system. The gesture which is captured using the webcam is being processed, and the action is performed accordingly.

By testing the model and evaluating the results, the AI-assisted virtual mouse system performs with great accuracy in various lighting conditions and performs consistently. Thus one can use the system in places where a mouse would not be desirable or practical. The system can be used to reduce the spread of germs and other harmful germs as the system operates using virtual gestures and does not have contact with any physical mouse or surfaces.

VIII. FUTURE SCOPE

Though a sophisticated system where communication between the user and the hardware system through the mouse has been created virtually, there are still some limitations in this project that will be worked on in the future. For example, in low light, the accuracy of the program may fall. For this, a training set for the model for low-light conditions has to be created, and there is still some difficulty in dragging and clicking with high precision. Nevertheless, this is a very big field of AI, and with the success of the current module, experimenting with similar but different interfaces can be made possible, like a virtual keyboard.

REFERENCES

[1] A. Oulasvirta, N. R. Dayama, M. Shiripour, M. John and A. Karrenbauer, "Combinatorial optimization of graphical user interface designs," IEEE, vol. 108, pp. 434-464, 2020.

[2] K. H. Shibly, S. Kumar Dey, M. A. Islam and S. Iftekhar Showrav, "Design and development of hand gesture based virtual mouse," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), pp. 1-5, 2019.

[3] Surya Narayan Sharma, Dr. A Rengarajan, "Hand gesture recognition using opencv and python," International Journal of Trend in Scientific Research and Development, Vol. 5, Issue-2, pp. 346-352, 2021.

[4] D. K. Vishwakarma and V. Grover, "Hand gesture recognition in low-intensity environment using depth images," 2017 International Conference on Intelligent Sustainable Systems (ICISS), pp. 429-433, 2017.

[5] Nikhil, Chinnam & Rao, Chukka & Prince, Brumancia & Kaliannan, Indira & Anandhi, T. & Ponnupillai, Ajitha, "Finger Recognition and Gesture based Virtual Keyboard," 5th International Conference on Communication and Electronics Systems (ICCES), pp. 1321-1324, 2020.

[6] Abhilash S S, Lisho Thomas, Naveen Wilson, Chaithanya C, "Virtual mouse using hand gesture," International Research Journal of Engineering and Technology (IRJET), Vol. 5, Issue 4, pp. 3903-3906 2018.

[7] Dinh-Son Tran, Ngoc-Huynh Ho, Hyung-Jeong Yang, Soo-Hyung Kim, Guee Sang Lee, "Real-time virtual mouse system using RGB-D images and fingertip detection," Multimed Tools Appl. 80, 10473–10490, 2021.

[8] 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), pp. 853-858, 2019.

[9] Alberto Fernández Villán, "Mastering OpenCV 4 with Python: A practical guide covering topics from image processing, augmented reality to deep learning with OpenCV 4 and Python 3.7," USA, Packt Publishing Limited, 29 March 2019.

[10] Google, "On-Device, Real-Time Hand Tracking with MediaPipe," Julia Evans. Updated Aug. 19, 2019. [Blog]. Available: https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html, Accessed on: March 15, 2022.

[11] Sung, George & Sokal, Kanstantsin & Uboweja, Esha & Bazarevsky, Valentin & Baccash, Jonathan & Bazavan, Eduard & Chang, Chuo-Ling & Grundmann, Matthias, "On-device Real-time Hand Gesture Recognition," 2021.

[12] En.wikipedia.org. 2022. Directed graph - Wikipedia. [online] Available at: https://en.wikipedia.org/wiki/Directed_graph Accessed on: 2 June 2022.

[13] Wei Liu, Anguelov D, Erhan D, et al, "SSD: Single Shot MultiBox Detector," Springer International Publishing, 2016.

[14] N. M. Krishna, R. Y. Reddy, M. S. C. Reddy, K. P. Madhav and G. Sudham, "Object Detection and Tracking Using Yolo," Third International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 1-7, 2021.

[15] Google, "Hand landmarks," Mediapipe, USA, Dec. 10, 2020. [Online]. Available: https://github.com/google/mediapipe/blob/master/docs/images/mobile/hand_landmarks.png, Accessed on: March 15, 2022.