A
Major Project Report
On

EYE BALL CURSOR MOVEMENT USING OPENCV

Submitted to ***CMREC, HYDERABAD***

In Partial Fulfillment of the requirements for the Award of Degree of
BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

Submitted
By

B.Bhargavi	**(218R1A6769)**
A.Raghavender	**(218R1A6766)**
T.Shreya Sri	**(228R5A6713)**
B.Nithin	**(218R1A6775)**

Under the Esteemed guidance of

Mr. M. Naveen Kumar

Assistant Professor, Department of CSE (DS)

Department of Computer Science & Engineering (Data Science)

CMR ENGINEERING COLLEGE

(UGC AUTONOMOUS)

(Approved by AICTE, NEW DELHI, Affiliated to JNTU, Hyderabad)
Kandlakoya, Medchal Road, R.R. Dist. Hyderabad-501 401)

2024-2025

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Accredited by NBA,Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad)

Kandlakoya, Medchal Road, Hyderabad-501 401

Department of Computer Science & Engineering (Data Science)

CERTIFICATE

This is to certify that the project entitled "**EYE BALL CURSOR MOVEMENT USING OPENCV"** is a Bonafide work carried out by

B.Bhargavi	**(218R1A6769)**
A.Raghavender	**(218R1A6766)**
T.Shreya Sri	**(228R5A6713)**
B.Nithin	**(218R1A6775)**

in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)** from CMR Engineering College affiliated to JNTU, Hyderabad, under our guidance and supervision. The results presented in this project have been verified and are found to be satisfactory. The results embodied in this project have not been submitted to any other university for the award of any other degree or diploma.

Internal Guide	**Major Project Coordinator**	**Head of the Department**	**External Examiner**
Mr. M. Naveen Kumar	**Mr. B. Kumaraswamy**	**Dr. M. Laxmaiah**	
Assistant Professor CSE (Data Science), CMREC	Assistant Professor CSE (Data Science), CMREC	Professor & HOD CSE (Data Science), CMREC	

DECLARATION

This is to certify that the work reported in the present project entitled **"EYE BALL CURSOR MOVEMENT USING OPENCV"** is a record of bonafide work done by us in the Department of Computer Science and Engineering (Data Science), CMR Engineering College, Hyderabad. The reports are based on the project work done entirely by us and not copied from any other source. We submit our project for further development by any interested students who share similar interests to improve the project in the future. The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

B.Bhargavi (218R1A6769)
A.Raghavender (218R1A6766)
T.Shreya Sri (228R5A6713)
B.Nithin (218R1A6775)

ACKNOWLEDGMENT

We are extremely grateful to **Dr. A. Srinivasula Reddy**, Principal and **Dr. M. Laxmaiah,** HOD, **Department of CSE (Data Science), CMR Engineering College** for their constant support**.**

We extremely thankful to **Mr. M. Naveen Kumar,** Assistant Professor, Internal Guide, Department of CSE, for his constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

We thank **Mr. B. Kumar Swamy** Major Project Coordinator for his constant support in carrying out the project activities and reviews.

We express our thanks to all staff members and friends for all the help and co- ordination extended in bringing out this project successfully in time.

Finally, we very much thankful to my parents who guided me for every step.

B.Bhargavi (218R1A6769)
A.Raghavender (218R1A6766)
T.Shreya Sri (228R5A6713)
B.Nithin (218R1A6775)

ABSTRACT

In this study, a specific human-computer interaction system using eyeball movement is presented. Conventionally, computer systems use a mouse as one of the primary data input devices. However, in this system, we use eyes instead of a mouse, providing a unique and innovative way of operating the computer through eyeball movements. The implementation work underlying this system for pupil identification uses a Raspberry Pi board to control the cursor of the personal computer. Furthermore, the Eye Aspect Ratio (EAR) technique is employed alongside OpenCV to detect the pupil and track eye movements.

The system utilizes an IP camera (Internet Protocol camera) to capture real-time video of the user's eyes. Advanced image processing algorithms are applied to analyze the movements and simulate corresponding mouse cursor actions on the screen. In addition to cursor control, the system is capable of detecting when the user focuses on an icon for a specific period, triggering a click operation. This feature is particularly beneficial for individuals with motor disabilities, providing them with an alternative means of interacting with digital devices.

The primary objective of this system is to enhance accessibility by offering hands-free computer control. It can significantly benefit users with conditions such as paralysis, muscular dystrophy, or spinal cord injuries. The implementation demonstrates the feasibility and accuracy of using eye-tracking technology for cursor control, paving the way for further developments in assistive technology. Through this system, users can achieve a more inclusive computing experience.

Keywords: Eyeball movement, Mouse, Raspberry pi, IP Cam (Internet Protocol camera).

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 Overview

Personal computers play a significant role in our daily lives, serving various purposes such as work, education, and entertainment. Conventional computer input methods, including keyboards and mice, remain dominant. However, individuals with mobility impairments face challenges using these traditional devices. This study proposes an innovative human-computer interaction system that uses eye movements to control the cursor, offering an accessible and inclusive solution.

Eye-tracking technology has evolved significantly over the years, offering promising applications in assistive technology, gaming, and healthcare. The development of cost-effective, user-friendly eye-tracking systems can provide independence to individuals with disabilities, allowing them to perform computer-related tasks without external assistance. This paper explores the potential of using a Raspberry Pi and OpenCV for an efficient, low-cost eye-tracking system.

The proposed system captures eye movements through an IP camera and translates them into cursor movements on the screen. Advanced image processing techniques, including the Eye Aspect Ratio (EAR) method, allow accurate detection of pupil position and blinking patterns. This hands-free control method not only enhances accessibility but also contributes to the broader field of human-computer interaction by improving non-intrusive input methods. This approach is beneficial in making technology inclusive and ensuring that even those with severe disabilities can participate in the digital world.

Another notable aspect of this system is its affordability and ease of implementation. By using open-source software and cost-effective hardware components, the solution is accessible to a wide range of users. The Raspberry Pi, a small yet powerful computing device, ensures that the eye-tracking system remains budget-friendly without compromising performance.

Overall, the system aims to bridge the gap between humans and machines by providing a seamless and intuitive interaction experience. This research contributes to the growing field of human-computer interaction and highlights the importance of developing inclusive technologies that cater to individuals with diverse needs**.**

1.2 Research Motivation

The motivation behind this research is to empower people with physical disabilities by providing a hands-free alternative to conventional input devices. Many existing solutions are costly and complex, limiting their accessibility. This research aims to develop a low-cost, efficient, and user-friendly system using a Raspberry Pi, an IP camera, and computer vision techniques. By employing eyeball movement tracking, the system offers greater independence to users with limited mobility.

Current assistive technologies for people with motor disabilities often involve expensive equipment or invasive procedures. While high-end eye-tracking systems exist, their high costs make them inaccessible to a large portion of the population. By developing an affordable alternative, this research ensures that individuals who require assistive devices can access them without financial constraints. Additionally, this project explores the potential for integrating eye-tracking systems into mainstream computing, making them more widespread and applicable beyond accessibility solutions.

Another key motivation is the increasing reliance on digital interfaces in everyday life. As technology advances, digital inclusion is becoming an essential factor in human development. Enabling alternative input methods will ensure that people with disabilities are not left behind in the digital age. Through this research, we aim to bridge the accessibility gap by developing an efficient and affordable eye-tracking system that allows users to interact with computers seamlessly.

Furthermore, eye-tracking technology holds immense potential in other fields, including gaming, virtual reality, healthcare diagnostics, and consumer behaviour analysis. This study's contribution can extend beyond accessibility by paving the way for innovative applications that rely on gaze-tracking data. Thus, the motivation is not only to improve accessibility but also to advance the broader field of human-computer interaction.

1.3 Problem Statement

The primary problem addressed in this research is the lack of accessible computer input methods for individuals with physical impairments. Traditional input devices like keyboards and mice are impractical for people with motor disabilities. This project aims to develop a reliable, cost-effective solution that allows users to control a computer cursor using only their eye movements. The system will also incorporate a mechanism for detecting focus and performing click operations based on eye gaze.

For individuals with limited motor functions, using conventional computer peripherals can be a significant challenge. While speech recognition and other alternative input methods exist, they may not always be suitable for users with speech impairments or other disabilities. An eye-controlled system provides a universal and intuitive approach that requires minimal physical effort, making it accessible to a wide range of users.

Additionally, existing eye-tracking technologies often rely on infrared sensors or high-end cameras, making them expensive and impractical for widespread use. This research seeks to overcome these limitations by leveraging open-source software, affordable hardware, and efficient image processing techniques. By creating a system that is both reliable and cost-effective, this project aims to make eye-tracking technology more accessible to those who need it most.

1.4 Applications

The proposed eyeball movement-based cursor control system has numerous applications, including:

1. **Assistive Technology:** Providing a means for individuals with physical disabilities to control a computer without using hands.
2. **Medical and Rehabilitation Centers:** Offering rehabilitation solutions for patients with temporary or permanent mobility impairments.
3. **Gaming and Virtual Reality:** Enhancing immersive gaming experiences through eye-tracking-based controls.
4. **Research and Development:** Facilitating research in fields such as neuroscience, psychology, and human-computer interaction.
5. **Smart Home Control:** Enabling users to manage smart home devices through eye movements for greater convenience.

Beyond accessibility, eye-tracking technology has the potential to improve various industries, including marketing, security, and user experience research. By analysing eye movement patterns, businesses can gain insights into consumer behaviour and optimize user interfaces. In security applications, gaze-tracking can be used for biometric authentication, providing a more secure alternative to traditional passwords. This system has the potential to bridge the accessibility gap and promote digital inclusion for people with mobility limitations. Future advancements may further refine this technology, making it even more efficient and widely applicable. With continued research and development, eye-tracking systems could become an integral part of everyday computing, benefiting both disabled individuals and mainstream users alike.

2. LITERATURE SURVEY

Over the past decade, artificial intelligence (AI) has significantly contributed to various subfields, including assistive technologies for individuals with disabilities. Different studies have explored eye- tracking systems for cursor control, aiming to improve accessibility and human-computer interaction. However, challenges remain in achieving accurate and real-time gaze tracking. A comprehensive overview of existing studies is presented in Table 1. Further exploration is required to enhance detection and localization approaches for real-time eye-tracking-based cursor control.

To enable cursor control via eye movements, Kanchan Pradhan et al. proposed a machine learning-based approach to interpret and track gaze patterns [8]. The study aimed to develop a system where users could control the cursor purely through eye movements. The authors investigated key aspects necessary for effective gaze-based control, emphasizing the importance of accurate and timely detection of eye movements alongside the selection of robust machine learning models.

Mohamed Nasor et al. explored the development of an eye-operated mouse cursor designed specifically for individuals with physical disabilities [9]. The study focused on designing an accessible interface where eye movements served as the primary input for cursor control. The authors highlighted technical challenges in reliable gaze-tracking implementation, concluding that achieving accurate and consistent gaze detection is crucial for assistive technology.

