Digital Image Processing Laboratory

Mini Project Report

on

EYE TRACKING USING OPENCV

Submitted by

SANGAMESH.J.B — **4SN20AI021**

SUHAS.R.S — **4SN20AI025**

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

(Visvesvaraya Technological University)

Under the Guidance of

Mrs. Daya Naik

Assistant Professor

Department of Artificial Intelligence and Machine Learning

SRINIVAS INSTITUTE OF TECHNOLOGY

[NAAC ACCREDITED]

MANGALURU-574143, KARNATAKA

2022 – 2023

SRINIVAS INSTITUTE OF TECHNOLOGY

[NAAC ACCREDITED]
MANGALURU -574143 –
KARNATAKA

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINELEARNING

CERTIFICATE

*This is to certify that the project entitled **"EYE TRACKING USING OPENCV ",** is an authentic record of the work carried out by*

SANGAMESH.J.B **4SN20AI021**

SUHAS.R.S **4SN20AI025**

as prescribed by Visvesvaraya Technological University, Belagavi, for VI Semester B.E. inArtificial Intelligence and Machine Learning during the year 2022-2023.

Mrs.Daya Naik	**Dr. Anoop.B.K**	**Dr. Shrinivasa Mayya D**
Project Guide	**Head of the Department**	**Principal**

Name of the Examiners	**Signature with Date**
1.	
2.	

ABSTRACT

This project aims a comprehensive study on the implementation of an eye tracking system using OpenCV, NumPy, dlib, and math libraries. The system utilizes advanced techniques in computer vision to accurately detect and track the movement of the eyes in real-time. By leveraging facial landmarks and eye segmentation, the system is able to precisely locate and analyze the behavior of the eyes. The project focuses on developing algorithms that process the captured video frames from the front camera and extract relevant features related to eye movements. By employing mathematical calculations using NumPy and OpenCV, the system determines the extent of left and right eye movements, enabling the quantification of gaze direction and eye blink patterns. The results of this eye tracking system can be displayed in real-time on the screen, providing valuable visual feedback for users. The system can also record and analyze eye movement data for further analysis and research purposes. Additionally, the project highlights the potential applications of eye tracking technology in areas such as human-computer interaction, psychology, medicine, and usability testing.

ACKNOWLEDGEMET

The satisfaction that accompanies the successful completion of any work would be incomplete without thanking the persons who made it perfect with their constant guidance and encouragement.

I take this opportunity to express my sincere thanks and indebtedness to my Project guide and co-ordinator, **Mrs. Daya Naik Assistant Professor, Department of Artificial Intelligence and Machine Learning**, for her support and guidance. Her vision and suggestions throughout the project period has been fundamental in the completion of the project.

I extend my warm gratitude to **Dr. Anoop B.K, H.O.D, Department of AIML**, for his constant support and advice that helped me to complete this project successfully.

I am extremely grateful to our beloved **Principal, Dr. Shrinivasa Mayya D** for his encouragement to come up with new ideas and advice to express them in a systematic manner.

I also like to thank all Teaching & Non-teaching staff of Srinivas Institute of Technology, Mangalore for their kind co-operation during the course of my work.

Finally, I am extremely thankful to my family and friends who helped me in my work and made the project a successful one.

Sangamesh.J.B

Suhas.R.S

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Eye tracking is a technology that holds significant potential in various fields, including human-computer interaction, medical research, and usability testing. It enables the measurement and analysis of eye movements, providing valuable insights into visual attention, cognitive processes, and user behavior. In recent years, the availability of powerful computer vision libraries, such as OpenCV, has facilitated the development of robust and efficient eye tracking systems.

The aim of this project is to design and implement an eye tracking system using OpenCV, along with complementary libraries such as NumPy, dlib, and math. The system utilizes the front camera of a device to capture real-time video framesand processes them to accurately track the movement of the eyes. By leveragingfacial landmarks and eye segmentation techniques, the system identifies and analyzes key features necessary for eye tracking.

