Gesture Works – One Stop Solution

Tanishq Agarwal, Bennett University, Pransh Gupta, Bennett University, Madhav Aggarwal, Bennett University

***Abstract*—**

In recent years, and especially in the last year, we have seen a shift in technology toward hands-free navigation, particularly during the Covid-19 era. As technology has advanced, there has been a shift from buttons to touch technology, and in recent years, there has been a significant advancement in gesture-controlled mechanisms. These technologies make it possible for even the most inexperienced user to navigate and use everyday technologies such as cell phones, iPads, computers, and so on.

Working on these steps, we are introducing GestureWorks, a touchless solution for all of your navigation needs. Currently, we have added two modules Gesture Controlled Virtual Mouse and OS based Voice Assistant and are looking forward to additional development in the future.

Keywords—

Face Recognition, Gesture-recognition, PC, Voice Assistant, Virtual Mouse, Webcam.

I. INTRODUCTION

As of now, Fourth Industrial Revolution is going on, also called Industry 4.0.

There has been a strong emphasis on automation, the convergence of physical and digital technologies, and the introduction of web 3.0 technologies such as Artificial Intelligence (AI), Deep Learning, Big Data, Cloud Computing, Robotics, Augmented Reality, IoT, and so on.

All of these enhanced digital technologies have begun to integrate totally with our daily activities such as shopping, entertainment, gaming, web search, and advertisements.

Technologies are more reliant on machines, and as a result, there has been progress in interactions with technology through a variety of methods such as gesture recognition, speech recognition, and so on.

In 1970s, Krueger introduced a new type of human computer interaction (HCI)[1][18] called Gesture-based interaction, which is a type of non-verbal / non-vocal communication where the hands or face, can communicate a specific message through movement. Using Human-Computer-Interaction (HCI)(2008)[1] to recognise hand gestures might help achieve the required ease and naturalness. When communicating with others, hand gestures have a significant role to play in transmitting information. There's something for everyone, from simple hand motions to more complex ones.

We can use our hand to point to anything (an object or a person), or we can use other basic hand or hand gestures expressed through physical articulations matched with their syntax & vocabulary, and also sign languages. As a result, individuals may interact more intuitively by employing hand gestures (2017) [17][16][5] as a device and then integrating them with computers.

Additionally, using voice commands leads to a completely hands-free navigation experience in which a user can communicate easily either through hand gestures or voice navigation with minimal hardware contact.

Gesture Controlled Virtual Mouse simplifies human computer interaction by utilising Hand Gestures and Voice Commands. Almost no direct contact is required with the computer. All I/O operations can be controlled remotely via static and dynamic hand gestures, as well as a voice assistant. This project employs cutting-edge Machine Learning and Computer Vision algorithms to recognise hand gestures and vocal instructions, and it works flawlessly without the usage of any additional hardware. It makes use of models such as CNN, which is implemented by Media Pipe and runs on top of pybind11.

It is divided into two modules: One that acts directly on the hands by utilising Media Pipe and is a hand detecting bot, while the second is a voice recognition bot called 'Dragon.'

II. RELATED WORK

Our software is the only hands-free navigation software that integrates various methods of input/navigation such as Hand Gestures, Speech Recognition, and other to be incorporated in the future, but it is the result of a decade of work and advancement in the field of Deep Learning and more specifically gesture recognition, which is the process of turning motions to their representation and then to commands for some purpose.

HAND GESTURE RECOGNITION:

The purpose of hand gesture recognition is to recognise specific hand movements as input and then process and map these motions as output for devices. Hand gesture (2021) [5][17] recognition features are divided into threecategories based on their extraction properties.

1) **Approaches Based on High-Level Features**:

Machine learning is a good place to look for high-level algorithms. These algorithms are concerned with classifying or interpreting a scene in its entirety. Body posture classification, face detection, human movement categorization, item detection and recognition, and so on are some of the features available.

These algorithms are concerned with teaching a system that recognizes or classify something, after which you give it a little more unknown input that it has never seen before, and its job is to either figure out what is going on in the scene or find a region of interest where it able to detect an action that the system has been trained to look for. When anything major happens, a high-level system recognises important locations in the scene.