Sivasangari et al. introduced an eye-gaze-driven cursor control system aimed at providing an alternative to traditional input devices, particularly for users with physical impairments [10]. The system utilized eye- tracking technology to allow hands-free computer navigation. The authors developed a pupil detection method using a Raspberry Pi and OpenCV, ensuring that cursor movement was accurately mapped to eye position. The system booted from an SD card containing the operating system, and upon application launch, the Raspberry Pi executed the eye-tracking cursor control.

Mathew, et al. [11] introduced an eye-gaze driven system for cursor control and home automation, targeting individuals with disabilities. Recognizing the critical role of computer access, this research explored eye movements as a viable communication tool for paralyzed individuals. Utilizing an eye-tracking methodology for movement acquisition, coupled with a simplified circuit design, the system aims to address Human-Machine Interface (HMI) challenges faced by disabled individuals.

Shubham, et al. [12] presented an eyeball movement-based cursor system, utilizing deep learning. This research focused on leveraging deep learning techniques to accurately interpret and translate eye movements into cursor control, aiming to enhance accessibility for individuals with limited mobility. The system explored the development of robust deep learning models capable of processing real-time eye- tracking data, converting subtle eye movements into actionable cursor commands. The study highlighted the potential of deep learning to improve the accuracy and responsiveness of eye-controlled interfaces, contributing to advancements in assistive technology.

Title	Objective	Method/Algorithm Used	Research Gaps
Eye Movement-based Cursor Control	To develop a system where users can control the cursor purely through eye movements.	Machine learning-based approach to interpret and track gaze patterns.	Requires accurate and timely detection of eye movements; selection of robust models is crucial.
Eye-Operated Mouse Cursor for Individuals with Disabilities	To design an accessible interface where eye movements serve as the primary input for cursor control.	Gaze-tracking technology for cursor movement recognition.	Achieving accurate and consistent gaze detection remains a challenge.
Eye-Gaze Driven Cursor Control System	To provide an alternative to traditional input devices, particularly for users with physical impairments.	Pupil detection method using Raspberry Pi and OpenCV for eye-tracking	Requires optimized processing for real-time tracking accuracy.
Eye-Gaze Driven System for Cursor Control and Home Automation	To enable cursor controland home automation for individuals with disabilities.	Eye-tracking methodology integrated with Human- Machine Interface (HMI) for home automation.	Design may limit scalability and advanced control features.
Eyeball Movement-Based Cursor Control Using Deep Learning	To enhance accessibility using deep learning techniques for eye-tracking-based cursor control.	Deep learning models process real-time eye-tracking data for cursor movement interpretation.	Accuracy and responsiveness of deep learning models require further optimization.

3. EXISTING SYSTEM

3.1 MATLAB-Based Eye Tracking System

MATLAB has been a fundamental tool in the development of eye-controlled systems, particularly for iris detection, pupil tracking, and cursor movement control. These systems utilize image processing and pattern recognition to track eye movement and estimate gaze direction. MATLAB's built-in Image Processing Toolbox and Computer Vision Toolbox provide various functions for analyzing eye images and detecting key features such as the pupil center, iris boundary, and sclera movement.

MATLAB-based eye-tracking systems are commonly used in:

Assistive Technologies – Enabling individuals with disabilities to control computers, wheelchairs, or home appliances.

Human-Computer Interaction (HCI) – Allowing hands-free operation of digital interfaces.

Medical Applications – Eye movement analysis for **neurological disorder detection** (e.g., Parkinson's disease, Alzheimer's).

Gaming and Virtual Reality (VR) – Enhancing immersive experiences by integrating gaze-based controls.

3.2 Working of MATLAB-Based Eye Tracking Systems

A typical MATLAB-based eye-controlled system follows these key steps:

1. Image Acquisition

- A webcam or infrared camera captures a live feed of the user's eyes.
- High-resolution images improve the accuracy of eye detection.

2. Preprocessing

- The captured image is converted to grayscale for faster processing.
- Noise reduction techniques (such as median filtering) are applied to remove unwanted artifacts.
- Contrast enhancement is performed using histogram equalization to make features like the iris and pupil more distinguishable.

3. Eye Detection and Pupil Tracking

- The Hough Transform or Circular Hough Transform (CHT) is used to detect circular shapes,

  identifying the iris and pupil boundaries.
- Edge detection techniques (such as Canny or Sobel edge detection) help locate the boundary between the iris and the sclera.
- MATLAB's region-based segmentation methods, like thresholding and morphological operations, are used to extract the pupil centroid.

4. Gaze Estimation

- Pupildisplacement is analyzed relative to a reference position.
- Movement is mapped to screen coordinates, enabling cursor control.
- Advanced systems implement geometric models to predict gaze direction based on the position of both eyes.

5. Cursor Control or Command Execution

- The detected gaze position is translated into mouse cursor movement.
- Blink detection algorithms can be used to trigger mouse clicks.
- Some systems integrate keyboard emulation, allowing users to select characters or commands using their gaze.

3.3 Drawbacks

Despite its advantages, MATLAB-based eye-tracking systems face several challenges and limitations:

- **Accuracy Issues** – MATLAB's traditional image processing techniques often fail to precisely locate the pupil center, leading to inaccurate gaze tracking.
- **High Computational Load** – MATLAB's algorithms, especially those involving Hough Transforms and edge detection, require significant processing power, making real-time tracking difficult.

- **Lighting Sensitivity –** Performance is heavily influenced by ambient lighting conditions. Changes in brightness, reflections, and shadows can affect detection accuracy.
- **Slow Response Time –** Compared to optimized C++ or Python-based OpenCV implementations, MATLAB systems can have higher latency, making real-time applications less responsive.
- **Frequent Recalibration Needed** – Users often need to calibrate the system multiple times, adjusting for eye movement variations and head position changes.
- **Limited Deep Learning Integration** – MATLAB's traditional image processing methods struggle to adapt to complex eye movements. Deep learning-based gaze estimation (e.g., using CNNs) is more effective but less commonly implemented in MATLAB.
- **Hardware Dependency** – Some MATLAB-based solutions require high-resolution cameras, infrared sensors, or specialized hardware, increasing cost and complexity.

4. PROPOSED METHODOLOGY

4.1 Overview

The proposed system aims to provide a hands-free computer control solution using eye movement tracking. At the core of this system is the Raspberry Pi 3, a compact and versatile microcomputer equipped with an ARM11 processor. The Raspberry Pi is chosen for its affordability, low power consumption, and compatibility with numerous peripherals. It serves as the central processing unit, managing data inputs from sensors and executing programmed algorithms for eye detection and tracking. The system is equipped with an IR (Infrared) sensor to detect eye movement, a monitor for display, and an SD card for operating system storage and data management.

The Raspberry Pi runs the Raspbian OS, a Debian-based operating system optimized for Raspberry Pi devices. Python is used as the primary programming language due to its flexibility and compatibility with OpenCV, a powerful library for image processing. The software modules involve capturing the eye's position using the IR sensor, processing the data to determine eye movement, and subsequently controlling the mouse cursor. The captured eye data is analyzed using algorithms to detect the direction of gaze, allowing the user to navigate the screen or select options without manual input. This system is particularly beneficial for people with physical disabilities, as it offers them a more accessible way to interact with digital devices.

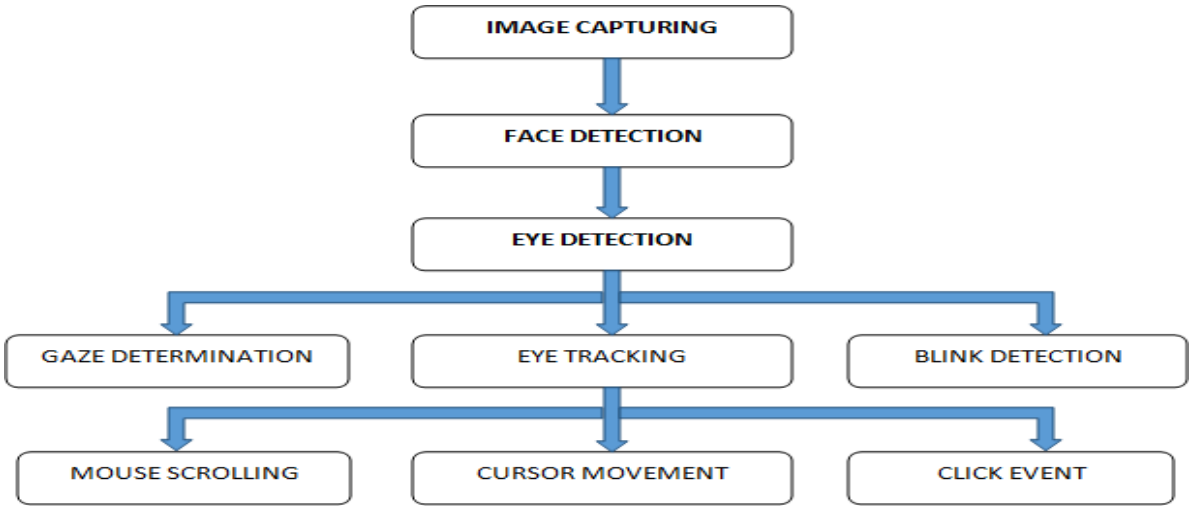


Fig. 4.1: Block diagram of proposed system

4.2 Components and Functionalities

The proposed system consists of multiple components that work together to achieve real-time gaze tracking and cursor control. The IR sensor plays a crucial role in detecting eye movements. Infrared sensors are ideal for this purpose as they can accurately track eye motion without interference from external light sources. By emitting infrared light and capturing the reflection from the user's eyes, the sensor can determine the position of the pupil and track changes in movement. This data is transmitted to the Raspberry Pi for further processing.

Additionally, the monitor connected to the Raspberry Pi serves as the primary display, providing a visual representation of the user's interactions. The SD card installed in the Raspberry Pi acts as the storage unit, housing the operating system, application software, and essential libraries. The use of USB adaptors ensures seamless connectivity between the Raspberry Pi, the IR sensor, and the display unit. This configuration makes the system portable and easy to set up in various environments, including homes, offices, and medical facilities. The simplicity of the hardware design contributes to the system's cost- effectiveness, making it a viable solution for users with different needs and budgets.

4.3 Eye Movement Detection Process

The eye movement detection process begins with the IR sensor capturing real-time images of the user's eyes. The sensor continuously monitors the pupil’s position and transmits this information to the Raspberry Pi for analysis. OpenCV, a widely used computer vision library, is utilized to process the captured images and detect the location of the pupil. The Eye Aspect Ratio (EAR) technique is applied to determine the eye state (open or closed) and identify any deliberate eye blinks that may be used as input commands.

Once the eye movement data is processed, the system interprets the directional gaze to control the mouse cursor. For instance, if the user’s gaze moves towards the left, the cursor moves in the corresponding direction. Similarly, looking up or down adjusts the cursor vertically. Additionally, intentional eye blinks can trigger actions such as mouse clicks, eliminating the need for physical interaction with the device. This intuitive control method ensures a seamless and responsive user experience, empowering individuals with mobility impairments to perform tasks independently.

4.4 Calibration and Accuracy

To ensure accuracy in gaze tracking, the system includes a one-time calibration process. During calibration, the user is required to follow a set of predefined points displayed on the screen. The IR sensor captures the user's eye movements at each point, establishing a reference for future gaze detection. This calibration data is stored on the SD card, allowing the system to adjust to individual differences in eye movement patterns. Additionally, the system employs algorithms to account for minor head movements, further enhancing tracking accuracy.

