DRIVER DROWSINESS DETECTION SYSTEM

MAIN PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF

BACHELOR OF TECHNOLOGY

in

Computer Science and Engineering

of

APJ Abdul Kalam Technological University

by

SANDRA ALICE JOHN (VAS20CS098)

(AN ISO 9001:2015 CERTIFIED INSTITUTION)

Department of Computer Science and Engineering

Vidya Academy of Science & Technology

Thalakottukara, Thrissur - 680 501

(http://www.vidyaacademy.ac.in)

May 2024

Department of Computer Science and Engineering
Vidya Academy of Science & Technology

Thalakottukara, Thrissur - 680 501

(http://www.vidyaacademy.ac.in)

(AN ISO 9001:2015 CERTIFIED INSTITUTION)

Certificate

This is to certify that the Main Project Report titled **"DRIVER DROWSINESS DETECTION SYSTEM"** is a bonafide record of the work carried out by **SANDRA ALICE JOHN (VAS20CS098)** of Vidya Academy of Science & Technology, Thalakkottukara, Thrissur - 680 501 in partial fulfillment of the requirements for the award of **Degree of Bachelor of Technology** in **Computer Science and Engineering** of **APJ Abdul Kalam Technological University**, during the academic year 2023-2024.

Project Guide

Ms. Mrithu A S

Assistant Professor, Dept. of CSE

Project Coordinators

Ms. Jency Babu

Assistant Professor, Dept. of CSE

Ms. Mithu Varghese

Assistant Professor, Dept. of CSE

Head of Department

Dr. Ramani Bai V

Professor, Dept. of CSE

Undertaking

I,

Sandra Alice John hereby undertake that the main project work entitled **"DRIVER DROWSINESS DETECTION SYSTEM"**, is carried out by me independently under the valuable guidance of **Ms. Mrithu A S**, Assistant Professor, Department of Computer Science and Engineering, Vidya Academy of Science and Technology, Thalakottukara, Thrissur in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** of **APJ Abdul Kalam Technological University**, during the academic year 2023-2024.

Thrissur **SANDRA ALICE JOHN (VAS20CS098)**

May 2024

Acknowledgement

I wish to record my indebtedness and thankfulness to all those, who helped me to prepare this report titled **"DRIVER DROWSINESS DETECTION SYSTEM"** and have helped me to present it in a satisfactory manner. I want to start by expressing gratitude to my project guide **Ms. Mrithu A S**, Assistant Professor, Dept. of Computer Science and Engineering, because of her valuable advice and guidance. I received motivation, encouragement and hold-up from her during the course of work.

I would like to thank my project coordinators **Ms. Jensy Babu** and **Ms. Mithu Varghese,** Assistant Professors, Dept. of Computer Science and Engineering, for the wholehearted support and guidance necessary to complete the project work.

I am thankful to **Dr. Ramani Bai V**, Head of Computer Science and Engineering Department, and our Principal **Dr. Saji C B**, for their sole co-operation.

I would like to extend my sincere gratitude to friends who have always been helpful, in preparing and presenting the report and in the discussion following the presentation. Last, but not least I wish to express my gratitude to God Almighty for his abundant blessings without which this effort would not have been successful.

SANDRA ALICE JOHN (VAS20CS098)

Eighth Semester B.Tech (2020 Admissions)

May 2024 Vidya Academy of Science & Technology, Thrissur - 680 501

ABSTRACT

In the present world, lots of road accidents take place due to the lack of attention and alertness of driver. This is termed as driver drowsiness. This leads to a lot of unfortunate situations causing adverse damage to human lives. The main goal of this research is the detection of driver drowsiness and an appropriate response to the detection. There are many methods which are based on the motion of the vehicle or based on the driver's behavior. One of the methods is the physiological method which helps in distracting the driver from drowsiness and making him alert. Other few methods require expensive sensors and deals with a lot of data. Therefore, the proposed system develops a system for detecting drowsiness in real time with proper procedure and accuracy which is acceptable. The driver's facial expressions are captured and recorded using a webcam. Every movement in each frame is detected using few techniques of image processing. The Eye Aspect Ratio, Mouth Opening Ratio, and Nose Length Ratio are calculated using the landmark points on the face. The calculated values are compared to the threshold values developed by the system and the difference in value leads to the detection. At the same time, the machine learning algorithms are also implemented in offline manner. Based on the classification, the system has successfully achieved 95.58% percentage of sensitivity and 100% of specificity using Support Vector Machine. This model system is compatible with all kinds of vehicles.

Contents

List of Figures

List of Symbols and Abbreviations

ISL	Indian Sign Language
SVM	Support Vector Machine
EAR	Ear Aspect ratio
MOR	Mouth Opening Ratio
GPS	Global Positioning System
HOG	Histogram of Oriented Gradients
NLR	Nose Length Ratio

Chapter 1

INTRODUCTION

1.1 General

Many road accidents which lead to death are because of drowsiness while driving. Drivers who drive long hours like truck drivers, bus drivers are likely to experience this problem. It is highly risky to drive with lack of sleep and driving for long hours will be more tiresome. Due to the drowsiness of the driver causes very dangerous consequences, it is estimated that 70,000 to 80,000 injures and crashes happen worldwide in a year. Even deaths have reached 1000-2000 every year. There are many unofficial deaths which are not confirmed by drivers that it was due to their drowsiness. This takes lives of many innocent people. It is a nightmare for a lot of people who travel across world. It is very important to identify the driver drowsiness and alert the driver to prevent crash.

The goal of this system is the detection of the indication of this fatigue of the driver. The acquisition system, processing system and warning system are the three blocks that are present in the detection system. The video of the driver's front face is captured by the acquisition system and it is transferred to the next stage which is, processing block. The detection is processed and if drowsiness of driver is detected, then the warning system gives a warning or alarm.

The methods to detect the drowsiness of the drive may be done using intrusive or non-intrusive method that is, with and without the use of sensors connected to the driver. The cost of the system depends on the sensors used in the system. Addition of more parameters can increase the accuracy of the system to some extent. The motivation for the development of cost effective, and real-time driver drowsiness system with acceptable accuracy are the motivational factors of this work. Hence, the proposed system detects the fatigue of the driver from the facial images, and image processing technology and machine learning method are used to achieve a cost effective and portable system.