The primary objectives of this project are as follows:

1. Detect and localize the eyes in real-time video frames using faciallandmarks and eye segmentation.
2. Track the movement of the eyes, enabling the measurement of gazedirection, eye rotations, and blinks.
3. Display the eye tracking results in real-time, providing immediate visualfeedback to the user.
4. Develop a reliable and efficient system that can be used for various applications, such as human-computer interaction, medical diagnostics. To achieve these objectives, the project will involve a series of steps, including preprocessing of video frames, detection of facial landmarks, segmentation of eyes, feature extraction, and mathematical calculations for analyzing eye movements. The implementation will make use of the OpenCV library for computer vision tasks, NumPy for numerical computations, dlib for facial landmark detection, and math for mathematical operations.

The outcomes of this project will contribute to the existing body of knowledge in image processing, computer vision, and eye tracking. The developed system willprovide a foundation for further research and practical applications in fields suchas human-computer interaction, psychology, and medicine.

1.1 Problem statement:

The aim of this project is to develop an eye tracking system using OpenCV and other relevant libraries to accurately track and analyze the movement of the eyes in real-time. The system should be able to detect and localize the eyes in video frames captured by the front camera, calculate the extent of left and right eye movements, and accurately count the number of blinks. The project aims to address the following challenges:

i. Eye Localization: Developing an algorithm to accurately detect and localize the eyes in real-time video frames, considering variations in lighting conditions, head movements, and different eye shapes and sizes.

ii. Eye Tracking: Implementing a robust eye tracking mechanism that can accurately track the movement of the eyes, including gaze direction, eye rotations, and blinks, while handling occlusions and other potential obstacles.

iii. Real-time Processing: Designing an efficient system that can process video frames in real-time, ensuring minimal latency and smooth performance, thereby enabling the system to provide immediate visual feedback to the user.

iv. Mathematical Analysis: Employing mathematical calculations and algorithms to quantify the extent of left and right eye movements, enablingthe system to provide precise measurements of eye behavior and gaze patterns.

v. User Interface: Creating a user-friendly interface to display the eye tracking results on the screen in a clear and understandable manner,allowing users to interpret and analyze the data easily

1.2 Objectives: -

i. To detect and localize the eyes accurately in real-time video frames captured by the front camera.

ii. To track the movement of the eyes, including gaze direction, eye rotations,and blinks.

iii. To calculate the extent of left and right eye movements using mathematicalcalculations and algorithms.

iv. To count the number of blinks accurately in real-time.

v. To develop a user-friendly interface to display the eye tracking results onthe screen.

vi. To ensure real-time processing of video frames with minimal latency.

vii. To evaluate the performance and accuracy of the eye tracking system usingvarious test scenarios and datasets.

viii. To explore potential applications of the eye tracking system in areas suchas human-computer interaction, medical research, and usability testing.

ix. To contribute to the field of computer vision and image processing byshowcasing the capabilities and potential of eye tracking technology.

By achieving these objectives, the project aims to provide a functional and reliable eye tracking system that can accurately track and analyze eye movement in real-time.

.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 Software specification

The eye tracking system developed in this project will be implemented using thefollowing software specifications:

1. Programming Language: Python - Python is a widely used programming language known for its simplicity and versatility. It offers a rich set of libraries and frameworks that are well-suited for computer vision and image processing tasks.
2. Libraries:
   a. OpenCV: OpenCV (Open-Source Computer Vision Library) is a popular open-source library that provides a comprehensive set of functions and algorithms for computer vision tasks. It offers efficientimage and video processing capabilities, including features like image filtering, object detection, and feature extraction, which are essential for eye tracking.
   b. NumPy: NumPy is a fundamental library for scientific computing inPython. It provides powerful numerical and mathematical operations, making it useful for calculations related to eye movement analysis and feature extraction.
   c. Dlib: Dlib is a robust library that offers state-of-the-art algorithms for facial landmark detection. It provides accurate localization of facial landmarks, including eye landmarks, which will be crucial foreye detection and tracking.
   d. math: The math library in Python provides various mathematical functions that will be utilized for calculations related to eye movement analysis, including angle calculations and statisticaloperations.
3. Development Environment: The development environment for the eye tracking system can beset up using Python's standard development tools or integrated development environments (IDEs) such as PyCharm, JupyterNotebook, or Visual Studio Code. These tools provide a convenient interface for coding, debugging, and executing Python scripts.

