IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# Virtual Mouse & Keyboard Using Hand Gesture

Prof. Jyotsna Nanajkar Professor, Department of Information Technology, Zeal College of Engineering and Research, Pune, Maharashtra

Ms Aishwarya Pisal, BE Students, Department of Information Technology, Zeal College of Engineering and Research Pune, Maharashtra.

Ms Arti Patil, BE Students, Department of Information Technology, Zeal College of Engineering and Research Pune, Maharashtra,

Ms Vaishnavi Sathe BE Students, Department of Information Technology, Zeal College of Engineering and Research Pune, Maharashtra,

Ms Prachi Sawale BE Students, Department of Information Technology, Zeal College of Engineering and Research Pune, Maharashtra

*Abstract:* The development of technology is evident in the modern world. Artificial intelligence is a feature of modern machines. This project also uses artificial intelligence in a minor way. This project shows how to use a camera to track finger movements in computer windows and how to use a single finger movement to manage the entire system. With the advancement of augmented reality technology and the gadgets we use on a daily basis, Bluetooth and wireless technology are becoming more and more common.
This article presents an intelligent virtual mouse that uses gestures and hand gestures to perform computer mouse operations using computer vision. The main purpose of this application is to perform computer mouse cursor functions and scrolling functions using a webcam or computer's built-in camera instead of using a traditional mouse. Hand and finger pointing using computer vision according to HCI. With the AI virtual mouse system, we can use the built-in camera or webcam to follow finger movements and play games

***Index Terms*** **-** Mediapipe , Autopy, GUI, gestures, OpenCv, Computer vision, Hand gesture recognization.

## 1. INTRODUCTION

In the era of touchless interaction and automation, traditional input devices like a physical mouse and keyboard are being reimagined through innovative technologies. This project, **"Virtual Mouse & Keyboard Using Hand Gesture"**, aims to develop a system that allows users to control their computer's mouse and keyboard functions using hand movements, without physically touching any hardware
The system uses a webcam to capture real-time video of hand gestures. With the help of computer vision libraries such as OpenCV and advanced hand tracking tools like MediaPipe, the system detects and interprets the user's finger positions and movements. These gestures are then mapped to specific mouse and keyboard actions using libraries like PyAutoGUI.

## WHAT IS VIRTUAL MOUSE?

The Virtual Mouse provides an infrastructure between the user and the system using only a camera. It allows users to interface with machines without the use of mechanical or physical devices, and even control mouse functionalities

• User makes specified hand gestures that is captured by the camera.
•Object recognition techniques are used to extract information from the capture.
•This is then translated to some meaningful event on the screen.
• Overcomes the use of physical devices.

### 1.1 MAIN PROBLEMS IN THE SUBJECT AREAS -

Hand gesture recognition can sometimes misinterpret gestures due to variations in lighting, background, or skin tone. Small errors in gesture detection can lead to inaccurate cursor movements or incorrect command execution. The system needs to interpret gestures in real time with minimal latency for a smooth and responsive user experience.
Processing video input or sensor data rapidly and efficiently requires optimized algorithms and sufficient computational power. Users need time to adapt to a new interaction paradigm that differs from traditional mouse/keyboard input. Designing intuitive and accessible gesture sets is essential to minimize this learning curve.

## 2.RELATED WORK-

In recent years, several researchers have worked on hand gesture-controlled virtual mouse and keyboard systems to enhance human-computer interaction. Early methods relied on hardware like sensor gloves and colored markers, which were uncomfortable and inaccurate. Later, computer vision-based techniques using simple webcams became more prominent. Frameworks like MediaPipe, combined with libraries such as OpenCV and NumPy, enabled real-time hand and finger tracking without additional sensors. Hand gestures were used for operations like cursor movement, clicking, scrolling, and volume control. Some models integrated voice assistants to improve accessibility. The proposed system improves on earlier models by eliminating the need for external devices and providing a lightweight, fast, and cost-effective solution. MediaPipe's hand landmark model allows accurate fingertip detection, while OpenCV handles video processing efficiently. Compared to older systems, this approach is easier to use, more scalable, and works well under various lighting and background conditions, making it suitable for real-world applications.
In vision-based systems, researchers started using frameworks like MediaPipe developed by Google, combined with OpenCV for image processing. MediaPipe provides a powerful hand landmark detection model, capable of identifying 21 key points on a hand in real-time. These landmarks help in recognizing static and dynamic gestures without the need for any wearable device. OpenCV helps capture live frames, process them, and integrate them with gesture-based logic for cursor movement, clicks, scrolls, and keyboard-like typing. In conclusion, the shift towards lightweight, AI-based gesture recognition systems like the one proposed — using MediaPipe, OpenCV, and simple Python libraries — provides a highly accessible and cost-effective alternative to traditional input devices, enhancing Human-Computer Interaction (HCI) for both general users and people with disabilities.