The use of adaptive algorithms also ensures consistent performance under varying lighting conditions. The IR sensor's ability to operate effectively in low-light environments makes it suitable for both indoor and outdoor use. Moreover, the Raspberry Pi's ARM11 processor is capable of handling real-time image processing tasks with minimal latency, resulting in a smooth and responsive user interface. Through regular calibration updates and software optimizations, the system maintains a high level of accuracy and reliability.

4.5 Advantages

OpenCV (Open Source Computer Vision Library) has become a widely adopted tool for implementing eye-controlled cursor systems due to its efficiency, accuracy, and real-time processing capabilities. Compared to traditional MATLAB-based approaches, OpenCV offers several advantages:

- **High Accuracy in Pupil Detection** – OpenCV provides advanced computer vision algorithms that can accurately detect and track the pupil and iris.
- **Real-Time Cursor Control** – OpenCV is optimized for real-time processing, enabling smooth and responsive cursor movements.
- **Works Well in Different Lighting Conditions** – OpenCV supports adaptive thresholding and contrast enhancement, making it robust in varying lighting conditions.
- **Faster Processing Speed** – OpenCV's and Python implementations offer **faster execution** than MATLAB's scripting environment.
- **Multi-Platform Support** – OpenCV supports Windows, Linux, macOS, and embedded systems

.

- **Reduced Calibration Needs** – Unlike MATLAB-based systems that require frequent recalibration, OpenCV-based systems use advanced feature tracking for stable gaze estimation.
- **Open-Source** – OpenCV is free and open-source, making it more accessible than proprietary MATLAB.
- **Cost-Effective –** Runs on low-cost hardware, including Raspberry Pi, smartphones, and webcams.

5. UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to or associated with UML.

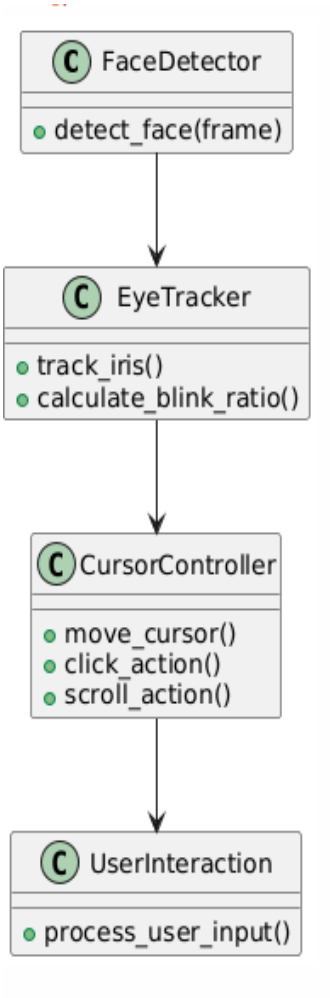
The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations process. The UML uses mostly graphical notations.

GOALS: The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can that develop and exchange meaningful model.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

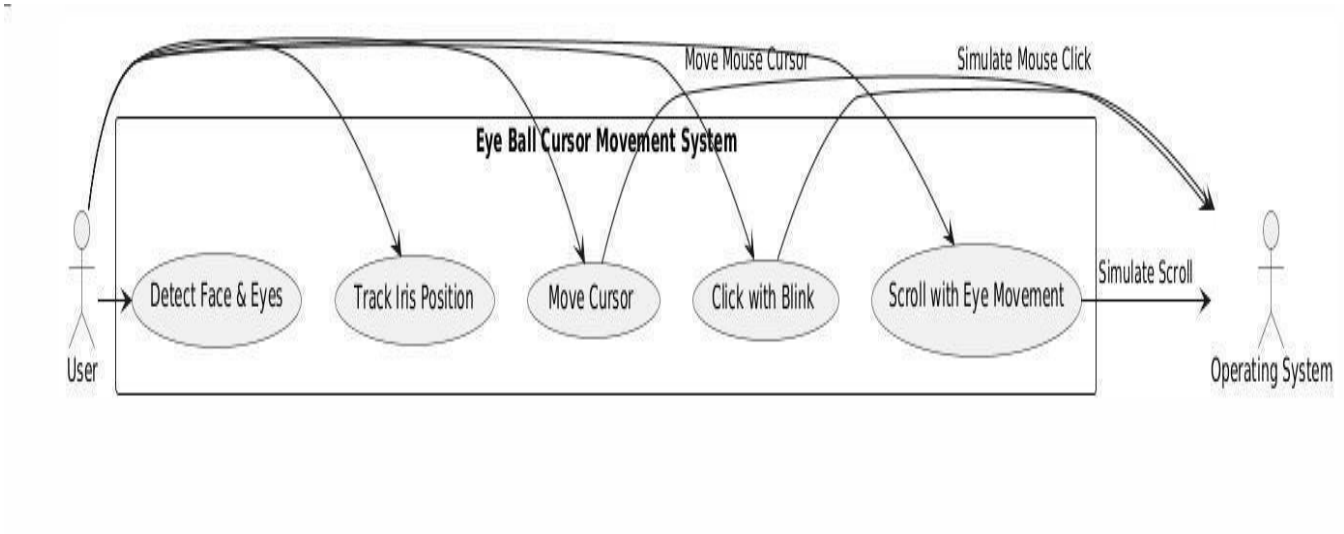
5.1 Class diagram

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.



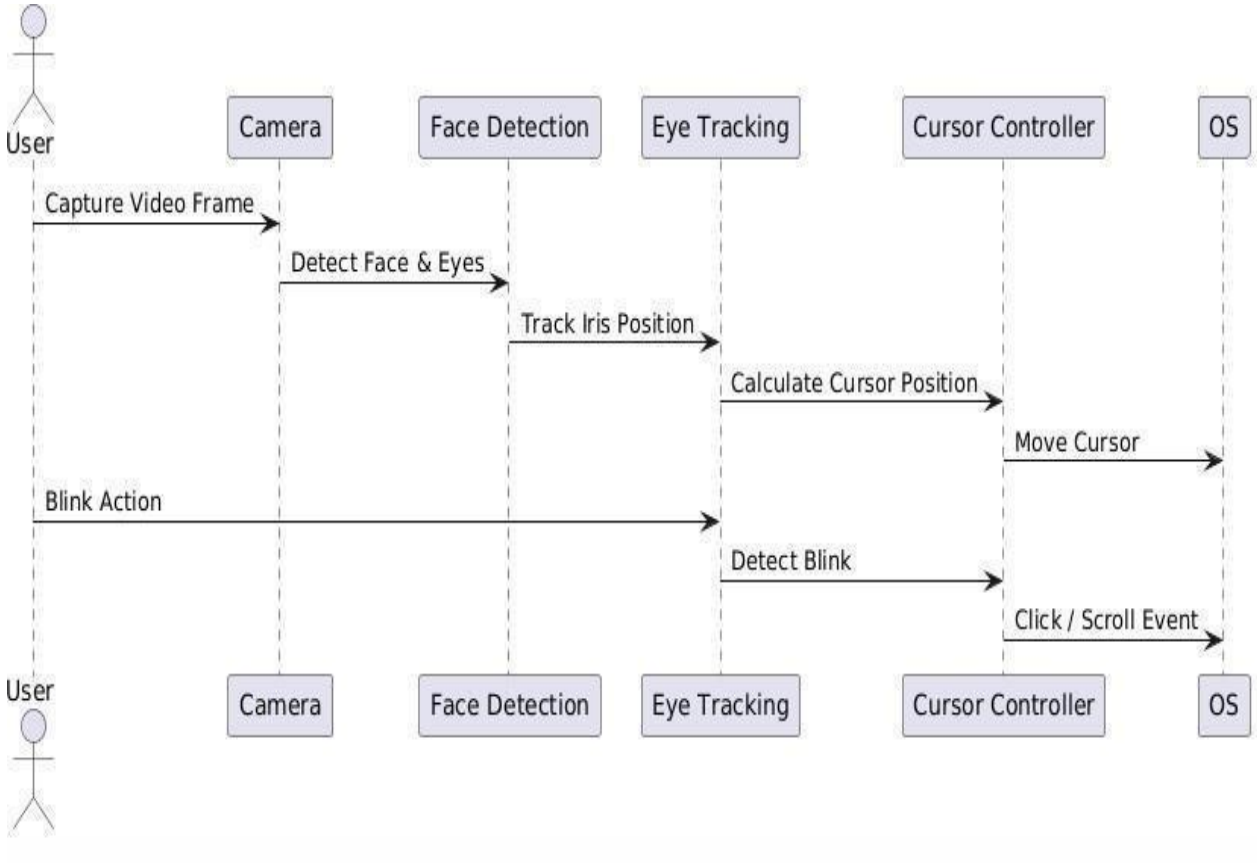
5.2 Use case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system are depicted.



