Magic Paint Board

A Project Report

Submitted by

ANIMESH GUPTA - 191B044

ANIRUDH DAMLE - 191B045

ASHISH PALIWAL – 191B074

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHONOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY

GUNA , MADHYA PRADESH (INDIA) – 473226

May 2022

DECLARATION

We hereby declare that the project entitled “Magic Paint Board” submitted for the B. Tech. (CSE) degree is our original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Anirudh Damle

Animesh Gupta

Ashish Paliwal

Place: Guna

Date:

CERTIFICATE

This is to certify that the project titled" Magic Paint Board " is the bona fide work carried out by Animesh, Anirudh & Ashish ,a student of B Tech (CSE) of Jaypee University of Engineering and Technology, Guna (M.P) during the academic year 2021-2022, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Signature of the Guide

Place: Guna

Date:

Abstract

This project is to describe about the hand gesture based on user interface that would able to replace mouse. In this project, we would focus to new system which we will use own hand to replace mouse and control the mouse pointer to do all mouse tasks in the computer via webcam. We would discuss about what advantages of system among existing system through several ways and more convenient than usual system. The technology that used is related in using Webcam and OpenCV libraries. We have been using Rapid Application Development (RAD) in developing my project.

There are four phases in this model: Requirement Planning, User Design, Construction Phase, and Cutover Phase. Rapid Application Development is slightly different from the Systems development life cycle (SDLC).

The Rapid Application Development is focus more on implementation and construction but less on design and documentation. Last but not least, we have arranged a time schedule to help me in developing the project so that we would be able to finish my project in time.

Acknowledgment

We would like to express our gratitude and appreciation to all those who gave us the opportunity to complete this project. Special thanks to our supervisor **Mr. Navaljeet Singh Arora** whose help, stimulating suggestions and encouragement helped us in all the time of development process and in writing this report. We also sincerely thanks for the time spent proofreading and correcting my many mistakes.
We would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.

Thanking you

Animesh Gupta
Anirudh Damle
Ashish Paliwal

LIST OF FIGURES

Table of content

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

This project is to describe about the hand gesture based on user interface that would able to replace mouse. In this project, we would focus to new system which we will use own hand to replace mouse and control the mouse pointer to do all mouse tasks in the computer via webcam. We would discuss about what advantages of system among existing system through several ways and more convenient than usual system. The technology that used is related in using Webcam and OpenCV libraries. We have been using Rapid Application Development (RAD) in developing my project.

There are four phases in this model: Requirement Planning, User Design, Construction Phase, and Cutover Phase. Rapid Application Development is slightly different from the Systems development life cycle (SDLC). The Rapid Application Development is focus more on implementation and construction but less on design and documentation. Last but not least, I have arranged a time schedule to help me in developing the project so that I would be able to finish my project in time.

In the current computer age, the common input devices to operate a computer are the mouse and the keyboard. Without them, a computer is just a monitor.

Besides that, using a mouse can be hazardous. Some people might need to make small, exact movements with their hand, fingers, and thumb when they are running a program, such as Photoshop. Furthermore, by positioning, scrolling, and clicking the mouse repeatedly, the muscle can become tired and overworked.

1.2 Project overview

In this time and age, technology is constantly achieving newer heights. The computer is no exception, too. The traditional user interface devices, namely the mouse and the keyboard are replaced with even better devices, such as the touch screen. The development of new technologies has realized and focused on the natural interaction between computer and human. Natural interaction consists of eyes, face, speech, and so on. But what device would feel more natural and intuitive than the hand gestures of users?

Hand gestures can be a new language to communicate with computer. Hand gesture is a new user-interface system which opens up a new direction for people to build the advanced device. For a successful communication, a sender and receiver must have the same set of information for particular gesture. Gesture is defined as a movement of body parts which is has certain message, to be communicated precisely between a sender and receiver. So that, message from each gesture is very important to understand what the computer need to do.

In the proposed system, the mouse pointer is controlled by user's hand by using hand gesture. Before that, the computer has to understand the instructions of each hand gestures. Through tracking and recognition by the computer, interaction between human and computer can be achieved.

1.2.1 Gesture and Voice Based interactive System

Gesture and Voice Based Interactive System is an interaction system which is aimed at the simulation of mouse function certain gestures of the fingers and also using various voice tags. In this system, they applied color tapes on the fingers and use gesture detection and color tracking technique to simulate mouse functions. They use Java Sphinx framework for simulation of the mouse events besides using voice commands.

