VIRTUAL INSCRIBER

Submitted in partial fulfillment of the requirements of the degree of

BACHELOR OF COMPUTER ENGINEERING

by

Shreyash Gupta (20102124)

Shreyash Divekar (20102103)

Dishank Jain (21202014)

Pratham Jain (20102072)

Guide:

Prof. Merlin Priya Jacob

Department of Computer Engineering

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

(2022-2023)

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

CERTIFICATE

This is to certify that the Mini Project 2B entitled "**VIRTUAL INSCRIBER"** is a bonafide work of **Shreyash Gupta (20102124), Shreyash Divekar (20102103), Dishank Jain (21202014), Pratham Jain (20102072)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Engineering.**

Guide:	Project Coordinator:	Head of Department
Prof. Merlin Priya Jacob	Prof. D.S. Khachane	Prof. S.H. Malave

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

Project Report Approval for Mini Project-2B

This project report entitled "**VIRTUAL INSCRIBER**" by **Shreyash Gupta, Shreyash Divekar, Dishank Jain, Pratham Jain** is approved for the partial fulfillment of the degree of ***Bachelor of Engineering*** in ***Computer Engineering***, ***2022-23***.

Examiner Name	Signature
1.__________________	______________
2.___________________	______________

Date:

Place:

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Shreyash Gupta (20102124)

Shreyash Divekar (20102103)

Dishank Jain (21202014)

Pratham Jain (20102072)

Date:

Abstract

Virtual Inscriber allows users to create and manipulate annotations using hand movements. It utilizes motion tracking technology to detect and interpret the user's movements, enabling them to draw lines of different colors in real-time without the need for a physical surface. This abstract highlights the key features and potential applications of Virtual Inscriber, including its portability, ease of use, and potential to enhance productivity in a range of fields, such as education, corporate and design. Professionals from the teaching field can use Virtual Inscriber while teaching their subject. During the period of pandemic online teaching was predominantly held, while teaching online teachers often used annotations on shared screens/virtual boards for better explanations. These annotations were only possible through a mouse or some hardware connected to the computing device, therefore teachers had no choice but to stay close to the desktop while teaching. In order to surpass this limitation Virtual Inscriber aims to make annotations possible through hand movements. Virtual Inscriber takes the input of live captured frames from the face camera of the desktop, it detects the color frames and traces them in order to perform the annotations, here the user must have the specific color in their hands to be detected. Virtual Inscriber enables the teachers to move freely within the range of the camera instead of being stationed to one place. The project uses this gap in developing motion-to-text converters which can serve as software for smart wearable devices for writing in the air. Writing is an integrated form of communication that can convey our thoughts. Typing and writing are standard ways to record information today. Letters or words are written in a relaxed space by marker or finger. These wearable devices can see and understand our actions . A computing process that attempts to recognize and interpret human gestures through the use of mathematical algorithms is known as gesture recognition. The program will use computer vision to track finger movement.The generated text can also be used for various purposes, such as texting, emails, etc.

Keywords - Gesture recognition , Annotations, Virtual inscriber.

CONTENTS

LIST OF FIGURES

Chapter 1
Introduction

Virtual Inscriber is a project that aims to provide a unique experience for professionals from teaching background, by allowing them to annotate on their live desktop/laptop using nothing but the air around them. The project uses machine learning algorithms to track the movements of the user's hand, allowing them to draw on a virtual canvas with a variety of different brushes and colors.

The main objective of the Virtual Inscriber project is to provide a new and innovative way for teachers to present their subjects or conduct their lectures. The project is designed to be accessible to anyone, regardless of their drawing ability or experience, and to be a fun and engaging way to explore technology.

One of the key features of the Virtual Inscriber project is its ease of use. The system is designed to be intuitive and easy to navigate, with a simple interface that allows users to select different brushes and colors, adjust the size and opacity of their brush strokes, and erase mistakes as needed. Another important aspect of the Virtual Inscriber project is its use of machine learning technology. The system is able to analyze the movements of the user's hand in real time, allowing it to track the position and orientation of the brush and adjust the stroke accordingly. This means that users can create fluid, natural-looking lines and shapes, without having to worry about the technical details of how to achieve them.

Overall, the Virtual Inscriber project brings up a new innovative way of presenting in the education/corporate sector. Its use of machine learning technology and intuitive interface make it accessible to a wide range of users, and its unique approach to presentations is sure to inspire creativity and experimentation. As the project continues to evolve and improve, it has the potential to become an essential tool for anyone who teaches or presents something to people, and anyone else looking to explore their creative potential.

Chapter 2
Literature Survey

1. Steve Dipaola (2016), “Using Artificial Intelligence Techniques to Emulate the Creativity of a Portrait” Conference: Electronic Visualisation and the Arts DOI:10.14236/ewic/EVA2016.32
This paper presents AI based methods for their operating principles and outputs that can be relevant to the field of computational creativity research and computational painterly rendering. Research uses several artificial intelligence techniques including Genetic Algorithms, Neural Networks and Deep Learning neural networks in an attempt to begin to understand and emulate this creative process. [1]

2. Saira Beg, M. Fahad Khan and Faisal Baig, "Text Writing in Air" Journal of Information Display Volume 14, Issue 4, 2013 ,DOI:10.1080/15980316.2013.860928 December 2013 Journal of Information Display 14(4)
This paper proposed a method tested on a Samsung Galaxy S3 android mobile phone. The proposed algorithm showed an average accuracy rate of 92.083% when tested for different shaped alphabets. More than 3000 different shaped characters were used. Here, more than 3000 different shaped characters were used. Our proposed system is the software based approach and relevantly very simple, fast and easy. It does not require sensors or any hardware rather than camera and red tape. Moreover, proposed methodology can be applicable for all disconnected languages but having one issue is that it is color sensitive in such a way that existence of any red color in the background before starting the character writing can lead to false results.[2]

