A Deep Learning Technique To Detect Drowsiness And Notifying through Mails and Alarm

A Project Report submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR.

In Partial Fulfillment of the Requirements for the Award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
BY

CH SAI LAKSHMI	**20121A3213**
K GAGANDEEP	**20121A3221**
G CHIDAMBARAM	**20121A3220**
K MAHESH	**20121A3229**

Under the Guidance of

Dr. P. Dhanalakshmi
Professor
Dept of CSE, SVEC

Department of Computer Science and Engineering
SREE VIDYANIKETHAN ENGINEERING COLLEGE
(Affiliated to JNTUA, Anantapuramu)
Sree Sainath Nagar, Tirupathi – 517 102

2020-2024

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University Anantapur)
Sree Sainath Nagar, A. Rangampet, Tirupati – 517 102, Chittoor Dist., A.P.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project Work entitled

A Deep Learning Technique To Detect Drowsiness And Notifying Through Mails and Alarm

is the bonafide work done by

CH SAI LAKSHMI	**20121A3213**
K GAGANDEEP	**20121A3221**
G CHIDAMBARAM	**20121A3220**
K MAHESH	**20121A3229**

In the Department of Computer Science and Engineering, Sree Vidyanikethan Engineering College, A. Rangampet. is affiliated to JNTUA, Anantapuramu in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering during 2020-2024.

This is work has been carried out under my guidance and supervision.

The results embodied in this Project report have not been submitted in any University or Organization for the award of any degree or diploma.

Internal Guide

Dr. P.Dhanalakshmi
Professor
Dept of CSE
Sree Vidyanikethan Engineering College
Tirupathi

Head

Dr. B. Narendra Kumar Rao
Prof & Head
Dept of CSE
Sree Vidyanikethan Engineering College
Tirupathi

INTERNAL EXAMINER

EXTERNAL EXAMINER

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION

To become a Centre of Excellence in Computer Science and Engineering by imparting high quality education through teaching, training and research.

MISSION

- The Department of Computer Science and Engineering is established to provide undergraduate and graduate education in the field of Computer Science and Engineering to students with diverse background in foundations of software and hardware through a broad curriculum and strongly focused on developing advanced knowledge to become future leaders.
- Create knowledge of advanced concepts, innovative technologies and develop research aptitude for contributing to the needs of industry and society.
- Develop professional and soft skills for improved knowledge and employability of students.
- Encourage students to engage in life-long learning to create awareness of the contemporary developments in computer science and engineering to become outstanding professionals.

- Develop attitude for ethical and social responsibilities in professional practice at regional, National and International levels.

Program Educational Objectives (PEO's)

After few years of graduation, the graduates of B.Tech. (CSE) will be:

1. Pursuing higher studies in Computer Science and Engineering and related disciplines.
2. Employed in reputed Computer and I.T organizations and Government or have established start-up companies.
3. Able to demonstrate effective communication, engage in teamwork, exhibit leadership skills, ethical attitude, and achieve professional advancement through continuing education.

Program Specific Outcomes (PSO's)

On successful completion of the Program, the graduates of B.Tech. (CSE) program will be able to:

PSO1: Use mathematical methodologies to model real-world problems, Employ modern tools and platforms for efficient design and development of computer-based systems.

PSO2: Apply adaptive algorithms and methodologies to develop intelligent systems for solving problems from interdisciplinary domains.

PSO3: Apply suitable models, tools, and techniques to perform data analytics for effective decision-making.

PSO4: Design and deploy networked systems using standards and principles, evaluate security measures for complex networks, and apply procedures and tools to solve networking issues.

Program Outcomes (PO's)

1. Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems (**Engineering knowledge**).

2. Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences (**Problem analysis**).

3. Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations (**Design/development of solutions**).

4. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions (**Conduct investigations of complex problems**).

5. Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations (**Modern tool usage**)

6. Apply to reason informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice (**The engineer and society**)

7. Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development (**Environment and sustainability**).

8. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice (**Ethics**).

9. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings (**Individual and team work**).

10. Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions (**Communication**).

11. Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments (**Project management and finance**).

12. Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change (**Life-long learning**).

Course Outcomes

CO1. Create/Design computer science engineering systems or processes to solve complex computer science engineering and allied problems using appropriate tools and techniques following relevant standards, codes, policies, regulations and latest developments.

CO2. Consider society, health, safety, environment, sustainability, economics and project management in solving complex computer science engineering and allied problems.

CO3. Perform individually or in a team besides communicating effectively in written, oral and graphical forms on computer science engineering systems or processes.

CO-PO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO3	PSO 4
CO1	3												3			
CO2		3												3		
CO3			3												3	
CO4				3											3	
CO5					3											3
CO6						3										
CO7							3									
CO8								3								
CO9									3							
CO10										3						
CO11											3					
CO12												3				

(Note: 3-High, 2-Medium, 1-Low)

DECLARATION

We hereby declare that this project report titled **A Deep Learning Technique To Detect Drowsiness And Notifying Through Mails and Alarm** is a genuine project work carried out by us, in the **B.Tech *(Computer Science and Engineering)*** degree course of **Jawaharlal Nehru Technological University Anantapur** and has not been submitted to any other course or University for the award of any degree by us.

Signature of the student

1.

2.

3.

4.

ACKNOWLEDGEMENT

We are extremely thankful to our beloved Chairman and founder **Dr. M. Mohan Babu** who took a keen interest to provide us with the infrastructural facilities for carrying out the project work.

We are highly indebted to **Dr. B. M. Satish**, Principal of the Sree Vidyanikethan Engineering College for his valuable support and guidance in all academic matters.

We are very much obliged to **Dr. B. Narendra Kumar Rao,** Professor & Head, Department of CSE, for providing us guidance and encouragement in the completion of this project.

We would like to express our indebtedness to the project coordinator, **Dr. Sunitha Gurram**, Professor, and the Department of CSE for her valuable guidance during the course of the project work.

We would like to express our deep sense of gratitude to **Dr. P. Dhanalakshmi**, Professor, Department of CSE, for the constant support and invaluable guidance provided for the successful completion of the project.

We are also thankful to all the faculty members of the CSE Department, who have cooperated in carrying out our project. We would like to thank our parents and friends who have extended their help and encouragement either directly or indirectly in the completion of our project work.

ABSTRACT

Drowsiness or Fatigue is the process of impairing an individual's mental and physical abilities. It can happen as a result of not getting enough sleep, extended periods of inactivity, or other factors like medication, alcohol, or clinical settings. The main consequences of being sleepy are the potential for fatalities and injuries. It distinguishes between awake and sleepy phases by analysing factors such as frame posture, eye movements, and facial expressions. Drowsiness can find using brainwave patterns, particularly alpha and theta frequencies. The current process includes OpenCV and deep learning model that helps in detecting face and eyes using 68 face land marks. If a person is drowsy then we will alert a person with the help of alarm and we will send mails to their family or friends that the person is drowsy and inform to kindly alert him, otherwise accidents may occur. The further development which we would like to do is to integrate a chatbot and that helps in keeping the person alert all the time.

The current model will have good accuracy and developments to get rid of accidents. The last purpose is to beautify avenue safety and mitigate injuries due to driving force drowsiness, contributing to the general nicely-being of society.

The ultimate goal is to enhance road safety and mitigate accidents caused by driver drowsiness, contributing to overall well-being of society.

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

1.1 Introduction:

Drowsiness, often referred to as sleepiness or somnolence, is a state characterized by a strong feeling of sleepiness and lethargy. It is a common occurrence in our daily lives and can be influenced by various factors such as inadequate sleep, long working hours, sleep disorders, medication, or certain medical conditions.

This state significantly impairs cognitive function, alertness, and reaction time, posing a serious threat to safety, especially when performing tasks that require focus and attention, such as driving or operating heavy machinery. According to the National Highway Traffic Safety Administration (NHTSA), drowsy driving was responsible for 91,000 crashes in 2017 alone, resulting in approximately 50,000 injuries and nearly 800 deaths. These statistics underscore the importance of addressing drowsiness to prevent accidents and ensure public safety.

The impact of drowsiness on cognitive function and reaction time can lead to delayed responses, reduced decision-making abilities, and impaired judgment, all of which increase the risk of accidents. Furthermore, drowsiness can also affect physical coordination and motor skills, further compromising an individual's ability to perform tasks safely. Therefore, developing effective methods for detecting and addressing drowsiness is crucial for mitigating its impact and preventing accidents and injuries. capacity to learn. To help future predictions, machine learning techniques are offered.

Fig 1. Drowsy person

Accident prevention can be significantly enhanced by employing advanced technologies such as image processing, electroencephalograph (EEG), artificial neural networks (ANN), and vocal and vehicle-based analysis systems, which are capable of analyzing driver drowsiness. While there are various models and methods available for detecting drowsiness, there is no single model that can accurately track whether a driver is drowsy or alert based on processed data alone.

Two popular methods for drowsiness detection include those that analyze facial expressions, eye movements, and other physiological indicators, as well as those that analyze video sequences. These models are trained using large datasets that consist of video samples labeled with drowsiness and alertness states. By training on such datasets, these models can accurately recognize and differentiate between drowsy and alert drivers.

Developing drowsiness detection or prevention devices as part of an accident-avoidance system is both challenging and critical. These devices must be highly accurate and reliable to effectively prevent accidents caused by drowsy driving. Ensuring the effectiveness of these devices requires rigorous testing and validation to guarantee their performance in real-world scenarios. Additionally, integrating these devices into vehicles and ensuring their compatibility with existing safety systems is essential for widespread adoption and maximum impact in preventing accidents and saving lives. Therefore,

while manufacturing drowsiness detection or prevention devices presents significant challenges, it is also a vital aspect of improving road safety and reducing the number of accidents caused by drowsy driving.

