Uniosun Journal of Engineering and Environmental Sciences (UJEES), Vol. 7 No. 2, Sept. 2025
DOI: https//doi.org/10.64980/ujees.v7i2.452 **Akinde et al.**
Published: September 30, 2025

AN INTELLIGENT-BASED ALGORITHM FOR DETERMINING DROWSY DRIVERS AND PREVENTION OF ROAD ACCIDENTS

Olusola K. AKINDE[1*], Toheeb A. OLALEYE[2], Michael O. IBITOYE[2], Victor RIZAMA[2], Sam TAIWO[3], Moses O. ADETONA[4]

[1]*Department of Electrical and Biomedical Engineering, Abiola Ajimobi Technical University, Ibadan, Nigeria*
[2]*Department of Electrical and Electronics Engineering, Federal University of Technology, Minna, Nigeria*
[3]*Department of Agricultural and Biosystems' Engineering, Olabisi Onabanjo University, Ibogu, Nigeria*
[4]*Department of Anatomy, College of Medicine, University of Ibadan, Nigeria*

[*]*Correspondent Author: olusola.akinde@tech-u.edu.ng*

Abstract
Driving demands full attention and focus, as any decrease in these could lead to accidents. Drowsiness, a condition that reduces a motorist's alertness, poses a significant risk, with delayed detection often resulting in accidents. Traditional methods for detecting drowsiness typically rely on sensors or invasive monitoring, which may not always be practical in everyday situations. This research aims to develop an intelligent algorithm for detecting drowsy drivers and preventing road accidents. The study adopts a physiological approach, utilizing computer vision and deep learning techniques to monitor and identify signs of drowsiness in real-time. The proposed method employs camera-based monitoring to detect symptoms such as yawning, heavy eyelids, and closed eyes from a large dataset of images of drowsy drivers. By using the advanced object detection model YOLOv8s, the system takes advantage of its real-time, high-accuracy facial landmark detection to analyze and assess the drowsiness levels of individuals. The model's performance, with precision at 0.895, recall at 0.914, and an F1 score of 0.9, demonstrates its effectiveness in accurately identifying drowsiness across various environmental conditions. This non-intrusive, efficient approach provides a viable solution for real-time, low-cost drowsiness detection, offering promising applications in driver's safety and autonomous vehicle technology.

Keywords
Drowsy drivers, Object detection, YOLOv8s, Non-intrusive

1. INTRODUCTION

Researchers have defined drowsiness in various ways, generally describing it as a biological state where the body transitions from wakefulness to sleep. It is characterised as a condition of sleepiness requiring rest, leading to symptoms that significantly affect performance. Additionally, drowsiness is understood as an inclination to fall asleep and a state in which awareness decreases due to sleep deprivation or fatigue. A common factor across these definitions is that drowsiness is primarily caused by lack of sleep, exhaustion, or fatigue [5], [9], resulting in reduced alertness - posing a serious risk to drivers, other road users, and property. The American National Highway Traffic Safety Administration (NHTSA) estimates that drowsy driving is responsible for approximately 100,000 accidents each year, leading to over 1,550 fatalities, 71,000 injuries, and $12.5 billion in property damage [11]. Moreover, the U.S. National Safety Council indicates that a significant percentage of drivers admit to falling asleep behind the wheel at least once a month, with 4% of these cases resulting in accidents. Research by the American Automobile Association (AAA) Foundation for Traffic Safety estimates that 320,000 drowsy driving crashes occur annually - more than three times the officially reported number. Of these, 109,000 crashes resulted in injuries, while approximately 6,400 were fatal. Researchers suggest that the true prevalence of drowsy driving fatalities may be over 350% higher than reported [1]. These figures are concerning, especially considering that a recent study found that 60% of adults admitted to driving while fatigued, and 37% acknowledged having fallen asleep at the wheel.

Drowsiness stages are generally classified into three categories: awake, non-rapid eye movement (NREM) sleep, and rapid eye movement (REM) sleep. NREM sleep is further divided into three distinct stages [13]: I: The transition from wakefulness to sleep, often associated with drowsiness, II: Light sleep, where heart rate and body temperature decrease, and III: Deep sleep, essential for physical and mental restoration. One of the primary challenges in developing an effective drowsiness detection system is acquiring accurate drowsiness

data. For safety reasons, inducing drowsiness in real-world driving conditions is not feasible, making it difficult to collect reliable data. As a result, drowsiness detection systems must be designed based on how drivers naturally enter a drowsy state, ensuring the accuracy and effectiveness of detection methods. Driver fatigue can be detected early, allowing for timely alerts that can help prevent potential accidents [5]. Drowsy drivers exhibit various signs, including frequent yawning, constant eye closure, and drifting out of their lanes.

Models for detecting driver drowsiness have been developed over the years. To avoid mishaps, researchers have proposed several models to identify sleepiness symptoms as soon as possible. The models for detecting driver drowsiness are classified into two, these are invasive and non-invasive. These can be further categorized into four, namely: image-based models, which use cameras to analyze the driver's facial expressions and movements; biological-based models, which use sensors on the driver's body to monitor bio-signals; vehicle-based models, which track the movement and behavior of the vehicle; and hybrid models, which combine two or more of these models [16]. Several researchers have made significant efforts in the accident detection and prevention schemes. Brief about their findings are referenced for inclusiveness. Invasive methods have utilized for drowsiness detection. It involves the use of sensors attached to or worn by drivers [2], leveraging majorly on IoT technologies [3] to monitor physiological and biological indicators of drowsiness. An IoT-based drowsiness detection system was designed in [23], it utilized the Wemos D1 Pro Esp8266 microcontroller, the MAX30102 sensor, and the Blynk application for real-time visualization. The system determines a driver's drowsiness level by analyzing heart rate data collected via the detection device. In another scheme, an innovative model was integrated into a real-world carriage system [8]. This system includes a wearable device and an embedded unit equipped with GPS, accelerometers, and gyroscopes sensors, and are installed in a Smart Box mounted on a bus. The wearable device constantly monitors the driver's heart rate and relays alerts in case of drowsiness to help prevent accidents. Also, if an accident occurs, the embedded system activates an alert function which promptly relay the incident's geolocation to a registered phone number via a mobile messaging service. In a parallel instant, a prototype system was developed to detect driver's drowsiness by analyzing facial landmarks, specifically focusing on the eye region, eye aspect ratio, and closure patterns. When signs of drowsiness are detected, the system issues an audible alert to the driver and notifies a third party [18]. An IoT-based vehicle monitoring and driver's assistance framework was designed by [15]. It was aimed at enhancing safety and smart fleet management. Their system incorporates a software algorithm for tracking of iris status, alongside hardware components such as cameras, a GPS module, a GSM communication module, and a microcontroller. These elements enable intelligent vehicle functionalities, including driver's identification, health monitoring, and over-speed detection. A "Stay Alert: Drowsiness Detection with IoT Technology" system designed by [14] facial recognition cameras, heart rate monitors, and accelerometers were integrated into an IoT network. This system continuously tracks physiological and behavioral indicators of drowsiness, such as eye movement, blink frequency, heart rate variations, and head position changes. Physiological monitoring schemes have also been proposed, utilizing signals such as electrooculogram (EOG) and electrocardiogram (ECG) to provide valuable insights into the driver's fatigue levels. But, a key restraint of these methods is the need for the driver to wear electrodes, which may be discomforting and tiring [17].