1.2.2 Hand gesture Recognition Using Computer Vision

Hand gesture recognition using computer vision is a system which allows user to interact with the computer by using video camera to interpret one-handed side language alphabet and number gestures. The systems use an optical method which is more practical, cost effective, and has no moving parts. The hardware chosen in this system is a single-color camera pointing down towards a desk surface a constant color with no special lighting. Detection of hand gestures are performed by comparing each RGB pixel values with ranges found.

1.2.3 Mouse Simulation using Two Colored tapes

A Mouse Simulation Using Two Colored tape is a system which can also be identified as ubiquitous computing. This system uses two different color tapes on the fingers. One of the color tapes is used for controlling cursor movement while the relative distance between the two-colored tapes will be used for click events of the mouse. The system uses only webcam to completely eliminate the mouse.

1.2.4 Current and limitation

Currently, there are many innovations that could replace the usual mouse to enable communication with the computer. Some of the existing systems consist of different small devices, such as gloves and color tapes. Needless to say, it is hard to operate a computer without any physical devices, other than the hands. In order to avoid situations like this, hand gesture based on user interface is another alternative to allow users to control the mouse pointer to operate the computer without any physical distance required.

1.2.5 Terminology

Abbreviation	Definition
HGBUI	Hand gesture-based user -interface
UML	Unified Modelling Language
RGB	Red-Green-Blue
SRS	Software Requirement Specification
SDD	Software Design Description
RAD	Rapid Application Development
RUP	Rational Unified Process

1.2.6 Method approach

Table 1.1 below shows the comparison of process models that had been found on selecting the best process model for this project.

Table 1.1 Process Models

Model	Stages	Scenario	Resource
Waterfall Model	5 Stages; Plan, Analyze, Design, Construct, Implement	Smaller systems with clearly verified requirements	When the software definition is stable, requirements and implementation of the product are both very well-understood. When requirements are fixed.
RAD	4 Stages; Planning and Requirement Gathering, Design, Construction, Cutover	Smaller Systems	When a good visual prototype is needed. When client's involvement in an ongoing basis.
RUP	4 Stages: Inception, Elaboration, Construction, Transition	Small parts of system that will be integrated into larger system	When system need web-enabled program development

From the models above, Waterfall Model is the oldest style model in the field of software engineering development. However once implementation phase started, it is relatively impossible to go back previous step. Hence if there's any error or problem in the design phase, system will get very complicated in implementation phase. RAD is more focused on developing small projects or large projects broken into smaller scale. Due to the goal of making development fast RAD focusses less on design and documentation and more on construction. RUP on the other hand covers both Waterfall and RAD's shortcomings by implementing iterative development and focusing more on architecture and risk management (Rooney, 2008). However RUP could result to more documentation, more design, and less on the software or system itself.

We used to develop in project is RAD Model and it is a very suitable to develop a system in a time constraint or limited amount of time. This model involves user's input along the way through development, prototyping, and other processes that could speed up the system or application development. RAD consists of four stages which are Planning and Requirement Gathering, Design, Construction, and Cutover. The details on every phase will be explained further in this report.

1.2.7 Scope of project

This scope of this project is defined below;

User

The user of this system will be using his/her own hand to control mouse pointer based on the user-interface.

System

This system able to tracking and recognition of the user's hand.

This system will be able to understand the movement of gesture from the hands

This system able to use without physical distance required.

1.2.8 Outline of the Project

This report is divided into 3 parts. The first part is told about the introduction, problem statement, objective and scope of the entire project. Meanwhile, review on existing systems and current system and limitation are also cover in this part. Besides, this part also contains the method of approach and terminology.

In second part, the main focus is discussed about the methodology have been used to develop the system. In addition, this part also explaining the user requirements, flow chart, diagrams and the method use for the whole development process.

The last part is about the conclusion of the technical report. This part will conclude and summarize the important point for the entire project.