2) **Low-Level Feature-Based Methodologies**:

Low-level feature recognition algorithms are generally focused with locating related points between pictures or discovering things that categorise as something even vaguely interesting at the most basic level conceivable - things like detecting edges or lines in an image (in addition to finding interesting points). Furthermore, anything directly dealing with pixel intensities or colours is what considered low level.

3) **Approaches Based on 3D Reconstruction**:

To totally achieve the hand impression, use the 3D model of features. According to research, properly segmenting the hand in skin colour requires similarity and high contrast of thebackdrop connected to the hand via structured light in order to bring in 3D depth data. Another employs a stereo camera to monitor several interest spots on the superficies of the hand, resulting in difficulty in handling robust 3D reconstruction, even though data including 3D provides essential information that can aid in the removal of ambiguity.

Furthermore, there are primarily three hand motion recognition systems.

1) **Machine Learning Approaches:**

For dynamic gestures, the results come from a stochastic process and technique based on statistical modelling, such as PCA, HMM, advanced particle filtering, and the condensation algorithm.

2) **Algorithm Approaches:**

Manual collection of encoded conditions and restrictions for use as gestures in dynamic gestures. Galveia(2014)[19] used a polynomial quation of the third degree to determine the dynamic component of the hand movements.

3) **Rule-based Approaches:**

With a set of pre-encoded rules and feature inputs, it's suitable for both dynamic and static gestures. The features of input motions are extracted and compared to the encoding rules for

recognised movements. Gestures are matched with a rule and authorised as recognised gestures.

Media pipe:

[9] Among the different machine learning frameworks available, we used mediapipe in our research. MediaPipe (2006) [2][13][14][9] is a cross-platform open-sourceMachine Learning framework for building multimodal andsophisticated machine learning pipelines. It may be used tobuild cutting-edge Machine Learning Models for applicationslike facial recognition, multihand tracking, item detection andtracking, and many more applications.

Mediapipe Holistic is one of the pipelines that comprises optimised face, hands, and posture components that enable for holistic tracking, allowing the model to identify hand and body positions as well as facial landmarks at the same time. One of the primary applications of MediaPipe (2019)[12][13][14] holistic is the detection of faces and hands and the extraction of critical features for transmission to a computer vision model.

Hand Landmark:

With a 3D hand-knuckle coordinate (Fig 1.), the exact location of 21 critical sites was determined. It is achieved inside the designated hand areas using regression that will immediately deliver the coordinate prediction, which is a model of the handlandmark in Media Pipe.

Each landmark's hand-knuckle has an x, y, and z coordinate, with x and y normalised to [0.0, 1.0] by picture width and height, respectively, and z representing the landmark's depth. At the wrist, you can learn about your ancestor's depth of landmark. The lower the value, the further away the landmark is from the camera.

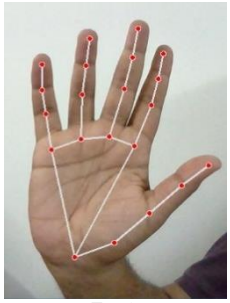


Fig 1.

0. WRIST;1. THUMB_CMC;
2. THUMB_MCP;
3. THUMB_IP;
4. THUMB_TIP
5. INDEX_FINGER_MCP;
6. INDEX FINGER_PIP;7. INDEX_FINGER_DIP
8. INDEX FINGER TIP;
9. MIDDLE_FINGER_MCP;10. MIDDLE FINGER_PIP
11. MIDDLE_FINGER_DIP;12. MIDDLE_FINGER_TIP;
13. RING_FINGER_MCP;14. RING_FINGER_PIP;
15. RING_FINGER_DIP;16. RING_FINGER TIP;
17. PINKY_MCP;18. PINKY_PIP;
19. PINKY_DIP;20. PINKY_TIP;

Voice Assistant:

A voice assistant uses speech recognition, language processing algorithms, and voice synthesis to listen to specific vocal commands and returns relevant information or executes specified tasks as desired by the user. It can give relevant information depending on the user's specific instructions, also known as intents, by filtering out background noise. While some voice assistants are entirely software-based and may be integrated into a wide range of devices, others are created particularly for a single device.

Voice assistants have a rich history that dates back over 100 years, which may seem odd given that applications like Siri have just been available for ten years. Voice assistants are already built into many of the products we use on a daily basis, including mobile phones, PCs, and smart speakers. There are numerous voice assistants that offer a very specialised feature set due to their large range of integrations, while others choose to be open ended to aid with nearly any circumstance at hand.