4. Hardware Requirements: The eye tracking system primarily relies on the front camera of a device, such as a laptop or a webcam, to capture real- time video frames. The hardware requirements include a device with a front camera capable of capturing video, along with sufficient processing power and memory to perform real-time video processing.

5. Compatibility: The developed eye tracking system should be compatible with different platforms, including Windows, macOS, and Linux. Compatibility with different versions of Python and the required librariesshould also be ensured to maximize usability and flexibility.

The software specification outlined above provides the foundation for developing a robust and efficient eye tracking system using OpenCV, NumPy,dlib, and math libraries. These specifications enable the implementation of key functionalities, such as eye detection, tracking, mathematical calculations, andreal-time processing, leading to an accurate and reliable eye tracking System

2.2 Hardware specification

The eye tracking system developed in this project requires the following hardwarespecifications:

1. Device: A computer or device capable of running the software and supporting real-time video processing. This can include laptops, desktop computers, or embedded systems with sufficient processing power and memory.
2. Front Camera: The system relies on a front-facing camera to capture real- time video frames of the user's face. The camera should be capable of capturing clear and high-resolution video, ensuring accurate eye detectionand tracking. Common options include built-in laptop/webcam cameras orexternal USB cameras.
3. Processor: A processor with sufficient speed and capabilities to handle real-time video processing. Ideally, a multicore processor with a clock speed of 2 GHz or higher is recommended to ensure smooth performance.
4. Memory: Adequate RAM is essential for efficient processing of video frames and running the necessary algorithms. A minimum of 4 GB of RAM is recommended, although higher amounts of RAM can be beneficial for more demanding applications.
5. Graphics Processing Unit (GPU): While not mandatory, a dedicated GPU can significantly accelerate certain computational tasks related to computervision and image processing. GPUs with CUDA support can be leveragedto speed up certain operations performed by the eye tracking system.
6. Display: A display screen with sufficient resolution and color accuracy to visualize the eye tracking results. This can be the built-in display of the device or an external monitor.
7. Operating System: The eye tracking system should be compatible with common operating systems such as Windows, macOS, or Linux. Compatibility with the specific version of the operating system should be ensured to maximize usability.
8. Connectivity: Adequate ports or wireless connectivity options should be available to connect the front camera to the device. This can include USB ports or wireless connections such as Wi-Fi or Bluetooth.
9. Power Supply: A reliable power source is required to ensure uninterruptedoperation of the device during eye tracking sessions. This can include a power outlet for desktop computers or a battery for laptops and portable devices.

CHAPTER 3

METHODOLOGY

3.1 Basic Information

The development of the eye tracking system involves the following steps and

Methodologies:

i. Data Acquisition: The system utilizes the front camera of a device to capture real-time video frames. The video frames are continuously acquired and processed for eye tracking.

ii. Preprocessing: The acquired video frames undergo preprocessing steps to enhance the quality and reduce noise. This can include operations such as resizing, grayscale conversion, and noise reduction techniques.

iii. Facial Landmark Detection: The dlib library is employed to detect and localize facial landmarks, including the landmarks corresponding to the eyes. These landmarks serve as reference points for accurately identifyingthe eye regions in the video frames.

iv. Eye Segmentation: Using the detected facial landmarks, the eye regions are segmented from the video frames. This process isolates the eyes and facilitates subsequent eye tracking operations.

