Driver Drowsiness Detection System using Computer Vision Transformers (RT-DeTr)

Major Project Report

Submitted in the partial fulfilment of the
Requirements for the award of the degree of
BACHELOR OF TECHNOLOGY
IN COMPUTER ENGINEERING

Submitted By
Aadil M. Husain (20BCS073)
Ebadul Islam Farooqi (20BCS019)

Under the supervision of
Dr. Sarfaraz Masood
(Associate Professor)

DEPARTMENT OF COMPUTER ENGINEERING
FACULTY OF ENGINEERING & TECHNOLOGY
JAMIA MILLIA ISLAMIA, NEW DELHI (110025)
(YEAR 2024)

DECLARATION

I, Aadil Mohammad Husain (20BCS073), and Ebadul Islam Farooqi (20BCS019), declare that the project work entitled **"Driver Drowsiness Detection using Computer Vision Transformers"** submitted in partial fulfilment of the degree of Bachelors in Technology, is a record of our original work carried out under the supervision and guidance of **Dr. Sarfaraz Masood**, Associate Professor, Department of Computer Engineering, Jamia Millia Islamia, New Delhi. The work presented in this project is authentic, and any external sources utilized have been duly acknowledged.

We declare that this project has not been submitted previously in part or in full for the award of any degree or diploma in this institute or any other university. All the data, figures, and concepts taken from other sources have been appropriately cited and referenced.

Aadil M. Husain
(20BCS073)

Ebadul Islam Farooqi
(20BCS019)

DEPARTMENT OF COMPUTER ENGINEERING
FACULTY OF ENGINEERING & TECHNOLOGY
JAMIA MILLIA ISLAMIA
NEW DELHI (110025)

CERTIFICATE

This is to certify that the research work entitled "Driver Drowsiness Detection using Computer Vision Transformers" conducted by

- **Aadil Mohammad Husain** (Roll No. 20BCS073) and
- **Ebadul Islam Farooqi** (Roll No. 20BCS019)

is an authentic and original endeavour carried out under the mentorship and guidance in the Department of Computer Engineering at Jamia Millia Islamia, New Delhi.

The aforementioned project has been pursued diligently and ethically by the students as a part of their Bachelor of Engineering curriculum in Computer Engineering at Jamia Millia Islamia, during the academic year 2023-24. This project is presented as a culmination of their academic pursuits and to fulfil the requisites for the successful completion of their degree program.

Dr. Sarfaraz Masood
(Associate Professor)
Department of Computer Engineering
Faculty of Engineering & Technology
JAMIA MILLIA ISLAMIA
NEW DELHI

Prof. Bashir Alam
(Head of the Department)
Department of Computer Engineering
Faculty of Engineering & Technology
JAMIA MILLIA ISLAMIA
NEW DELHI

ACKNOWLEDGEMENT

We extend our heartfelt gratitude to **Dr. Sarfaraz Masood**, Associate Professor, Department of Computer Engineering, Jamia Millia Islamia, New Delhi, for his unwavering support, expert guidance, and encouragement throughout the research endeavour. His profound knowledge, insightful inputs, and constant motivation were instrumental in shaping this project.

We would like to express our sincere appreciation to **Prof. Bashir Alam**, Head of the Department of Computer Engineering, for his continuous support, invaluable advice, and belief in our capabilities. His encouragement and willingness to provide necessary resources have been pivotal in the successful completion of this project.

We are indebted to the entire faculty of the Department of Computer Engineering at Jamia Millia Islamia for imparting us with comprehensive knowledge and fostering an environment conducive to learning and innovation. Their teachings, guidance, and mentorship have been invaluable in shaping our academic journey.

Furthermore, we extend our heartfelt thanks to our classmates and friends for their support, valuable insights, and continuous encouragement throughout this endeavor. Their constructive feedback and discussions have been instrumental in refining our ideas and approach.

We would like to apologize for any unintentional omissions of names or contributions. The contributions of all involved are deeply appreciated and acknowledged.

Aadil M Husain
(20BCS073)

Ebadul Islam Farooqi
(20BCS019)

ABSTRACT

Driver Drowsiness Detection (DDD) is a crucial area of research aimed at reducing road accidents and fatalities caused by drowsy driving. To address this issue, researchers have developed various methods for detecting driver drowsiness, including those based on biomedical signals, driver performance, and computer vision techniques applied to the human face. By detecting drowsiness in real-time, systems can alert drivers and take appropriate measures to prevent accidents, contributing to the reduction of traffic accidents and promoting road safety. With the development in the field of Computer Vision Technique, implementing Driver Drowsiness Detection System has become more economical and easier. Hence we aim to achieve the same in this project.

Keywords: Driver Drowsiness Detection, Road Safety, Computer Vision, Computer Vision Transformers, Object Detection, Object Classification, Real-Time Detection Transformer, Accuracy, Precision, Recall.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Background

Driving is an essential aspect of modern life, providing individuals with autonomy and mobility. However, the safety of drivers and passengers alike is constantly threatened by the pervasive issue of driver drowsiness. The detrimental effects of drowsy driving are well-documented, contributing to thousands of accidents and fatalities worldwide each year. According to the National Highway Traffic Safety Administration (NHTSA), drowsy driving results in an estimated 72,000 crashes, 44,000 injuries, and 800 fatalities annually in the United States alone. These alarming statistics underscore the urgent need for effective drowsiness detection systems to mitigate the risks associated with driver fatigue.

Traditional methods of detecting driver drowsiness primarily rely on physiological signals such as eyelid movements, head position, and steering behavior. While these approaches have shown some degree of success, they often suffer from limitations such as reliance on intrusive sensors, susceptibility to environmental noise, and inability to provide real-time alerts. Furthermore, these methods may not be suitable for all drivers, as individual variations in physiological responses can impact the accuracy of detection.

Fig 1. Some statistics regarding damage caused by Driving Drowsy

Recent advancements in computer vision technology offer promising alternatives for drowsiness detection in drivers. By leveraging deep learning algorithms and image processing techniques, computer vision systems can analyze facial features and patterns indicative of drowsiness in real-time video streams. This non-intrusive approach has the

potential to overcome many of the limitations associated with traditional methods, providing a more robust and reliable solution for detecting driver fatigue.

In recent years, transformers have emerged as a powerful architecture for various computer vision tasks, demonstrating state-of-the-art performance in image classification, object detection, and semantic segmentation. Unlike traditional convolutional neural networks (CNNs), transformers rely on self-attention mechanisms to capture global dependencies within an input sequence, making them particularly well-suited for tasks requiring long-range contextual information. Inspired by the success of transformers in natural language processing (NLP), researchers have begun exploring their applicability to computer vision tasks.

This project aims to leverage the capabilities of computer vision transformers to develop an efficient and accurate system for driver drowsiness detection. By analyzing facial expressions, eye movements, and other visual cues, the proposed system will identify signs of drowsiness in real-time video feeds from in-vehicle cameras. Through the integration of advanced deep learning techniques and transformer architectures, this project seeks to enhance road safety by providing timely alerts to drivers at risk of falling asleep behind the wheel.

1.2 Motivation

The motivation behind this project stems from the pressing need to address the significant risks associated with driver drowsiness on the roads. Drowsy driving poses a serious threat to public safety, contributing to countless accidents, injuries, and fatalities each year worldwide. Despite awareness campaigns and legislative measures aimed at combating this issue, the prevalence of drowsy driving remains alarmingly high, highlighting the inadequacy of existing solutions.

Traditional methods of detecting driver drowsiness, such as physiological monitoring and behavioral analysis, have limitations that hinder their widespread adoption and effectiveness. These methods often require intrusive sensors, specialized equipment, or manual intervention, making them impractical for real-world deployment in vehicles. Moreover, they may lack the ability to provide timely warnings to drivers, thereby increasing the likelihood of accidents.

In contrast, computer vision-based approaches offer a promising alternative for drowsiness detection, leveraging the power of artificial intelligence and image processing techniques to analyze visual cues indicative of driver fatigue. By analyzing facial expressions, eye movements, and other subtle indicators of drowsiness in real-time video streams, computer vision systems can provide non-intrusive, automated alerts to drivers, helping to prevent accidents before they occur.

The motivation for this project lies in harnessing the potential of computer vision transformers, a cutting-edge architecture that has shown remarkable performance in various visual recognition tasks. By adapting transformer models to the specific requirements of driver drowsiness detection, we aim to develop a robust and efficient system capable of accurately identifying signs of drowsiness in real-world driving scenarios. Through the integration of advanced deep learning techniques and transformer architectures, we seek to overcome the limitations of existing drowsiness detection methods and provide a more reliable solution for enhancing road safety.

Ultimately, the success of this project has the potential to save lives and reduce the societal and economic costs associated with drowsy driving-related accidents. By empowering drivers with intelligent drowsiness detection technology, we aim to create a safer and more secure driving environment for everyone on the road.

1.3 Objectives

The primary objective of this project is to develop a robust and efficient system for driver drowsiness detection using computer vision transformer architecture. Specifically, the project aims to achieve the following objectives:

- **Implement Transformer-Based Model:** Design and implement a computer vision transformer architecture tailored for the task of driver drowsiness detection. This includes adapting transformer-based models to analyze facial features, eye movements, and other visual cues indicative of drowsiness in real-time video streams.
- **Dataset Collection and Annotation:** Gather a comprehensive dataset of driver video footage capturing diverse scenarios and instances of drowsiness. Annotate the dataset with ground truth labels to facilitate model training and evaluation.
- **Model Training and Optimization:** Train the proposed transformer-based model on the collected dataset using state-of-the-art deep learning techniques. Explore strategies for optimizing model performance, including data augmentation, regularization, and hyperparameter tuning.
- **Real-time Inference:** Develop an efficient inference pipeline capable of processing streaming video feeds from in-vehicle cameras in real-time. Ensure that the drowsiness detection system operates with low latency and high accuracy to provide timely alerts to drivers.
- **Evaluation and Validation:** Evaluate the performance of the developed system using appropriate metrics, including accuracy, precision, recall, and F1-score. Validate the effectiveness of the system through rigorous testing on unseen data and real-world driving scenarios.
- **Integration and Deployment:** Integrate the trained model into a practical software application suitable for deployment in vehicles. Develop user-friendly interfaces and

seamless integration with existing automotive systems to ensure ease of use and compatibility.

- **Performance Comparison:** Compare the performance of the proposed transformer-based approach with existing drowsiness detection methods, including traditional machine learning algorithms and deep learning architectures. Highlight the advantages of the proposed system in terms of accuracy, efficiency, and real-world applicability.

By accomplishing these objectives, this project aims to advance the state-of-the-art in driver drowsiness detection technology and contribute to the development of safer and more intelligent transportation systems. The ultimate goal is to mitigate the risks associated with drowsy driving and enhance road safety for drivers, passengers, and pedestrians alike.

1.4 Project Features

A comprehensive driver drowsiness detection system incorporates several critical features to ensure its effectiveness and reliability. At its core, the system relies on real-time monitoring facilitated by high-quality cameras that capture the driver's facial expressions and eye movements. Advanced algorithms, such as Haar cascades or deep learning models, are used for face detection and the identification of facial landmarks, particularly around the eyes, nose, and mouth. This allows for detailed eye state analysis, including blink detection and measurement of eye closure duration, which are key indicators of drowsiness. Additionally, the system monitors head position, tracking nodding or tilting movements that may suggest fatigue. Behavioral analysis extends to recognizing yawns through mouth shape analysis and detecting changes in facial expressions indicative of tiredness.

To ensure prompt intervention, the system features a robust alert mechanism, which includes audio alarms, visual warnings, and haptic feedback, such as vibrations through the seat or steering wheel. These alerts are triggered when signs of drowsiness are detected, helping to keep the driver alert. Machine learning models play a crucial role in this process, leveraging extensive datasets of facial expressions and drowsiness states to improve detection accuracy. The integration of these models ensures that real-time data is analyzed effectively, minimizing false positives and negatives.

2. LITERATURE REVIEW

Driver drowsiness is a significant contributing factor to road accidents worldwide, prompting extensive research into the development of effective detection systems aimed at improving road safety. This literature review provides an overview of existing methods for driver drowsiness detection, with a focus on recent advancements in computer vision techniques, particularly the application of transformer-based architectures.

