Table of Contents 1

1. INTRODUCTION

Gesture recognition is an important part of computer science with the aim of understanding human gestures through algorithms. Computer vision-based gesture recognition enables people to communicate more naturally with machines. Interactive presentation systems use advanced Human Computer Interaction (HCI) techniques to provide a more convenient and user-friendly interface for giving commands to the machine, such as page up/down controls, open menu, exit app, etc. Compared with traditional mouse and keyboard control, the experience is significantly improved with these techniques.

The advantage is that it is less affected by the environment. Users can interact with computers at any time, and it has less constraint on users, enabling computers to accurately and timely understand the user's instruction. The instructions do not require any mechanical assistance making the whole process efficient in an overall manner.

Gestures are timely, vivid, intuitive, flexible and visual in the process of human computer interaction. They can soundlessly complete interaction and successfully break the gap between reality and the virtual world. Introduction of gesture recognition into the online teaching world would open gates of new opportunities to students and teachers making the whole experience more interactive and effective.

Input units have been a major part of computing machines, however technology has much more to offer. The proposed project believes in the idea of making the process of giving inputs to machine more time efficient as well as an effective process which will decrease the gap between input and output cycle providing a more user friendly interface that is also capable of customization for special needs of situation as well as user.

1.1. Purpose of the Project

Online teaching is a rather new concept which has recently come to day-to-day practice making the whole process not so friendly for the teachers who have been usings chalks and markers and students who are habitual of studying in a real time classroom. Writing on an online board is a task that mostly no teacher/ user of online teaching platforms can avoid. It is generally rather difficult for users to write using a standard mouse/ mousepad. The uneasiness of the whole process consumes a lot of time and makes the classroom less efficient by minimising the output. Stylist does make the job easier but it increases the input cost and adds on to the cluttering of the workspace also taking extra space in the laptop bag. Similarly, there are various challenges that users of online teaching platforms all over the world face which are not been dealt with eagerness.

The process of online teaching, office meetings, presentations, etc will become a lot easier if the minimum input required would just be the user and some room light. The proposed project aims in minimising the hassles listed above by introducing gesture and body recognition as a major method of providing commands to the computer.
The common problem of writing on online boards faced by users is solved by simply providing the option of writing directly on the screen by using the user's mere finger as a pointer and palm as an eraser. This makes the whole teaching experience more time sufficient, user friendly, interactive and productive.
The gesture recognition method also makes multitasking easier. Eg- a user can drag an image on the screen, while resizing it accordingly and can also write or erase on the screen/ image at the same time. This process is not limited to images and can be used simultaneously with pdf files, slideshows, word files, etc.

This method removes the responsibility from the user to own the appropriate input units and use them correctly to give commands. By using the gesture recognition methods, the user can give commands in time to the machine and doesn't really have to worry about learning to use the various input units and wait for them to ask the machine to perform a certain task.

Simple method of hand gesture recognition task is given below:

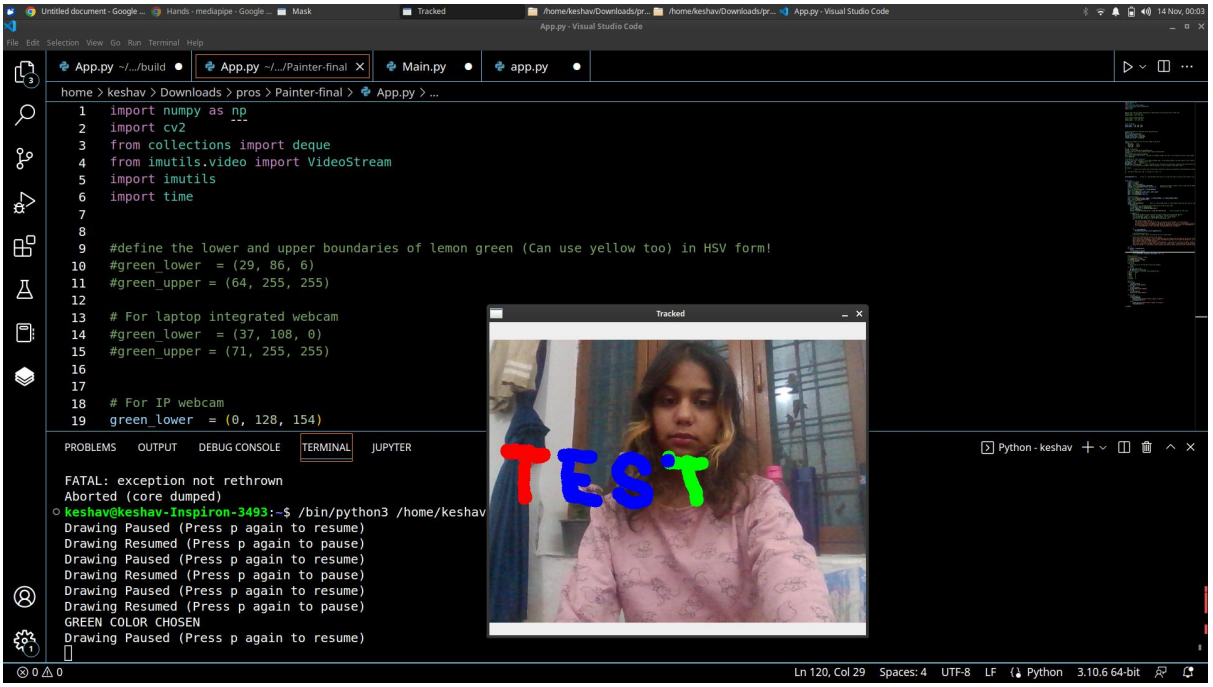


Fig 1.1 *Testing painter from the menu*

Data gloves are a popular method used to perform similar activities however the proposed projects discourage the use of the same to minimise usage of multiple units. The removal of data gloves from the project is supported by the well known framework proposed by Google named MediaPipe.

