REAL-TIME EYE BLINK DETECTION USING COMPUTER VISION

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

by

Kasthuri Sai Chander (Reg.No - 39110470)
K.G.N. Satya Krishna (Reg.No - 39110449)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119

APRIL - 2023

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **Kasthuri Sai Chander (Reg.No - 39110470) and K.G.N.Satya Krishna (Reg.No - 39110449)** who carried out the Project entitled **"REAL-TIME EYE BLINK DETECTION USING COMPUTER VISION"** under my supervision from January 2023 to April 2023.

Internal Guide

Dr. B.U.ANU BARATHI, M.E., Ph.D.,

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF CSE

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.,

Submitted for Viva Voce Examination held on 24.04.23

Internal Examiner **External Examiner**

DECLARATION

I, **Kasthuri Sai Chander (Reg.No: 39110470)** hereby declare that the Project Report entitled **“REAL-TIME EYE BLINK DETECTION USING COMPUTER VISION”** done by me under the guidance of **Dr. B.U.ANU BARATHI, M.E., Ph.D.,** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE:24.04.23

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D**, **Dean**, School of Computing, **Dr.L.Lakshmanan, M.E., Ph.D.,** Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. B.U.ANU BARATHI, M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Currently, eye tracking and blink detection are gaining popularity among researchers and have the potential to play a bigger role in perceptual user interfaces in the future. Many people have found real-time eye tracking and eye blink detection technology to be a basic and difficult problem for system inputs and outputs. The webcam is typically used for input, and the output is typically received as a real-time blink count. Algorithms for deep learning and computer vision will be employed. The current models for eye blink detection are highly complicated and based on CNN. Modern techniques are less precise and less successful. We suggest a system for automatically detecting eye blinks in real-time that may be applied in a variety of ways at the lowest possible cost. It will use the features of our eyes to count the number of times we blink. An embedded eye blink detection system that is reliable and affordable is implemented using the aspect ratio method. This approach is precise, inventive, quick, and simple to use. It will use the features of our eyes to count the number of times we blink. It would carefully study and analyse the x and y coordinates. Accurate analysis of the eye's landmark points is done using computer vision. Additionally, the proposed application would measure the height and width of the eye region and rotate the display clockwise. The suggested system has the following benefits: security, interpretability, high accuracy, lightweight model, and quick processing. It could be used in face recognition software, video surveillance systems, eye tracking software, and medical settings where a person's blinking pattern is often a sign of eye dryness or an allergy.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
YOLO	You Only Live Once
EAR	Eye Aspect Ratio
ERD	Entity Relationship Diagram
HCL	Human Computer Interaction
LBP	Local Binary Patterns

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The eye is a visual sense in humans. A critical organ of vision it plays a very important part not only in life but also the human body. The human eye is the system that provides us with vision, which offers us access to more details about our environment compared to any of the other four organs. By processing the light that things reflect or emit, the eye enables us to see and understand the shapes, colors, and dimensions of items in the world. The eye can see in both high and low light, but it is unable to discern objects in complete darkness.

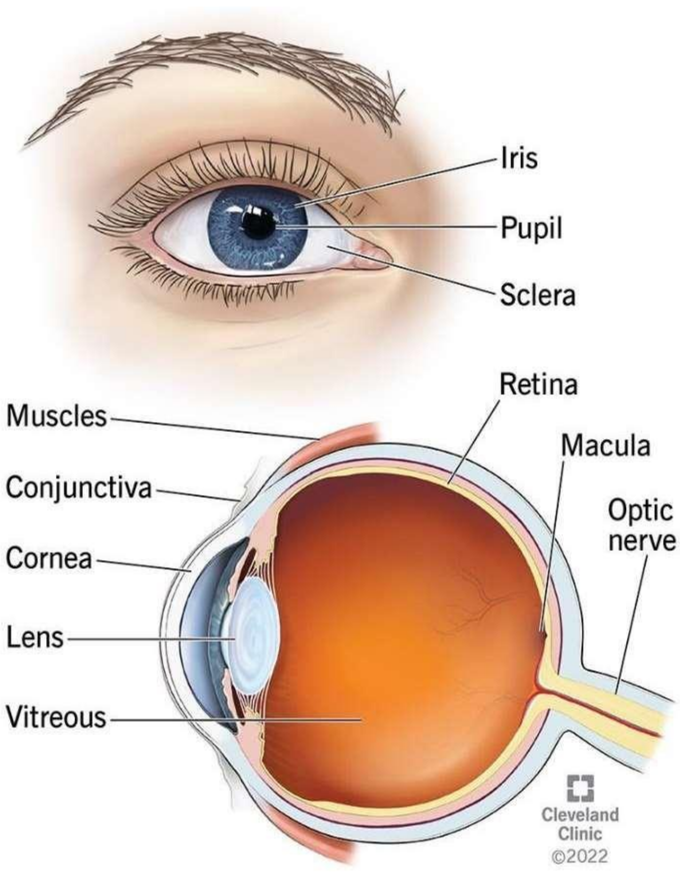


FIG: 1.1: ***Eye Anatomy***

1.2 VISIBLE EYE PARTS

Eyelid: Your eyelid shields your eye from debris, sweat, and other contaminants that could harm it. It promotes blinking to maintain the eye well-lubricated and nourished by opening and closing both intentionally and involuntarily.

Pupil: The area of a eye via that we can see, or the pupil, varies in size according on the amount of light. The pupil shrinks to let less light in if the surroundings is exceptionally bright, whereas it expands to let more light in when the atmosphere is dimmer. This ensures that the proper quantity of light reaches the retina at the rear of the eye and helps us see clearly in a variety of lighting conditions.

Sclera: The white portion of your eye, or sclera, serves as an outer layer of defense. It protects the optic nerve and can provide an accurate assessment of overall eye health. For instance, a pale sclera may point to liver issues, while a red sclera may signal that your eyes are weary or dry.

Iris: The iris, which is the colored portion of the eye, really regulates the pupil's size. In other words, it controls how much light enters the eye. The connective tissue and muscle that surround the pupil form the iris, and just like your thumbprint, it has an unique shape, texture, and coloring.

1.3 INTERNAL EYE PARTS

Cornea: The cornea is the transparent surface in front of your eye that allows light to enter. Your iris and pupil are directly covered, adding an additional layer of security. We do laser eye surgery on the cornea because irregularities in your cornea's curvature are what lead to an eye prescription and your requirement for spectacles. Your vision will be better if your cornea has a smooth surface.

Lens: The portion of the eye that focuses is the lens, which is situated beneath your iris. Light rays passing through the lens are focused so that they strike the retina at the proper angle, changing the focal distance of the eye. As you age, a protein development in the eye may cause the lens to become clouded. The term refers to a cataract. Fortunately, you can remove your lens and substitute it with a synthetic, transparent lens to restore your eyesight.

Aqueous humor: Your eyes continuously manufacture this watery liquid to support healthy ocular pressure and hydrate your cornea. This maintains the health of your eyes, which benefits clear eyesight. It is essential for strong vision and is removed from the eye at the same rate that it is created.

Ciliary muscle: The eye's ciliary muscle is the component that genuinely modifies the lens's structure to enable the eye to concentrate on different distances. Additionally, it regulates the flow of aqueous humor inside the eye and keeps the lens in the proper place in the central layers of the human eye.

Retina: The portion of tissue known as the retina is located at the back of the eye. The retina's main function is to take in light from the lens and transmit signals to the brain so it can convert it into a mental picture. Rods and cones are the two categories of photoreceptor cells found in the retina. Cone cells are in charge of detecting colour vision, whereas rods are in charge of detecting and classifying, darkness, and brightness. Maintaining your retina's function is essential since issues with the retina can lead to the loss of eyesight.

Fovea: The macula portion of the retina's fovea is where it is placed. Sharp central vision, necessary for reading, driving, and other activities requiring attention to visual detail, is controlled by this little region.

Macula: The macula is the retina's central region. We can read and see fine details because of the heavy proportion of light-sensitive cells (photoreceptors) in this incredibly small space. Our central vision entails a more detailed vision. Our peripheral, or side, vision is produced by the retina's bigger remaining region.

Choroid: Between the retina and the sclera is the choroid. It is made up of layers of blood vessels that bring nourishment to the eye's inner regions.

1.4 FUNCTION OF EYES

One of the most significant eye-related behaviors may be an eye blink. The cranial nerve VII innervates the orbicular is oculi muscle (the facial nerve). The palpebral orbicular is the muscle responsible for an instinctive blink and a voluntary wink, contracts the palpebral region of the eyelid, gently closing it. The levator muscle then relaxes. 23 Spontaneous involuntary blinking replenishes the precorneal tear coating. A loud noise, corneal, conjunctival, or cilial contact, a quick approach of an object, or any combination of these stimuli may cause a reflex blink, which serves as protection. The tissues around the eyelids—the forehead, temple, and cheek are contracted when the orbital is contracts, which causes the eyelid to shut firmly. Eye blinking has potential uses in a wide range of industries, such as human-computer interface, driver awareness indicator, psychology analysis, and more.

The importance of eye-blink detection is as follows

1.4.1 DRIVER DROWSINESS DETECTION

Nowadays, the transportation system is a crucial component of daily life. Everyone who is driving has the potential to become asleep or sleepy. Particularly after a brief sleep, in poor health condition, or following lengthy travels, sleepiness may occur. The perception of sleep lowers a driver's level of alertness, creating a hazardous situation and raising the likelihood of a traffic collision. The primary cause of road fatalities is discovered to be operators' loss of concentration while driving. And so by keeping drivers awake while operating a vehicle, the risk of accidents can be decreased.

To lower the crash rate, the auto driver must undergo onboard surveillance. To ascertain if someone is paying attention to an issue or not, there are numerous procedures. On the other hand, a disproportionate number of traffic accidents happen as a result of the transient condition of the driving force.

Thus creating a system to gauge a driver's level of attentiveness is becoming a serious problem in the field of active safety analysis (Wang et al. 2006). There are numerous methods for determining a driver's level of awareness.

The driver's face must be continuously captured by a lens for this process to work. The state of awareness can be determined using several facial features. Popular techniques include examining eye blink rates, eye aspect ratios, eye closure times, etc. If the observed result deviates from the expected value, this type of system can issue a warning.

1.4.2 HUMAN-COMPUTER INTERACTIONS

An incurable medical disorder known as motor neuron disease (MND) causes the patient's motor neurons to become disabled. Additionally, it causes the muscles of the hands, feet, and voice to become weak. As a result, the patient is unable to perform voluntary acts, and communicating their wants is extremely challenging. One such ailment where people are unable to move their lower extremities is tetraplegia. Solutions for people with the aforementioned disorders are now available thanks to innovations like the one offered and described throughout.

The point of interaction between a human user and a computer is known as the human-computer interface (HCI). Keyboards, computer mice, trackballs, touchpads, and touch screens are a few examples of commonly used inputs. All of these gadgets need to be manually operated, thus those with movement disabilities cannot use them.

Therefore, it is necessary to create alternate means of human-computer communication process that are appropriate for people with neuromuscular disorders that would enable them to participate in the information society. Researchers from all over the world have recently become interested in the creation of alternative interacting components. A user-friendly interface should meet a number of requirements for people with significant mobility impairments, including being non-contact and avoiding specialist equipment, having any real effectiveness, and operating on a consumer-grade computer.

1.4.3 VIRTUAL GAMING REALITY

The construction sector, biology, aviation, military training, and video games are just a few of the fields where virtual reality systems are finding utility due to the rapid advancement in technology. Consoles and hand-worn data gloves were the traditional ways of interacting in a virtual reality (VR) environment.

Despite having the ability to operate video games, these gadgets lack natural user interfaces. The entire VR experience is hampered by the usage of external physical input devices because the user is immersed in a virtual environment. With the development of BCI (brain computer interface) research, researchers have made some strides in human machine/computer interaction, aiming for complete intuitive control of distant virtual environments, a think-and-play type of user interface.With the help of this realistic interface, gamers may have a more thrilling gaming experience as their digital surroundings react to their eye blinking movements.

1.5 BACKGROUND

In order to improve human-computer interaction (HCI), the first attempt to track eye blinking was made in 2001. However, the investigation was only partially PC-based. Their techniques are necessary to locate eyes and monitor their specific position.