3. G Chandan, “Real Time Object Detection and Tracking Using Deep Learning and OpenCV” ISBN:978-1-5386-2456-2,vol.12, no.12 jan 2020
This reference describes popular object detection algorithms as Region-based Convolutional Neural Networks (RCNN), Faster-RCNN, Single Shot Detector (SSD) and You Only Look Once (YOLO). Amongst these, Faster-RCNN and SSD have better accuracy, while YOLO performs better when speed is given preference over accuracy. Deep learning combines SSD and Mobile Nets to perform efficient implementation of detection and tracking. This algorithm performs efficient object detection while not compromising on the performance.[3]

4. S Guennouni, A Ahaitouf and A Mansouriss , “Multiple object detection using OpenCV on an embedded platform”2014 Third IEEE International Colloquium in Information Science and Technology (CIST), 2014, pp. 374-377.
The proposed application deals with real time systems implementation and the results give an indication of where the cases of object detection applications may be more complex and where it may be simpler. With the advancement in the video surveillance and image processing, object detection has known a rising interest in the computer visualization industry. However, achieving high performance and near-real-time object detection is a key concern in both large-scale systems and embedded platforms. Therefore, a reliable and accurate near real-time object detection application, running on an

embedded system, is crucial, due to the rising security concerns in different fields. [4]

Overall, these papers and articles demonstrate the potential of Python and AI algorithms for creating virtual canvas with various styles and techniques.

Research Paper	ANALYSIS
1. Steve Dipaola (2016), ‘Using Artificial Intelligence Techniques to Emulate the Creativity of a Portrait Painter DOI:10.14236/ewic/EVA2016.32 Conference: Electronic Visualisation and the Arts.	AI based methods for their operating principles and outputs that can be relevant to the field of computational creativity research and computational painterly rendering
2. Saira Beg, M. Fahad Khan and Faisal Baig, "Text Writing in Air," Journal of Information Display Volume 14, Issue 4, 2013	The proposed method tested on a Samsung Galaxy3 android mobile phone. The proposed algorithm showed an average accuracy rate of 92.083% when tested for different shaped alphabets. More than 3000 different shaped characters were used. .
3. G Chandan, “Real Time Object Detection and Tracking Using Deep Learning and OpenCV” IEEE Xplore Compliant CFP18N67-ART; ISBN:978-1-5386-2456-2	This reference describes popular object detection algorithms are Region-based Convolutional Neural Networks (RCNN), Faster-RCNN, Single Shot Detector (SSD) and You Only Look Once (YOLO).
4. S Guennouni, A Ahaitouf and A Mansouriss ,“Multiple object detection using OpenCV on an embedded platform”, 2014 Third IEEE International Colloquium in Information Science and Technology (CIST), 2014, pp. 374-377.	The proposed application deals with real time systems implementation and the results give an indication of where the cases of object detection applications may be more complex and where it may be simpler.

Chapter 3
Problem Statement, Objective & Scope

Problem Statement: -

The problem that Virtual Inscriber aims to solve is the limitation of traditional annotations using mouse or any other hardware device connected to desktop/laptop. Many times these limitations present a problem for the presenter as it forces the presenter to be stationed at one place near the computing device for doing annotations. Additionally, these tools often lack the tactile feedback and natural feel of presenting and explaining, it is human nature to do hand gestures while explaining or presenting something. Virtual Inscriber seeks to address these limitations by providing a platform for presentations that is accessible, intuitive, and natural, using machine learning technology to track the movements of the user's hand and create fluid, natural-looking strokes on a virtual canvas. By doing so, Virtual Inscriber aims to inspire creativity and exploration, and provide a new and innovative way for people to express themselves.

Objective: -

- Enable user to effortlessly annotate on the screen of the device using hand movements
- To create a virtual canvas as an interface for performing annotations.
- To detect the hand movements as a color marker.
- Perform morphological operations to increase accuracy of annotations.
- **Learning from existing art**: The Virtual Inscriber could be trained on a large corpus of existing art to learn and replicate the styles and techniques of various artists. This could include classical art, modern art, and contemporary art, among others.
- **Creativity and innovation:** While replicating existing styles is important, a Virtual Inscriber could also be designed to generate new and innovative artwork. This could involve incorporating randomness or other algorithms that allow the AI to create unique pieces of art that are not based on any particular style.
- **User interactivity:** A Virtual Inscriber could be designed to interact with users in real-time. Users could provide feedback on the artwork generated by the AI and make suggestions for modifications, leading to a collaborative creation process.
- **Commercial applications:** A Virtual Inscriber could have commercial applications in the design, marketing, and advertising industries. For example, the AI could be used to generate artwork for branding and marketing campaigns or to create custom designs for clients.

Scope: -

- To ensure that the interface is very simple and easily understandable by the user.
- The user should be able to draw what he wishes to draw without any interruptions.
- In the future, this is useful for making kids learn drawing in schools in an interactive way.
- Also, the scope of this system is mainly used as a powerful means of communication for the deaf,which means implementing this project can help in:
- It helps people with hearing impairments to communicate well.
- Various purposes, such as sending messages, e-mails, etc., as the generated text can also be used for that.
- The user will be able to annotate in multiple colors through the color palette provided in the frame of face camera
- The user will be able to clear all the annotations using a clear all button provided with the color palette on the frame of face camera
- Besides the face camera frame window the user will be provided with three more windows
- A whiteboard window, this window will replicate all the annotations done on the camera window, these annotations will be visible on a plain whiteboard
- A mask window, this window will show the mask generated of the input taken from camera frame, we will be able to notice the morphological operations performed on the mask window
- A track bar window, this window will decide which color is to be detected for tracing, the user will be able to change the color by manipulating the values of track bar.