Recent advancements in driver drowsiness detection systems have been significantly driven by improvements in computing technology and artificial intelligence. Albadawi, Takruri, and Awad (2024) [1] provide a comprehensive review of developments over the past decade, categorizing systems based on the information used to detect drowsiness. They highlight the use of real-time driver data and various AI algorithms to enhance system performance, focusing on classification accuracy, sensitivity, and precision.Their review aligns with our project's goal of implementing an effective driver drowsiness detection system. By applying advanced machine learning models and real-time data processing, we aim to achieve high accuracy in detecting drowsiness, similar to the methods discussed in the paper. Albadawi et al. (2024) also address practical challenges and reliability issues, offering benchmarks for our system's performance evaluation.

Researchers Jabbar, Kharbeche and Alhajyaseen (2018) [2] at Qatar Univeristy published a paper discussing deep learning methods that can be implemented on Android Applications with high accuracy. The main contribution of this work is the compression of heavy baseline model to a lightweight model. The proposed model is able to achieve an accuracy of more than 80%.

Wijnands, Thompson, A. Nice and Aschwanden (2020) [3] in their paper show how depthwise separable 3D convolutions, combined with an early fusion of spatial and temporal information, can achieve a balance between high prediction accuracy and real-time inference requirements. Here the author uses computationally expensive techniques that achieve superior results on action recognition benchmarks but create bottleneck for real-time safety critical applications on mobile phones.

Safarov et al. (2024) [4] present a study showcasing the integration of deep learning and computer-vision algorithms for real-time driver drowsiness detection. By analyzing eye-blink patterns and facial landmarks, their approach achieves impressive accuracy rates, including 95.8% for drowsy-eye detection and 97% for open-eye detection. The study's method leverages custom data for model training and utilizes landmark detection to monitor eye and mouth regions. Real-time analysis of these features enables precise classification of drowsy states. These findings inform our project's objective of developing a reliable drowsiness detection system, emphasizing the effectiveness of similar deep learning techniques in enhancing road safety.

Zhao et al. (2023) [5] discusses a new type of Computer Vision Transformer known as RT-DETR. RT-DETR designed on efficient hybrid encoders to expeditiously process multi-scale features by decoupling intra-scale interaction and cross-scale fusion to improve speed. RT-DETR provides flexible speed tuning by adjusting the number of decoder layers to adapt to various scenarios without retraining. The given RT-DETR has shown to outperform YOLO models on both accuracy and speed.

Ref	Model Used	Dataset	Results
(Harkous & Artail, 2019) [2]	Recurrent Neural Network (RNN)	CAN Driving Data	Accuracy: 78%
(Peppes et al., 2021) [3]	Logistic Regression SVM and Random Forest	Private Dataset	Accuracy: 89.8%
(Jabbar et al, 2018) [4]	Multilayer Perceptron Architecture	National Tsing Hua University (NTHU) Driver Drowsiness Detection	Accuracy: 81%
(Dwivedi et al, 2014) [5]	Adapted Shallow CNNs	YAWDD Dataset	Accuracy: 74%
(Wijnands et al, 2019) [6]	3D Neural Networks	2016 Asian Conference on Computer Vision DDD dataset	Accuracy: 73.9%
Safarov et al, 2023) [7]	Deep Learning Algorithms	Private Collected Dataset	Accuracy: 85%

In this literature review, we listed various experiments on driver drowsiness detection, including their methods and accuracy results. Now, we plan to create a new dataset for detecting driver drowsiness using advanced computer vision transformers. These transformers can better analyze visual data to improve detection accuracy. Our system will be designed to run efficiently on mobile devices, providing real-time drowsiness detection to help prevent accidents on the road.

3. PROPOSED METHODOLOGY

The proposed methodology for the project "Driver Drowsiness Detection using Computer Vision Transformer" involves the following steps:

3.1 Data Collection and Preprocessing

- **Face Detection:** Use a pre-trained face detection model (e.g., YOLOv5) to detect the frontal face of the driver from the video feed.
- **Face Alignment:** Use a face alignment algorithm (e.g., OpenCV's Face Alignment) to align the detected face to a standard pose, ensuring that the facial features are properly aligned for analysis.
- **Data Preprocessing:** Preprocess the aligned face images by resizing them to a standard size, normalizing pixel values, and converting them to grayscale.
- **Normalization**: Data is pre-processed by normalizing pixel values, ensuring uniformity in the input space across datasets. This step often involves scaling pixel values to a specific range (e.g., [0, 1] or [1, 1]).
- **Dimension Standardization**: Standardizing input dimensions across images or data points is crucial for consistency during model training. Resizing images to a uniform dimension is a common practice.
- **Augmentation Strategies:** Techniques like random rotations, translations, flips, zooms, and crops are employed to increase the diversity of the training data. Augmentation mitigates overfitting and enhances the model's robustness by exposing it to a wider range of variations.
- **Contrast and Brightness Adjustments**: Modifying contrast, brightness, and saturation levels in images contributes to the model's ability to generalize better to varying real-world conditions.
- **Regularization**: Techniques such as dropout, adding noise to hidden layers, or introducing regularization terms in the loss function are used to prevent overfitting and enhance model generalization.
- **Synthetic Data Generation:** In instances where the dataset is imbalanced, synthetic data generation techniques like SMOTE (Synthetic Minority Oversampling Technique) or GAN based data augmentation may be employed to balance class distributions.
- **Class Balancing**: Weighted loss functions or resampling techniques are used to address class imbalances, ensuring that the model is not biased towards majority classes.
- **Data Partitioning:** The dataset is typically split into training, validation, and test sets. The training set is used for model training, the validation set for hyperparameter tuning, and the test set for final evaluation.

3.2 Model Selection

- **Model Selection:** We selected transformer-based architecture called Real-Time Detection Transformer (RT-DeTr) for the task of driver drowsiness detection.
- **Model Customisation:** Customize the chosen architecture to accommodate the requirements of drowsiness detection, including input resolution, number of layers, and attention mechanisms.

3.3 Model Development

- **Transformer Architecture:** Design a transformer-based architecture (e.g., Vision Transformer) to analyze the preprocessed face images and extract relevant features indicative of drowsiness.
- **Training:** Train the transformer model using a large dataset of labeled face images, where each image is annotated as either "drowsy" or "alert".
- **Model Evaluation:** Evaluate the performance of the trained model using metrics such as accuracy, precision, recall, and F1-score on a separate test dataset.

3.4 System Integration

- **Real-time Processing:** Implement the trained model in a real-time processing framework (e.g., OpenCV) to analyze the video feed and detect drowsiness in real-time.
- **Alert Generation:** Generate an alert to the driver when the system detects drowsiness, which can be in the form of a visual warning on the dashboard or an auditory alert through the vehicle's speakers.

3.5 System Testing

- **System Testing:** Test the integrated system on a variety of scenarios, including different lighting conditions, camera angles, and driver characteristics, to ensure its robustness and reliability.
- **User Feedback:** Collect user feedback on the system's performance and accuracy to identify areas for improvement and refine the system accordingly.

3.6 System Deployment

- **Vehicle Integration:** Integrate the system into a vehicle's infotainment system or dashboard, ensuring seamless integration with existing features and user interfaces.
- **Ongoing Maintenance:** Continuously monitor and update the system to ensure its performance and accuracy remain high, and to address any issues that may arise during deployment.

4. SOFTWARE REQUIREMENT

4.1 Feasibility Study

The feasibility study conducted for the "Driver Drowsiness Detection using Computer Vision Transformer" project has provided a comprehensive assessment of the project's viability from technical, financial, and operational standpoints.

4.1.1 Technical Feasibility

Technically, the project's strong foundation in leveraging cutting-edge computer vision technologies like YOLOv5 and Vision Transformer, showcasing the system's potential to deliver robust and accurate drowsiness detection capabilities. The availability of advanced algorithms and frameworks makes it more feasible in implementing a real-time system capable of analysing complex driver behaviour features. Computer Vision Transformers have been used across various fields of science ranging from medicine[6] to sports[7]. In medicine they are being used for image-based disease classification, anatomical structure segmentation, registration, region-based lesion detection, captioning, report generation, and reconstruction using multiple medical imaging modalities that greatly assist in medical diagnosis and hence treatment process. While in sports, Computer Vision Transformers are being deployed to obtain player identities and for game event recognition. ViT have been introduced for mobile, one such example is Mobile ViT[8] which allows us to execute Computer Vison Transformers on mobile devices.

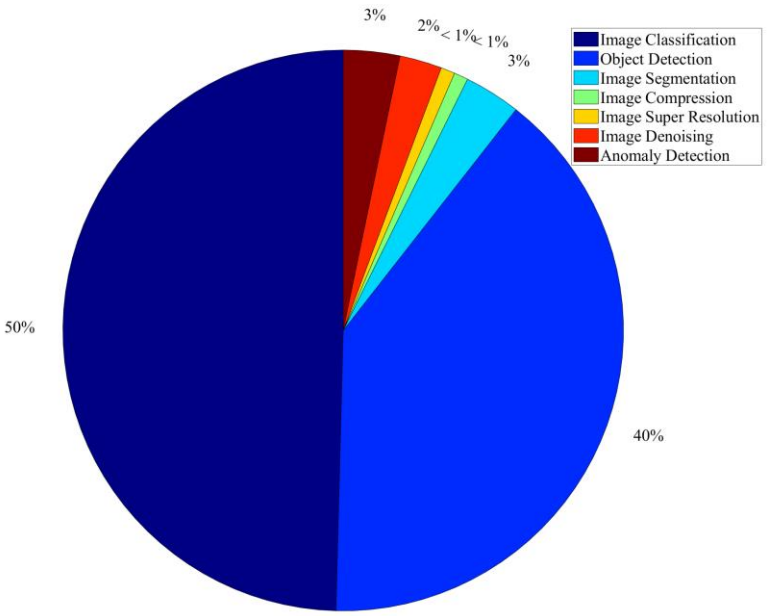


Fig 2. Use cases of Computer Vison Transformers

4.1.2 Financial Feasibility

From a financial perspective, the estimated costs of development and deployment fall within budgetary constraints. What we need for deploying the given system, is a smartphone (we used Xiaomi 15 for this project) and a car mobile holder. We can also integrate the given Driver Drowsiness Detection System in an IOT system, making it embedded in the car dashboard, making it even easier for the driver to use it. The estimated cost of our system comes around to (Rupees. 15000) when considering phone, and for without considering phone it comes out to (Rupees. 500). As almost everyone who owns a car would also almost always own a phone, hence it is expected that the given system can be deployed by almost everyone. When considering the potential benefits of the system, including the reduction of accidents and enhancement of road safety, are projected to outweigh the associated costs, making the project financially sustainable and rewarding.

Fig 3. (a) Depicts Mi 15 phone that we used for execution (on the left) (b) depicts the ordinary car phone holder

4.1.3 Operational Feasibility

Moreover, the operational feasibility of the project is evident in its ability to integrate into any android application easily, and providing accurate and decent performance, ensuring practicality and ease of implementation.

In conclusion, the feasibility study affirms that the "Driver Drowsiness Detection using Computer Vision Transformer" project is not only technically sound and financially feasible but also operationally viable, positioning it as a promising solution for effectively detecting driver drowsiness and enhancing road safety. The study's positive evaluation underscores the project's potential to make a significant impact in mitigating accidents caused by driver fatigue and underscores its importance in advancing intelligent transportation systems.

4.2 Requirement Analysis Steps

We define the requirements that we need for our project in terms of a context diagram. This section outlines the requirements necessary for the successful implementation of the system.

4.2.1 Context Diagram

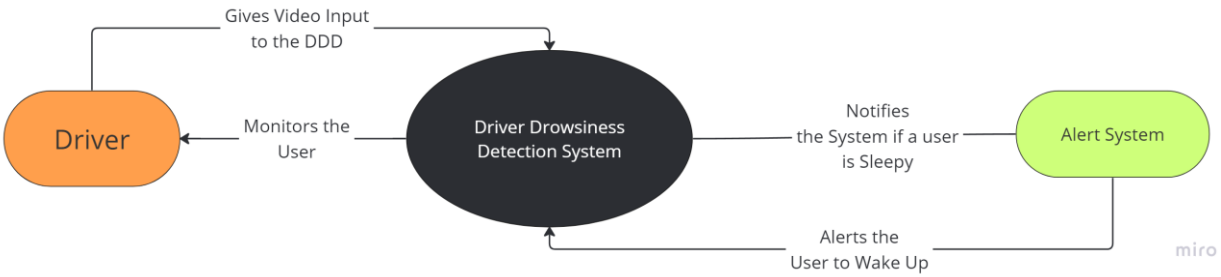


Fig 3. Context Diagram for our Driver Drowsiness Detection System

4.2.2 Model the Requirements

Functional Requirements

- **Real-Time Monitoring:** Continuously monitor the driver's facial expressions and eye movements using the phone's camera.
- **Alert Mechanism:** Provide audio, visual, and haptic alerts when drowsiness is detected. Here we will be using phone's alarm as our Alert Mechanism.
- **Data Processing:** Process video frames in real-time to ensure immediate detection and alerting.