## 3.Methodology

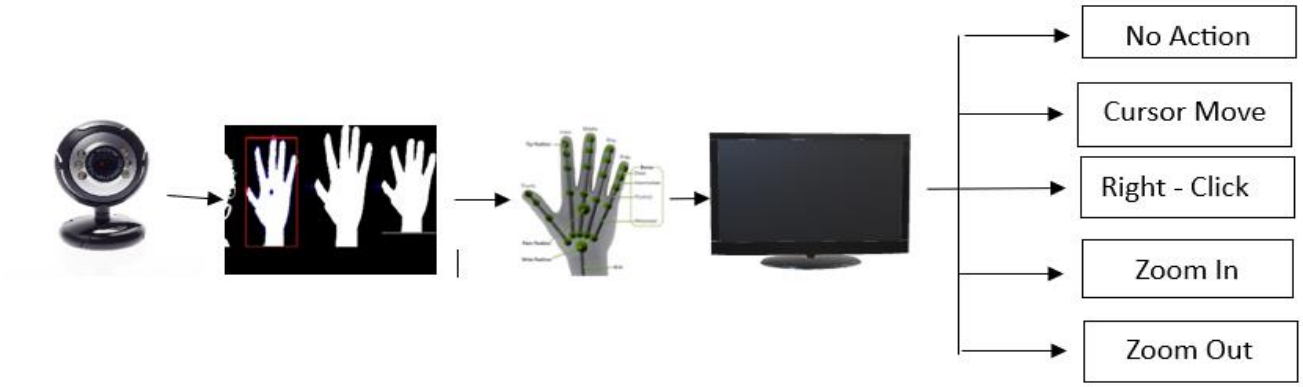



**Fig 3.1. System Model**

### 3.1 Camera Settings

The runtime operations are managed by the webcam of the connected laptop or desktop. To capture a video, we need to create a Video Capture object. Its argument can be either the device index or the name of a video file. Device index is just the number to specify which camera. Since we only use a single camera we pass it as '0'. We can add additional camera to the system and pass it as 1, 2 and so on. After that, you can capture frame-by-frame. But at the end, don't forget to release the capture. We could also apply color detection techniques to any image by doing simple modifications in the code

### 3.2 Capturing frames

The infinite loop is used so that the web camera captures the frames in every instance and is open during the entire course of the program. We capture the live feed stream, frame by frame. Then we process each captured frame which is in RGB (default) color space to HSV color space. There are more than 150 color-space conversion methods available in OpenCV. But we will look into only two which are most widely used ones, BGR to Gray and BGR to HSV.

### 3.3 Display the frame

The imShow() is a function of HighGui and it is required to call the waitKey regulerly. The processing of the event loop of the imshow() function is done by calling waitKey. The function waitKey() waits for key event for a "delay" (here, 5 milliseconds). Windows events like redraw, resizing, input event etc. are processed by HighGui. So we call the waitKey function, even with a 1ms delay.

### 3.4 Mouse Movement

We have to first calculate the center of both detected red object which we can easily do by taking the average of the bounding boxes maximum and minimum points. now we got 2 co-ordinate from the center of the 2 objects we will find the average of that and we will get the red point shown in the image. We are converting the detected coordinate from camera resolution to the actual screen resolution. After that we set the location as the mouse_position. but to move the mouse pointer it will take time. So we have to wait till the mouse pointer reaches that point. So we started a loop and we are not doing anything there we are just waiting will the current mouse location is same as assigned mouse location. That is for the open gesture.

### 3.5 Clicking

The next step is to implement the close gesture. The operation is performed by clicking the object and dragging it. It is similar to the open gesture, but the difference is we only have one object here so we only need to calculate the center of it. And that will be placed on the location where we will position our mouse pointer. Instead of mouse release operation we will be performing a mouse press operation.