We utilised pyttsx3, Speech Recognition, and other modules to create Voice Assistant. Our program (Fig 2.) will currently consist of three modules: Gesture controller, voice assistant, and Bot. Following the execution of the software, two types of input will be available: direct hand movements and voice instructions via the Dragon app. A series of hand movements such as pinching, waving of the hand, and coordinated

movement of multiple different fingers will operate the virtual mouse as well as other activities such as voice control, screen brightness, selection, and so on.

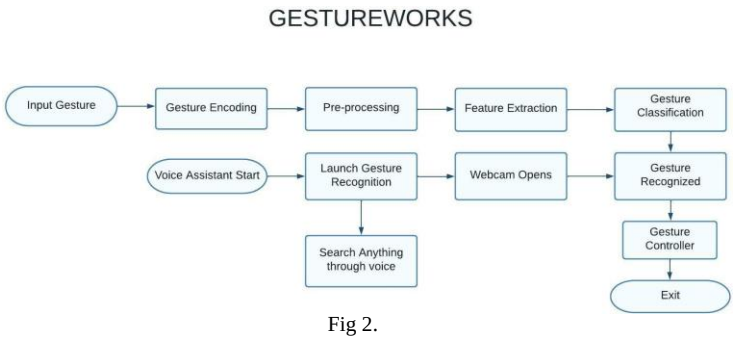


Fig 2.

IV. Proposed Methodology

We have two basic sources of input: hand images through webcam for virtual mouse and speech recognition for voice assistant. After receiving the input, pre-processing and feature extraction are performed, and the required functions are carried out in accordance with the desired hand symbols or input speech.

The following are the workings of several libraries and modules utilized in the process:

(A) pyttsx3:

It is a Python text-to-speech conversion package. It operates offline, unlike other libraries, and also is consistent both with Python 2 and Python 3.

(B) Speech Recognition:

It is a library that performs Speech Recognition tasks with the help of several engines and APIs, both online and offline.

(C) pynput:

Controlling and monitoring input devices is possible using this library. Input and monitoring through mouse and keyboard are currently supported.

(D) pyautogui:

It is a Python GUI automation package for humans that works across platforms. Controls the mouse and keyboard programmatically.

(E) Wikipedia:

It is a Python package that makes it simple to retrieve and interpret data from Wikipedia.

(F) comtypes:

It is a little Python COM package written in even less than 10,000 code lines and built upon the ctypes FFI library.

It enables for the creation, use, and implementation of customised & dispatch-based COM interface in Python. It is compatible with both Windows and Windows 64-bit.

(G) pycaw:

It stands for Python Core Audio Windows Library and is used for Audio Utilities in both Python2 & Python3.

(F) screen-brightness-control:

A Python utility for adjusting the intensity of your display.

It Supports Windows and the majority of Linux distributions.

(G) eel:

It is a small Python library for creating basic offline HTML/JS GUI programmes similar to Electron, having complete access to Python features and resources.

(H) OpenCV:

OpenCV is a vision - based package that includes object identification picture processing techniques. The computer vision library OpenCV is a python programming language library that may be used to create real-time computer vision applications. The OpenCV package is often used to analyse images and videos, as well as perform analyses such as face and object detection.

(I) Mediapipe:

MediaPipe framework is often used by developers to design and analyse systems using graphs, as well as to create systems for application usage. The stages in a MediaPipe-based system are performed in the pipeline architecture. The pipeline developed may be executed on a variety of platforms, providing for extensibility in desktop and mobile devices environments. The MediaPipe system is comprised of three major components: performance evaluation, a framework for obtaining sensor data, as well as a collection of reusable components known as calculators.

The pipeline is a graph made up of parts called calculators, each of which is linked by streams through which data

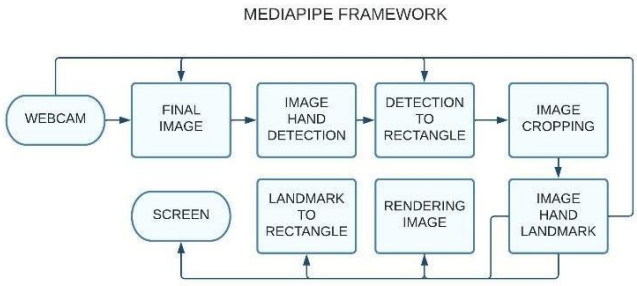


Fig 3.

packets pass. Developers may change or define own calculators everywhere in the graph, allowing them to create their own application.

