EYEPOINT – IRIS RESEARCH AND DEVELOPMENT

A PROJECT REPORT

Submitted by

ASHWIN SANJAY J (211420104030)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report "**EYEPOINT - IRIS RESEARCH AND DEVELOPMENT**" is the bonafide work of "**ASHWIN SANJAY J (211420104030)**" who carried out the project under my supervision.

Signature of HOD with Date	**Signature of Supervisor with Date**
Dr. L. JABASHEELA, M.E., Ph.D.,	**DR. N. PUGHAZENDI, M.E., Ph.D.,**
PROFESSOR AND HEAD,	**PROFESSOR,**
Department of Computer Science and Engineering,	Department of Computer Science and Engineering,
Panimalar Engineering College,	Panimalar Engineering College,
Chennai-600123	Chennai-600123

Certified that the above candidate was examined in the Project Viva-Voce Examination held on........................

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

I, **ASHWIN SANJAY J (211420104030)** hereby declare that this project report titled **EYEPOINT - IRIS RESEARCH AND DEVELOPMENT**, under the guidance of **DR. N. PUGHAZENDI M.E.,Ph.D.,** is the original work done by me and I have not plagiarized or submitted to any other degree in any university by me.

ASHWIN SANJAY J

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHIKUMAR, M.E., Ph.D.,** and **Dr. SARANYASREE SAKTHIKUMAR B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr. K. Mani, M.E., Ph.D.,** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. L. JABASHEELA, M.E., Ph.D.,** for the support extended throughout the project.

We would like to thank my Project Guide,**DR. N. PUGHAZENDI M.E.,Ph.D.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

Additionally, we would like to express our gratitude to **T-MACHINE SOFTWARE SOLUTIONS PVT. LTD.** for their collaboration, support, and providing the necessary resources that significantly contributed to the realization of this project.

PROJECT CERTIFICATE

ABSTRACT

EyePoint is an advanced eye-tracking software that revolutionizes user interaction by enabling precise mapping of a user's gaze onto a computer monitor. Utilizing cutting-edge Al algorithms, EyePoint is adept at real-time iris tracking, even with standard webcams, making it a cost-effective alternative to more expensive systems. Its adaptability to various camera setups and lighting conditions, combined with an intuitive web-based calibration tool, ensures ease of use and consistent performance across diverse user environments. The 'Fun Window' feature enhances user engagement with real-time gaze visualization, while robust data security measures safeguard user privacy. Designed for cross-platform compatibility, EyePoint's versatile applications range from enhancing user experience in interactive learning to providing insights in user behavior research. Its unique blend of precision, affordability, and user-centric design positions EyePoint as a transformative tool in the eye-tracking technology landscape, making advanced gaze tracking accessible for a wide array of applications. CNN model for iris tracking uses the latest techniques in computer vision and deep learning. This method involves finding the iris accurately through image processing, spotting the pupil efficiently, and adjusting the segmentation and normalization for different environments. It also includes extracting specific features to recognize distinct iris patterns and using accurate matching methods for precise recognition. Additionally, there are measures to prevent fake attempts, make real-time processing faster, and improve the reliability of these methods through adaptive learning.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

Gaze tracking is an innovative technology that focuses on monitoring and interpreting the direction of a person's gaze, providing valuable insights into human-computer interaction, psychology, and physiological studies. This cutting- edge technology utilizes specialized hardware and software to accurately capture and analyze eye movements, allowing for a deeper understanding of user behavior and engagement.

One of the primary applications of gaze tracking is in human-computer interaction (HCI). By precisely tracking where a user is looking on a screen, devices can adapt their interfaces in real-time, enhancing user experience and efficiency. This technology has widespread applications in fields such as virtual reality (VR), augmented reality (AR), gaming, and assistive technologies. In VR and AR environments, gaze tracking enables more natural interactions by allowing users to control elements simply by looking at them.

In psychology and neuroscience, gaze tracking serves as a valuable tool for studying cognitive processes and behavior. Researchers can gain insights into attention, perception, and decision-making by analyzing patterns of eye movement. This aids in understanding how individuals process information and make decisions, contributing to advancements in fields like cognitive science and neuromarketing.

`

In the healthcare sector, gaze tracking finds applications in diagnosing and monitoring neurological disorders. It enables early detection of conditions affecting eye movement, such as Alzheimer's disease or traumatic brain injuries. Additionally, gaze tracking is incorporated into assistive technologies to empower individuals with motor impairments, allowing them to interact with computers and devices through eye movements.

1.2 PROBLEM DEFINITION

In the rapidly advancing field of user interaction and technology, a significant challenge presents itself: the accurate mapping of a user's gaze onto a computer monitor. This task demands precise determination of the x, y pixel coordinates where the user is looking, while also adapting to a variety of camera positions and diverse hardware configurations. The complexity of this endeavor is further amplified by the varying qualities of standard cameras, which are not originally designed for the nuanced requirements of iris detection.

Difficulty in Tracking IRIS with Standard Cameras:

Standard cameras struggle with precise iris detection due to non- optimal placement and variable imaging capabilities.

These cameras are not specifically designed for iris tracking, leading to accuracy issues under different environmental conditions and user behaviors.

Cost of Advanced Technology:

Advanced systems like the Tobii are precise but costly, limiting their widespread use.

Variability in Lighting Conditions:

Difficult to ensure reliable performance under diverse lighting conditions, from dimly lit rooms to bright office environments.

User Diversity:

The system must adapt to variations in eye characteristics, maintaining accuracy and equity across different user demographics.

Integration with Existing Systems and Software:

Seamless integration with various operating systems and software applications is essential, without needing extensive modifications.

Low Latency Processing:

Real-time or near-real-time processing of eye movement data is crucial for responsive interaction.

Handling Glasses and Contact Lenses:

The system needs to track the iris accurately for users wearing glasses or contact lenses.

System Calibration:

Developing an intuitive and efficient calibration process adaptable to varied webcam setups and user environments.

Hardware Limitations:

Overcoming limitations of standard webcam hardware, such as lower resolution and frame rate, to maintain tracking accuracy.

1.2 EXISTING SYSTEM

In the architecture of the existing eye-tracking system, the integration of an eye-tracking device stands out as a critical element to ensure the system's accuracy and reliability. This dedicated hardware component employs sophisticated sensors, cameras, or infrared technology to capture intricate details of the user's eye movements. The captured images or videos undergo thorough analysis, facilitated by advanced algorithms, to precisely determine the user's gaze point with subpixel accuracy. This eye-tracking device serves as the system's lens into the user's visual interactions, providing the necessary data for comprehensive gaze tracking. The importance of this hardware becomes particularly pronounced in scenarios where fine-grained accuracy is paramount, such as in research applications, human-computer interaction studies, or immersive virtual reality experiences. Moreover, the eye-tracking device plays a pivotal role in calibration processes, adapting the system to the unique characteristics of each user's eyes. This personalized calibration ensures that the system can accurately interpret individual gaze patterns, blinks, and other nuances, contributing to an enhanced and tailored user experience. As the

The existing system continues to evolve, the foundational role of the eye-tracking device underscores its significance in achieving not only accurate gaze tracking but also unlocking the full potential of applications that rely on precise and nuanced eye movement data.

1.3 PROPOSED SYSTEM

In the proposed system, a groundbreaking approach is introduced by leveraging normal cameras to achieve high-accuracy eye tracking. Departing from the necessity of specialized eye-tracking devices, this innovative strategy harnesses the potential of standard, high-resolution cameras to capture and interpret eye movements with exceptional precision. By implementing advanced computer vision algorithms and machine learning techniques, the proposed system transforms ordinary cameras into powerful tools for gaze tracking, ensuring a seamless and accurate analysis of users' eye behaviors.

This novel approach not only simplifies the hardware requirements but also enhances the accessibility of eye-tracking technology, making it more widely applicable across various devices and scenarios. The utilization of normal cameras retains a user-friendly and non-intrusive quality, as users can engage in eye tracking without the need for dedicated, specialized equipment. The proposed system's reliance on standard cameras opens the door to a myriad of possibilities, from integrating gaze tracking into everyday applications to expanding its utility in fields such as human-computer interaction, gaming, and virtual reality. As the system continues to evolve, this pioneering utilization of normal cameras represents a paradigm shift in eye-tracking technology, offering a potent combination of accessibility, accuracy, and versatility.

CHAPTER 2

LITERATURE SURVEY

2.1 Evaluation of appearance-based eye tracking calibration data selection

Yuqing Li, Yinwei Zhan, Zhuo Yang have proposed that the calibration process is a critical aspect of gaze estimation, with person-specific methods often outperforming person-independent ones.

Classical approaches involve subjects staring at targets on a screen, recording eye images and target locations, extracting features, and determining parameters for mapping functions to compute gaze points. However, appearance-based eye tracking lacks a clear calibration method. Recent works propose innovative solutions, such as using gaze redirection networks, fine-tuning latent parameters, or employing architectures requiring minimal calibration data.

This study addresses the challenge of calibration data selection's impact on person-specific gaze estimation. The researchers trained a person-independent convolutional neural network (CNN) and employed support vector regression (SVR) as a regressor for calibration. The CNN serves as a gaze estimator, considering factors like illumination conditions, head motion, image resolution, and the number of subjects. The team chose the GazeCapture dataset for training due to its diversity in subjects and images.

In calibration, the study introduces a unique approach by using features from the fully connected (FC) layer of the CNN to train an SVR regressor. The calibration data distribution is evaluated using two types: rectangle and ellipse. The results indicate that increasing the number of calibration targets, despite being few, reduces gaze estimation error. However, there is a limitation to the error reduction with a further increase in the number of calibration targets.

The impact of calibration target shape on estimation error is also explored. Results show differences in performance for rectangle and ellipse distributions, with the rectangle distribution achieving the best performance with a 6.9-centimeter error.

This research highlights the importance of careful calibration data selection in appearance-based eye tracking. Despite using a person-independent CNN, the study demonstrates that a thoughtful calibration strategy, even with a minimal number of targets, significantly improves gaze estimation accuracy,2020.[11]

2.2 Eye Movement Classification Using CNN

Face Detection and Eye Detection in Video:

Milu Prince, Neha Santhosh, Nimitha Thankachan, Reshma Sudarsan Ms.Anjusree V.K has proposed dlib library and OpenCV for face and eye detection in videos.. The frontal face detector from dlib is utilized, along with facial landmarks for eye detection. The facial landmark file "shape predictor 68 face landmarks.dat" is employed. The video is processed frame-wise, converting each frame to grayscale for face detection. The facial landmark function is then used to detect the eyes. The eyes' locations in the facial landmark file are specified as points, facilitating eye extraction.

Dataset Creation and Pre-processing:

The Eye Chimera dataset is utilized, containing images of faces with different eye movements. After detecting eyes, they are cropped separately and resized to 50x50 pixels. Due to insufficient images for each eye movement type, left and right eyes are combined, doubling the dataset. Data augmentation includes image flipping to increase dataset variety. Grayscale conversion and histogram equalization enhance training efficiency and contrast. For blink detection, open and closed eye datasets are created, incorporating manual and CEW dataset images. Resizing and histogram equalization are applied to these datasets.

Training of the Model:

The dataset is split into 80% training and the rest for validation and testing. The CNN comprises three convolution stages, each followed by ReLu and max pooling layers. Fully connected layers join the outputs, and a sigmoid activation function classifies open and closed eyes.

This model handles three classes of eye movements: left, right, and center. The dataset consists of 536 preprocessed images. The CNN includes three convolution stages with ReLu and max pooling, followed by two additional similar stages. Fully connected layers join the outputs, and a softmax function in the final layer predicts the class with the highest score.

This system architecture involves face and eye detection using dlib and OpenCV, dataset creation and pre-processing from the Eye Chimera dataset, and training . The models utilize convolutional neural networks (CNNs) with specific architectures for each task,2020.[12]

2.3 Classification of Eye Tracking Data using a Convolutional Neural Network

Yuehan Yin, Chunghao Juan, Joyram Chakraborty, Michael P. McGuire stated the system that classifying users#39; eye movements during short intervals by converting records into a 2D array based on screen resolution.. Each array represents 10 seconds of eye tracking data, with JSON format chosen for storage. The JSON representation includes fields like data_id, gp_image (2D array of gaze points), gp_sequence (timestamp sequence and gaze point coordinates), interface, nationality, participant_name, timestamp_begin, and timestamp_end.