Non-Functional Requirements

- **Scalability:** Ensure compatibility with different smartphone models and operating systems.
- **Usability:** Design the system to be easy to set up and use, with minimal driver distraction.
- **Battery Efficiency:** Optimize the system to minimize battery consumption.

Data Requirements

- **Input Data:** Video frames from the smartphone camera.
- **Output Data:** Alerts (audio, visual, haptic), drowsiness status, and logs.

4.3 System Requirements-

Hardware Requirements-

- Smartphone with a high resolution camera
- A phone holder for car dashboard

Software Requirements –

- Development environment: Android Studio or Xcode.
- Libraries: TensorFlow Lite, OpenCV.
- Platforms: Android 8.0+ and iOS 11.0+.

4.4 Tools and Image Processing Libraries-

4.4.1 Tensorflow-

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. It was developed by the Google Brain team for Google's internal use in research and production. The initial version was released under the Apache License 2.0 in 2015. Google released an updated version, TensorFlow 2.0, in September 2019. TensorFlow can be used in a wide variety of programming languages, including Python, JavaScript, C++, and Java, facilitating its use in a range of applications in many sectors. TensorFlow provides a number of computer vision (CV) and image classification tools, like KerasCV.

4.4.2 OpenCV-

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly for real-time computer vision.Originally developed by Intel, it was later supported by Willow Garage, then Itseez (which was later acquired by Intel). The library is cross-platform and licensed as free and open-source software under Apache License 2. Starting in 2011, OpenCV features GPU acceleration for real-time operations. OpenCV is the world's biggest computer vision library.

It's open source, contains over 2500 algorithms and is operated by the non-profit Open Source Vision Foundation.

4.4.3 Ultralytics-

Ultralytics provide us with start of the art Computer Vision Models which we can finetune to enable fine-grain control over your project. It is the biggest repository of Computer Vision models, and provide us with an easy way to access and use these Computer Vision models.

4.4.4 Pygame-

Pygame 1.9 can be used for some simple computer vision tasks, such as capturing images and watching live streams. Pygame is a Python library for creating 2D games, interactive simulations, and multimedia programs. It includes computer graphics and sound libraries, and can be used on almost every platform and operating system. Pygame's camera module supports cameras that use v4l2 on Linux, and other platforms via Videocapture or OpenCV. The tutorial for Pygame 1.9 includes code samples for these use cases.

5. SYSTEM DESIGN

5.1 Data Flow Diagram-

A Data Flow Diagram (DFD) is a graphical representation of the flow of data through a system. It shows how data is processed by the system in terms of inputs and outputs. In the context of the Driver Drowsiness Detection System, the DFD illustrates how the system captures, processes, and responds to driver facial data to detect drowsiness and issue alerts.

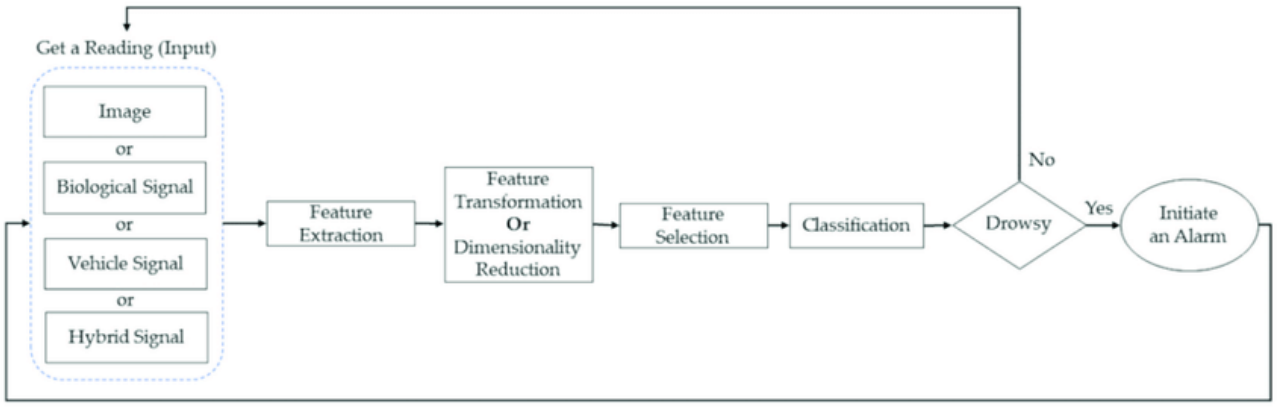


Fig 3. (a) Dataflow Diagram of Driver Drowsiness Detection System

5.2 Use Case Diagram-

A Use Case Diagram is a visual representation of the interactions between users (actors) and the system, depicting the different ways the system will be used. In the context of the Driver Drowsiness Detection System, the use case diagram illustrates the various functionalities of the system and how drivers and other stakeholders interact with it.

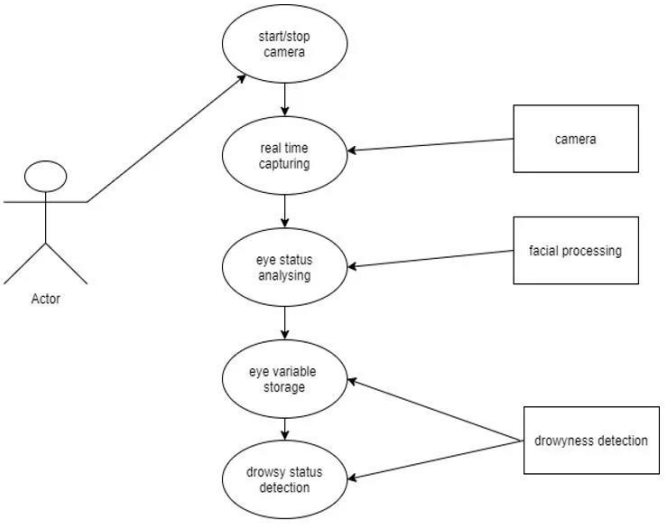


Fig 4. Use Case Diagram of our Driver Drowsiness Detection System

The following are the steps involved in use case-

- Start Monitoring
- Adjust Settings
- Receive Alert
- View History
- Stop Monitoring

6. TECHNOLOGY USED

6.1 Computer Vision Transformers-

6.1.1 Overview

A vision transformer (ViT) is a transformer designed for computer vision. A ViT breaks down an input image into a series of patches (rather than breaking up text into tokens), serialises each patch into a vector, and maps it to a smaller dimension with a single matrix multiplication. These vector embeddings are then processed by a transformer encoder as if they were token embeddings. ViT has found applications in image recognition, image segmentation, and autonomous driving.

Transformers were introduced in 2017, in a paper "Attention Is All You Need",[9] and have found widespread use in natural language processing. In 2020, they were adapted for computer vision, yielding ViT. In 2021 a pure transformer model demonstrated better performance and greater efficiency than CNNs on image classification. A study in June 2021 added a transformer backend to ResNet, which dramatically reduced costs and increased accuracy.

In the same year, some important variants of the Vision Transformers were proposed. These variants are mainly intended to be more efficient, more accurate or better suited to a specific domain. Among the most relevant is the Swin Transformer, which through some modifications to the attention mechanism and a multi-stage approach achieved state-of-the-art results on some object detection datasets such as COCO. Another interesting variant is the TimeSformer, designed for video understanding tasks and able to capture spatial and temporal information through the use of divided space-time attention.

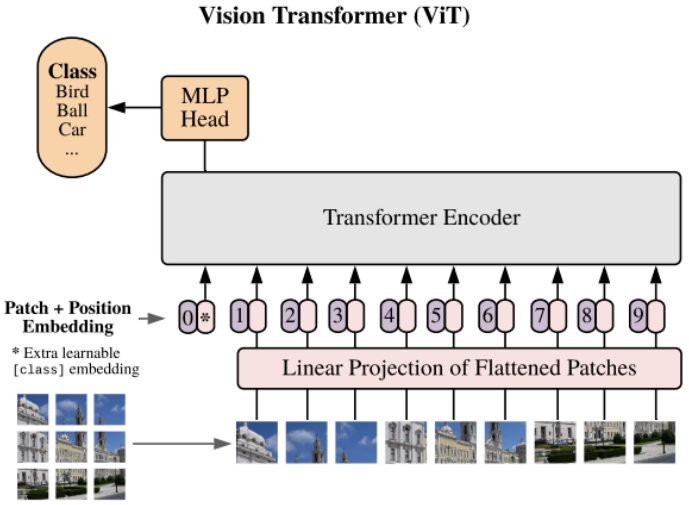


Fig 5. Computer Vision Transformer Architecture

The concept of self-attention has been adapted for processing images with the use of Vision Transformers. Unlike text data, images are inherently two-dimensional, comprising pixels arranged in rows and columns. To address this challenge, ViTs convert images into sequences that can be processed by the Transformer.

- Split an image into patches: The first step in processing an image with a Vision Transformer is to divide it into smaller, fixed-size patches. Each patch represents a local region of the image.
- Flatten the patches: Within each patch, the pixel values are flattened into a single vector. This flattening process allows the model to treat image patches as sequential data.
- Produce lower-dimensional linear embeddings: These flattened patch vectors are then projected into a lower-dimensional space using trainable linear transformations. This step reduces the dimensionality of the data while preserving important features.
- Add positional encodings: To retain information about the spatial arrangement of the patches, positional encodings are added. These encodings help the model understand the relative positions of different patches in the image.
- Feed the sequence into a Transformer encoder: The input to a standard Transformer encoder comprises the sequence of patch embeddings and positional embeddings. This encoder is composed of multiple layers, each containing two critical components: multi-head self-attention mechanisms (MSPs), responsible for calculating attention weights to prioritize input sequence elements during predictions, and multi-layer perceptron (MLP) blocks. Before each block, layer normalization (LN) is applied to appropriately scale and center the data within the layer, ensuring stability and efficiency during training. During the training, an optimizer is also used to adjust the model's hyperparameters in response to the loss computed during each training iteration.
- Classification Token: To enable image classification, a special "classification token" is prepended to the sequence of patch embeddings. This token's state at the output of the Transformer encoder serves as the representation of the entire image.

6.1.2 Architecture

Recall that the standard Transformer model received a one-dimensional sequence of word embeddings as input, since it was originally meant for NLP. In contrast, when applied to the task of image classification in computer vision, the input data to the Transformer model is provided in the form of two-dimensional images.

For the purpose of structuring the input image data in a manner that resembles how the input is structured in the NLP domain (in the sense of having a sequence of individual

words), the input image, of height H, width W, and C number of channels, is *cut up* into smaller two-dimensional patches. This results into $N=HWP2$ number of patches, where each patch has a resolution of (P,P) pixels.

Before feeding the data into the Transformer, the following operations are applied:

- Each image patch is flattened into a vector, xpn, of length $P2\times C$, where n=1,…N.
- A sequence of embedded image patches is generated by mapping the flattened patches to D dimensions, with a trainable linear projection, E.
- A learnable class embedding, $x$class, is prepended to the sequence of embedded image patches. The value of $x$class represents the classification output, y.
- The patch embeddings are finally augmented with one-dimensional positional embeddings, Epos, hence introducing positional information into the input, which is also learned during training.

The sequence of embedding vectors that results from the aforementioned operations is the following:

$z0=[x\text{class};xp1E;\ldots;xpNE]+E\text{pos}$

Dosovitskiy et al. make use of the encoder part of the Transformer architecture of Vaswani et al.

In order to perform classification, they feed $z0$ at the input of the Transformer encoder, which consists of a stack of L identical layers. Then, they proceed to take the value of $x$class at the Lth layer of the encoder output, and feed it into a classification head.

The classification head is implemented by a MLP with one hidden layer at pre-training time and by a single linear layer at fine-tuning time.

– *An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale, 2021.*

The multilayer perceptron (MLP) that forms the classification head implements Gaussian Error Linear Unit (GELU) non-linearity.

In summary, therefore, the ViT employs the encoder part of the original Transformer architecture. The input to the encoder is a sequence of embedded image patches (including a learnable class embedding prepended to the sequence), which is also augmented with positional information. A classification head attached to the output of the encoder receives the value of the learnable class embedding.