1.3 Hardware Specifications

Operating system

Windows 8 or later
macOS Sierra 10.12 or later
Ubuntu 14.04+,
Debian 8+,
openSUSE 13.3+, or
Fedora Linux 24+

Device	Specification
Laptop	Lenovo G400S --Black Processor Intel Core i5 - 3230M 2.6GHz Memory 4GB DDR3 RAM Storage 1TB Serial - ATA Hard Disk Graphics NVIDIA GeForce GT 720M 2GB Graphics Optical Drive DVD Writter Display 14" HD Display (16:9) Wide Screen Bluetooth Yes Camera Integrated Web Camera Communications WiFi 802.11b/g/n, Ethernet Port Ports 2 x USB 3.0 Port 1 x USB 2.0 Port 1 x HDMI 1 x VGA 1 x Headphone / Microphone Combo Card Slot Multi in One Card Reader Software Genuine Windows 8
Webcam	1MP Fixed Focus CMOS camera on the laptop

1.4 Software Requirements

Item	Specifications
Microsoft Window 8	Support Architecture: -32bit(x86) -64bit(x86) Hardware Requirement: -1 gigabyte (GB) RAM (32-bit) or 2 GB RAM (64-bit) -16 GB available hard disk space (32-bit) or 20 GB (64- bit) -DirectX 9 graphics device with WDDM 1.0 or higher driver
OpenCV	OpenCV 2.4.7
Microsoft office Home and Student 2010	-Microsoft office words 2010, -Microsoft office Power point 2010,
Microsoft Visual Studio 2010 professional	-64bits

CHAPTER 2

LITERATURE SURVEY

2.1 Existing system

There are many other applications present in the market but it is not using AI we have use hand gesture in this project, based on user interface that would able to replace mouse. In this project, we would focus to new system which we will use own hand to replace mouse and control the mouse pointer to do all mouse tasks in the computer via webcam. We would discuss about what advantages of system among existing system through several ways and more convenient than usual system. The technology that used is related in using Webcam and OpenCV libraries. We have been using Rapid Application Development (RAD) in developing my project.

2.2 Proposed system

In the proposed system, the mouse pointer is controlled by user's hand by using hand gesture. Before that, the computer has to understand the instructions of each hand gestures. Through tracking and recognition by the computer, interaction between human and computer can be achieved.

In this time and age, technology is constantly achieving newer heights. The computer is no exception, too. The traditional user interface devices, namely the mouse and the keyboard are replaced with even better devices, such as the touch screen. The development of new technologies has realized and focused on the natural interaction between computer and human. Natural interaction consists of eyes, face, speech, and so on. But what device would feel more natural and intuitive than the hand gestures ofusers?

Hand gestures can be a new language to communicate with computer. Hand gesture is a new user-interface system which opens up a new direction for people to build the advanced device. For a successful communication, a sender and receiver must have the same set of information for particular gesture. Gesture is defined as a movement of body parts which is has certain message, to be communicated precisely between a sender and receiver. So that, message from each gesture is very important to understand what the computer used to do.

2.3 Feasibility analysis

- Product safety: Product safety is an important factor in determining the technical and market feasibility of the idea.

- Market gap: Filling a viable gap in the market is key to the success of your new product or service.

- Keep R&D simple: Technical problems need to be considered at an early stage.

- Dependence on other: Dependent on linkages with other products or systems (e.g., hardware that requires software to drive it)

- Customer usage: Idea can be translated into a usable or user-friendly product or service. Goal should be to produce a functional product or service, and to produce it in a way that it becomes attractive to the consumer, within the price points the market can bear.

2.4 Algorithm and Workflow

This is the most exciting part of our system. Writing involves a lot of functionalities. So, the number of methods used for controlling the system is equal to these number of actions involved. The basic functionalities we included in our system are:

- 1. Writing Mode –
  In this state, the system will trace the fingertip coordinates and stores them.

- 2. Color Mode –
  The user can change the color of the text among the various available colors.

- 3. Clear Screen –
  Say if the user goes wrong, we need a function to add a quick backspace.