For database setup, a single JSON file with multiple JSON documents is generated using Python and stored in MongoDB. Indexes on the data_id field facilitate data retrieval. Feature engineering involves using maximum filters and min-max scaling on the gp_image field's 2D arrays. Another approach connects gaze points based on timestamp sequence to generate scan paths, enhancing feature recognition. The research emphasizes enhancing machine learning models by preprocessing data.

The choice of the deep-learning framework and methodology revolves around the adoption of Keras, primarily due to its seamless integration with TensorFlow, a widely-used open-source machine learning library. Keras is favored for its user-friendly interface and abstraction, making it more accessible, especially for those who are new to deep learning.

To prepare the data for model training and evaluation, the 70/30 holdout method is employed for data partitioning. This method involves splitting the dataset into two subsets: a training set, which constitutes 70% of the data, and a testing set, which comprises the remaining 30%. This partitioning strategy is pivotal for assessing the model's generalization performance on unseen data.

The study focuses on two distinct classification tasks: the classification of web user interfaces and the classification of nationalities. To handle these tasks, evaluation metrics play a crucial role in quantifying the model's performance. The metrics employed include accuracy, recall, specificity, precision, and negative predictive value. These metrics offer a comprehensive understanding of the model's effectiveness in correctly identifying instances belonging to different classes.

In the context of deep learning, the study employs a confusion matrix, which is a valuable tool for visualizing the performance of a classifier. This matrix provides a breakdown of true positives, true negatives, false positives, and false negatives. It is particularly useful in scenarios where class imbalances exist, as it allows for a more nuanced assessment of the model's strengths and weaknesses.

Furthermore, the Keras framework facilitates the development and training of models on central processing units (CPUs), offering computational efficiency. The study delves into two classification tasks characterized by unbalanced classes. To handle this, one-hot encoding is applied to the class labels, ensuring that the model can effectively learn and differentiate between the distinct classes, even in the presence of imbalances.

The study adopts a comprehensive approach to model evaluation, considering various metrics and visualization tools. This approach is especially relevant when dealing with real-world scenarios where datasets may exhibit class imbalances, and a nuanced evaluation is necessary for robust model performance assessment.Feature engineering involves applying maximum filters and min-max scaling to enhance the visibility of gaze points and scan paths.

This approach aims to improve model accuracy by providing more informative features than raw data alone.

Overall, The study involves the preprocessing of eye tracking data, storage in MongoDB, and feature engineering to enhance machine learning model performance. The use of Keras facilitates model development, and evaluation metrics provide a comprehensive assessment of classifier effectiveness. The research emphasizes the importance of feature engineering in improving model accuracy, especially for tasks involving unbalanced classes,2020.[13]

2.4 Eye Tracking System Using Convolutional Neural Network

Shatha Iskandar , Zahraa Jaara , Mohamad Abou Ali, Fatima Sbeity , Abdallah Kasem and Lara Hamawy has proposed the system that aims to detect iris movement by creating a contour around the eye using a convolutional neural network (CNN) model.. Mean squared error (MSE) is chosen as the evaluation metric due to the continuous nature of the variables, representing pixel coordinates of the iris in image

The system's approach involves leveraging a Convolutional Neural Network (CNN) for the classification of predicted iris points based on training data. The process initiates with the tracing of iris contours using the "VGG Image Annotator," which captures fifteen points in both x and y coordinates. These annotated points serve as the foundation for training the CNN model to predict iris movements.

To implement the CNN model, Google Colaboratory, a product of Google Research, is chosen. The model is trained using nearly accurate points obtained from the annotation process. This training equips the model to predict the location of the iris in different images, enabling the detection of movements in various directions.

The dataset preparation phase is meticulous, starting with the loading of 500 images. The dataset is then split into training and testing sets using an 80-20 ratio, facilitating the model's development and subsequent validation. The dlib library proves instrumental in extracting the eye area, resulting in images of dimensions 77 by 55 pixels. To enhance computational efficiency, the images are further resized to 416 by 416 pixels using the OpenCV library.

A critical aspect of dataset preparation involves creating a dataset with experimentally obtained equations. This dataset is specifically tailored for iris identification and serves as pre-training data for the CNN model. While acknowledging the availability of a larger dataset for potentially better results, computational constraints necessitate the use of a dataset comprising 500 images.

The construction of the CNN model is achieved through the Keras library, employing sequential layers. The architecture incorporates six convolution layers, five max-pooling layers, flattening layers, three dense layers, and two dropout layers. The design prioritizes the detection of spatial features through kernels, with Rectified Linear Unit (ReLU) activation functions introducing non-linearity. The number of output filters for each convolution layer is set at 32, and the kernel size is defined as (3,3).

The compiled model utilizes an Adam optimizer with a learning rate of 0.0001 for efficient training. The mean squared error (MSE) is chosen as the loss function, emphasizing the minimization of the difference between actual and predicted points. The model undergoes training with 250 epochs.

This systematic approach demonstrates a comprehensive integration of tools and techniques, emphasizing the significance of accurate dataset preparation and the utilization of a well-structured CNN model for iris movement detection

Before training, the model is compiled with an Adam optimizer, mean squared error loss function, and accuracy metric. The training involves 250 epochs. The mean squared error graph demonstrates a significant drop from around 50,000 to approximately 280, indicating a reduction in the difference between actual and predicted points.

Accuracy, though not as significant as MSE in this context, hovers around 50%, considering the continuous nature of the variables. The fluctuation in training and validation accuracy is attributed to the limited and non-diverse training data, leading to occasional incorrect predictions.

Images from the tested dataset display the lining placed during training and the lining predicted by the model. The difference between the two linings is relatively small, showcasing the system's capability to predict iris movements accurately.

The proposed system demonstrates effective iris movement detection using a CNN model, with MSE as a suitable evaluation metric for continuous variables. Despite some fluctuation in accuracy, the model performs well in predicting iris contours in images from the tested dataset,2023.[14]

2.5 Eye Tracking Analysis Using Convolutional Neural Network

Narayana Darapaneni, Meghana D Prakash, Bibek Sau, Meghasyam Madineni, Rahul Jangwan, Anwesh Reddy Paduri, Jairajan K P, Mugdha Belsare,PradeepMadhavankutty proposed the study employs various algorithms for gaze tracking, ranging from classic edge detection and ellipse fitting to modern deep learning techniques like Convolutional Neural Networks (CNN), Deep Neural Networks (DNN), and OpenCV. Opencv, an open-source computer vision and machine learning software library, serves as a common infrastructure for computer vision applications. Specifically, DNN is utilized for classifying images as valid or invalid, coupled with OpenCV for preprocessing tasks such as edge detection, ellipse fitting, rotations, blur, exposure adjustments, reflection, and compression.

CNNs are chosen due to their high learning capacity, particularly in tasks involving large datasets, making them suitable for eye tracking challenges. They are employed to extract key features from images in the Gazecapture dataset, focusing on details about the eyelid, pupil, and eye area. The decision to use CNN is influenced by its consistent positive results in image classification tasks over time.

For the implementation phase, the Gaze Capture dataset (2016) is selected, which is crowdsourced through an iPad app. The dataset ensures gaze accuracy by incorporating a mechanism to verify that individuals are looking in the correct direction. The dataset comprises a vast number of frames, and approximately 50,000 images are chosen for training due to its size. Data preprocessing involves using frames with detected faces and eyes, and Euclidean distance is employed as the loss function during training.

The study includes exploratory data analysis (EDA) on face, face grid, and left and right eye detection. The images extracted from the dataset undergo transformations such as blurring, contouring, edge detection, embossing, and smoothing. Two separate models are tested, one for accuracy in detecting left and right eyes and another for predicting gaze points. The results show high accuracy in eye detection, with training accuracy at 97% and testing accuracy at 98%. Analysis indicates slight overfitting in the model, suggesting room for improvement, potentially through transfer learning for gaze points.

The final model is designed to determine the accuracy of gaze points using CNN, expressed in Euclidean distance between actual and predicted gaze points. The study emphasizes the recent success of CNNs in computer vision, especially in gaze estimation models. The research aims to contribute valuable insights to enhance eye-tracking performance using deep learning techniques, hoping to benefit researchers across various fields interested in leveraging gaze as a cue for understanding human behavior,2022.[10]

CHAPTER 3

THEORETICAL BACKGROUND

3.1 IMPLEMENTATION ENVIRONMENT

3.1.1 HARDWARE REQUIREMENT

For an eye-tracking website, detailing the hardware requirements is crucial for users and developers to understand the necessary equipment for optimal performance.

Camera:

High-resolution webcam or specialized eye-tracking camera. Minimum 720p for standard use; higher resolution for more precision. 30 frames per second (fps) or higher for real-time tracking.

Computer System:

Quad-core processor (or equivalent) for real-time processing. 6GB or higher for smooth operation. Dedicated GPU for better performance, especially in 3D applications.

3.1.2 SOFTWARE REQUIREMENT

Operating System:

Windows, macOS, Linux: Depending on the website's compatibility.

Web Browser:

Support for modern browsers like Google Chrome, Mozilla Firefox, or Safari.

Internet Connection:

High-speed internet for seamless data transmission. Minimum of 5 Mbps speed for smooth interaction.

3.2 SYSTEM ARCHITECTURE

System architecture refers to the conceptual structure and high-level design of a software system or application. It defines the arrangement and interaction of the system's components to fulfill specific requirements, such as functionality, performance, scalability, and maintainability.

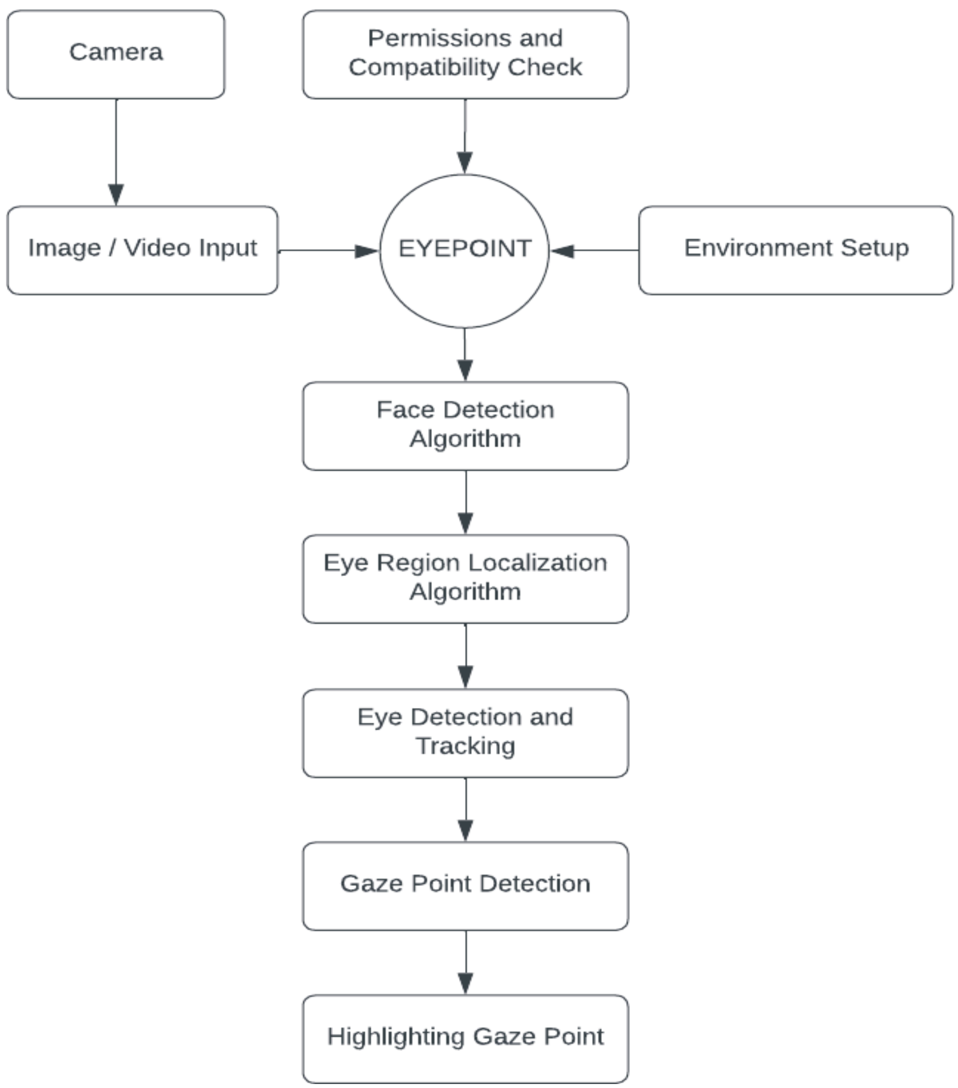


Fig 3.1 Architecture diagram for Gaze Tracking

The image depicts a system architecture for face detection and gaze point detection. Here's a breakdown of the components and their interactions:
Components:

Camera:

This captures images or video streams of the system.