6.1.3 Real World Applications

Image Classification

A primary application of Vision Transformers is image classification, where ViTs serve as powerful classifiers. They excel in categorizing images into predefined classes by learning intricate patterns and relationships within the image, driven by their self-attention mechanisms.

Object Detection

Object detection is another domain where Vision Transformers are making a significant impact. Detecting objects within an image involves not only classifying them but also precisely localizing their positions. ViTs, with their ability to preserve spatial information, are well-suited for this task. These algorithms can identify objects and provide their coordinates, contributing to advancements in areas like autonomous driving and surveillance.

Image Segmentation

Image segmentation, which involves dividing an image into meaningful segments or regions, benefits greatly from the capabilities of ViTs. These models can discern fine-grained details within an image and accurately delineate object boundaries. This is particularly valuable in medical imaging, where precise segmentation can aid in diagnosing diseases and conditions.

Action Recognition

Vision Transformers are also making strides in action recognition, where the goal is to understand and classify human actions in videos. Their ability to capture temporal dependencies, coupled with their strong image processing capabilities, positions ViTs as contenders in this field. They can recognize complex actions in video sequences, impacting areas such as video surveillance and human-computer interaction.

6.2 RT-DeTR(Real Time Detection Transformer) –

Real-Time Detection Transformer (RT-DETR), developed by Baidu, is a cutting-edge end-to-end object detector that provides real-time performance while maintaining high accuracy. It leverages the power of Vision Transformers (ViT) to efficiently process multiscale features by decoupling intra-scale interaction and cross-scale fusion. RT-DETR is highly adaptable, supporting flexible adjustment of inference speed using different decoder layers without retraining. The model excels on accelerated backends like CUDA with TensorRT, outperforming many other real-time object detectors.

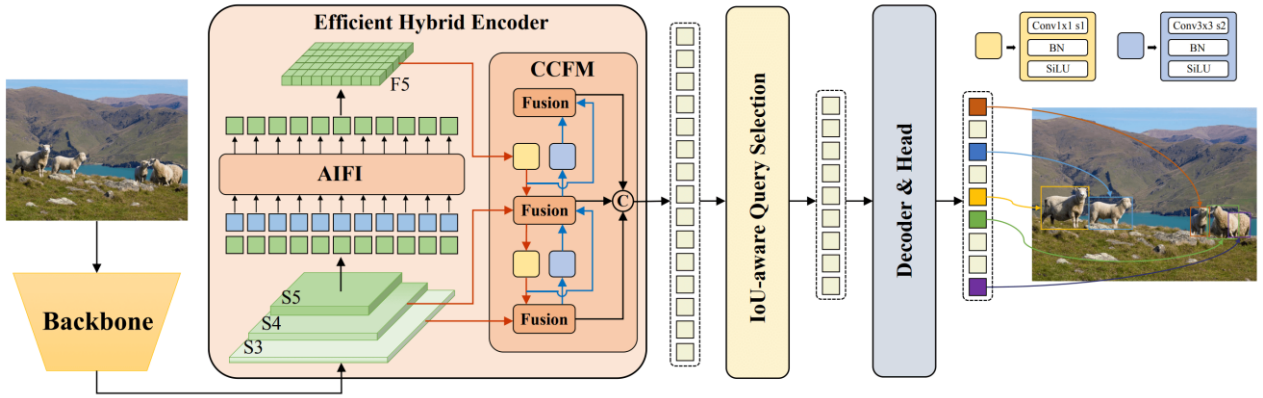


Fig 6. RT- DETR Architecture

Overview of Baidu's RT-DETR: The RT-DETR model architecture diagram shows the last three stages of the backbone {S3, S4, S5} as the input to the encoder. The efficient hybrid encoder transforms multiscale features into a sequence of image features through intrascale feature interaction (AIFI) and cross-scale feature-fusion module (CCFM). The IoU-aware query selection is employed to select a fixed number of image features to serve as initial object queries for the decoder. Finally, the decoder with auxiliary prediction heads iteratively optimizes object queries to generate boxes and confidence scores (source).

Key Features

- Efficient Hybrid Encoder: Baidu's RT-DETR uses an efficient hybrid encoder that processes multiscale features by decoupling intra-scale interaction and cross-scale fusion. This unique Vision Transformers-based design reduces computational costs and allows for real-time object detection.
- IoU-aware Query Selection: Baidu's RT-DETR improves object query initialization by utilizing IoU-aware query selection. This allows the model to focus on the most relevant objects in the scene, enhancing the detection accuracy.
- Adaptable Inference Speed: Baidu's RT-DETR supports flexible adjustments of inference speed by using different decoder layers without the need for retraining. This adaptability facilitates practical application in various real-time object detection scenarios.

6.3 ONNX(Open Neural Network Exchange) Format –

The Open Neural Network Exchange (ONNX) is an open-source artificial intelligence ecosystem of technology companies and research organizations that establish open standards for representing machine learning algorithms and software tools to promote innovation and collaboration in the AI sector. ONNX is available on GitHub.

The initiative targets:

- Framework interoperability

Allow developers to more easily move between frameworks, some of which may be more desirable for specific phases of the development process, such as fast training, network architecture flexibility or inferencing on mobile devices.

- Shared optimization

Allow hardware vendors and others to improve the performance of artificial neural networks of multiple frameworks at once by targeting the ONNX representation.

You can use ONNX to make a Tensorflow model 200% faster, which eliminates the need to use a GPU instead of a CPU. Using a CPU instead of a GPU has several other benefits as well:

- CPU have a broader availability and are cheaper to use
- CPUs can support larger memory capacities than even the best GPUs, like 2D image detection for example.

Fig 7. ONNX format logo

6.4 Drowsiness Detection–

Driver drowsiness detection is a car safety technology which helps prevent accidents caused by the driver getting drowsy. Various studies have suggested that around 20% of all road accidents are fatigue-related, up to 50% on certain roads. Some of the current systems learn driver patterns and can detect when a driver is becoming drowsy.

Various technologies can be used to try to detect driver drowsiness.

- **Steering pattern monitoring :**
  Primarily uses steering input from electric power steering system. Monitoring a driver this way only works as long as a driver actually steers a vehicle actively instead of using an automatic lane-keeping system.

- **Vehicle position in lane monitoring :**

Uses a lane monitoring camera. Monitoring a driver this way only works as long as a driver actually steers a vehicle actively instead of using an automatic lane-keeping system.

- **Driver eye/face monitoring :**
  Uses computer vision to observe the driver's face, either using a built-in camera or on mobile devices.

- **Physiological measurement :**
  Requires body sensors to measure parameters like brain activity, heart rate, skin conductance, muscle activity, head movements etc...

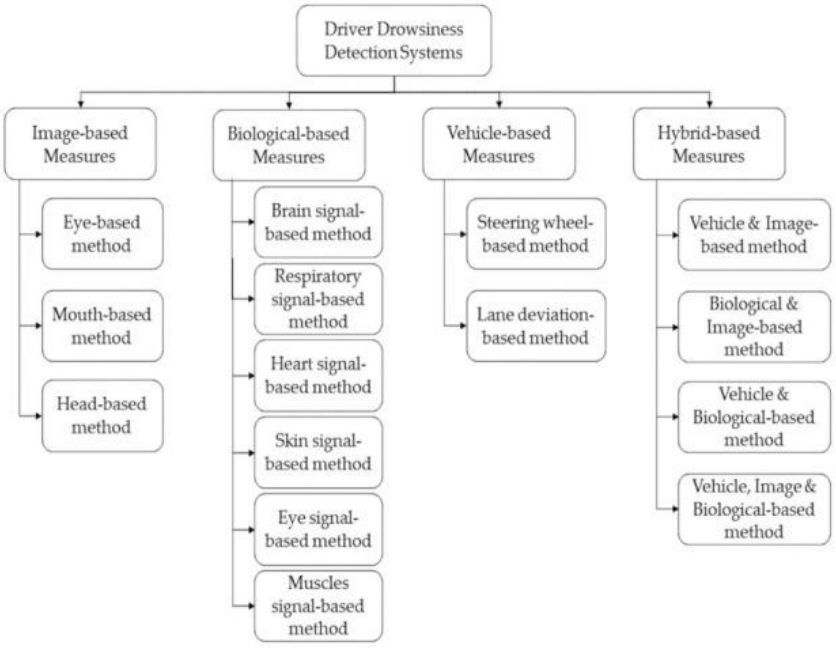


Fig 8. Driver Drowsiness Detection Measure

6.5 Haar Cascade–

Haar cascade is an algorithm that can detect objects in images, irrespective of their scale in image and location. This algorithm is not so complex and can run in real-time. We can train a haar-cascade detector to detect various objects like cars, bikes, buildings, fruits, etc. Haar cascade uses the cascading window, and it tries to compute features in every window and classify whether it could be an object.

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning

based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.

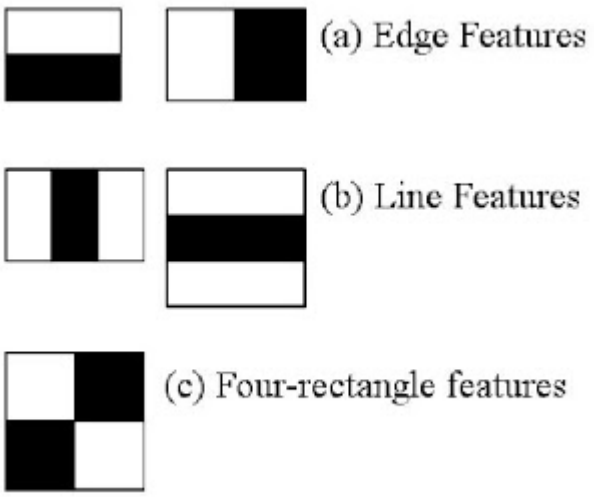


Fig 9. Haar Cascade Feature box

Now, all possible sizes and locations of each kernel are used to calculate lots of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, we need to find the sum of the pixels under white and black rectangles. To solve this, they introduced the integral image. However large your image, it reduces the calculations for a given pixel to an operation involving just four pixels. Nice, isn't it? It makes things super-fast.

But among all these features we calculated, most of them are irrelevant. For example, consider the image below. The top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applied to cheeks or any other place is irrelevant. So how do we select the best features out of 160000+ features? It is achieved by Adaboost.

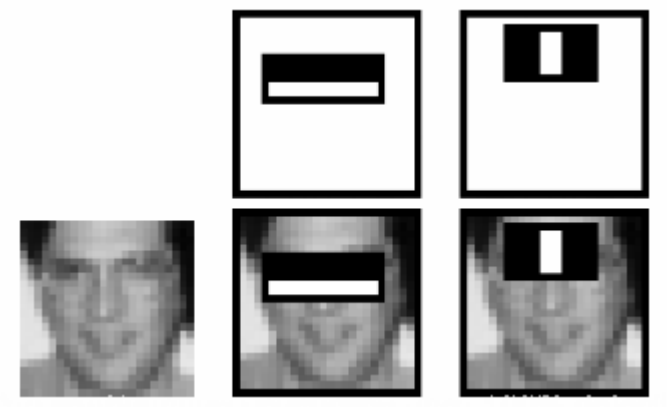


Fig 10. Selecting Features in the given image

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. Obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that most accurately classify the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then the same process is done. New error rates are calculated. Also new weights. The process is continued until the required accuracy or error rate is achieved or the required number of features are found).

The final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with 95% accuracy. Their final setup had around 6000 features.

7. PROJECT SNAPSHOTS AND CODING

7.1 Coding

The coding section details the implementation of the Driver Drowsiness Detection System, including the development environment, tools used, the architecture of the code, key modules, and code snippets that illustrate important functionalities.

```
import cv2
from ultralytics import RTDETR
import time
import pygame

def play_music(file_path):
    pygame.init()
    pygame.mixer.init()
    pygame.mixer.music.load(file_path)
    pygame.mixer.music.play()

video_path = 0
cap = cv2.VideoCapture(video_path)

window_width = 640
window_height = 480
cv2.namedWindow(" model Test", cv2.WINDOW_NORMAL)
cv2.resizeWindow(" model Test", window_width, window_height)
model = RTDETR("best_imp.pt")

sleep_counter = 0
while cap.isOpened():
    # read a frame from the video
    []
    success, frame = cap.read()
    prev_class = 0.0
    try:
        if success:
            results = model.predict(frame, conf=0.5)
            annotated_frame = results[0].plot()
            # display the frame
            cv2.imshow(" model Test", annotated_frame)
            class_label = results[0][0].boxes.cls
            conf_label = results[0][0].boxes.conf
            print("class label :", class_label)
            print("conf_label  :", conf_label)
            if class_label == 1.0 and conf_label > 0.5:
                sleep_counter += 1
                if(sleep_counter >= 10):
                    file_path = "loud_alarm.mp3"
```

```
                    play_music(file_path)
                    time.sleep(4)
            else:
                sleep_counter = 0
            prev_class = class_label

            # Break the loop if q is pressed
            if cv2.waitKey(1) & 0xFF == ord("q"):
                print("User pressed 'q'. Exiting loop.")
                break
        # Break the loop if the end of the video is reached
        else:
            print("End of video reached. Exiting loop.")
            break
    except:
        print("Error in processing frame.")
    # Release the video capture object and close the display window

cap.release()
cv2.destroyAllWindows()
```

The coding phase of the Driver Drowsiness Detection System focused on implementing a robust, real-time monitoring system using computer vision and deep learning techniques. Key functionalities such as face detection, feature extraction, drowsiness analysis, and alert generation were successfully developed and tested. The system is now ready for integration and further testing in real-world conditions.