The reviewed non-invasive studies focused on various algorithms that uses camera-based machine-learning techniques for drowsiness detection [10]. a non-intrusive Advanced Driver Assistance System (ADAS) was designed to detect and alert drivers experiencing drowsiness. The approach used involved analyzing 60-second sequences of recorded facial images. To minimize false positives, they applied two alternative detection solutions, utilizing recurrent and convolutional neural networks along with deep learning schemes to extract numerical features from captured images, which are then processed using a fuzzy logic-based system. Two methods were proposed in [12] with three scenarios for drowsiness alert systems. The first method utilized facial landmarks to detect blinks and yawns based on adapted threshold values for each driver. The second method employed deep learning techniques, using two adaptive deep neural networks based on MobileNet-V2 and ResNet-50V2. This approach processes video frames to automatically learn driver's activity features. By leveraging transfer learning, the system addresses the challenge of limited training datasets, ensuring faster training while keeping the benefits of deep neural networks. In the study carried out by [21], a deep-learning-based driver drowsiness detection system for brain-computer interfaces (BCI) using functional near-infrared spectroscopy (fNIRS) was adopted. Brain signals related with drowsiness were collected from 13 healthy participants while they operated a car simulator. A continuous-wave fNIRS system focused on the prefrontal and dorsolateral prefrontal cortices to measure brain activity. Deep neural networks (DNN) were used to classify drowsy and alert states, while convolutional neural networks (CNN) processed color map images to identify the most relevant channels for detecting brain activity over time. Their CNN-based model achieved an outstanding average accuracy of 99.3%, demonstrating its capability to distinguish between drowsy and alert states aptly. The study in [19] advanced various architectures for analyzing facial and drowsiness detection performance using deep learning techniques. To assess the driver's state, facial regions covering the entire face were utilized. The face detection algorithms applied include: Viola-Jones, DLib, and YOLOv3. For grouping, a

revised LeNet Convolutional Neural Network (CNN) architecture was used for drowsiness detection. The study showed an approach to identifying a drowsy driver using an NIR camera. Among the face detection methods evaluated, YOLOv3 outperformed Viola-Jones and DLib in terms of accuracy. The modified LeNet model was used for taxonomy, achieving a system accuracy of 97% at 20 frames per second. But, in cases where the face was partly occluded, such as by a hand, detection was impaired [19].

Despite the feats by some researchers in driver's drowsiness detection schemes, realising an optimal balance between accuracy, practicality, and reliability is a challenge. Vehicle-based systems are often impacted by external factors such as road conditions, while physiological methods, though highly accurate, rely on invasive equipment like electrodes, making them impractical for everyday use. These constraints highlight the need for efficient and non-intrusive method. This study thus aims to design a Driver Drowsiness Detection (DDD) System that leverages computer vision and deep learning to precisely identify signs of drowsiness in real time using a camera-based monitoring setup. This approach aims to maintain high accuracy across diverse eco-friendly conditions while ensuring drivers' comfort and removal of intrusive devices.

2. MATERIALS AND METHOD

The detection of a drowsy driver was achieved using object detection machine learning based model weaved around deep learning paradigm, a powerful branch of artificial intelligence. This method involved training an artificial neural network model with multiple layers to automatically learn and extract patterns associated with drowsiness symptoms, such as yawning, heavy eyelids, and closed eyes, from a large dataset of drowsy driver images [4], [7]. The process begins with acquiring image data captured by cameras of drowsy drivers in various environmental, lighting, and positional conditions. These images were then pre-processed through tasks like data cleaning, resizing, augmentation, class balancing, and validation. During training, the model learns to detect and classify drowsiness symptoms by extracting key features from the pre-processed images. Post-processing techniques, such as model fine-tuning, hyper-parameter tuning, and error analysis are applied to optimize performance. The model is then tested on diverse datasets to evaluate its robustness in detecting drowsy driver symptoms under varying conditions, ensuring its effectiveness in real-time monitoring [6], [7], [19]. Figure 1 is the process flow illustrating the model:

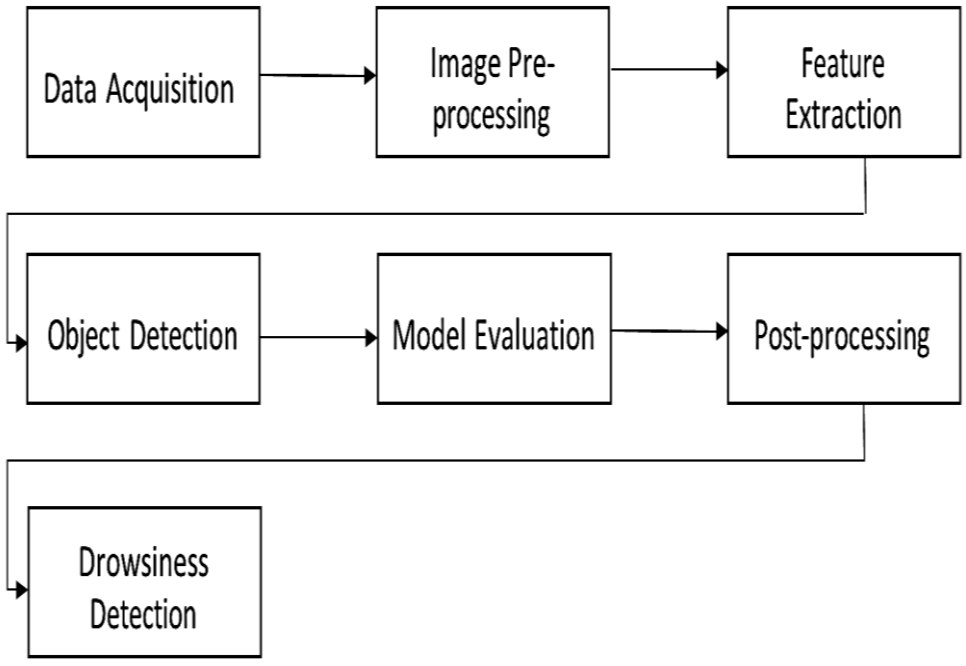

Figure 1: Driver Drowsiness Detection Pipeline Flow

2.1 You Only Look Once Version 8 (YOLO v8)

The drowsiness detection system in this study utilizes the "You Only Look Once" version 8 (YOLO v8), a cutting-edge single-stage object detection algorithm known for its speed and efficiency in real-time applications. Built on a Convolutional Neural Network (CNN) and tailored for processing visual data [2], YOLO's primary advantage is its ability to detect objects in a single forward pass through the network, removing the need for multiple processing stages. Unlike old two-stage detectors such as RCNN, YOLO parts an image into a grid, where each section predicts bounding boxes and class chances for objects within its area. To refine detections,

non-maximum suppression (NMS) was used, removing overlapping boxes to ensure each object is identified only once. This streamlined method (Figure 2) enables YOLO to deliver rapid and precise forecasts, making it highly apt for real-time driver drowsiness detection [6], [20] and [22].