Chapter 4

Proposed System Architecture

In order to overcome the current limitations of annotations through physical touch, the project is designed to enable user to perform annotations through hand movements i.e without touching anything physically.Virtual Inscriber is a system that allows users to create annotations using natural hand gestures and movements in the air, without the need for physical tools such as a pen or tablet. The system utilizes motion capture technology to track the user's hand movements and translate them into digital strokes on a virtual canvas.

The system consists of two main components: a hardware component and a software component. The hardware component includes a camera or sensor array that captures the user's hand movements and transmits them to the software component. The software component includes an application that processes the data from the sensor and generates the corresponding annotations.To use the system, users simply need to stand in front of the sensor or camera and make natural hand gestures and movements in the air. The system will detect these movements and convert them into digital strokes on the virtual canvas. Users can select different colors and brush sizes to customize their annotations, and can also save and export their annotations to share with others.One of the key benefits of Virtual Inscriber is its intuitive and natural user interface. Unlike traditional drawing tools that require specialized training and equipment, Virtual Inscriber allows users to draw using familiar hand gestures and movements. This makes the system accessible to a wider range of users, including children and individuals with disabilities.

Overall, Virtual Inscriber represents an innovative and exciting approach to digital drawing, with the potential to revolutionize the way we think about and interact with digital media.

- ## Architecture / Block Diagram:-

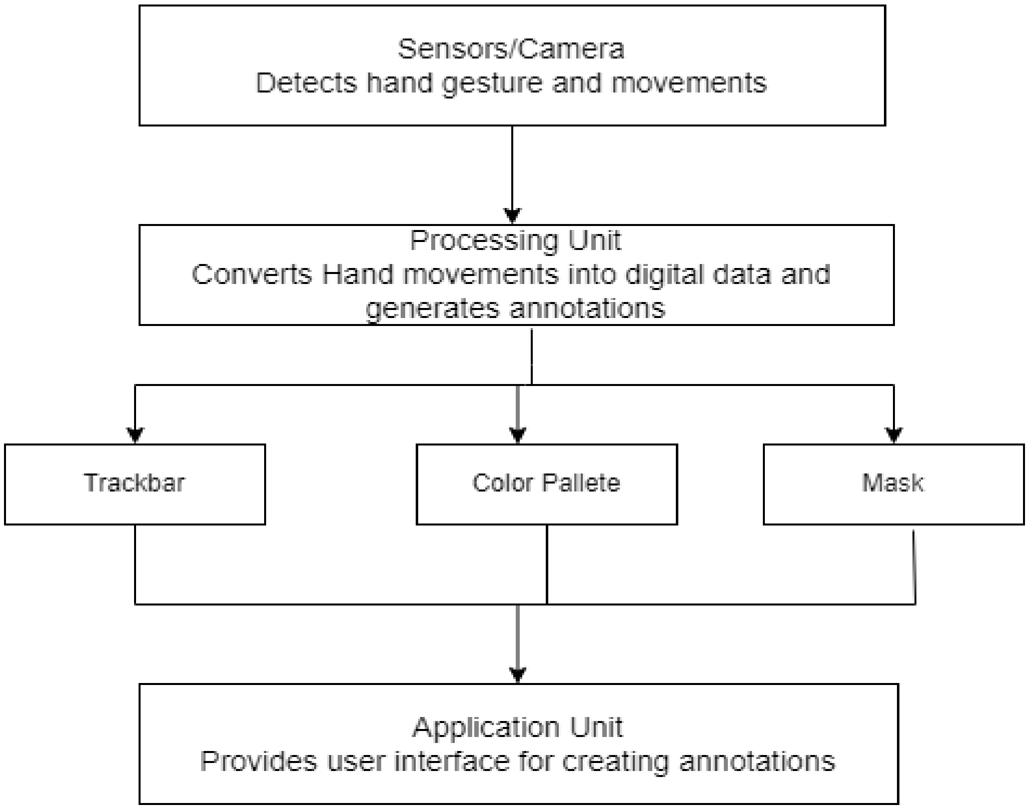


Fig 4.1 Architecture diagram

The architecture diagram shows the three main components of the Virtual Inscriber system:

Sensor/Camera: This component captures the user's hand gestures and movements and sends them to the processing unit. The sensor could be a camera or an array of sensors that track hand movements.

Processing Unit: This component converts the hand gestures and movements into digital data and generates the corresponding annotations. It may also perform additional processing, such as filtering or smoothing the data to improve accuracy.

Application: This component provides the user interface for creating annotations using the Virtual Inscriber system. Users can select different colors and brush sizes.

Overall, the Virtual Inscriber system uses a modular architecture that separates the input, processing, and output components. This design enables the system to be easily customized and adapted to different hardware and software configurations.

- **Data Flow Diagram (Level 0 & Level 1):-**

DFD 0

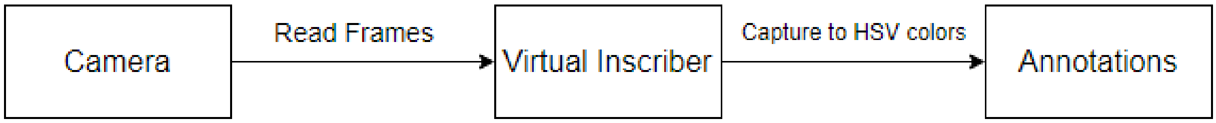


Fig 4.2.1 Data Flow Diagram (DFD 0)

The above Data flow diagram represents the procedure of operations performed by Virtual Inscriber. DFD 0 gives us a very abstract view of Virtual Inscriber, it tells us about the function of Virtual Inscriber.