1.2 Objectives of the Work

The primary objective of driver drowsiness detection using machine learning is to enhance road safety by developing a proactive system that can identify signs of driver fatigue and intervene before accidents occur. The specific goals include:

Accident Prevention: The main aim is to reduce the number of accidents caused by drowsy driving, which is a significant contributor to road accidents and fatalities.

Early Detection: Implementing machine learning algorithms to analyze driver behavior and facial features allows for the early detection of signs of drowsiness or fatigue.

Real-time Monitoring: The system is designed to monitor the driver's state continuously in real-time, providing timely alerts or warnings if signs of drowsiness are detected.

Intervention Mechanism: Integration with a warning system enables the implementation of interventions, such as alarms or alerts, to prompt the driver to take corrective actions or pull over for rest.

Safety Improvement for Long-Hour Drivers: Targeting individuals with long driving hours, such as truck and bus drivers, the system aims to mitigate the risks associated with extended periods of driving.

Cost-Effective Solutions: Striving to develop a cost-effective solution ensures that the technology is accessible and can be widely adopted, especially in commercial vehicles where long-distance driving is common.

Portable and Practical Implementation: The research aims to create a system that is not only effective but also portable, allowing for practical implementation in various vehicles without significant logistical challenges.

Utilizing Machine Learning: Leveraging machine learning techniques enables the system to learn and adapt to different driving conditions and individual driver behaviors, improving accuracy over time.

1.3 Motivation for this work

- Every year various people lose their lives as a result of destructive road setbacks all through the planet and languid driving is one of the fundamental drivers of road accidents and passing.
- Fatigue and smaller than expected rest at the driving controls are often the fundamental driver of certifiable setbacks.
- Therefore, identification of driver's weakness and its sign is the fundamental objective.

Chapter 2

LITERATURE REVIEW

2.1 Literature Survey

2.1.1 Real-Time Driver-Drowsiness Detection System Using Facial Features

- **Authors**: Wanghua Deng and Ruoxue Wu
- **Source** : IEEE
- **Year** : 2019

This system is a real-time driver-drowsiness detection system. The system analyzes facial features captured through video images to identify signs of fatigue, such as yawning, blinking, and the duration of eye closure. Key features include a new face-tracking algorithm for improved accuracy and a method for detecting facial regions based on 68 key points. By combining features from the eyes and mouth, this paper claims to achieves around 92% accuracy in detecting driver fatigue, providing a valuable tool for alerting drivers and potentially preventing accidents due to drowsy driving.

2.1.2 Drowsiness Detection System using Eye Aspect Ratio Technique

- **Authors** :S. Sathasivam, A. K. Mahamad, S. Saon, A. Sidek, M. M. Som and H. A. Ameen
- **Source** : IEEE
- **Year** : 2020

This paper proposes an image detection drowsiness system using the Eye Aspect Ratio (EAR) technique to address the growing issue of car accidents caused by driver fatigue. Employing a setup with the Pi camera, Raspberry Pi 4, and GPS module, the system continuously monitors the driver's eye closure state in real time. Experimental results, conducted under various conditions including wearing spectacles, dim light, and microsleep scenarios, demonstrate a 90% accuracy in detecting drowsiness. The system aims to significantly enhance driver vigilance, contributing to increased road safety by mitigating the primary cause of car accidents—driver fatigue.

2.1.3 Driver Drowsiness Detection and Alert System

- **Authors** : R Kannan, Palamakula Jahnavi, M Megha
- **Source** : IEEE
- **Year** : 2023

This paper focuses on addressing the global issue of driver fatigue, a leading cause of accidents. A prototype drowsiness detection system has been developed, utilizing Python and Dlib for monitoring the driver's eyes. The system aims to enhance safety through real-time, non-intrusive monitoring, sounding an alarm if signs of drowsiness, such as prolonged eye closure, are detected. The proposal also includes a hardware component based on infrared light to further improve detection accuracy. By combining software and hardware solutions, the project contributes to the advancement of driver safety, aligning with the broader goal of preventing accidents caused by drowsy driving.

2.1.4 Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio

- **Authors** : Sukrit Mehta, Sharad Dadhich, Sahil Gumber and Arpita Jadhav Bhatt
- **Source** : SSRN
- **Year** : 2019

The proposed system addresses the critical issue of drowsy driving, a major cause of road accidents and fatalities worldwide. The study introduces a lightweight, real-time driver's drowsiness detection system implemented as an Android application. Using image processing techniques, the system analyzes facial landmarks, Eye Aspect Ratio (EAR), Mouth Opening Ratio (MOR) and Nose Length Ratio in video frames to identify signs of fatigue. The incorporation of machine learning, specifically the random forest classifier, yields promising results, with the proposed model achieving an 97% accuracy rate. This research presents an innovative and practical approach to detecting driver drowsiness, contributing to ongoing efforts to improve road safety.

Chapter 3

SYSTEM DESIGN

3.1 Methodology

The proposed driver drowsiness system involves capturing video images of the driver and analyzing their facial expression. The system updates the driver's eye variable storage and alerts them if they fall asleep. To prevent road accidents caused by driver fatigue, advanced technology such as video-based monitoring systems can be used. To calculate the eye aspect ratio and the yawn threshold for each frame, six points of the eye and mouth are visualized. The EAR is the ratio of Euclidean distance between vertical and horizontal eye points. The yawn threshold is calculated by subtracting the mean of distance of upper and lower lip from the axis.

Libraries:

NumPy: NumPy stands for Numerical Python. NumPy is a fundamental Python library for numerical computing, offering efficient multi-dimensional array operations, mathematical functions, and linear algebra capabilities. It is widely used in scientific computing, data analysis, and machine learning due to its speed, versatility, and ease of use.

NumPy's array manipulation and mathematical functions are invaluable for driver drowsiness detection systems. It efficiently handles data representation, feature extraction, and model training, contributing to accurate and efficient detection algorithms.