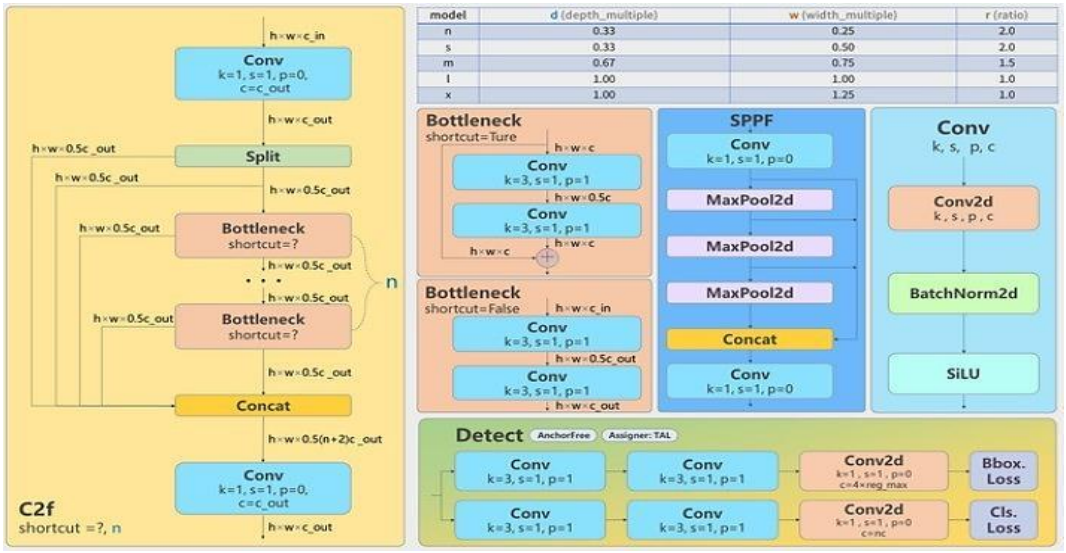

Figure 2: Architecture of YOLO v8.

The YOLO v8 algorithm as depicted in Figure 2.0 consists of 3 major blocks through which all processing is done namely: the backbone, the neck, and the head.

i. **Backbone:** The backbone, also known as the feature extractor, is essential for identifying meaningful patterns from an input image of a drowsy driver. It starts by detecting basic elements like edges and textures in the initial layers and gradually extracts features at various scales to enhance the accuracy of detection. Finally, it generates a rich hierarchical representation of the input image, serving as the foundation for subsequent processing stages.

ii. **Neck:** The neck serves as a link between the backbone and the head, enabling feature fusion and improving contextual awareness. By integrating feature maps from different stages of the backbone, it enhances the network's ability to accurately detect objects of various sizes. Next, it incorporates contextual information to improve detection accuracy by accounting for the broader scene context. Lastly, the neck reduces spatial resolution and dimensionality, optimizing computational efficiency and increasing processing speed.

iii. **Head:** The head is the concluding part of the network, tasked with generating the final outputs. It predicts bounding boxes for potential objects in the image and assigns confidence scores to indicate the probability of an object's presence. Additionally, it classifies the detected objects within the bounding boxes according to their respective categories.

In this research project, YOLO v8s is the specific variant of the YOLO model used for driver drowsiness detection due to its lightweight nature and optimized speed. Compared to the other YOLO v8 variants such as YOLO v8l, and YOLO v8x, YOLO v8s is specifically designed for faster inference while maintaining sufficient accuracy. Its smaller model size allows it to operate efficiently on devices with limited computational resources, such as mobile processors or edge devices, ensuring that the systems can process live video data in real-time. This is particularly important in driver drowsiness detection, where quick identification of symptoms like eyelid movements of facial expressions can be captured. The fast inference time provided by YOLO v8s minimizes latency, making it highly suitable for safety-focused applications where even a slight delay in detection could be hazardous [22]. On the other hand, larger models like YOLO v8x, or YOLO v8l while providing better accuracy, require more computational power and longer training and processing times, which can slow down the system in real-time applications. Thus, YOLO v8s achieves an ideal balance between speed, efficiency, and detection accuracy, making it the most suitable option for real-time driver drowsiness detection.

2.2 Data Acquisition

A dataset comprising 18,140 images of drivers was collected, featuring different states of a driver such as "happy", "Neutral", "Eyes closed", "Heavy Eyes", "Yawn" and "Bent Neck". These images were sourced from diverse online platforms, including Roboflow, Kaggle, Google Datasets, and YouTube. Figure 3.0 depicts various samples from the dataset collection and Table 1.0 shows class distribution by image.

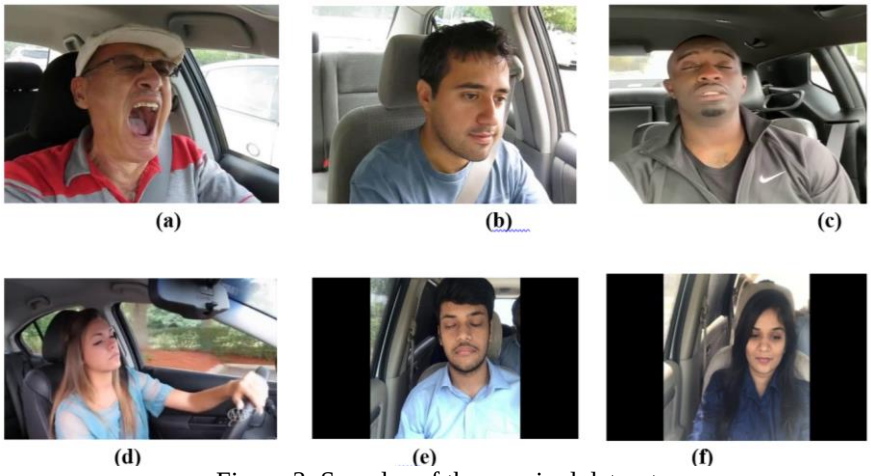

Figure 3: Samples of the acquired dataset

Table 1: Class Distribution by Images

Class		Images	Total	Percentage (%)
Drowsy Symptoms	Eye Closed	3299	10104	55,700
	Bent Neck	11		
	Heavy Eyes	3908		
	Yawn	2886		
Non-Drowsy Symptoms	Happy	111	8036	44,300
	Neural	7925		

2.3 Data Cleaning and Structuring

To properly prepare the data for further use, the 18,140 images were adequately cleaned and sorted into different sections based on the condition of the driver by removal of blurred images or images of low resolution, irrelevant images that did not depict any drowsy symptoms, etc. Moreover, due to the minimal representation of the "Bent Neck" category (11 images), it was appropriately excluded from the dataset to avoid negatively affecting the dataset. The remaining data was re-organized into four symptom categories: "yawn", "eyes closed", and "heavy eyes" indicating drowsiness while the "eyes opened" category predominantly represented non-drowsiness. This is because the initial "neutral" and "happy" sections were merged to collectively represent non-drowsy symptoms.