DFD 1

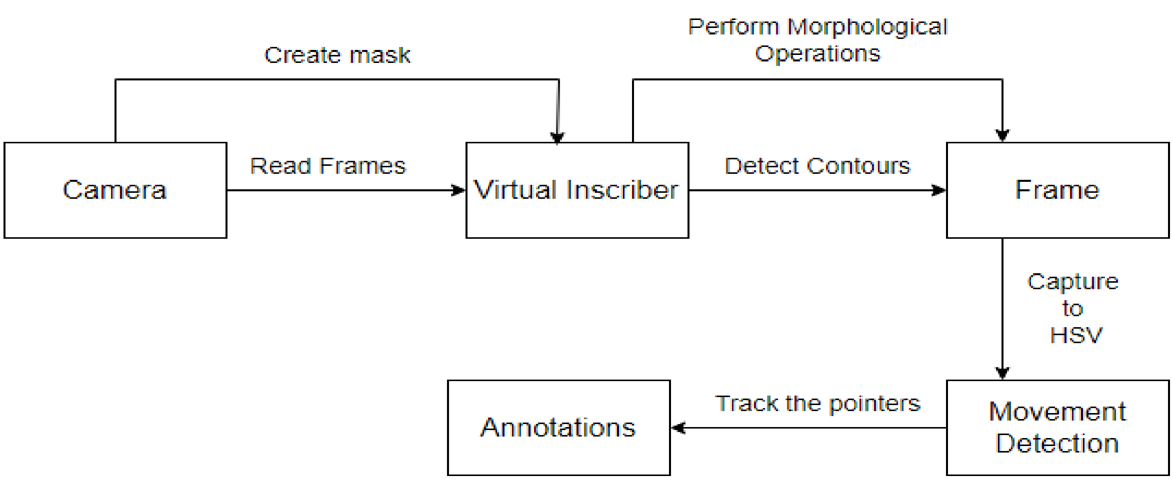


Fig 4.2.1 Data Flow Diagram (DFD 1)

DFD level 1 goes into details of functioning; it includes various components like frames and movement detection used in Virtual Inscriber.

- **Use Case Diagram:-**

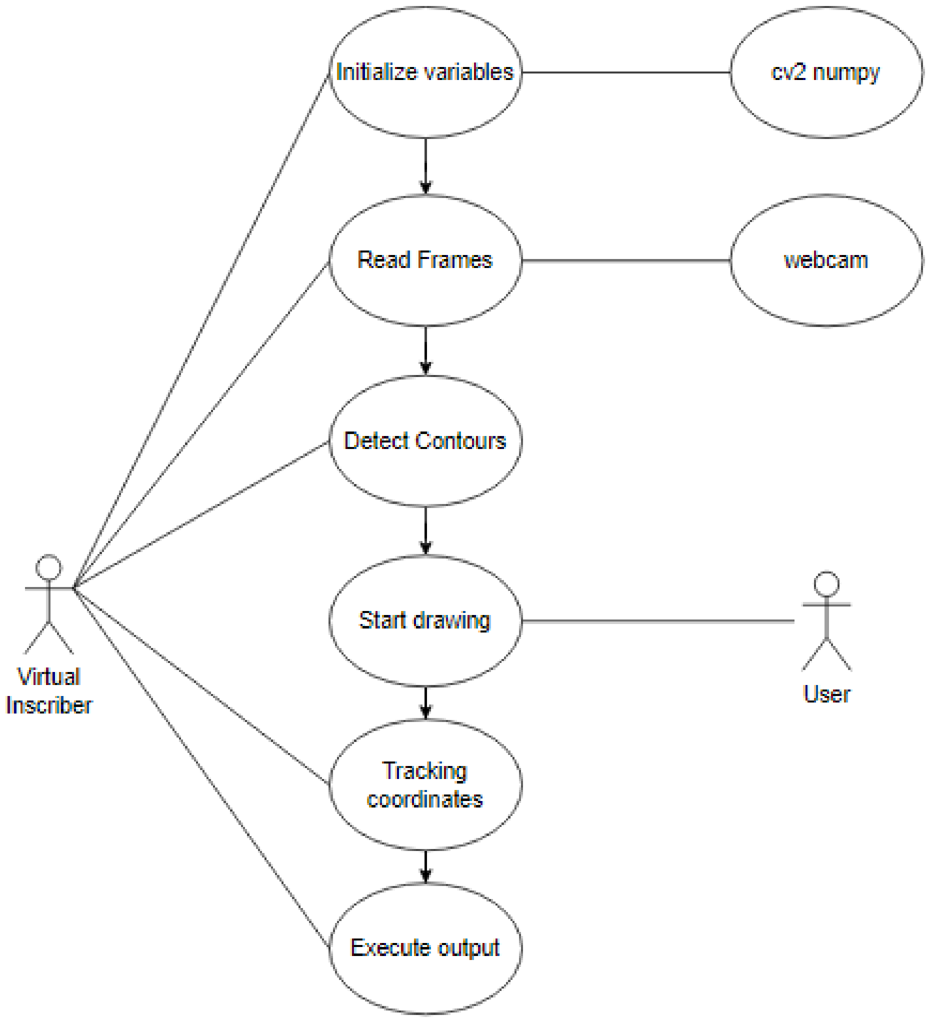


Fig 4.3 Use Case Diagram

The use case diagram in fig 4.3 provides us with the information on how different entities of Virtual Inscriber interact with each other and coordinate to fulfill the objective of Virtual Inscriber. There are two basic actors involved namely Virtual Inscriber (software) and the User, Virtual inscriber deals with multiple components as it has to perform the operations. The user comes into picture only at user interface while using Virtual Inscriber, where it annotates using some hand movements.