CHAPTER 3
SYSTEM ANALYSIS & DESIGN

3.1Use case diagram

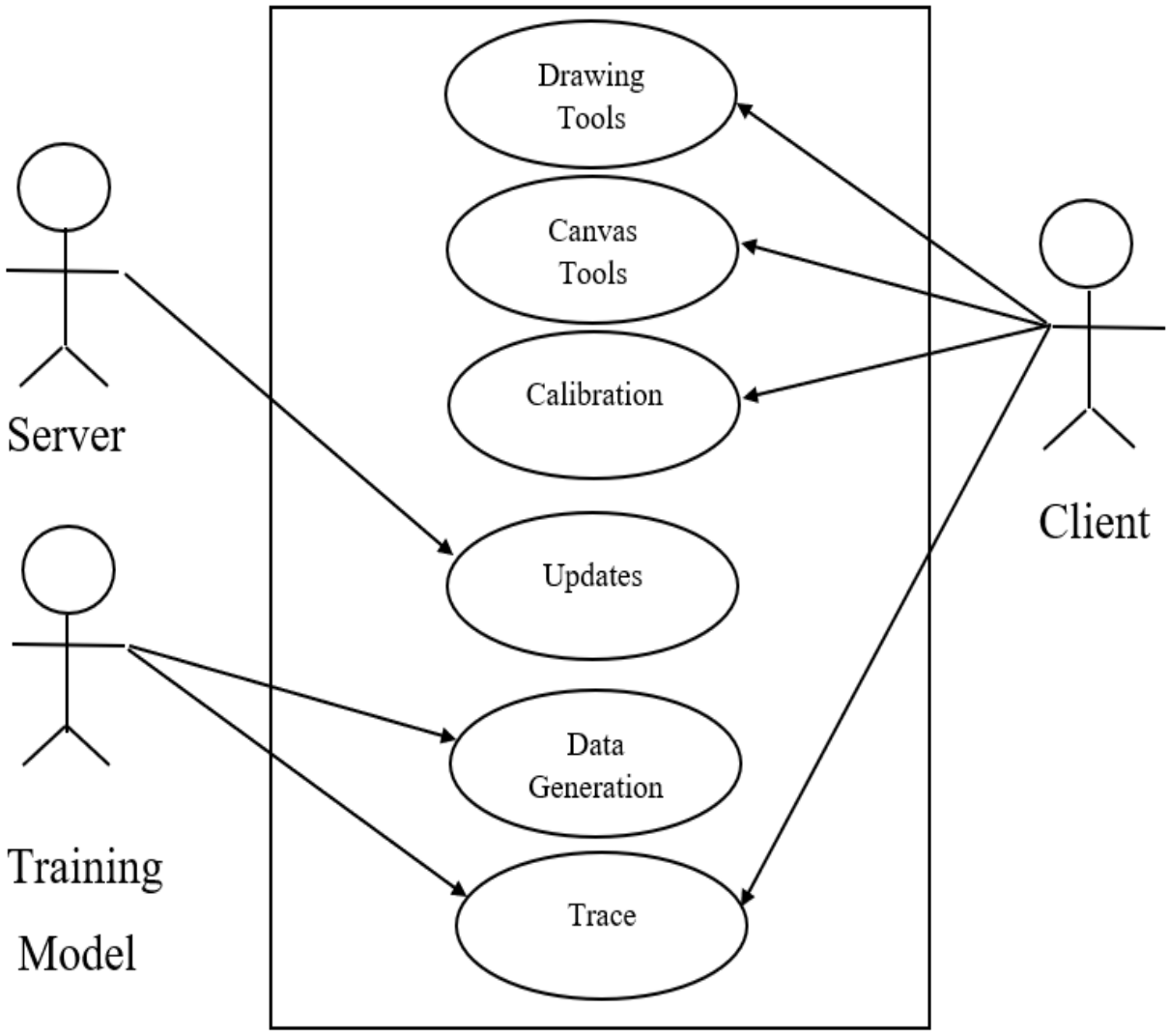

Fig 3.1

Explanation:

- Drawing Tools:
  Users can use the pen tool for writing on the screen. Also, they can use the 'Clear' button for the clearance of the entire screen.

- Canvas Tools:
  Users can select between multiple-colored pens available i.e., Red, Blue, Green and one reprogrammable color choice.

- Calibration:
  Manual calibration added for fine tuning the color detection in the contour.

- Updates:
  Users will experience periodic bug fixes and updates from the server.

- Trace:
  Masks the color detected in the environment to be used as a pen.