The inclusion of the non-drowsy symptoms i.e. negative (-ve) samples was essential for improving the model's robustness and overall performance. By integrating these negative categories, the model was adept at differentiating between drowsy and non-drowsy drivers significantly enhancing its generalization capabilities and detection accuracy.

2.4 Labeling/ Annotation

The dataset of 18, 140 images was carefully labeled in the YOLO format using the Labeling software application as shown in Figure 4.0. The software is an open-source graphical annotation tool that facilitates the labeling and annotation of images with bounding boxes. Each time a label or annotation is created using the software, it automatically exports the parameters of the labeled objects to a .txt file for storage. For every annotated image, the associated class ID, along with the x and y coordinates, width, and height of the bounding box, was documented in a `.txt` file. These details provided the precise information required to define the bounding boxes, enabling the model to accurately identify and localize objects within the image. Table 2 entails explanation of each of the data label.

Table 2: Data Labels and Description

Data Labels	**Description of Data Labels**
Class ID:	This parameter designates the category or label of the object detected in the image. Since the model aims to differentiate between drowsy and non-drowsy drivers, as well as various facial indicators, the class indicates which category the detected expression falls into. For example, a detected expression could be labeled as "drowsy" or "not drowsy," depending on the classification model's output.
X-coordinate	This parameter represents the x-coordinate of the top-left corner of the bounding box surrounding the detected object. It specifies the horizontal position of the bounding box's starting point, measured in pixels from the left edge of the image.
Y-coordinate	This parameter represents the y-coordinate of the bounding box's top-left corner, specifying the vertical position where the bounding box starts. It is measured in pixels from the top edge of the image.
Width	This parameter denotes the width of the bounding box, indicating how wide it is starting from the x-coordinate. The width is measured in pixels and helps determine the horizontal span of the detected object.
Height	This parameter represents the height of the bounding box, specifying how tall it is starting from the y-coordinate. The height is measured in pixels and aids in determining the vertical span of the detected facial expression.

Figure 4: Samples of the annotated (i.e. labelled) dataset.

2.5 Data Preparation

To adequately prepare the data for training, the following preparatory measures were put in place:

i. **Class Balancing:** To ensure accurate classification by the model, it was essential to achieve a balanced representation of all classes. This balance enhances the model's efficiency by ensuring an even distribution of weights across the different class categories (i.e., yawn, eyes closed, eyes opened … etc). Figure 5.0 depicts the class distribution of the dataset.

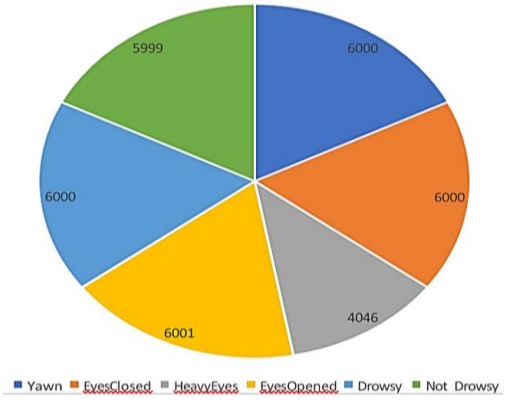

Figure 5: Class distribution of the dataset.

ii. **Label File Modification:** All bounding box annotations in the .txt files were formatted consistently as ***class_ID, x, y, width, and height*** to maintain uniformity across the dataset. Also, extra blank lines identified within the label files were removed to prevent potential errors during training. This change ensured that all label files adhered to the expected YOLO format.

2.6 Data Pre-Processing and Augmentation

Following data cleaning, structuring, and preparation phase, the entire dataset was uploaded to the Roboflow Workspace, a platform made to streamline computer vision workflows. Roboflow enabled the pre-processing and increase of the driver drowsiness dataset, ensuring that it was optimized for training the YOLO v8 model. The platform's robust tools enabled efficient handling of large datasets, while its pre-processing abilities ensured constancy and quality across all images. The specific pre-processing and structuring methods are stated below:

2.6.1 Pre-processing techniques

i. **Auto-Orientation:** This step automatically corrected the orientation of images regardless of the original orientation during capture while making sure that they were properly aligned for accurate detection during training.

ii. **Image Resizing:** All images were resized to a resolution of 640 x 640 pixels to align with YOLO v8 training standards and to ensure uniform dimensions across the dataset while facilitating quicker GPU training and preserving the critical details of each image.

2.6.2 Augmentation techniques

i. **Vertical Flips:** Images were flipped vertically to increase variability and improve the model's ability to generalize. It is mathematically represented as eq.1:

$$\begin{matrix} x \\ y \\ 1 \end{matrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{matrix} x \\ y \\ 1 \end{matrix} \quad (1)$$

$x'=y'$, $x'=-y'$

ii. **Rotation:** This technique was used to alter the orientation of images by rotating them at specified angles. This process generated new training data instances introducing variations in the images' positioning. By including rotated images in the dataset, the model learned to recognize objects regardless of their orientation, improving its ability to handle different perspectives and angles. The process is represented by the eqs.2 and 3 below:

$$x2 = \cos(\theta) * (x1 - x) - \sin(\theta) * (y1 - y) + x \quad (2)$$

$$y2 = \sin(\theta) * (x1 - x) - \cos(\theta) * (y1 - y) + y \quad (3)$$

iii. **Shear:** This method distorted the images slightly along a given axis, introducing variation and helping the model become resilient to minor distortions in real-world settings. Mathematically, it is described in eq 4:

$$xyy^n = [10\ k1]xy \quad (4)$$

where:

$x' = x + k * y$ (the new x - coordinate)

$x' = y$ (unchanged y - coordinate)

iv. **Gaussian Blur:** smooths images and reduces noise. This augmentation helps the model learn to identify important features even when images are slightly blurred or have low detail. It also simulates scenarios where the input data might be captured with poor focus. It is mathematically depicted in equation 5:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{x^2-y^2}{-2\sigma^2}} \quad (5)$$

v. **Median Blur:** Median filtering is especially effective at removing "salt and pepper" noise (random bright or dark pixels) while keeping the edges in an image. This technique is helpful for improving the model's performance in detecting features in noisy or degraded milieus. Mathematically, it is represented as eq.6:

$$New\ Value = Median\ (Window\ Values) \quad (6)$$

vi. **To Gray:** Converting images to grayscale removed color information, forcing the model to rely on texture, shape, and intensity patterns. This augmentation ensured the model should perform well even in settings where color data was unavailable or unreliable, such as low light conditions.

vii. **Contrast Limited Adaptive Histogram Equalization (CLAHE):** This enhances contrast in low-contrast images. This helps the model to detect subtle details that may be obscured in poorly lit or overexposed images. It also prepares the model to handle variations in lighting and exposure in real-world scenarios.The training batch image collage in the Figure 6 depicts some of the augmentation techniques applied. Following data pre-processing and augmentation, the original dataset of 18,129 images was successfully expanded to 20,927 images.