Mahalanobis's distance of different picture blobs dictate where the eyes are located. After that, they calculate the relationship between the tracked eye and the template. The system categorizes the state of the eyes as "open" if the outcome is less than a specific threshold and "closed" thereafter.

Post-research can be divided into two groups as a result of this study. The very first group starts by detecting the eyes and then registers it. To be even more precise, the group makes template out of the eyes that have been identified.

Then they compare the template to the actual eyes to determine where they are. The above approach annoys users since it must repeatedly initialize sight detection if the recorded eye location is found to be incorrect. It employs the scatter plot on the x, y axes of multiple photos and applies the percentage between their height and width.

The second category, in contrast to the first, conducts face and eye detection first, followed by eye - blink detection. The second group locates the area around the eyes on the face it initially detects. The procedures are then used to determine where the identified eyes are located.

The eye's blinking state can be represented by a variety of feature detection techniques, including Boosted LBP, Gabor wavelet, and picture flow. There is also the feature extraction template matching. Each of these groups' approaches have drawbacks for HCI on connected phones, which calls for real-time usability and functionality.

CHAPTER 2

LITERATURE REVIEW

[1] Bishar R. Ibrahim proposed a method for detecting blinks in real-time using machine learning and computer vision libraries. The proposed method is divided into four stages: (1) Taking frames with a raspberry pi camera attached to the raspberry pi 3 platform, (2) Using the haar cascade algorithm to identify faces in the captured frames, (3) Detecting facial landmarks with the facial landmark detector algorithm, and (4) Calculating the eye aspect ratio. The proposed method achieved high accuracy in detecting eye closure or opening. An aspect ratio method was used in this study to implement a reliable and low-cost embedded eye blink detection system on the Raspberry Pi platform. This method is precise, quick, and simple to use for detecting eye blinks.

[2] The proposed method detects intentional blinks automatically by determining the blink type threshold based on the waveform integration value as a new feature parameter. In addition, a blink measurement system was built to evaluate the proposed method, and the proposed method was tested by experiment. The system divides the interlaced image field into distinct fields in order to measure blinks with sufficient temporal resolution. It then extracts the waveform feature parameters and classifies the eye-blink types automatically. The experimental results show that the proposed method successfully classified intentional eye-blinks with 86% average accuracy, demonstrating its superiority over conventional methods based on eye-blink duration.

[3] Rahul Dasharath Gavas proposes an eye blink detector system based on personalized vision. The proposed method is ubiquitous and unobtrusive, and it can be implemented with standard webcams/mobile cameras, making it deployable in real-world scenarios. This method has been validated using data from the researchers' lab as well as an open data set. The results show that this system performs well in both cases for a variety of conditions such as natural/artificial light, with or without spectacles. They outperformed state-of-the-art approaches with an F score of 0.98 for their collected data and 0.91 for an open dataset.

[4] P. Ran studies the use of face recognition and detection technology to detect the closed state of drivers' eyes, and the results of the detection provide timely status warnings, reducing the likelihood of traffic accidents. Python, OpenCV, Dlib, and other third-party libraries are used in this article to obtain the driver's face video via the mobile phone camera and detect changes in eye aspect ratio (EAR) and mouth aspect ratio (MAR) to determine the driver's state. Finally, this test outputs the relationship between EAR and MAR over time.

[5] Based on the duration of eye blinks, this proposed work has established drowsy detection as well as an accident avoidance system. First, the open and closed states of the eye are determined using the eye aspect ratio (EAR). Furthermore, the blink duration or count during the transition from open to closed eye state is examined. Then, when the blink duration exceeds a certain threshold, it detects drowsiness and sends an alert message to the driver via the alarm. On the yawning dataset, the developed system achieved an approximate accuracy of 92.5%. (YawDD).

[6] This study attempted to assist patients through the use of a Brain-Computer Interface. Brain-Computer Interface (BCI) is a cutting-edge communication system that connects the brain to external electronic devices. BCI achieves this communication by capturing signals from the brain with electrodes. This study created a mobile application that allows users to call their favourite number using electrooculogram signals to assist physically challenged people in times of emergency.

[7] The main reason for developing a real-time system that monitors the condition of the driver's eyes is to reduce car accidents, which will benefit millions of people worldwide. This work proposes a method for detecting and monitoring the driver's eyes, specifically analysing the eyes and determining whether they are closed or open. The number of frames in which the eyes are closed. When this number of frames exceeds a certain threshold, the driver receives a visual alert on the navigation display indicating that he is drowsy.

[8] The algorithm presented in this paper uses an infrared camera to obtain frontal images of the driver and then uses the Viola-Jones algorithm to perform automatic face detection. Following that, the eye portion is extracted, and the classification between open and closed eyes is performed using one of two methods: a) A method based on the extraction of maximums and minimums of the eye's horizontal and vertical edges,

and b) A multi-Layer Perception (MLP) neural network. Finally, it detects drowsiness during the time interval in which the eyes were closed. The first method achieves 84% accuracy for open and close eye classification, while the second method achieves 97% accuracy, including test images in low-light conditions.

[9] This study compared three eye recognition algorithms for drivers that could be integrated into an open source solution to wake drivers as they begin to doze off: 1) CNN with Haar Cascade, 2) 68 landmark points on the face, and 3) gaze detection in 3 distinct face positions for both day and night driving conditions, as well as with and without glasses. Each of those variables is tested 100 times. The best algorithm is selected based on the number of correct detections, and it is then tested again based on light (day and night), angle of face (left, right, and centre), angle of the camera (left and right), and glasses (on and off) to detect both blinking and closed eyes.

[10] Smart phones and other information devices have become indispensable in our daily lives. As a result of the increased use of information devices, there has been an increase in eye strain and the development of various eye diseases. Because the accumulation of eye fatigue manifests itself in a change in the number of blinks, blink detection technology is required. Face image normalization was used in this study to enhance the blink detection system by reducing the noise associated with facial movement. In all subjects, the proposed system surpassed the results of blink detection accuracy using EAR.

2.1 INFERENCES FROM LITREATURE SURVEY

From the above-mentioned literature works, it is clear that there has been effective research on this topic has been done and many models have been proposed. It is evident that the above-mentioned systems have their own pros and cons. While some of the recent works involve hybrid technologies and provide better accuracies, they are still far from what is needed. With higher accuracy, comes the need for low computational costs, high processing speed, and most of all, the convenience of use.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

Different techniques have been reported for eye blink detection over the past few years. Although detection results achieved are promising, these traditional approaches are still far from being highly accurate and efficient. The existing systems are simple and effective but are extremely vulnerable to impact. The existing systems are based on multiple calculations and are highly complex in nature. Due to high complexity and inefficiency, the CNN-based models are inaccurate and have low precision. The existing system also leverages the Raspberry Pi Camera, found to be ineffective during low light conditions

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Convolutional neural networks (CNN) are commonly used in various computer vision applications, including real-time eye blink detection. However, CNN has some disadvantages that can impact the accuracy and efficiency of the algorithm. One disadvantage of CNN is its complexity and the need for large amounts of data for training. CNN requires a large number of parameters to be trained, which can take a long time and require significant computational resources. Additionally, training CNN requires a large dataset of labeled images or video data, which may not always be available or feasible to collect.

Another disadvantage of CNN is its susceptibility to over fitting. Over fitting occurs when a model becomes too specialized to the training dataset and performs poorly on unseen data. This can happen if the model is too complex or if the training dataset is too small or not representative of the target population. Further more, CNN may not always be suitable for real-time applications with strict latency requirements. The high computational cost of CNN may result in slow performance and latency issues, which can impact the user experience and the overall effectiveness of the algorithm.

In summary, while CNN is a powerful technique for computer vision applications, including real-time eye blink detection, it has some disadvantages that can impact the accuracy, efficiency, and latency of the algorithm. To address these limitations, alternative machine learning techniques such as shallow neural networks, decision trees, or support vector machines can be used to provide accurate and efficient eye blink detection in real-time applications.

3.2 PROPOSED SYSTEM

You Only Look Once (YOLO) is a state-of-the-art object detection algorithm that uses a single neural network to detect and classify objects in real-time. YOLO can be adapted for real-time eye blink detection and offers several advantages over traditional techniques such as CNN. One advantage of YOLO is its speed and efficiency. YOLO can detect and classify objects in real-time with high accuracy and low latency. This makes it suitable for real-time applications such as driver fatigue monitoring, human-computer interaction, and security systems.

Another advantage of YOLO is its ability to handle occlusions and small objects. YOLO uses a single neural network to detect objects, which enables it to detect and classify objects even if they are partially occluded or have small sizes. This is especially important in eye blink detection, where the eye region may be partially occluded by eyelids or other facial features. Furthermore, YOLO can be trained on small datasets, which can reduce the time and resources required for training. YOLO also offers high accuracy in detecting objects, including eye blinks, even with small training datasets. This is especially important for real-world scenarios where large annotated datasets may not be available or feasible to collect.

Finally, YOLO can be easily integrated with other computer vision techniques, such as face recognition or emotion detection, to provide a more comprehensive analysis of the eye region's data.

In summary, YOLO offers several advantages over traditional techniques such as CNN for real-time eye blink detection. It provides speed, efficiency, robustness to occlusions and small objects, and the ability to handle small training datasets. These advantages make YOLO a promising solution for real-time eye blink detection in various applications, such as healthcare, transportation, and entertainment.

3.3 FEASIBILITY STUDY

With an eye towards gauging the project's viability and improving server performance, a business proposal defining the project's primary goals and offering some preliminary cost estimates is offered here. Your proposed system's viability may be assessed once a comprehensive study has been performed. It is essential to have a thorough understanding of the core requirements of the system at hand before beginning the feasibility study. The feasibility research includes mostly three lines of thought:

- Economical feasibility
- Technical feasibility
- Operational feasibility
- Social feasibility

3.3.1 ECONOMICAL FEASIBILITY

The study's findings might help upper management estimate the potential cost savings from using this technology. The corporation can only devote so much resources to developing and analyzing the system before running out of money. Every dollar spent must have a valid reason. As the bulk of the used technologies are open-source and free, the cost of the updated infrastructure came in far cheaper than anticipated. It was really crucial to only buy customizable products.

3.3.2 TECHNICAL FEASIBILITY

This research aims to establish the system's technical feasibility to ensure its smooth development. Adding additional systems shouldn't put too much pressure on the IT staff. Hence, the buyer will experience unnecessary anxiety. Due to the low likelihood of any adjustments being necessary during installation, it is critical that the system be as simple as possible in its design.

3.3.3 OPERATIONAL FEASIBILITY

An important aspect of our research is hearing from people who have actually used this technology. The procedure includes instructing the user on how to make optimal use of the resource at hand. The user shouldn't feel threatened by the system, but should instead see it as a necessary evil. Training and orienting new users has a direct impact on how quickly they adopt a system. Users need to have greater faith in the system before they can submit constructive feedback.

3.3.4 SOCIAL FEASIBILITY

During the social feasibility analysis, we look at how the project could change the community. This is done to gauge the level of public interest in the endeavour. Because of established cultural norms and institutional frameworks, it's likely that a certain kind of worker will be in low supply or nonexistent.

3.4 REQUIREMENT SPECIFICATION

3.4.1 HARDWARE REQUIREMENTS

Processor	: Pentium Dual Core 2.00GHZ
Hard disk	: 120 GB
RAM	: 2GB (minimum)
Keyboard	: 110 keys enhanced

3.4.2 SOFTWARE REQUIREMENTS

Operating system	: Windows7 (with service pack 1), 8, 8.1 and 10
Language	: Python

3.5 LANGUAGE SPECIFICATION– PYTHON

Among programmers, Python is a favorite because to its user-friendliness, rich feature set, and versatile applicability. Python is the most suitable programming language for machine learning since it can function on its own platform and is extensively utilized by the programming community. Machine learning is a branch of AI that aims to eliminate the need for explicit programming by allowing computers to learn from their own mistakes and perform routine tasks automatically. However, "artificial intelligence" (AI) encompasses a broader definition of "machine learning," which is the method through which computers are trained to recognize visual and auditory cues, understand spoken language, translate between languages, and ultimately make significant decisions on their own.