The effects of drowsiness pose a serious danger not only to the individual experiencing it but also to others on the road. Drowsiness can significantly impair driving ability, leading to an increased risk of accidents. One of the most concerning effects of drowsiness is the slowing of response times, which restricts the driver's ability to react quickly to potential hazards. This means that drowsy drivers may have difficulty braking quickly or swerving in time to avoid obstacles, increasing the likelihood of accidents.

Moreover, drowsiness also impairs judgment and attention, making it more likely for drivers to make poor decisions while behind the wheel. The reduced cognitive function and decreased alertness associated with drowsiness further compound the risks of driving while tired. At the extreme end, fatigue can lead to "microsleeping," which involves brief moments of unconsciousness, even at high speeds. These microsleeps can have catastrophic consequences, resulting in accidents and serious injuries.

The implications of these outcomes underscore the importance of getting an adequate amount of sleep and avoiding driving when drowsy. By prioritizing rest and recognizing the dangers of driving while tired, individuals can help prevent accidents and keep themselves and others safe on the road. Additionally, raising awareness about the dangers of drowsy driving and implementing effective drowsiness detection and prevention measures are crucial steps in reducing the number of accidents caused by drowsiness.

Drowsiness detection technology plays a crucial role in preventing accidents by alerting drivers when they are at risk of falling asleep behind the wheel. By providing timely warnings, these systems give drivers the opportunity to pull

over and rest, reducing the likelihood of accidents caused by drowsy driving. As a result, drivers are less fatigued, more alert, and able to fully focus on the

road, thereby enhancing overall road safety. Furthermore, drowsiness detection gadgets contribute to raising awareness about the dangers of drowsy driving. By alerting drivers to their fatigue levels and the importance of rest, these devices serve as a reminder for everyone to ensure they are adequately rested before embarking on a journey. This increased awareness helps to foster a culture of responsible driving and encourages individuals to prioritize their well-being and safety. While drowsiness detection technology is a valuable tool in preventing accidents, it is important to remember that it is not a substitute for responsible driving behavior. Drivers should still take responsibility for their own well-being and safety by listening to their bodies and taking breaks when needed, even if they have not received a drowsiness alert. Ultimately, by combining the use of drowsiness detection technology with responsible driving practices, we can work towards creating safer roads for everyone.

This project aims to develop a drowsiness detection system utilizing advanced deep learning techniques and Internet of Things (IoT) components. The system will continuously monitor physiological and behavioral cues, such as eye movements, head position, and facial expressions, which are indicative of drowsiness. By analyzing these cues in real-time, the system will accurately detect periods of reduced alertness in individuals. Upon detecting signs of drowsiness, the system will promptly trigger an alarm to alert the individual and nearby individuals, thereby preventing potential accidents or injuries. In addition to the immediate alert, the system will also send a notification via email to designated contacts, informing them of the drowsy state of the individual. This notification will prompt the contacts to take appropriate action, such as reaching out to the drowsy individual or providing assistance if necessary.

By integrating deep learning techniques and IoT components, this drowsiness detection system aims to provide an effective and reliable solution for preventing accidents caused by drowsy driving or other activities. The real-time monitoring and accurate detection capabilities of the system will help

ensure the safety of individuals in various settings, such as driving or operating heavy machinery, ultimately reducing the risk of accidents and injuries.

1.2 Statement of Problem

Drowsiness while driving or operating heavy machinery is a significant cause of accidents, often resulting in fatalities and severe injuries. To address this issue, the project aims to develop a drowsiness detection system that can accurately identify signs of drowsiness and promptly notify the user to take necessary precautions. Additionally, the system will incorporate an AI chatbot to engage the driver in conversation, helping to maintain alertness and prevent drowsiness.

1.3 Objectives:

1. Develop a system for detecting drowsiness by capturing live webcam recordings.
2. Utilize deep learning techniques to minimize false detections and prevent accidents.
3. Integrate an Alert system to notify when a person is drowsy.
4. Automating mails to the nearer ones and sending messages.

1.4 Scope

The scope of the project aims to develop a system capable of detecting driver drowsiness in real-time and alerting them to prevent accidents. The system will utilize computer vision techniques to analyze the driver's facial features and monitor their eye movements and blinking patterns. Through the use of a camera installed in the vehicle, the system will continuously track the driver's

face. If signs of drowsiness are detected, such as prolonged eye closure or abnormal head movements, an alarm will be triggered to alert the driver,

preventing potential accidents caused by drowsy driving. Additionally, the system will be designed to ensure minimal distraction to the driver while providing effective alerts to ensure road safety.

1.5 Applications

1.5.1 Road Safety: Drowsiness detection systems can be integrated into vehicles to alert drivers when they are becoming drowsy, helping to prevent accidents due to driver fatigue.

1.5.2 Workplace Safety: Drowsiness detection systems can be used in industries such as manufacturing, construction, and healthcare to prevent accidents caused by drowsy workers operating heavy machinery or performing critical tasks.

1.5.3 Health Monitoring: Drowsiness detection systems can be used to monitor the sleep patterns and alertness levels of individuals, helping to identify and manage sleep disorders such as sleep apnea and narcolepsy.

1.5.4 Consumer Electronics: Drowsiness detection systems can be integrated into wearable devices such as smartwatches and fitness trackers to provide real-time feedback on the user's alertness levels.

1.5.5 Medical Monitoring: Drowsiness detection systems can be used in hospitals and healthcare facilities to monitor patients for signs of drowsiness and alert healthcare providers when intervention is needed.

1.6 Limitations

1.6.1 Variability in conditions: The model might not perform well under various lighting conditions, different camera angles, or when the person is wearing glasses or has different facial features.

1.6.2 Limited to frontal face detection: Most drowsiness detection models rely on detecting faces, and they might fail if the face is not directly in front of the camera or if the face is partially obscured.

1.6.3 Data bias: The performance of the model might be biased based on the dataset it was trained on, leading to poor generalization to new or diverse conditions.

1.6.4 Processing power: Deep learning models can be computationally expensive, especially if the application requires real-time processing, which might limit their deployment on low-power devices.

1.6.5 Dependency on hardware: The performance of the system may vary depending on the quality and specifications of the camera and hardware used.

CHAPTER-2

LITERATURE SURVEY

In **Priyanka et al. [1]** and her co-authors come up with a Prototype of accident detection and alert system that comprise of Arduino UNO, GSM, GPS, SW420 Sensor, 2-ch Relay and MQ-3 Alcohol sensor, which detect modifications in the vibrations and as well as alcohol using the hardware units and utilize accompanying software to dump the code in Arduino The alerts messages will be sent to the displaying OLED device when the values more than the threshold or conditioned extensive to receive.

Yuman Albadawi et al. [2] employed the image perceivable DDD system as a public dataset in Computer Vision Lab. The advantages are real time monitoring and it prevents accidents and it enhances safety. High cost and it depends on accuracy and privacy concerns are observed as drawbacks in the model.

Fernando E. Quiroz, Jr. et al. [3] utilizes a vision-based approach, employing a histogram of oriented gradient-based face detector and a facial landmark detector to extract eye areas. The advantages are it has wide applicability and it focuses more on road safety. The drawbacks include limited evaluation information, reliance on a predefined threshold for the eye aspect ratio, and potential sensitivity to individual variability in facial features or expressions.

Priyadarsini et al. [4], devised a system dependent on computer vision techniques, the face of the driver who was seen on the coloured video in the car was found. Thereafter, eye tracking is done to identify the locations of the driver’s eyes which will be used as templates for subsequent frames. The images of the eye that is being monitored are used to identify drowsiness issues through the generation of warning alarms. Image processing's job is to identify the driver’s face and then extract the driver's eye image for drowsiness detection.

Mehta et al. [5] have created a mobile application capable of detecting facial landmarks and subsequently estimating the EAR and ECR. These metrics are then utilized to predict driver drowsiness through machine learning models, achieving an accuracy rate of 84%. Smart glass that has been innovated by the new business Ellice-Healthy allows to monitor somnolence as a result of automatic blinking, eye recording, and recording of vital signs. With the beeping and the driver's somnolence easily detectable by the smart glass, it would cut in cutting down the driver's fatigue.

Md. Yousuf Hossain et al. [6] propose a real-time approach based on the Raspberry Pi, pi Camera, and a buzzer for detecting drowsiness without affecting the operation of the system. The Haar cascade classifier is utilized to detect facial landmarks. For a typical human eye, the eye aspect ratio (EAR) is approximately 0.25. The system's judgement by gazing only at the eye aspect ratio (EAR) for detection of drowsiness may lead to the limited accuracy neglecting other indicators of drowsiness, making it possible to get false positives or negatives.

Hrishikesh B. Juvale et al. [7] used images generated from the infra-red night vision camera and a specific threshold can be put into place to turn the images into binary. Besides that, the efficiency of this paper is brought about by pixel isolation. Slope approach is used to calibrate and provide accuracy as slopes are calculated between clusters, which ultimately lead to the formation of clusters representing the pupils. It is exit from the intrusive way and has a high benefit. One of the major challenges is Dependency and artifact recognition and Low accuracy.

Roopalakshmi R et al. [8] propose a method to detect drowsiness by blinking eye count. Raspberry Pi, pi camera and also alerts are sounded off by a buzzer and a vibrator when a system is implemented. A Haar cascade, on the other hand, assists in identifying a face that has eyes. If human fatigue condition can be detected for more than 2 seconds, then the blinking rate of eyes is affected. The simple eye blink data which is count will overlook other critical signs of driver fatigue and reduce the sensitivity in which the drowsiness is detected precisely.