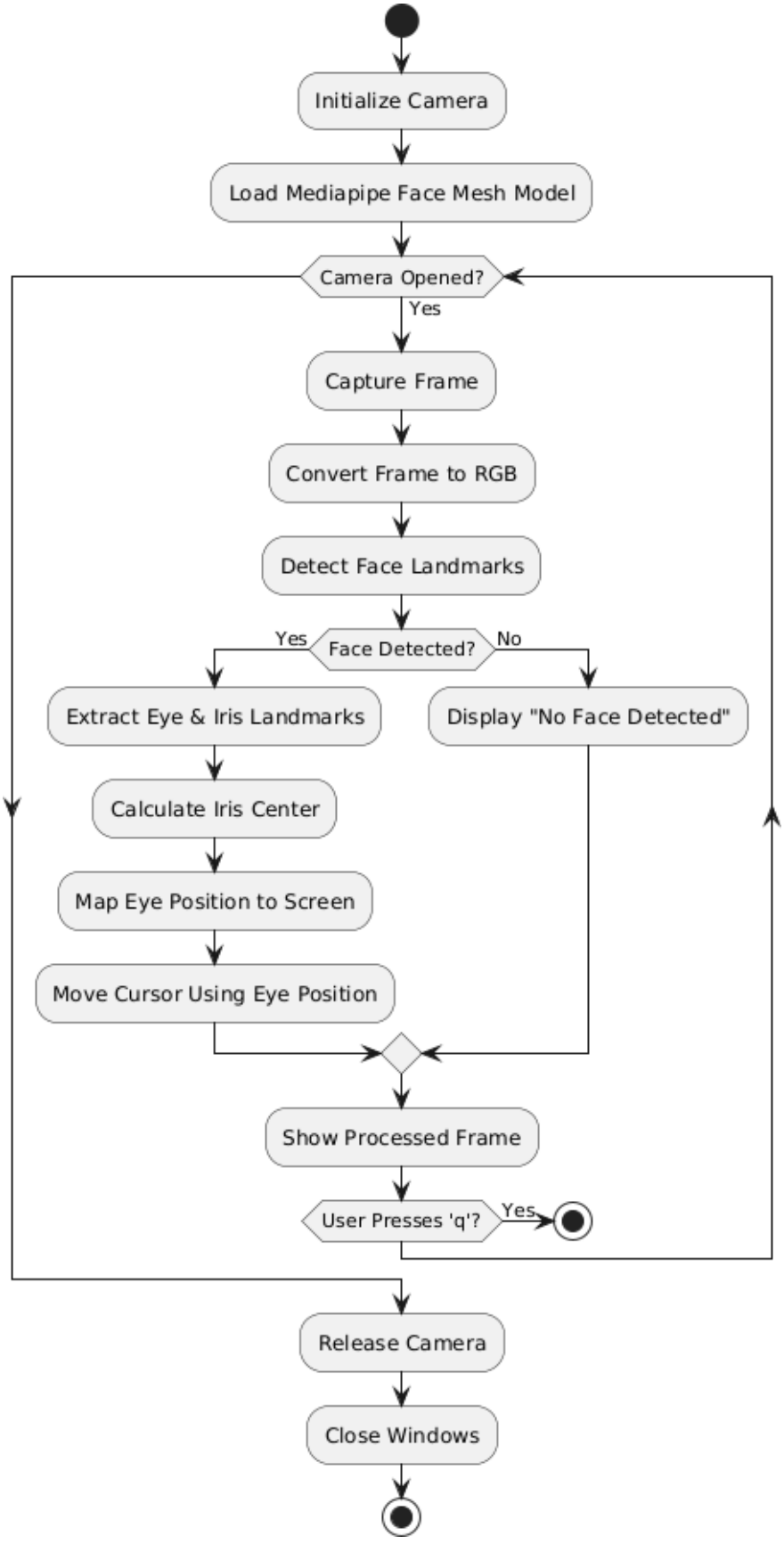
5.3 Sequence Diagram

A **sequence diagram** in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows, as parallel vertical lines ("lifelines"), different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.



5.4 Activity Diagram

Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration, and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.


Initialize Camera
Load Mediapipe Face Mesh Model
Camera Opened?
Yes
Capture Frame
Convert Frame to RGB
Detect Face Landmarks
Face Detected?
Yes
No
Extract Eye & Iris Landmarks
Display "No Face Detected"
Calculate Iris Center
Map Eye Position to Screen
Move Cursor Using Eye Position
Show Processed Frame
User Presses 'q'?
Yes
Release Camera
Close Windows

6. SOFTWARE ENVIRONMENT

6.1 What is Python?

Below are some facts about Python.

- Python is currently the most widely used multi-purpose, high-level programming language.
- Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.
- Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
- Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard libraries which can be used for the following :

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv,pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python

Let's see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet of Things. This is a way to connect the language with the real world.

6. Simple and Easy

When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. These further aids to the readability of the code. Python is widely recognized for its readability, making it one of the most beginner-friendly programming languages. Its clean and simple syntax closely resembles natural English, making it easy to understand and write.

8. Object- Oriented

This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

9. Free and Open-Source

Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

Anydoubts till now in the advantages of Python? Mention in the comment section.

Advantages of Python Over Other Languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or windows. Programmers need to learn different languages for different jobs but with Python, you can professionally professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

Disadvantages of Python

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle. The reason it is not so famous despite the existence of Brython is that it isn't that secure.

3. Design Restrictions

As you know, Python is dynamically-typed. This means that you don't need to declare the type of variable while writing the code. It uses duck-typing. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

4. Underdeveloped Database Access Layers

Compared to more widely used technologies like JDBC (Java DataBase Connectivity) and ODBC (Open DataBase Connectivity), Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

6. 2 History of Python

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it. "Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So, I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my

own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

Python Development Steps

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of lists, dict, str and others. It was also object oriented and had a module system. Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced.

This release included list comprehensions, a full garbage collector and it was supporting unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it. Some changes in Python 7.3:

- Print is now a function.
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g., a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e., int. long is int as well.
- The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behavior.
- Text Vs. Data Instead of Unicode Vs. 8-bit .

Purpose

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is

6.3 Modules Used in project

TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable Programming across a range oftasks. It is a symbolic math library and is also used for machine Learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed bythe Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object-oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

6.4 Install Python Step-by-Step in Windows and Mac

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high- level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

How to Install Python on Windows and Mac

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e., operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So, the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here. The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

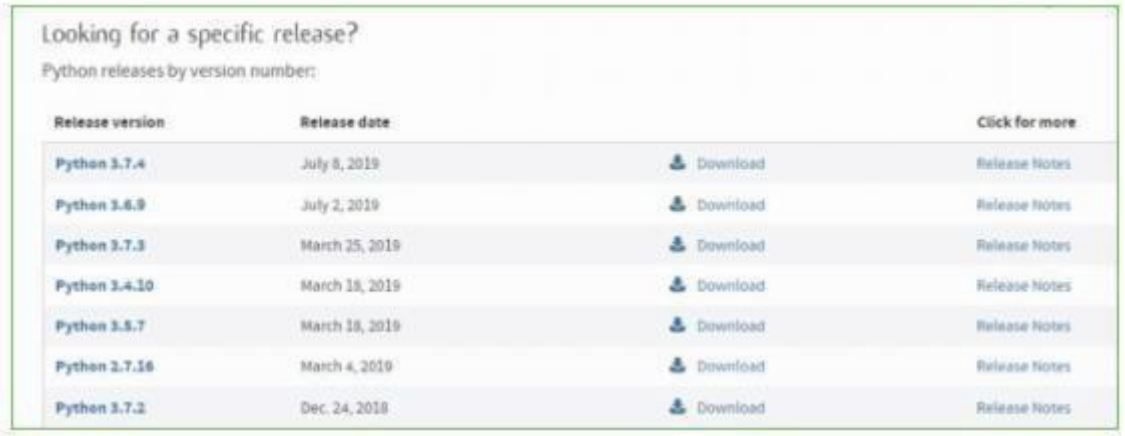
Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org

Now, check for the latest and the correct version for your operating system. Step

2: Click on the Download Tab.

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4ef7b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	642fb4fa7583daff1e442cbabcee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5ddd605c38217a45773bf5e4a936b243f	28982845	SIG
Windows help file	Windows		d63999573a2c98b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b09c8cf6df9ec9b9abe83184a4372ba2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0adf76de6db30c1afd3e563e00	26883968	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28c81c6088bd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab38d1f8841879fda94133574139d8	6741628	SIG
Windows x86 executable installer	Windows		33cc8d2942a64446a3d8d41476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa6d317df82c3098e3ea371d87c	1324608	SIG

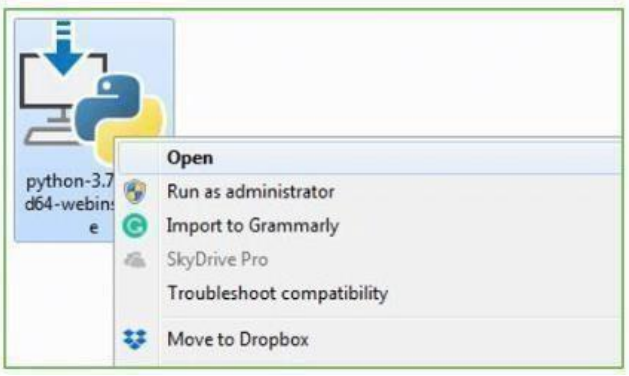
1. To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
2. To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e., Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



Step 2: Before you click on Install Now, make sure to put a tick on Add Python 3.7 to PATH.

Step 3: Click on Install NOW After the installation is successful. Click on Close.

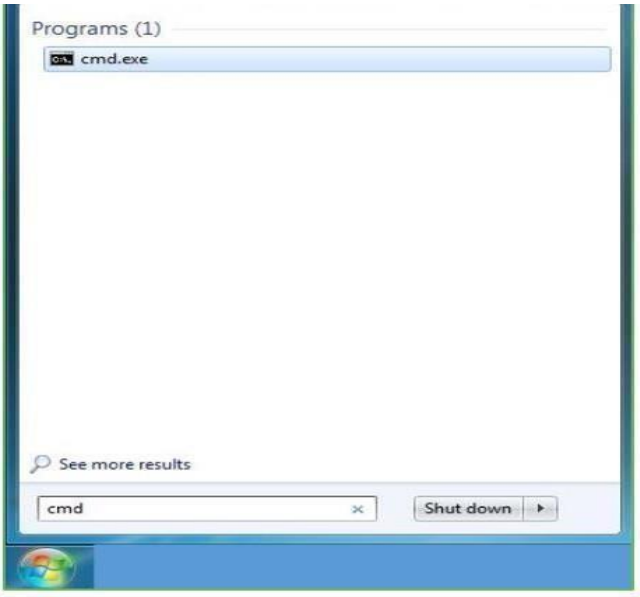
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes. Verify

the Python Installation

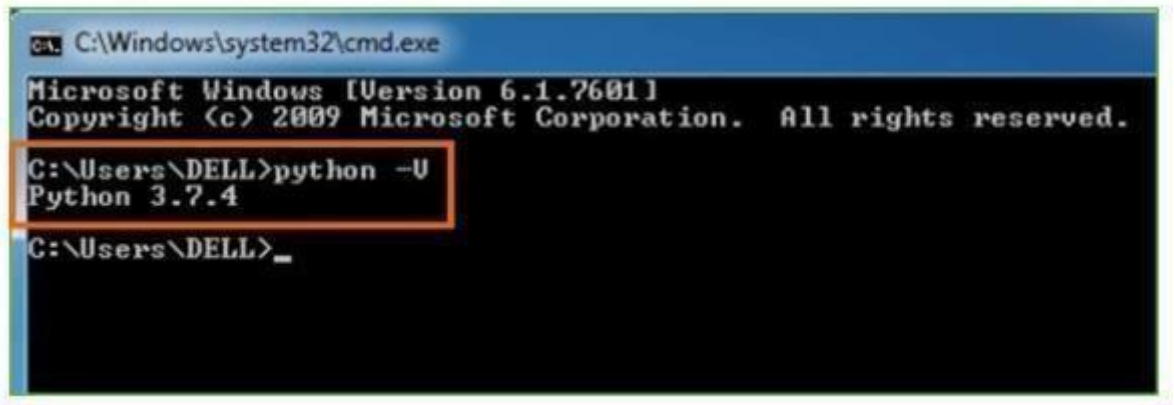
Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.



Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type python –V and press Enter.



Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

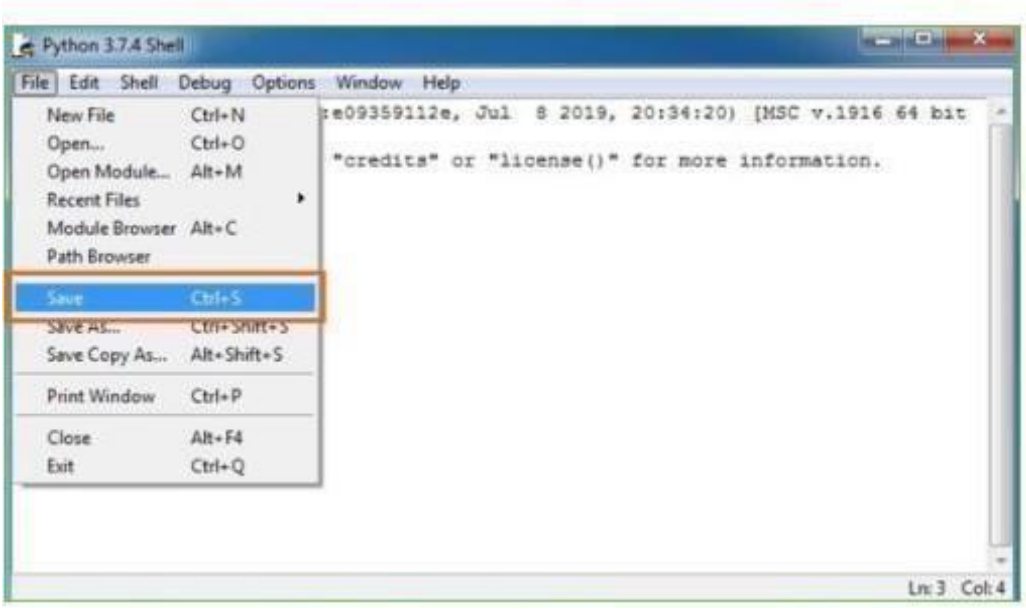
Check how the Python IDLE works

Step 1: Click on Start

Step 2: In the Windows Run command, type "python idle".

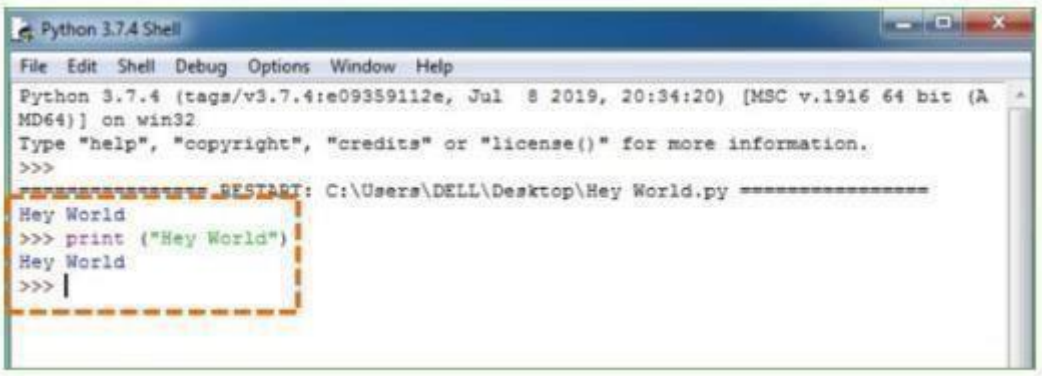
Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. Click on File > Click on Save



Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g., enter print ("Hey World") and Press Enter.



You will see that the command given is launched. With this, we end our tutorial on how to install Python. You have learned how to download python for windows into your respective operating system.

Note: Unlike Java, Python does not need semicolons at the end of the statements otherwise it won't work.

7. SYSTEM REQUIREMENTS SPECIFICATIONS

Requirements Specification:

Requirement Specification provides a high secure storage to the web server efficiently. Software requirements deal with software and hardware resources that need to be installed on a serve which provides optimal functioning for the application. These software and hardware requirements need to be installed before the packages are installed. These are the most common set of requirements defined by any operation system. These software and hardware requirements provide a compatible support to the operation system in developing an application.

7.1 Hardware Requirements:

The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

1. System : Pentium IV 2.4 GHz.
2. Hard Disk : 100 GB.
3. Monitor : 15 VGA Color.
4. RAM : 1 GB.

7.2 Software Requirements:

The software requirements specify the use of all required software products like data management system. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

For developing the application the following are the Software Requirements:

1. Python IDLE 3.7 version (or)
2. Anaconda 3.7 (or)
3. Jupiter (or)
4. Google Colab

8. FUNCTIONAL REQUIREMENTS

8.1 Output Design

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization
- Internal Outputs whose destination is within organization and theyare the
- User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.

Output Definition

The outputs should be defined in terms of the following points:

- Type of the output
- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

It is not always desirable to print or displaydata as it is held on a computer. It should be decided as which form of the output is the most suitable.

8.2 Input Design

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood bythe user.

Input Stages

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

Input Types

It is necessaryto determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department's communications to the system?
- Interactive, which are inputs entered during a dialogue.

Input Media

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

- Type of input
- Flexibility of format
- Speed
- Accuracy
- Verification methods
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easyto use
- Portability

Keeping in view the above description of the input types and input media, it can be said that

most of the inputs are of the form of internal and interactive. As Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

Error Avoidance

At this stage care is to be taken to ensure that input data remains accurate form the stage at which it is recorded up to the stage in which the data is accepted by the system. This can be achieved only by means of careful control each time the data is handled.

Error Detection

Even though every effort is made to avoid the occurrence of errors, still a small proportion of errors is always likely to occur, these types of errors can be discovered by using validations to check the input data.

Data Validation

Procedures are designed to detect errors in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data. Whenever an invalid data is keyed in, the system immediately prompts the user and the user has to again key in the data and the system will accept the data only if the data is correct. Validations have been included where necessary.

The system is designed to be a user friendly one. In other words the system has been designed to communicate effectively with the user. The system has been designed with popup menus.

8.3 User Interface Design

It is essential to consult the system users and discuss their needs while designing the user interface:

User Interface Systems Can Be Broadly Classified As:

- User initiated interface the user is in charge, controlling the progress of the user/computer dialogue. In the computer-initiated interface, the computer selects the next stage in the interaction.

In the computer-initiated interfaces the computer guides the progress of the user computer dialogue. Information is displayed and the user response of the computer takes action or displays further information.

User Initiated Interfaces

User initiated interfaces fall into two approximate classes:

- Command driven interfaces: In this type of interface the user inputs commands or queries which are interpreted by the computer.
- Forms oriented interface: The user calls up an image of the form to his/her screen and fills in the form. The forms-oriented interface is chosen because it is the best choice.

Computer-Initiated Interfaces

The following computer – initiated interfaces were used:

- The menu system for the user is presented with a list of alternatives and the user chooses one; of alternatives.
- Questions – answer type dialog system where the computer asks question and takes action based on the basis of the users reply.

Right from the start the system is going to be menu driven, the opening menu displays the available options. Choosing one option gives another popup menu with more options. In this way every option leads the users to data entry form where the user can key in the data.

Error Message Design

The design of error messages is an important part of the user interface design. As user is bound to commit some errors or other while designing a system the system should be designed to be helpful by providing the user with information regarding the error he/she has committed.

This application must be able to produce output at different modules for different inputs.

8.4 Performance Requirements

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be

designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for anysystem can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system
- The existing system is completely dependent on the user to perform all the duties

8.5 Non Functional Requirements

All the other requirements which do not form a part of the above specification are categorized as Non-Functional needs. A system perhaps needed to gift the user with a show of the quantity of records during info. If the quantity must be updated in real time, the system architects should make sure that the system is capable of change the displayed record count at intervals associate tolerably short interval of the quantity of records dynamic. Comfortable network information measure may additionally be a non-functional requirement of a system.

The following are the features:

1. Accessibility
2. Availability
3. Backup
4. Certification
5. Compliance
6. Configuration Management
7. Documentation
8. Disaster Recovery
9. Efficiency (resource consumption for given load)
10. Interoperability

8.6 Feasibility Study:

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operation Feasibility
- Economical Feasibility

8.6.1 Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new System?
- Will the proposed system provide adequate response to inquiries, regardless of the number or Locations of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

8.6.2 Operational Feasibility

User-friendly

Customer will use the forms for their various transactions i.e. for adding new routes, viewing the routes details. Also the Customer wants the reports to view the various transactions based on the constraints. These forms and reports are generated as user-friendly to the Client.

Reliability

The package wills pick-up current transactions on line. Regarding the old transactions, User will enter them in to the system.

Security

The web server and database server should be protected from hacking, virus etc

Portability

The application will be developed using standard open source software (Except Oracle) like Java, tomcat web server, Internet Explorer Browser etc these software will work both on Windows and Linux o/s. Hence portability problems will not arise.

Availability

This software will be available always.

Maintainability

The system uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front-end and the 2nd tier is the database, which uses My-Sql, which is the back-end.

The front-end can be run on different systems (clients). The database will be running at the server. Users access these forms by using the user-ids and the passwords.

8.6.3 Economic Feasibility

The computerized system takes care of the present existing system's data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports. It should be built as a web based application with separate web server and database server. This is required as the activities are spread throughout the organization customer wants a centralized database. Further some of the linked transactions take place in different locations.

9. METHODOLOGY

SDLC (Software Development Life Cycle) – Umbrella Model

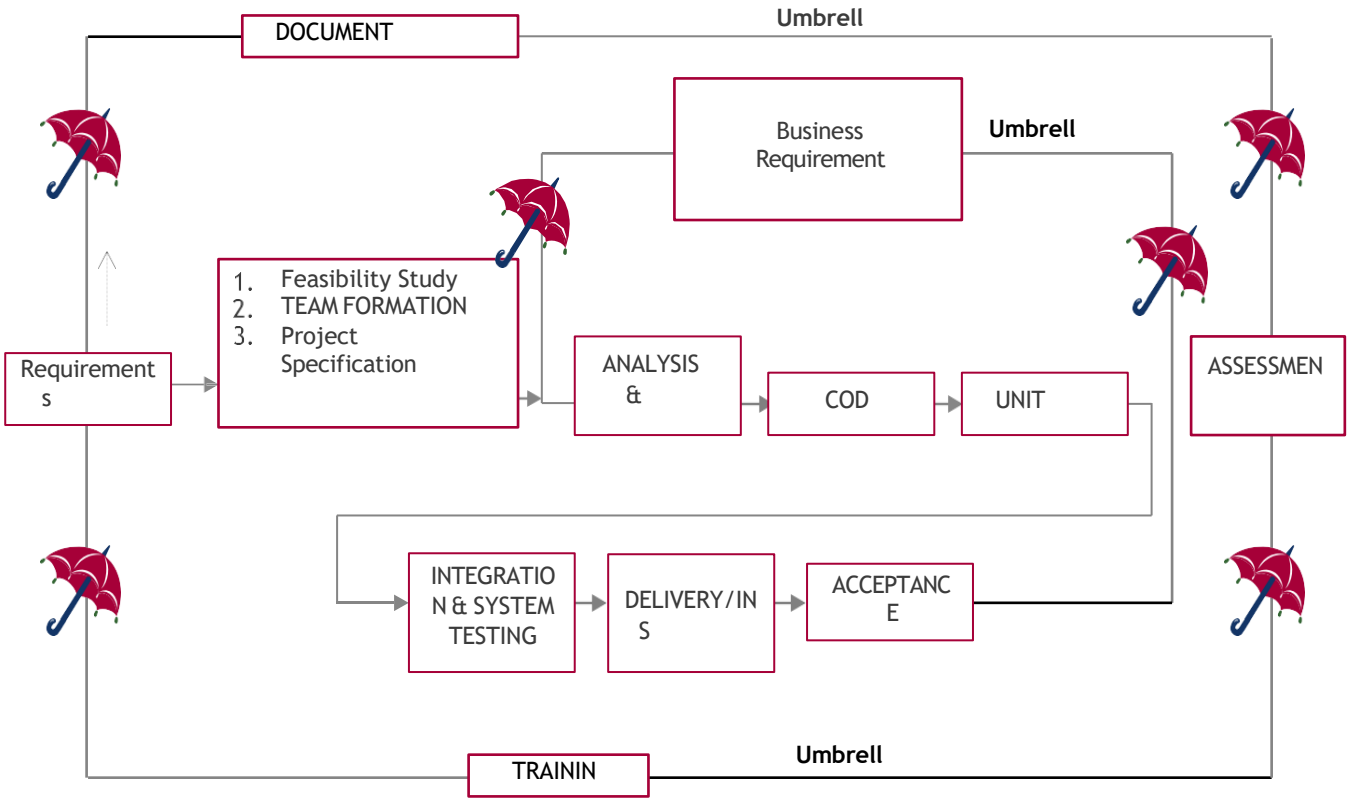


Fig no. 9.1 Umbrella model

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

Requirements Gathering Stage

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports.