### 3.6 Drag

In order to implement the dragging we introduce a variable 'pinchflag'. It will be set to 1 if it was clicked earlier. So after clicking whenever we find the open gesture we check if the pinch flag is set to 1. If it is set to one then Drag operation is performed otherwise the mouse move operation is performed.

## 4. ALGORITHMS AND TOOLS

**1) Mediapipe** : It is a cross-platform framework that's mostly used for building multimodal pipeline in machine learning, and it's an open-source framework by Google. MediaPipe framework is most useful for cross-platform development work since this framework is constructed by statistic data. The MediaPipe framework works as multimodal, which implies this framework is applied to face detection, facemesh, iris scanner, pose detection, hand detection, hair segmentation object detection, motion tracking and objection. The MediaPipe framework is that the best option to the developer for building, analysing and designing the systems performance in the form of graphs, and it's also been used for developing various application and systems within the cross-platform (Android, IOS, web, edges devices). The involving steps in our proposed system uses MediaPipe framework as pipeline structure configuration. This pipeline structure create and run in various platforms which allowing scalability in mobile and desktop system also. The MediaPipe package gives us three reliability, they're like performance, evaluation, and creation which is retrieving by the sensor data, and using a set of components. Mediapipe processing inside a graph which is defines flowing of packet between nodes.

**2) PyAutoGUI :** PyAutoGUI lets your Python scripts control the mouse and keyboard to automate interactions with other applications. The API is designed to be simple. PyAutoGUI works on Windows, macOS, and Linux, and runs on Python 2 and 3. PyAutoGUI has several features:

- Moving the mouse and clicking in the windows of other applications
- Sending keystrokes to applications (for example, to fill out forms).
- Take screenshots, and given an image (for example, of a button or checkbox), and find it on the screen.
- Locate an application’s window, and move, resize, maximize, minimize, or close it (Windows-only, currently).
- Display alert and message boxes

**3) OpenCV**:

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being an Apache 2 licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million.

**4) Numpy:**

• **POWERFUL N-DIMENSIONAL ARRAYS** -Fast and versatile, the NumPy vectorization, indexing, and broadcasting concepts are the de-facto standards of array computing today.

• **NUMERICAL COMPUTING TOOLS** -NumPy offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more.

• **INTEROPERABLE** - NumPy supports a wide range of hardware and computing platforms, and plays well with distributed, GPU, and sparse array libraries.

• **PERFORMANT**- The core of NumPy is well-optimized C code. Enjoy the flexibility of Python with the speed of compiled code.

• **EASY TO USE** -NumPy’s high level syntax makes it accessible and productive for programmers from any background or experience level.

• **OPEN SOURCE**- Distributed under a liberal BSD license, NumPy is developed and maintained publicly on GitHub by a vibrant, responsive, and diverse community.

## 5. Architecture

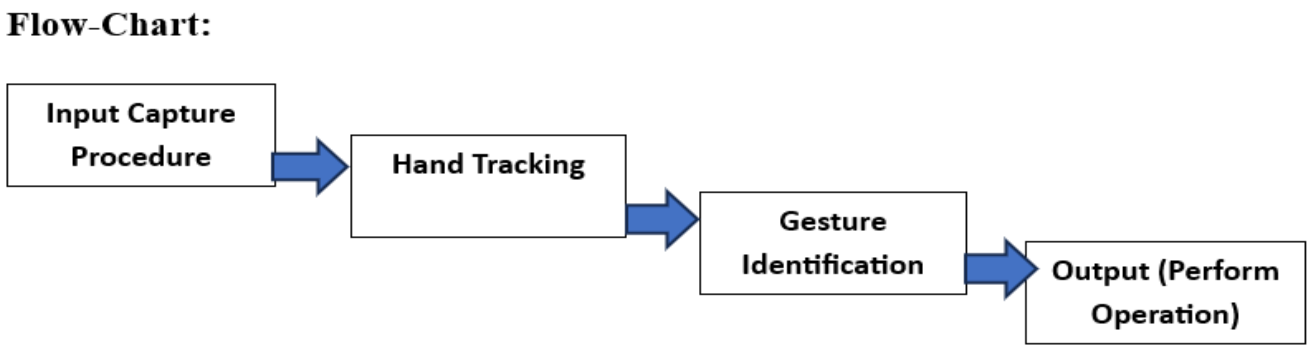



**Fig 5.1 Architecture Of Hand Gesture Mouse Control**

**Step 1- Image Frame Acquisition:**
The initial step in implementing the Hand Gesture Mouse Interface is image frame acquisition. This involves capturing frames from the webcam in real-time to gather data on hand gestures performed by the user.