Chowdhury et al. [9] reviewed various physiological sensors used to determine the level of drowsiness in a driver. The observed vital indicators for evaluating drowsiness were electrocardiography, respiratory belt, electrography, electrooculography, electromyography, galvanic skin response, skin temperature, and hybrid techniques. In the section of the EEG methods, the authors incorporated articles using the spectral power functions, event related potentials and entropies. The dilemma of measurement intrusion: drivers should be considered.

Liu et al. [10] suggested a facial feature capturing-based algorithm for fatigue detection, which involves eye closure duration, head nodding as well as yawning, and feeding them into a convolutional gamma fatigue network. Firstly, the algorithm detects the driver's eyes and mouth by means of the cascade CNNs. The application of an algorithm that is solely based on the observable signs, like the length of eye closing cycle, the frequency of nodding of the head and the count of yawns might reduce its capability to identify the subtler signs of fatigue and thus might cause occasional false positive and negative results in real life of driving

CHAPTER -3

ANALYSIS

System Analysis is the detailed study of the various operations performed by the system and their relationships within and outside the system. The breakdown of something into parts so that the entire system may be understood.

System Analysis is concerned mainly with understanding or being aware of the problem, identifying the relevant variables which are used for decision-making, and analyzing and synthesizing them to obtain optimal solutions. Another view of it is a Problem-Solving technique that breaks down a system into different parts and it studies how those parts will interact to accomplish their purpose.

3.1 Existing System:

Detecting drowsiness is a broad term for using different technologies or methodologies to monitor anyone's state of alertness or sleepiness. The existing systems for detecting drowsiness are: The existing systems for detecting drowsiness are:

1. In Fig 2 (below) is described. CNN classifier being used to perceive drowsiness and now a dataset passed to the CNN model.

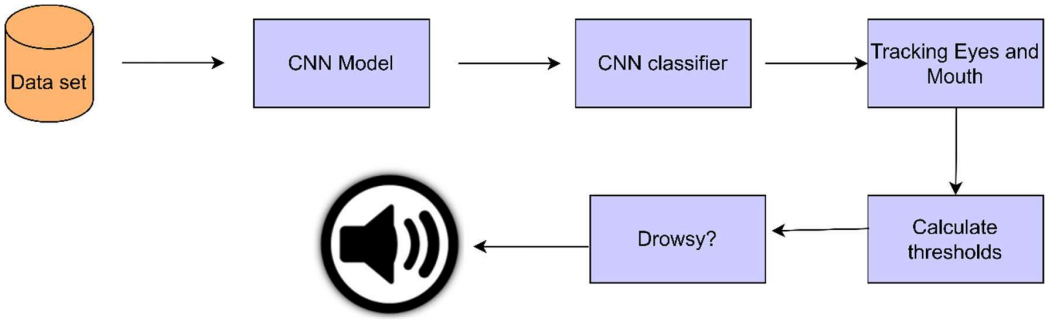

Fig 2. Detecting drowsiness using CNN Model

The process of detecting drowsiness using Convolutional Neural Networks (CNNs) involves the following steps: The process of detecting drowsiness using Convolutional Neural Networks (CNNs) involves the following steps:

i)Data Collection: Dataset applied for the training process should be well labelled and be composed of images or videos of people both in alerted and sleepy states. These photos would be displayed in such a way as to cover the whole range of facial expressions, but also the features that differentiate/indicate being awake.

ii)Data Preprocessing: Adjust the images before training by scaling, changing (normalizing) the pixel values, and reversing the images to create a tailorized training process. This part contributes to the fact of the CNN discovering the features, which determine the drowsiness directly.

iii) Model Architecture: Design a model based on a CNN, where images will serve as the input and the model will be fed by multiple layers of activation which will be used for feature extraction and the last one will be fully connected layer which will be used for classification. The final level usually presents neurons using the sigmoidal function, which means a binary classification (Status: alert/drowsy).

iv)Training: Implement a convolutional neural network (CNN) on the saved dataset you have just prepared. When learning the data set the model will get used to recognize the image features and patterns that indicate tiredness. Using a loss function applicable to this scenario and optimizer with cross entropy loss used.

v)Validation: Validate the model by training it on a separate set of data that was not used during training to see if it provides a good generalized performance on new cases. Compare the options of the hyperparameters, architecture and the training methods that have the best effect on performance.

vi)Testing: The evaluation of the trained CNN on the test set which contains data examples from real-life scenarios will identify the CNN reaction speed and the promise for potential further use.

vii)Deployment: Involve the trained CNN where possible such as inside the vehicles, workplaces, and other place to, at the same time, understand the level of drowsiness and the alerts that signal drowsiness.

Algorithm 1 Driver Drowsiness Detection Model using CNN

Require: Input shape $(32, 32, 3)$

1: Initialize model
2: Define optimizer: Adam (0.0001)
3: Compile model with loss function: categorical crossentropy, metrics: accuracy
4: Define data augmentation: rotation range 20, zoom range 0.2, horizontal flip
5: Load dataset
6: Preprocess dataset: Resize images to $(32, 32)$, One-hot encode labels
7: Split dataset into training and testing sets (80% training, 20% testing)
8: Train model with data augmentation for 200 epochs
9: Test the data

Advantages:

1. Feature learning

2. High Accuracy.

3. Robustness to variations.

4. Real time processing.

Disadvantages:

1. Large Dataset requirement

2. Overfitting.

3. Data bias

4. Limited to facial features

3.2 Proposed System:

One of the dangerous problems that is related to road safety, bringing forth severe injuries, loss of lives, and financial load, is exhaustion/drowsiness. The automatic progressive drowsiness that comes with the gradual process of going from fully alert while awake to falling asleep is a severe factor in almost all of the unfortunate car accidents that result in fatalities.

3.2.1 ALERT GUARD

Drowsiness detection and notification through an integrated system of IoT, deep learning model, and OpenCV offer a comprehensive solution to enhance road safety. It uses cameras or IR sensors installed in vehicles, OpenCV processes the streaming video data, detecting facial features crucial for assessing drowsiness, such as eye closure and head movements. Concurrently, a deep learning model using Dlib, is employed to analyse these features, identifying patterns indicative of drowsiness with high accuracy. The IoT component enables seamless communication between the detection system and the alarm mechanism. When the deep learning model identifies drowsiness in the driver, it triggers an alarm through the IoT network, alerting the driver in real-time to take necessary corrective action or pull over for rest. This integrated approach not only offers proactive safety measures but also showcases the potential of combining emerging technologies to mitigate risks associated with driver fatigue, ultimately contributing to safer roads for all.

The whole process starts when the system is initiated. It uses a webcam to scan the user's face and relies on libraries like dlib and OpenCV to recognize the face. One of the key issues which directly correlates with road traffic safety, thus causing injuries and deaths of drivers, and a large financial burden, is the phenomenon of drowsiness/fatigue. The fact that most of the serious auto crashes that cause great loss of life are the result of the unique monotony of the gradual process of drifting to sleep from full wakefulness when one is driving is as crucial as any of the factors involved.

If the Eye aspect ratio value is greater than 0.25, it will be interpreted that the person is premade. The EAR (Ethogram of Attention and Reactivity) Value will be less than or equal to 0.25 if the sleeper is drowsy. Such a two-stage method makes it possible to understand drowsiness which is determined by above-mentioned threshold value of EAR. It is a binary call, yes or no, drowsy or not drowsy.

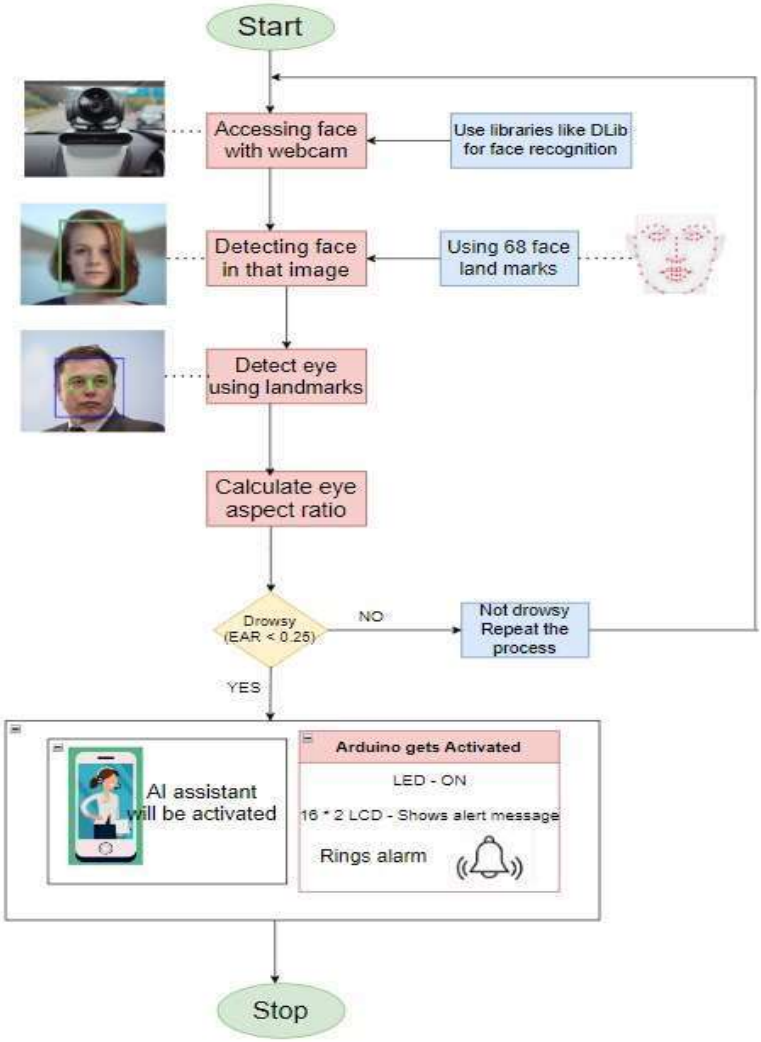

Fig 3. Drowsiness detection using Alert Guard system

1. Accessing face using live webcam: Utilize the Raspberry Pi camera module to access live webcam feed for face detection.