- **Sequence Diagram :-**

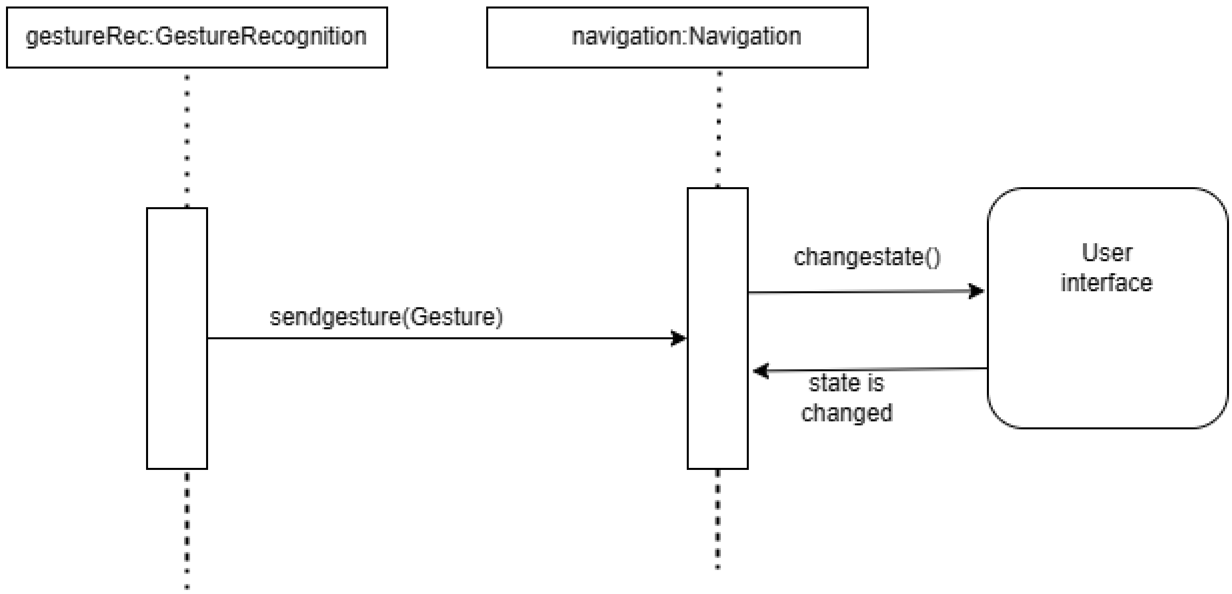


Fig 4.4 Sequence Diagram

The above figure 4.4 revolves around the gesture recognition and tracing of the hand movements in order to make the annotations possible. The sequence diagram starts with gesture recognition detecting and sending gesture information to the navigation component where the navigation component changes the state based upon the gestures received from the gesture recognition components.

The user interface component receives the changestate() message from navigation component and acts upon it by performing the changes and responds with the state changed message to the navigation component

- **Activity Diagram:-**

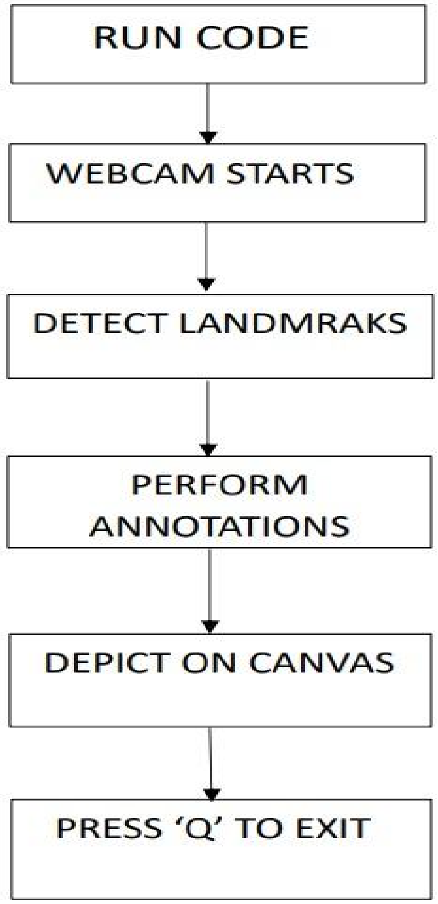


Fig 4.5 Activity Diagram

The activity diagram tells us about the flow of functions performed by Virtual Inscriber. The flow starts with the initiation of the webcam and then the picture is being monitored live. The contours of the detected hand movements are assigned on the basis of which they are traced. The traced lines are converted into HSV color annotations with help of OpenCV. The annotations are depicted on designed canvas, canvas consists of the color palettes provided with help of OpenCV functions. This color palettes are used with the help of hand movements. The user is able to select a specific color by hovering his hand near its color button.