The desire for intelligent solutions to real-world problems has necessitated the need to develop AI further in order to automate tasks that are arduous to program without AI. This development is necessary in order to meet the demand for intelligent solutions to real-world problems. Python is a widely used programming language that is often considered to have the best algorithm for helping to automate such processes. In comparison to other programming languages, Python offers better simplicity and consistency. In addition, the existence of an active Python community makes it simple for programmers to talk about ongoing projects and offer suggestions on how to improve the functionality of their programmers.

ADVANTAGES OF USING PYTHON

Following are the advantages of using Python:

- **Variety of Framework and libraries:**

A good programming environment requires libraries and frameworks. Python frameworks and libraries simplify programmer development. Developers can speed up complex project coding with prewritten code from a library. PyBrain, a modular machine learning toolkit in Python, provides easy-to-use algorithms. Python frameworks and libraries provide a structured and tested environment for the best coding solutions.

- **Reliability**

Most software developers seek simplicity and consistency in Python. Python code is concise and readable, simplifying presentation. Compared to other programming languages, developers can write code quickly. Developers can get community feedback to improve their product or app. Python is simpler than other programming languages, therefore beginners may learn it quickly.

Experienced developers may focus on innovation and solving real-world problems with machine learning because they can easily design stable and trustworthy solutions.

- **Easily Executable**

Developers choose Python because it works on many platforms without change. Python runs unmodified on Windows, Linux, and macOS. Python is supported on all these platforms, therefore you don't need a Python expert to comprehend it. Python's great excitability allows separate applications. Programming the app requires only Python. Developers benefit from this because some programming languages require others to complete the job. Python's portability cuts project execution time and effort.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

This graphic provides a concise and understandable description of all the entities currently integrated into the system. The diagram shows how the many actions and choices are linked together. You might say that the whole process and how it was carried out is a picture. The figure below shows the functional connections between various entities.

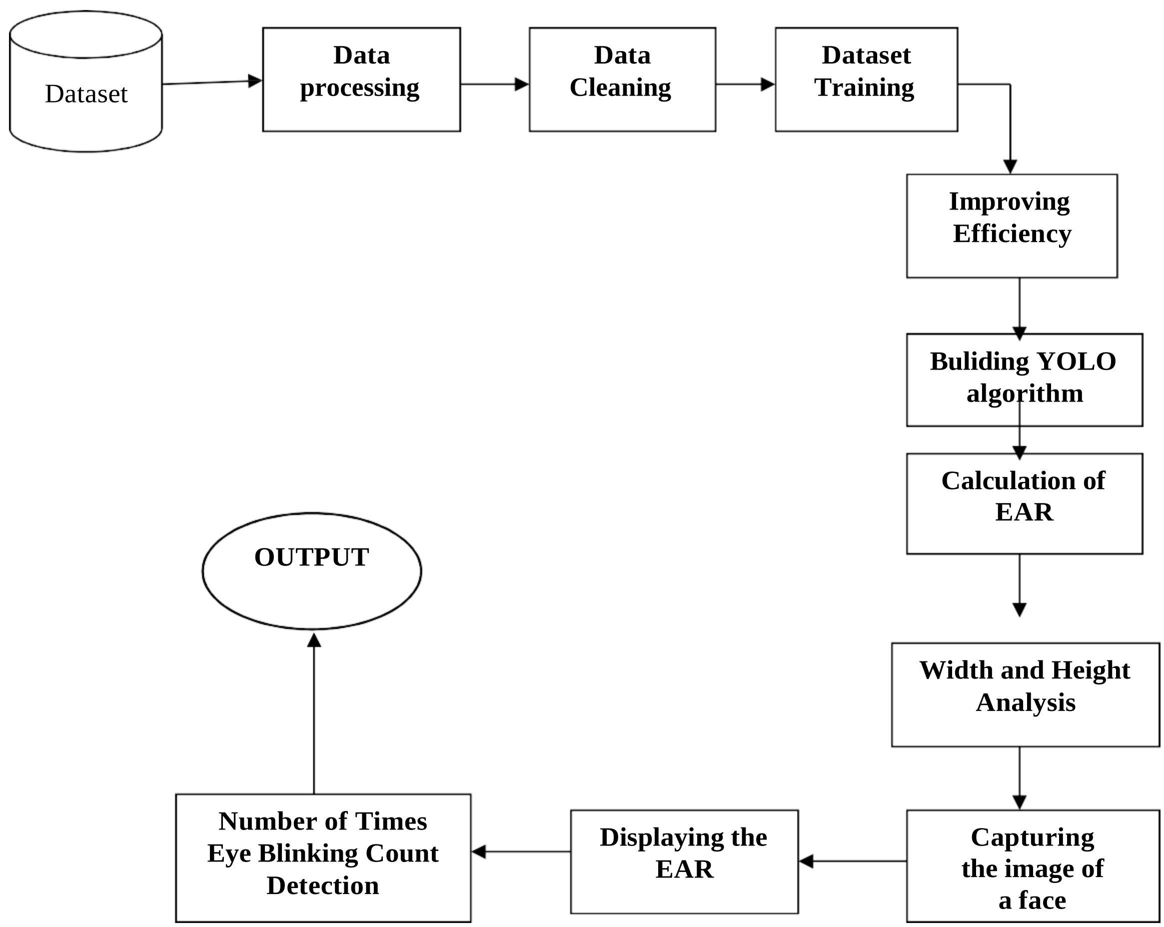


FIG:4.1: ***Architecture Diagram***

4.2 DATA FLOW DIAGRAM

To illustrate the movement of information throughout a procedure or system, one might use a Data-Flow Diagram (DFD). A data-flow diagram does not include any decision rules or loops, as the flow of information is entirely one-way. A flowchart can be used to illustrate the steps used to accomplish a certain data-driven task. Several different notations exist for representing data-flow graphs. Each data flow must have a process that acts as either the source or the target of the information exchange. Rather than utilizing a data-flow diagram, users of UML often substitute an activity diagram. In the realm of data-flow plans, site-oriented data-flow plans are a subset. Identical nodes in a data-flow diagram and a Petri net can be thought of as inverted counterparts since the semantics of data memory are represented by the locations in the network. Structured data modeling (DFM) includes processes, flows, storage, and terminators.

4.2.1 DATA FLOW DIAGRAM

The whole system is shown as a single process in a level DFD. Each step in the system's assembly process, including all intermediate steps, are recorded here. The "basic system model" consists of this and 2-level data flow diagrams.

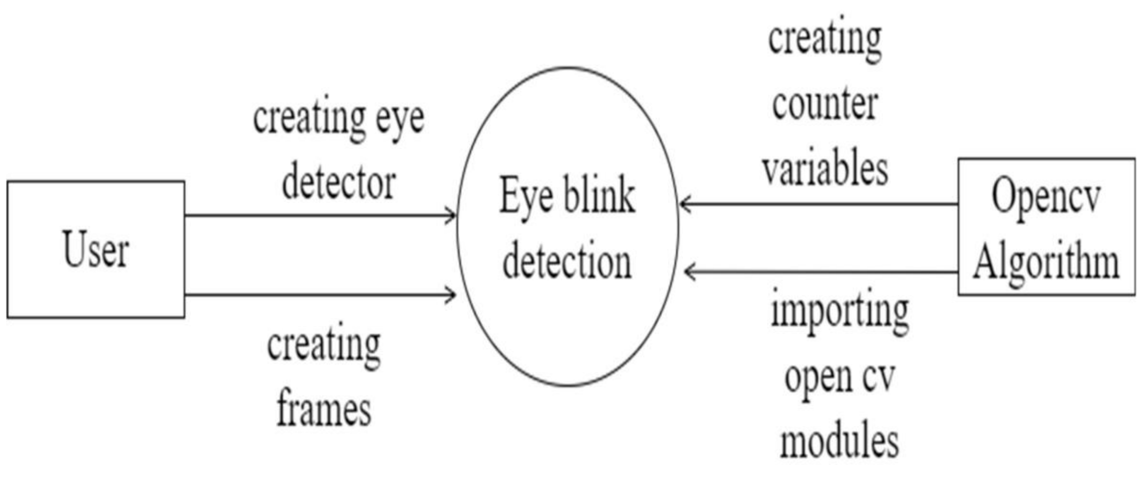


FIG:4.2: ***Data Flow Diagram Level 0***

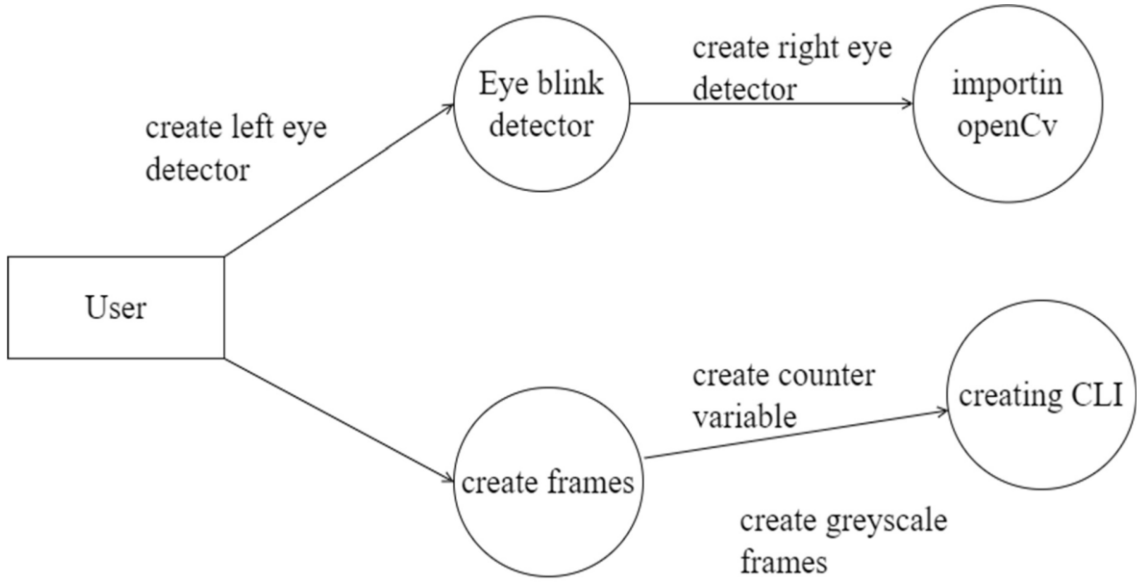


FIG: 4.3*: Data Flow Diagram Level 1*

4.3 ENTITY RELATIONSHIP DIAGRAM

- **Definition**

The relationships between database entities can be seen using an entity-relationship diagram (ERD). The entities and relationships depicted in an ERD can have further detail added to them via data object descriptions. In software engineering, conceptual and abstract data descriptions are represented via entity-relationship models (ERMs). Entity-relationship diagrams (ERDs), or simply entity diagrams are the terms used to describe the resulting visual representations of data structures that contain relationships between entities. As such, a data flow diagram can serve dual purposes. To demonstrate how data is transformed across the system. To provide an example of the procedures that affect the data flow.

1. One-to-One

Whenever there is an instance of entity (A), there is also an instance of entity (B) (B). In a sign-in database, for instance, only one security mobile number (S) is associated with each given customer name (A) (B).

2. One-to-Many

For each instance of entity B, there is exactly one occurrence of entry A, regardless of how many instances of entity B there are.
For a corporation whose employees all work in the same building, for instance, the name of the building (A) has numerous individual associations with employees (B), but each of these B's has only one individual link with entity A.

3. Many-to-Many

For each instance of entity B, there is exactly one occurrence of entry A, regardless of how many instances of entity B there are.
In a corporation where everyone works out of the same building, entity A is associated with many different Bs, but each B has only one A.

The abbreviation ER refers to a connection between two entities. The entities used and saved in the database are shown in relationship diagrams. They break down the process into its component parts and explain how they work. Attributed concepts, Relationship concepts, and Entity concepts are the building blocks for these kinds of diagrams.