Dlib: Dlib is a C++ toolkit widely used for computer vision tasks. It offers algorithms for facial landmark detection, object detection, and face recognition. With its Python bindings, it's accessible to Python developers. Dlib's robustness and cross-platform compatibility make it suitable for applications like driver drowsiness detection.

Scipy: Scipy is a Python library for scientific computing that extends NumPy's functionality. It includes tools for optimization, integration, interpolation, linear algebra, signal processing, and statistics. With Scipy, users can perform advanced mathematical operations and solve complex scientific problems efficiently in Python.

MediaPipe: MediaPipe is an open-source framework developed by Google that offers solutions for building real-time perception pipelines. It provides ready-to-use machine learning models and components for various tasks such as object detection, pose estimation, hand tracking, and facial recognition. MediaPipe is designed to be efficient, scalable, and easy to use, making it suitable for a wide range of applications, including augmented reality, gesture recognition, and human-computer interaction.

cv2: cv2, short for OpenCV (Open Source Computer Vision Library), is a popular open-source library for computer vision and image processing tasks in Python. It offers a wide range of functions and algorithms for tasks such as image manipulation, object detection, feature extraction, video analysis, and more. With cv2, developers can perform various operations on images and videos, making it a versatile tool for applications ranging from simple image editing to complex computer vision projects.

imutils: Imutils is a convenience library in Python designed to make common image processing tasks easier with OpenCV. It simplifies tasks such as resizing, rotating, and displaying images, as well as converting between different color spaces and drawing shapes on images. Imutils provides a collection of functions that streamline the use of

OpenCV, making it a valuable tool for rapid prototyping and development of computer vision applications.

3.2 System Architecture

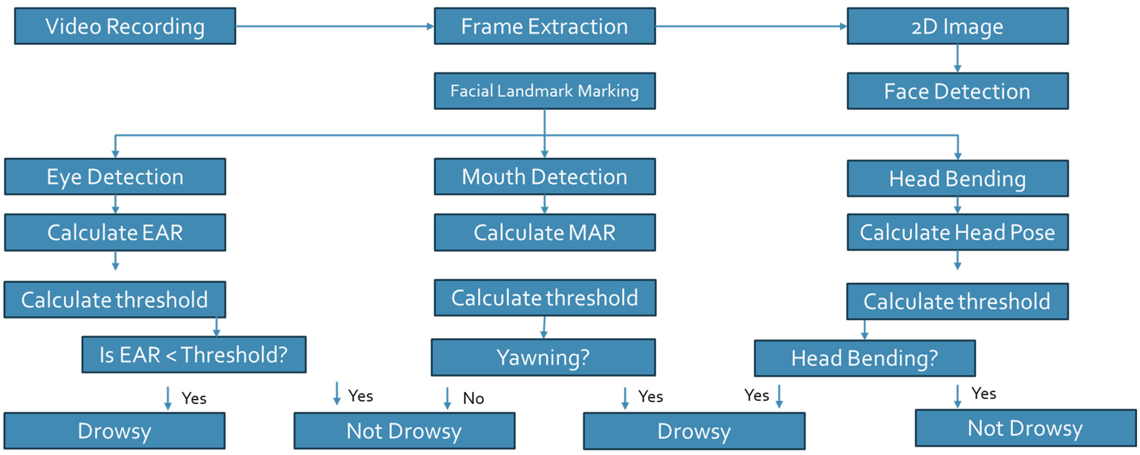


Figure 3.1: System Architecture.

The Figure 3.2 depicts the system architecture for drowsiness detection system using machine learning. The system architecture for driver drowsiness detection is a multi-step process designed to effectively monitor and prevent driver fatigue-related accidents. It begins with the capture of real-time video input from a camera positioned in front of the driver. The video feed is then subjected to facial expression analysis, with a specific focus on eye status. The analysis involves the use of image processing techniques to detect eye closure, yawning and face bending, which are key indicators of drowsiness.

The algorithm employed in the system calculates the Eye Aspect Ratio (EAR), yawn threshold and head pose estimation to determine the driver's drowsiness status. The EAR is a measure of the ratio of the Euclidean distances between specific coordinates on the eyes, while the yawn threshold is calculated based on the distances of coordinates of the upper lip and lower lip from a reference axis and the head pose is calculated using geometric techniques based on the detected 3D coordinates of facial landmarks. This involves solving the Perspective-n-Point (PnP) problem to estimate the head's pose relative to the camera. These calculations enable the system to accurately assess the driver's level of drowsiness.

In the event that drowsiness is detected, the system triggers alerts to notify the driver, thereby providing a proactive mechanism to prevent potential accidents. The architecture of the system integrates advanced technology, such as video-based monitoring systems, to enhance road safety. By leveraging these technologies and algorithms, the system aims to effectively monitor and address driver fatigue, ultimately contributing to the reduction of road accidents caused by drowsiness.

The comprehensive approach underscores the significance of leveraging advanced technologies and machine learning algorithms to develop effective drowsiness detection systems, with the overarching goal of saving lives and preventing injuries on the roads.

3.3 Data Flow Diagram

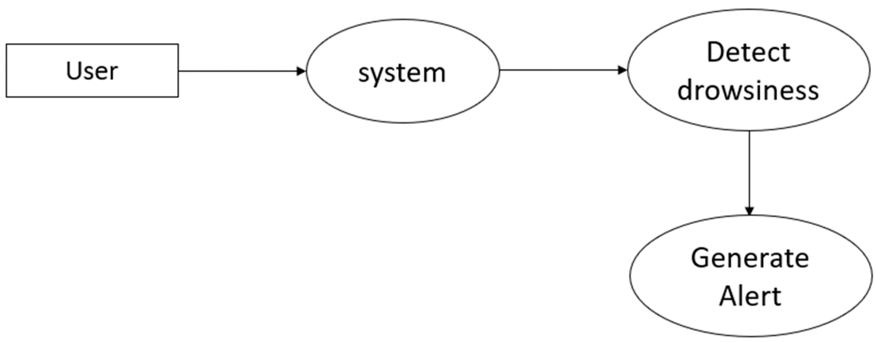


Figure 3.2: Data Flow Diagram Level 1.