Chapter 5
Project Planning

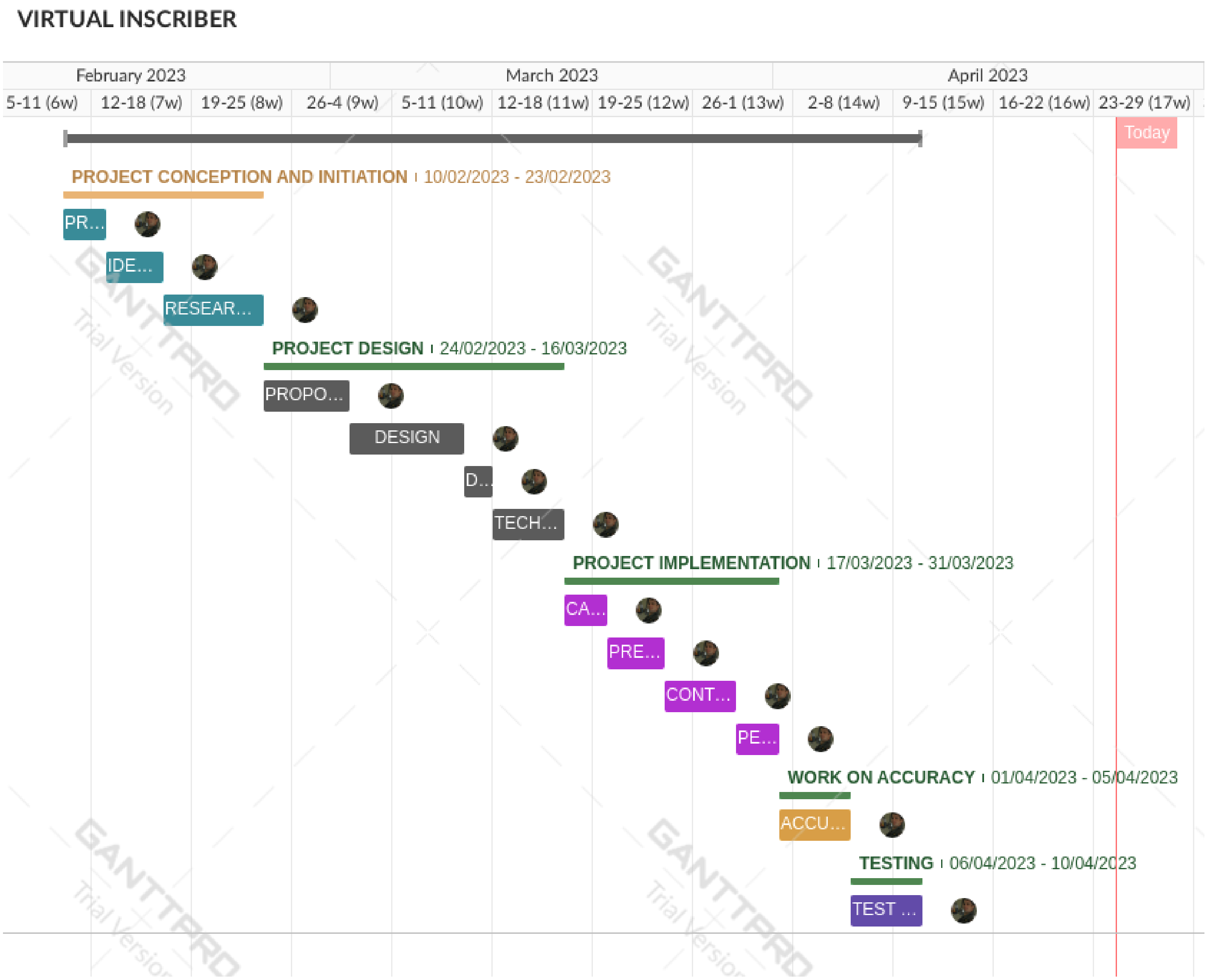


Fig 5.1 Gantt Chart

Chapter 6
Experimental Setup

Software Requirements: -

1) Operating System: Virtual Inscriber is compatible with Windows, macOS, and Linux operating systems.
2) Graphics Processing Unit (GPU): A dedicated graphics processing unit (GPU) is required for Virtual Inscriber to perform optimally.
3) Motion Tracking Software: Virtual Inscriber relies on motion tracking software to detect and interpret the user's hand gestures. Therefore, it requires motion tracking software that is compatible with the operating system being used.
4) API Libraries: Virtual Inscriber uses application programming interface (API) libraries for its motion tracking technology. These libraries need to be installed and configured correctly for Virtual Inscriber to function properly.
5) Internet Connection: An internet connection is required to download and install Virtual Inscriber, as well as to receive software updates.

Hardware Requirements: -

6) CPU: Virtual Inscriber requires a processor with a clock speed of at least 2.4 GHz. A multi-core processor is recommended for better performance.
7) GPU: A dedicated GPU with at least 2 GB VRAM is required for Virtual Inscriber to perform optimally. The GPU should support OpenGL 3.2 or later.
8) RAM: A minimum of 4 GB RAM is required for Virtual Inscriber to function. However, 8 GB or more is recommended for optimal performance.
9) STORAGE: Virtual Inscriber is compatible with Windows, macOS, and Linux operating systems.
10) OS: Virtual Inscriber is compatible with Windows, macOS, and Linux operating systems.

Chapter 7
Implementation Details

1. HAND TRACKING USING OPENCV

Hand tracking is based on color detection rather than contour detection. This involves selecting a specific range of colors that correspond to the hand and creating a binary mask of the hand region based on those colors. The contours of the hand can then be detected in the binary mask and tracked using the steps outlined above.OpenCV provides several functions and algorithms that can be used for hand tracking, including contour detection, feature detection, and machine learning-based approaches. The choice of algorithm depends on the specific requirements of the application, such as real-time performance, accuracy, and robustness to lighting conditions and occlusions.

2. TRACKBARS

When the trackbars are arranged, we will get the realtime esteem from the trackbars and make a range. This reach is a numpy structure which is utilized to be passed in the capacity cv2.inrange(). This capacity returns the Mask on the hued object. This Mask is a high contrast picture with white pixels at the situation of the ideal tone.

3. CONTOUR DETECTION

Recognizing the Position of Colored item at fingertip and shaping a circle over it. We are playing out some morphological procedure on the Mask, to make it liberated from contaminations and to distinguish shape without any problem. That is Contour Detection.

4. FRAME PROCESSING

Following the fingertip and drawing focuses at each position for air material impact. That is Frame Processing.

5. ALGORITHMIC OPTIMIZATION

Making the code efficient to work the program without a hitch. Algorithmic Optimization.

6. ANNOTATE USING OPENCV

OpenCV provides a wide range of tools and functions for annotating images and video streams, and the specific technique used depends on the requirements of the application. By using these techniques, you can create informative visualizations and improve the accuracy and interpretability of your computer vision system.

Functionality includes:-

Drawing primitives: OpenCV provides several functions for drawing basic geometric shapes on an image, such as lines, rectangles, circles, and polygons. These can be used to highlight specific regions of interest or indicate the location of detected objects.