3.2 Sequence diagram

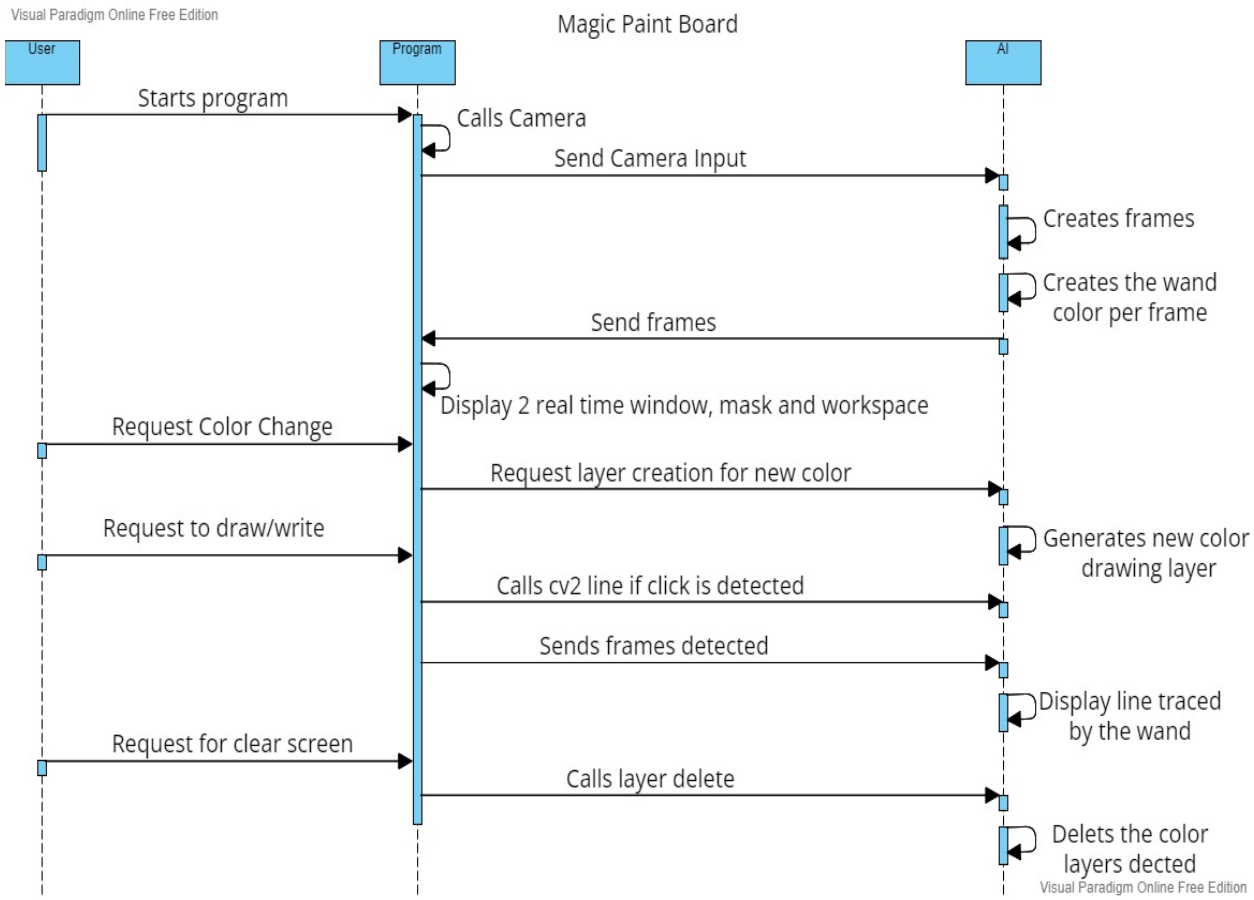

Fig 3.2

Explanation:

- User will start the program.
- Program will call the camera.
- Program will send camera input(data) AI.
- AI will create frames.
- Then AI will create the wand color per frame.
- AI will then send frames to program.
- Program will display in 2 real time window, mask and workspace.

Color change

- If request color change, program layer creates for new color, AI will generate new drawing layer.

Request to draw/write

- Request to draw/write, program will call cv2 line if click detected, sends frames detected.
- Display line traced by the wand.

Request for clear screen

- Request for clear screen, program will call layer delete, AI will delete the color layers detected.