Permissions and Compatibility Check:

This component ensures the system has permission to access the camera and verifies compatibility between the camera and system.

Image/Video Input:

This block represents the raw image or video data coming from the camera.

Eyepoint:

`This component is responsible for eye detection and gaze point tracking within the images or video frames. It likely performs the following functions:

Eye Region Localization Algorithm:

Identifies the area of the eye within the image or frame.

Eye Detection and Tracking:

Detects the presence of eyes and tracks their movement over consecutive frames.

Gaze Point Detection:

Estimates the user's gaze direction based on the eye position within the image or frame.

Environment Setup:

This component might handle calibrating the system for optimal performance in the specific environment where it's used.

Highlighting Gaze Point:

This functionality visually indicates the estimated gaze location on the image or video frame, likely for visualization or debugging purposes.

3.3 System Design

3.3.1 Database Architecture

Table 3.1: Table of User

COLUMN NAME	DATA TYPE	CONSTRAINT
user_id	int	Primary Key
username	varchar	Not null
password	varchar	Not null
email	varchar	Not null
registration_date	date	Not null

Table 3.2 : Table of Calibration

COLUMN NAME	DATA TYPE	CONSTRAINT
caliberation_id	int	Primary Key
screen_width	int	Not null
screen_height	int	Not null
user_id	int	Foreign key(User)
caliberation_timestamp	timestamp	Not null
caliberation_point_x	int	Not null
caliberation_point_y	int	Not null
user_response_x	int	Not null
user_response_y	int	Not null
caliberation_accuracy	float	Not null

Table 3.3 Table of Gaze Data

COLUMN NAME	DATA TYPE	CONSTRAINT
gaze_id	int	Primary Key
user_id	int	Foreign key(User)
frame_no	int	Foreign key(Frames)
gaze_timestamp	timestamp	Not null
gaze_point_x	int	Not null
gaze_point_y	int	Not null

Table 3.3: Table of Frames

COLUMN NAME	DATA TYPE	CONSTRAINT
frame_no	int	Primary Key
user_id	int	Foreign key(User)
image_data	blob	Not null
timestamp	timestamp	Not null

3.3.2 Module Diagram

Use Case Diagram:

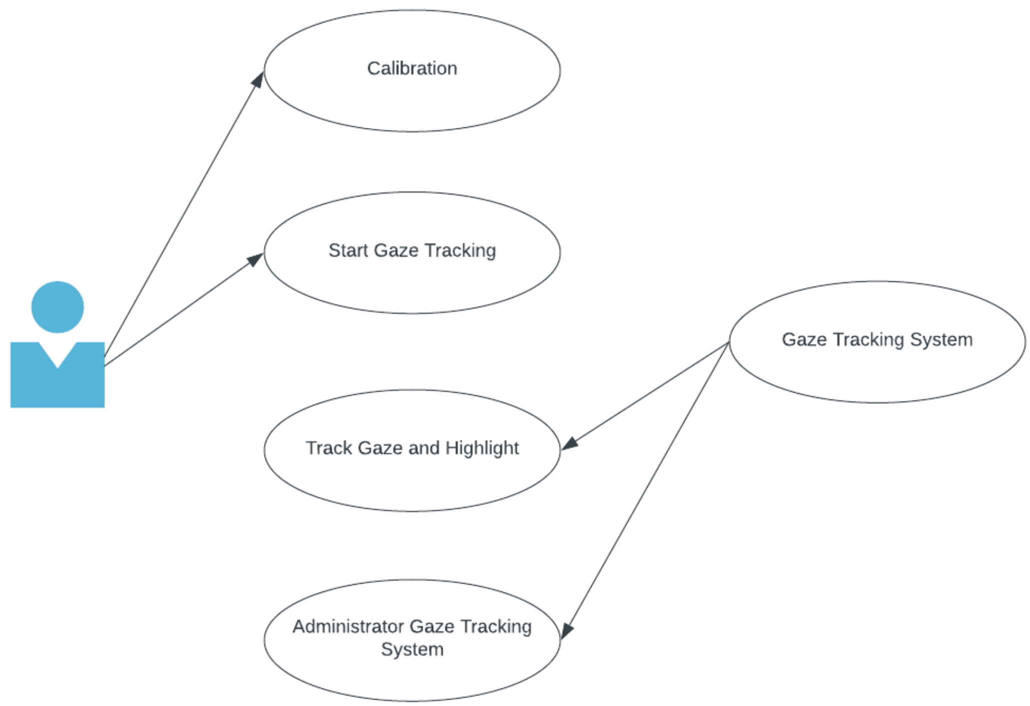


Fig 3.2: Use Case Diagram for Gaze Tracking

The use case diagram depicts a gaze tracking system. Let's break down the interactions between the actors and the system

Administrator:

The actor mentioned in the scenario is responsible for initiating, terminating, and calibrating the gaze tracking system. This actor serves as an interface between the system itself and the administrator, facilitating the management and control of the gaze tracking functionality.

The Gaze Tracking System, on the other hand, is the core component of the system. It interacts with both the administrator and itself, offering a range of functionalities. These include starting and stopping gaze tracking sessions, performing calibration procedures to ensure accurate tracking, and actively monitoring and highlighting the user's gaze.

The use case diagram provides a high-level overview of the interactions between the Gaze Tracking System and the administrator. It emphasizes the key actions and responsibilities of both parties without delving into the specifics of how users interact with the system to have their gaze tracked or highlighted.

While the actor manages the overall operation of the gaze tracking system, the system itself orchestrates the various functions required for accurate gaze tracking, calibration, and user interaction. The use case diagram serves as a visual representation of these interactions at a high level, providing a conceptual understanding of the system's functionality and its interaction with the administrator.

Sequence Diagram :

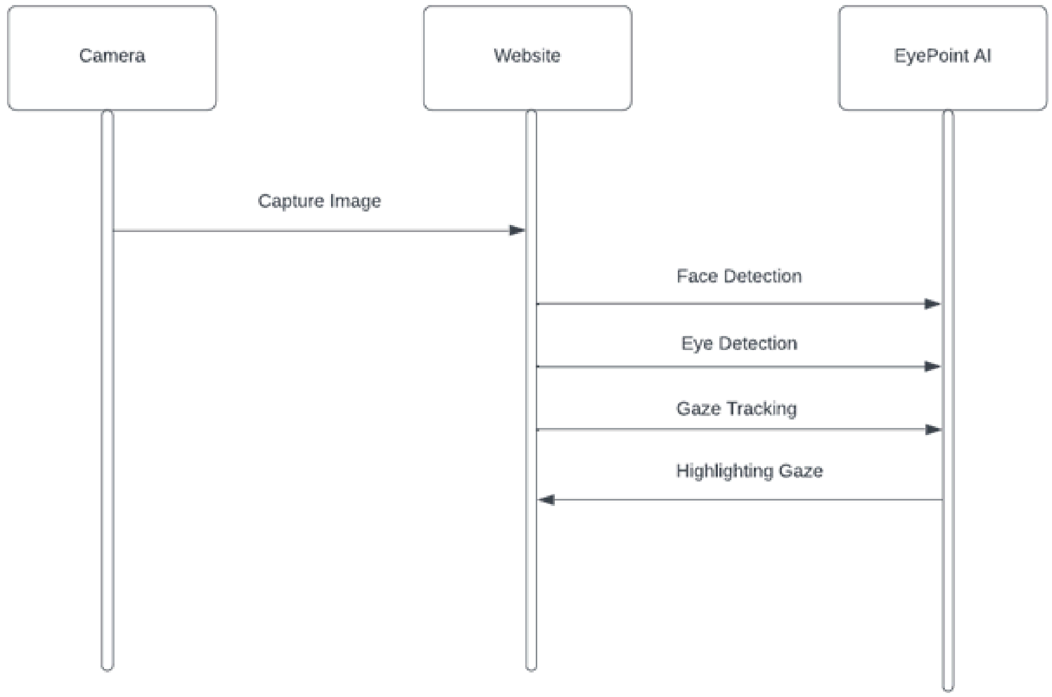


Fig 3.3: Sequence Diagram For Gaze Tracking

The sequence diagram depicts the process of taking a picture using a camera and sending it to a website with gaze tracking functionality. Here's a breakdown of the interactions between the objects.

Camera:

This lifeline represents the digital camera that captures the image.

Website:

This lifeline represents the website that will receive and display the captured image, along with its gaze tracking functionalities.

EyePoint Al (assumed):

This lifeline likely represents an eye-tracking software or library used by the website to analyze gaze data. It might not directly interact with the camera in this scenario.

The sequence of messages flows as follows:

Capture Image:

The diagram initiates with the camera capturing an image, likely triggered by a user pressing a button on the camera or the website sending a capture command (depending on the specific setup).

Send Image to Website:

The camera sends the captured image data to the website.

Website Receives Image:

The website receives the image data from the camera.

Face Detection:

The website performs face detection on the received image to identify the presence and location of a face within the picture.

Eye Detection:

If a face is detected, the website proceeds to detect the eyes within the face region of the image.

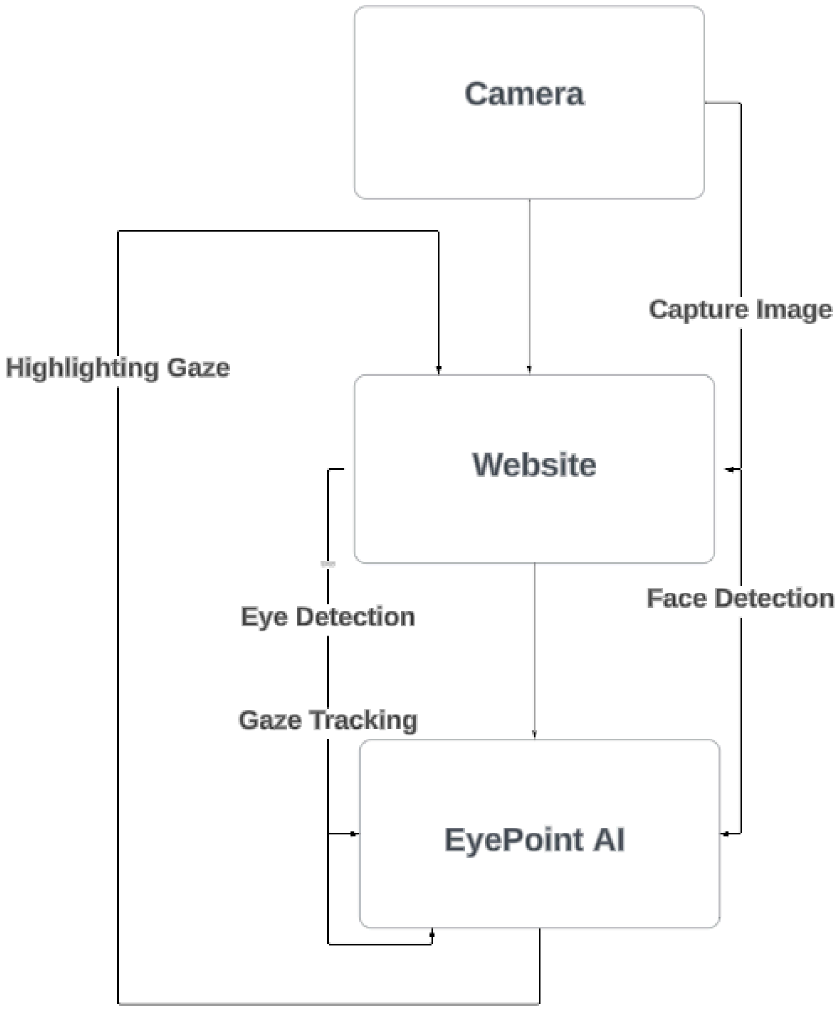
Gaze Tracking:

The depiction of gaze tracking here might be simplified. It likely refers to the website using EyePoint AI to analyze the eye positions within the image and estimate the user's gaze direction at the time the picture was captured. This information might be used for website analytics purposes.