v. Eye Tracking: The segmented eye regions are tracked across consecutive video frames to determine eye movements. Various techniques can be usedfor eye tracking, such as optical flow, template matching, or feature-based tracking algorithms. The chosen technique should handle variations in lighting conditions, occlusions, and other challenges.

vi. Mathematical Calculations: Using the tracked eye regions, mathematical calculations are performed to quantify the extent of left and right eye movements. These calculations may involve measuring angles, distances, or pixel movements, depending on the specific requirements of the eye tracking system.

vii. Blinks Detection: By analyzing the eye regions and their movements, the system can identify blinks. Blinks are typically detected based on changesin eye aspect ratio or changes in the intensity profile of the eye region.

viii. Real-time Display: The eye tracking results, including eye movements and blink counts, are displayed on the screen in real-time. This can be implemented through a graphical user interface (GUI) or overlaying the results on the video frames.

ix. Performance Evaluation: The developed eye tracking system is evaluated using various test scenarios and datasets to assess its accuracy, robustness,and real-time performance. The evaluation involves comparing the system's results with ground truth data or manual annotations.

x. Iterative Refinement: Based on the evaluation results, the system is refinedand optimized to improve accuracy, efficiency, and user experience. Iterative refinement may involve adjusting parameters, employing advanced algorithms, or incorporating user feedback.

The implementation of the eye tracking system utilizes the OpenCV, NumPy, dlib, and math libraries in Python. These libraries provide essential functions, algorithms, and mathematical tools required for image processing, feature extraction, and mathematical calculations.

Throughout the development process, documentation and version control practices are followed to ensure maintainability and reproducibility of the system.

By following this methodology, the eye tracking system can accurately detect, track, and analyze eye movements in real-time, providing valuable insights into visual attention and human behavior.

3.2 FLOW CHART

Description:

The flowchart begins here.The system captures video input from a camera . An if condition is evaluated based on a certain criterion . If the condition is false, it means that the system cannot detect the required landmarks properly, and an error message is displayed. If the condition in the if statement is true, the system proceeds to detect the required landmarks in the captured video. After successfully detecting the landmarks, the system focuses on analysing eye movements. It determines if the eyes are looking to the left, right, or center.Based on the eye movement analysis, the system generates a result or output. The result could be displayed on a screen. The flowchart is graphically in displayed below in figure 3.2.1