MediaPipe is a cross platform pipeline framework to build custom machine learning solutions for live and streaming media. The framework was open sourced by Google and is currently in the alpha stage.

1.2 Target Beneficiary

Body Recognition
The foremost target of the project was to identify specific body parts as well as the whole body and its structure. This will further be used as the base of the project.

Setting Gesture Commands
After achieving the Body recognition, gestures were listed and given commands accordingly. This process in the project majorly dealt with landmarks on the hand and differences of distance among them.

Online Educational Platform
The project aims to put forward a platform which is more interactive and accessible by a variety of users. It will deal with controlling computer devices via Gesture Recognition. The system will focus on interaction of human hands with the computer by recognizing the hand movements of the user through a camera without touching. This system will also provide interactive U.I., sending a message, controlling the menu, managing media etc. This will open opportunities for new experiences for the users with the chance to have a more viable and interactive experience easily, quickly and hands free.

Integration
In the given project there is a scope of integration of the suggested methods with existing platforms such as google meet, blackboard, zoom, etc to increase the potency of the integrated applications as well as experience of the user.

1.3 Project Scope

Controlling Computer Devices via Gesture Recognition System has various uses which also include using multi-tasking task completion. This technique can be used with various project and in various ways according to one's imagination and needs as this method is flexible and highly customizable.
The practice can bring significant changes into online teaching methods by providing teachers tools that are easily accessible and students a more real time experience. Some features accessed by gestures :

- Markers that can draw on the screen using finger
- Erasers to clear the screen
- Sharing various types of file formats on screen
- Menu accessed by hand gestures
- Online board to write on with hand gestures, etc.

Using body gestures to provide commands to the computing machine reduces the gap between reality and the machine world which further makes the experience of the user more extraordinary and real. Because of the usability of gesture recognition, this idea and its algorithms can be used in many other areas like game consoles, tablet PC's etc., Virtual Realities and so on.

Moreover, the major parameters of this technology such as customization and minimization of input devices can be used to create interfaces for specially abled people, older population or younger children who are new to technology.

2. Project Description

The model of this project can be describe into different phases

1. Extracting video frames using OpenCV

First importing OpenCV, it will access the camera and start taking the video frame by frame. Now the variable is storing the first frame and sending it to the thread made from MediaPipe so it can read the frame using stream.read() function.[11]
Here is how reading frame works:

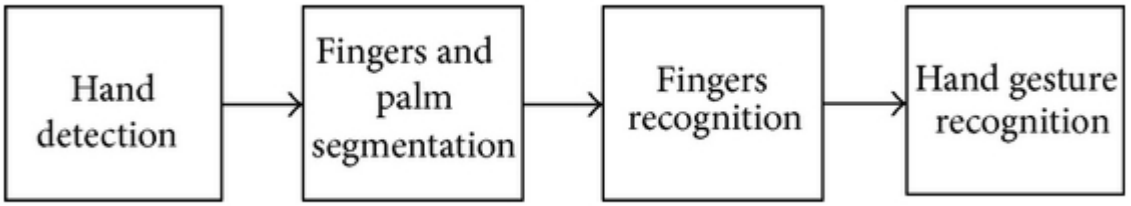


Fig 2.1: *Flow chart for gesture recognition*

2. Reading the Frame using Mediapipe

While reading the first frame it captured, the mediapipe solution will detect the hand as this is a well trained framework. After that it will make the landmarks on the palm, elbow and shoulders and provide a defined name and number to those landmarks [15].

3. Commands using conditions

Now after storing the updated frame, conditions can be used to give suitable commands to the program. These conditions and landmarks can be used to provide various outputs according to the user.

2.1 Reference Algorithm

MediaPipe

MediaPipe offers open source cross-platform, customizable ML solutions for live and streaming media.

Mediapipe Framework

Today, there are many frameworks or libraries of machine learning for hand gesture recognition. One of them is MediaPipe. The MediaPipe is a framework designed to implement production-ready machine learning that must build pipelines to perform inference over arbitrary sensory data, has published code accompanying research work, and build technology prototypes . In MediaPipe, graph modular components come from a perception pipeline along with the function of inference model function, media processing model, and data transformations. Graph of operations are used in others machine learning such as Tensor flow , MXNet, PyTorch[, CNTK, OpenCV 4.0. [13]

Using MediaPipe for hand gesture recognition has been researched by Zhang, before, using a single RGB camera for AR/VR application in a real-time system that predicts a hand skeleton of the human. We can develop a combined MediaPipe using other devices. The MediaPipe implements pipeline consists of two models for hand gesture recognition as follows :