Highlight Gaze:

This highlights the estimated gaze location on the image for visualization purposes.

Collaboration Diagram:



Camera:

This lifeline represents the digital camera that captures the image.

Website:

This lifeline represents the website that will receive and display the captured image, along with its gaze tracking functionalities.

EyePoint Al (assumed):

This lifeline likely represents an eye-tracking software or library used by the website to analyze gaze data. It might not directly interact with the camera in this scenario.

The sequence of messages flows as follows:

Capture Image:

The diagram initiates with the camera capturing an image, likely triggered by a user pressing a button on the camera or the website sending a capture command (depending on the specific setup).

Send Image to Website:

The camera sends the captured image data to the website.

Website Receives Image:

The website receives the image data from the camera.

Face Detection:

The website performs face detection on the received image to identify the presence and location of a face within the picture.

Eye Detection:

If a face is detected, the website proceeds to detect the eyes within the face region of the image.

Gaze Tracking:

The depiction of gaze tracking here might be simplified. It likely refers to the website using EyePoint AI to analyze the eye positions within the image and estimate the user's gaze direction at the time the picture was captured. This information might be used for website analytics purposes.

Highlight Gaze:

This highlights the estimated gaze location on the image for visualization purposes.

Activity Diagram :

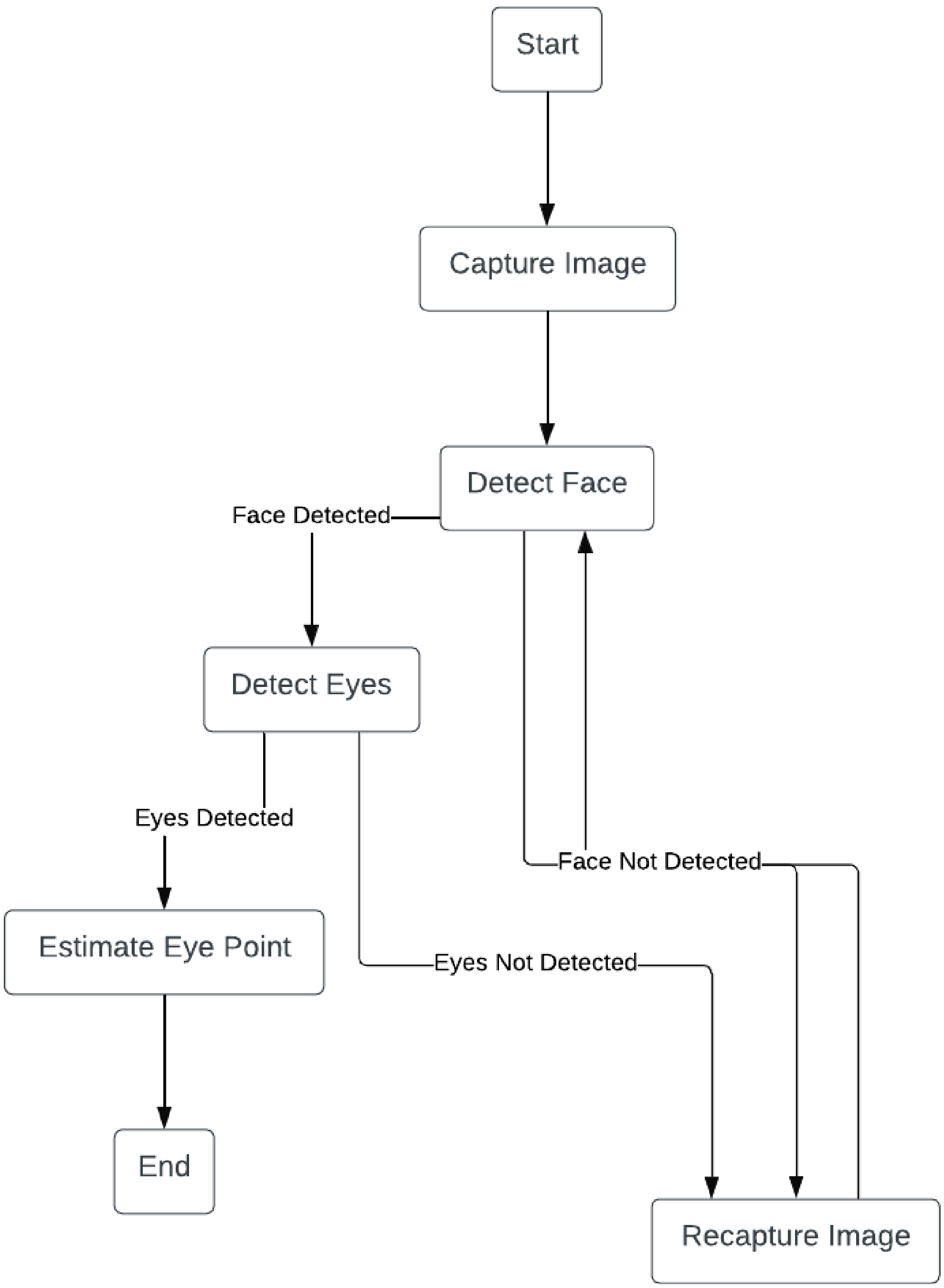


Fig 3.4: Activity Diagram For Gaze Tracking

The activity diagram depicts the steps involved in a face detection system. Here's a breakdown of the activities and their flow:

Start: The process begins here.

Capture Image: The system captures an image, likely using a camera.

Face Detected ?: A decision diamond checks if a face is detected in the captured image.

Yes: The process continues to the "Detect Eyes" activity.

No: If no face is detected, the process ends (represented by the "End" terminal symbol).

Detect Eyes: The system attempts to detect the presence of eyes within the captured image.

Eyes Detected?: Another decision diamond checks the outcome of eye detection.

Yes: The process proceeds to the "Estimate Eye Point" activity.

No: If no eyes are detected, the system ends (represented by the "End" terminal symbol).

Estimate Eye Point: The system estimates the location of the eyes within the image. This might involve calculating the center of each eye or other relevant data points.

End: The process reaches a successful conclusion.

The activity diagram outlines the conditional flow of a face detection system. It starts by capturing an image and checks if a face is present. The process terminates if no face or eyes are found in the captured image.

Class Diagram :

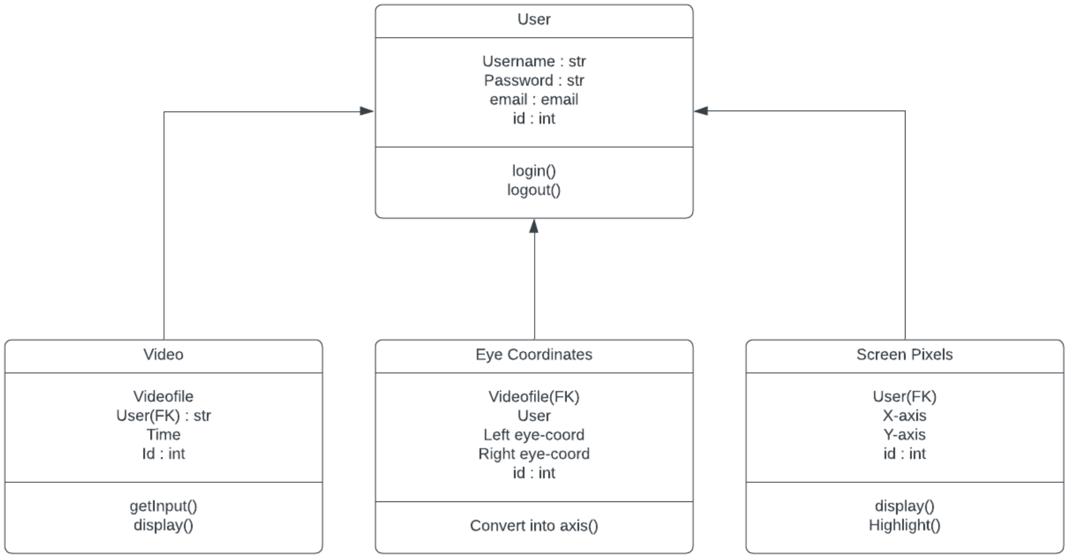


Fig 3.5: Class Diagram For Gaze Tracking

The class diagram depicts a system with several classes involved in eye tracking, likely using appearance-based methods with CNNs. Here's a breakdown of the classes and their relationships:

Classes:

User:

This class likely represents the user whose eye movements are being tracked. It might hold attributes like username, password, email, and ID for user identification. It also has methods for login and logout, possibly related to user authentication for the system.

Video File:

This class represents a video file that might be used to capture eye movements. It has attributes like filename and potentially a foreign key referencing the User class, indicating which user the video file is associated with.

Eye Coordinates:

This class likely stores the coordinates of the user's eyes within an image or frame. It might have attributes for left eye X and Y coordinates, right eye X and Y coordinates, and an ID for the data point.

Screen Pixels:

This class might represent the dimensions of the screen where the eye movements are being tracked. It could have attributes for width and height in pixels and an ID for the data point.

Time:

This class likely stores timestamps associated with the eye tracking data. It might have an attribute for the timestamp itself and an ID for the data point.

Relationships:

User:

Has a one-to-many relationship with VideoFile: One user can have many video files associated with them. This is likely represented by the foreign key in the VideoFile class referencing the User class.

Video File:

Can have one-to-many relationships with both EyeCoordinates and Time. A video file can have many timestamps and eye coordinate data points associated with it.

Data Flow Diagram :

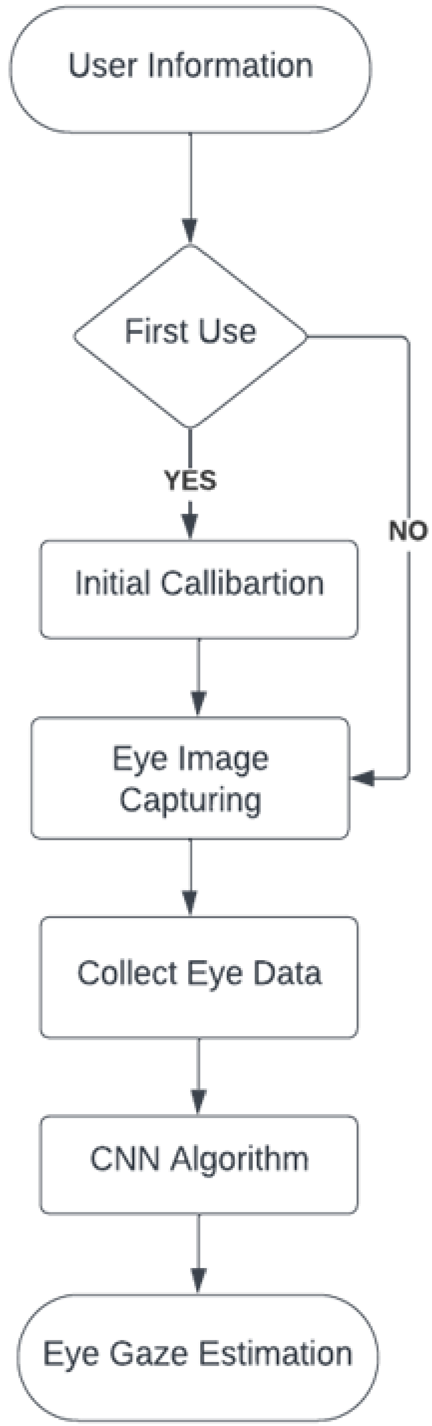


Fig 3.6: Data Flow Diagram For Gaze Tracking

The Flow Diagram depicts the steps involved in a Convolutional Neural Network (CNN) algorithm for eye gaze estimation. Here's a breakdown of the actions and decisions involved:

Start:

The process begins here.

User Information:

This section likely refers to information collected about the user, potentially including calibration data gathered during a setup phase.

First Use?:

A decision diamond checks if this is the first time the user is using the system.

Yes: The process moves to the "Initial Calibration" step.

No: The system skips calibration and proceeds to eye data collection, assuming calibration data is already available for this user.

Initial Calibration (if first use):

This step captures eye image data, likely through a camera or specialized eye tracking hardware. It might involve multiple captures to collect sufficient data for calibration.