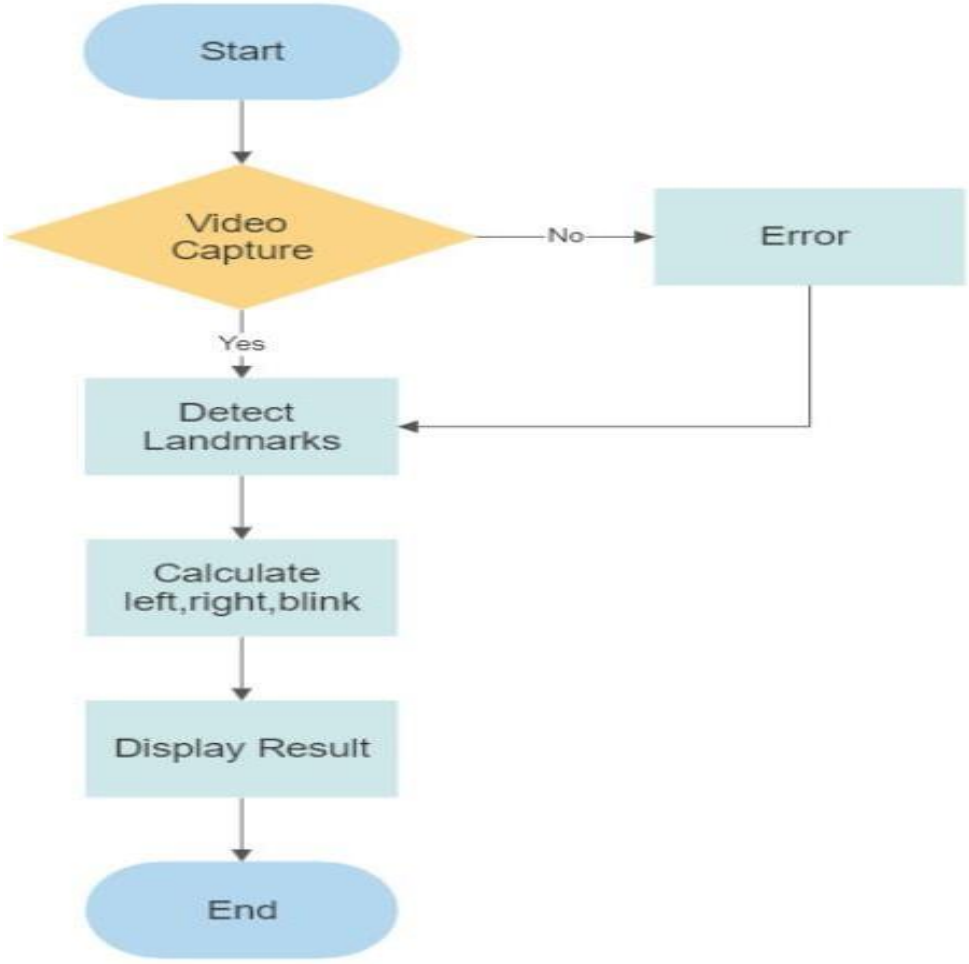

Figure 3.2.1: Flow chart of Eye Tracking

CHAPTER 4

IMPLEMENTATIONS

The implementation of the "Eye Tracking using OpenCV" project involves several key steps and components. Here is an overview of the main implementations:

1. Setting up the Development Environment:
    a. Install Python and relevant libraries such as OpenCV, NumPy, dlib,and math.
    b. Set up the chosen integrated development environment (IDE) or use the command line for coding.
2. Video Capture and Preprocessing:
    a. Access the front camera using OpenCV's Video Capture module to capture real-time video frames.
    b. Preprocess the frames by resizing, converting to grayscale, and applying noise reduction techniques to improve accuracy.
3. Facial Landmark Detection:
    a. Utilize the dlib library to detect and localize facial landmarks,particularly the landmarks corresponding to the eyes.
    b. Extract the eye regions using the detected landmarks as reference points.
4. Eye Segmentation:
    a. Segment the eye regions from the video frames to isolate the eyes for further analysis.
    b. Apply techniques such as image thresholding or image segmentationalgorithms to obtain accurate eye regions.
5. Eye Tracking:
    a. Track the eye regions across consecutive video frames using techniques like optical flow, template matching, or feature-based tracking.

6. Mathematical Calculations:
    a. Perform mathematical calculations and computations using NumPyand math libraries to quantify the extent of left and right eyemovements.
    b. Calculate angles, distances, or other relevant metrics to analyze eyebehavior and gaze patterns.
7. Blinks Detection:
    a. Implement algorithms to detect blinks based on changes in eyeaspect ratio or intensity profile of the eye regions.
    b. Count the number of blinks in real-time by monitoring thechanges in eye state.
8. Real-time Display and Visualization:
    a. Develop a user-friendly interface to display the eye tracking resultsin real-time.
    b. Overlay the eye movements, gaze direction, blink counts, and otherrelevant information on the video frames or display them in a separate window.
9. Performance Evaluation and Optimization:
    a. Evaluate the performance and accuracy of the eye tracking systemusing various test scenarios and datasets.
    b. Compare the results with ground truth data or manual annotationsto assess the system's effectiveness.
    c. Optimize the algorithms, parameters, and processing techniques toenhance accuracy, real-time performance, and robustness.
10. Documentation and Reporting:
    a. Document the implementation details, including the methodologies, algorithms, and libraries used.
    b. Provide a comprehensive project report outlining the system's functionalities, performance, and potential applications.
    c. Include code documentation and instructions for future reference and reproducibility.