2. Detecting face in the image: Implement face detection algorithm to locate faces in the captured images.

3. Detecting eyes using 68 face landmarks: Utilize the 68 face landmarks to accurately detect and locate eyes within the detected faces.

4. Calculate EAR (Eye Aspect Ratio): Calculate the Eye Aspect Ratio (EAR) to determine the level of drowsiness based on eye closure.

5. If a person is drowsy then ring the alarm and send messages: Trigger an alarm and send alert messages if the calculated EAR indicates that the person is drowsy.

ADVANTAGES:

1.Real time detection

2.Cross platform compatibility

3.Reduced false Alarms

4.Continuous Improvement

DISADVANTAGES:

1.Hardware Requirements

2.Data privacy concerns

3.Sensitivity to environmental factors.

Algorithm 1 Driver Drowsiness Detection Algorithm

Require: Video capture, Face detector, Facial landmarks predictor
Ensure: Alert when driver is drowsy

```
Initialize video capture
Load face detector and facial landmarks predictor
Initialize sleep, drowsy, active, status, and color variables
while True do
    Read frame from video capture
    Convert frame to grayscale
    Detect faces in grayscale frame
    for each detected face do
        Get coordinates of the face
        Draw rectangle around the face on the frame
        Detect facial landmarks
        Compute eye blink ratio for left and right eyes
        if either eye blink ratio is low then
            Increment sleep counter and reset drowsy and active counters
            if sleep counter exceeds threshold then
                Display "SLEEPING !!!"
                Send email alert
            end if
        else if either eye blink ratio is moderate then
            Increment drowsy counter and reset sleep and active counters
            if drowsy counter exceeds threshold then
                Display "Drowsy !"
                Send email alert
            end if
        else
            Increment active counter and reset sleep and drowsy counters
            if active counter exceeds threshold then
                Display "Active :)"
            end if
        end if
        for each facial landmark do
            Draw circle around the landmark on the face frame
        end for
    end for
    Display frame with annotations
    if escape key is pressed then
        Exit loop
    end if
end while
```

3.3 Software & Hardware Requirements:

SOFTWARE REQUIREMENTS

- Programming language – Python(Version – 3.11)
- Framework – Dlib
- Libraries – Numpy, Pandas, OpenCV, Matplotlib
- Development Environment – IDLE

HARDWARE REQUIREMENTS

- Processor – intel core i3 or above
- Memory – Atleast 8 GB RAM
- Storage – 256 GB SSD or more
- Stable Internet Connectivity

CHAPTER -4

DESIGN

The most creative and challenging part of system development is System Design. The Design of a system can be defined as a process of applying various techniques and principles for the purpose of defining a device, architecture, modules, interfaces and data for the system to satisfy specified requirements. For the creation of a new system, the system design is a solution to "how to" approach.

UML DIAGRAMS:

Drowsiness detection systems aim to enhance safety in various contexts, particularly in transportation and critical operational environments, by monitoring and alerting individuals when signs of drowsiness are detected. In a UML (Unified Modeling Language) diagram for such a project, you might typically start with a use case diagram to identify the actors interacting with the system and the specific functionalities it provides, such as monitoring, analysis, and alerting. Following this, a class diagram would help represent the key classes involved, like sensors, data processing modules, and alert mechanisms, along with their attributes and relationships. Additionally, sequence diagrams could illustrate the interactions between these classes over time, depicting the flow of information and control as the system detects drowsiness events and triggers appropriate responses. State diagrams could further clarify the different states the system and its components may transition through, such as awake, drowsy, and alerting. Finally, activity diagrams could outline the high-level processes involved, from data acquisition to decision-making and response generation, providing a comprehensive overview of the system's functionality and behavior.

4.1 Use Case Diagram:

In a use case diagram for drowsiness detection, the primary actors typically include the system itself and the user. The system interacts with various components such as sensors for detecting drowsiness indicators like eye closure and head position. Use cases would include functionalities like data collection, analysis, and alerting. For instance, a use case might depict the system collecting data from sensors, analyzing it to determine if the user is drowsy, and then triggering an alert if necessary. Additional use cases could involve user interaction, such as adjusting sensitivity settings or reviewing drowsiness detection history. Overall, the use case diagram provides a clear visual representation of the system's functionality and the interactions between its components and users in the context of detecting drowsiness.

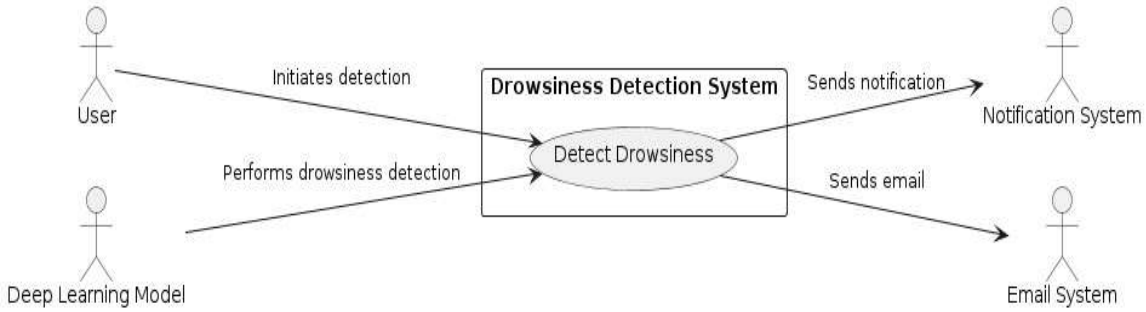

Fig 4.1 – Use-Case Diagram for detecting drowsiness

4.2 Class Diagram:

class diagram for drowsiness detection, essential classes include sensors, data processing modules, and alert mechanisms. Sensors capture data like eye movement, processed by modules to detect drowsiness using defined algorithms or thresholds. Alert mechanisms trigger warnings based on detected drowsiness. Relationships show how these classes interact, like sensors feeding data to processing modules.

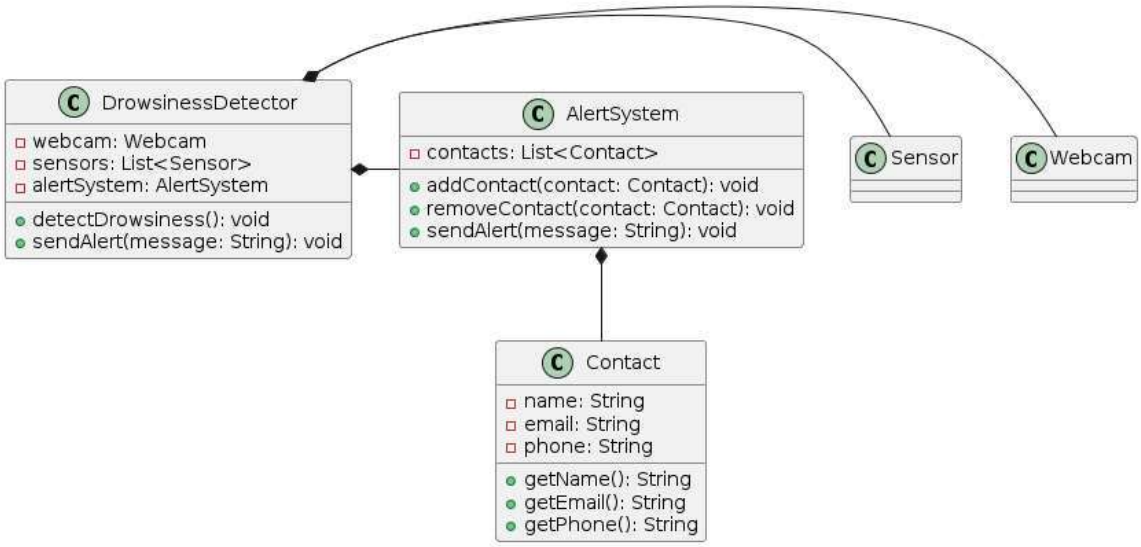

Fig - 4.2 Class Diagram for detecting drowsiness

4.3 Sequence Diagram:

In a sequence diagram for detecting drowsiness, the flow of interactions between system components and external entities is illustrated over time. It typically begins with an actor initiating the drowsiness detection process, such as the system's user. The diagram then depicts the sequence of messages exchanged between components, like sensors, data processing modules, and alert mechanisms. For example, it may show the sensor class sending data to the data processing module, which then analyzes the data to detect signs of drowsiness. If drowsiness is detected, the alert mechanism class is invoked to trigger an appropriate response, such as sounding an alarm or notifying the user. Sequence diagrams help visualize the dynamic behavior of the system, showcasing how different components collaborate to achieve drowsiness detection in real-time scenarios.