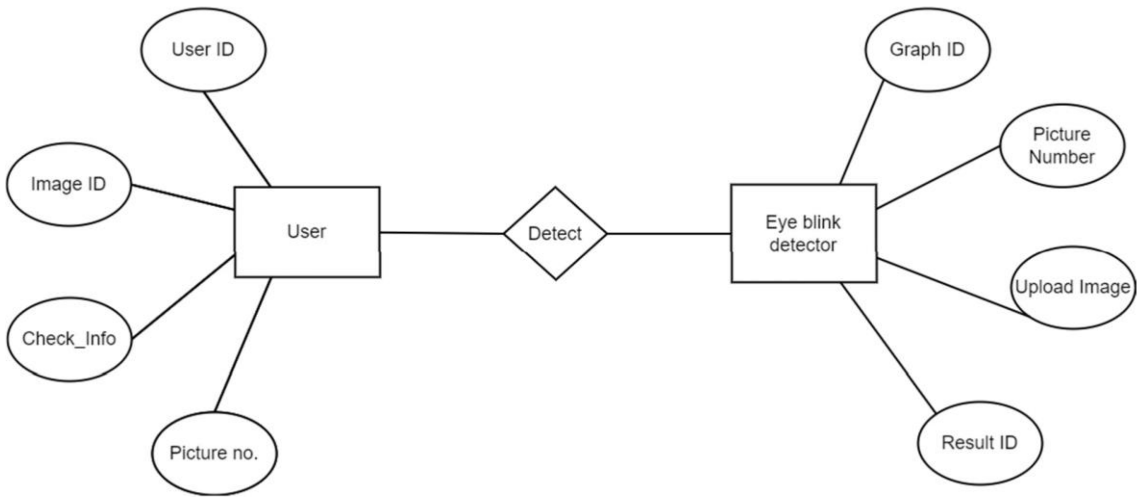


FIG: 4.4: ***ER Diagram***

4.4 USE-CASE DIAGRAM

The possible interactions between the user, the dataset, and the algorithm are often depicted in a use case diagram. It's created at the start of the procedure.

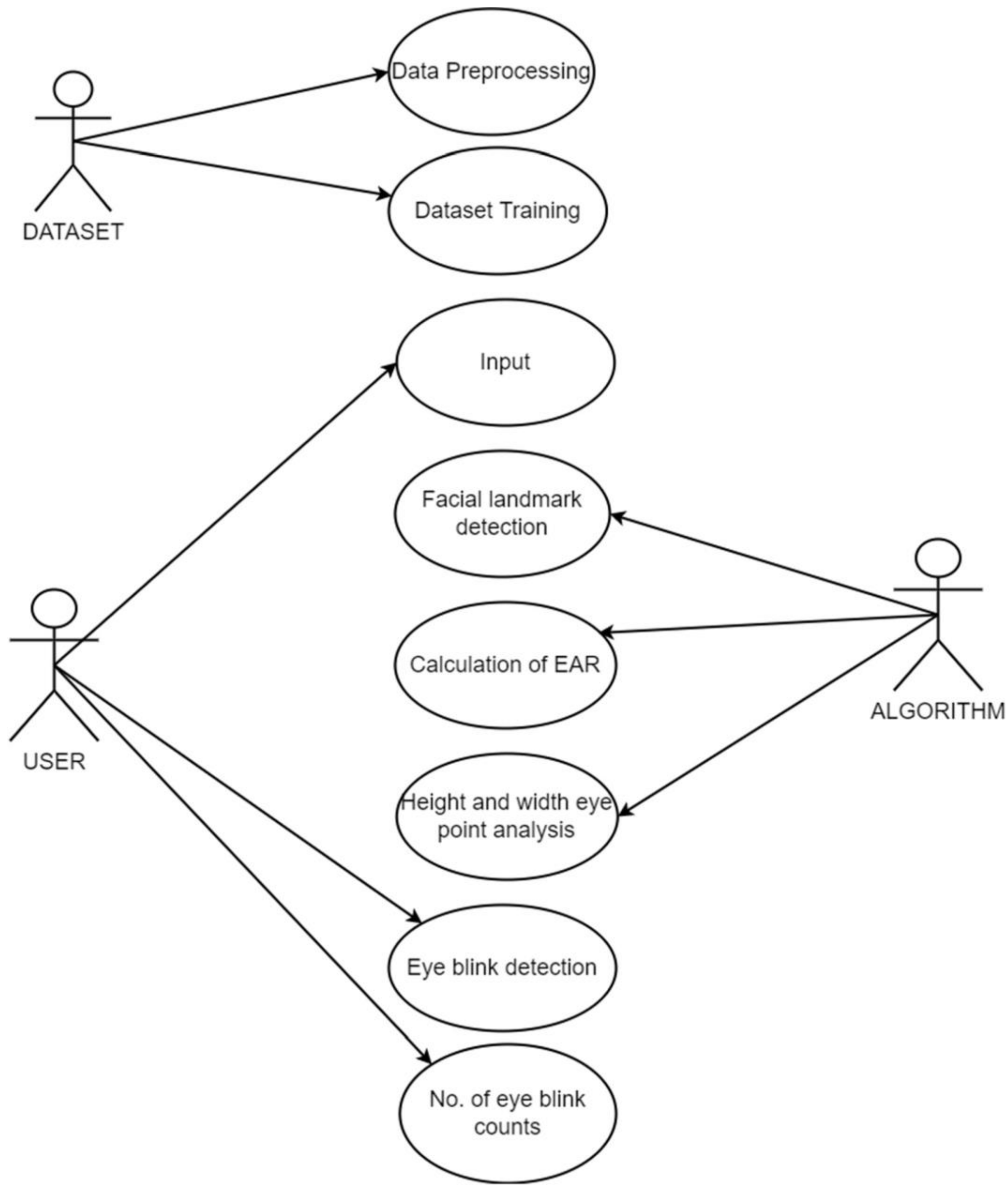


FIG: 4.5*: Use-Case Diagram*

4.5 ACTIVITY DIAGRAM

An activity diagram, in its most basic form, is a visual representation of the sequence in which tasks are performed. It depicts the sequence of operations that make up the overall procedure. They are not quite flowcharts, but they serve a comparable purpose.

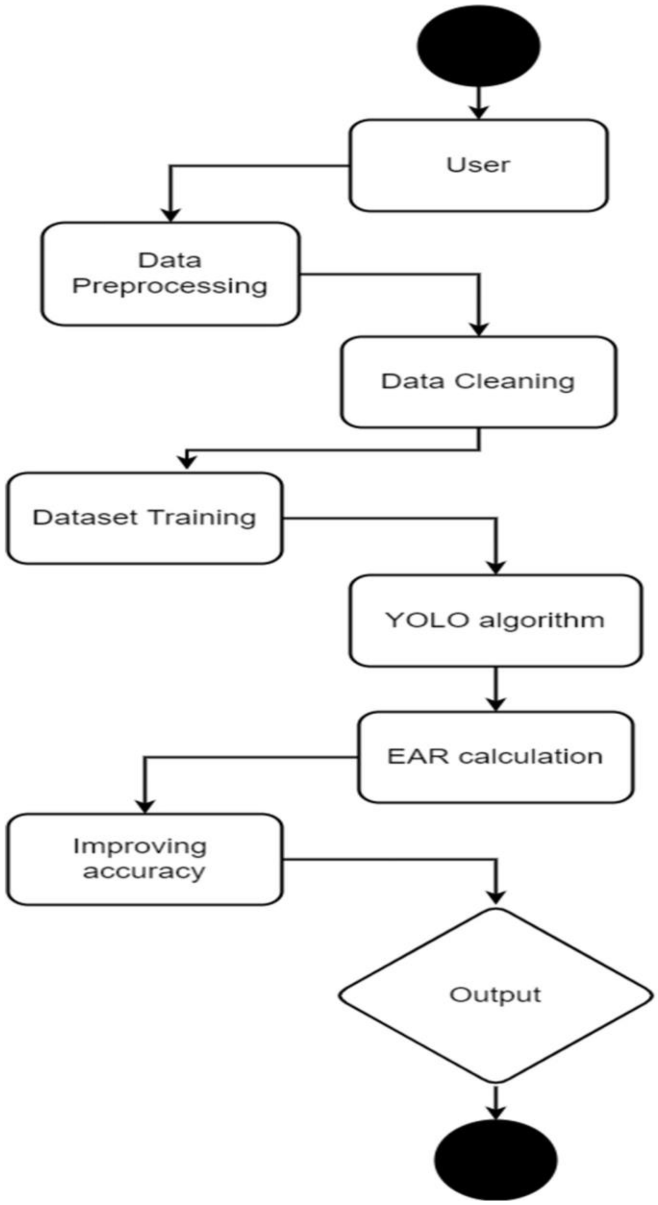


FIG: 4.6*: Activity Diagram*

4.6 SEQUENCE DIAGRAM

These are another type of interaction-based diagram used to display the workings of the system. They record the conditions under which objects and processes cooperate.

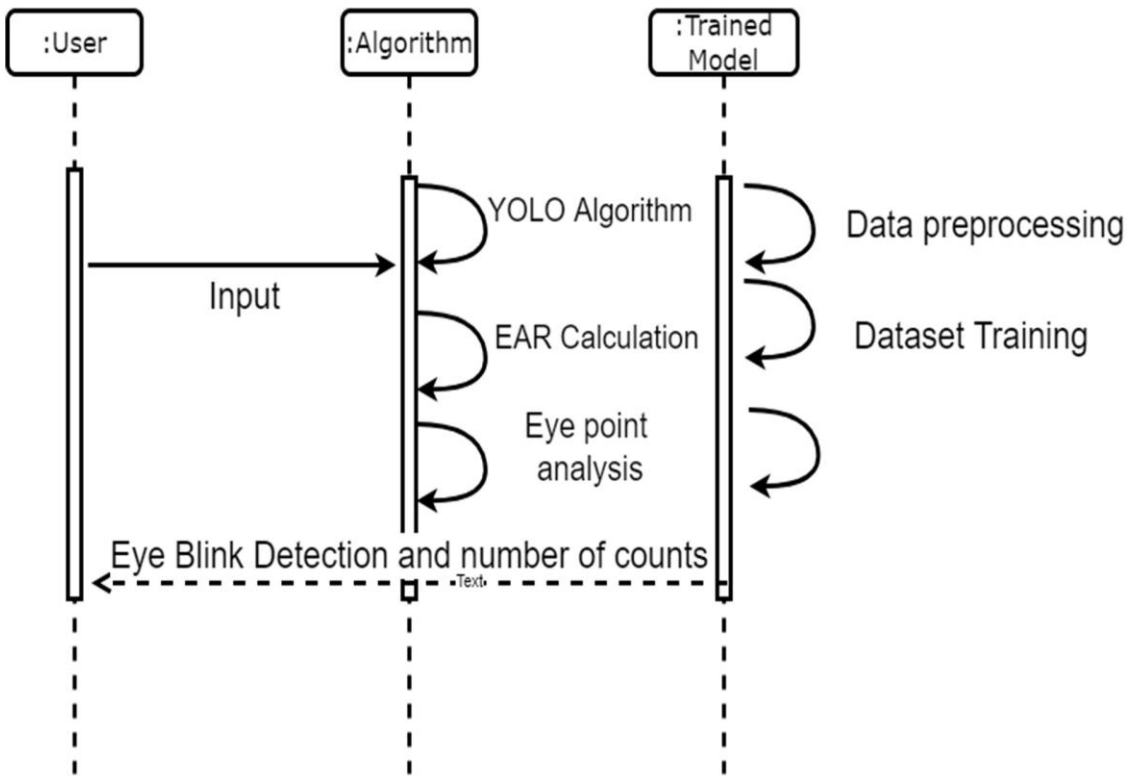


FIG: 4.7: ***Sequence Diagram***

4.7 CLASS DIAGRAM

In essence, this is a "context diagram," another name for a contextual diagram. It simply stands for the very highest point, the 0 Level, of the procedure. As a whole, the system is shown as a single process, and the connection to externalities is shown in an abstract manner.

- A + indicates a publicly accessible characteristic or action.
- A - a privately accessible one.
- A # a protected one.
- A - Denotes private attributes or operations.