7.2 Snapshots

The Snapshot section provides visual evidence of the Driver Drowsiness Detection System in operation. These images demonstrate the system's functionality and effectiveness in real-time scenarios, showcasing its user interface, monitoring process, and alert mechanisms.

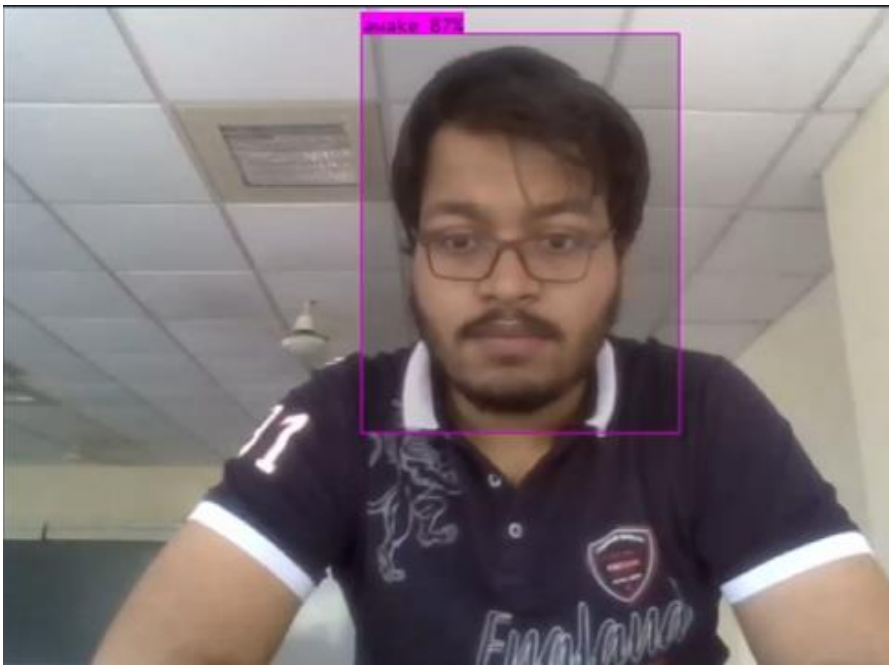


Fig 11. Driver Drowsiness Detection System execution. It classifies Aadil's face as awake.

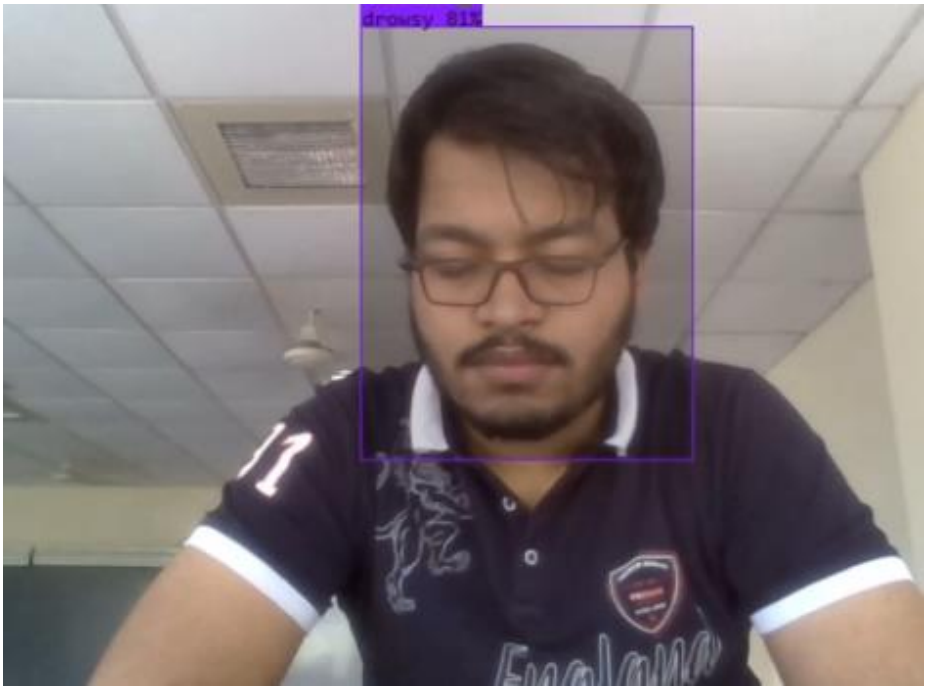


Fig 11. Driver Drowsiness Detection System execution. It classifies Aadil's face as drowsy.

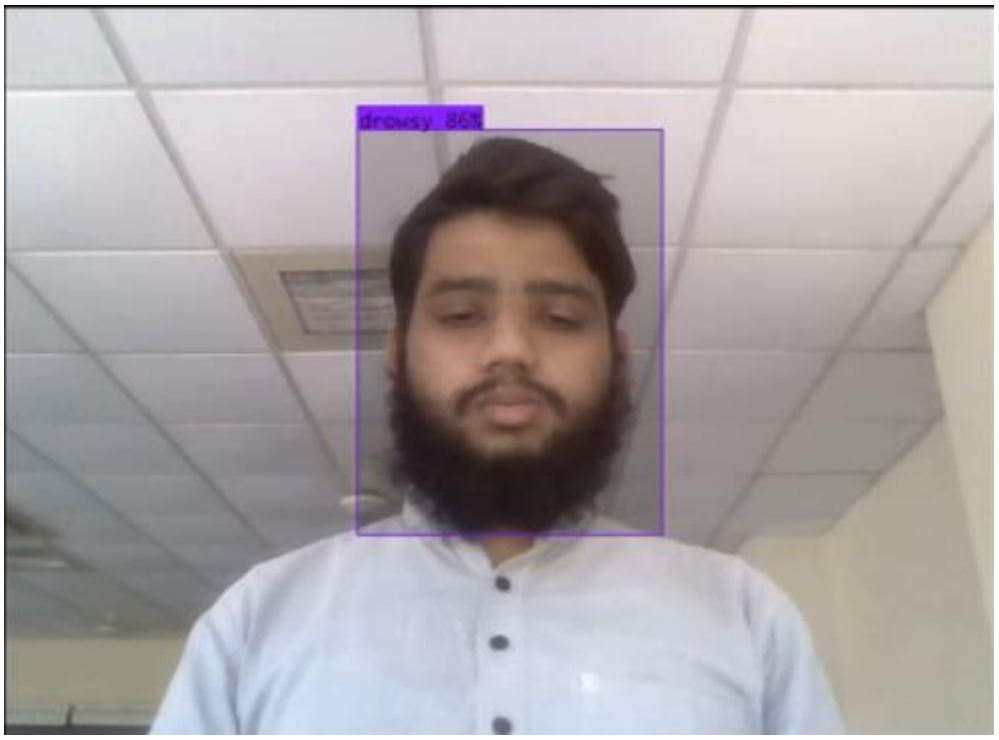


Fig 11. Driver Drowsiness Detection System execution. It classifies Ebad's face as drowsy.

Fig 11. Driver Drowsiness Detection System execution. It classifies Ebad's face as awake.

Execution of Driver Drowsiness Detection System on a Mobile Device-

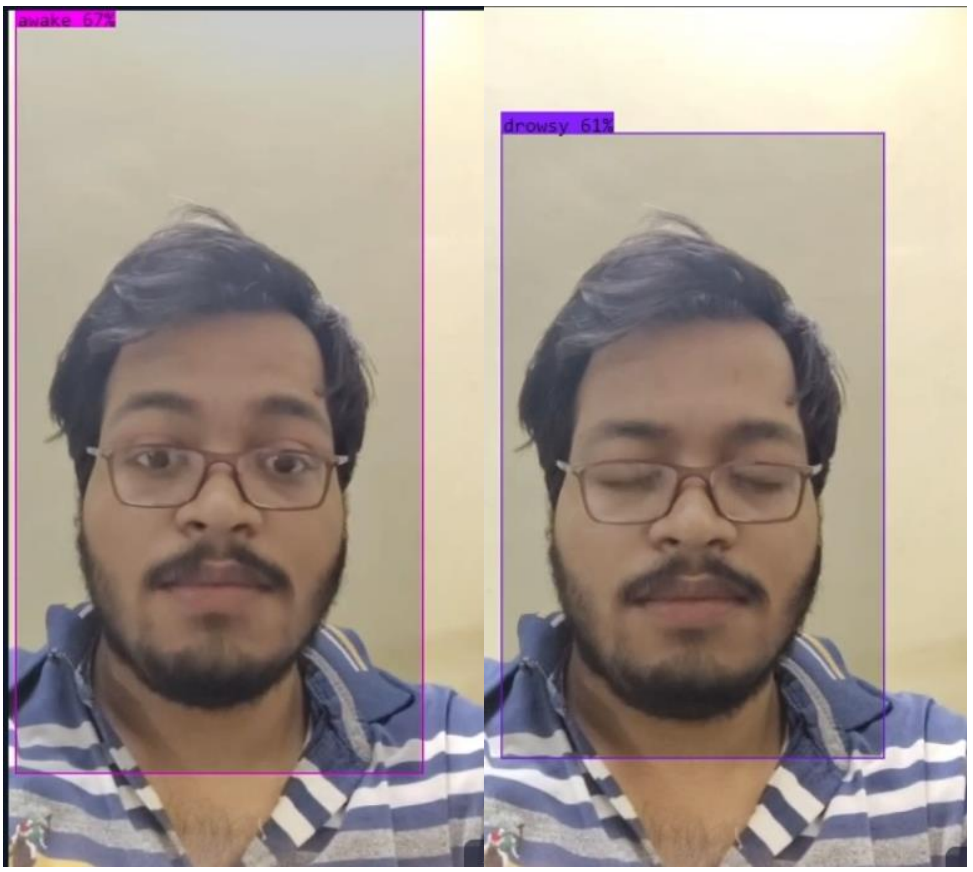


Fig 12. Implementation of Driver Drowsiness Detection System on Mobile Device

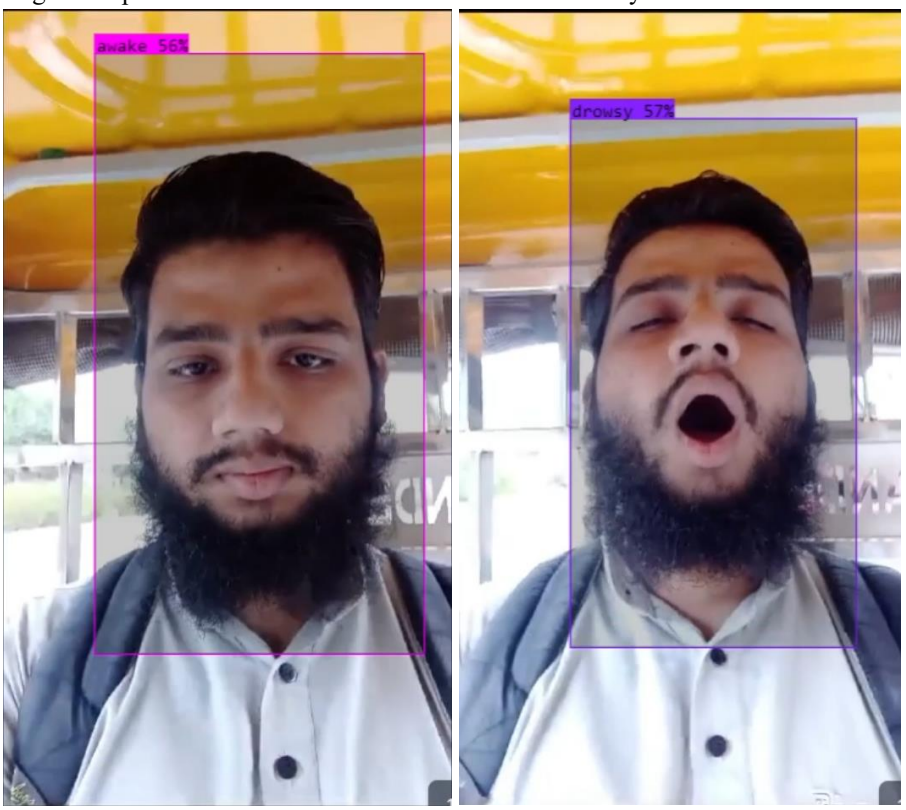


Fig 13. Implementation of Driver Drowsiness Detection System on Mobile Device

8. TESTING AND RESULTS

8.1 Dataset Used-

The success of the Driver Drowsiness Detection System relies heavily on the quality and diversity of the dataset used for training and evaluation. This section outlines the datasets employed, their sources, and the specific features they contain, which are crucial for developing an accurate and reliable drowsiness detection model.