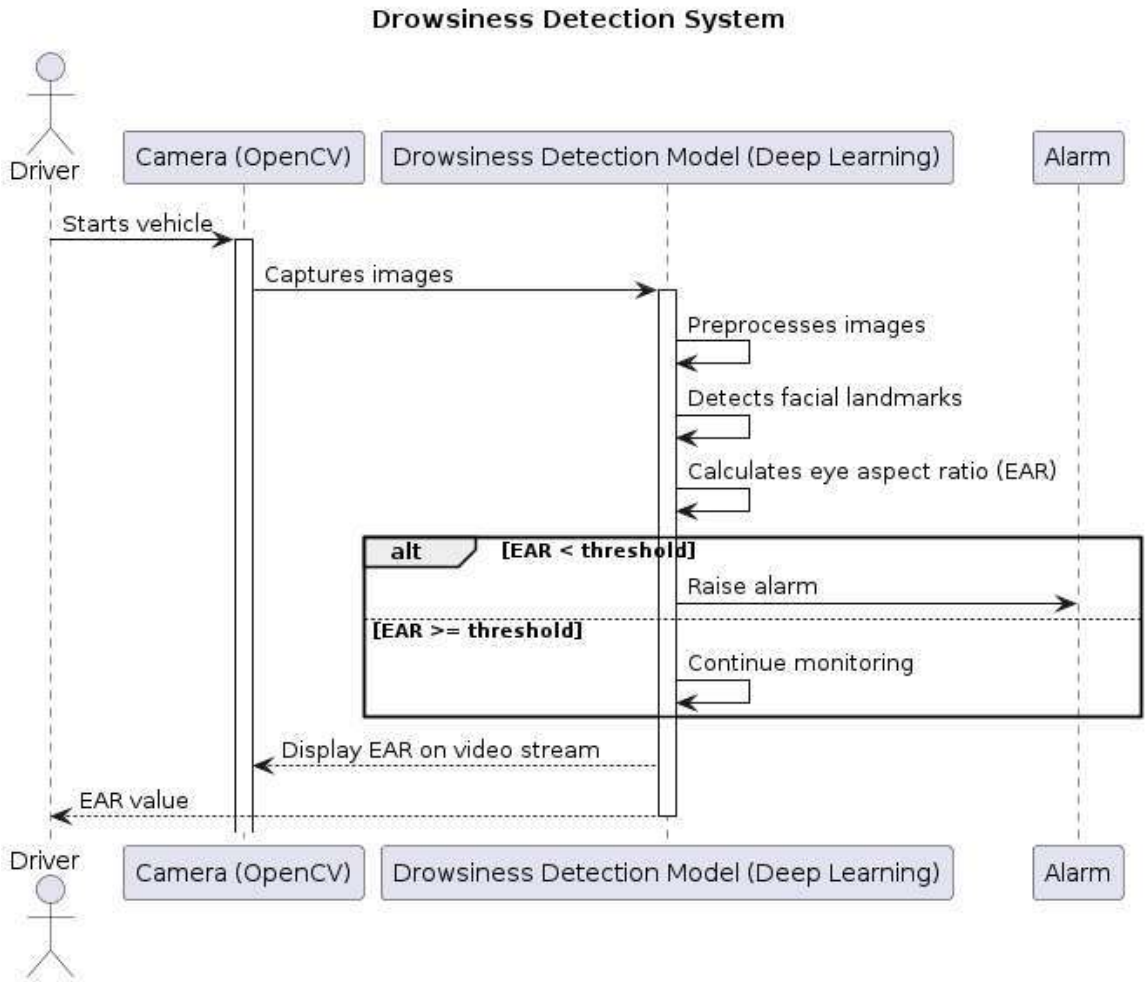

Fig 4.3- Sequence Diagram for detecting drowsiness

4.4 ER Diagram:

An Entity-Relationship (ER) diagram for detecting drowsiness would focus on the relationships between the various entities involved in the system. Entities could include sensors, data processing modules, users, and alert mechanisms. Relationships would illustrate how these entities interact and depend on each other to facilitate drowsiness detection. For example, sensors would have a one-to-many relationship with data processing modules, indicating that multiple sensors contribute data to the processing module for analysis. Users might have a one-to-one relationship with the alert mechanism, showing that each user is associated with their own alert settings and notifications.

Attributes associated with each entity would further detail their characteristics, such as sensor types or user preferences. Overall, the ER diagram provides a structured representation of the entities, their attributes, and the relationships between them in the context of drowsiness detection.

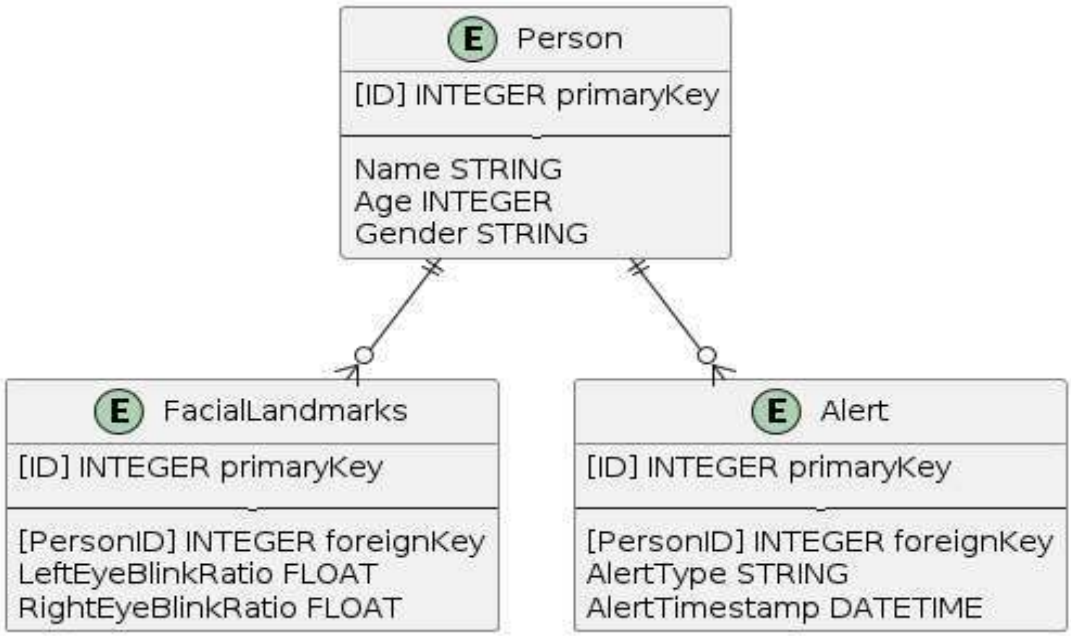

Fig 4.4- ER Diagram for detecting drowsiness

4.5 DFD Diagram:

A Data Flow Diagram (DFD) for detecting drowsiness visualizes data flow within the system. It includes processes (like data collection and analysis), data stores (where data is stored), data flows (movement of data between processes and stores), and external entities (such as sensors or users). In this context, data flows from sensors to processing modules for analysis and then to alert mechanisms for response generation.

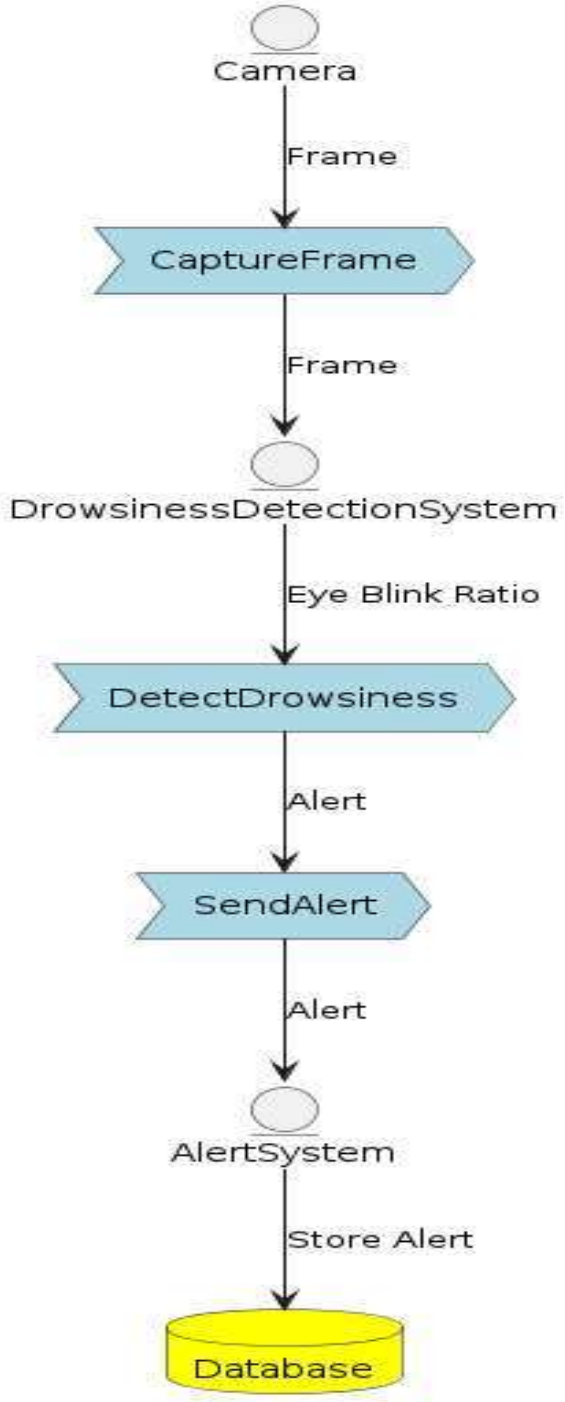

Fig-4.5 Data Flow Diagram for Predicting drowsiness

4.6 Block Diagram

It below block Diagram describes the basic steps for the identification of drowsiness using a deep learning technique and Open CV.

Fig 4.6 Block Diagram for Predicting detecting drowsiness

CHAPTER 5

IMPLEMENTATION

Detecting drowsiness in drivers is crucial for preventing accidents caused by fatigue. In this implementation, we use computer vision techniques to detect drowsiness by analyzing facial landmarks. We'll monitor the driver's eye movement and raise an alert if signs of drowsiness are detected.

5.1 Dataset Used:

In drowsiness detection, a basic dataset comprising images or video frames is used. These frames contain the facial features of the driver, including 68 facial landmark points. These landmark points provide crucial information about the driver's facial expressions, eye movements, and head pose. By analyzing these features, the drowsiness detection system can identify signs of drowsiness in the driver. The dataset is essential for training and testing the drowsiness detection algorithm, ensuring its accuracy and reliability in real-world scenarios.