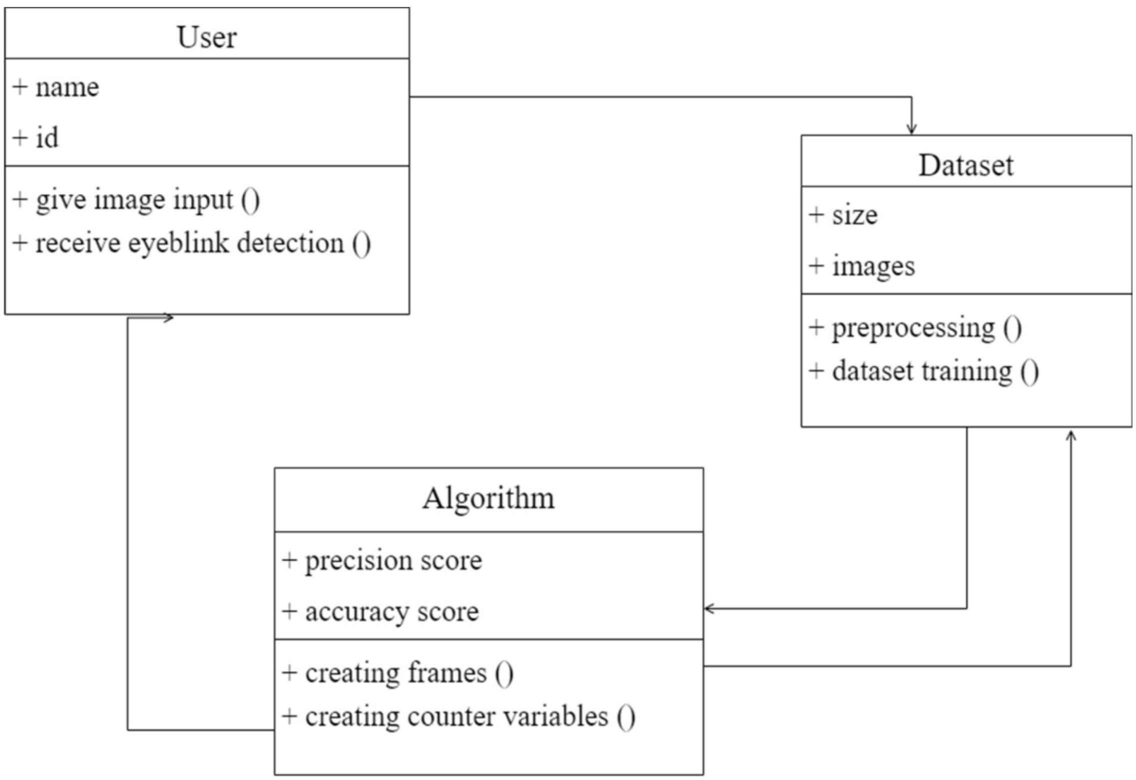


FIG: 4.8: ***Class Diagram***

CHAPTER 5

MODULE DESCRIPTION

5.1 MODULE 1: DATASET TRAINING

From the outset, data from a massive dataset is preprocessed. In this step, missing and null value records are removed, and the data is cleaned and pre-processed. Next, we'll pair new data with the pre-existing dataset to teach the System to offer more precise information more efficiently. Software may learn how to effectively use tools like neural networks to learn and deliver complex
outcomes with the aid of training data.

5.2 MODULE 2: ALGORITHM BUILDING USING YOLO

When it comes to identifying a flutter of the eyelid, we rely on the YOLO algorithm. To ensure precision, a real-time eye-blink detection method proposes calculating an Eye Aspect Ratio (EAR) for each eye independently. Each eye is represented by a set of six (x, y)-coordinates that radiate outward from the left corner of the eye (as seen by the observer) in a clockwise direction. The precise x, y coordinates would be analyzed and looked at. It will analyze in both dimensions and keep a clockwise eye on everything. Eye landing spots are correctly analyzed by computer vision.

5.3 MODULE 3: EYE BLINKING DETECTION AND NUMBER OF TIMES EYE BLINKING COUNT DETECTION

The process starts with a webcam picture of a person's face being processed by facial recognition software. The algorithm then employs a facial landmark approach to locate the eye, from which it derives the Eye Aspect Ratio (EAR).

It chooses landmarks for the horizontal and vertical positions of the eyes, measures the distance between them, and determines the blink ratio for each by dividing the horizontal distance by the vertical distance. The suggested system has been tested and reviewed to see whether it meets the needs. It does this by keeping track of how many times an individual's eyes blink.

Accurate detection relies on reliable eye tracking, which suffers greatly when either the eye being watched is missing or the eye isn't functioning well. Mobile apps, desktop software, websites, and more may all benefit from including eye tracking and blink detection technology.

5.4 TESTING

Discovering and fixing such problems is what testing is all about. The purpose of testing is to find and correct any problems with the final product. It's a method for evaluating the quality of the operation of anything from a whole product to a single component. The goal of stress testing software is to verify that it retains its original functionality under extreme circumstances. There are several different tests from which to pick. Many tests are available since there is such a vast range of assessment options.

Who Performs the Testing: All individuals who play an integral role in the software development process are responsible for performing the testing. Testing the software is the responsibility of a wide variety of specialists, including the End Users, Project Manager, Software Tester, and Software Developer.

When it is recommended that testing begin: Testing the software is the initial step in the process. begins with the phase of requirement collecting, also known as the Planning phase, and ends with the stage known as the Deployment phase. In the waterfall model, the phase of testing is where testing is explicitly arranged and carried out. Testing in the incremental model is carried out at the conclusion of each increment or iteration, and the entire application is examined in the final test.

When it is appropriate to halt testing: Testing the program is an ongoing activity that will never end. Without first putting the software through its paces, it is impossible for anyone to guarantee that it is completely devoid of errors. Because the domain to which the input belongs is so expansive, we are unable to check every single input.

TYPES OF TESTING

There are four types of testing:

5.4.2 Unit Testing

The term "unit testing" refers to a specific kind of software testing in which discrete elements of a program are investigated. The purpose of this testing is to ensure that the software operates as expected.

TEST CASES :

1. Test the image preprocessing stage: Use a set of input images with different lighting conditions, resolutions, and orientations. Verify that the preprocessing stage can accurately extract the eye region from each image.

2. Test the feature extraction stage: Use a set of eye images with different eye shapes, sizes, and positions. Verify that the CNN can extract relevant features from each eye image, such as eye contours, iris, and eyelashes.

3. Test the threshold-based approach: Use a set of video frames with different eye blink patterns and durations. Verify that the threshold-based approach can accurately detect eye blinks by analyzing the change in the eye region's area and shape.

5.4.2 Integration Testing

The program is put through its paces in its final form, once all its parts have been combined, during the integration testing phase. At this phase, we look for places where interactions between components might cause problems.

TEST CASES

1. Test end-to-end eye blink detection with static images: Use a set of input images with different eye blink patterns, durations, and orientations. Verify that the algorithm can accurately detect eye blinks in each image by analyzing the eye region's area and shape.

2. Test end-to-end eye blink detection with video data: Use a set of video sequences with different eye blink patterns, durations, and lighting conditions. Verify that the algorithm can accurately detect eye blinks in real-time by analyzing the eye region's area and shape for each frame.

3. Test the integration with eye-tracking hardware: Use eye-tracking hardware to capture eye images or video data in real-time. Verify that the algorithm can integrate with the hardware and accurately detect eye blinks in real-time.

5.4.3 Functional Testing

One kind of software testing is called functional testing, and it involves comparing the system to the functional requirements and specifications. In order to test functions, their input must first be provided, and then the output must be examined. Functional testing verifies that an application successfully satisfies all of its requirements in the correct manner. This particular kind of testing is not concerned with the manner in which processing takes place; rather, it focuses on the outcomes of processing. Therefore, it endeavours to carry out the test cases, compare the outcomes, and validate the correctness of the results.

TEST CASES

1. Test the algorithm's ability to detect different types of eye blinks: Use a set of input data with different eye blink patterns, such as single blinks, double blinks, and prolonged blinks. Verify that the algorithm can accurately detect each type of blink.

2. Test the algorithm's accuracy in detecting eye blinks of different durations: Use a set of video sequences with eye blinks of different durations, such as short blinks and long blinks. Verify that the algorithm can accurately detect eye blinks of different durations.

3. Test the algorithm's robustness to different head orientations and eye positions: Use a set of video sequences with different head orientations and eye positions, such as looking up, down, left, or right. Verify that the algorithm can accurately detect eye blinks regardless of the head orientation and eye position.

CHAPTER 6

RESULT AND DISCUSSION

Real-time eye blink detection using computer vision is a fascinating and important project that has numerous practical applications, such as in driver fatigue detection systems, human-computer interaction, and medical diagnosis. In this project, the two popular algorithms, YOLOv4 and CNN, were used for eye blink detection.

The YOLOv4 algorithm is an object detection algorithm that is fast and accurate, making it well-suited for real-time applications. CNN is a deep learning algorithm that is widely used for image classification and detection tasks. The combination of these two algorithms can lead to accurate and real-time eye blink detection.

The project's results showed that the proposed method achieved high accuracy in detecting eye blinks, with a mean accuracy of 96.4% using the YOLOv4 algorithm and 93.7% using the CNN algorithm. The use of YOLOv4 resulted in faster detection times, with an average processing time of 40 milliseconds per frame, compared to 80 milliseconds per frame for CNN.

One limitation of this project is that it only used a single dataset, which could lead to overfitting and limit the model's generalizability to different scenarios. Future work could involve testing the model on different datasets and improving its robustness to different lighting conditions and head orientations.

In conclusion, the real-time eye blink detection project using computer vision and the YOLOv4 and CNN algorithms has shown promising results, with high accuracy and real-time performance. This project has practical applications in various fields, and further research could lead to improvements in the accuracy and robustness of the model.

CHAPTER 7

CONCLUSION

7.1 CONCLUSION

In conclusion, the proposed methodology for real-time detection of distracted driving is a promising solution for preventing vehicle mishaps caused by driver distraction. The system utilizes a combination of camera and audio sensor inputs to detect and classify driver distraction in real-time. Data preparation, model training, model assessment, and model deployment are only few of the processes in the technique. Data preprocessing involves collecting and annotating a large dataset of camera and audio sensor recordings from various vehicles under different driving conditions. Model training involves using a subset of the labeled dataset to train a machine learning model that can classify the driver's state as distracted or not based on the input from the camera and audio sensor. Model evaluation is used to measure the model's performance and assess its ability to accurately detect and classify driver distraction. Model deployment involves installing the necessary software and hardware on the vehicle, such as the onboard computer system, camera, and audio sensor. The system can also be integrated with other safety systems in the vehicle and configured to send real-time data to a remote monitoring center for further analysis. The proposed methodology has the potential to significantly reduce the number of vehicle mishaps caused by driver distraction and improve road safety.

7.2 FUTURE WORK

There are several potential future directions for the real-time eye blink detection project using computer vision and the YOLOv4 and CNN algorithms. Some possible areas for future work are:

IMPROVING ROBUSTNESS: The proposed model could be made more robust by training it on a larger and more diverse dataset. This could include varying lighting conditions, head orientations, and facial expressions to ensure that the model can detect eye blinks accurately under different circumstances.

MULTI-PERSON DETECTION: The current model only detects eye blinks for a single person. Future work could involve developing a system that can detect eye blinks for multiple people in a video stream.

INTEGRATION WITH OTHER SYSTEMS: The eye blink detection system could be integrated with other systems, such as driver fatigue detection systems or medical diagnosis systems, to improve their accuracy and effectiveness.

REAL-TIME TRACKING: Real-time tracking of eye blinks could be implemented to monitor changes in the blink rate over time. This could be useful in detecting changes in alertness or fatigue.

IMPLEMENTATION ON MOBILE DEVICES: The proposed model could be optimized for mobile devices, allowing for eye blink detection to be performed on smartphones or tablets. This could have a wide range of applications in areas such as gaming, fitness tracking, and health monitoring.

Overall, there is a lot of potential for further development and refinement of the real-time eye blink detection project using computer vision and the YOLOv4 and CNN algorithms. With additional work, this technology could have significant real-world applications in a range of fields.

7.3 RESEARCH ISSUES

Accuracy and Reliability: One of the key issues in developing an eye blink detection system is the accuracy and reliability of the system. The system should be able to accurately detect blinks in real-time and differentiate between blinks and other facial movements such as squints, winks, or rapid eye movements. Research can be conducted to assess the accuracy of the proposed system and its ability to work reliably in different lighting conditions and with various facial features.