1. A palm detector model processes the captured image and turns the image with an or -iented bounding box of the hand,

2. A hand landmark model processes on cropp -ed bounding box image and returns 3D hand k -ey points on hand.

3. A gesture recognizer that classifies 3D hand key points then configuration them into a disc -rete set of gestures.

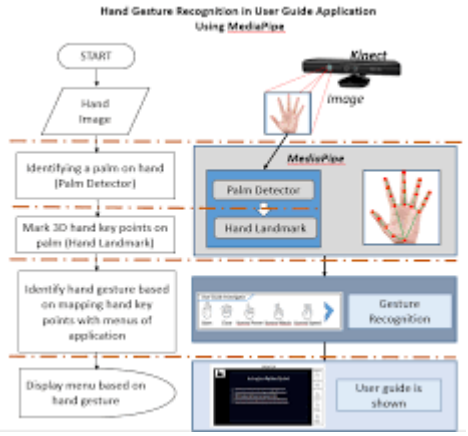


Fig 2.2: *Workflow Method Research.[4]*

MediaPipe Hands

Palm Detection Model

To detect initial hand locations, This model optimised for real-time uses in a manner similar to the face detection model. Detecting hands is a decidedly complex task: this model has to work across a variety of hand sizes with a large scale span (~20x) relative to the image frame and be able to detect occluded and self-occluded hands. Whereas faces have high contrast patterns, e.g., in the eye and mouth region, the lack of such features in hands makes it comparatively difficult to detect them reliably from their visual features alone. Instead, providing additional context, like arm, body, or person features, aids accurate hand localization.This method addresses the above challenges using different strategies. First, it trains a palm detector instead of a hand detector, since estimating bounding boxes of rigid objects like palms and fists is significantly simpler than detecting hands with articulated fingers. In addition, as palms are smaller objects, the non-maximum suppression algorithm works well even for two-hand self-occlusion cases, like handshakes. Moreover, palms can be modelled using square bounding boxes (anchors in ML terminology) ignoring other aspect ratios, and therefore reducing the number of anchors by a factor of 3-5. Second, an encoder-decoder feature extractor is used for bigger scene context awareness even for small objects (similar to the RetinaNet approach). Lastly, it minimises the focal loss during training to support a large amount of anchors resulting from the high scale variance.[12][13]

With the above techniques, it achieves an average precision of 95.7% in palm detection. Using a regular cross entropy loss and no decoder gives a baseline of just 86.22%.

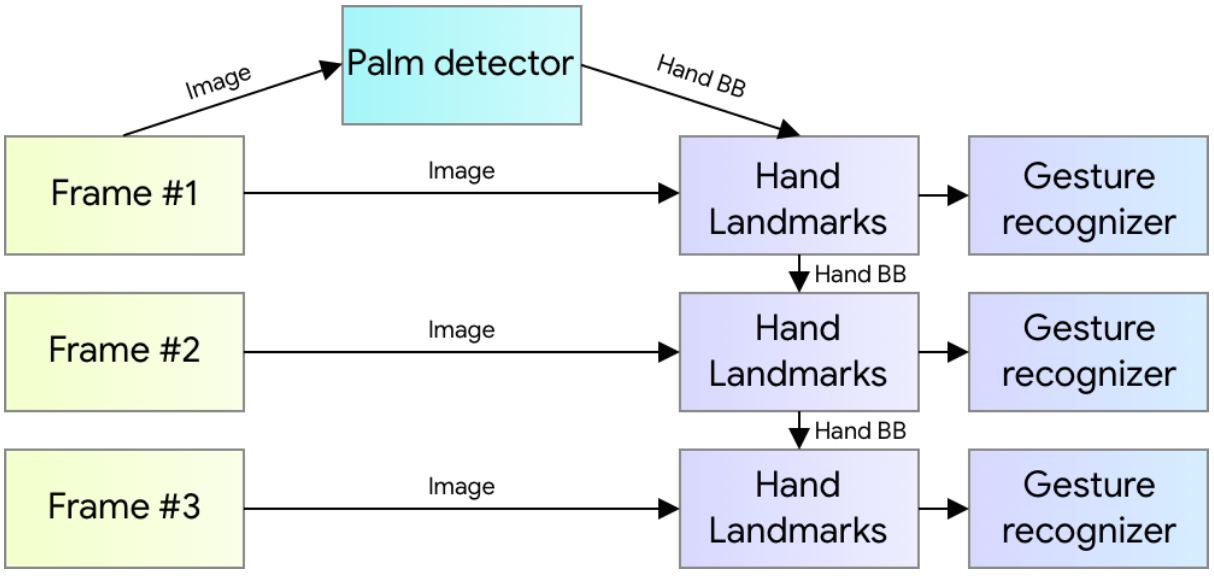


Fig 2.3 *Hand Perception Pipeline Overview [13]*

1. Hand Landmark Model

After the palm detection over the whole image our subsequent hand landmark model performs precise keypoint localization of 21 3D hand-knuckle coordinates inside the detected hand regions via regression, that is direct coordinate prediction. The model learns a consistent internal hand pose representation and is robust even to partially visible hands and self-occlusions.[12]

To obtain ground truth data, we have manually annotated ~30K real-world images with 21 3D coordinates, as shown below (we take Z-value from image depth map, if it exists per corresponding coordinate). To better cover the possible hand poses and provide additional supervision on the nature of hand geometry, we also render a high-quality synthetic hand model over various backgrounds and map it to the corresponding 3D coordinates.