Color mapping: In some cases, it can be helpful to apply a color map to an image to visualize certain features or properties. OpenCV provides functions for applying color maps, such as cv2.applyColorMap(), which can be used to highlight temperature variations, depth information, or other characteristics of the image.

Overlaying images: Another way to annotate an image is to overlay it with another image or mask. For example, you can use a mask to highlight the pixels that correspond to a detected object or use an image to indicate the orientation or pose of the object.

Image processing: OpenCV provides a range of image processing functions such as filtering, edge detection, morphology operations, thresholding, histogram equalization, and many more.

Object detection and tracking: Object detection and tracking are important computer vision tasks that can be accomplished using OpenCV, a popular open-source computer vision library. OpenCV provides various algorithms and tools for object detection and tracking, including Haar Cascades, HOG (Histogram of Oriented Gradients), and deep learning-based approaches

Feature detection and extraction: OpenCV includes algorithms for detecting and extracting local features in images, such as SIFT (Scale-Invariant Feature Transform), SURF (Speeded Up Robust Features), and ORB (Oriented FAST and Rotated BRIEF).

Camera calibration and stereo vision: OpenCV provides tools for calibrating cameras and computing stereo vision, which is useful for applications such as 3D reconstruction and object recognition.

Machine learning: OpenCV includes machine learning algorithms such as support vector machines, k-nearest neighbors, and decision trees, which can be used for various image and video processing tasks.

Chapter 8
Result

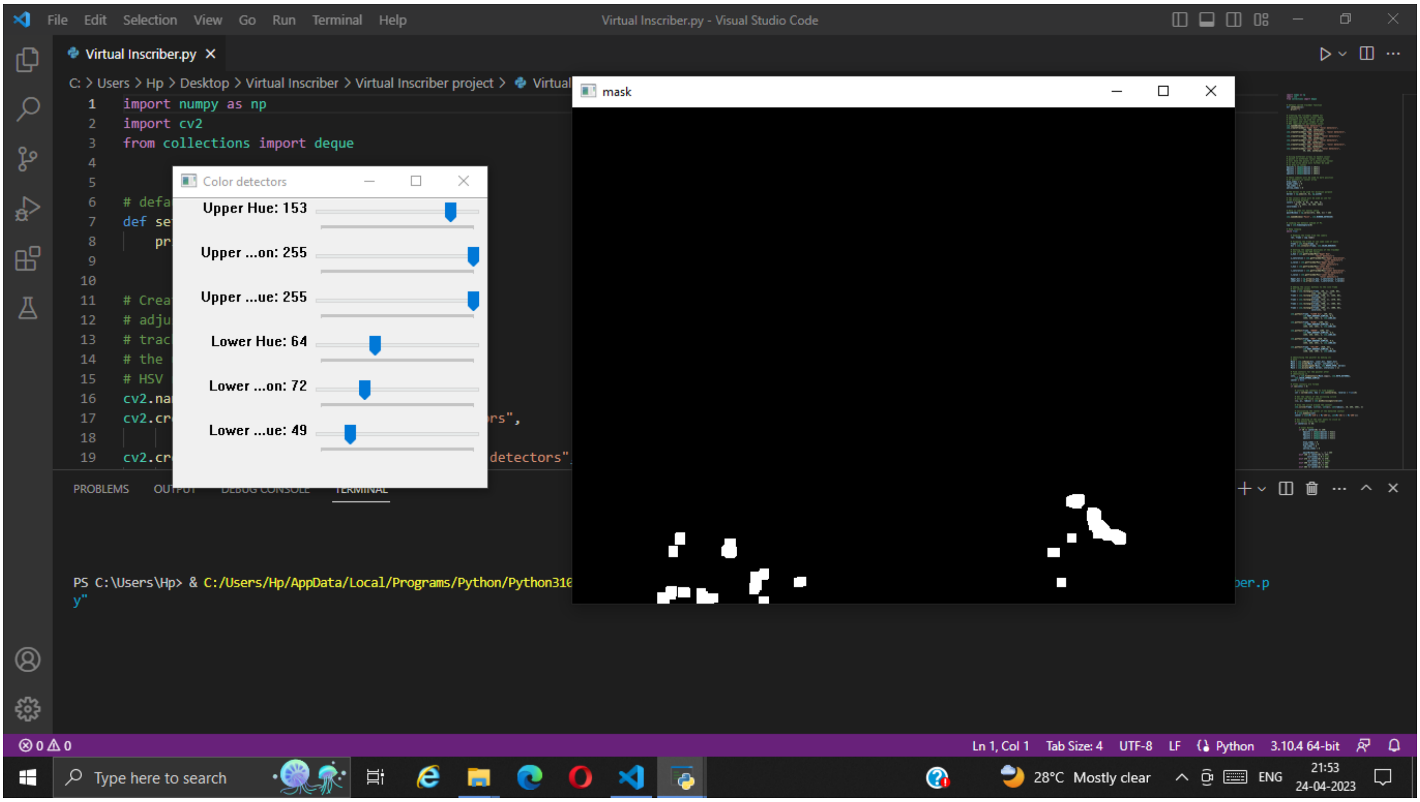
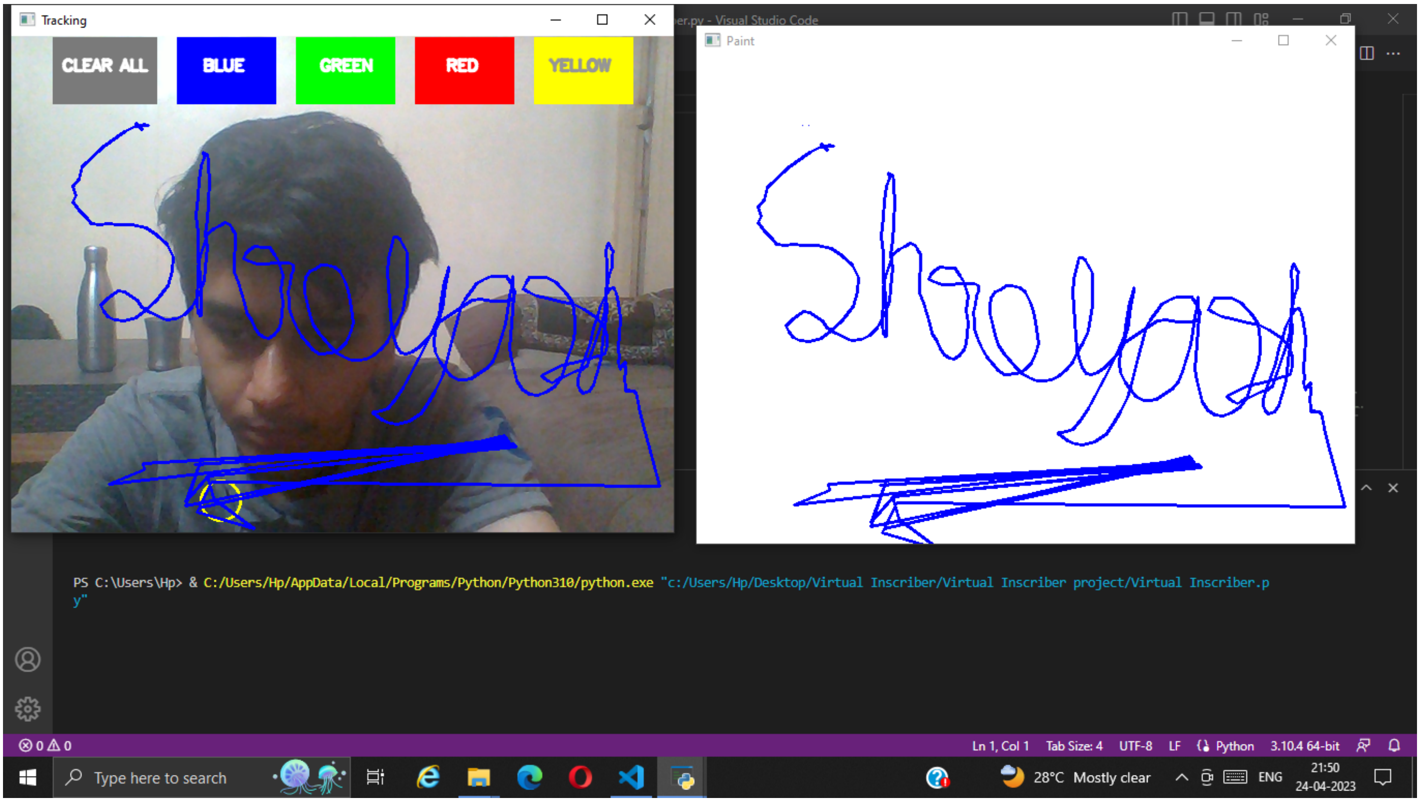