Dataset size : 97358 KB

Type of file : DAT file

Number of landmarks : 68

Purpose : Detecting face and eyes in a live video

5.2 Implementation

5.2.1 Capture Video Stream:

To begin, it initializes the camera to capture video frames using OpenCV. This is achieved by creating a VideoCapture object and passing 0 as an argument, which represents the default camera. To capture the video stream, the script utilizes the OpenCV library, which provides functions for handling video streams and images.

cv2.VideoCapture(0) creates a VideoCapture object cap to capture video frames.

The argument 0 represents the default camera connected to the system. If multiple cameras are connected, you can specify the camera index (e.g., 1, 2, etc.) to select a different camera.

5.2.2 Detect Faces:

This utilizes the dlib library to create a face detector object. Then, it detects faces in the grayscale video frame using the created face detector object. The detected faces are stored as bounding boxes. get_frontal_face_detector helps in detecting the face in a video stream.

5.2.3 Detect Facial Landmarks:

Next, a pre-trained facial landmark detector model is loaded using dlib. This model is capable of detecting 68 facial landmark points in a given face. These landmark points provide crucial information about the driver's facial features, such as the eyes, nose, and mouth. This step is essential as it identifies specific points on the driver's face, such as the eyes, which are crucial for analyzing eye movements and detecting drowsiness.

5.2.4 Compute Eye Aspect Ratio (EAR):

EAR defines a function to calculate the Eye Aspect Ratio (EAR) for each eye. EAR is a measure of eye openness and is computed using the Euclidean distances between specific landmark points of the eyes. The formula to compute EAR is (A + B) / (2.0 * C).

5.2.5 Detect Drowsiness:

In this step, thresholds for EAR and consecutive frames with low EAR are defined. If the calculated EAR falls below the predefined threshold, a counter is incremented. If this counter exceeds a certain number of consecutive frames, it indicates drowsiness.

5.2.6 Send Alert:

Finally, it sends an email alert when drowsiness is detected. It uses a predefined function to send an email, including information about the detected drowsiness. This alert helps in timely intervention, preventing potential accidents due to driver drowsiness.

CHAPTER 6

EXECUTION PROCEDURE AND TESTING

Drowsy driving is a significant cause of accidents, injuries, and fatalities on roads worldwide. To address this issue, advanced driver assistance systems (ADAS) can be deployed to detect drowsiness and alert the driver in real-time. In this project, we'll implement a drowsiness detection system using OpenCV and deep learning techniques.

6.1 Execution Procedure:

6.1.1 Installing all the required libraries

```
pip install opencv-python-headless
pip install dlib
pip install numpy
```

Open CV, dlib, numpy are the needed libraries in order to detect drowsiness.

A. Open CV : OpenCV (Open Source Computer Vision Library) is a popular open-source computer vision and machine learning software library, widely used for real-time image and video processing.
B. Dlib : dlib is used for face detection and facial landmark detection. It provides pre-trained models for detecting faces and extracting facial landmarks, which are crucial for monitoring eye movements and detecting drowsiness accurately.
C. Numpy : NumPy is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. In drowsiness detection, NumPy is used for numerical operations and array manipulation.

6.1.2 Importing the packages

Import necessary libraries such as OpenCV, dlib, numpy, and smtplib.

```
import cv2
import dlib
import numpy as np
import smtplib
```

6.1.3 Load Pre-trained Models:

Load the pre-trained face detector and facial landmark predictor models.

```
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
```

To load the pre-trained face detector and facial landmark predictor models in Python using the dlib library, you first need to import dlib. Then, you can load the face detector using the get_frontal_face_detector() function. This detector is capable of detecting faces in images. Next, you can load the facial landmark predictor using the shape_predictor() function, passing the path to the pre-trained model file"shape_predictor_68_face_landmarks.dat". This model is capable of predicting 68 facial landmarks such as the locations of eyes, nose, mouth, etc. Once loaded, these models can be used to detect faces in images and to locate facial landmarks accurately.

6.1.4 Define a function to calculate EAR

EAR stands for Eye Aspect Ratio. It is a mathematical calculation used in drowsiness detection systems to estimate how open a person's eyes. The threshold value of EAR is 0.25.

```
def eye_aspect_ratio(eye):
    A = np.linalg.norm(eye[1] - eye[5])
    B = np.linalg.norm(eye[2] - eye[4])
    C = np.linalg.norm(eye[0] - eye[3])
    ear = (A + B) / (2.0 * C)
    return ear
```

6.1.5 Capture the video stream

To start capturing the video stream from the camera using OpenCV, you need to create a VideoCapture object. The argument 0 passed to VideoCapture() indicates that you want to use the default camera available on your system. If you have multiple cameras, you can specify the index of the camera you want to use (e.g., 1 for the second camera, 2 for the third camera, and so on).

```
cap = cv2.VideoCapture(0)
```

6.1.5 Process the video stream

Video stream continuously monitor a video stream, the program reads frames and converts them to grayscale. It then detects faces in each frame and identifies facial landmarks. Using these landmarks, it calculates the Eye Aspect Ratio (EAR) for each eye. By comparing the EAR values, it determines if the eyes are closed. If drowsiness is detected, an alarm is triggered to alert the user. This process ensures real-time monitoring for signs of drowsiness in the video stream.

```
import cv2
import numpy as np
import dlib
from imutils import face_utils
import serial
import pywhatkit
import time
import smtplib
cap = cv2.VideoCapture(0)
hog_face_detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("C:\\Users\\chinn\\Downloads\\shape_predictor_68_face_landmarks.dat")

sleep = 0
drowsy = 0
active = 0
status=""
color=(0,0,0)

def compute(ptA,ptB):
        dist = np.linalg.norm(ptA - ptB)
        return dist

def blinked(a,b,c,d,e,f):
        up = compute(b,d) + compute(c,e)
        down = compute(a,f)
        ratio = up/(2.0*down)

        if(ratio>0.25):
                return 2
        elif(ratio>0.21 and ratio<=0.25):
                return 1
        else:
                return 0
```

```
while True:
    _, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = hog_face_detector(gray)

    for face in faces:
        x1 = face.left()
        y1 = face.top()
        x2 = face.right()
        y2 = face.bottom()

        face_frame = frame.copy()
        cv2.rectangle(face_frame, (x1, y1), (x2, y2), (0, 255, 0), 2)

        landmarks = predictor(gray, face)
        landmarks = face_utils.shape_to_np(landmarks)

        left_blink = blinked(landmarks[36],landmarks[37],
        landmarks[38], landmarks[41], landmarks[40], landmarks[39])
        right_blink = blinked(landmarks[42],landmarks[43],
        landmarks[44], landmarks[47], landmarks[46], landmarks[45])

        if(left_blink==0 or right_blink==0):
            sleep+=1
            drowsy=0
            active=0
            if(sleep>6):

                time.sleep(2)
                status="SLEEPING !!!"
                color = (0,0,255)
```

```
        elif(left_blink==1 or right_blink==1):
            sleep=0
            active=0
            drowsy+=1
            if(drowsy>6):

                time.sleep(2)
                status="Drowsy !"

                color = (0,0,255)

        else:
            drowsy=0
            sleep=0
            active+=1
            if(active>6):

                time.sleep(2)
                status="Active :)"
                color = (0,0,255)

        cv2.putText(frame, status, (100,100), cv2.FONT_HERSHEY_SIMPLEX, 1.2, color,3)
        if(status=="Drowsy !" or status=="SLEEPING !!!"):
           #pywhatkit.sendwhatmsg("+919866596051", "Person is drowsy. Kindly alert!", 11,50)
```

```
           email=input("EMAIL : ")
           rec_mail=input("REC_EMAIL : ")
           l=rec_mail.split(',')
           sub=input("SUB:")
           msg=input("MESSAGE:")
           text=f"Subject: {sub}\n\n{msg}"
           for i in range(len(l)):
              server=smtplib.SMTP("smtp.gmail.com",587)
              server.starttls()
              server.login(email,"guqgmfyhgmejgdnm")
              server.sendmail(email,l[i],text)
              print("Email has been sent to "+rec_mail)
              #break


        for n in range(0, 68):
            (x,y) = landmarks[n]
            cv2.circle(face_frame, (x, y), 1, (255, 255, 255), -1)

cv2.imshow("Frame", frame)
#cv2.imshow("Result of detector", face_frame)
key = cv2.waitKey(1)
if key == 27:
    break
```

6.2 Testing:

Testing is an important part of model development and involves evaluating the performance of the trained models on a previously unseen dataset. This is done to ensure that the models have not to overfit the training data and can generalize well to new data. The testing dataset is used to evaluate the

performance of the models by comparing the predicted values to the actual values. Metrics such as accuracy, precision, recall, F1-score, and AUC-ROC can be used to evaluate the performance of the models.

6.2.1 Data Preprocessing Testing:

Verify that the preprocessing steps handle various types of input data correctly, including handling missing values, scaling numerical features, and encoding categorical variables. Test the preprocessing pipeline with different types of data to ensure consistency and correctness.

6.2.2 Model Training Testing:

Validate that the model training process converges correctly and produces expected model parameters. Test the training pipeline with different configurations and datasets to ensure stability and reproducibility.

6.2.3 Model Prediction Testing:

Test the model prediction logic to verify that it produces accurate predictions for input data. Use synthetic or real-world data to evaluate the model's performance across different scenarios and drowsiness levels. Verify that the model outputs are consistent with expectations and align with ground truth labels or expert assessments.

6.2.4 Edge Case Testing:

Test the drowsiness detection system with edge cases and boundary conditions to ensure robustness and reliability. Include test cases with extreme or unusual input data to verify that the system handles them gracefully and produces sensible predictions.