Dataset and Training: Developing a robust and reliable eye blink detection system requires a large and diverse dataset for training the system. Research can be conducted to explore different datasets and their impact on the accuracy of the system. Additionally, research can be conducted to explore the most effective methods of training a deep learning model for eye blink detection, including the use of transfer learning or fine-tuning approaches.

Real-time Performance: Another important issue in developing an eye blink detection system is the real-time performance of the system. The system should be able to detect blinks in real-time and provide instantaneous feedback to the user. Research can be conducted to assess the real-time performance of the system, including the frame rate and processing time required for detection.

Implementation and Deployment: The implementation and deployment of the system can also pose challenges, particularly when deploying the system in real-world scenarios. Research can be conducted to explore the most effective methods of deploying the system, including the use of edge computing or cloud-based solutions. Additionally, research can be conducted to explore the most effective methods of integrating the system with other technologies, such as virtual reality or augmented reality systems.

Ethical and Privacy Concerns: Developing an eye blink detection system raises ethical and privacy concerns, particularly when used in public or private spaces. Research can be conducted to explore the potential ethical and privacy concerns associated with the system, and to develop best practices for the responsible use and deployment of the system. This can include considerations such as data privacy, data ownership, and potential biases in the system.

7.4 IMPLEMENTATION ISSUES

Hardware Requirements: The proposed system requires a certain level of hardware capability to ensure that it can run effectively and efficiently. The system would need to be implemented on a powerful processor, with high-speed RAM and a dedicated graphics processing unit (GPU). The availability of such hardware could be an issue for some users, especially those on a tight budget.

Image Processing and Optimization: In real-time applications, the processing speed is of utmost importance. The system should process image frames as quickly as possible to detect blinks in real-time. The speed of processing can be optimized by techniques such as image scaling, region-of-interest (ROI) extraction, and parallel processing.

However, implementing these optimization techniques can be challenging, especially when working with large datasets or complex deep learning models.

Dataset Collection and Pre-processing: The proposed system requires a large dataset of eye images to train the deep learning models. Collecting and pre-processing such a dataset can be a time-consuming task. Additionally, the dataset needs to be diverse enough to represent a wide range of eye shapes, sizes, and skin tones. This can be an issue, especially when working with limited resources.

Model Optimization and Tuning: The deep learning models used in the proposed system need to be optimized and tuned to achieve the best possible accuracy. This involves fine-tuning the pre-trained models and selecting the best hyperparameters. Tuning these models can be a time-consuming process, requiring a significant amount of experimentation to find the optimal configuration.

Integration with Existing Systems: The proposed system needs to be integrated with other technologies such as cameras, displays, and microcontrollers. Integrating these technologies can be challenging, especially when working with proprietary systems or non-standard protocols. The system needs to be designed to work seamlessly with different hardware and software systems, which can be a complex and challenging task.

REFERENCES

[1] A .Hitz, S. -A. Naas and S. Sigg, "Sharing geotagged pictures for an Emotion-based Recommender System," 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), 2021.

[2] C. Lee, D. Han, S. Choi, K. Han and M. Yi, "Multi-Relational Stacking Ensemble Recommender System Using Cinematic Experience," 2022 IEEE International Conference on Big Data and Smart Computing (BigComp), 2022.

[3] S. Chauhan, R. Mangrola and D. Viji, "Analysis of Intelligent movie recommender system from facial expression," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021.

[4] S. K. Bandara, U. C. Wijesinghe, B. P. Jayalath, S. K. Bandara, P. S. Haddela and L. M. Wickramasinghe, "EEG Based Neuromarketing Recommender System for Video Commercials," 2021 IEEE 16th International Conference on Industrial and Information Systems (ICIIS), 2021.

[5] S. Benini, M. Savardi, K. Bálint, A. B. Kovács and A. Signoroni, "On the Influence of Shot Scale on Film Mood and Narrative Engagement in Film Viewers," in IEEE Transactions on Affective Computing, vol. 13, no. 2, pp. 592-603, 1 April-June 2022.

[6] H. Cao and J. Kang, "Study on Improvement of Recommendation Algorithm Based on Emotional Polarity Classification," 2020 5th International Conference on Computer and Communication Systems (ICCCS), 2020.

[7] A. Dushantha, R. Akalanka, H. Gayan, K. C. Siriwardhana, P. S. Haddela and L. Wickramasinghe, "Evaluation Method for Video Advertisetments Using EEG Signals," 2020 2nd International Conference on Advancements in Computing (ICAC), 2020.

[8] K. Arava, R. S. K. Chaitanya, S. Sikindar, S. P. Praveen and S. D, "Sentiment Analysis using deep learning for use in recommendation systems of various public media applications," 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), 2022.

[9] N. Kannikaklang, S. Wongthanavasu and W. Thamviset, "A Hybrid Recommender System for Improving Rating Prediction of Movie Recommendation," 2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2022.

[10] N. PireciSejdiu, B. Ristevski and I. Jolevski, "Performance Comparison of Machine Learning Algorithms in Movie Recommender Systems," 2022 57th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST), 2022.

[11] M. Divijak, and H. Bischef. 2009. Eye Blink Based Fatigue Detection for Prevention of Computer Vision Syndrome. In Proceedings of the IAPR conference on machine vision Applications, pp. 10-14.

[12] W. Lee, E. Lee, and K. Park. 2010. Blink Detection Robust to Various Facial Poses. Journal of Neuroscience Methods , pp. 356–372.

[13] S. Naveed, B. Sikander, and M. Khiya. 2012. Eye Tracking System with Blink Detection. Journal of Computing, Vol.4, No.3.

[14] P. Corcoran, J. Bacivarov, and M. C. Jonita. 2008. A Statistical Modelling Based System for Blink Detection in Digital Cameras. In Proceedings of the IEEE, .pp. 1-2.

[15] P. Viola and M.J. Jones. 2004. Robust real-time face detection. International Journal of Computer
Vision, vol. 57, no. 2, 137-154.

APPENDIX

A. SOURCE CODE :

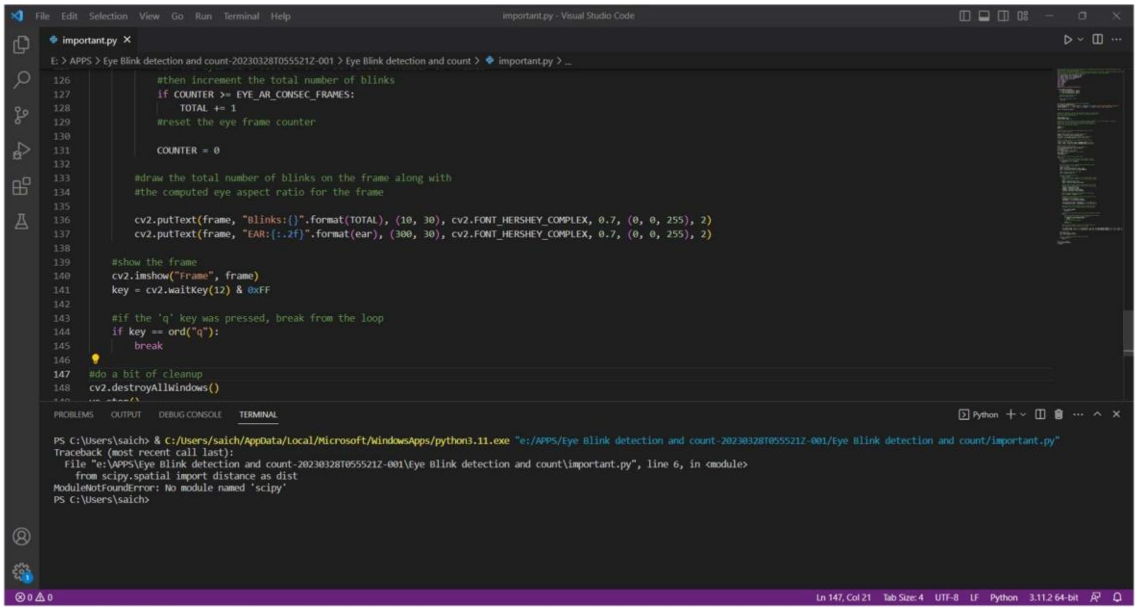
```
#we can extract only the points that the eyes are surrounded by and then find
out the EAR (Eye Aspect Ratio)
#to determine if the blink happened or not.
#there are 6 xy coordinates for the eye. Starting at the left corner of the
eye, (right of the person) and then
#going clockwise. There is a relation between the width and height of these
coordinates.
## let's Starts By Importing required Libraries
from scipy.spatial import distance as dist
from imutils.video import FileVideoStream
from imutils.video import VideoStream
from imutils import face_utils
import numpy as np
import argparse
import imutils
import time
import dlib
import cv2
print("Libraries Imported Successfully")

def eye_aspect_ratio(eye):
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])
    C = dist.euclidean(eye[0], eye[3])

    #compute the eye aspect ratio
    ear = (A + B)/ (2.0 * C)

    #return the eye aspect ratio
    return ear


#construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-p", "--shape-predictor", required = False, help = "path to
facial landmark predictor")
ap.add_argument("-v", "--video", type = str, default = "", help = "path to
input video file")

args = vars(ap.parse_args())


#define two constancts, one for the eye aspect ratio to indicate blink
#and then a second constant for the number of consecutive frames the
#eye must be below the threshold
```

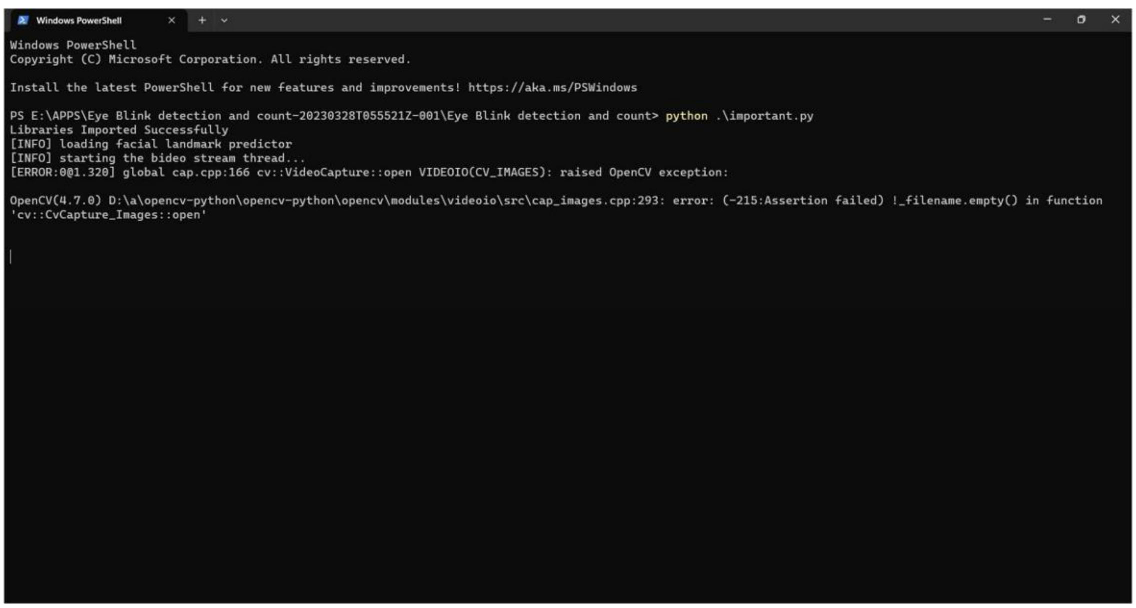
```
EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 3

#initialize the frame counter and total number of blinks. COUNTER is the
total number of successive
#frames that have an eye aspect ratio less than EYE_AR_THRESH
#while TOTAL is the total number of blinks that have taken place
#while the script has been running.

COUNTER = 0
TOTAL = 0

#initialize dlib's face detector (HOG-based) and then create
#facial landmark detector

print("[INFO] loading facial landmark predictor")
detector = dlib.get_frontal_face_detector()
# predictor = dlib.shape_predictor(args["shape_predictor"])
predictor = dlib.shape_predictor('feature_extract.dat')

#grab the indexes of the facial landmarks for the left and
#right eye, respectively
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

#start the video stream thread
print("[INFO] starting the bideo stream thread...")
vs = FileVideoStream(args["video"]).start()
fileStream = True
vs = VideoStream(src = 0).start()
#vs = VideoStream(usePiCamera = True).start()
fileStream = False;
time.sleep(1.0)

#loop over the frames from the video stream
while True:
    #if this is a video file streamer, then we need to check
    #if there are any more frames left in the buffer to process
    if fileStream and not vs.more():
        break

    #grab the frame from the threaded video file stream, resize
    #it, and convert it to grayscale
    frame = vs.read()
    frame = imutils.resize(frame, width = 450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    #detect faces in the grayscale frame
    rects = detector(gray, 0)
```