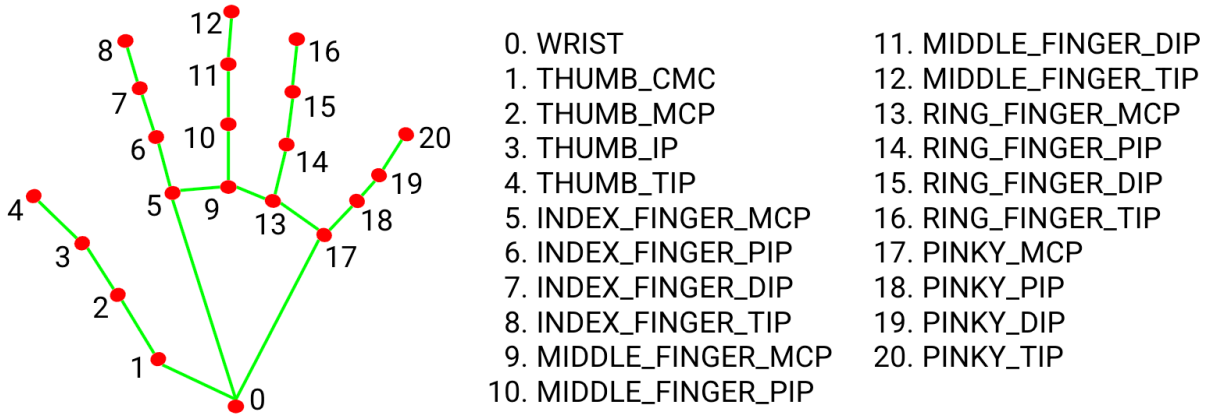


Fig 2.4: *Hand Landmark in MediaPipe [15]*

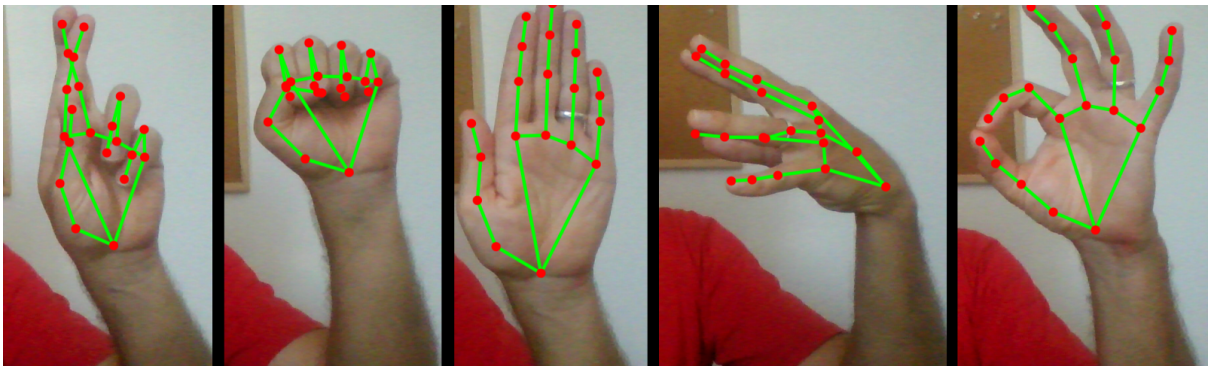
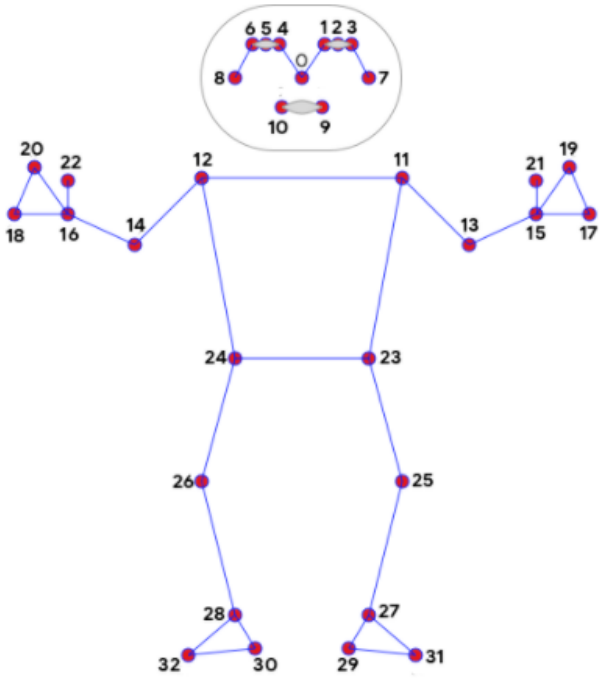


Fig 2.5: *Live video hand landmarks using mediapipe*

MediaPipe Pose

Pose Landmark Model (BlazePose 3D)

The landmark model in MediaPipe Pose predicts the location of 33 pose landmarks (see figure below).