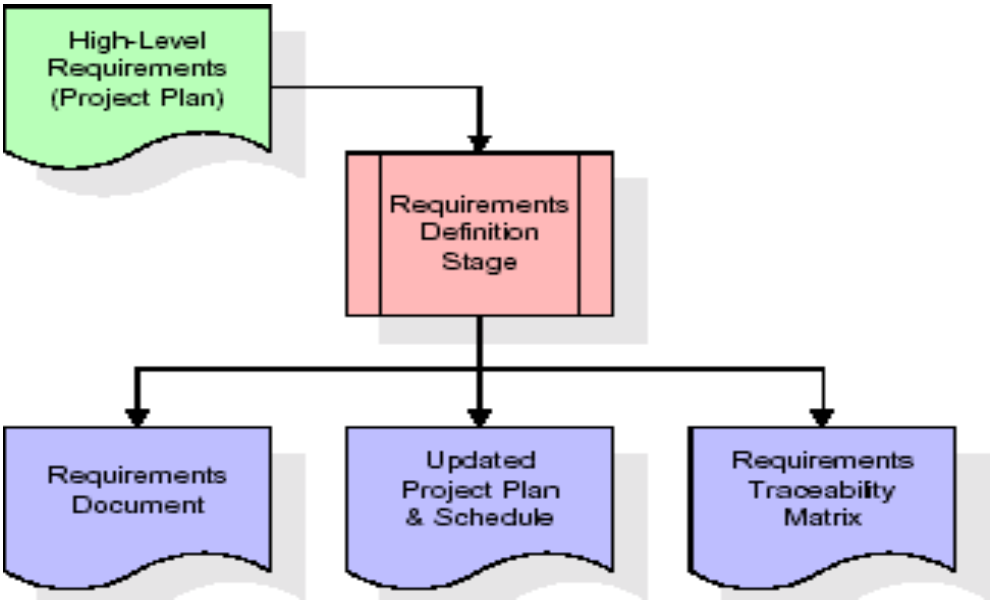


Fig no. 9.2 Requirements Gathering stage

These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are not included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

Analysis Stage

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.

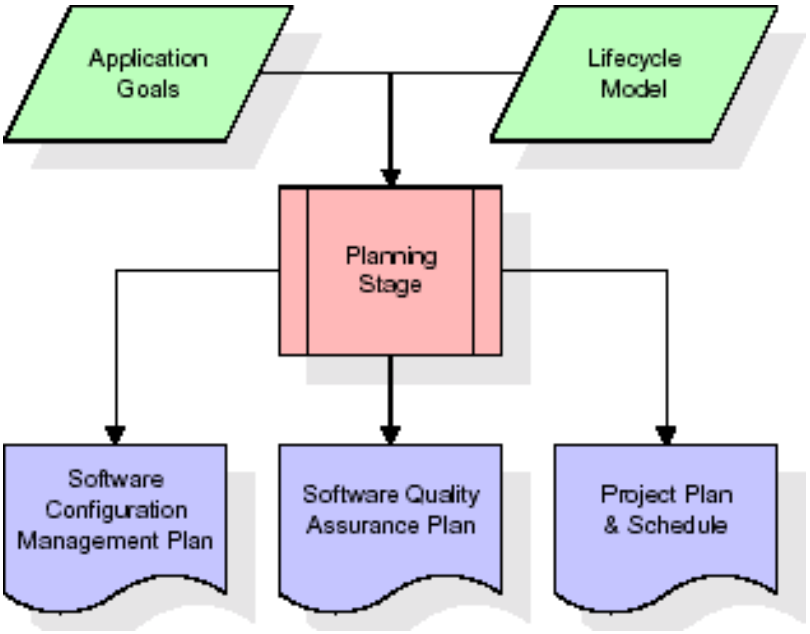


Fig no. 9.3 Analysis stage

The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

Designing Stage

The design stage takes as its initial input the requirements identified in the approved of the document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

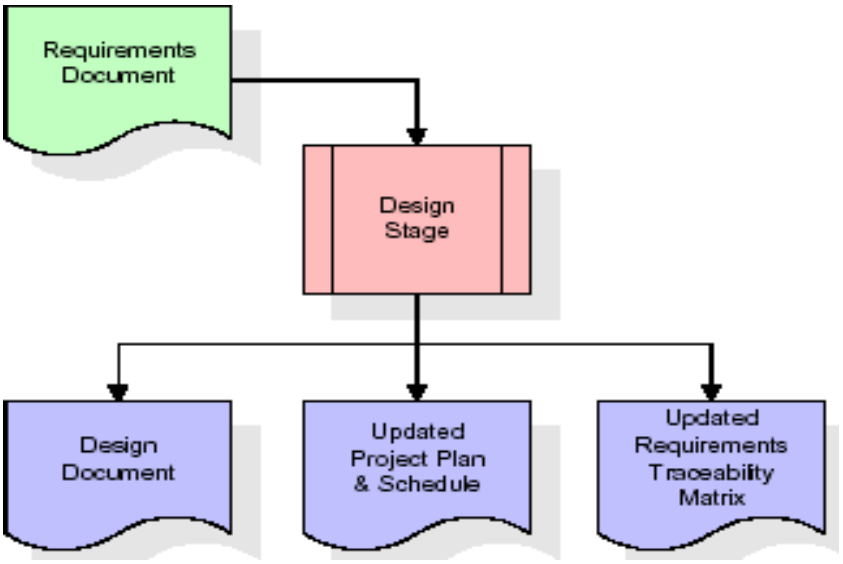


Fig no. 9.4 Designing stage

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

Development (Coding) Stage

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases

will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

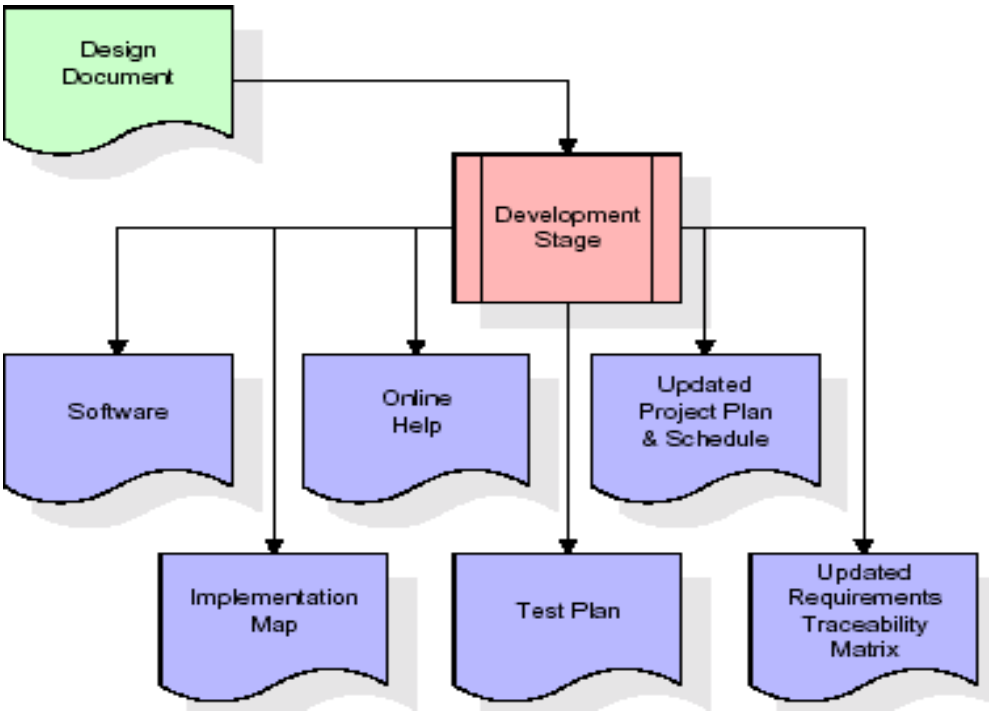


Fig no. 9.5 Coding stage

Integration & Test Stage

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

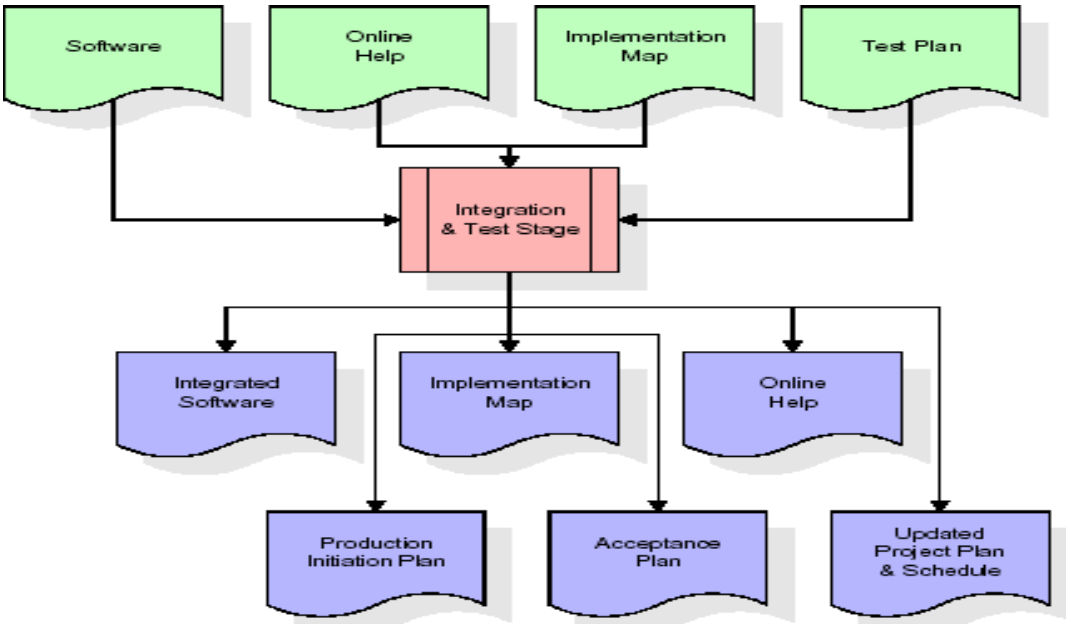


Fig no. 9.6 Integration and Testing Stage

Installation & Acceptance Test

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.