**Step 2 - Hand Gesture:**
In the interaction scenario, an individual executes distinct hand gestures within the proximity of a camera or sensor embedded in a device. These gestures encompass a spectrum of movements, positions, or signs articulated through hand motions. Each gesture carries its unique significance, contributing to the nuanced language communicated to the system.

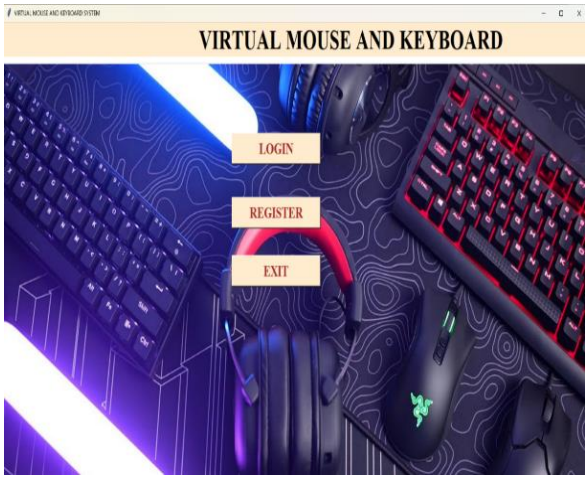
**Step 3 - Image Captured by Device:**
Upon execution, the device's camera swiftly captures either a single image or a sequence of images encapsulating the intricacies of the performed hand gesture. Subsequently, the device's software meticulously processes these images, dissecting the visual data to discern the essence and intent behind the gesture's manifestation.

**Step 4 - Action to be Performed Based on Hand Gestures:**
In this stage of the process, the software embedded within the device undertakes a meticulous analysis of the captured images, aiming to discern and interpret the intricate nuances of the hand gesture presented. Drawing from its programmed algorithms, the software identifies the specific gesture enacted by the user. Depending on the recognized gesture, the device is then primed to execute a predefined action, such as propelling the car forward or applying the brakes with precision and responsiveness.

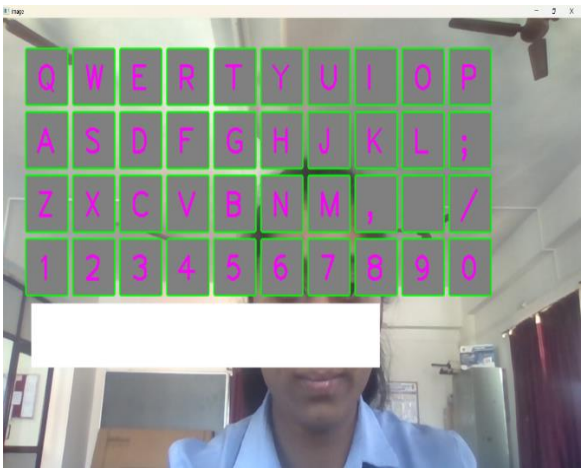
## 6.Results

The virtual mouse system using hand gesture recognition was successfully implemented and tested. The system accurately detected and tracked hand movements in real-time using a standard webcam. Key functionalities such as:

- **Cursor movement** based on hand position
- **Left click** using a specific finger gesture
- **Right click** using another gesture
- **Drag and drop** feature through continuous gesture recognition
- **Smooth cursor motion** using interpolation technique