Figure 6: A training batch collage containing few augmentation techniques applied.

2.6.3 The Train-Test Split

The dataset, comprising 20, 927 images, was divided into three subsets using a ratio of 90:5:5. This split ensured an effective allocation of data for model training, validation, and testing,

i. **Training Set (90%):** The training set containing 18,835 images was the largest subset and was used to train the YOLO v8 model. It provided the model with sufficient examples to learn patterns, features, and relationships between the input data and corresponding labels.

ii. **Validation Set (5%):** The validation set, comprising 1,046 images, was used during training to assess the model's performance on unseen data. It played a crucial role in tuning hyperparameters, detecting overfitting, and ensuring the model could generalize well to new data.

iii. **Testing Set (5%):** The testing set, which also contained 1,046 images, was kept aside for the final evaluation of the model after training. It offered an unbiased measure of the model's accuracy, robustness, and capability to handle real-world data across various environmental conditions.

2.6.4 Model Building

The driver drowsiness detection model was developed on the Google Colab platform, utilizing an Intel® Core™ i5-7300U CPU @ 2.60GHz in conjunction with the Tesla T4 GPU accelerator. The Tesla T4 is equipped with 16 GB GDDR6 VRAM and is powered by 2560 CUDA cores with a boost clock of 1.59 GHz. This GPU was chosen for its high performance in deep learning tasks, offering fast computation due to its Turing architecture and support for CUDA 12.2. The deep learning model was developed using transfer learning, a potent approach that utilizes pre-trained models to accelerate the training process. Transfer learning enabled the pre-trained model to benefit from the knowledge embedded in models before trained on large datasets. This pre-existing knowledge allowed the model to quickly adapt to a new custom dataset with

minimal computational cost and reduced training time. By fine-tuning the final layers, the model was tailored to detect driver drowsiness even with the limited dataset.

However, before training the model, certain hyper-parameters were selected and these are shown in Table 3. Properly setting these hyper-parameters was crucial since they knowingly influenced the model's ability to learn from the data and converge to an optimal solution. In addition, the settings determined how quickly the model learned, how well it generalized to new data, and how effectively it avoided overfitting, thus playing a pivotal role in achieving accurate and reliable predictions.

Table 3: Hyper-parameters used and their respective values.

Number of Epochs.	**100**	Total number of complete passes through the network for every iteration
Learning Rates	0.01	Controls the step size for updating the weights during training.
Input images sizes.	640x640	The resolution of the input images
Total number of images	6	The resolution of the input images
Batch sizes	SILU	Sigmoid Linear Activation Function also known as Swish.
Momentum	0.9	Control input of the previous gradient to the current update, helping smoothen training and accelerate convergence.
Optimizer	SDG	Stochastic Gradient Decent.
Workers	8	The number of workers for data loading.
Pre-trained weights	349	Loaded weights from YOLO v8s model.
Layers	168	Total number of trainable and non-trainable parameters used in the YOLO v8 model architecture.
Parameters	11,127,906	Total number of trainable and non-trainable parameters used in the YOLO v8 model architecture
IoU Threshold	0.07	Intersection over Union threshold for detection during training.
Maximum Detections	300	Maximum Number of Detection per images.
Weight Decay	0.005	Regularization term added to lost role to avert overfitting by penalizing large weights.
Warmup Epochs	3.0	Number of Initial of Epochs with a gradually increasing rate to stabilize early training
Patience	100	Early Stopping Patience.

3. RESULTS AND DISCUSSION

The performance of the model was assessed using several key evaluation metrics, which include: Precision, Recall, F1 Score, IoU Intersection over Union (IoU), and Mean Average Precision. These metrics provided insights into how effectively the model had learned from the training data. Table 4.0 highlights the description of each metric and its significance.

Table 4: Model evaluation metrics

Metric	Values
Precision 0.	0.895
Recall	0.914
Map@50	0.763
Map@50-95	0.952
F1-Score	0.900
Box Loss	0.732
Class loss	0.400

3.1 Precision: measures the accuracy of the model's positive predictions. Specifically, it quantifies how many of the instances that the model flagged as indicating drowsiness (positive detections) are actually correct. The precision value of 0.895 as shown in Figure 7 indicates that for 89.5% of the times the model detected drowsiness, the driver was indeed drowsy (i.e., True Positive) leaving an error rate of about 10.5% (False Positives). Mathematically, it is calculated using eq.7:

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)} \quad (7)$$

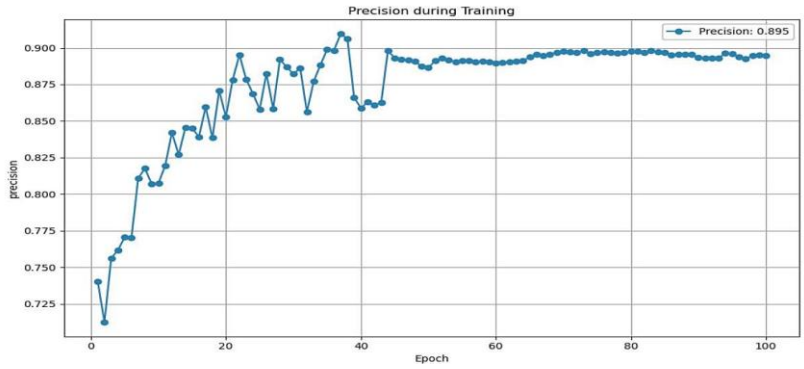

Figure 7: Precision curve during training evaluation metrics.