.

The Figure 3.2 and Figure 3.3 shows the data flow diagram of drowsiness detection system using machine learning. The data flow diagram for the driver drowsiness detection system illustrates the flow of information and processes within the system. It typically consists of processes, data stores, data flow, and external entities.

Level 1: Provides a high-level overview of the entire system.

Main Processes in level 1 Data Flow Diagram includes:

- Input Processing
- Drowsiness Detection
- Alert Activation

External Entities in level 1 Data Flow Diagram includes:

- Driver
- Vehicle Camera
- Alert System

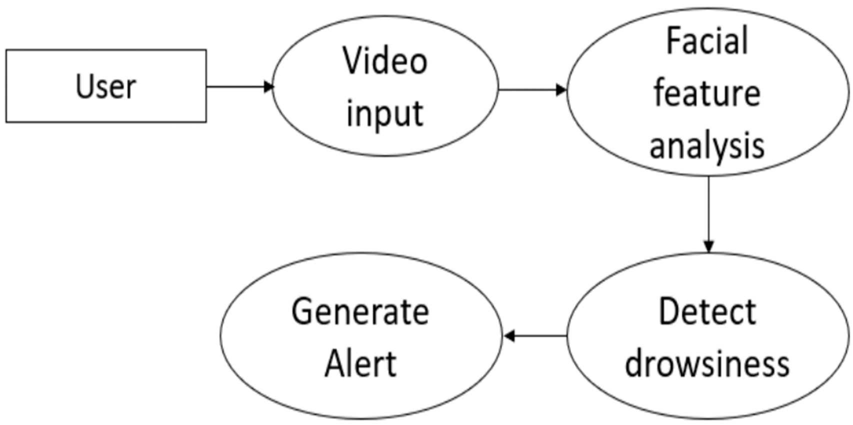


Figure 3.3: Data Flow Diagram Level 2.

Level 2: Expands on the processes from Level 1 and provides more detail.

Main Processes in level 1 Data Flow Diagram includes:

- Input Processing
- Facial Detection
- Landmark Detection
- Feature Extraction
- Drowsiness Detection

- Alert Activation

External Entities in level 1 Data Flow Diagram includes:

- Driver
- Vehicle Camera
- Alert System

In the context of driver drowsiness detection, the data flow diagram would depict the flow of video input, facial expression analysis, and drowsiness status determination.

Processes: These represent the specific tasks or operations within the system. In the context of driver drowsiness detection, processes may include capturing real-time video, facial expression analysis, eye status analysis, and drowsiness status determination.

Data Stores: These are repositories of data within the system. In this context, data stores may include storage for video input, facial expression data, and drowsiness status records.

Data Flow: This represents the flow of data between processes and data stores. For driver drowsiness detection, data flow would illustrate the movement of video input through the facial expression analysis process, the eye status analysis process, and ultimately to the drowsiness status determination process.

External Entities: These are sources or destinations of data outside the system. In the context of driver drowsiness detection, external entities may include the camera capturing the video input and the driver who receives alerts based on the drowsiness status.

The data flow diagram provides a visual representation of how data moves through the system, from input capture to the determination of drowsiness status and subsequent alerts. It serves as a valuable tool for understanding the flow of information and processes within the driver drowsiness detection system.

3.4 Use Case Diagram

The use case diagram illustrates the flow of interactions between the driver, the camera capturing facial features, the facial feature analysis module detecting drowsiness, the

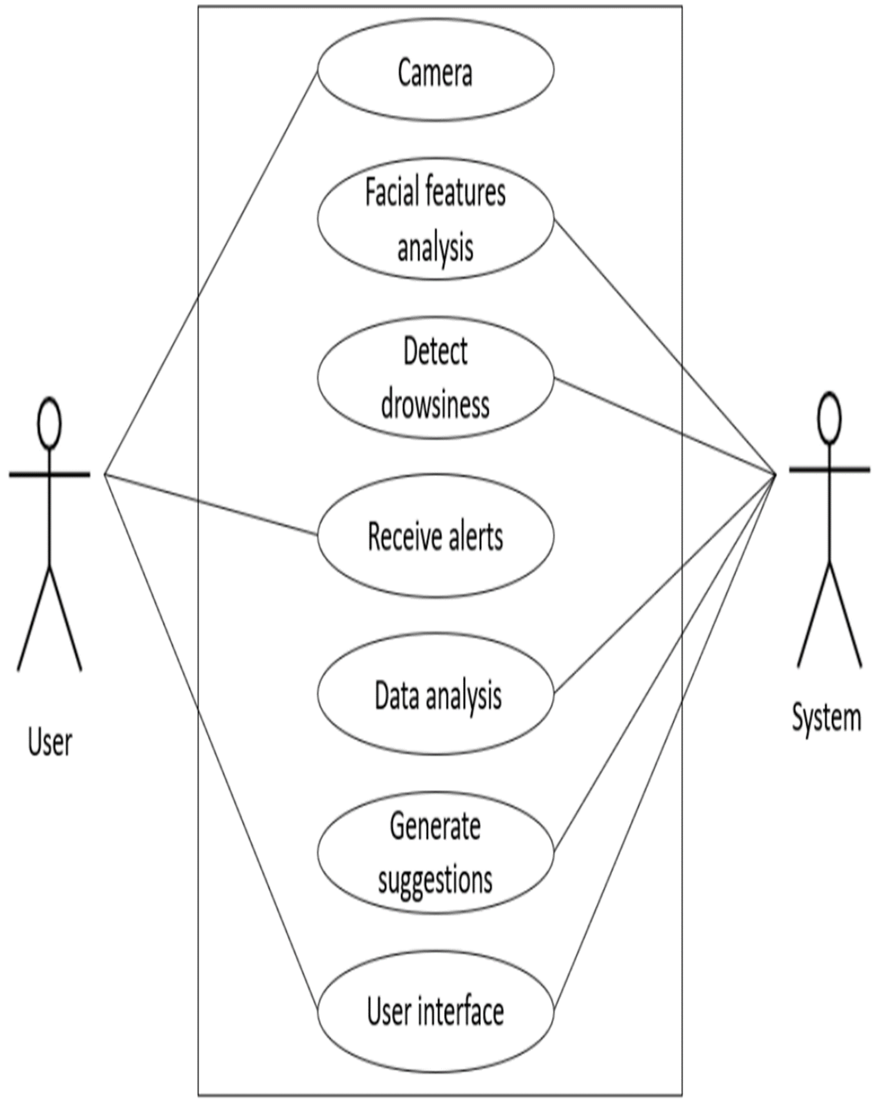


Figure 3.4: Use-Case Diagram.

alert system triggering alerts, and the data analysis module processing and analyzing drowsiness data.