Whenever the calculators & streams are joined, they form a data-flow diagram. It's made with MediaPipe, and then each node is just a calculator that's linked together via streams. A single-shot detector type is employed for real-time detection and recognition of a hand or palm. The MediaPipe (2019)[12][13][14] employs the single-shot detector concept. Initially, the hands detection module trains for a hand detection model since palms are easier to train. Furthermore, the minimal suppression works much better on tiny things like hands or fists. A hand landmark model is made up of 21 joints or knuckles co-ordinates located in the hand area.

V. Functions & Features

The system's many functions and characteristics are described below.

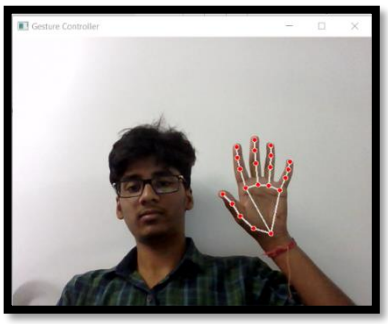
The suggested AI virtual mouse technology is based on images taken by a laptop or PC's camera. This video capture object made using Python's computer vision module OpenCV, using which the webcam begins recording footage. The web camera records images and sends them towards the GestureWorks. The GestureWorks use a camera to capture each frame until the application is terminated. As illustrated in the accompanying code, the video frames are converted to RGB from BGR colour space in order to detect the hands in the movie frame by frame.

Pseudo Code:

```
def start(self):
image = cv2.cvtColor(cv2.flip(image, 1),
cv2.COLOR_BGR2RGB)
image.flags.writeable = False
results = hands.process(image)
image.flags.writeable = True
image = cv2.cvtColor(image,
cv2.COLOR_RGB2BGR)
```

Gesture Recognition:

(A) Neutral Gesture:

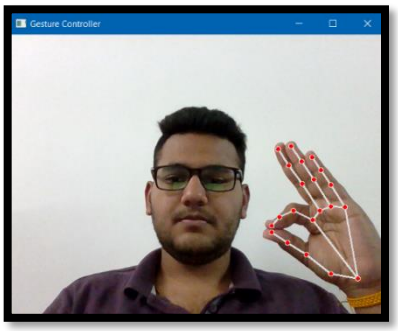


It is used to pause/stop the execution of the current gesture by moving hand in inwards and outwards direction.

(B) Move Cursor:

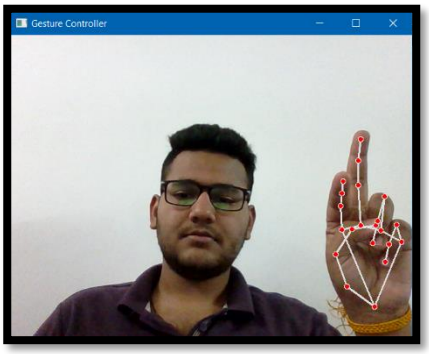
The middle of the index and middle fingers is designated as the cursor. The pointer is moved to the desired position with this motion. The cursor movement speed is related to the speed of the hand.

(C) Scrolling:



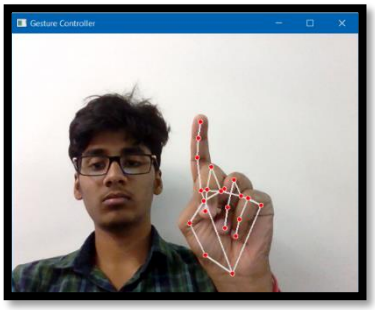
Scroll horizontally and vertically with Dynamic Gestures. The scroll speed is related to the distance advanced by the pinch gesture from the starting position. Vertical and lateral pinch movements control vertical and horizontal scrolling, respectively.

(D) Left Click:



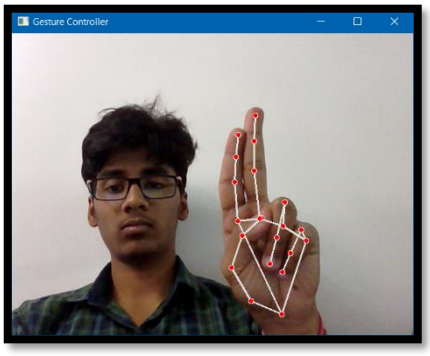
The cursor is defined by the area between the index and middle fingers. For the left click to occur, the index finger is directed downward.