Fig 8.1 Mask and Trackbar

The above screenshot shows mask and trackbar windows, the mask is used for monitoring and performing morphological operations. The trackbar is used to set the values of a specific color which is to be detected and traced.



This screenshot displays two windows, the camera frame and the whiteboard frame. The user performs annotations with hand movements on the camera frame and those annotations are replicated on the whiteboard

Chapter 9
Conclusion

To avoid the use of mouse and difficulty to draw using it in the existing systems , this project Virtual Inscriber helps us a lot. We can easily draw or present our imagination just by waving our hand . This can be used in different aspects like teaching, drawing etc. This helps us to reduce the use of hardware components like mouse, touch screen etc. This can also be used as a base project for various systems that require hand tracking. The project discussed in this paper also helps to improve creativity in people. This helps us to teach and draw easier than earlier. In future, we can also use this project as a base project for many other hand tracking projects. We can also use this in sign language detection , virtual mouse etc. In the future, progress on Artificial Intelligence will improve the efficiency of writing in the air.

With Virtual Inscriber being a base model it can be further developed to generate transcripts of all annotations performed, these transcripts can be stored in a text file in the device of the user or can be provided as a downloadable file at the end of each session. For the project to be a real world working model Virtual Inscriber will have to be integrated with operating systems of user's devices so that while screen sharing the user will be able to use Virtual Inscriber as a tool for live annotations on the shared screen. The project can be further expanded by doing collaborations with online meeting services, giving the utility of Virtual Inscriber as an additional feature to the online meeting services.

The project of Virtual Inscriber has led us to understand the concepts of live color detection and live movement tracking, in the process we got a practical experience of Computer Vision as a field and we understood the usage of the OpenCV library.

Chapter 10
References

[1] Steve Dipaola (2016), ‘Using Artificial Intelligence Techniques to Emulate the Creativity of a Portrait Painter DOI:10.14236/ewic/EVA2016.32 Conference: Electronic Visualisation and the Arts.

[2] Saira Beg (2013), M. Fahad Khan and Faisal Baig, "Text Writing in Air," Journal of Information Display Volume 14, Issue 4.DOI:10.1080/15980316.2013.860928 December 2013
 Journal of Information Display 14(4)

[3]G Chandan, “Real Time Object Detection and Tracking Using Deep Learning and OpenCV”IEEE Xplore Compliant Part Number:CFP18N67-ART; ISBN:978-1-5386-2456-2.

[4] S Guennouni, A Ahaitouf and A Mansouriss , “Multiple object detection using OpenCV on an embedded platform”, 2014 Third IEEE International Colloquium in Information Science and Technology (CIST), 2014, pp. 374-377.