Chapter 4

IMPLEMENTATION

4.1 Requirements

4.1.1 Visual Studio Code

Visual Studio Code, commonly abbreviated as VS Code, is a versatile and lightweight code editor developed by Microsoft. It stands out for its cross-platform compatibility, making it accessible on Windows, macOS, and Linux operating systems. With its clean and intuitive user interface, developers benefit from features like syntax highlighting, code completion, and code folding, enhancing productivity. Moreover, VS Code integrates seamlessly with Git, facilitating version control directly within the editor. Its extensibility is another highlight, supporting a vast array of extensions for language support, debugging tools, and themes. Furthermore, developers appreciate its integrated terminal, enabling them to execute shell commands and scripts without leaving the editor. Combined with built-in debugging support, task automation, and IntelliSense functionality, Visual Studio Code offers a robust development environment suitable for various programming tasks.

4.1.2 Python

Python is a versatile and widely-used programming language known for its simplicity and readability. It's favored by developers for its ease of learning and extensive libraries, making it suitable for various applications, including web development, data analysis, artificial intelligence, and scientific computing. Python's syntax emphasizes readability and clarity, with indentation serving as a fundamental aspect of its structure. With a vast ecosystem of third-party packages and frameworks such as Django, Flask, NumPy, and TensorFlow, Python empowers developers to build complex and innovative solutions efficiently. Its interpreted nature allows for rapid prototyping and iterative development, while its cross-platform compatibility ensures code can run seamlessly on different operating systems. Overall, Python's versatility, simplicity, and robust community support make it a go-to choice for developers across industries and skill levels.

Python is commonly used in driver drowsiness detection systems due to its versatility, extensive libraries, and ease of use. In such systems, Python is employed for various tasks, including:

- Image Processing: Python's libraries such as OpenCV are utilized for processing the video feed captured by the camera installed in the vehicle. Image processing techniques are applied to detect and track the driver's face and facial landmarks.

- Machine Learning: Python's machine learning libraries like scikit-learn and TensorFlow are utilized to train models for detecting signs of drowsiness based on extracted features such as eye closure, yawn, and head movements. These models can classify whether the driver is alert or drowsy based on the input data.

- Data Analysis: Python's data analysis libraries such as pandas and NumPy are used to analyze and preprocess the data collected from the driver's behavior, including the frequency and duration of eye closure, yawns, and head movements.

- Alert Mechanism: Python scripts are employed to activate alert mechanisms when drowsiness is detected. This could include generating audible alarms, displaying visual alerts on the dashboard, or triggering haptic feedback through steering wheel vibrations.

- System Integration: Python facilitates the integration of various components within the drowsiness detection system, such as capturing input from the camera, processing the data, running machine learning algorithms, and activating alerts, providing a cohesive solution.

4.1.3 Anaconda

Anaconda is a popular open-source distribution of the Python and R programming languages for data science and machine learning tasks. It provides a comprehensive package management system and environment management tool, making it easier for developers and data scientists to install, manage, and deploy various libraries and packages required for their projects.

Here are some key features of Anaconda:

Package Management: Anaconda includes the Conda package manager, which allows users to easily install, update, and remove packages from their Python or R environments. Conda handles dependencies and ensures compatibility between different packages, simplifying the process of setting up development environments.

- Environment Management: With Anaconda, users can create isolated environments to manage project dependencies and avoid conflicts between different packages or versions. This helps ensure reproducibility and maintainability of projects by encapsulating all dependencies within a specific environment.

- Pre-installed Libraries: Anaconda comes with a wide range of pre-installed libraries and tools commonly used in data science and machine learning, including NumPy, pandas, scikit-learn, TensorFlow, and Jupyter Notebook. This eliminates the need for manual installation and configuration of these libraries, allowing users to start working on their projects immediately.

- Cross-Platform Compatibility: Anaconda is available for Windows, macOS, and Linux operating systems, making it suitable for developers and data scientists working on different platforms.

- Integration with Jupyter Notebooks: Anaconda seamlessly integrates with Jupyter Notebooks, providing an interactive computing environment for data analysis, visualization, and documentation. Users can create and share Jupyter notebooks directly from the Anaconda interface.

- Community Support: Anaconda has a large and active community of users and contributors who provide support, tutorials, and resources for getting started with data science and machine learning using Anaconda.

4.1.4 OpenCV

OpenCV is an open-source computer vision and machine learning library that provides tools for image and video processing, object detection, feature extraction, and more. It supports various platforms and offers pre-trained models for common tasks like object detection and recognition.

OpenCV plays a crucial role in drowsiness detection systems, particularly in analyzing video feeds to identify signs of drowsiness in drivers. Here's how OpenCV can be utilized in such systems:

- Face Detection: OpenCV provides robust face detection algorithms, such as Haar cascades or deep learning-based methods, which can locate faces within video frames. Detecting the driver's face is the first step in analyzing their behavior.

- Facial Landmark Detection: OpenCV can be used to detect facial landmarks, such as eye corners, nose, and mouth, which are essential for assessing drowsiness indicators like eye closure or head movement.

- Eye Tracking: By tracking the driver's eye movements, OpenCV can detect patterns such as prolonged eye closure or slow eyelid movements, indicating drowsiness.

- Image Processing: OpenCV offers a wide range of image processing functions that can enhance the quality of video frames, remove noise, and improve the accuracy of feature detection algorithms.