6.2.5 Integration Testing:

Test the integration between different components of the drowsiness detection system, including data preprocessing, model training, and

prediction. Ensure that data flows smoothly through the pipeline and that transformations are applied correctly at each step.

6.2.6 Data Quality Testing:

Evaluate the quality of the dataset used for training and testing the drowsiness detection system. Test for data completeness, consistency, and correctness to ensure that the data is reliable and representative of the target population. Identify and address any data anomalies or inconsistencies that may affect the performance of the system.

CHAPTER 7
RESULT AND PERFORMANCE EVALUATION

Detecting drowsiness is crucial, especially in contexts where alertness is essential for safety, like driving or operating heavy machinery. We have developed two methods for drowsiness detection: static and dynamic.

1. **Static Detection** : For static detection, we have compiled a dataset consisting of 726 images of open eyes and 726 images of closed eyes. When a user uploads an image, our system compares it with the images in the dataset. If the system finds a match with a closed-eye image, it indicates that drowsiness is present.

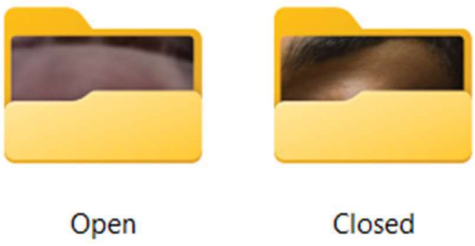

Fig. Static data of open and closed eyes of 726 files each

Training dataset uses two classes of the dataset. Characteristics of the dataset are as follows —

i)The dataset contains a total of 1452 images in two categories.
ii)Each category has 726 images.
iii)The dataset is already balanced, so no need to balance the dataset.
iv)Class Labels — 'Open Eye' and 'Closed Eye'.
v)Class Labels were encoded such that 0 represents Open Eye and 1 illustrates Closed Eye.

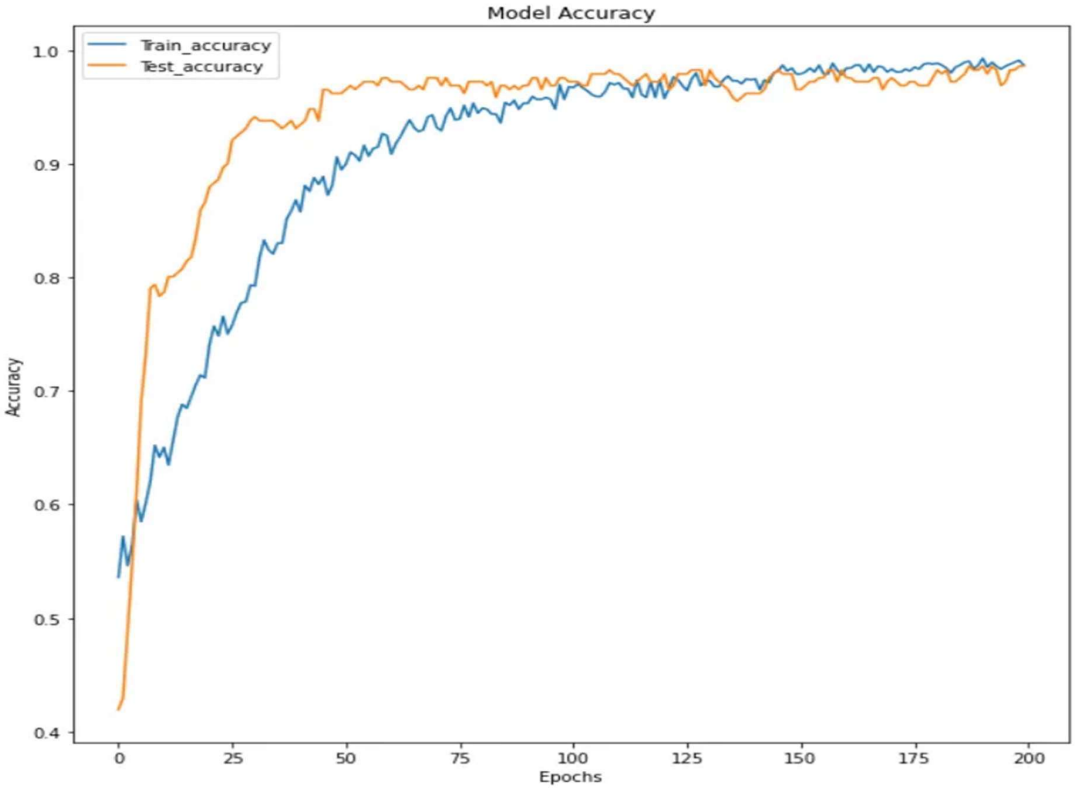

Fig . Accuracy curve for train and test accuracy in drowsiness detection

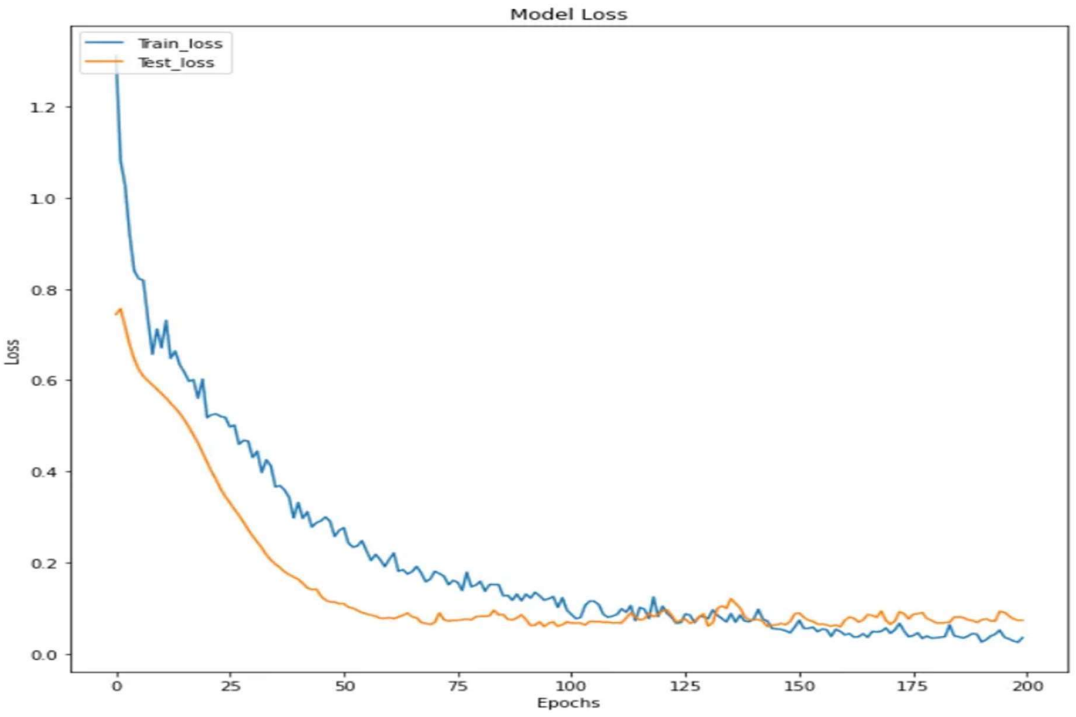

Fig. Loss curve for train and test loss in drowsiness detection

2. **Dynamic Detection:** Dynamic detection, on the other hand, utilizes a live webcam feed. The system continuously records video from the webcam, and through real-time analysis, it detects signs of drowsiness. If drowsiness is detected in the video stream, an alert is triggered, warning the user or operator of the situation. This method is particularly useful in scenarios where continuous monitoring is required, such as in transportation systems or surveillance. In the dynamic detection system using a live webcam, there are two stages that can be predicted: the drowsy state and the active state.

A. Drowsy State:

During the drowsy state, the system monitors the user's behavior in real-time through the webcam feed. It analyzes various factors such as eye movement, blink patterns, head pose, and facial expressions. When the system detects signs indicating that the user is becoming drowsy, such as frequent blinking, prolonged eye closure, or nodding off, it classifies the user as being in a drowsy state.

Fig. Drowsy state

B. Active State:

Conversely, when the system observes that the user is alert, attentive, and exhibiting normal behavior, it classifies the user as being in an active state. In this state, the system does not detect any signs of drowsiness or fatigue, and the user is considered to be fully alert and attentive.

Fig. Active state

By continuously monitoring the user's behavior and classifying it into either the drowsy state or the active state, the system can provide real-time alerts to prevent accidents or errors caused by drowsiness, especially in critical situations such as driving or operating machinery. We can send alert to their nearer ones through mails and messages. We can automate mails and messages if a person is drowsy.

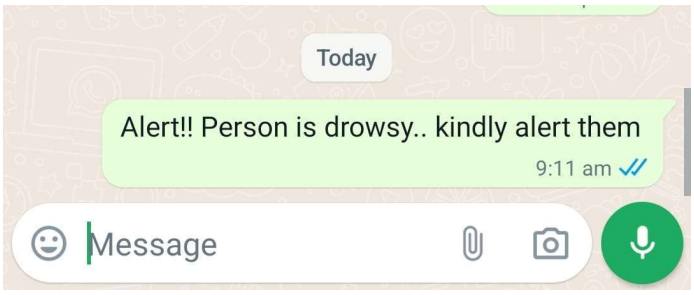

Fig. Notification Sent to Nearest Contact Regarding Drowsiness Detection through mails and messages

A notification has been sent to the nearest contact informing them that the individual has been identified as drowsy. This notification serves as

an alert to ensure that appropriate action can be taken to prevent any potential risks or accidents associated with drowsiness.

Here's a comparison of performance between different techniques used in drowsiness detection.