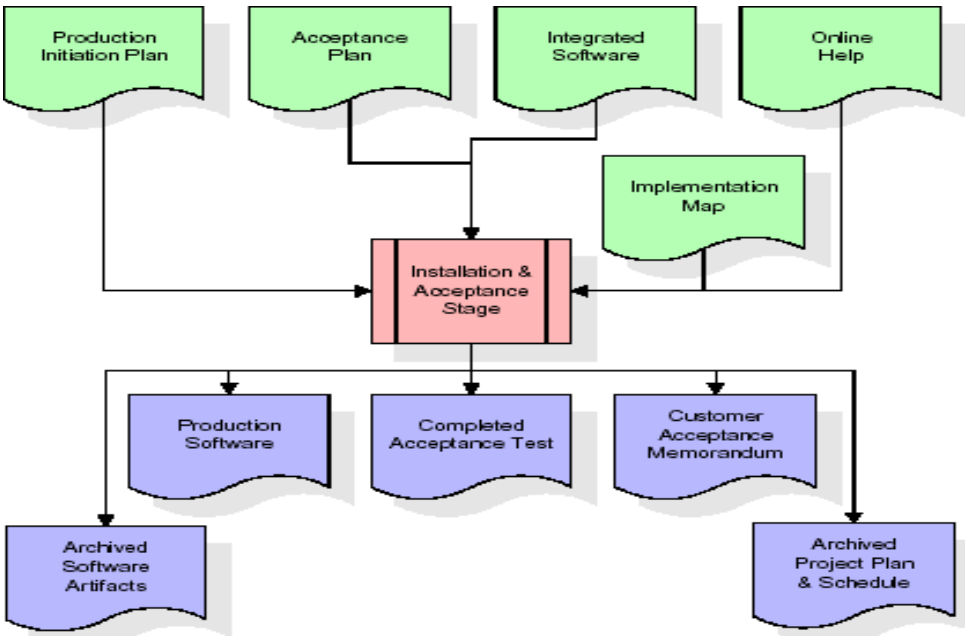


Fig no.9.7 Installation

Maintenance

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category.

10. SYSTEM TESTING

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. The following is the description of the testing strategies, which were carried out during the testing period.

10.1 System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

10.2 Module Testing

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

10.3 Integration Testing

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system

10.4 Acceptance Testing

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

11. SOURCE CODE

```
import cv2 as cv
import numpy as np
import mediapipe as mp
import math
import pyautogui
scroll = False
cAct = ""
pAct = ""
sCot=0
rbCot=0
lbCot=0
mp_face_mesh = mp.solutions.face_mesh
mp_drawing = mp.solutions.drawing_utils
# load face detection model
mp_face = mp.solutions.face_detection.FaceDetection(
    model_selection=1, # model selection
    min_detection_confidence=0.5 # confidence threshold
)

cap = cv.VideoCapture(0)

LEFT_IRIS = [474, 475, 476, 477]
RIGHT_IRIS = [469, 470, 471, 472]

LEFT_EYE =[ 362, 382, 381, 380, 374, 373, 390, 249, 263, 466, 388, 387, 386, 385,384, 398 ]
RIGHT_EYE=[ 33, 7, 163, 144, 145, 153, 154, 155, 133, 173, 157, 158, 159, 160, 161 , 246 ]

cursor_speed = 10
screen_width, screen_height = pyautogui.size()

def landmarksDetection(img, results, draw=False):
    img_height, img_width= img.shape[:2]
```

```
    # list[(x,y), (x,y). ..]
    mesh_coord = [(int(point.x * img_width), int(point.y * img_height)) for point in
results.multi_face_landmarks[0].landmark]
    if draw :
        [cv.circle(img, p, 2, (0,255,0), -1) for p in mesh_coord]

    # returning the list of tuples for each landmarks
    return mesh_coord

def euclaideanDistance(point, point1):
    x, y = point
    x1, y1 = point1
    distance = math.sqrt((x1 - x)**2 + (y1 - y)**2)
    return distance

def blinkRatio(img, landmarks, right_indices, left_indices):
    # Right eyes
    # horizontal line
    rh_right = landmarks[right_indices[0]]
    rh_left = landmarks[right_indices[8]]
    # vertical line
    rv_top = landmarks[right_indices[12]]
    rv_bottom = landmarks[right_indices[4]]
    # draw lines on right eyes
    # cv.line(img, rh_right, rh_left, utils.GREEN, 2)
    # cv.line(img, rv_top, rv_bottom, utils.WHITE, 2)

    # LEFT_EYE
    # horizontal line
    lh_right = landmarks[left_indices[0]]
    lh_left = landmarks[left_indices[8]]

    # vertical line
    lv_top = landmarks[left_indices[12]]
    lv_bottom = landmarks[left_indices[4]]
```

```
    rhDistance = euclaideanDistance(rh_right, rh_left)
    rvDistance = euclaideanDistance(rv_top, rv_bottom)

    lvDistance = euclaideanDistance(lv_top, lv_bottom)
    lhDistance = euclaideanDistance(lh_right, lh_left)

    reRatio = rhDistance/rvDistance
    leRatio = lhDistance/lvDistance

    ratio = (reRatio+leRatio)/2
    return reRatio,leRatio


with mp_face_mesh.FaceMesh(max_num_faces=1,refine_landmarks=True,
min_detection_confidence=0.5, min_tracking_confidence=0.5) as face_mesh:
    while True:
        ret, frame = cap.read()
        if not ret:
            break
        frame = cv.flip(frame, 1)
        rgb_frame = cv.cvtColor(frame, cv.COLOR_BGR2RGB)
        img_h, img_w = frame.shape[:2]
        results = mp_face.process(rgb_frame)

        if not results.detections:
            print('No faces detected.')
        else:
            for detection in results.detections: # iterate over each detection and draw on image
                mp_drawing.draw_detection(frame, detection)



        results = face_mesh.process(rgb_frame)
```

```
    if results.multi_face_landmarks:
        mesh_coords = landmarksDetection(frame,results,False)
        reRatio,leRatio = blinkRatio(frame,mesh_coords,RIGHT_EYE,LEFT_EYE)
        mesh_points=np.array([np.multiply([p.x, p.y], [img_w, img_h]).astype(int) for p in
results.multi_face_landmarks[0].landmark])

        (l_cx, l_cy), l_radius = cv.minEnclosingCircle(mesh_points[LEFT_IRIS])
        (r_cx,r_cy), r_radius = cv.minEnclosingCircle(mesh_points[RIGHT_IRIS])

        A= mesh_points[4]
        # cv.circle(frame,A, int(l_radius), (0, 255, 0), 1, cv.LINE_AA)

        if reRatio>5.5 and leRatio>5.5:
            cAct = "scroll"
            if(cAct == pAct):
                sCot = sCot+1
                if(sCot>=5):
                    sCot = 0
                    print("SCROLL SWITCH ")
                    if scroll:
                        print("-- OFF")
                        scroll = False
                    else:
                        print("-- ON")
                        scroll = True



        if not scroll:
            if reRatio >5.5 and leRatio<5.5:
                cAct = "rblink"
                if(cAct == pAct):
                    rbCot = rbCot+1
                    if(rbCot>3):
                        rbCot=0
```

```
            print("RIGHT BLINK")
            pyautogui.click(button='right')
            cv.circle(frame,(240,320),5,(255,0,0),3)
    #CEF_COUNTER +=1

    if leRatio >5.5 and reRatio<5.5:
        cAct = "lblink"
        if(cAct == pAct):
            lbCot = lbCot+1
            if(lbCot>3):
                lbCot=0
                print("LEFT BLINK")
                pyautogui.click(button='left')
                cv.circle(frame,(240,320),5,(0,0,255),3)

center_left = np.array([l_cx, l_cy], dtype=np.int32)
center_right = np.array([r_cx, r_cy], dtype=np.int32)

cv.circle(frame, center_left, int(l_radius), (255, 0, 0), 1, cv.LINE_AA)
cv.circle(frame, center_right, int(r_radius), (255, 0, 0), 1, cv.LINE_AA)

cursor_xo = int((center_left[0] + center_right[0]) / 2)
cursor_yo = int((center_left[1] + center_right[1]) / 2)

print(cursor_xo,cursor_yo)

cursor_x = int(cursor_xo * screen_width / frame.shape[1])
cursor_y = int(cursor_yo * screen_height / frame.shape[0])
nose_y = int(A[1]*screen_height/frame.shape[0])

if scroll:
    if cursor_yo < 210:
        print("SCROLLING UP... ")
        pyautogui.scroll(40)
    if cursor_yo >210 and cursor_yo<223:
```

```
                print("NO SCROLL .... ")
            if cursor_yo > 223:
                print("SCROLLING DOWN. ..")
                pyautogui.scroll(-40)

        try :
            pyautogui.moveTo((cursor_x-280)*1.5, (cursor_y-350)*4, duration=0.1)
        except:
            print("OUT OF FRAME !!")
        pAct = cAct
    cv.imshow("img", frame)
    key = cv.waitKey(1)
    if key ==ord("q"):
        break
cap.release()
cv.destroyAllWindows()
```

12. RESULTS AND DISCUSSIONS

12.1 Implementation Description

This project is mainly predicting the eyeball movements. At first we need to identify the facial landmarks By the use of landmarks we can easily identify the movements of the eye. We can detect eyeball movements, eye blinks in a video and also predict emotions. Understanding the Dlib's facial landmark finder. Dlib's model, allows us to predict the 68 2D facial landmarks accurately. The 68 facial landmarks picture is s shown below.

Fig 12.1.1 Picturing the 68 facial landmark positions

Only the eyes are taken into account while determining eye ball motions. The eye is designated by 6 (x, y) coordinates, starting from the far left corner and proceeding clockwise to the right, encompassing the remaining region of the eye, as seen in the image.

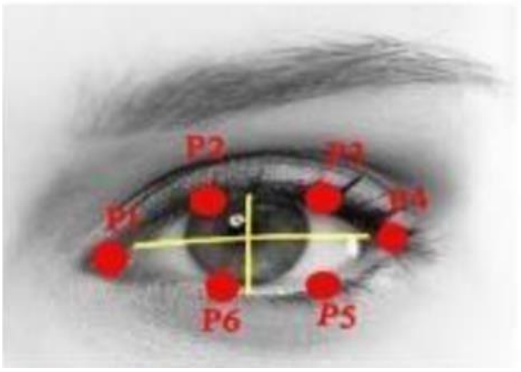


Fig 12.1.2 The 6 facial points linked with the eye

Based on the study work done in real-time Eye Blink Detection utilising Face Landmarks system, an equation can be derived that satisfies the relationship between all 6 facial co-ordinates known as eye aspect ratio and may be computed as follows:

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

In the equation above, points p1 through p6 are two-dimensional facial landmark positions. The numerator of the equation is substituted with the distance between the eye's vertical points, and the denominator with the distance between the eye's horizontal points. The eye aspect ratio is nearly constant when the eyes are wide open, but it goes to zero when the person blinks.