3.3 Flow chart

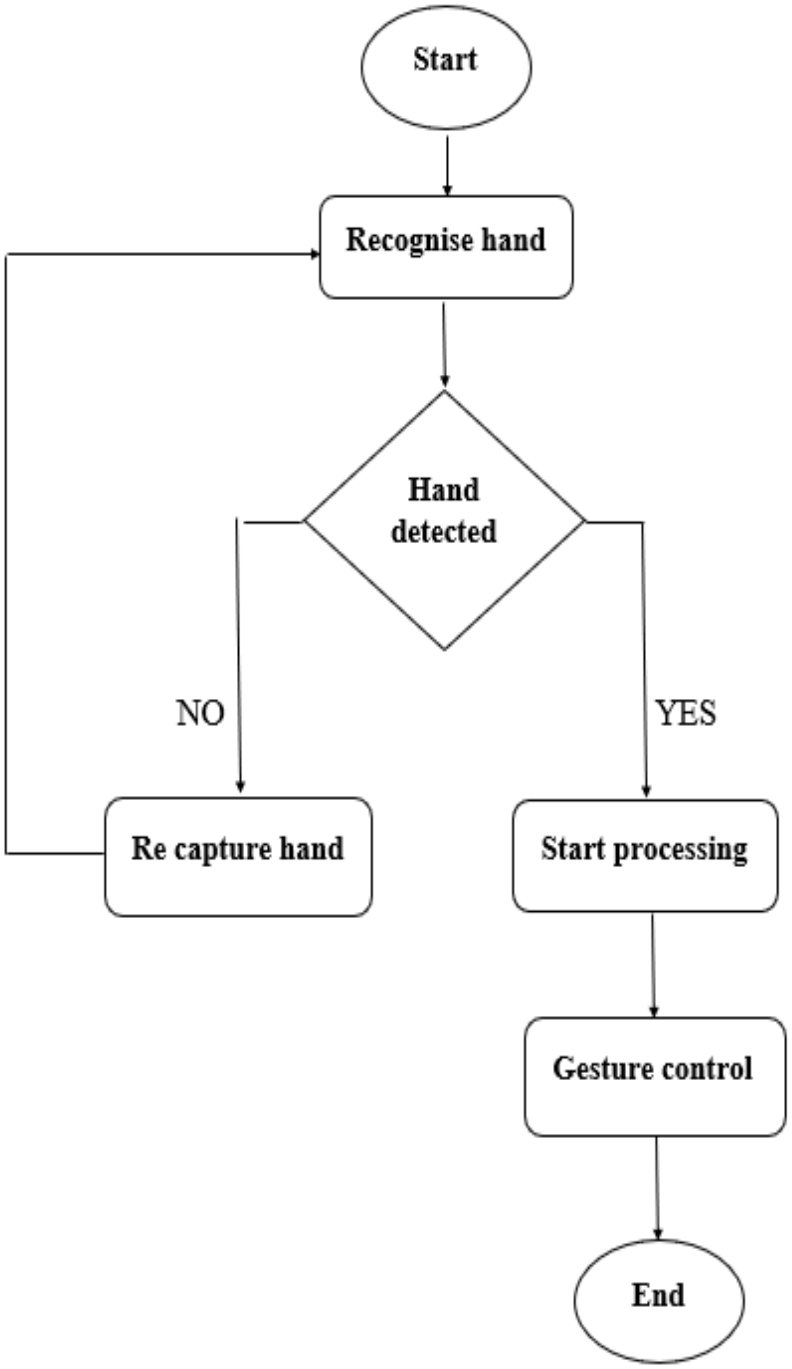

Fig 3.3

Explanation:

- The user will start.
- Camera will detect hands.
- Hands get detected.
- If no then recapture hands and recognize it.
- If yes then start processing.
- Gesture control will start.
- End the process.

3.4 Basic concepts and tools

3.4.1 Introduction to Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

3.4.2 Introduction to Tkinter

Tkinter is an open source, portable graphical user interface (GUI) library designed for use in Python scripts.
Tkinter relies on the Tk library, the GUI library used by Tcl/Tk and Perl, which is in turn implemented in C. Therefore, Tkinter can be said to be implemented using multiple layers.
Several competing GUI toolkits are available to use with the Python language, namely:

- wxPython: a wrapper extension for wxWindows, a portable GUI library originally developed for the C++ language. It is the second most popular GUI toolkit for Python since it is considered excellent for complex interface design.
- JPython (Jython): since it is implemented in java, JPython has access to Java GUI libraries, namely SWING and AWT. Recently, JTkinter has been implemented and provides a Tkinter port to JPython using the Java Native Interface (JNI).
- PyKDE / PyQt, PyGTK: these packages provide an access to KDE and Gnome GUI libraries to python scripts.
- Win32all.exe: provides access to Microsoft Foundation Classes (MFC) to python scripts. It is limited to run on MS Windows only
- WPY: a GUI library that can be used on both Microsoft Windows and UNIX X Windows. This library uses the MFC coding style.
- X11: a library based on the X Windows and Motif libraries allowing excellent control of the X11 environment, but are limited to run on X Windows OS's only.