Table 1. Comparison of performance between various techniques(in %)

Techniques	Accuracy	Sensitivity	Precision
SVM	85	80	87
KNN	82	75	84
CNN	90	88	92
Alert Guard	93	90	92

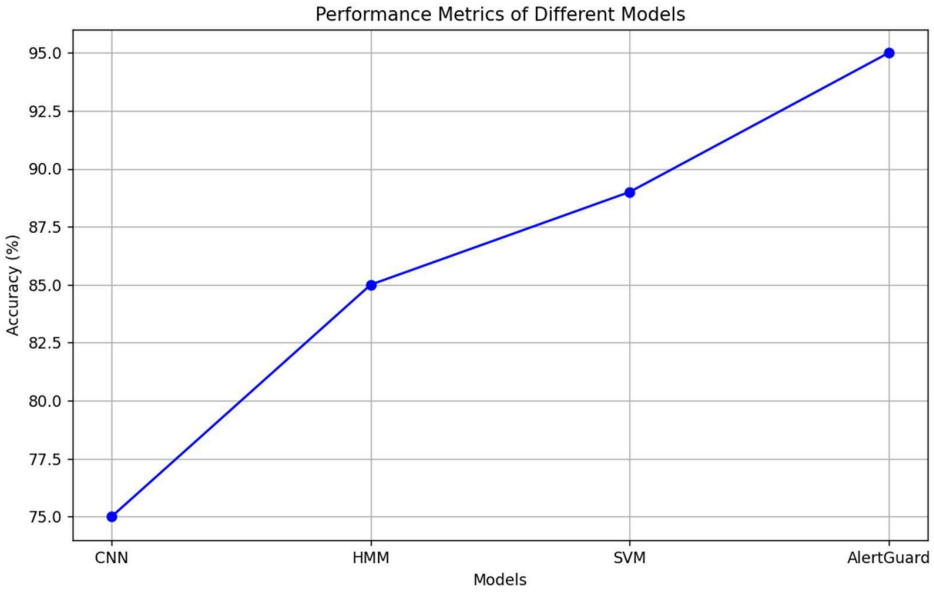

Fig. Performance Analysis of various techniques in detecting drowsiness

CHAPTER-8

CONCLUSION AND FUTURE WORK

Alert Guard an ensemble based deep learning algorithm named Alert Guard is developed for detecting drowsiness and notifying through an Alarm using live webcam. Here a live webcam is used instead of training a predefined dataset to the model. The results show that the performance of the model in detecting the driver drowsiness as compared to other models. Performance metrics were employed to measure and compare the models' efficiency. In further the current algorithm can be integrated with the chatbot to converse with the person to avoid drowsiness and we can also intimate to the nearby vehicles by establishing V2V communication.

The sensor is also used to detect if a person has taken alcohol. In future it can be implemented in schools and colleges to alert the lecturers to find if the students are sleepy. The vibration sensor and GPS is used for detecting accidents and the location is sent through an Email. We can also further extend the project by integrating the virtual assistants or AI chatbots to engage a person in conversation and reduce the sleepiness of a driver.

APPENDIX

PYTHON CODE :

```
#Importing OpenCV Library for basic image processing functions
import cv2
# Numpy for array related functions
import numpy as np
# Dlib for deep learning based Modules and face landmark detection
import dlib
#face_utils for basic operations of conversion
from imutils import face_utils
import serial
import time
s = serial.Serial('COM3',9600)
#Initializing the camera and taking the instance
cap = cv2.VideoCapture(0)

#Initializing the face detector and landmark detector
hog_face_detector = dlib.get_frontal_face_detector()
predictor =
dlib.shape_predictor("C:\\Users\\chinn\\Downloads\\shape_predictor_68_
face_landmarks.dat")

#status marking for current state
sleep = 0
drowsy = 0
active = 0
status=""
color=(0,0,0)
```

```
def compute(ptA,ptB):
        dist = np.linalg.norm(ptA - ptB)
        return dist

def blinked(a,b,c,d,e,f):
        up = compute(b,d) + compute(c,e)
        down = compute(a,f)
        ratio = up/(2.0*down)

        #Checking if it is blinked
        if(ratio>0.25):
                return 2
        elif(ratio>0.21 and ratio<=0.25):
                return 1
        else:
                return 0



while True:
        _, frame = cap.read()
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        faces = hog_face_detector(gray)
        #detected face in faces array
        for face in faces:
                x1 = face.left()
                y1 = face.top()
                x2 = face.right()
                y2 = face.bottom()
```

```
face_frame = frame.copy()
cv2.rectangle(face_frame, (x1, y1), (x2, y2), (0, 255, 0), 2)

landmarks = predictor(gray, face)
landmarks = face_utils.shape_to_np(landmarks)

#The numbers are actually the landmarks which will show eye
left_blink = blinked(landmarks[36],landmarks[37],
landmarks[38], landmarks[41], landmarks[40], landmarks[39])
right_blink = blinked(landmarks[42],landmarks[43],
landmarks[44], landmarks[47], landmarks[46], landmarks[45])

#Now judge what to do for the eye blinks
if(left_blink==0 or right_blink==0):
        sleep+=1
        drowsy=0
        active=0
        if(sleep>6):
                s.write(b'a')
                time.sleep(2)
                status="SLEEPING !!!"
                color = (0,0,255)

elif(left_blink==1 or right_blink==1):
        sleep=0
        active=0
        drowsy+=1
        if(drowsy>6):
                s.write(b'a')
```

```
                    time.sleep(2)
                    status="Drowsy !"
                    color = (0,0,255)

            else:
                drowsy=0
                sleep=0
                active+=1
                if(active>6):
                    s.write(b'b')
                    time.sleep(2)
                    status="Active :)"
                    color = (0,0,255)

            cv2.putText(frame, status, (100,100),
cv2.FONT_HERSHEY_SIMPLEX, 1.2, color,3)

            for n in range(0, 68):
                (x,y) = landmarks[n]
                cv2.circle(face_frame, (x, y), 1, (255, 255, 255), -1)

        cv2.imshow("Frame", frame)
        #cv2.imshow("Result of detector", face_frame)
        key = cv2.waitKey(1)
        if key == 27:
            break
```

ARDUINO CODE :

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);
const int buzzer_Pin = 8;
const int led_Pin = 9;
char sleep_status = 0;

void setup() {
    lcd.init();                    // initialize the lcd
  lcd.init();
  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(3,0);
  lcd.print("Hello, world!");
  delay(200);
  Serial.begin(9600);
  pinMode(buzzer_Pin, OUTPUT);
  pinMode(led_Pin, OUTPUT);
  lcd.begin(16, 2);
  lcd.print("Driver Sleep ");
  lcd.setCursor(0,2);
  lcd.print("Detection SYSTEM");
  digitalWrite(buzzer_Pin, LOW);
  digitalWrite(led_Pin, LOW);
}
void loop()
{
```

```
  while (Serial.available() > 0)
{
  sleep_status = Serial.read();
  if(sleep_status == 'a')
  {
     lcd.clear();
     lcd.print("Please wake up");
     digitalWrite(buzzer_Pin, HIGH);
     digitalWrite(led_Pin, HIGH);
     delay(2000);
     digitalWrite(buzzer_Pin, LOW);
     digitalWrite(led_Pin, LOW);
     delay(100);
  }
  else if(sleep_status == 'b')
  {
     lcd.clear();
     lcd.print("All Ok");
     lcd.setCursor(0,2);
     lcd.print("Drive Safe");
     digitalWrite(buzzer_Pin, LOW);
     digitalWrite(led_Pin, LOW);
     delay(2000);
  }
  else
  {
    /* Do Nothing */
  }
}}
```

REFERENCES:

[1] Priyanka, S., and S. Shanthi. "A Review on Drowsiness Prediction System using Deep Learning Approaches." In 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), pp. 1079-1084. IEEE, 2022.

[2] Albadawi, Yaman, Maen Takruri, and Mohammed Awad. "A review of recent developments in driver drowsiness detection systems." Sensors 22, no. 5 (2022): 2069.

[3] Quiroz Jr, Fernando E., Renz Joven S. Madeja, and Christian Rick A. Jaro. "Vision-based Drowsiness Detection and Alarm System." (2020).

[4] Dr.S.Priyadarsini, ChahakAgarwal.D, Deshiya Narayan.M. "Driver Drowsiness Detection System Using Raspberry Pi"(2019)

[5] Mehta, Sukrit, Sharad Dadhich, Sahil Gumber, and Arpita Jadhav Bhatt. "Real-time driver drowsiness detection system using eye aspect ratio and eye closure ratio." In Proceedings of international conference on sustainable computing in science, technology and management (SUSCOM), Amity University Rajasthan, Jaipur-India. 2019.

[6] Hossain, Md Yousuf, and Fabian Parsia George. "IOT based real-time drowsy driving detection system for the prevention of road accidents." In 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), vol. 3, pp. 190-195. IEEE, 2018.

[7] Juvale, Hrishikesh B., Anant S. Mahajan, Ashwin A. Bhagwat, Vishal T. Badiger, Ganesh D. Bhutkar, Priyadarshan S. Dhabe, and Manikrao L. Dhore. "Drowsy detection and alarming system (DroDeASys)." In Proceedings of the World Congress on Engineering and Computer Science. 2017.

[8] Roopalakshmi, R., Jayantkumar A. Rathod, Ashwitha S. Shetty, and K. Supriya. "Driver drowsiness detection system based on visual features." In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 1344-1347. IEEE, 2018.

[9] Chowdhury. "Sensor applications and physiological features in drivers' drowsiness detection: A review." IEEE sensors Journal 18, no. 8 (2018): 3055-3067.

[10] Liu. "Convolutional two-stream network using multi-facial feature fusion for driver fatigue detection." Future Internet 11, no. 5 (2019): 115.