3.2 Recall: quantifies the model's ability to detect all actual instances of driver drowsiness. Contrariwise, it tells us how many of the truly drowsy drivers were correctly identified by the model. As shown in Figure 8, the model initially experienced a stochastic start due to its early phase of learning, where it was still trying to learn the underlying features across various categories. However, over time, the model stabilized, and its performance steadily improved, culminating in the final result. With the recall value of 0.914, the model successfully detected drowsiness 91.4% of the time when the driver was actually drowsy, the model successfully detected and flagged drowsiness. However, this also implies that 8.6% of the time, when the driver was indeed drowsy, the system failed to detect it (i.e. False Negatives). It is mathematically expressed as eq. 8 below:

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FP)} \quad (8)$$

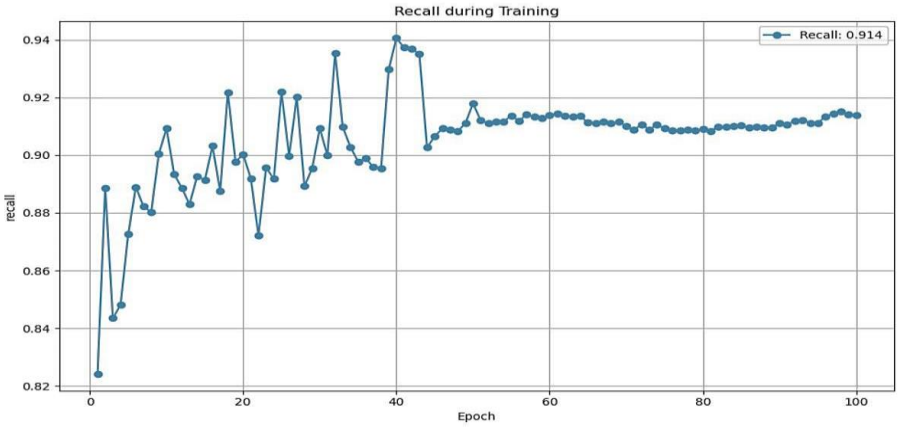

Figure 8: Recall curve during training evaluation metrics.

3.3 F1-Score: is the harmonic mean of precision and recall, offering a balanced measure of a model's ability to accurately identify positive instances while minimizing both false positives and false negatives. It ranges from 0 to 1, where a score of 1 indicates perfect precision and recall. The F1-score is calculated by using a decision boundary to classify predicted probabilities into distinct classes. In this scenario, the model correctly classifies an instance as belonging to a positive class if its predicted probability exceeds the threshold of 0.376. The overall score of 0.9 as indicated in Figure 9 shows that the model is highly effective in detecting drowsy drivers (high recall) and reducing false alarms (high precision) at the threshold of 0.376. Mathematically, it is calculated using eq. 9:

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

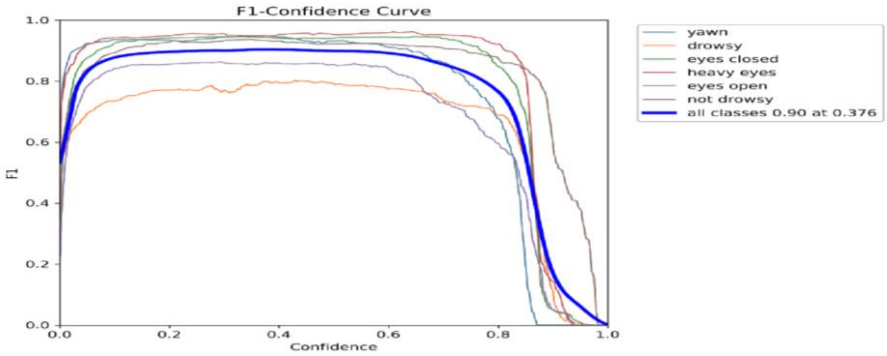

Figure 9: F1 Score - Confidence Curve During Training.

3.4 Mean Average Precision (mAP)@50: indicates the calculated value at an Intersection over Union (IoU) of 0.50 (i.e. 50%). This implied that when the model predicts drowsiness or a symptom like a yawn, eyes closed, heavy eyelids, etc, such is taken as precise detection if the predicted area overlaps with the ground truth by at least 50%. From the driver drowsiness model, the mAP@50 of 0.952 showed that the model is 95.2% accurate in identifying and localizing driver drowsiness and its signs with a good degree of overlap where a symptom is present. From Figure 10, it can be observed that at the beginning of the training, the mAP@50 experienced an abrupt increase and at some point from 40 epochs, the value stabilized and was on a constant increase afterward until it got to 100 epochs as shown in eq. 10.

$$mA\ P_{50} = \frac{1}{N}\sum_{50}^{n}(R_{50} - R) * P_{50} \tag{10}$$

where: R_{50} *is the recall at an IoU threshold of 50%*
P_{50} *is the precision at an IoU threshold of 50%.*
n signifies the number of classes (i.e 6) the model is detecting.

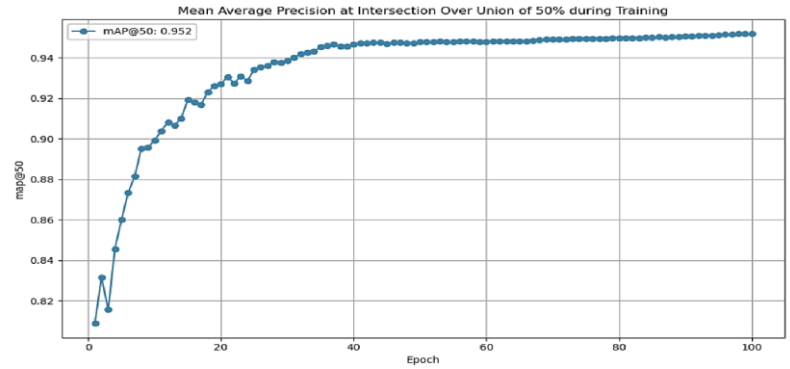

Figure 10: mAP@50 During Training

3.5 mAP@50-95: provides a stricter evaluation of the model's detection performance. The value of 0.763 indicates that, on average, the model performs well across a wide range of Intersections over Union (IoU) thresholds, meaning it can accurately detect drowsiness symptoms not only when the bounding box localization is precise (at 50% overlap) but also when the localization is less perfect (e.g at 70%, 80%, and higher IoU values). This shows the model's robustness in handling varying degrees of overlap. Furthermore, from Figure 11, it can be observed that the mAP@50-95 steadily increased with each complete pass of the entire training dataset (i.e. 18, 835 images) through the YOLO model. This indicates that as the model (eq. 110was exposed to more data, it improved its ability to detect drowsiness symptoms with increasing precision across multiple IoU thresholds, further validating the effectiveness of the model in real-world scenarios.

$$mA\ P_{50-95} = \frac{1}{N}\sum_{50}^{-955n}(R_{50-95} - R) * P_{50-95} \quad (11)$$

where**:** $R_{50\ -\ 95}$ is the recall between an IoU threshold of 50 – 95% $P_{50\ -\ 95}$ is the precision between an IoU threshold of 50 - 95% n represents the number of classes, that is, the model is detecting.