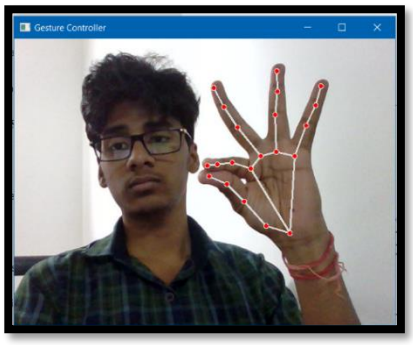
(E) Right Click:



The cursor is defined by the area between the index and middle fingers. For the right click to occur, the middle finger is directed downward.

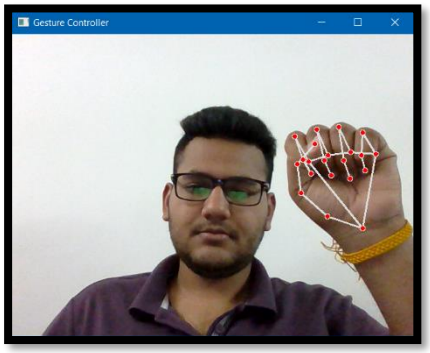
(F) Double Click:





For the double click to occur, the index finger and the middle finger are joined together.

(G) Drag & Drop:

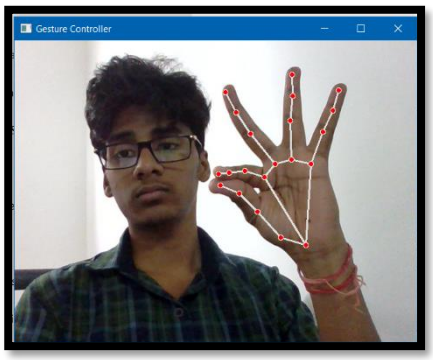


Drag-and-drop functionality is demonstrated with this gesture. It is possible to use this function to move or transfer files from one location to another.

(H) Multiple Item Selection:

Select numerous things by dragging the cursor over them with your closed fist.

(I) Brightness Control:

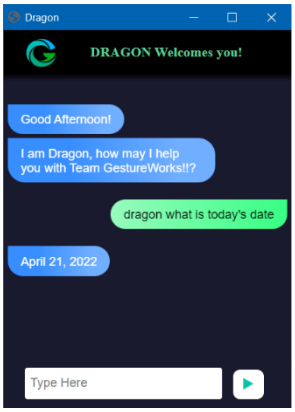


The rate of brightness increase/decrease is related to the distance travelled by the pinch motion from the starting point (left/right).

(J) Volume Control:

The rate of volume increase/decrease is related to the distance travelled by the pinch motion from the starting position(upwards/downwards).

Voice Assistant:



(A) Launch/Stop Gesture Recognition

Launch: Turns on the camera to recognise hand gestures. Stop: Stops gesture recognition and turns off the camera.

(B) Google Search

Google is used to do a search for the specified text.

(C) Current Date/Time

The current time and date are returned by specifying "Dragon What's the time/date?.

(D) File Navigation

It lists/opens/changes the current/specified directory.

(F) Sleep/Wakeup Dragon

Sleep: It pauses the voice recognition function until it wakes up Wakeup: It resumes the voice recognition function.

(G) Copy/Paste

Copy: It copies the selected text in the Clipboard. Paste: It pastes the text which is in Clipboard.

(H) Exit

The voice assistant thread is terminated.

VI. RESULTS

The notion of enhancing interaction with people using computer vision is presented below:

Cross-validation of the Gesture Works is problematic due to the smaller number of datasets available. Gestures recognition have been tested in a variety of lighting situations, as well as at varied ranges from the webcam for monitoring and recognition of gestures.

To summarise the data reported in Table 1, an experimental test was performed. The test was carried out in various lighting conditions and at various distances from of the screen, and each participant checked the Gesture Works. The experimental findings are summarised in Table 1.