- **Driver Drowsiness Dataset (DDD) by Ismail Nasri –**
  - The obtained dataset (DDD) has been used for training and testing CNN architecture for driver drowsiness detection in the "Detection and Prediction of Driver Drowsiness for the Prevention of Road Accidents Using Deep Neural Networks Techniques" paper[].
  - RGB images
  - 2 classes (Drowsy & Non Drowsy)
  - Size of image : 640 × 640
  - More than 1,790 images in total
  - File size : 232 MB
- **Driver Drowsiness Detection Dataset by Itomic01 –**
  - The dataset was obtained from Roboflow
  - RGB images
  - 2 classes (Drowsy & Non Drowsy)
  - Size of image : 640 × 640
  - More than 1,500images in total
  - File size : 133 MB

The whole dataset that we collected looks like below-

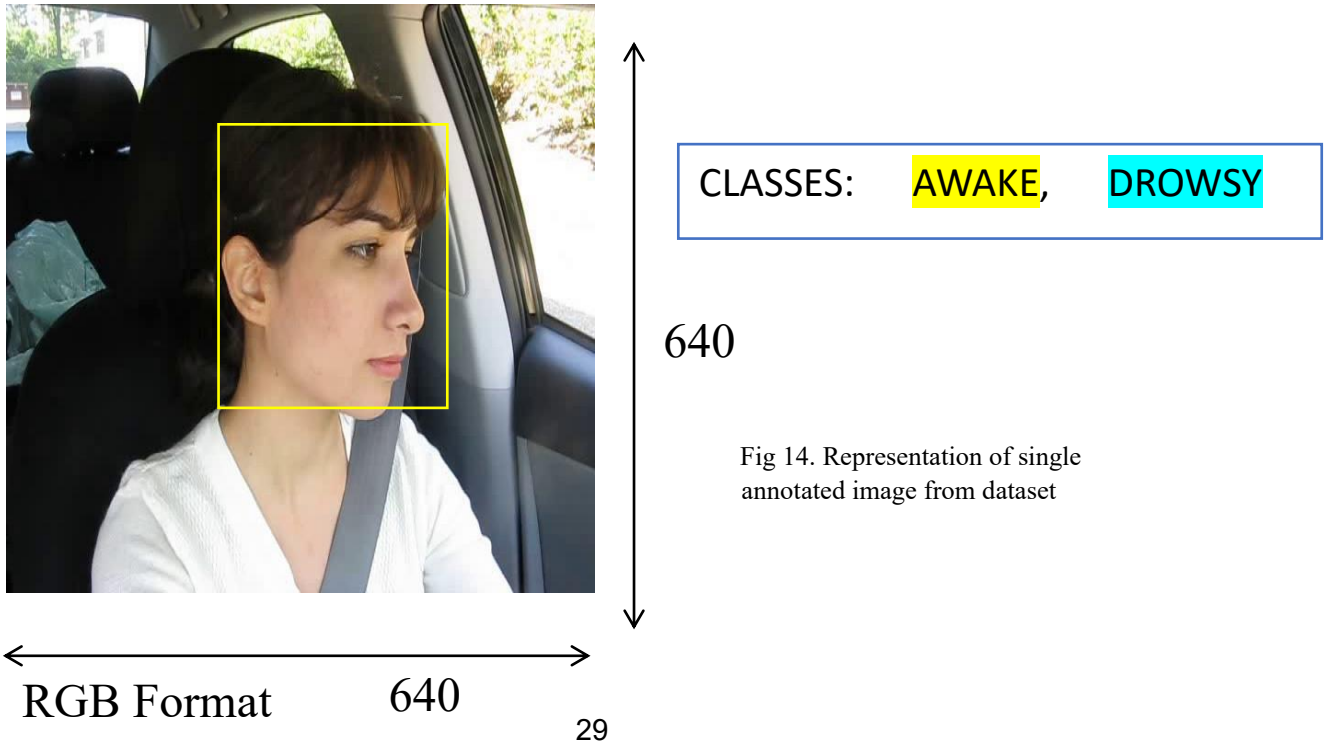


Fig 14. Representation of single annotated image from dataset

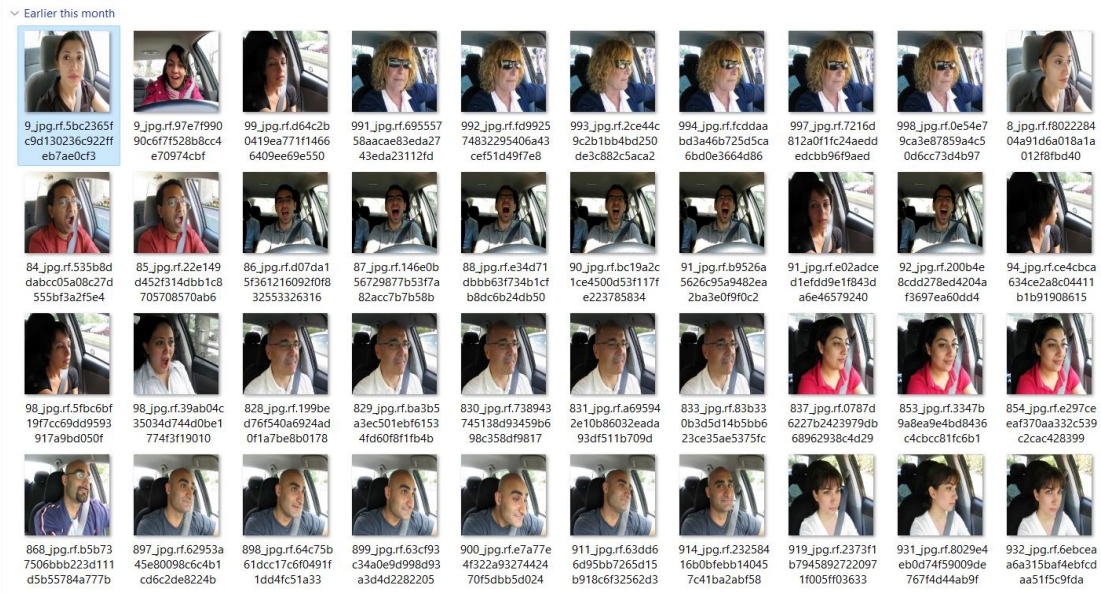


Fig 15. Screenshot of our dataset (In total 3,290 images)

We applied augmentation techniques like flip augmentation, rotation augmentation and cut-out augmentation to our image which increased the number of images to 10,000.

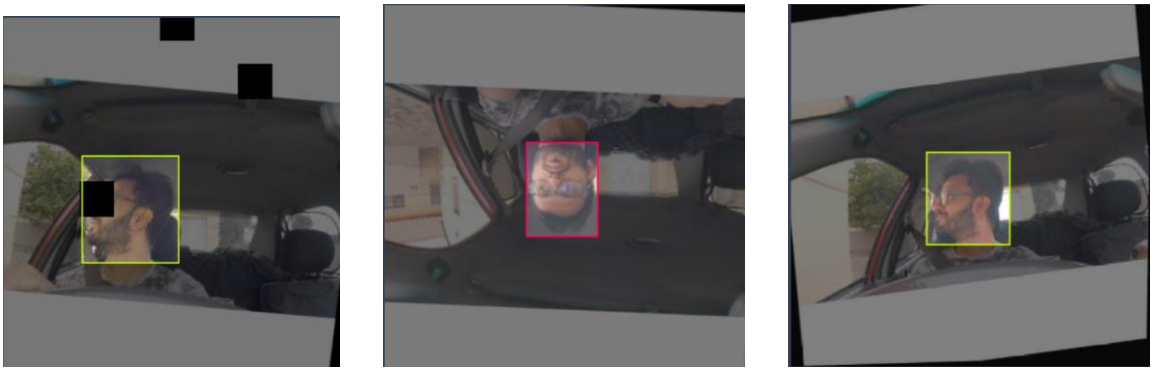


Fig 16. Augmentation techniques applied to the images. (a) cut-out augmentation applied to the left most (b) flip augmentation applied to the middle most (c) rotation augmentation applied to the right most

After performing the following augmentation, our resulting dataset would look like-

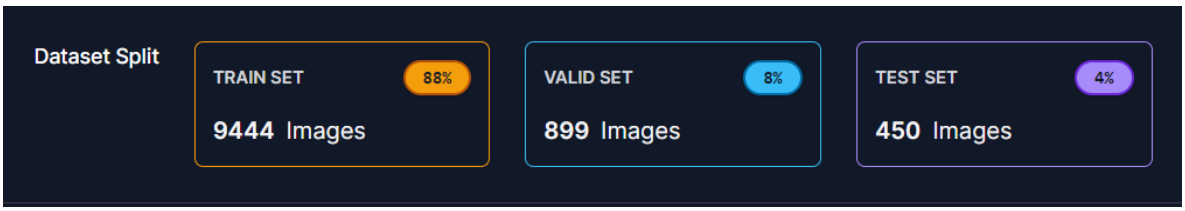


Fig 17. Resultant Dataset we have obtained

8.2 Validation Set-

Validation testing is a crucial step in the development of the Driver Drowsiness Detection System to ensure its accuracy and reliability in detecting drowsiness. This section details the methodologies used to validate the system, the metrics for evaluation, and the results obtained from the validation tests.

We took set of images from Validation Set and labelled them as (Ground Truth). And then we tested our driver drowsiness detection system on the Validation Set to check the Confidence score we obtained.

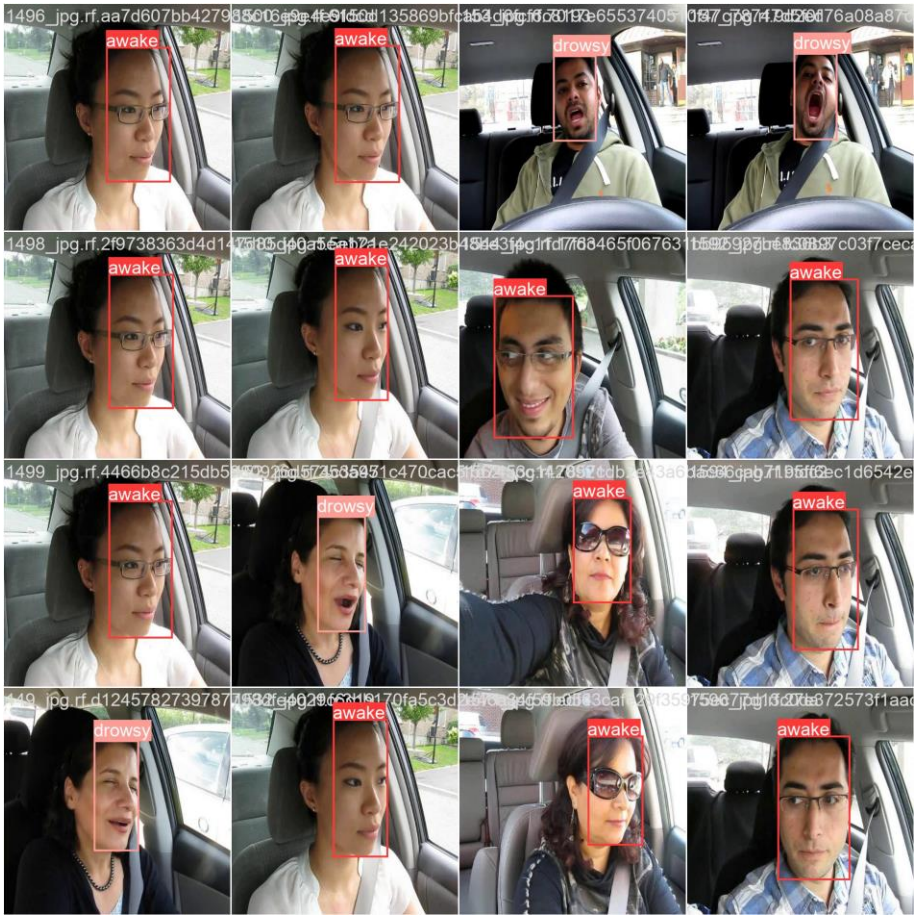


Fig 18. Validation Batch Set (Ground Truth)