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

Fig 2.6: *Body pose landmark in MediaPipe [13]*

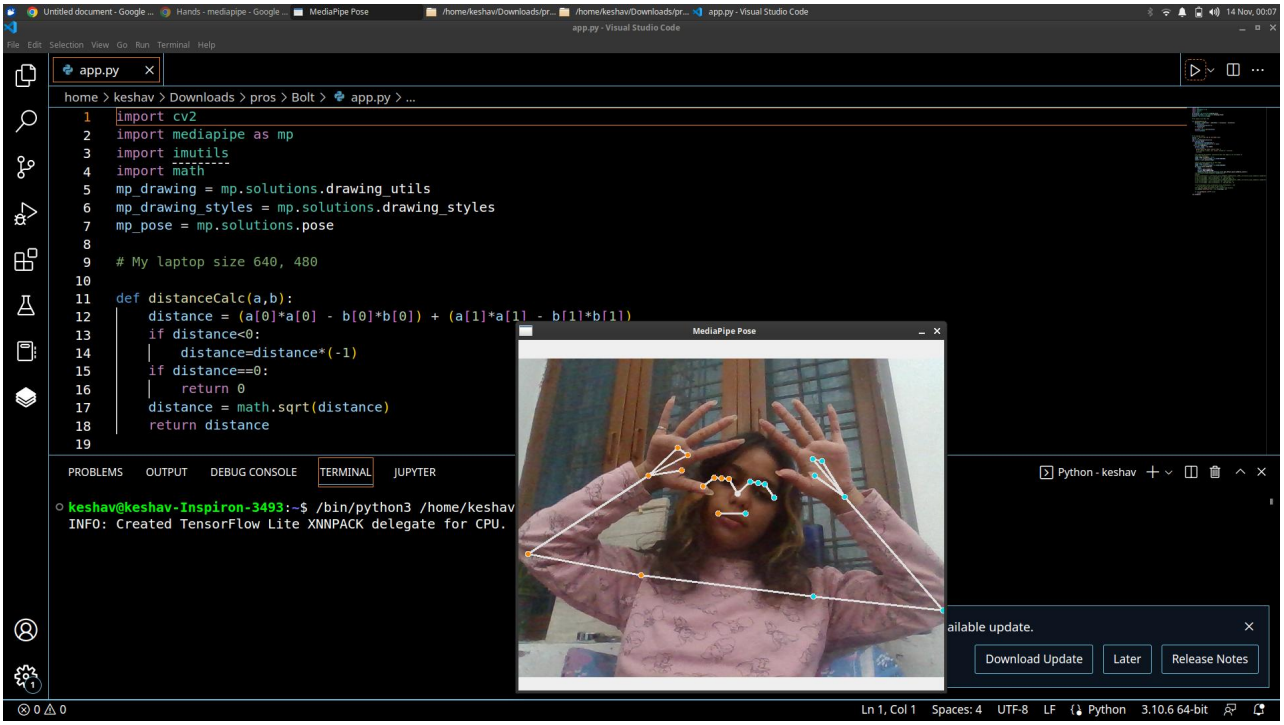


Fig 2.7: *Interface using Gesture Recognition*

2.2 SWOT Analysis

Strengths

1. Free and open source
2. Low-cost
2. Software easy and fast to use
3. Light Weight Device
4. Easy Identification

Weakness

1. Limited Tracking Accuracy
2. Tracking Area limited to the field
3. Limited to a user

Opportunities

1. More needed tools
2. Remove limitations(no. Of user, lighting.etc)
3. Frame Rate control

Threats

1. Undefined gesture control intrusion recovery process

2.3 Project Features

Variations in image plane and pose: The hands in the image vary due to rotation, translation and scaling of the camera pose or the hand itself. The rotation can be both in and out of the plane.

Lighting Condition and Background: As shown in Figure 1.1 light source properties affect the appearance of the hand. Also, the background, which defines the profile of the hand, is important and cannot be ignored.

2.4 Design and Implementation Constraints

Although the software utilises different methods of filtering and object detection, the system still varies under different environmental lights. This system cannot guarantee to perform correct hand detection under very bright light sources such as direct sunlight or having bright lights in the background in the case of windows. These factors affect the algorithm due to the fact that the light softens the skin colour needed for detection, thus rendering the skin detector ineffective to detect accurately.