Collect Eye Data:

This action continuously captures eye image data for gaze estimation.

CNN Algorithm:

The collected eye image data is then fed into the CNN algorithm.

Eye Gaze Estimation:

The CNN algorithm processes the eye image data and estimates the user's gaze direction.

Output:

The estimated gaze direction is the final output of the process. This information can be used for various applications, such as controlling an interface through eye movements or analyzing visual attention patterns.

The flow chart doesn't show details about how the CNN is trained. Training likely involves a separate dataset of eye images paired with corresponding gaze directions.

The initial calibration step might involve guiding the user to look at specific points on the screen to establish a reference for accurate gaze estimation

CHAPTER 4

SYSTEM IMPLEMENTATION

4 .1 MODULE DESCRIPTION

4.1.1 AI ENGINE

The AI Engine serves as the driving force behind sophisticated eye-tracking software, deciphering intricate eye movement data. This robust engine facilitates meticulous analysis, empowering the software to comprehend and respond to users' gaze behavior with exceptional precision. By leveraging advanced artificial intelligence, the system interprets nuanced ocular interactions, providing a seamless and accurate interface for a wide array of applications, from enhancing user experiences in virtual reality environments to revolutionizing accessibility features in human-computer interactions.

4.1.1.1 Eye Tracker Processing

The Eye Tracker Processing system seamlessly integrates state-of-the-art algorithms designed to capture and interpret eye movements in real-time. Leveraging advanced computer vision techniques, this sophisticated technology excels at accurately analyzing gaze data. The system's ability to process eye movements in real-time opens the door to a myriad of applications, with gaze-based navigation standing out prominently.

Key Features

- **Gaze Point Detection**
  - Gaze point detection is an advanced process that accurately pinpoints the exact location where a user is directing their visual attention, either

on a digital screen or within a physical environment. Employing specialized technologies such as eye-tracking, it enables the identification of the specific point of focus with high precision. This technology is crucial for various applications, ranging from human-computer interaction in user interfaces to scientific research in understanding visual behavior. It enhances the user experience by allowing systems to respond to gaze cues, providing a nuanced understanding of user engagement and preferences**.**

- **Precise Gaze Coordinates:** Our system guarantees the precise determination of the user's gaze point, achieving sub-pixel accuracy to provide highly detailed and accurate gaze coordinates. This level of precision enhances the reliability of gaze tracking, allowing for fine-grained analysis and interaction in applications ranging from user interfaces to research environments.

- **Precision:** Precision is a key attribute of our system, significantly improving the dependability of interactions. This heightened precision is particularly valuable in applications where detailed gaze tracking is essential, ensuring accurate and nuanced user engagement.

▪ **Real-time Eye Movement Analysis**

- Our real-time eye movement analysis offers immediate insights into user gaze behavior, facilitating quick adaptations in dynamic interactive environments. This capability allows for swift adjustments and optimizations, enhancing the overall responsiveness and user experience in real-time scenarios.

- **Classification of Eye Movements:** Our system employs a sophisticated classification of eye movements, differentiating between rapid, slow, and smooth pursuit movements. This classification offers nuanced insights into user visual behavior, allowing for a detailed understanding of how users engage with content and interfaces.

Continuous Tracking: Our system excels in continuous, real-time tracking of eye movements, guaranteeing responsive and seamless monitoring in dynamic interactive experiences. This capability allows for the uninterrupted observation of users' gaze behavior, ensuring precise and timely adjustments to optimize the interactive environment. With this feature, our system provides a reliable foundation for applications demanding continuous and instantaneous eye movement analysis, such as virtual reality interfaces and interactive simulations.

- **Pupil and Blink Analysis:**
  - Our system meticulously analyzes changes in pupil behavior and effectively manages blinks, ensuring accurate tracking of your gaze location. By closely monitoring these subtle cues, the system gains a deeper understanding of your focus and engagement, providing more precise insights into your visual interaction. This attention to details enhances the overall accuracy and reliability of gaze tracking, contributing to a more nuanced comprehension of user behavior.
  - **Detection and handling blinks:** Through advanced algorithms, our system achieves continuous and uninterrupted gaze tracking by effectively detecting and managing blinks. This sophisticated approach enhances the overall accuracy of gaze tracking, ensuring a seamless and reliable monitoring of the user's visual attention. By incorporating blink handling mechanisms, our system maintains precision even in

dynamic scenarios. Measuring how your pupils react and detecting blinks helps us better understand how engaged and focused you are.

4.1.1.2 Calibrator

A calibrator plays a pivotal role in ensuring the precise alignment of eye- tracking systems. By guiding users through a sequence of specific gaze points, it meticulously optimizes the calibration process. This calibration is crucial for achieving and maintaining accuracy in eye-tracking data, as it tailors the system to the unique characteristics of each user's gaze behavior. The calibrator enhances the reliability of eye-tracking applications by fine-tuning the system to the individual's eyes, ensuring that the captured data is aligned with their true gaze, leading to improved precision and overall performance in various applications, from research studies to interactive user interfaces and accessibility features.

Key Features

- **Collect Eye Movement Data**
  - We collect data on the movement of your eyes to gain insights into your points of focus and attention. This information allows us to understand and analyze the areas or objects that capture your visual interest, contributing to a more comprehensive understanding of your engagement and preferences..

  Calibration check: During the data collection process, we prioritize accuracy in our tracking system by continuously monitoring and adjusting it in real-time based on the dynamic movements of your eyes. This meticulous approach ensures that the gathered data reflects the most precise and up-to-date information, contributing to reliable insights into your visual behavior and attention.

- **Distance Estimation**
    - Our system determines your distance from the screen, allowing it to grasp your perspective and tailor the gaze analysis accordingly. This capability enhances the accuracy of gaze tracking by accounting for variations in viewing distances, ensuring a more personalized and effective user experience.

    - **Dynamic Adjustment:** Utilizing dynamic adjustment, our system automatically adapts based on your distance from the screen, enhancing the user experience for optimal comfort. This responsive feature ensures that the system continuously optimizes its settings to accommodate variations in viewing distances, contributing to a seamless and user- friendly interaction.

- **Automatic Calibration Trigger**
    - Incorporating an automatic calibration trigger, our system intuitively recognizes when it's necessary to assess and fine-tune its settings. This feature ensures that the system proactively manages calibration, maintaining optimal performance and accuracy over time without requiring manual intervention.
    - **Timely Recalibration:** This functionality ensures the system consistently operates at its best by automatically recalibrating when necessary, optimizing performance and maintaining accuracy over

time. This proactive approach minimizes the need for manual intervention, providing users with a seamless and reliable experience.

4.1.1.3 Pixel Locator

- The pixel locator functions as a digital guide within an eye-tracking system, serving the crucial role of identifying specific points on the screen. This technology is instrumental in ensuring that the system comprehends the spatial location of various elements, thereby enhancing overall accuracy for precise interactions. the pixel locator identifies and maps specific points or coordinates on the screen that correspond to the user's gaze. This information is then utilized by the system to accurately determine where the user is looking at any given moment. By providing a detailed and real-time mapping of screen elements, the pixel locator contributes to a more refined and precise interaction experience.

Key Features

- **Pixel Locator**
  - Pixel tracking within our system achieves remarkable accuracy, precisely identifying the locations of elements on your screen with exceptional precision. This high-resolution capability ensures that the system can pinpoint and track visual elements down to the pixel level, contributing to a superior level of precision in gaze tracking.
- **Sub-pixel Accuracy:**

  The system goes beyond standard accuracy by offering a super sharp focus, enabling it to zoom in with sub-pixel accuracy. This advanced feature enhances the precision of gaze tracking, allowing for incredibly detailed interactions and ensuring a level of reliability that exceeds conventional tracking methods.

- **Dynamic Pixel Mapping**
  - **Real - time Updates:** Our system dynamically maps pixel positions in real-time as you shift your gaze, offering continuous updates on your points of focus. This dynamic mapping ensures that the system adapts seamlessly to your visual exploration, providing accurate and timely information about the elements you are concentrating on.
  - **Adaptive Pixel tracking:** Adapting to alterations in the surroundings, our system ensures consistent and accurate pixel tracking, even in the face of environmental changes. This adaptive capability allows the system to maintain reliable gaze tracking by dynamically responding to shifts in the visual context, contributing to a robust and adaptable user experience.

Coordinate Mapping

- **Accurate Mapping:** Our system meticulously maps the locations of elements on the screen, providing precise coordinates that pinpoint the exact positions of visual elements. This accurate mapping allows for detailed insights into the specific locations of objects, enhancing the system's overall capability in gaze tracking and interaction analysis.
- **Synchronized Tracking:** By integrating the detailed mapping of on-screen elements with information about your gaze, our system achieves perfect synchronization in tracking. This holistic approach ensures that the system's gaze tracking aligns the mapped coordinates, resulting in precise and synchronized insights into your visual interactions.

4.1.2 WEB APPLICATION

Our web application is seamlessly integrated with eye-tracking software, boasting a user-friendly interface that facilitates effortless access to eye-tracking features. This intuitive design ensures a smooth and enjoyable experience for users, allowing them to harness the full potential of eye-tracking functionalities without unnecessary complexities. Whether for research, gaming, or accessibility purposes, our application aims to enhance user interaction by providing a cohesive platform that maximizes the benefits of eye-tracking technology, making it accessible and user-friendly for a diverse audience.

4.1.2.1 Sys – Info Collector

- The Sys-Info Collector is a vital component that gathers crucial system information, guaranteeing optimal performance and compatibility with the eye-tracking system. By collecting key data, it enables seamless integration, ensuring the eye-tracking system functions at its best by adapting to the unique specifications of the user's system for a superior and tailored experience.

Key Features

- **Camera Permissions**
  - **User Prompts:** Our system ensures a seamless and user-friendly experience when prompting users for camera access. We strategically time these prompts to occur only when necessary, ensuring they are not intrusive. Clear and concise messaging explains why camera access is needed, reassuring users about the purpose. The visual design of the

prompts is consistent with our app's aesthetic, making them easy to notice and engage with. Feedback is provided promptly after the user grants or denies access, ensuring a transparent and informative process.

- **Permission Handling:** To ensure smooth handling of permission requests, we have designed our web application to make it easy for users to grant camera access. We provide clear and concise explanations about

  why camera access is needed and how it will enhance their experience. Our user-friendly interface presents the permission request in a non-intrusive manner, allowing users to grant access with just a few clicks.

- **Collect Screen Dimensions**
  - **Screen width Detection:** To collect screen dimensions, our system automatically detects the screen width of the user's device. This information is used to set up the calibration process within the screen dimensions, ensuring an optimal viewing experience. By accurately determining the screen width, we can tailor the calibration process to fit the user's device, enhancing usability and visual quality. This approach allows us to deliver a consistent and optimized user interface across different devices and screen sizes. By dynamically adapting to the user's screen width, we ensure that the calibration process is accurate and effective.

- **Device Compatibility Check**
  - **Compatibility:** We conduct a thorough check to determine if your device is compatible with eye tracking technology, ensuring optimal functionality. This compatibility check is crucial to ensure that our system works seamlessly with your device, providing you with the best

possible experience. By verifying compatibility, we can guarantee that our eye tracking features will perform reliably and accurately on your device. This proactive approach allows us to identify and address compatibility issues upfront, minimizing disruptions.

- **Pop-ups:** We conduct a comprehensive compatibility check to ensure your device is compatible with our eye tracking technology. In cases where compatibility is lacking, we provide informative pop-up messages. These messages guide you toward suitable alternatives, ensuring a seamless experience with our system. This proactive approach demonstrates our commitment to providing a user-friendly and accessible eye tracking solution for all users. Furthermore, our pop-up

  messages include clear instructions on how users can resolve compatibility issues, such as updating their device's software or using a different browser.

- **Browser Compatibility Check**
  - **Compatibility:** We perform rigorous testing to confirm that our system works seamlessly with popular web browsers. This ensures that users can access and use our system without any compatibility issues, regardless of the browser they prefer. This commitment to browser compatibility allows us to reach a wider audience and provide a consistent user experience across different platforms.Thorough testing process this covers a range of popular web browsers, including Chrome, Firefox,

    Safari, and Edge, among others

- **Recommendations:** We conduct thorough compatibility checks to ensure our system works smoothly with popular web browsers. Following these checks, we encourage users to select the best-performing browser for an enhanced experience. Our clear recommendations are based on our compatibility testing and performance analysis, aiming to provide users with optimal performance and compatibility. By guiding users towards the most suitable browser, we help them achieve the best possible experience when using our platform.