CHAPTER 5

RESULTS/SNAPSHOTS

The "Eye Tracking using OpenCV" project produces real-time results that can be visualized and analyzed. Here are some of the possible results and snapshots:

1. Eye Localization: The system successfully detects and localizes the eyes in the video frames captured by the front camera. Snapshots can display the outlined eye regions or bounding boxes around the eyes.
2. Eye Tracking: The implemented eye tracking mechanism accurately tracks the movement of the eyes in real-time. Results can be shown through visualindicators such as arrows or lines representing the gaze direction and eye movements.
3. Gaze Heatmap: By aggregating the eye tracking data over time, a gaze heatmap can be generated to visualize the areas of the screen that receivedthe most visual attention. The heatmap can be overlaid on the video frames or displayed separately.
4. Blink Detection: The system accurately detects and counts the number of blinks in real-time. The blink count can be displayed as a numerical value or as an incrementing counter on the screen.
5. Statistical Analysis: The system provides statistical analysis of eye movements, including metrics such as average eye movement distance, gaze angle, or blink frequency. These results can be displayed as numericalvalues or graphical representations.
6. Real-time Display: The eye tracking results are displayed on the screen inreal-time, either overlaid on the video frames or in a separate window. The display can include information such as gaze direction, eye movement metrics, blink count, and any other relevant data.

7. User Interface: The developed user interface allows users to interact with the eyetracking system and interpret the results easily. It can include controls for startingand stopping the eye tracking process, adjusting parameters, and toggling between different visualizations.

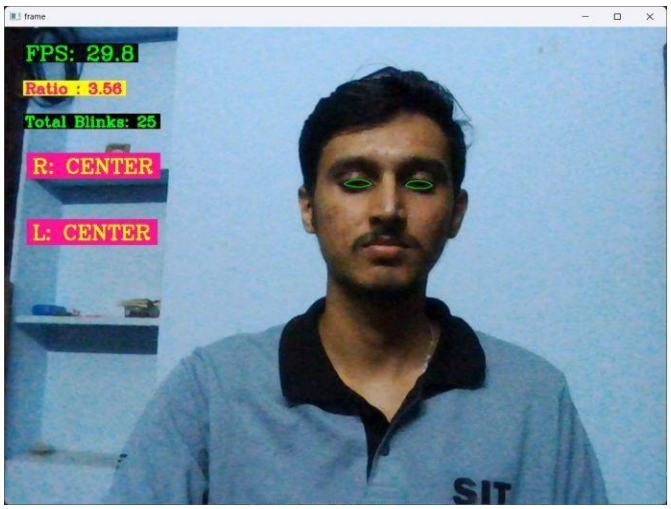

Figure 5.1 : Center Eye Movement

When the eye movement is at the center, Eye ball typically indicates that the participant's gaze is fixated on a central point of interest within their field of view.

Figure 5.2 : Right Eye Movement

The eye movement is predominantly towards the right, Eye ball indicates that the participant's gaze is directed towards the right side of the visual stimulus.