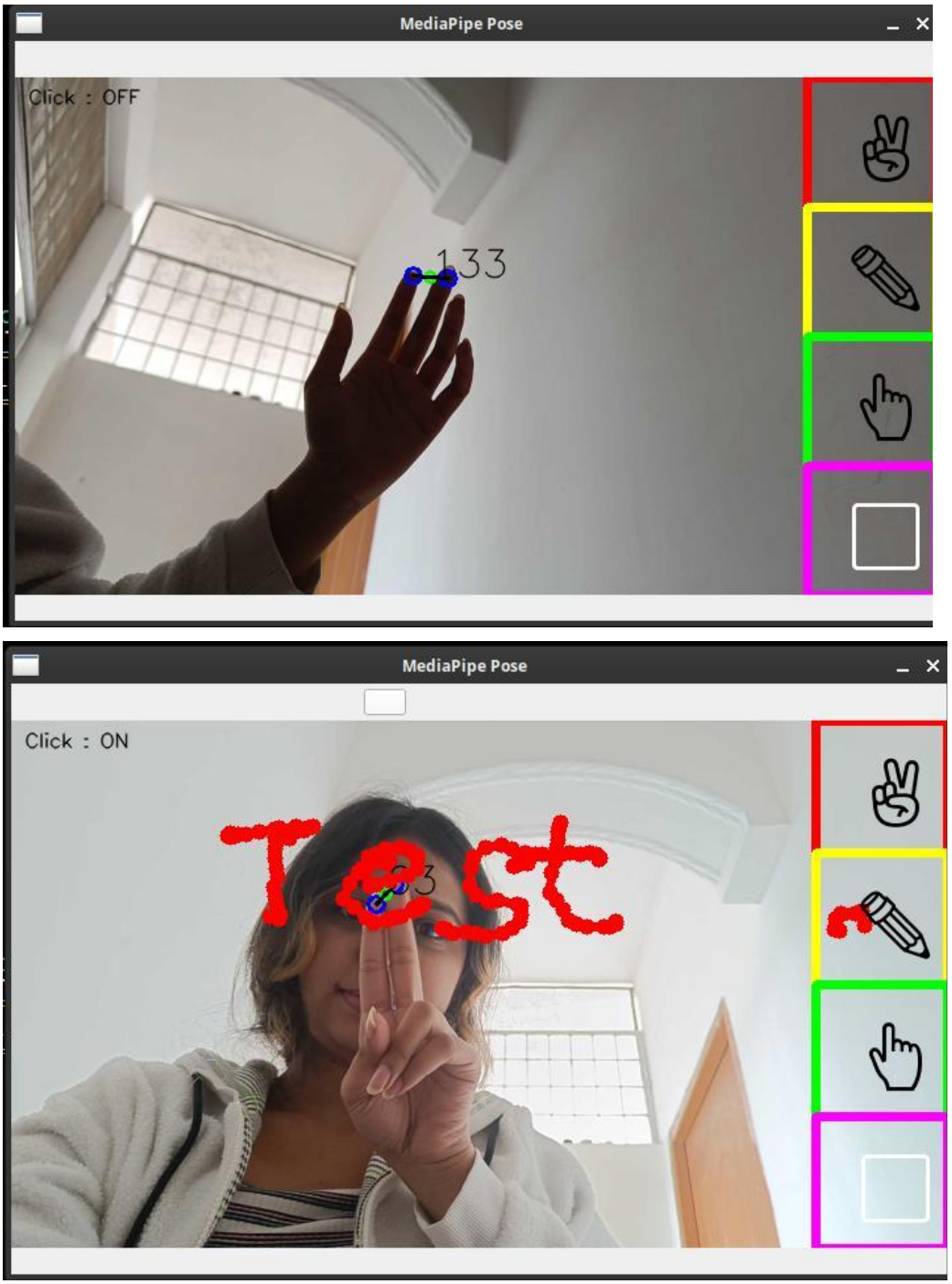
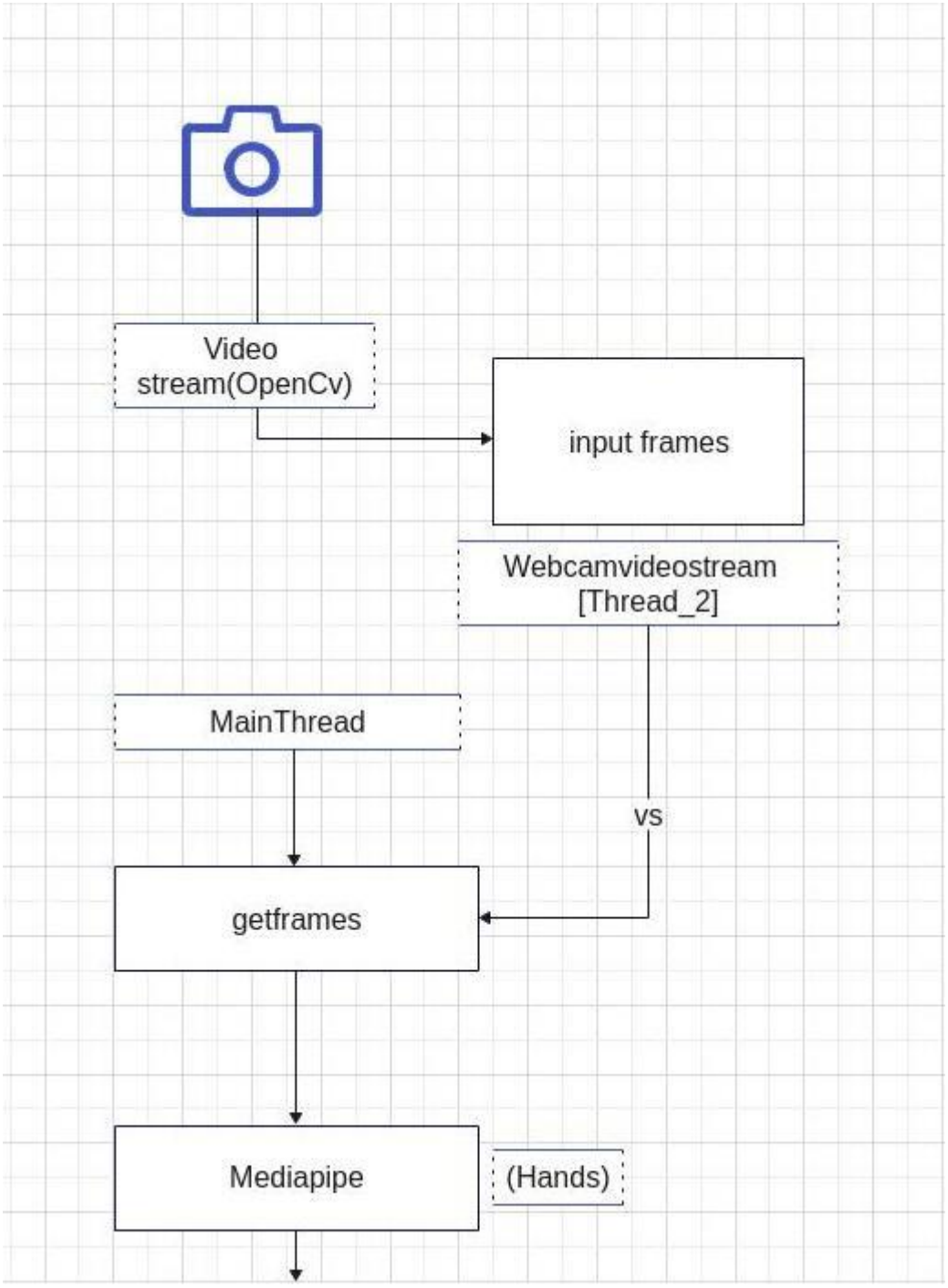