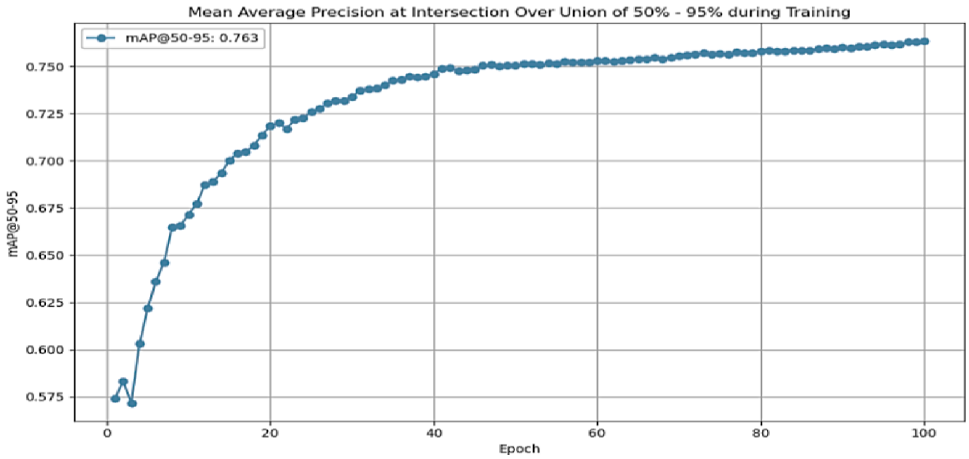

Figure 11: mAP@50-95 During Training

3.6 Confusion Matrix: A confusion matrix is a diagram, as in figure 12, used to assess the performance of a model's classification, particularly in terms of accuracy, precision, recall, and F1 score. With confusion matrix, possible scenarios driver's status while driving could be described. Some of this instances are as follows: True Positive (TP), which is the number of instances correctly predicted as positive, where the model exactly identifies a drowsy driver; False Positive (FP), this is the number of instances incorrectly predicted as positive (e.g the model mistakenly labels a non-drowsy driver as drowsy); True Negative (TN), implying the number of instances correctly predicted as negative (e.g. the model correctly identifies a non-drowsy driver); and the False Negative (FN), this indicate the number of instances wrongly classified as negative (for instance, the model does not recognize a drowsy driver.

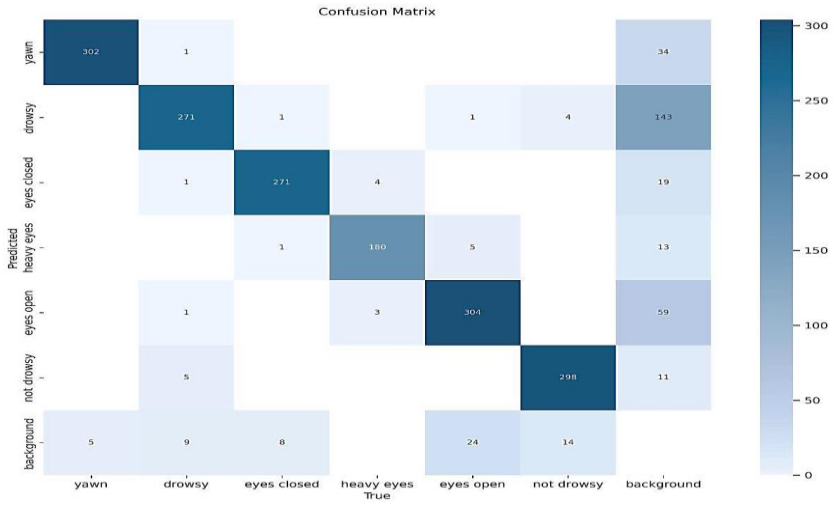

Figure 12: Confusion matrix of the model after training.

From figure 12, the model correctly predicted 302 cases of yawning symptoms but falsely classified 5 background images as yawning. In the drowsy class, the model accurately identified 271 cases. But it misclassified 1 instance of eyes closed, 5 instances of eyes open, and 9 background images as drowsy. For the eyes closed symptom, the model correctly predicted 271 instances but made an error by classifying 1 instance as heavy eyes and 8 instances as background images. In the case of heavy eyes, the model rightly identified 180 cases but misclassified 3 instances as eyes open and 4 as eyes closed. Also, it predicted 304 instances of eyes open correctly but misread 5 instances as heavy eyes, 1 as drowsy, and 24 as background images.

Finally, in the non-drowsy sort, the model correctly predicted 298 instances but wrongly classified 4 cases as drowsy and 14 instances as background. Figure 13 shows the results of samples of the model's performance on some of the image frames.

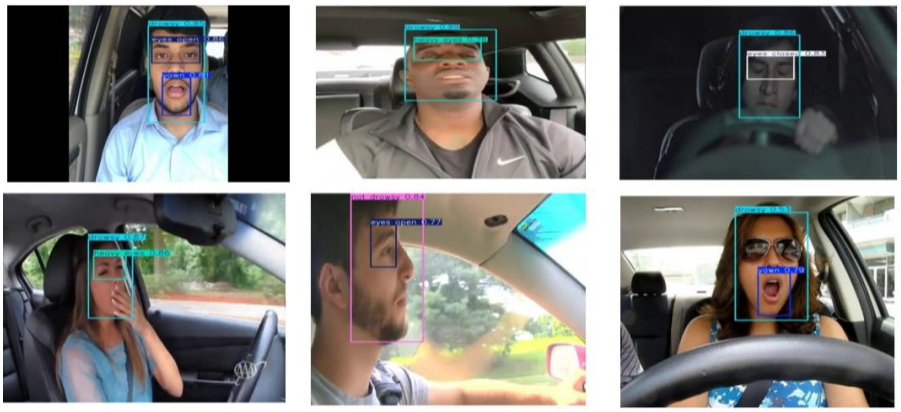

Figure 13: Samples of the model's performance on some of the image frames.

In brief, towards the achievement of the developed object detection machine learning based model capable of precisely finding and extracting facial features and signs of drowsiness such as yawning, heavy eyelids, or closed eyes from a driver's face using images and video footage captured by a camera, figure 12 showed the level. From figure 12, the model correctly predicted 302 cases of yawning symptoms but falsely classified 5 background images as yawning. In the drowsy class, the model precisely identified 271 cases. But, it misclassified 1 instance of eyes closed, 5 instances of eyes open, and 9 background images as drowsy. For the eyes closed symptom, the model correctly predicted 271 instances but made an error by classifying 1 instance as heavy eyes and 8 instances as background images. In the cases of heavy eyes, the model rightly identified 180 cases but misclassified 3 instances as eyes open and 4 as eyes closed. Also, it predicted 304 instances of eyes open correctly but misread 5 instances as heavy eyes, 1 as drowsy, and 24 as background images.