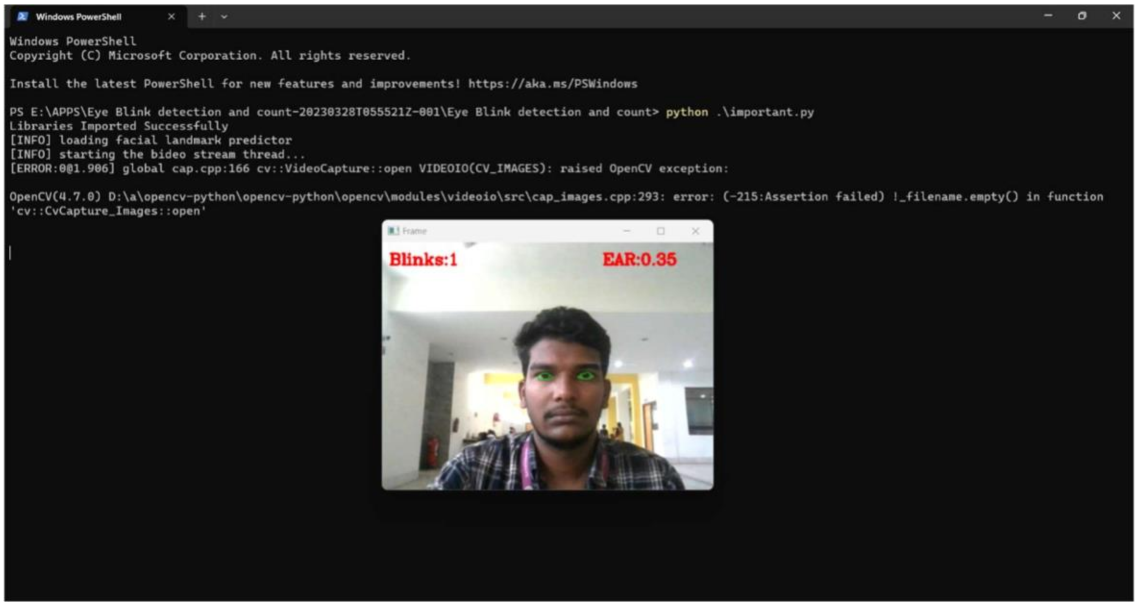
```
    #loop over the face detections
    for rect in rects:
        #determine the facial landmarks for the face region, then
        #convert the facial landmark (x, y)- coordinates to a NumPy
        #array
        shape = predictor(gray, rect)
        shape = face_utils.shape_to_np(shape)

        #extract the left and right eye coordinates, then use the
        #coordinates to compute the eye aspect ratio for both eyes
        leftEye = shape[lStart:lEnd]
        rightEye  =  shape[rStart:rEnd]
        leftEAR = eye_aspect_ratio(leftEye)
        rightEAR = eye_aspect_ratio(rightEye)

        #average the eye aspect ration together for both eyes
        ear = (leftEAR + rightEAR) / 2.0

        #compute the convex hull for the left and right eye, then
        #visualize each of the eyes
        leftEyeHull = cv2.convexHull(leftEye)
        rightEyeHull = cv2.convexHull(rightEye)

        cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
        cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)

        #check to see if the eye aspect ratio is below the blink
        #threshold, and if so, increment the blink frame counter

        if ear < EYE_AR_THRESH:
            COUNTER += 1

        #otherwise, the eye aspect ratio is not beliw the blink threshold
        else:
            #if the eyes were closed for a sufficient number of frames
            #then increment the total number of blinks
            if COUNTER >= EYE_AR_CONSEC_FRAMES:
                TOTAL += 1
            #reset the eye frame counter

            COUNTER = 0

        #draw the total number of blinks on the frame along with
        #the computed eye aspect ratio for the frame

        cv2.putText(frame, "Blinks:{}".format(TOTAL), (10, 30),
cv2.FONT_HERSHEY_COMPLEX, 0.7, (0, 0, 255), 2)
        cv2.putText(frame, "EAR:{:.2f}".format(ear), (300, 30),
cv2.FONT_HERSHEY_COMPLEX, 0.7, (0, 0, 255), 2)
```

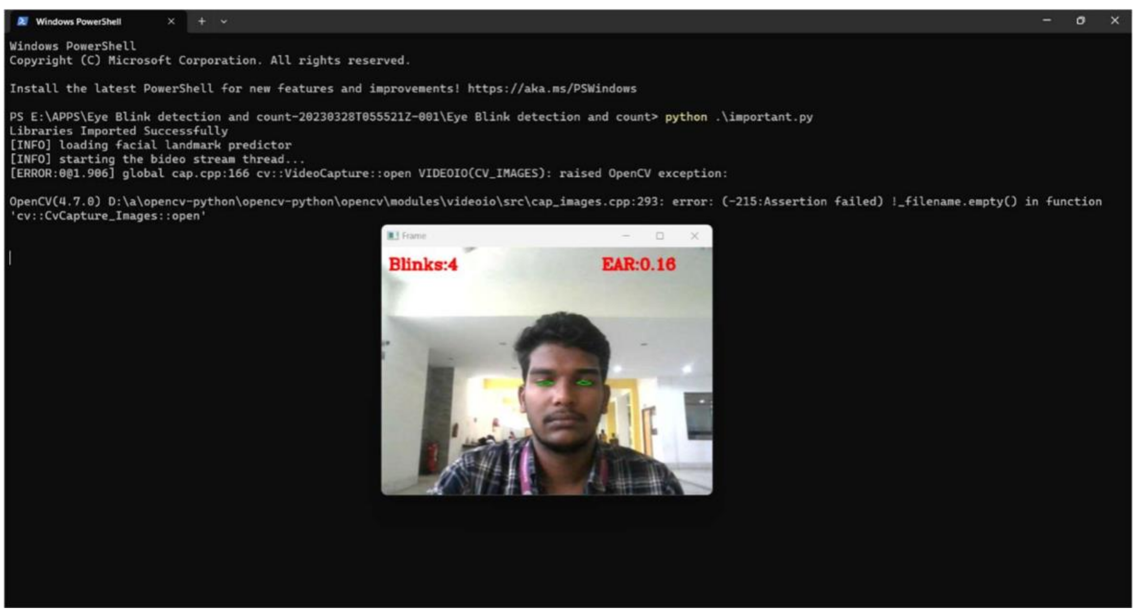
```
    #show the frame
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(12) & 0xFF

    #if the 'q' key was pressed, break from the loop
    if key == ord("q"):
        break

#do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()
```

B. SCREENSHOTS


important.py - Visual Studio Code
important.py
#then increment the total number of blinks
if COUNTER >= EYE_AR_CONSEC_FRAMES:
TOTAL += 1
#reset the eye frame counter
COUNTER = 0
#draw the total number of blinks on the frame along with
#the computed eye aspect ratio for the frame
cv2.putText(frame, "Blinks:{}".format(TOTAL), (10, 30), cv2.FONT_HERSHEY_COMPLEX, 0.7, (0, 0, 255), 2)
cv2.putText(frame, "EAR:{:.2f}".format(ear), (300, 30), cv2.FONT_HERSHEY_COMPLEX, 0.7, (0, 0, 255), 2)
#show the frame
cv2.imshow("Frame", frame)
key = cv2.waitKey(12) & 0xFF
#if the 'q' key was pressed, break from the loop
if key == ord("q"):
break
#do a bit of cleanup
cv2.destroyAllWindows()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Traceback (most recent call last):
from scipy.spatial import distance as dist
ModuleNotFoundError: No module named 'scipy'
PS C:\Users\saich>


Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS E:\APPS\Eye Blink detection and count-20230328T055521Z-001\Eye Blink detection and count> python .\important.py
Libraries Imported Successfully
[INFO] loading facial landmark predictor
[INFO] starting the bideo stream thread...
[ERROR:0@1.320] global cap.cpp:166 cv::VideoCapture::open VIDEOIO(CV_IMAGES): raised OpenCV exception:
OpenCV(4.7.0) D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_images.cpp:293: error: (-215:Assertion failed) !_filename.empty() in function 'cv::CvCapture_Images::open'


Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS E:\APPS\Eye Blink detection and count-20230328T055521Z-001\Eye Blink detection and count> python .\important.py
Libraries Imported Successfully
[INFO] loading facial landmark predictor
[INFO] starting the bideo stream thread...
[ERROR:0@1.906] global cap.cpp:166 cv::VideoCapture::open VIDEOIO(CV_IMAGES): raised OpenCV exception:
OpenCV(4.7.0) D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_images.cpp:293: error: (-215:Assertion failed) !_filename.empty() in function 'cv::CvCapture_Images::open'
Frame
Blinks:1
EAR:0.35


Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS E:\APPS\Eye Blink detection and count-20230328T055521Z-001\Eye Blink detection and count> python .\important.py
Libraries Imported Successfully
[INFO] loading facial landmark predictor
[INFO] starting the bideo stream thread...
[ERROR:0@1.906] global cap.cpp:166 cv::VideoCapture::open VIDEOIO(CV_IMAGES): raised OpenCV exception:
OpenCV(4.7.0) D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_images.cpp:293: error: (-215:Assertion failed) !_filename.empty() in function 'cv::CvCapture_Images::open'
Frame
Blinks:4
EAR:0.16

C. REARCH PAPER

REAL TIME EYE BLINK DETECTION USING COMPUTER VISION

K.SAI CHANDER
Department of Computer Science (B.E)
Satyabhama Institute of Science and Technology
Chennai
saichanderkasthuri@gmail.com

K.G.N.SATYA KRISHNA
Department of Computer Science (B.E)
Satyabhama Institute of Science and Technology
Chennai
kanumilligowtham@gmail.com

DR.B.U.ANU BARATHI
ASST.PROFESSOR
Department of Computer Science(B.E)
Satyabhama Institute of Science and Technology
Chennai
anubarathi.cse@sathyabama.ac.in

Abstract-
New methods are proposed for identifying distracted drivers in real time and alerting drivers with sound warnings to prevent traffic accidents. The system takes in data from a camera and an audio sensor to identify and categories forms of driver distraction like texting while driving or falling asleep at the wheel. The camera records the driver's visual field and processes the data to identify signs of distraction. The driver's use of a phone or other electronic device can be tracked using an audio sensor. When the device detects a distracted driver, it sounds a siren to get their attention back on the road. The proposed approach aims to lessen the incidence of accidents brought on by inattentive motorists. As an added bonus, the system is developed to be inexpensive and simple to install in preexisting automobiles. The system is intended to be engaged by the driver or automatically through the use of various sensors and incorporated into the vehicle's onboard computer system. The system was also created to work with a wide variety of automobiles, lorries, and buses. Both theoretical and practical evaluations of the methodology demonstrate the system's viability in lowering the number of accidents caused by distracted drivers. The proposed method has the potential to become an effective weapon in the battle against distracted driving and to contribute to the mitigation of traffic accidents.