- Alert Generation: OpenCV can trigger alerts or warnings when signs of drowsiness are detected, such as sounding alarms, vibrating the driver's seat, or displaying visual cues on a dashboard.
- Real-time Processing: OpenCV's efficient algorithms allow for real-time processing of video streams, enabling continuous monitoring of the driver's behavior and immediate response to drowsiness indicators.
- Integration with Machine Learning: OpenCV can be integrated with machine learning models trained to recognize drowsiness patterns from video data. This combination enhances the accuracy and reliability of drowsiness detection systems.

4.1.5 TensorFlow

TensorFlow is an open-source machine learning framework developed by Google for building and training machine learning models. It provides a comprehensive ecosystem of tools, libraries, and resources to support various machine learning tasks, including deep learning, neural networks, and numerical computation.

Key features of TensorFlow include:

- Flexibility: TensorFlow offers flexibility in building machine learning models, supporting a wide range of tasks from simple linear regression to complex deep learning architectures.
- High-Level APIs: TensorFlow provides high-level APIs such as Keras, which simplifies the process of building and training neural networks. Keras offers a user-friendly interface for defining and configuring neural network layers, making it accessible to beginners and experts alike.
- Scalability: TensorFlow is designed to scale seamlessly from individual devices to large distributed systems. It supports distributed training across multiple CPUs, GPUs, and TPUs (Tensor Processing Units), enabling users to train models on large datasets efficiently.

- TensorFlow Extended (TFX): TensorFlow Extended is a platform for deploying and managing machine learning models in production. It includes components for data validation, model analysis, and serving, providing end-to-end support for the machine learning lifecycle.

- TensorBoard: TensorFlow includes TensorBoard, a visualization toolkit for visualizing and monitoring the training process and model performance. TensorBoard offers interactive dashboards for analyzing metrics, visualizing computational graphs, and debugging models.

- Community and Ecosystem: TensorFlow has a vibrant community of developers, researchers, and contributors who actively contribute to the framework's development. The TensorFlow ecosystem includes libraries and tools for tasks such as data preprocessing, model interpretation, and reinforcement learning.

- Support for Different Platforms: TensorFlow supports multiple programming languages, including Python, C++, and JavaScript, making it accessible to developers working in different environments.

4.2 Acquiring Data:

At first, the video is captured and recorded using a laptop webcam and then the frames are released and proposed in a laptop. Then after the completion of extraction of frames, the picture implementation techniques are proposed on the 2-dimensional picture of the recorded video. Now, the required driver information is produced. The volunteers are asked to focus on the laptop webcam and perform activities like continual eye blinking and closing, mouth yawning, and head bending. The webcam is adjusted to capture the video for 20 to 30 min.

4.3 Face Identification:

The driver's face is identified first after the frames have been extracted. The proposed system, Haar Cascade and SVM algorithms are used for face extraction. Following that, the negative examples of same size are considered for HOG descriptors and then results are evaluated. Typically, the count of negative examples is more than positive examples. Then after getting the characteristics of both the groups, a direct SVM algorithm is used for classifying the required task. To get more detection and better accuracy for SVM algorithm, strong negatives are used. In this detection, after the guidance, the SVM classifier is investigated on labeled data and the incorrect positive example characteristic values are reused for guidance purpose. To test the picture, the window used for positive examples are rendered over the image and then the required result or output will be classified for each window location. Then based on the results obtained and considering the various sample results, the higher value samples are considered for drowsiness identification and the boundary outline is drawn on the face. This minimum elimination steps will reduce the overlapping and redundant bounding in the bounding box.

4.4 Locating Face Points:

After the detection of faces, the further step is to locate the points on the human face like eyes, mouth, nose etc. Then the photograph that used for face detection has to be normalized to minimize the distance factor between camera and the driver. Therefore, the face photograph is resized with the width of 500 pixels and converted into gray scale pictures. Then normalization is done for regression trees. It will be approximate the positions of location points on the face from pixel intensities. In this process, the square error loss is reduced by using gradient boost learning. Different priors are used to discover various structures. By considering this procedure, the location of the boundaries of eyes, mouth and nose are noted and shown in Table I. Then the location points on face are marked and shown in fig.2. The red points on the figure are used for further identification.

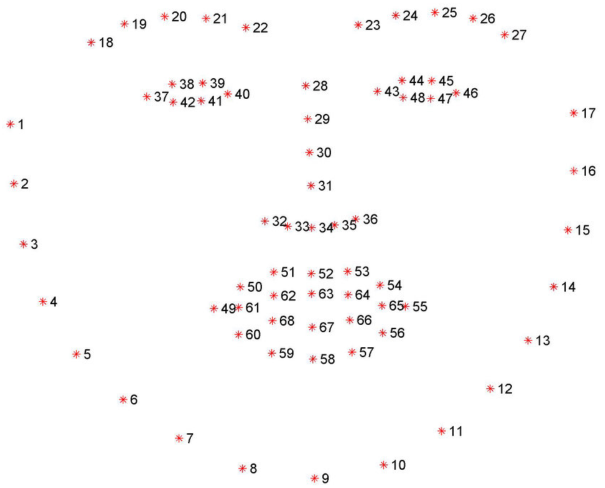


Figure 4.1: Facial Landmarks

Parts	Landmark Points
Mouth	[13-24]
Right eye	[1-6]
Left eye	[7-12]
Nose	[25-28]

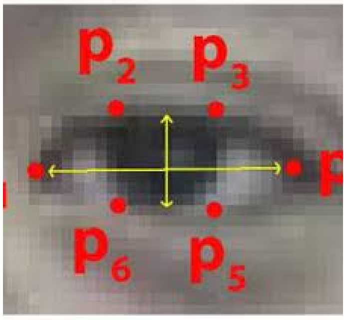
Table I: The location points on face

4.5 Feature Extraction:

After marking the points on face, the drowsiness features are calculated as given below. Eye Aspect Ratio (EAR): From the boundary points on the eyes, the EAR is evaluated as inverse ratio of width of eyes to the height of eyes. The mathematical formula for EAR is given as,