TABLE I

FINGER TIP GESTURE	MOUSE FUNCTION PERFORMED	SUCCESS	FAILURE	ACCURACY
Tip ID 1 or both tip IDs 1 and 2 are up	Mouse movement	95	0	95
Tip IDs 0 and 1 are up and the distance between the fingers is <30	Left button click	94	1	94
Tip IDs 1 and 2 are up and the distance between the fingers is <40	Right button click	95	3	95
Tip IDs 1 and 2 are up and the distance between the fingers is >40 and bothfingers are moved up the page	Scroll up function	95	0	95

Tip IDs 1 and 2 are up and the distance between the fingers is >40 and both fingers are moved down the page	Scroll down function	91	1	91
All five Tip IDs 0,1,2,3,4 are up	No action performed	94	1	94
RESULT		564	6	94

According to Table 1, GestureWorks attained an accuracy of around 94 percent. We may conclude that GestureWorks functioned effectively based on its 94 percent accuracy. as seen in the Table 1, overall accuracy for "Right Click" is poor since this is the most difficult motion for computer to grasp. Because the gesture utilised to achieve the specific mouse function is more difficult, overall accuracy for right - clicking is low. Also, the accuracy for all other motions is good and high. In comparison to earlier techniques for web mouse, our model performed exceptionally well, with 94 percent accuracy. Figure 14 depicts an accuracy graph.

TABLE II

Existing Models	Accuracy(%)
Virtual mouse system using RGB-D images and fingertip detection[15]	92.13
Palm and finger recognition based [16]	78
Hand Gesture-based virtual mouse [17]	78
GestureWorks	94

Classification metrics Testing

(A). Accuracy:

Graph:

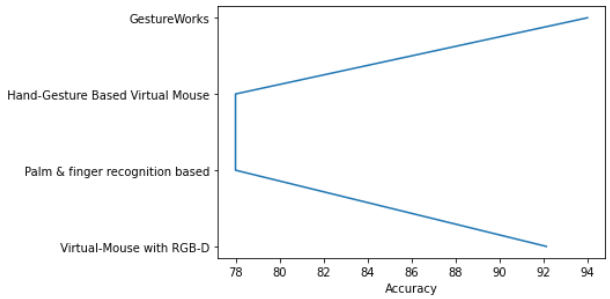


Fig 4.

(B). Confusion Matrix

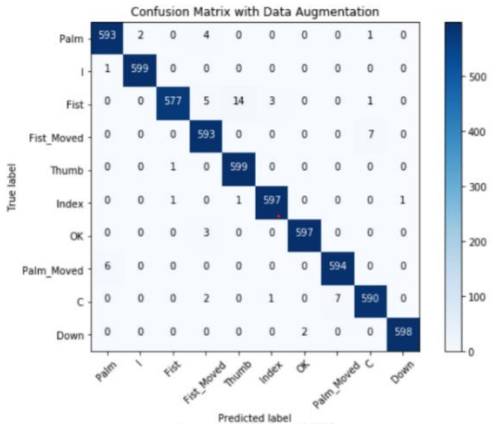


Fig 5.

VII. CONCLUSION

The primary goal of the GestureWorks is to operate mouse cursor functionalities with hand gestures rather than a hardware mouse. The suggested system may be realised by employing a webcam or an in-built camera that recognises hand motions and hand tips and analyses these frames to execute the specific mouse tasks.

Based on the model's findings, we can conclude that the Gesture Works worked extremely well and has more accuracy than current models, and that the model solves most of the constraints of existing systems. GestureWorks may be utilised for real-world applications since the suggested model is more accurate, and it can also be used to minimise the

spread of COVID-19 because it can be operated virtually using hand movements rather to the standard physical mouse. The model has various limitations, such as a little loss in precision in the right - clicking mouse operation and some difficulty in simply dragging and dropping to pick text. As a result, we will strive next to address these constraints by upgrading the fingertip detecting algorithm to give more accurate results.

VIII. APPLICATIONS

GestureWorks is beneficial for a variety of applications; it may be used to save space while using a real mouse, as well as in situations when we cannot use a physical mouse. The method reduces the need for gadgets while also improving human-computer connection.

Important Applications are:

The suggested model has a higher accuracy of 94 percent, which is much higher than other proposed virtual mouse models, and it has various applications.

In the midst of the COVID-19 condition, this is not safe to utilise the equipment by touching them since doing so may result in the virus spreading, GestureWorks may be used to operate the PC mouse operations without using the real mouse.

The system can also be used to operate robots and automated systems without the need of gadgets.