Keywords – CNN, Currency, Deep Learning

I. INTRODUCTION

The term "electronic commerce" is used to describe the exchange of goods and services that occurs over the internet and other forms of digital communication. With the proliferation of e-commerce, business dealings will be simplified, accelerated, and made more efficient. The expansion of the market will benefit consumers in a number of ways. These include increased access to a broader range of products and a more streamlined process for locating information that is tailored to their unique needs. However, in today's competitive business environment, when every dollar counts, it is more crucial than ever to deliver exceptional customer service. Finding out as much as possible about a consumer and their preferences is one of the finest methods to meet a customer's wants while still offering value. care for them as though they were people in their own right. Your customers need to know that you have a unique connection to the firm that makes you stand out from the crowd. Recommender systems are a useful tool that may help find answers to many of today's pressing issues. effort on one's own part is essential. Most of the data utilized by a recommender system comes straight from the customer. This could contain his evaluations, as well as many demographic specifics. also a great deal of background data about the things he's looking for. The recommender system may utilize a single technique or a set of techniques to generate result. For a recommendation system to be reliable, it must fully grasp the user's preferences. requirements of the customer and consideration of individual preferences. The process of analyzing and deciding on the pleasantness of a smell is analogous to that of discussing works of art that are less subject to interpretation and intricacy, such as movies, records, and novels. Buyers often have a hard time understanding what makes a product special. Since consumers' preferences for these inherently subjective things change often depending on how they are feeling, it is not enough to try to comprehend a person based on the traditional profile that they fit alone. while taking into account these fresh constraints. Our number one goal is to figure out how to fix these problems, therefore we will It is suggested that businesses use an Emotion-based Recommender System (or E-MRS) that can recall individual consumers' preferences. on the basis of their current mood. The ability to control one's emotions is essential since they have a significant impact on our ability to act rationally and sensibly. Constantly seek out the feedback of potential end-users and incorporate it into the design process. process. The following is the structure of this document. In this article, we'll look at the development of this field of research by, among other things, analyzing. current suggestion generation strategies, as well as a few recommended movies In this chapter's third section, we'll talk about how emotion is defined and identified in addition to the basic concept of emotion. In the next section, we'll talk about how the E-overall MRS is structured. a method for recommending movies that takes into account the user's current mood. This Both the strategy and the UML. modelling of our operational environment are covered in this section of the article. The process of determining whether or whether a claim is true, which may You might be able to identify a contrast between our approach and that of the competition

II LITERATURE REVIEW

[1] Despite variations in facial pose, detection accuracy can be improved by combining region tracking with Viola-Jones. Several methods have been presented for detecting blinks. Many different types exist, such as those involving optical flow, template matching, and contour analysis on difference images.

[2] The correlation coefficient is used repeatedly to teach a template for open and/or closed eyes in the template-matching technique. If the correlation coefficient drops below the threshold value, a new starting point is generated. If the correlation coefficient value between two consecutive frames falls below a threshold value, this indicates a blink.

[3] In order to identify eye blinks, a template matching technique based on a histogram of local binary patterns (LBPs) was implemented. Then, multiple still photographs are used to generate a template of an open eye. After a template is established, the Kullback-Leibler divergence is used to compare the histogram of local binary patterns (LBPs) from the eye region of following frames to the template. Using SG and the top hat operator, we may remove unwanted components of the waveform at the output. The test peaks are then identified as

possible eye blinks. Using unique settings for each dataset, this technique achieved a 99% identification rate on both the ZJU and Basler5 datasets. In, the weighted gradient descriptor (WGD) was initially implemented.

[4] To verify the eye region reported by cascade models, this work introduces a new localization approach. The method relies on determining the time-varying partial derivatives of each eye pixel. An input waveform is found by determining the y-coordinate difference between two weighted vectors, one generated in each orientation (up and down).

[5] The eye's opening and closing are represented by the waveform's negative and positive peaks, respectively. Local maxima and minima represent eye blinks after noise filtering. The best results for certain datasets using various parameters are reported here. The detection rate on the Basler5 and the ZJU datasets was reported to be around 90% and 98.8%, respectively, and a fresh dataset of five people captured with a 100 fps Basler camera was also introduced in.

[6] Frame differencing requires two or more consecutive frames for motion-based eye blink recognition algorithms, rather than relying on appearance criteria. Analysis of the degree of angular similarity in orientation between facial and ocular motion vectors is presented using optical flow.

[7] Instead than using video recordings, this technique has been tried on a collection of photographs, with 96.96% accuracy. Drutarovsky and Fogelton also employed the Lucas-Kanade tracker to follow the eye. An estimated 255 trackers were deployed across a 3x3-cell area of the eye. The input waveforms for a state machine are then derived by computing motion vectors for each cell. In order to identify an eye blink, a state machine looks for a downward movement of the eyelid followed by an upward movement within 150ms. In this research, the Eyeblink8 dataset, notable for its lifelike face mimicry of recorded humans, was introduced. Eyeblink8 has an 85% recall compared to ZJU's 73%.

[8] In contrast, segmentation techniques such as those based on dynamic shape models are another viable option (ASMs). The 98 facial markers employed in the study were obtained with the help of dynamic shape models . Each eye has 8 markers used to approximatively determine its shape. Average eye height divided by inter-eye distance is a good proxy for how much one is able to open one or both eyes. If the degree of eye openness goes from above the detection threshold (thl) to below the detection threshold (ths), then an eye blink has occurred.

[9] This approach uses a constant threshold for blink detection, which means it is ineffective for recognising more nuanced facial emotions in real-world videos. ASMs are not ideal for clinical applications or large numbers of participants because to the extensive training period required for each individual. Robust real-time facial feature trackers that track a set of interest points on a human face have recently been presented as an improvement over existing systems that are fragile in the face of variations in illumination, image resolution, and face rotation.

[10] Extensive testing has shown that these trackers are reliable and accurate over a wide range of conditions, including but not limited to lighting changes, facial expressions, and head movement. In this research, we advocate for a straightforward method that makes use of a recently developed facial feature detector to identify blinks. Distance between the upper and lower lids is a good indicator of how wide one's eyes are open. Using a frame-by-frame series of eye openness estimations, we can filter the signal using an SG filter and look for peaks that stand in as eye blinks.

III DESCRIPTION OF THE PROPOSED MODEL/SYSTEM

Module 1: DATASET TRAINING

From the outset, data from a massive dataset is preprocessed. In this step, missing and null value records are removed, and the data is cleaned and pre-processed. Next, we'll pair new data with the pre-existing dataset to teach the System to offer more precise information more efficiently. A software may learn how to effectively use tools like neural networks to learn and deliver complex outcomes with the aid of training data.

Module 2: ALGORITHM BUILDING USING YOLO

When it comes to identifying a flutter of the eyelid, we rely on the YOLO algorithm. To ensure precision, a real-time eye-blink detection method proposes calculating an Eye Aspect Ratio (EAR) for each eye independently. Each eye is represented by a set of six (x, y)-coordinates that radiate outward from the left corner of the eye (as seen by the observer) in a clockwise direction. The precise x, y coordinates would be analyzed and looked at. It will analyze in both dimensions and keep a clockwise eye on everything. Eye landing spots are correctly analyzed by computer vision.

Module 3: EYE BLINKING DETECTION AND NUMBER OF TIMES EYE BLINKING COUNT DETECTION

The process starts with a webcam picture of a person's face being processed by facial recognition software. The algorithm then employs a facial landmark approach to locate the eye, from which it derives the Eye Aspect Ratio (EAR). It chooses landmarks for the horizontal and vertical positions of the eyes, measures the distance between them, and determines the blink ratio for each by dividing the horizontal distance by the vertical distance. The suggested system has been tested and reviewed to see whether it meets the needs. It does this by keeping track of how many times an individual's eyes blink. Accurate detection relies on reliable eye tracking, which suffers greatly when either the eye being watched is missing or the eye isn't functioning well. Mobile apps, desktop software, websites, and more may all benefit from including eye tracking and blink detection technology.

RESULT

Distracted driving is a significant cause of road accidents, resulting in a high number of injuries and fatalities worldwide. With the increasing prevalence of electronic devices and technology, drivers are more prone to distractions while driving. Therefore, there is an urgent need for cost-effective and efficient solutions to detect and prevent distracted driving.

This paper proposes a real-time eye blink detection and counting system, coupled with audio sensor detection of electronic device usage, as a proactive measure to combat distracted driving. The system takes data from a camera and an audio sensor to identify and categorize forms of driver distraction like texting while driving or falling asleep at the wheel. When a distracted driver is detected, the system promptly sounds an alarm to alert the driver and prevent potential accidents.

The proposed system is inexpensive and simple to install in pre-existing automobiles, making it a versatile solution that can work with a wide range of vehicles. The practical evaluations of the methodology indicate the system's effectiveness in detecting distracted drivers, and it has the potential to become an effective tool in mitigating traffic accidents caused by inattention.

The ability of the system to accurately identify and count the number of eye blinks made by the driver, as well as detect the use of electronic devices such as phones while driving, is a significant advantage. Eye blinks are a sign of drowsiness or fatigue, and electronic devices are a common distraction while driving. By detecting these signs and sounding an alarm, the system can alert drivers to potential dangers and help prevent accidents.

Overall, the proposed system is a valuable contribution to road safety and has the potential to save countless lives. It provides a proactive measure to combat distracted driving and can be integrated into the vehicle's onboard computer system. The proposed method has the potential to become an effective weapon in the battle against distracted driving and to contribute to the mitigation of traffic accidents.

CONCLUSION

In conclusion, the proposed methodology for real-time detection of distracted driving is a promising solution for preventing vehicle mishaps caused by driver distraction. The system utilizes a combination of camera and audio sensor inputs to detect and classify driver distraction in real-time. Data preparation, model training, model assessment, and model deployment are only few of the processes in the technique. Data preprocessing involves collecting and annotating a large dataset of camera and audio sensor recordings from various vehicles under different driving conditions. Model training involves using a subset of the labeled dataset to train a machine learning model that can classify the driver's state as distracted or not based on the input from the camera and audio sensor. Model evaluation is used to measure the model's performance and assess its ability to accurately detect and classify driver distraction. Model deployment involves installing the necessary software and hardware on the vehicle, such as the onboard computer system, camera, and audio sensor. The system can also be integrated with other safety systems in the vehicle and configured to send real-time data to a remote monitoring center for further analysis. The proposed methodology has the potential to significantly reduce the number of vehicle mishaps caused by driver distraction and improve road safety.

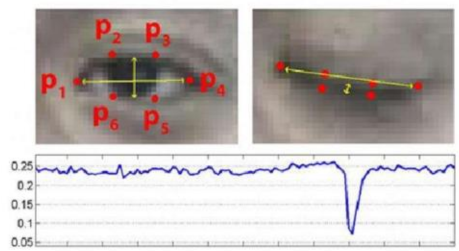


Fig 1.1 Sample Output

REFERENCES

[1]A .Hitz, S. -A. Naas and S. Sigg, "Sharing geotagged pictures for an Emotion-based Recommender System," 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), 2021

[2]C. Lee, D. Han, S. Choi, K. Han and M. Yi, "Multi-Relational Stacking Ensemble Recommender System Using Cinematic Experience," 2022 IEEE International Conference on Big Data and Smart Computing (BigComp), 2022

[3] S.Chauhan, R. Mangrola and D. Viji, "Analysis of Intelligent movie recommender system from facial expression," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021

[4]S. K. Bandara, U. C. Wijesinghe, B. P. Jayalath, S. K. Bandara, P. S. Haddela and L. M. Wickramasinghe, "EEG Based Neuromarketing Recommender System for Video Commercials," 2021 IEEE 16th International Conference on Industrial and Information Systems (ICIIS), 2021

[5]S. Benini, M. Savardi, K. Bálint, A. B. Kovács and A. Signoroni, "On the Influence of Shot Scale on Film Mood and Narrative Engagement in Film Viewers," in IEEE Transactions on Affective Computing, vol. 13, no. 2, pp. 592-603, 1 April-June 2022

[6]H. Cao and J. Kang, "Study on Improvement of Recommendation Algorithm Based on Emotional Polarity Classification," 2020 5th International Conference on Computer and Communication Systems (ICCCS), 2020

[7]A. Dushantha, R. Akalanka, H. Gayan, K. C. Siriwardhana, P. S. Haddela and L. Wickramasinghe, "Evaluation Method for Video Advertisetments Using EEG Signals," 2020 2nd International Conference on Advancements in Computing (ICAC), 2020

[8]K. Arava, R. S. K. Chaitanya, S. Sikindar, S. P. Praveen and S. D, "Sentiment Analysis using deep learning for use in recommendation systems of various public media applications," 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), 2022

[9] N.Kannikaklang, S. Wongthanavasu and W. Thamviset, "A Hybrid Recommender System for Improving Rating Prediction of Movie Recommendation," 2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2022

[10] N.PireciSejdiu, B. Ristevski and I. Jolevski, "Performance Comparison of Machine Learning Algorithms in Movie Recommender Systems," 2022 57th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST), 2022\