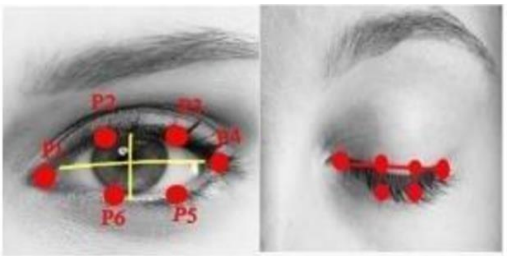


Fig 12.1.3 Landmarks of the eye when the eye is fully open (left) and landmarks of the eye when eye is closed(right)

When the user's eyes are fully open, the eye aspect ratio will be greater and will remain consistent throughout time, as shown in Figure 4. (Left). When a person blinks, the eye aspect ratio drops dramatically and approaches zero.Furthermore, the eye aspect ratio remains constant throughout time and gradually approaches zero. The number then climbs, indicating that the subject has blinked once.

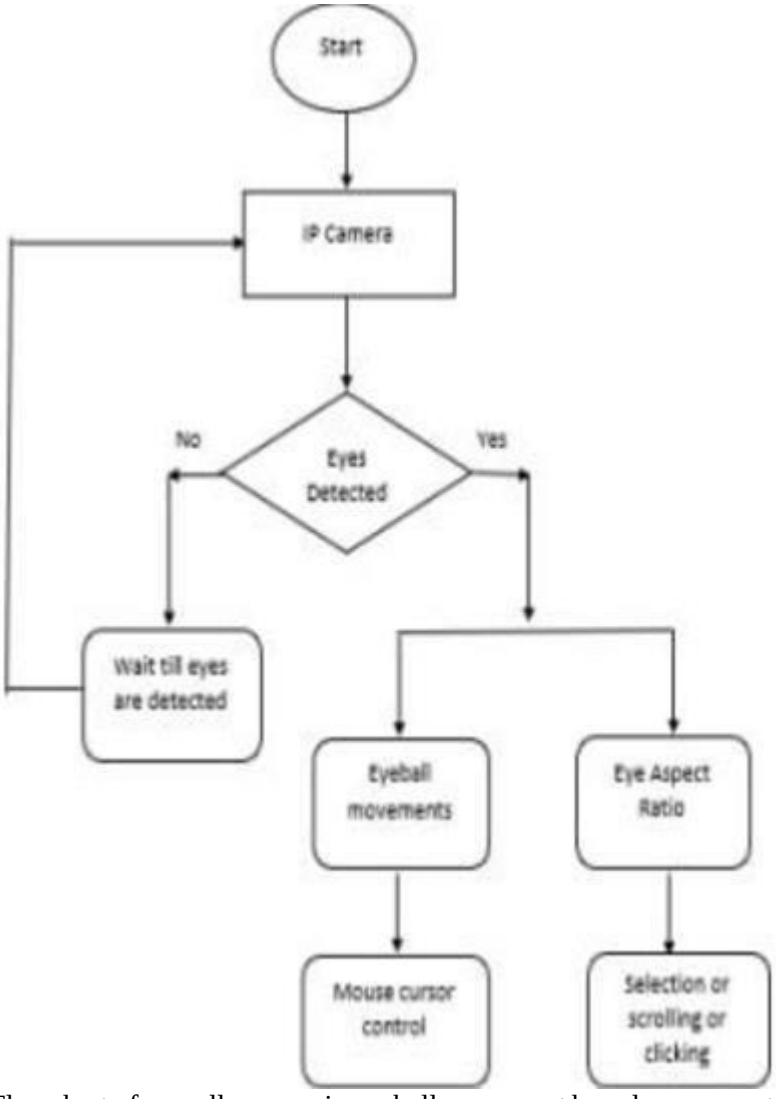


Fig 12.1.4 Flowchart of overall process in eyeball movement based cursor control

The flowchart above depicts the overall procedure for controlling a cursor using ocular movement using a Raspberry Pi and OpenCV. The Raspberry Pi is the core aspect of the processing module, which uses an Internet Protocol camera to track facial expression. The camera captures the image after waiting for the eyes to be recognised. OpenCV technology's photo handling method is used to distinguish eyes. The mouse pointer can be controlled based on eyeball movement, and blinking of the eyes is used to calculate the Eye Aspect Ratio (EAR) to execute various actions such as clicking, scrolling or selecting.

12.2 Results

Change the current working directory before starting the Python software, and then launch the Python software by entering the commands in the terminal as shown in Figure 6. We obtain a frame window after typing the commands in the terminal, and the camera starts capturing the face that reads the input. if the user wants to move the cursor upward, the Anchor point should be moved to up.

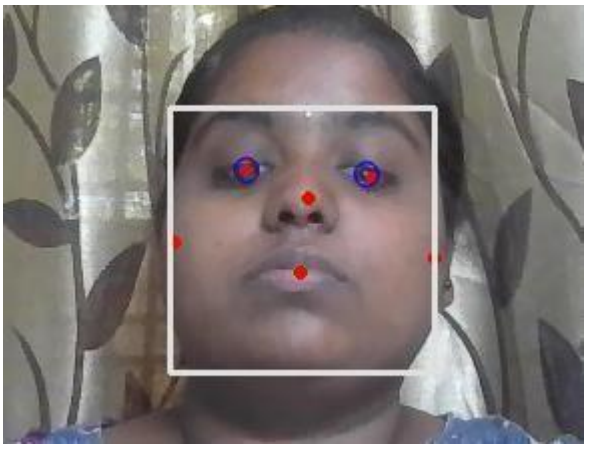


Fig 12.2.1 If the user wants to move the cursor towards upward direction then the user should move the Anchor point to up

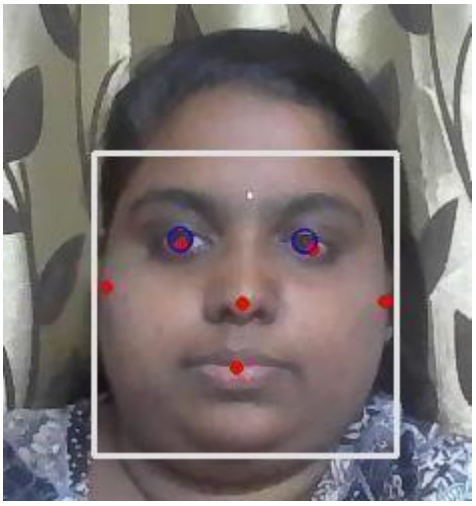


Fig 12.2.2 If the user wants to move the cursor towards downward direction then the user should move the Anchor point to down

Fig 12.2.3 If the user wants to move the cursor towards left direction then the user should move the Anchor point to left

Fig 12.2.4 If the user wants to move the cursor towards right direction then the user should move the anchor point to right

13. CONCLUSION AND FUTURE SCOPE

Conclusion

The cursor control using eye ball movement. Without using the hands we can operate the computers. This technology is contracted to replace the conventional computer screen pointing device for the use of disabled persons. The movement of cursor automatically moves by adjusting the position where of eyesight. It is mostly used for the disabled and paralysed people. Without the help of other person they can use the computers. This work can be extended to implement efficient movements to perform the click events and also cover the total mouse function of the system and to cover the total human-computer interface system using eye blink. Technology also extended to the eyeball movement and eye blinking to get the efficient and accurate movement. In future, many people who are unable to operate a standard computer mouse or keyboard because of disabilities of their hands or arms, can get possible alternative in multimodal system, which allows controlling a computer without using standard mouse and keyboard. Using head movements to control the cursor across the computer screen and by using the speech for giving the control commands. Automatic speech recognition and head tracking in joint multimodal action are combined to operate the system.

Future Scope

The concept of controlling a computer cursor using eye movement presents numerous possibilities for future development and enhancement. As technology continues to advance, the system can be refined to offer greater accuracy, faster response times, and improved user experiences. The future scope of this technology spans multiple domains, including assistive technology, gaming, healthcare, and smart home automation.

References

[1]. Khare, Vandana, S. Gopala Krishna, and Sai Kalyan Sanisetty. "Cursor Control Using Eye Ball Movement." In 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), vol. 1, pp. 232-235. IEEE, 2019.

[2]. Neil Castellino and Michelle Alva, “An image-based eye controlled assistive system for paralytic patients”, IEEE Conference Publications, 2017.

[3]. Bukhalov, Aleksei, and Viktoriia Chafonova. "An eye tracking algorithm based on hough transform." In 2018 International Symposium on Consumer Technologies (ISCT), pp. 49-50. IEEE, 2018.

[4]. Shu-Fan Lin, Xuebai Zhang, ShyanMing Yuan, "Eye Tracking Based Control System for Natural Human Computer Interaction", Computational Intelligence and Neuroscience ,2017.

[5]. Geetha and M. Mangaiyarkarasi, “Cursor Control System Using Facial Expressions for Human-Computer Interaction”, Vol 8 Issue 1 APRIL 2014, ISSN: 0976-1353 International Journal of Emerging Technology in Computer Science & Electronics, pp 30-34.

[6]. Florina Ungureanu, Robert Gabriel Lupu and Valentin Siriteanu, “Eye tracking mouse for human-computer interaction”, IEEE Conference Publications, 2013.

[7]. Suree Pumrin and Chairat kraichan, “Face and eye tracking for controlling computer functions”, IEEE Conference Publications, 2014.

[8]. Kanchan Pradhan, Sahil Sayyed, Abhishek Karhade, Abhijeet Dhumal and Sohail Shaikh, "Eyeball Movement Based Cursor Using Machine Learning", International Research Journal of Modernization in Engineering Technology and Science, April,2023.

[9]. Mohamed Nasor et al., "Eye-controlled mouse cursor for a physically disabled individual", 2018 Advances in Science and Engineering Technology International Conferences (ASET),2018.

[10]. Sharon Mathew et al., "Eye movement-based cursor control and home automation for disabled people", 2019 International Conference on Communication and Electronics Systems (ICCES), 2019. [12]. Rane Shubham et al., "Eyeball Movement-Based Cursor Using Deep Learning", International Research Creative Research Thoughts (IJCRT), June ,2021.

[11]. Cheng- Chih Wu and Ting-Yun Hou, Tracking Students' Cognitive Processes During Program Debugging‖—An Eye-Movement Approach, IEEE, 2019.

[12]. Qiang Ji and Zhiwei Zhu, Novel Eye Gaze Tracking Techniques Under Natural Head Movement, IEEE, vol. 54, no. 12, DECEMBER , 2007.

[13]. Seung-Jin Baek and Young-Hyun Kim, Eyeball Model-based Iris CenterLocalization for Visible Image based Eye-Gaze Tracking Systems, IEEE, 2013.

[14]. Jun-Seok Lee, Kyung-hwa Yu, Sang-won Leigh, Jin-Yong Chung and Sung-Goo Cho, "Method for controlling device on the basis of eye ball motion and device therefor", January ,2018.

[15]. S. K. B. Sangeetha, "A survey on Deep Learning Based Eye Gaze Estimation Methods", *Journal of Innovative Image Processing (JIIP)*, vol. 3, no. 03, pp. 190-207, [6]. Florina Ungureanu, Robert Gabriel Lupu and Valentin Siriteanu, "Eye tracking mouse for human- computer interaction", IEEE Conference Publications, 2021.

[16]. A. Sivasangari et al., "Eyeball-based cursor movement control", 2020 International conference communication single processing ,2020.