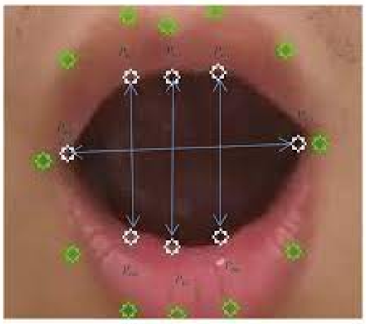
$$EAR = \frac{(p_2 - p_6) + (p_3 - p_5)}{2(p_4 - p_1)}$$

where i is the point marked as i in facial landmarking (i-j) gives the distance between the points i and j. When the eyes are open EAR is max and when the eyes are closed, the EAR is approximated to zero. So, the decreasing value of EAR indicates the closing of eyes and identifies the drowsy behavior of the driver. Mouth Opening Ratio (MOR):



Mouth Opening Ratio is defined as the identification of yawning of the driver and ultimately drowsiness detection is done. The mathematical formula for calculating the MOR is given as,

$$MOR = \frac{(p_{15} - p_{23}) + (p_{16} - p_{22}) + (p_{17} - p_{21})}{3(p_{19} - p_{13})}$$



The MOR ratio is maximum when the mouth of driver is open and if it remains same for some time, then yawning condition is indicated and if it decreases towards zero, then the condition is considered as normal, and not yawning. Nose Length Ratio (NLR): When the head of the driver is down along with vertical axis, then by using the bending angle of head the drowsiness alert is identified. The focal plane of the webcam is directly proportional to head bending, and by using this head bending is calculated. Normally, the nose makes an acute angle with the webcam. The acute angle increases when head moves upward and vice versa. Therefore, the nose length ratio is given as nose length to average length of nose, which also measures the head bending. The mathematical formula for NLR is given as,

$$NLR = \frac{nose\ length\ (p_{28} - p_{25})}{average\ nose\ length}$$

4.6 Classification:

After obtaining the values of eyes, mouth and nose, the next step is to identify the drowsiness from the SVM frames. The flexible value of threshold is required for the drowsiness. Then, algorithms like SVM are used for the classification of the information.

For obtaining the threshold values for eyes, mouth and nose the initial condition of the driver is assumed to be in normal state. This step is known as setup phase. The EAR for nearly 100 to 300 frames will be recorded, and from this an average of 150 to 200 higher values are recognized as strong threshold values for EAR. The values which are high, in which eyes are not closed is considered. If threshold is more than the tested value, then it is identified as eyes closed. For every person, the size of eye will be different; so, this makes the impact to decrease the setup for person to person. For computing MOR values, the threshold value of frames is calculated based on the condition that the mouth is not open. If the threshold is less than the test value that is, when the mouth is open, the yawning is identified. Using the head bending feature, the angle between access and head can be determined in terms of nose length ratio. NLR values range from 0.8 to 1 in normal condition of head and varies with head bending up and down. The average of the NLR is measured as mean of lengths of nose in the setup assuming that head is not bend. After obtaining the threshold values, it is tested. Then, if at least one of the eyes, mouth and nose identifiers are not satisfied, then drowsiness alert is indicated. In practical, for example out of 75 frames at least 70 frames satisfy drowsiness conditions for one or more features, then the overall system is identified as drowsiness detection and the driver is alerted with the alarm sound.

To overcome the thresholding problem, a single value of threshold is considered, and this threshold value depend on EAR. To obtain the average value of EAR, 150 higher EAR values from 300 frames are considered. If the threshold value is more than EAR value, then the driver is at danger. By considering yawn and head bending, this EAR threshold will be increased, and it is dispersed into more frames. The yawning and head bending frames are combined to get flexible value of threshold. If EAR value is more than obtained threshold value, then that condition is treated as drowsiness, and this indicates to alert. In head bending situation, if the head is down, then the frames are considered for a drowsiness alert.

The machine learning algorithms and the threshold factors are used for the identification of driver's drowsiness, from the values obtained from EAR, MOR, and NLR. Earlier, these features were used for the analysis of classification from feature space to individual. However, here it is used for the principal component analysis.

After converting the values obtained from threshold, whether the features are significant for classes or not is tested. If three factors give five percent significance, this classification based on Bayesian Classifier and SVM algorithm is used.

4.7 Design

- Data Acquisition: Use a camera to capture real-time facial images of the driver. Optionally, utilize a microphone to record audio signals for yawn detection. Gather data from inertial sensors (e.g., accelerometers) to monitor head movements.

- Preprocessing: Apply basic image processing techniques to detect and extract facial landmarks, including eye and mouth regions. Analyze audio signals to detect patterns indicative of yawning. Process data from inertial sensors to measure head movements. Feature Extraction:

- Extract features from the preprocessed data relevant to eye closure, yawning, and head bending detection. Features may include eye closure duration, mouth opening width, and head tilt angle.

- Machine Learning Models: Train simple machine learning classifiers, such as Support Vector Machines (SVMs) using labeled data. Develop separate models for eye closure, yawning, and head bending detection based on the extracted features.

- Decision Fusion: Combine the outputs of individual detection models to make a collective decision about the driver's drowsiness level. Use a straightforward decision rule, such as a voting scheme or threshold-based approach, to determine drowsiness.
- Alerting Mechanism: Implement a basic alerting mechanism to notify the driver when drowsiness is detected. Provide simple visual or auditory alerts, such as flashing lights or beeping sounds, to grab the driver's attention.
- Real-Time Processing: Ensure that the system can process data and make drowsiness predictions in real-time to enable timely alerts. Optimize algorithms for efficiency to minimize processing delays.
- Testing and Validation: Conduct basic testing and validation of the system using simulated scenarios or controlled experiments. Evaluate the system's performance in terms of accuracy and responsiveness.
- Deployment and Maintenance: Deploy the system in a test environment, such as a driving simulator or controlled vehicle setting. Monitor the system's performance and collect feedback for further improvements. Maintain the system by updating models and algorithms as needed based on real-world usage.