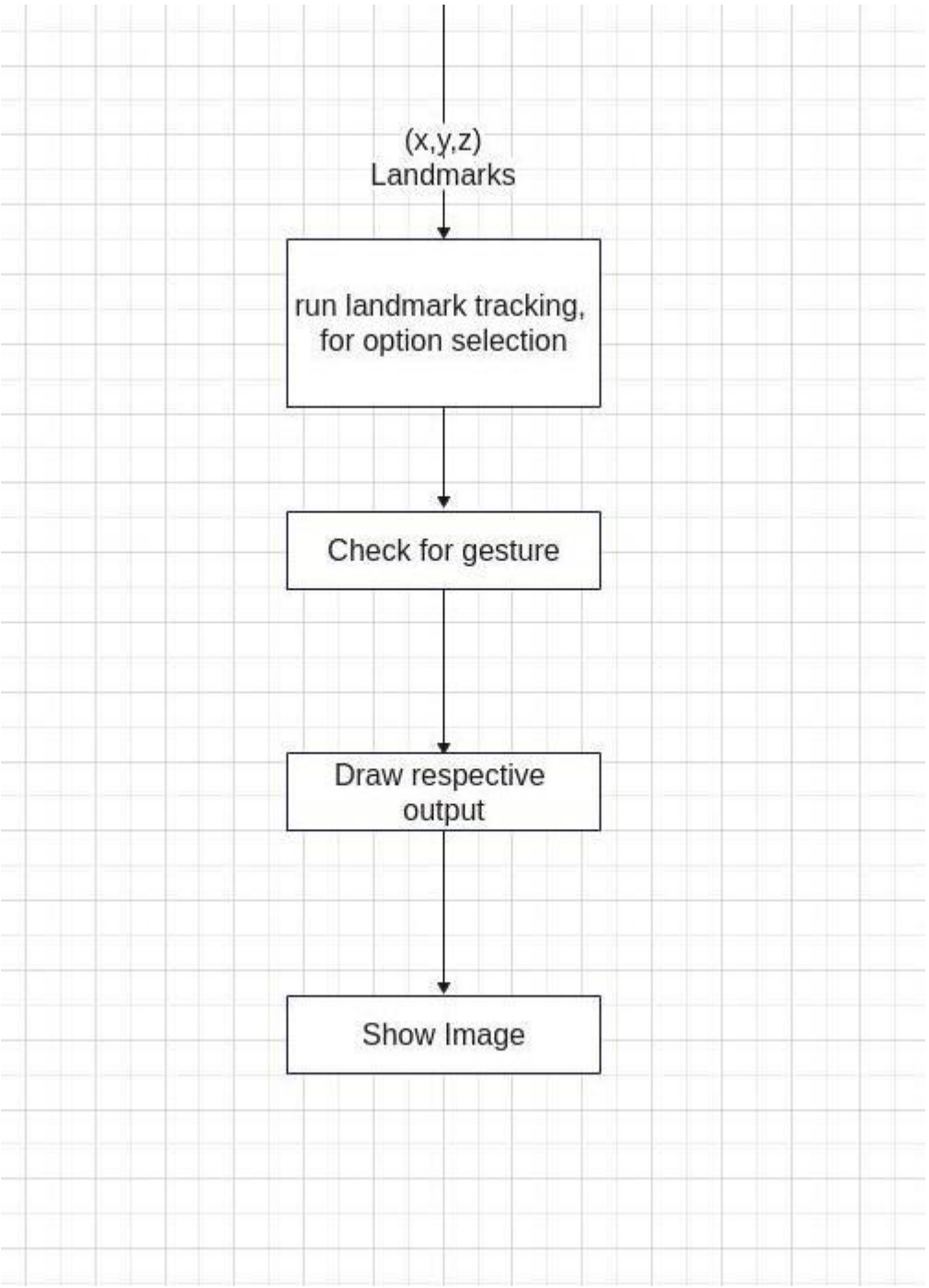