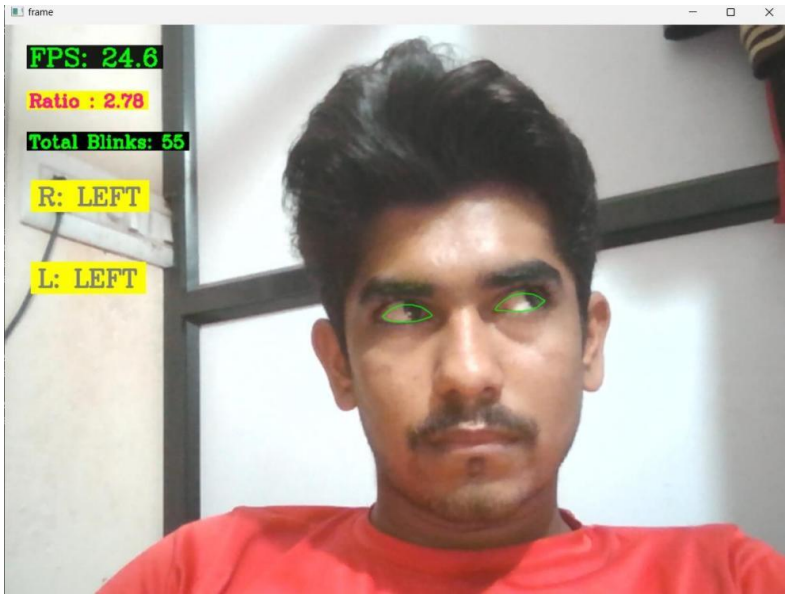

Figure 5.3 : Left Eye Movement

The eye movement is predominantly towards the left, Eye ball indicates that the participant's gaze is directed towards the left side of the visual stimulus.

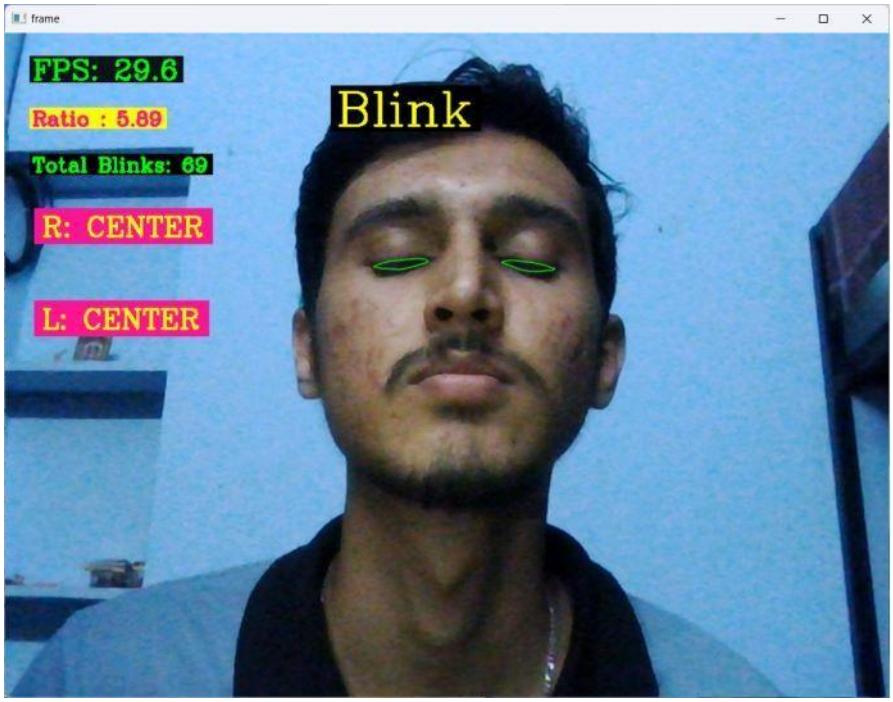

Figure 5.4 : Blink of Eye

If the eye movement is detected as blinking, it indicates that the person being tracked is closing and opening their eyelids rapidly.