3.4.3 Introduction to Open CV

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 a BSD-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. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching streetview images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers

3.4.4 Introduction to Pandas

Pandas is a Python library for data analysis. Started by Wes McKinney in 2008 out of a need for a powerful and flexible quantitative analysis tool, pandas have grown into one of the most popular Python libraries. It has an extremely active community of contributors.

Pandas is built on top of two core Python libraries—matplotlib for data visualization and NumPy for mathematical operations. Pandas acts as a wrapper over these libraries, allowing you to access many of matplotlib's and NumPy's methods with less code. For instance, pandas. plot () combines multiple matplotlib methods into a single method, enabling you to plot a chart in a few lines.
Before pandas, most analysts used Python for data munging and preparation, and then switched to a more domain specific language like R for the rest of their workflow. Pandas introduced two new types of objects for storing data that make analytical tasks easier and eliminate the need to switch tools: Series, which have a list-like structure, and Data Frames, which have a tabular structure

3.4.5 Introduction to NumPy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

At the core of the NumPy package, is the object. This encapsulates *n*-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences:

- NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). Changing the size of an will create a new array and delete the original.
- The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy) objects, thereby allowing for arrays of different sized elements.
- NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.
- A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays. In other words, in order to efficiently use much (perhaps even most) of today's scientific/mathematical Python-based software, just knowing how to use Python's built-in sequence types is insufficient - one also needs to know how to use NumPy arrays.

CHAPTER 4

Result / Outputs / Conclusion

Result

- A simple, easy, optimized, smooth running software which is an alternative of hand drawings and gestures.

- Which can write /paint anything on screen virtually.

Conclusion

- The system has the potential to challenge traditional writing methods. It eradicates the need to carry a mobile phone in hand to jot down notes, providing a simple on- the-go way to do the same. It will also serve a great purpose in helping especially abled people communicate easily.
- Even senior citizens or people who find it difficult to use keyboards will able to use system effortlessly. Extending the functionality, system can also be used to control IoT devices shortly. Drawing in the air can also be made possible.
- The system will be an excellent software for smart wearables using which people could better interact with the digital world. Augmented Reality can make text come alive. There are some limitations of the system which can be improved in the future.
- Firstly, using a handwriting recognizer in place of a character recognizer will allow the user to write word by word, making writing faster.
- Secondly, hand-gestures with a pause can be used to control the real-time system instead of using the number of fingertips.
- Thirdly, our system sometimes recognizes fingertips in the background and changes their state. Air-writing systems should only obey their master's control gestures and should not be misled by people around.

CHAPTER 5
References

- Researchers Explore Machine Learning Calibration

By Pure AI Editors | March 2021 - https://pureai.com/articles/2021/03/03/ml-calibration.aspx

- Distance Formula & Training for Tracing Ping Pong AI

By GitHub | March 2021 - https://github.com/AProjectAMonth/Ping-Pong

- Object detection with TensorFlow model and OpenCV

By Gabriel Casemiro | Jul 2016 - https://towardsdatascience.com/object-detection-with-tensorflow-model-and-opencv-d839f3e42849

CHAPTER 6
Screenshots

Fig 6.1

Selection tool using hand detection :

Two fingers are used to select the tools.

Fig 6.2


Drawing:
After selection of tool, user can draw with any thing on board using any fingure\thumb.

Fig 6.3

Eraser Tool:

Using a tool mentioned above, user can use eraser to erase any painted thing on board.

Fig6.4

Steps to execute/run:

Step 1: Install pre-requirements i.e., Python, NumPy, Pandas

Step 2: Install an IDE. E.g.- VS Code, Atom, PyCharm

Step 3: Run the program