- **Bandwidth Assessment**
  - **Evaluation:** We have implemented a bandwidth assessment system to evaluate the user's internet connection. This system analyzes the data transmission capabilities to ensure they are sufficient for optimal eye tracking performance. By conducting this assessment, we can provide users with a seamless and reliable eye tracking experience, even in environments with varying internet speeds.The bandwidth assessment system monitors the connection's speed and stability, adjusting the eye tracking performance to match the available bandwidth. This dynamic adjustment ensures that users receive the best possible experience regardless of their internet connection quality.
- **System Resource Utilization**
  - **Monitoring:** Our system includes efficient monitoring to check the availability of the camera, ensuring optimal use of system resources. This monitoring process helps us manage system resources effectively, reducing unnecessary strain and improving overall performance. By

checking the camera's availability, we can ensure that resources are allocated efficiently, enhancing the user experience.This proactive approach also helps prevent conflicts with other applications that may require camera access, ensuring a smoother user experience. By efficiently managing system resources.

4.1.2.2 Landing Page

- The Landing Page serves as the initial point of entry for users accessing the Eye Point website. It is designed to captivate visitors with engaging visual elements, including graphics and images, and an eye-catching color scheme. The Landing Page provides an overview of Eye Point's key features, highlighting its benefits and value proposition**.**

Key Features

- **Engaging Visual Design**
  - **Innovative Design:** Our engaging visual design incorporates eye-catching graphics and images to enhance the eye-tracking experience. By using a professional and modern color scheme, our design aims to convey the innovative nature of the technology. This approach not only makes the user interface visually appealing but also creates a more immersive and engaging experience for users. Through innovative design choices, we strive to enhance the overall user experience and highlight the advanced features of our eye-tracking technology. By combining aesthetics with functionality, we aim to provide a seamless and enjoyable user experience.
  - Our eye-tracking system features engaging visual design elements,

including eye-catching graphics and images. These visual components are designed to enhance the overall user experience and make the eye-tracking experience more visually engaging. By incorporating these design elements, we aim to create an interface that is not only functional but also visually appealing and immersive.

- **Interactive Demo or Video**
  - **Demo:** We offer an interactive demo that allows users to directly experience the features of our eye-tracking technology. This demo provides a hands-on experience, showcasing the capabilities of our system and allowing users to explore its functionalities. By providing an interactive demo, we aim to give users a better understanding of how our eye-tracking technology works and how it can benefit them.
  - **Explanatory Video:** In addition to the interactive demo, we provide an informative video that offers a comprehensive overview of our eye-tracking technology and its applications. This video is designed to give users a clear understanding of the technology's benefits and how it can be used in various applications. By offering both an interactive demo and an explanatory video, we aim to provide users with a complete picture of our eye-tracking technology and its potential uses.

- **Key Features Overview**
  - **Highlighting:** We provide a quick overview highlighting the key features of our eye-tracking system to pique user interest. This overview is designed to give users a glimpse of the capabilities and benefits of our technology, encouraging them to explore further. By

highlighting the main features, we aim to showcase the value proposition of our eye- tracking system and demonstrate how it can address their needs. By showcasing the key features in this manner, we aim to capture users' interest and encourage them to explore the full range of capabilities offered by our eye-tracking system.

- **Detailed Descriptions:** We offer detailed descriptions that explain the significance and potential benefits of each feature of our eye-tracking system. These descriptions provide users with a comprehensive understanding of how each feature works and how it can benefit them. By providing detailed explanations, we aim to help users make informed decisions about using our technology and understand how it can meet their needs. This approach enhances the overall user experience by empowering users with the knowledge they need to make the most of our eye-tracking technology.

- **User-Friendly Navigation**
  - **Easy Access:** Our eye-tracking system includes a user-friendly navigation system with a simple and intuitive menu. This menu is designed to make it easy for users to navigate through different sections of the system, allowing them to access the features they need quickly and efficiently. By providing easy access to key sections, we aim to enhance the overall user experience and make our eye-tracking system more user- friendly.
  - Our primary focus is on creating navigation elements that prioritize a seamless user experience, ensuring accessibility across all pages of our eye-tracking system. We strive to design intuitive menus and interfaces that make it effortless for users to find and access the features they

need. By emphasizing usability and accessibility, we aim to provide a user- friendly navigation experience that enhances overall satisfaction and ease of use for all users.

4.1.2.3 Environment Setup

- The Environment Setup component is an integral part of EyePoint, guiding users through the setup and calibration process for precise eye tracking. It provides interactive tutorials featuring on-screen visual cues and progress indicators, ensuring users follow each step accurately. This facilitates an efficient and user-friendly calibration experience, enhancing the accuracy of the EyePoint system.

Key Features

- **Start Calibration Button**

  - **Clear Button:** Our system includes a clear button in the setup interface, specifically designed to encourage users to start the calibration process easily. This button is prominently displayed, making it easy for users to locate and initiate calibration. By providing a clear and intuitive interface for starting calibration, we aim to streamline the setup process and ensure that users can quickly and easily begin using our eye-tracking system.

  - The start calibration button in our system is designed to be interactive, encouraging user engagement and making the calibration process straightforward. This button is prominently displayed in the setup interface, prompting users to initiate calibration with ease. By making the button interactive and visually appealing, we aim to enhance the user experience and ensure that users can start the calibration process effortlessly.Additionally, the button's design is optimized to be intuitive, ensuring that users can easily identify it as the starting point for calibrating the eye-tracking system.

- **Ensure Visibility of Face**
  - **Face Positioning:** Our system includes a face positioning feature that prompts users to position their face within a designated area to start calibration. This feature ensures that the user's face is properly aligned for accurate eye tracking. By prompting users to position their face correctly, we enhance the visibility of the face and improve the accuracy of the calibration process. This feature is designed to make calibration more efficient and effective, ultimately improving the overall performance of our eye-tracking system.This feature helps ensure that the user's face is consistently positioned for optimal tracking, leading to more reliable and accurate results.
- **Instructions and Guidelines**
  - **Clear Instruction:** Our system provides clear, step-by-step instructions for setting up and calibrating the camera, ensuring optimal performance of the eye-tracking system. These instructions guide users through the setup process, making it easy for them to configure the system correctly. By providing clear guidance, we aim to help users get the most out of our eye-tracking technology and ensure that it functions optimally for their needs.These instructions are designed to be easy to follow, even for users with little technical knowledge, ensuring a smooth setup process. Additionally, the guidelines include tips and best practices for calibrating the camera, helping users achieve the most accurate eye-tracking results. Overall, our goal is to provide

comprehensive instructions that empower users to set up and calibrate the system with confidence.

Visual Guide: Our system includes visual guides to assist users in accurately positioning their camera for precise eye tracking. These visual guides are designed to complement the step-by-step instructions, providing users with a clear visual reference to ensure that the camera is positioned correctly. By including visual guides, we aim to make the setup process more intuitive and help users achieve optimal eye-tracking accuracy.These visual guides are accompanied by explanatory text, ensuring that users understand the importance of proper camera positioning for optimal performance. Additionally, the visual guides are designed to be user-friendly, providing clear and easy-to-follow instructions for users of all experience levels.

- **Calibration Screen**
  - **User-Friendly Calibration:**Our system features a user-friendly calibration screen designed to make the calibration process simple and easy for users. The interface is intuitively designed, with features that collect eye data to ensure precise and accurate gaze tracking. that users can calibrate their eye-tracking system smoothly, leading to more accurate results. The interface also provides real-time feedback, allowing users to adjust their positioning for optimal calibration.The calibration screen includes interactive elements that guide users through the calibration process step by step, making it easy to understand and follow. Additionally, the screen provides clear instructions and visual cues to help users position their eyes correctly,Overall, the design of the

calibration screen is focused on providing a seamless and intuitive experience for users, enhancing usability of our eye-tracking system.

- **Interactive Tutorial**
  - **Visual Cues:** Our system includes an interactive tutorial with on-screen visual cues to guide users through the calibration process. These visual cues are designed to make the calibration process more user-friendly and easy to understand. The interactive nature of the tutorial allows users to learn at their own pace, ensuring that they can calibrate their eye-tracking system accurately. By providing this interactive tutorial, we aim to enhance the overall user experience and make the calibration process more accessible to all users.The visual cues in the interactive tutorial help users understand the calibration process better by providing clear and concise instructions.
- **Progress Indicator**
  - **Tracker:** Our system includes a progress tracker on the calibration screen, providing users with a clear visual indication of their progress in completing the calibration process. This tracker helps users stay informed about their progress and gives them a sense of accomplishment as they move through the calibration steps. By including this progress indicator. The progress indicator also helps users understand the remaining steps in the calibration process, allowing them to anticipate what comes next. This feature contributes to a more organized and efficient calibration experience, ensuring that users can complete the process with confidence.

4.1.2 4 iCleanser – AI

- iCleanser AI stands as a cutting-edge image processing software utilizing sophisticated artificial intelligence algorithms. Primarily focusing on facial identification, noise reduction, and frame reduction, this advanced tool enhances image quality through intelligent processing. By

Leveraging AI, iCleanser ensures optimal visual clarity, making it particularly adept for refining facial recognition tasks, minimizing noise, and streamlining frames for improved overall image quality. iCleanser AI is an advanced image processing software that leverages artificial intelligence to enhance image quality. Its focus on facial identification, noise reduction, and frame reduction makes it an ideal tool for a wide range of applications, including facial recognition, video editing, and animation. With iCleanser AI, users can ensure optimal visual clarity and improved image quality.

Key Features

- **Face Detection**
  - **Accuracy:** Our system uses a smart algorithm to accurately detect faces, ensuring precision in identification.The accuracy of face detection systems can vary depending on several factors, including the quality of the input Data, the sophistication of the detection algorithm, and the specific conditions under which the system operates.The quality and resolution of the input images or video frames significantly impact the accuracy of face detection. High-resolution images with clear, well-lit faces are generally easier to detect accurately and low-resolution images are also recognised.The choice of detection algorithm significantly impacts accuracy. Advanced algorithms based on deep learning techniques, such as convolutional neural networks

(CNNs), tend to achieve higher accuracy compared to traditional methods like Viola- Jones or Histogram of Oriented Gradients (HOG). Deep learning models can learn complex patterns and features directly from data, leading to more robust and accurate face detection.

- **Robust:** It remains effective even in changing lighting conditions, providing robust face detection for reliable performance.A robust face detection system achieves high accuracy in detecting faces in various types of images or video frames, including those with different lighting conditions, backgrounds, poses, expressions, occlusions, and ages. It minimizes false positives (incorrectly detecting non-faces as faces) and false negatives (missing actual faces).A robust face detection system

  adapts to changes in the environment or input data, such as variations in image resolution, aspect ratio, or quality. It can handle real-world challenges, such as noisy or low-quality images, without significantly compromising accuracy.

Frame Reducer Mechanism

- **Reduction:** Reduce captured frames in real-time, making the system faster and more efficient.By reducing the number of frames in the video, the overall data size of the recording is decreased. This can be beneficial for storage purposes, especially in scenarios where storage space is limited or where large amounts of video data need to be transmitted over networks with restricted bandwidth.Processing and analyzing video data can be computationally intensive. By reducing the frame count, the computational resources required for tasks such as video editing, analysis, or playback are minimized, leading to improved efficiency and performance.

- **Cleanser**
  - **Noise Reduction:** We use techniques to reduce noise and enhance image quality for clearer visuals.Applying spatial filters, such as Gaussian smoothing or median filtering, to smooth out noise while preserving edge details.Using frequency-domain filters, such as Wiener or Butterworth filters, to attenuate noise in specific frequency bands.Noise reduction improves image quality by reducing distractions and enhancing clarity, making it easier to discern details and patterns in the image.
  - **Remove Distortions:** By removing artifacts and distortions, our system ensures improved accuracy in the captured information..Removing these distortions improves the accuracy and fidelity of the captured information.Removing distortions enhances the accuracy and reliability of the captured information, making it more suitable for analysis, interpretation, and decision-making tasks.Removing distortions enhances the accuracy and reliability of the captured information,

    making it more suitable for analysis, interpretation, and decision-making tasks.