Fig 2.8: *Screenshots of user guide application*

2.7 Design Diagram




(x,y,z)
Landmarks
run landmark tracking, for option selection
Check for gesture
Draw respective output
Show Image

3. System Requirements

Hardware

RAM: 6 GB	Disk Space: 10GB
Processor:	Intel i5 gen 10 or Ryzen 3 3250u (or higher
Software:	VS code
Environment Software:	VS CODE
Language:	Python
Operating System:	Windows or Linux

For a smoother experience the system requirement below is recommended

Hardware

RAM: 8 GB	Disk Space: 10GB
Processor:	Intel i7 or Ryzen 5 5600x (or higher
Software:	VS code
Environment Software:	VS CODE
Language:	Python
Operating System:	Windows or Linux

3.1 User Interface

This software shall be easy to use for all users with minimal instructions. 100% of the languages on the graphical user interface (GUI) shall be intuitive and understandable by non-technical users.

This software shall be operable in all lighting conditions. Regardless of the brightness level in the user's operating environment, the program shall always detect the user's hands.

3.2 Software Interface

Face Detection

This software shall utilise a face detection system to filter out faces from the video capturing device. By applying face detection, the system can disregard the region where the face is located and thus reduce the amount of calculation needed to perform hand detection.

Hand Calibration

Depending on the user's preferences, the system shall perform adjustments according to user's dominant hand. This means that if the user is right-handed, the mouse control gesture mode should be recognized near the right side of the face instead of the whole field of view. This is achieved through trigonometry maths conversions.

Mouse Movement Gesture Control Mode

After obtaining the location of the hand, the software shall use the detected location as the mouse cursor point. As the user moves his/her hands, the mouse should follow promptly on the screen.

Browsing Gesture Control Mode

The software shall allow the user to use the "Browsing Gesture Mode". In this mode, the user's hand gesture will only be recognized for commands including previous page, next page, scroll up and scroll down

4. Non-functional Requirements

This software shall minimise the use of the Central Processing Unit (CPU) and memory resources on the operating system. When the software is executing, the software shall utilise less than 80% of the system's CPU resource and less than 100 megabytes of system memory.

4.1 Performance requirements

This software shall minimise the number of calculations needed to perform image processing and hand gesture detection. Each captured video frame shall be processed within 350 milliseconds to achieve 3 frames per second performance. Requiring a minimum of four cores to perform the complex calculation of image capturing and processing

4.2 Security requirements

Web applications are available via network access, so it is difficult. If not possible, to limit the population of the end-user who may access the applications. In order to make the product sensitively connect and provide secure mode, this method was implemented throughout the infrastructure that supports the web application and within the application itself. Web Applications have become heavily integrated with critical corporate and database. E-commerce applications extract and then store sensitive customer information.

4.3 Software Quality Attributes

Portability

This software shall be 100% portable to all operating platforms that support OOPs (Object Oriented Programming). Therefore, this software should not depend on the different operating systems.

Extensibility

The software shall be extensible to support future developments and add-ons to the HGR software. The gesture control module of HGR shall be at least 50% extensible to allow new gesture recognition features to be added to the system.

5. Other Requirements

Basic Hardware requirements such as a pc or a laptop, a camera, a microphone and other such as a light stable atmosphere.

6 References

[1] Ustunug A, Cevikcan, Industry 4.0: Managing The Digital Transformation, Springer Series in Advanced Manufacturing, Switzerland. 2018. DOI: https://doi.org/10.1007/978-3-319-57870-5.

[2] Hamed Al-Saedi A.K, Hassin Al-Asadi A, Survey of Hand Gesture Recognition System. IOP Conferences Series: Journal of Physics: Conferences Series 1294 042003. 2019. DOI: https://doi.org/10.1088/1742-6596/4/042003.

[3] S.Rautaray S, Agrawal A. Vision Based Hand Gesture Recognition for Human Computer Interaction: A Survey. Springer Artificial Intelligence Review. 2012. DOI: https://doi.org/10.1007/s10462-012-9356-9.

[4] 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.

[5] C.Chua, H. Guan, Y.Ho, Model-Based 3D Hand Posture Estimation From a Single 2D Image. Image and Vision Computing vol.20, 2002, pp. 191-202.

[6] M.Panwar, Hand Gesture Recognition Based on Shape Parameters, In International Conferences: Computing Communication and Application (ICCCA), 2012.

[7] Marco Maisto, An Accurate Algorithm for Identification of Fingertips Using an RGB-D Camera, IEEE Journal on Emerging and Selected Topics in Circuits and System, 2013. pp. 272-283.

[8] Wu Xiayou, An Intelligent Interactive System Based on Hand Gesture Recognition Algorithm and Kinect, In 5th International Symposium on Computational Intelligence and Design.2012

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

[10] Abadi M, Barham P, Chen J et.al, Tensorflow: A System for Large-Scale Machine Learning, In 12th USENIX Symposium on Operating System Design and Implementation (OSDI), USA, 2016,
https://www.usenix.org/conference/osdi16/technica l-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] MediaPipe: On-Device, Real Time Hand Tracking, In https://ai.googleblog.com/2019/08/on-devicereal-time-hand-tracking-with.html. 2019. Access 2021.

[14] Grishchenko I, Bazarevsky V, MediaPipe Holositic – Simultaneoue Face, Hand and Pose Prediction on Device, Google Research, USA, 2020, https://ai.googleblog.com/2020/12/mediapipeholistic-simultaneous-face.html, Access 2021.

[15] MediaPipe Github: https://google.github.io/mediapipe/solutions/hands. Access 2021.