Chapter 5

EVALUATION AND RESULT

5.1 Result

The output of the system is dynamically generated based on the real-time analysis of the driver's actions. If the system detects prolonged periods of eye closure, frequent yawning, or excessive head bending beyond predetermined thresholds, it triggers an alert mechanism to notify the driver of their drowsy state. This alert can take various forms, including audible alarms, visual alerts displayed on the vehicle's dashboard or windshield, or haptic feedback through steering wheel vibrations or seat vibrations.

The severity of the alert and the type of notification can be tailored based on the detected level of drowsiness and the driver's responsiveness to previous alerts. For example, if the system detects mild drowsiness, it may activate a subtle visual alert on the dashboard, whereas in cases of severe drowsiness, it may trigger a louder audible alarm accompanied by steering wheel vibrations to ensure the driver's immediate attention.

Additionally, the system may log information about drowsiness events, including timestamps and the severity of detected drowsiness, for further analysis or reporting purposes. This data logging enables post-analysis of driving patterns and drowsiness trends, which can be used to refine the system's algorithms and improve its effectiveness over time.

Overall, the system for driver drowsiness detection serves as a proactive safety measure, continuously monitoring the driver's behavior and providing timely alerts to prevent accidents resulting from fatigue-induced impairment. By leveraging multiple detection components and adaptive alert mechanisms, the system enhances road safety and contributes to a safer driving experience for all road users.

Figure 5.1, Figure 5.2 and Figure 5.3 shows normal states of a driver where the driver is looking forward or left and right to the mirrors. Figure 5.4 and Figure 5.5 depicts drowsy state of driver where the head of the driver is bent in the upward or downward directions. Figure 5.6 shows drowsy state of driver where eye closure is detected and Figure 5.7 shows drowsy state of driver where yawns are detected.

Figure 5.1: Normal state: Looking Forward

Figure 5.2: Normal state: Looking Left

Figure 5.3: Normal state: Looking Right

Figure 5.4: Drowsy state: Looking Up

Figure 5.5: Drowsy state: Looking Down

Figure 5.6: Drowsy state: Drowsiness Alert

Figure 5.7: Drowsy state: Yawn Alert

INPUT	EYE DETECTION FREQUENCY	YAWN DETECTION FREQUENCY	NOSE LENGTH DETECTION FREQUENCY	DROWSINESS DETECTION FREQUENCY
1	100%	100%	100%	100%
2	95%	90%	100%	95%
3	100%	95%	92%	95.67%
4	100%	100%	90%	96.67%
5	100%	100%	95%	98.33%

Figure 5.8: Table: Result

Chapter 6

CONCLUSION AND SCOPE OF FUTURE WORK

6.1 Conclusion

The driver drowsiness detection using machine learning algorithms underscores the critical importance of addressing driver fatigue as a significant cause of road accidents. The proposed algorithm, which focuses on image processing techniques and physiological signals, represents a significant advancement in the field of driver drowsiness detection. By calculating eye aspect ratio and yawn threshold, the algorithm aims to effectively detect drowsiness and sound alarms when necessary, thereby contributing to the overall goal of minimizing accidents caused by driver drowsiness.

Furthermore, this emphasizes the need for continued research and development in this area, as it directly impacts public safety and has the potential to significantly reduce the number of accidents on the roads. By referencing related studies and existing methodologies, the research provides a comprehensive overview of the current state of the field and highlights the ongoing efforts to address this critical issue.

In conclusion, the project serves as a valuable contribution to the ongoing efforts to improve road safety and reduce the impact of driver drowsiness on accidents. It underscores the significance of leveraging machine learning algorithms and advanced technologies to develop effective drowsiness detection systems, ultimately aiming to save lives and prevent injuries on the roads.

6.2 Future Scope

The proposed system can be loosened up further to have plentiful security features, for instance, simply a specific number of people can have experts get to or work the vehicle. If there ought to be an event of an undertaking to burglary, the vehicle engine doesn't start or an alert sounds. An image of the robber is taken in an endeavored burglary and shipped off the proprietor of the vehicle who can enlist an argument against the cheat of the vehicle.

Bibliography

[1] W. Deng and R. Wu, ”Real-Time Driver-Drowsiness Detection System Using Facial Features,” in IEEE Access, vol. 7, pp. 118727-118738, 2019, doi: 10.1109/ACCESS.2019.2936663.

[2] S. Sathasivam, A. K. Mahamad, S. Saon, A. Sidek, M. M. Som and H. A. Ameen, ”Drowsiness Detection System using Eye Aspect Ratio Technique,” 2020 IEEE Student Conference on Research and Development (SCOReD), Batu Pahat, Malaysia, 2020, pp. 448-452, doi: 10.1109/SCOReD50371.2020.9251035.

[3] R. Kannan, P. Jahnavi and M. Megha, ”Driver Drowsiness Detection and Alert System,” 2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS), Raichur, India, 2023, pp. 1-5, doi: 10.1109/ICICACS57338.2023.10100316.

[4] Mehta, Sukrit and Dadhich, Sharad and Gumber, Sahil and Jadhav Bhatt, Arpita, Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio (March 20, 2019). Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India, February 26-28, 2019.

[5] Jongseong Gwak, Akinari Hirao and Motoki Shino, “An Investigation of Early Detection of Driver Drowsiness Using Ensemble Machine Learning Based on Hybrid Sensing”, Institute of Industrial Science, The University of Tokyo, Japan, 2020.

[6] Prakash Choudhary, Rahul Sharma, Gautam Singh, Smarjeet Das, Professor Sumedh G. Dhengre, “A Survey Paper on Drowsiness Detection & Alarm System

for Drivers", International Research Journal of Engineering and Technology, Pune, Maharashtra, India, 2016.

[7] Singh Himani Parmar, Mehul Jajal, Yadav Priyanka Brijbhan, "Drowsy Driver Warning System Using Image Processing", Electronics & Communication, GEC, Bharuch, Gujarat.

[8] Adrian Rosebrock, "Drowsiness Detection with OpenCV", pyimagesearch.com/2017/05/08/drowsiness- detectionopencv/, 2017.

Department of Computer Science and Engineering

Vidya Academy of Science & Technology

Thalakkottukara, Thrissur - 680 501

(http://www.vidyaacademy.ac.in)