- **Image Enhancement**
  - **Sharpening Filters:** Sharpening filters to make captured images clearer and improve visual details.Unsharp masking is one of the most widely used sharpening techniques. It involves creating a blurred version of the original image (the "mask") and then subtracting this

blurred version from the original to produce a sharpened image. The degree of sharpening is controlled by adjusting the strength of the mask and the radius of the blur.Gradient-based sharpening techniques use gradient information to enhance image details. These techniques calculate the gradient magnitude and orientation at each pixel and then adjust pixel values based on the gradient direction to emphasize edges and suppress noise.

- **Uploader**

  - By reducing noise and enhancing resolution, we optimize the system's image quality for better overall performance.The uploader component captures individual frames from a video stream or image sequence. These frames typically represent snapshots of the scene captured by a camera or generated by some other source.In some cases, the uploader may operate asynchronously, allowing it to continue capturing and uploading frames while awaiting responses from the recognition service. This asynchronous processing enables real-time or near-real- time recognition of streaming video data.Overall, the uploader for frames recognition facilitates the seamless transfer of video frames from a local source to a remote recognition service, enabling real-time analysis, detection, and classification of objects or patterns within the video stream.

4.1.2.5 Backend Receiver

- The Backend Receiver is a pivotal component that collects and processes data from the eye-tracking system, ensuring seamless communication and supporting the website's functionality. By facilitating the smooth transfer of information, it plays a crucial role in enhancing the user experience by ensuring accurate and timely updates from the eye- tracking system and responsive platform.

Key Features

- **Data Storage**
  - The backend system receives and stores face and eye-tracking data from users, facilitating the recognition of the user's face. This data enables the system to analyze and track facial features, contributing to personalized user identification and enhancing user experience. Additionally, the backend serves as a repository for the collected data, allowing for continuous analysis and improvement of face and eye-tracking algorithms. It plays a crucial role in maintaining a comprehensive record of user interactions, supporting further developments in facial recognition technology and user-centric applications.
  - **Scalable:** The system employs a scalable storage infrastructure to efficiently handle and organize data, providing seamless performance scalability as the user base expands. This ensures optimal functionality and responsiveness, even with increasing volumes of face and eye-tracking information.

- **Image Quality Check**
  - The system conducts analyses of lighting conditions to guarantee optimal visibility and clarity for accurate face and eye-tracking results. This helps enhance the performance and reliability of the recognition process across varying lighting environments.
  - **Resolution Check:** The system verifies the resolution of uploaded images to ensure they meet the required standards.This step contributes to maintaining the quality of the image data used in the system.

- **Image Processor**
  - System processes facial data quickly and effectively for prompt results.
  - **Compatibility for Different Image Formats**: The system boasts compatibility with diverse image formats, ensuring flexibility in handling an array of facial data types. This adaptability enhances its capability to process and analyze facial information seamlessly.

4.1.3 FUN WINDOW

- The application employs real-time gaze visualization, incorporating interactive elements to elevate user engagement and deliver a distinctive and immersive experience. By providing a dynamic representation of the user's gaze in real-time, augmented with interactive elements, it enhances the overall user experience, making interactions more engaging and personalized. This feature adds a layer of interactivity and responsiveness, creating a unique and immersive environment for users within the application.

4.1.3.1 Fun Box

- The interactive application showcases a real-time visualization of the user's gaze, significantly elevating user engagement and delivering a distinctive interactive experience. By dynamically presenting the user's gaze movements, the application creates an immersive environment that not only captures attention but also facilitates personalized interactions.

This real-time visualization feature enhances the overall user experience, making the application both engaging and uniquely interactive.

Key Features

- **Real-time Gaze Visualization**
  - **Display Gaze Point:** This feature lets you track your screen using your eyes and display the user's gaze point in real-time on the screen.Displaying the gaze point involves visually indicating where a person is looking on a screen or in an environment. This is commonly done in real-time using eye-tracking technology.Once the gaze point coordinates are calculated, they are overlaid onto the screen or environment in real-time. This is often achieved by superimposing a visual indicator, such as a dot or crosshair, at the calculated coordinates. The indicator moves dynamically as the person's gaze shifts, providing a continuous representation of their point of focus.
  - **Smooth and Responsive:** Minimize processing delays and optimize rendering to achieve real-time updates. It ensures smooth and responsive gaze visualization.The eye-tracking system should have a high sampling rate, meaning it captures eye movements frequently and accurately. This ensures that changes in gaze direction are detected promptly, leading to smoother and more responsive visualizations.Proper calibration of the eye-tracking hardware is essential for accurate gaze tracking. Calibration ensures that the system accurately maps eye movements to screen coordinates, reducing errors and improving responsiveness

- **Pixel Highlighting Animation**
  - **Highlighting:** This feature animates a highlight around your current gaze point on the screen, making it visually clear where you're looking.Highlighting the gaze using animation, such as a pointer or laser, is a technique commonly employed in eye-tracking systems to provide visual feedback about where a user is looking. This helps users understand and interact with the system more intuitively.A laser animation simulates the effect of a laser beam emanating from the user's gaze point. The laser beam may appear as a thin line or dot that extends from the gaze position, providing a visual trace of the user's attention path.
  - **Customization:** Users can customize the animation style, ensuring a personalized and convenient highlighting experience based on individual preferences.Users can choose the color of the gaze pointer or laser animation. Offering a color picker or a predefined set of color options allows users to select a color that contrasts well with the background and is easy for them to perceive.Users can adjust the size of the gaze pointer or laser animation. Providing options to increase or decrease the size allows users to customize the visibility of the animation based on their preferences and viewing conditions.Users can select the shape of the gaze pointer or laser animation. Offering different shapes such as arrows, circles.Providing default settings for gaze highlighting animations ensures that users have a starting point if they do not wish to customize the animations themselves.

- **User-Customizable Fun Window Themes**
  - Our system offers Customizable fun window themes, allowing users to personalize the appearance of their interface.Users can access a settings menu or control panel where they can browse and select from a range of pre-installed themes.Themes may include options such as vibrant colors, whimsical patterns, retro-inspired designs, or themed graphics.Users can select a background image or texture to personalize their desktop background.The operating system include a set of default themes that cater to a wide range of preferences and usability needs.
  - **Themes:** Users can choose from a variety of themes, adding a touch of fun and individuality to their window displays.Themes can also have implications for accessibility, as they affect factors such as color contrast, font readability, and icon clarity. Accessible themes prioritize usability and readability for users with visual impairments or other accessibility needs.Themes can contribute to user engagement by offering novelty and variety. Users may enjoy exploring different themes and switching between them to refresh their digital experience or express their personality.

- **Compatibility with Different Screen Resolutions**
  - **Responsive:** Fun Window is designed with responsive design techniques, ensuring adaptability to various screen sizes and resolutions.Users may have different devices with varying screen

resolutions, such as desktop monitors, laptops, tablets, or smartphones. Ensuring that themes are compatible with different resolutions helps maintain a consistent user experience across devices, regardless of the screen size or aspect ratio.user-friendly experience across diverse devices and usage scenarios. This helps maximize usability, accessibility, and user satisfaction across the entire spectrum of screen sizes and resolutions.

CHAPTER 5
RESULTS AND DISCUSSION

5.1 PERFORMANCE PARAMETERS / TESTING

5.1.1 TESTING OBJECTIVES

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding error, if it exists. The tests are inadequate to detect possibly present errors. The software more or less confirms the quality and reliable standards.

5.1.2 TESTING LEVELS

System testing is a stage of implementation which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

Unit Testing

In the lines of strategy, all the individual functions and modules were put to the test independently. By following this strategy all the errors in coding were identified and corrected. This method was applied in combination with the White and Black Box testing Techniques to find the errors in each module.

Integration Testing

Data can be lost across the interface; one module can have an adverse effect on others. Integration testing is a systematic testing for constructing program structure. While at the same time conducting tests to uncover errors associated within the interface. Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order sets is conducted. The objective is to take unit tested

modules and combine them to test it as a whole. Thus, in the integration-testing step all the errors uncovered are corrected for the next testing steps.

Validation Testing

The outputs that come out of the system are as a result of the inputs that go into the system. The correct and the expected outputs that go into the system should be correct and proper. So this testing is done to check if the inputs are correct and they are validated before it goes into the system for processing.

Acceptance Testing

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required. This is done in regard to the following point:

1. Input screen design
2. Output screen design

An acceptance test has the objective of selling the user on the validity and reliability of the system. It verifies that the system's procedures operate to system specifications and that the integrity of important data is maintained. Performance of an acceptance test is actually the user's show. User motivation is very important for the successful performance of the system. After that a comprehensive report is prepared.

5.2 RESULTS & DISCUSSION

5.2.1 TEST CASE REPORT

TEST CASE ID	ACTION TO BE PERFORMED	EXPECTED RESULT	ACTUAL RESULT	PASS / FAIL
1.	Selecting "Fun Window" Button.	Display Fun Window Page	Displaying Fun Window Page	Pass
2.	Selecting "Eye Tracking" Button.	Display Eye Tracking Page	Displaying Eye Tracking Page	Pass
3.	Selecting "Start Calibration" Button	Display Compatibility Check Page	Displaying Compatibility Check Page	Pass
4.	Selecting "Start Calibration" Button	Display Compatibility Check Page	Displaying Instruction s Page	Fail

5.	Selecting "Read Instructions" Button	Display Instructions Page	Displaying Instructions Page	Pass

Fig 5.1 Testing Table For Modules

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

Eye tracking technology represents a transformative force in various fields, revolutionizing the way we interact with computers, advancing psychological research, and offering novel applications in healthcare. The ability to accurately monitor and interpret eye movements provides valuable insights into human behavior, cognition, and engagement.

In human-computer interaction, eye tracking has reshaped user experience by enabling dynamic adaptations to interface elements based on users' gaze patterns. This has profound implications for virtual and augmented reality, gaming, and assistive technologies, creating more intuitive and responsive interactions.

Psychological and neuroscience studies have benefited significantly from eye tracking, allowing researchers to delve deeper into understanding attention, perception, and decision-making processes. The technology has become an invaluable tool for unraveling the mysteries of the mind and gaining insights into cognitive functions.

Healthcare applications, particularly in neurological diagnosis and assistive technologies, highlight the potential of eye tracking to make a positive impact on individuals' lives. Early detection of neurological disorders and empowering those with motor impairments are tangible outcomes of this technology, showcasing its capacity to contribute to improved healthcare outcomes.

While the potential is vast, challenges such as ensuring accuracy and addressing individual variability persist. Ongoing advancements in hardware, machine learning, and data processing are actively addressing these challenges, fostering the continued growth and adoption of eye tracking technology.

As we stand at the forefront of this technological frontier, the future holds promising developments. The evolution of eye tracking technology is set to redefine our understanding of human-computer interaction, psychological processes, and healthcare diagnostics. With its increasing integration into diverse domains, eye tracking stands as a testament to the dynamic interplay between technological innovation and our quest for a deeper comprehension of the human experience.

6.2 FUTURE ENHANCEMENTS

Future enhancements in eye tracking technology hold the potential to further amplify its impact across various domains. The continuous evolution of this technology is likely to bring about advancements in precision, adaptability, and accessibility. Here are potential future enhancements:

Real-time Adaptability:

Advancements in machine learning and artificial intelligence could enable eye tracking systems to adapt in real-time based on users' changing behaviors. This adaptability can lead to more personalized and responsive user interfaces, especially in applications like gaming, virtual reality, and augmented reality.

Expanded Application in Healthcare:

Eye tracking technology is likely to play an increasingly significant role in healthcare. This may involve refining its application in early detection and monitoring of neurological disorders, as well as exploring new avenues in mental health diagnostics and treatments.

Integration with Wearable Devices:

Future eye tracking solutions may integrate seamlessly with wearable devices, such as smart glasses or headsets. This integration can enhance user experience and open up new possibilities in hands-free control, navigation, and information retrieval.

Eye Tracking in Autonomous Vehicles:

Eye tracking technology could be integrated into the design of autonomous vehicles to monitor driver attention and alertness. This can contribute to the development of safer and more efficient transportation systems by preventing accidents caused by driver distraction or fatigue.

Cross-Device and Cross-Platform Compatibility:

Future enhancements may focus on improving the interoperability of eye tracking across different devices and platforms. This can facilitate a consistent user experience and enable the seamless transition of eye tracking data between various applications and devices.