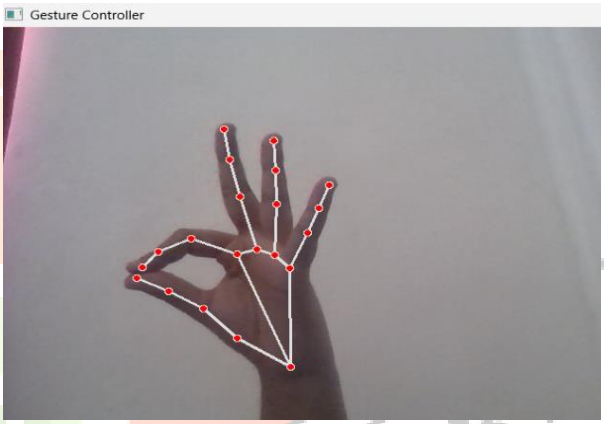



Login Page

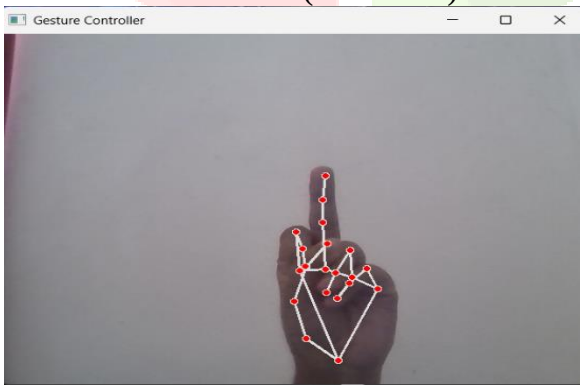



keyboard

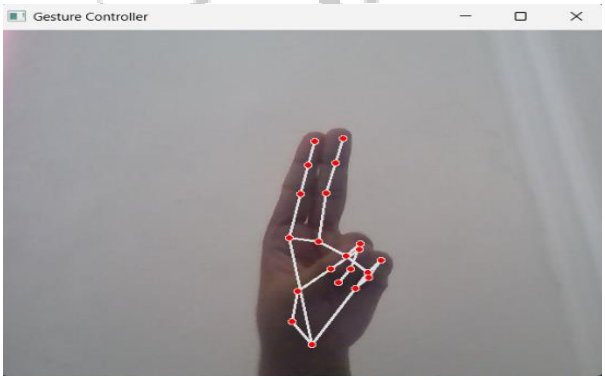

Palm(No Action)



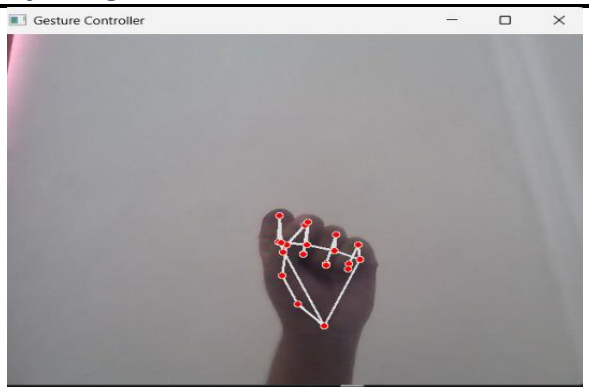

Brightness



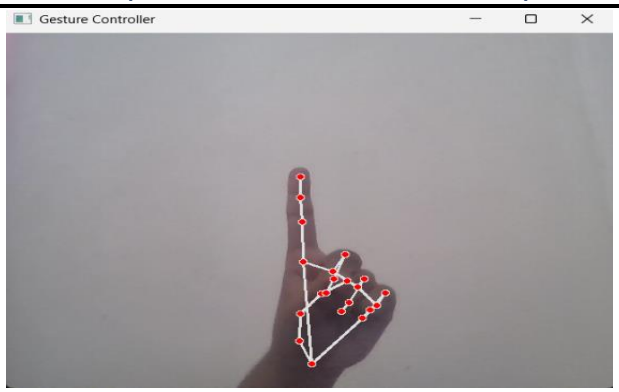

Left Click




Double Click




Drag & Drop




Right Click

## 7.Conclusion

The development of a virtual mouse using hand gesture recognition and keyboard integration showcases the potential of human-computer interaction without traditional hardware. By utilizing computer vision techniques and real-time gesture tracking, users can control the mouse pointer and perform click actions through simple hand movements. This not only offers an innovative and touchless way of interacting with computers but also enhances accessibility for individuals with physical limitations.

The project demonstrates the successful integration of technologies such as OpenCV, MediaPipe, and Python, delivering a low-cost, flexible, and efficient alternative to conventional input devices. With further enhancements in gesture recognition accuracy, latency reduction, and user adaptability, this system could find applications in various fields including gaming, smart homes, presentations, and assistive technologies.

In conclusion, this virtual mouse system serves as a promising step towards the future of intuitive and contactless user interfaces.