CHAPTER 6
ADVANTAGES AND DISADVANTAGES

Advantages: -

1. Accurate Eye Tracking: The eye tracking system using OpenCV offers accurate tracking of eye movements, allowing for precise analysis of gazedirection, eye rotations, and blink detection.
2. Real-time Processing: The system performs eye tracking and analysis in real-time, providing instant results and feedback. This is beneficial for applications that require immediate responses, such as human-computer interaction or usability testing.
3. Non-Intrusive: The eye tracking system relies solely on video frames captured by the front camera, making it non-intrusive and comfortable forusers. It does not require any additional sensors or devices attached to the user's eyes.
4. Cost-Effective: Implementing eye tracking using OpenCV is relatively cost-effective compared to specialized eye tracking hardware or commercial solutions. It leverages existing hardware (e.g., front camera) and open-source software libraries, reducing the need for expensive equipment.
5. Flexibility and Customization: The system can be customized and adapted to specific requirements or research objectives. Developers have the flexibility to modify algorithms, adjust parameters, and integrate additionalfunctionalities based on their needs.
6. Wide Range of Applications: Eye tracking using OpenCV has a wide range of applications, including human-computer interaction, user experience research, medical diagnostics, and attention analysis. Its versatility makesit suitable for various industries and research fields.

Disadvantages: -

1. Sensitivity to Lighting Conditions: The accuracy of the eye tracking systemcan be affected by variations in lighting conditions. Poor lighting orextreme lighting conditions may lead to less accurate eye tracking results.
2. Dependency on Camera Quality: The quality of the front camera plays a crucial role in the system's performance. Lower-resolution or low-qualitycameras may result in less accurate eye tracking and affect the overall system performance.
3. Limited Tracking Range: The eye tracking system using OpenCV has limitations in tracking eye movements beyond the camera's field of view. It may not accurately capture eye movements that occur outside thecaptured video frame
4. Sensitivity to Occlusions: The system may face challenges in accurately tracking eye movements when there are occlusions, such as eyeglasses, long hair, or objects obstructing the view of the eyes. These occlusions canhinder the system's ability to detect and track the eyes effectively.
5. Calibration Requirements: For optimal performance, the eye tracking system may require initial calibration to establish a baseline and adapt to individual users. Calibration processes may add complexity and time to thesetup and usage of the system.
6. Computational Demands: Real-time eye tracking and analysis can be computationally demanding, especially on lower-powered devices. Processing high-resolution video frames at a fast frame rate may require adequate processing power and memory resources.

CHAPTER 7

CONCLUSION

This project demonstrates the feasibility and effectiveness of implementing eye tracking functionalities using open-source libraries and computer vision techniques. By leveraging the power of OpenCV, along with additional libraries like NumPy, dlib, and math, the project successfully tracks eye movements, detects blinks, and provides real-time analysis of gaze direction.

Through the implementation process, several key steps were followed, including data acquisition, preprocessing, facial landmark detection, eye segmentation, eyetracking, mathematical calculations, and real-time display. These steps, combinedwith accurate algorithms and techniques, enable the system to provide reliable results and insights into human visual attention and eye behavior.

In conclusion, the "Eye Tracking using OpenCV" project provides a valuable toolfor various applications, including human-computer interaction, usability testing,medical research, and attention analysis.

BIBLIOGRAPHY

1. Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media.
2. Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of SoftwareTools.
3. King, D. E. (2009). Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research, 10, 1755-1758.
4. NumPy Contributors. (2021). NumPy: A fundamental package for scientific computing with Python. Retrieved from https://numpy.org
5. dlib contributors. (2021). Dlib: A modern C++ toolkit containing machinelearning algorithms and tools for creating complex software. Retrieved from http://dlib.net
6. OpenCV Contributors. (2021). OpenCV: Open Source Computer Vision Library. Retrieved from https://opencv.org
7. Chollet, F., et al. (2015). Keras: The Python Deep Learning library. Retrieved from https://keras.io
8. Acharya, T., et al. (2017). Eye Movement Analysis: A Literature Survey. Journal of Eye Movement Research, 10(1), 1-28.
9. Guo, Y., et al. (2019). An Eye-Tracking System for Human Computer Interaction Using Webcam. In Proceedings of the 14th International Conference on Computer Science & Education (ICCSE), 50-54.
10. Pusiol, G., et al. (2021). An Open-Source Eye Tracking System for Interactive Learning Environments. Sensors, 21(5), 1751.