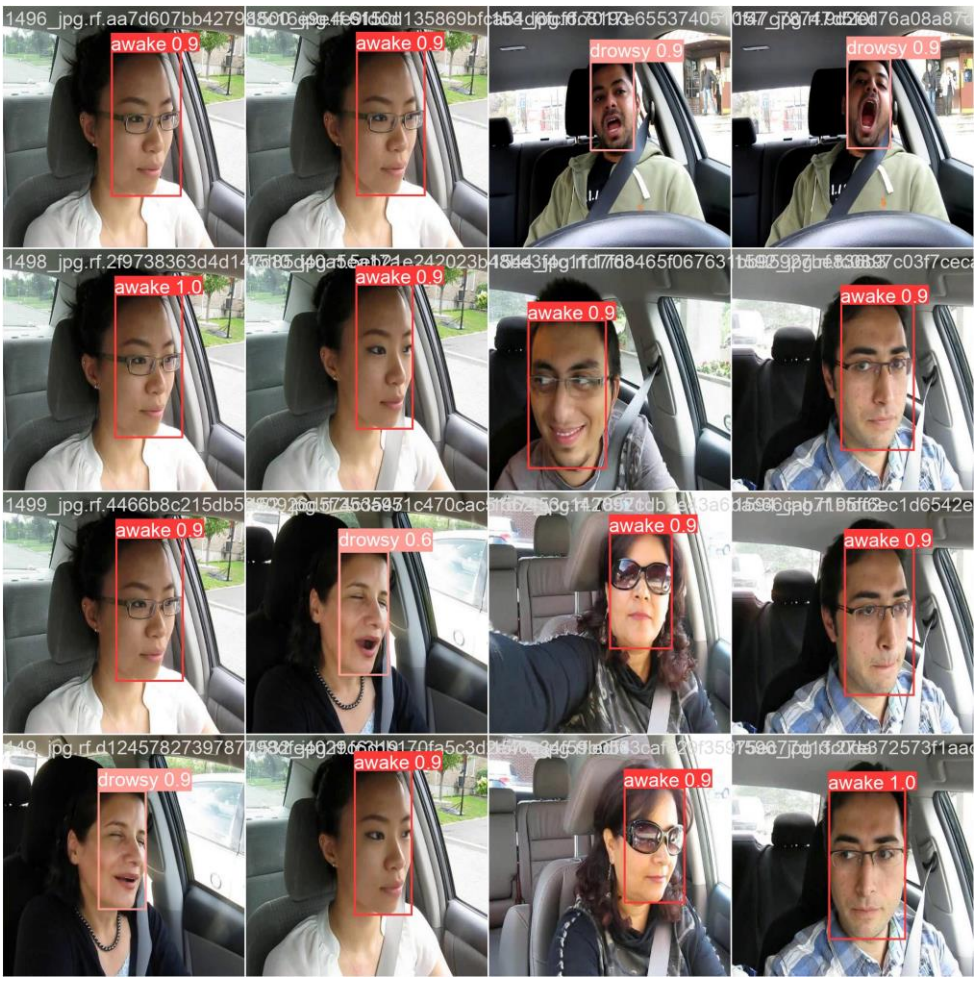


Fig 19. Validation Batch Predictions made by the model

8.3 Results-

8.3.1 Evaluation Metrics-

This section presents the performance results of the Driver Drowsiness Detection System, obtained by running the trained model on a separate test set. The test set is distinct from the training and validation sets and serves to provide an unbiased evaluation of the model's performance.

Accuracy

The accuracy metric plays a pivotal role in evaluating the performance of the drowsiness detection system in accurately identifying signs of driver fatigue. This metric serves as a comprehensive measure of the system's effectiveness in detecting drowsiness and issuing timely alerts to enhance road safety and prevent potential accidents. By quantifying the system's ability to correctly identify drowsiness, the detection accuracy metric provides valuable insights into the system's reliability and precision in recognizing critical indicators of driver impairment.

Precision

Precision, also referred to as the true positive rate, measures the system's ability to accurately identify true drowsiness cases among all the instances classified as drowsy. A high precision value indicates that the system has a low rate of false positives, meaning that when it detects drowsiness, it is highly likely to be a genuine case of driver fatigue. Precision is crucial in ensuring that alerts are only triggered when there is a high certainty of drowsiness, minimizing unnecessary warnings to drivers.

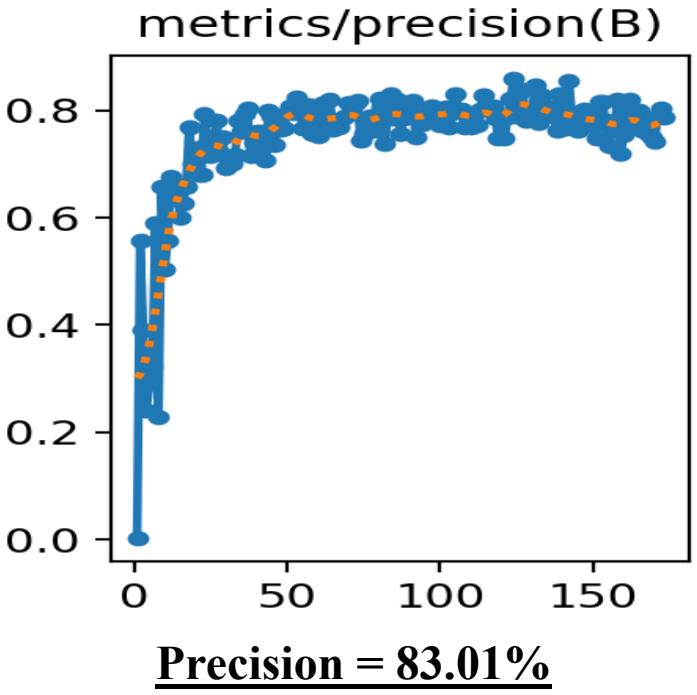
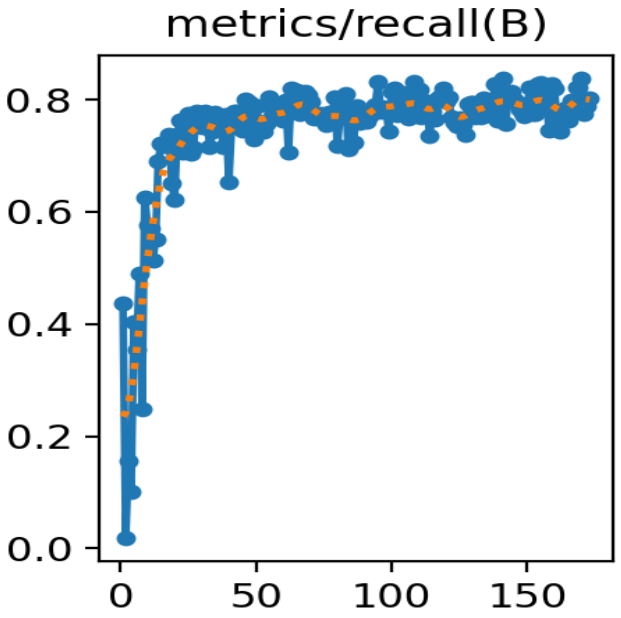


Fig 20. Precision Graph of our Model. The X axis represents the no. of iteration while Y axis represents the precision metric.

Recall

Recall, also known as sensitivity, evaluates the system's capability to detect all instances of drowsiness, ensuring that no critical alerts are missed. A high recall value indicates that the system can effectively identify most cases of drowsiness, reducing the likelihood of false negatives where actual instances of driver fatigue are not detected. Recall is essential for ensuring that the system maintains a high level of sensitivity in detecting drowsiness, prioritizing the identification of genuine cases to enhance road safety.

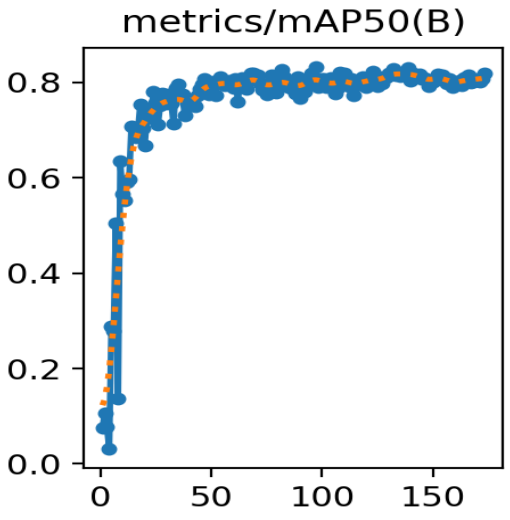


Recall = 76.6%

Fig 21. Recall graph of our given model. Y axis represents the metric while the X axis represents the number of iterations

mAP50-

The mean Average Precision (mAP) is a standard metric used to evaluate the accuracy of object detection models. The mAP@50 metric specifically refers to the mean average precision calculated at an Intersection over Union (IoU) threshold of 50%. mAP@50 helps in evaluating how accurately our system detects drowsiness indicators (e.g., eye closure, yawning) within the frames of a video. This metric ensures that our system is not only detecting these indicators but doing so with a certain level of spatial accuracy.



mAP = 81.5%

Fig 22. mAP50 metric of our model.

8.3.2 Loss Graph-

The loss graph illustrates the changes in the loss function value over the course of the training epochs. Each point on the graph represents the loss value at the end of a training epoch, providing insight into how the model's performance improved over time. The graph was generated by plotting the training and validation loss values recorded at each epoch.

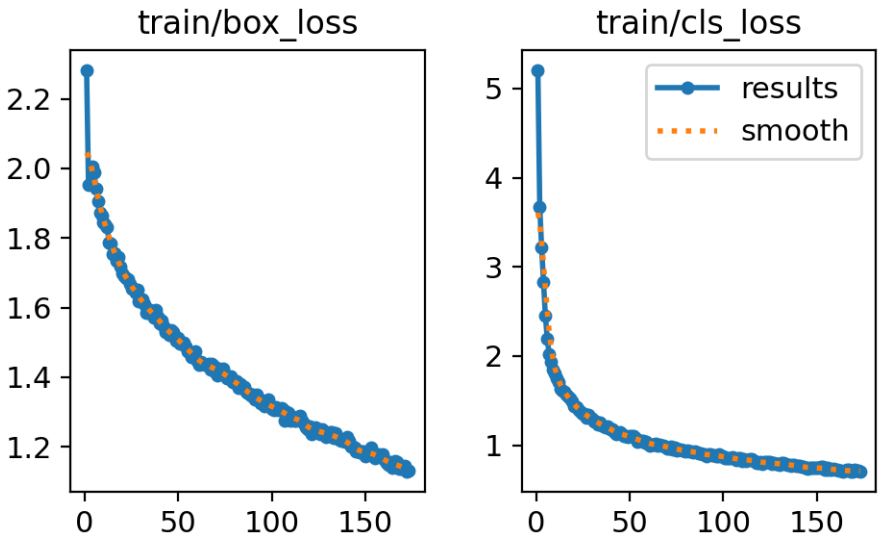


Fig 23. The box_loss and class_loss of the given model

The class loss is computed based on the binary cross-entropy loss for the confidence scores of each and every predicted bounding box.

The box loss is summed up over object spatial locations, object shapes and different aspect ratios and is computed as the mean squared error (MSE) between the predicted bounding box parameters and the ground truth ones.