It can also be used to draw 2D and 3D graphics using hand gestures.

It can also be used to play VR/AR games without the need for wireless or cable mouse devices. People with hand issues can utilise this technology to control the computer's mouse operations. The suggested system, such as HCI, can be utilised to control robots in the field of automation.

The suggested system in design and architecture may be utilised for digitally designing and prototyping.

IX. REFERENCES

[1] Pantic M, Nijholt A, Pentland A, Huanag TS, Human-Centered Intelligent Human-Computer Interaction (HCI2): How Far We from Attaining It? International Jounal of Autonomous and Adaptive Communications Systems (IJAACS), vol.1 no.2, 2008. pp 168-187. DOI: 10.1504/IJAACS.2008.019799.

[2] Lugaresi C, Tang J, Nash H, McClanahan C, et al. MediaPipe: A Framework for Building Perception Pipelines. Google Research. 2019. https://arxiv.org/abs/2006.10214.

[3] Y.Li, Hand Gesture Recognition Using Kinect, 2012.

[4] E. Holden, Visual Recognition of Hand Motion, Ph. D Thesis Departement of Computer Science, University of Western. 1997.

[5] Yang M.H, Ahuja N, Tabb M, Extraction of 2D Motion Trajectories and its Application to Hand Gesture Recognition. IEEE Trans on PAMI vol.29. 2002. pp 1062-1074.

[6] Dewaele G, Devernay F, Horaud R. Hand Motion from 3D Point Trajectories and Smooth Surface Model. In Processing of 8th ECCV. 2004.

[7] Lee H, Kim J. An HMM-Based Threshold Model Approach for Gesture Recognition. IEEE Trans on PAMI vol.21. 1999. pp 961-973.

[8] Wilson A, Bobick A, Parametric Hidden Markov Models for Gesture Recognition. IEEE Trans. On PAMI vol.21, 1999. pp.884-900.

[9] Lugaresi C, Tang J, Nash H et.al, MediaPipe: A Framework for Perceiving and Processing Reality. GoogleResearch. 2019.

[10] Abadi M, Barham P, Chen J et.al, TensorFlow: A Systemfor Large-Scale Machine Learning, In 12th USENIX Symposium on Operating System Design and Implementation (OSDI), USA, 2016, https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

[11] Matveev D, OpenCV Graph API. Intel Corporation. 2018.

[12] Zhag F, Bazarevsky, Vakunov A et.al, MediaPipe Hands: On – Device Real Time Hand Tracking, Google Research. USA. 2020. https://arxiv.org/pdf/2006.10214.pdf.

[13] Media Pipe: On-Device, Real Time Hand Tracking, In https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html. 2019. Access 2021.

[14] MediaPipe GitHub: https://google.github.io/mediapipe/solutions/hands. Access 2021.

[15] D.-S. Tran, N.-H. Ho, H.-J. Yang, S.-H. Kim, and G. S. Lee, "Real-time virtual mouse system using RGB-D images and fingertip detection," *Multimedia Tools and Applications Multimedia Tools and Applications*, vol. 80, no. 7, pp. 10473–10490, 2021.

[16] A. Haria, A. Subramanian, N. Asokkumar, S. Poddar, and J. S. Nayak, "Hand gesture recognition for human computer interaction," *Procedia Computer Science*, vol. 115, pp. 367–374, 2017.

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

[18] Wiskott, L., Fellous, J.-M., Kruger, N., und von der Malsburg, C.:

Face recogn ¨ ition by elastic bunch graph matching. IEEE Transactions on Pattern Analysis and Machine, Intelligence. 19(7):775–779. 1997.
[19] B. Galveia, T. Cardoso and Y. Rybarczyk, Adding Value to the Kinect SDK, Creating a Gesture Library, 2014.
[20] A Review on Gesture Controlled Virtual Mouse Romanshu Agrekar, Samyak Bhalerao, Mohiuddin Gulam, Vaibhav Gudadhe and Kalpana Bhure First published: 5 January 2022.
[21] Artificial Intelligence-Enabled Sensing Technologies in the 5G/Internet of Things Era: From Virtual Reality/Augmented Reality to the Digital Twin Zixuan Zhang,Feng Wen,Zhongda Sun,Xinge Guo,Tianyiyi He,Chengkuo Lee First published: 29 March 2022