## 8. References

[1] Sanka Sarkar, Indrani Naskar Sourav Sahoo Sayan Ghosh, Sumanta Chatterjee “A Vision Base Application For Virtual Mouse Interface Using Hand Gesture” In International Journal Of Innovation Science And Research Technology ISSN-2456-216 Volume.11 pp.1-6. no. 6 November 2021

[2] Kavita R, Janasruti S U, Lokitha S, Tharani G “Hand Gesture Controlled Virtual Mouse Artificial Intelligence” in International Journal Of Innovation Science And Research Technology -ISS(O)-2395-4396 Volume-9 no-2 pp.1-14 2023.

[3] Neerja Arora “Artificial Intelligence Based Virtual Mouse Using Hang Gestures” Journal of Computer Science Volume-2 no-2 July December-2023 pp.1-4

[4] Meenachi R., Nandan C, Swaroop H G, Varadharaju S “Virtual Mouse Using Hand Gesture” in IJCSPUB ISSN-2250-1770 vol-13 pp.1-5, May 2023

[5] Changhyun Jeon, Oh-Jin Kwon, Dongil Shin, And Dongkyoo Shin “Hand-Mouse Interface Using Virtual Monitor Concept for Natural Interaction” IEEE Volume-5 pp.1-8 2017.

[6] Jungpil Shin1, (Member, IEEE), And Cheol Min Kim “Non-Touch Character Input System Based on Hand Tapping Gestures Using Kinect Sensor” IEEE Volume-5 no-1 pp.-10 May 2017.

[7] Rares Pogoreanu and Radu Gabriel Bozomitu “Wireless Gyro-mouse for Text Input on a Virtual Keyboard” IEEE Volume-10 pp.1-4 2022.

[8] Prithvi J, S Shree Lakshmi, Suraj Nair and Sohan R Kumar and “Gesture Controlled Virtual Mouse with Voice Automation” International Journal of Engineering Research & Technology ISSN: 2278-0181 Volume-12 no-4 pp. 1-4 April-2023.

[9] G N Srinivas, S Sanjay Pratap, V S Subrahmanyam, K G Naga Priya, A Venkata Srinivas Rao “Virtual Mouse Control Using Hand Gesture Recognition” IRJET ISSN-2395=0056 Volume-10 no-02 pp.-1-5 Feb-2023.

[10] Aabha Waichal, Mauli Gandi Srushti Bhagwat, Amruta Bhanji Shalaka Deore, Shubhangi Ingale “Hand Gesture Recognition Based Virtual Mouse Using CNN” International journal of Computer Science 0975-8887 Volume-184 No-20 pp. 1-5 July-2022.

[11] Akshay Krishan, Ann Treese Raphi, Arjun Anirudha, Meera George "AI Virtual Keyboard For Typing" International Journal of Engineering Research & Technology Volume-11 no-04 pp.1-5 2023.
[12] G M Trupti, Chandhan kumar, Dheeraj P, Vilas, Prasanna Kumar. S. Shivaraddi "Virtual Mouse Using Hand Gesture" International Journal of Advanced Research in Computer and Communication Engineering SSN (O) 2278-1021 Volume 13, no- 4, pp.1-5 April 2024.
[13] Sneha Thorat, Prajakta Suryawanshi, Kalyani Mate, Kavita Joshi "Virtual Keyboard" Dr D.Y Patil Institute of Engineering Management and Research Pune, Volume-8, no-4, Journal of Emerging Technologies and Innovative Research (JETIR), Volume-8, no-4 pp.1-4 April 2021.
[14] Renato de Sousa Gomide, Luiz Fernando Batista Loja, Rodrigo Pinto Lemos, Edna Lúcia Flores, Francisco Ramos Melo, Ricardo Antonio Gonçalves Teixeira "A new concept of assistive virtual keyboards based on a systematic review of text entry optimization techniques" Volume 32, no-2 pp.1-6 March 2016.
[15] Sowjanya M N, Sweatha R, Radha K G, Indresh BS, Kusuma N and Amit Bhupal Pattar "Virtual Keyboard Using Machine Learning" International Journal of Development Research Volume=08, no-1, pp.1-4 18555-18558, pp. 1-4 January-2018.
[16] Ue-Hwan Kim, Sahng-Min Yoo , and Jong-Hwan Kim , Fellow "I-Keyboard: Fully Imaginary Keyboard on Touch Devices Empowered by Deep Neural Decoder" IEEE Transactions On Cybernetics volume-13, no-01 pp.1-13 2019