Advancements in Remote Eye Tracking:

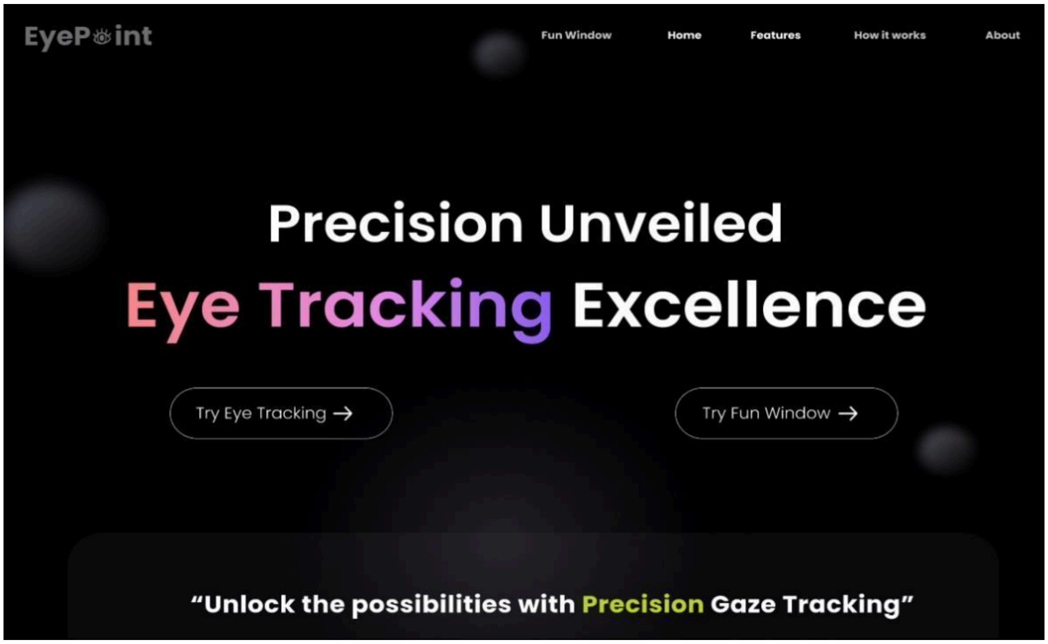
Remote eye tracking technology, which allows eye movements to be tracked without physical contact with the user, may see improvements. This can enhance user comfort and extend the range of applications in areas like market research, UX testing, and virtual collaboration.

Accessibility Features:

Future developments may prioritize enhancing accessibility features, making eye tracking technology more inclusive for individuals with diverse abilities. This can involve refining gaze-based interaction methods and ensuring that eye tracking is easily adaptable to different user needs.As eye tracking becomes more pervasive, future enhancements may include robust ethical considerations and privacy measures. Developers and researchers may focus on ensuring responsible use, data security, and transparency in collecting and processing eye tracking information.

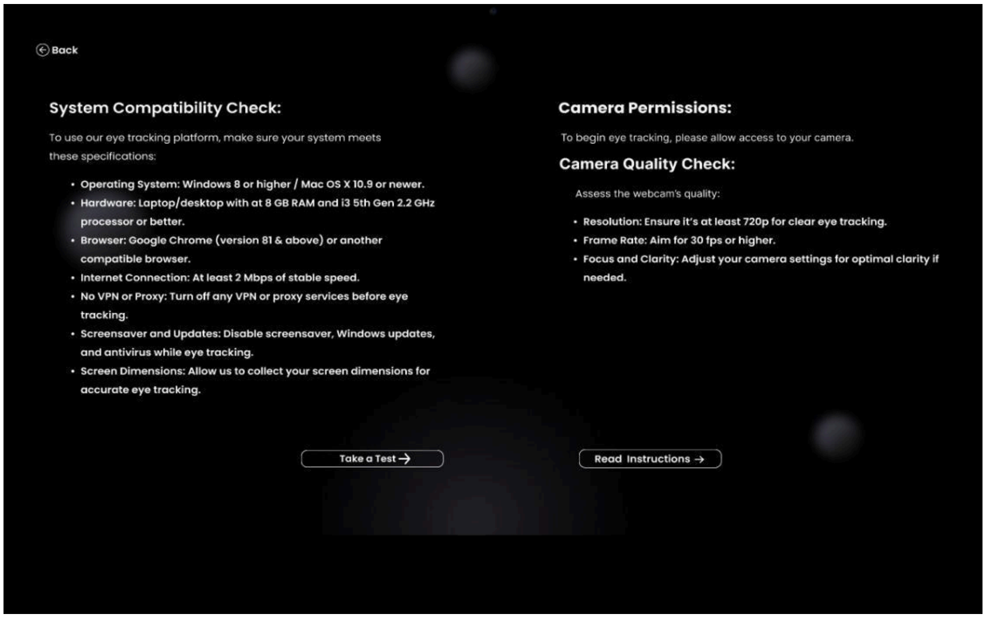
APPENDICES

A1 EyePoint Landing Page

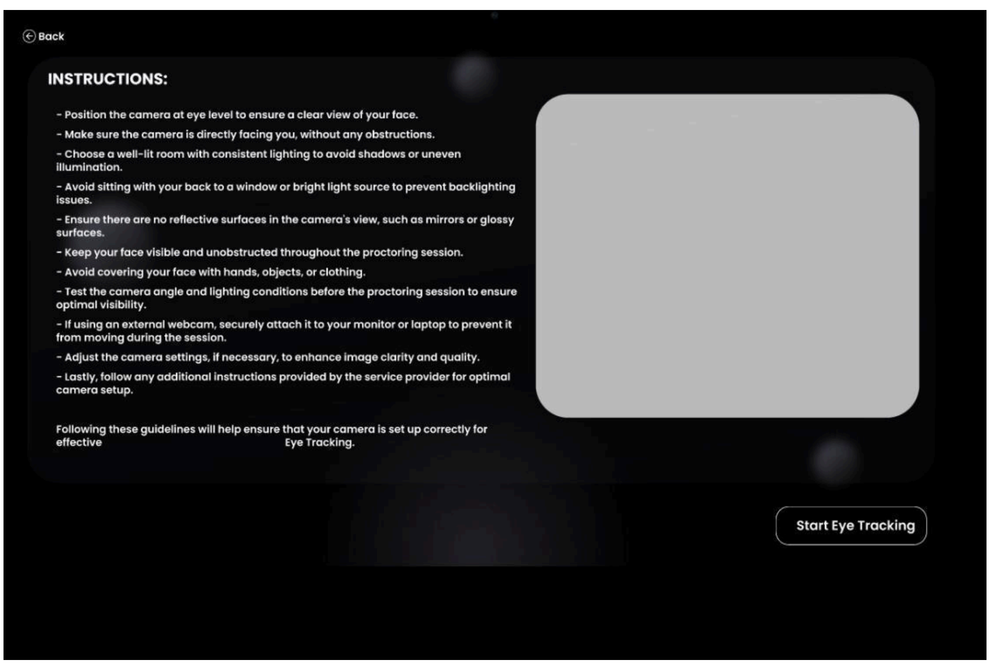


A2 Fun Window Landing Page

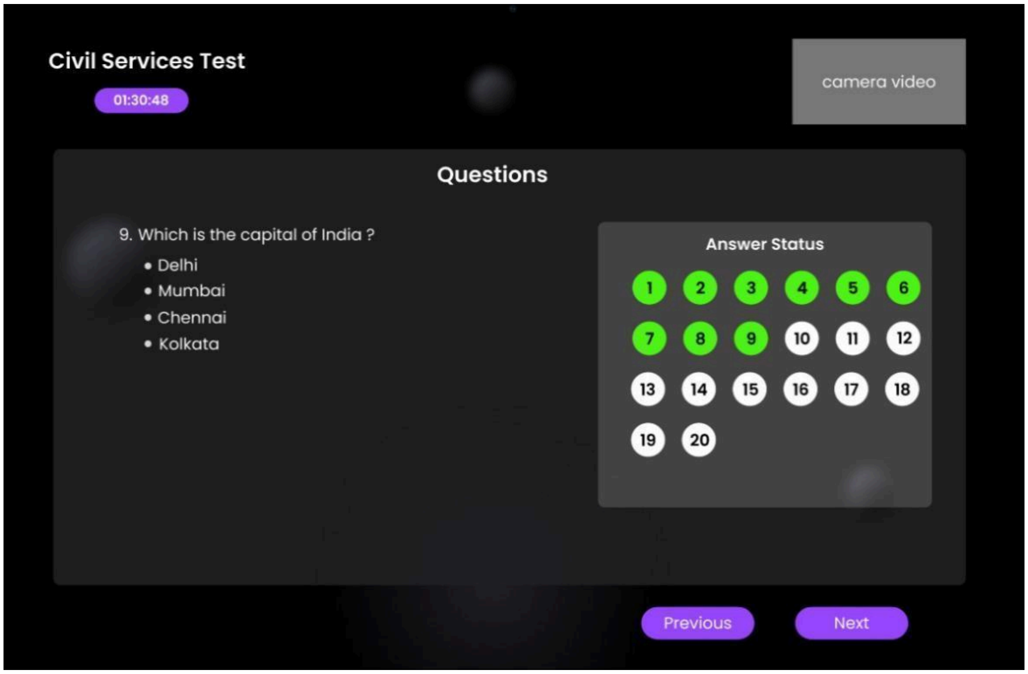
A3 System Compatibility Check Page



A4 Eye Tracking Instruction Page



A5 Exam Monitoring Page



CHAPTER 7
REFERENCE
S

[1] A. Pas˘ aric˘ a, R. G. Bozomitu, V. Cehan, and R. G. Lupu, “Pupil˘ Detection Algorithms for Eye Tracking Applications,” no. January 2018, 2015, doi: 10.1109/SIITME.2015.7342317.

[2] V. Clay, P. Konig, and S. K¨ onig, “Eye Tracking in Virtual Reality Eye¨ Tracking in Virtual Reality,” no. April, 2019, doi: 10.16910/jemr.12.1.3.

[3] I. Rakhmatulin and A. T. Duchowski, “Deep neural networks for low cost eye tracking,” Procedia Comput. Sci., vol. 176, no. October, pp. 685–694, 2020, doi: 10.1016/j.procs.2020.09.041.

[4] J. Griffin and A. Ramirez, “Convolutional Neural Networks for Eye Tracking Algorithms,” arXiv Prepr. arXiv1612.04335, 2018.

[5] B. Brousseau and J. Rose, “Hybrid Eye-Tracking on a Smartphone with CNN Feature Extraction and an Infrared 3D Model,” pp. 1–21, 2020, doi: 10.3390/s20020543.

[6] W. Fuhl, T. Santini, G. Kasneci, and W. Rosenstiel, “based real time Robust Pupil Detection.”

[7] D. Cazzato, M. Leo, C. Distante, and H. Voos, “When I Look into Your Eyes : A Survey on Computer Vision Contributions for Human Gaze Estimation and Tracking,” 2020, doi: 10.3390/s20133739.

[8] M. Kaczorowska, “Brain sciences Interpretable Machine Learning Models for Three-Way Classification of Cognitive Workload Levels for EyeTracking Features,” 2021.

[9] R. G. Bozomitu, P. Alexandru, T. Daniela, and C. Rotariu, “for RealTime Applications,” pp. 20–23, 2019.

[10] Narayana Darapaneni, Meghana D Prakash, Bibek Sau, Meghasyam Madineni, Rahul Jangwan, Anwesh Reddy Paduri, Jairajan K P, Mugdha Belsare,PradeepMadhavankutty, Eye Tracking Analysis Using Convolutional Neural Network,2022.

[11] Yuqing Li; Yinwei Zhan; Zhuo Yang, Evaluation of appearance-based eye tracking calibration data selection, 2020.

[12] Milu Prince, Neha Santhosh, Nimitha Thankachan, Reshma Sudarsan Ms.Anjusree V.K, Eye Movement Classification Using CNN, 2020.

[13] Yuehan Yin, Chunghao Juan, Joyram Chakraborty, Michael P. McGuire, Classification of Eye Tracking Data using a Convolutional Neural Network, 2020.

[14] Shatha Iskandar , Zahraa Jaara , Mohamad Abou Ali, Fatima Sbeity , Abdallah Kassem, Lara Hamawy. Eye Tracking System Using Convolutional Neural Network, 2020