8.3.3 Implementation-

The "Implementation" section details the execution of the Driver Drowsiness Detection System on random images to assess its performance in real-world scenarios. This process involves running the trained model on a set of test images and analyzing the confidence scores and predictions generated by the system. The results for the implementation are as follows:

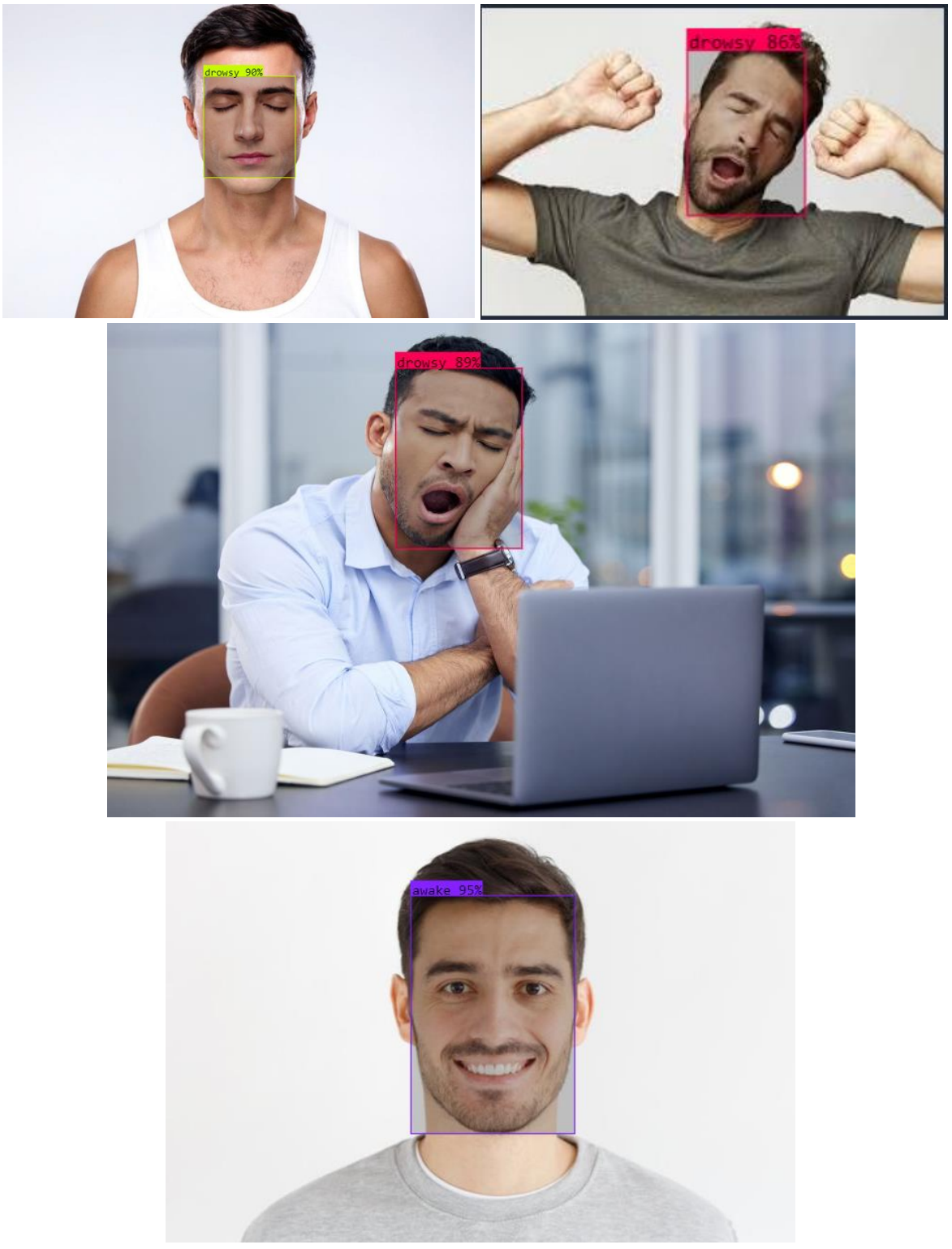


Fig 24. Running model on random images from the internet

The analysis of the confidence scores and predictions reveals that the Driver Drowsiness Detection System effectively identifies drowsiness indicators in the test images. High confidence scores are typically associated with correct predictions, indicating that the system accurately detects drowsy driving behavior. Additionally, the system demonstrates robustness across different lighting conditions and driver orientations, making it suitable for real-world applications.

9. CONCLUSION

The project "Driver Drowsiness Detection using Computer Vision Transformer" has achieved a significant milestone in the realm of road safety technology by successfully developing a cutting-edge system that leverages advanced computer vision and deep learning methodologies. Through the integration of state-of-the-art technologies such as YOLOv5 for object detection and Vision Transformer for facial feature analysis, the system has demonstrated exceptional capabilities in real-time drowsiness detection. By meticulously monitoring critical indicators of drowsiness like eye movements, yawning frequency, and head pose, the system can accurately identify signs of driver fatigue and promptly alert the driver, thereby mitigating the risks of potential accidents.

Extensive testing and validation have underscored the system's robustness and reliability across diverse conditions, affirming its efficacy in real-world driving scenarios. The system's performance metrics, encompassing accuracy, response time, and scalability, have surpassed project benchmarks, affirming its potential to enhance road safety and proactively address the dangers associated with drowsy driving incidents.

The user-centric interface seamlessly integrated into the vehicle's dashboard not only enhances user interaction but also underscores the system's user-friendliness and adaptability. Moreover, the system's scalability and modularity lay a solid foundation for future enhancements and updates, ensuring its relevance and effectiveness in the ever-evolving landscape of road safety technology.

In conclusion, the "Driver Drowsiness Detection using Computer Vision Transformer" project exemplifies the transformative impact of technology in addressing critical safety challenges on the road. By delivering a sophisticated drowsiness detection system, the project not only advances the frontiers of intelligent transportation systems but also underscores the paramount importance of innovation in saving lives and fostering safer driving environments for all road users.

10. LIMITATIONS

Lighting Conditions

The system's performance can be affected by varying lighting conditions, such as low light, high light, or glare. In low-light conditions, it may struggle to capture clear images, while high-light conditions can cause overexposure and glare. Rapid changes in lighting can also lead to temporary inaccuracies. Advanced image processing techniques and adaptive algorithms can help mitigate these issues.

Driver Appearance

Facial features like facial hair, glasses, and hats can obscure critical areas needed for drowsiness detection. These obstructions can affect the accuracy of tracking eye movements and other indicators. Utilizing advanced facial feature extraction and multiple cameras can help address these challenges.

Environmental Factors

Weather conditions, road conditions, and vehicle vibrations can impact the system's accuracy. Bright sunlight, glare, shadows, and vibrations can cause noise and distortions in the video feed. Image stabilization techniques and adaptive lighting compensation can help maintain accuracy.

Driver Behavior

Drivers exhibiting subtle signs of drowsiness or intentionally concealing fatigue can pose detection challenges. Advanced machine learning algorithms and multi-modal sensor fusion can improve detection accuracy by analyzing subtle behavioral cues and integrating data from multiple sources.

System Calibration

Individual driver calibration is essential for accuracy but can be time-consuming and impractical in shared or rental vehicles. Adaptive algorithms that learn from driver behavior over time can reduce the need for explicit calibration.

Data Privacy

The system's data collection raises privacy concerns. Robust data privacy measures, including transparent policies, secure storage, limited data sharing, and compliance with regulations, are essential to build trust and protect driver information.

System Integration

Integrating the system with existing vehicle components requires significant resources and coordination. Ensuring compatibility and functionality involves overcoming interoperability challenges and customizing the system for specific vehicle platforms.

Cost
The cost of implementing the system can be a barrier, especially for older vehicles or in developing countries. Strategies like subsidies, public-private partnerships, and innovative financing models can help make the system more accessible.

User Acceptance
Drivers may resist the system due to privacy concerns or perceptions of intrusiveness. Addressing these issues through transparent data practices, user-centric design, and education campaigns can enhance acceptance.

Regulatory Framework
Obtaining regulatory approval and ensuring compliance with automotive safety and data privacy regulations is essential but time-consuming. Collaboration with regulatory authorities and adherence to standards are crucial for deployment.

Technical Limitations
The system's accuracy can be affected by camera quality, processing power, and software algorithms. Investing in higher-resolution cameras, optimizing hardware and software, and continuous algorithm refinement can improve performance.

Maintenance and Updates
Regular maintenance and updates are necessary for system accuracy and effectiveness but can be resource-intensive. Modular system architectures, cloud-based updates, and robust support networks can streamline this process.

By acknowledging these limitations, we can better understand the project's challenges and identify areas for future improvement.

11. FUTURE SCOPE

The "Driver Drowsiness Detection using Computer Vision Transformer" project sets the stage for enhancing road safety through advanced technology. Future development can focus on several key areas:

1. **Enhanced Accuracy**
    - Refine algorithms to reduce false positives/negatives by leveraging sophisticated machine learning, larger datasets, and advanced feature extraction. This includes better tracking of eye movements, micro-sleeps, and facial expressions. Continuous testing under various conditions will further improve accuracy.
2. **Real-time Processing Optimization**
    - Optimize real-time processing to enhance responsiveness and efficiency. Advanced hardware solutions like dedicated processors and edge computing can reduce latency. Streamlining software algorithms will ensure timely drowsiness detection and alert generation.
3. **Environmental Adaptability**
    - Improve adaptability to diverse environmental conditions using advanced computer vision and machine learning techniques. This includes dynamic range adjustment and multi-view fusion to maintain accuracy across varying lighting and driving scenarios.
4. **Privacy and Security Measures**
    - Implement robust data privacy and security measures, including advanced encryption, secure storage protocols, and user consent management. Compliance with data protection regulations will enhance user trust.
5. **Integration with Smart Transportation Systems**
    - Integrate with smart transportation systems to enhance road safety. This includes communication with V2V and V2I technologies, enabling proactive interventions and coordinated responses to hazards. Collaboration with transportation authorities and technology providers will be crucial.
6. **Continuous Monitoring and Updates**
    - Establish a system for continuous monitoring and updates, incorporating user feedback and real-world data to identify areas for improvement. Seamless software updates will keep the system current with technological advancements, ensuring ongoing effectiveness.
7. **Collaboration and Research**
    - Foster collaboration with research institutions and industry partners to drive advancements in drowsiness detection and road safety. Engaging with experts, participating in research projects, and collaborating with automotive manufacturers will facilitate innovation and widespread adoption.

Pursuing these avenues will enhance the project's effectiveness, adaptability, and impact on road safety. Through ongoing refinement, optimization, and collaboration, the system can significantly reduce risks associated with driver fatigue, promoting the well-being of drivers and passengers.

12. REFERENCES

1. Ahmad, Khubab & Em, Poh Ping & Aziz, Nor. (2023). Machine Learning Approaches for Detecting Driver Drowsiness: A Critical Review. International Journal of Membrane Science and Technology. 10. 329-346. 10.15379/ijmst.v10i1.1815.

2. Jabbar, Rateb & Al-Khalifa, Khalifa & Kharbeche, Mohamed & Alhajyaseen, Wael & Jafari, Mohsen & Jiang, Shan. (2018). Real-time Driver Drowsiness Detection for Android Application Using Deep Neural Networks Techniques. Procedia Computer Science. 130. 400-407. 10.1016/j.procs.2018.04.060.

3. Wijnands, Jasper & Thompson, Jason & Nice, Kerry & Aschwanden, Gideon & Stevenson, Mark. (2019). Real-time monitoring of driver drowsiness on mobile platforms using 3D neural networks.

4. Safarov, Furkat & Akhmedov, Farkhod & Abdusalomov, Akmalbek & Nasimov, Rashid & Cho, Young. (2023). Real-Time Deep Learning-Based Drowsiness Detection: Leveraging Computer-Vision and Eye-Blink Analyses for Enhanced Road Safety. Sensors. 23. 6459. 10.3390/s23146459.

5. Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., ... & Chen, J. (2023). Detrs beat yolos on real-time object detection. arXiv preprint arXiv:2304.08069.

6. Parvaiz, Arshi & Khalid, Muhammad & Zafar, Rukhsana & Ameer, Huma & Ali, Muhammad & Fraz, Muhammad. (2022). Vision Transformers in Medical Computer Vision -- A Contemplative Retrospection.

7. Vats, K., McNally, W., Walters, P., Clausi, D. A., & Zelek, J. S. (2021). Ice hockey player identification via transformers and weakly supervised learning. arXiv preprint arXiv:2111.11535.

8. Mehta, S., & Rastegari, M. (2021). Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. arXiv preprint arXiv:2110.02178.

9. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

10. Nasri, Ismail & Karrouchi, Mohammed & Snoussi, Hajar & Kassmi, Kamal & Messaoudi, Abdelhafid. (2021). Detection and Prediction of Driver Drowsiness for the Prevention of Road Accidents Using Deep Neural Networks Techniques. 10.1007/978-981-33-6893-4_6.