Besides, the enhancement of the accuracy of the model for facial features recognition linked with drowsiness, focusing on reducing false positives and false negatives to ensure reliable performance. The obtained precision value of 0.895 (89.5%) as shown in figure 7 showed that for 89.5% of the times the model detected

drowsiness, the driver was indeed drowsy (i.e. True Positive) leaving an error rate of about 10.5% (False Positives). Similarly, with the recall value of 0.914, the model successfully detected drowsiness 91.4% of the time when the driver was actually drowsy, the model successfully detected and flagged drowsiness. However, this also implies that 8.6% of the time, when the driver was indeed drowsy, the system failed to detect it (i.e. False Negatives).

The overall score of 0.9 as indicated in figure 9 showed that the model is highly effective in detecting drowsy drivers (high recall) and reducing false alarms (high precision) at the threshold of 0.376. Also, the value of mAP@50 (i.e. 0.952) showed that the model is 95.2% accurate in identifying and localizing driver drowsiness and its signs with a good degree of overlap where a symptom is present.

4. CONCLUSION

The novelty of this research lies in addressing the significant limitations of existing state-of-the-art driver drowsiness detection systems. While most systems merely determine whether a driver is drowsy or not, they fail to identify the underlying symptoms that justify this classification. To bridge this gap, this study developed a monitoring method using YOLO v8 to build a custom object detector model tailored specifically for detecting driver drowsiness symptoms such as yawning, closed eyes, and heavy eyelids, and determining whether the driver is drowsy or not.

The model was enhanced to effectively generalize across various scenarios by applying a range of data augmentation techniques, such as horizontal and vertical flips, rotations, and other transformations. This approach ensured the model's robustness to different environmental conditions and viewpoints. This innovative approach not only reduces computational costs and training time but also provides an efficient and reliable solution for the early detection of driver drowsiness. So, it mitigates the risk of severe accidents, reduces substantial economic losses, and enhances public health and safety.

REFERENCES

[1] Albadawi, Y., Takruri M., and Awad M. (2022). "A Review of Recent Developments in Driver Drowsiness," https: www.mdpi.com, pp. 1–41, 2022.

[2] Das S., Pratihar S., Pradhan B., Jhaveri R. H., and Benedetto F., (2024). "IoT-Assisted Automatic Driver Drowsiness Detection through Facial Movement Analysis Using Deep Learning and a U-Net-Based Architecture," Inf., vol. 15, no. 1, pp. 1–30, doi: 10.3390/info15010030.

[3] Dian F. J., Vahidnia R., and Rahmati A. (2020). "Wearables and the Internet of Things (IoT), Applications, Opportunities, and Challenges: A Survey," IEEE Access, vol. 8, pp. 69200–69211, doi: 10.1109/ACCESS.2020.2986329.

[4] Dipu M. T. A., Hossain S. S., Arafat Y., and Rafiq F. B. (2021). "Real-time Driver Drowsiness Detection using Deep Learning," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 7, pp. 844–850, doi: 10.14569/IJACSA.2021.0120794.

[5] Hooda R., Joshi V., and Shah M. (2022). "A comprehensive review of approaches to detect fatigue using machine learning techniques," Chronic Dis. Transl. Med., vol. 8, no. 1, pp. 26–35, doi: 10.1016/j.cdtm.2021.07.002.

[6] Hussain M. (2024). "YOLOv1 to v8: "Unveiling Each Variant - A Comprehensive Review of YOLO," IEEE Access, vol. 12, no. February, pp. 42816–42833, 2024, doi: 10.1109/ACCESS.2024.3378568.

[7] Hussain M., (2023). "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature towards Digital Manufacturing and Industrial Defect Detection," Machines, vol. 11, no. 7, doi: 10.3390/machines11070677.

[8] Irsan M., Hassan R., Hasan M. K. Lam M. C., Hassain W. H. M. W., I A. H., Armed A. S. A. M. S. (2022). "A Novel Prototype for Safe Driving Using Embedded Smart Box System," Sensors, vol. 22, no. 5, p. 1907, 2022, doi: 10.3390/s22051907.

[9] Kikame E. I., Obiadi B. N., Enete I. C., Mbadugha C. L., and Aniegbuna I. A. I. (2024). "Nigerians and their Adaptation to their Deadly Road Conditions: The Case of Fatigue-Related Road Accidents," vol. 10, no. 7, pp. 1–28, 2024, doi: 10.56201/ijemt.v10.no7.2024.pg1.28.

[10] Magán E., Sesmero M. P., Alonso-Weber J. M., and Sanchis A., (2022). "Driver Drowsiness Detection by Applying Deep Learning Techniques to Sequences of Images," Appl. Sci., vol. 12, no. 3, 2022, doi: 10.3390/app12031145.

[11] NHTSA, "Drowsy Driving: Avoid Falling Asleep Behind the Wheel | NHTSA." https://www.nhtsa.gov/risky-driving/drowsy-driving (accessed Feb. 15, 2025).

[12] Phan A. C., Nguyen N. H. Q., Trieu T. N., and Phan T. C., (2021). "An efficient approach for detecting driver drowsiness based on deep learning," Appl. Sci., vol. 11, no. 18, doi: 10.3390/app11188441.

[13] Raghuvaran V., Priya P., and Vidhya V. (2018). "An IoT Based Reliable Driver Drowsiness Detection System for Automobiles," Int. J. Res. Anal. Rev., vol. 5, no. 3, pp. 246–253.

[14] Rani G. S., A. Anusha A., Yamini A., Rao I. S., Chowdary C. G. K. (2024). "Stay Alert: Drowsiness Detection with IoT Technology," Int. J. Mod. Trends Sci. Technol., vol. 10, no. 02, pp. 46–52, doi: 10.46501/ijmtst1002007.

[15] Rudrusamy B., Teoh H. C., Pang J. Y., Lee T. H., and Chai S. C. (2023). "IoT-Based Vehicle Monitoring and Driver Assistance System Framework for Safety and Smart Fleet Management," Int. J. Integr. Eng., vol. 15, no. 1, pp. 391–403, doi: 10.30880/ijie.2023.15.01.035.

[16] Sahayadhas A., Sundaraj K., and Murugappan M. (2012). "Detecting driver drowsiness based on sensors: A review," Sensors (Switzerland), vol. 12, no. 12, pp. 16937–16953, doi: 10.3390/s121216937.

[17] Shaik M. E. (2023). "A systematic review on detection and prediction of driver drowsiness," Transp. Res. Interdiscip. Perspect., vol. 21, no. January, p. 100864, 2023, doi: 10.1016/j.trip.2023.100864.

[18] Shariffuddin N. A., and Ling Y. T. (2023). "IoT-based Eyes Closure Alert System for Drowsy Driver," Trends Undergrad. Res., vol. 6, no. 1, pp. c1-6, doi: 10.33736/tur.2812.2023.

[19] Sinha A., Aneesh R. P., and Gopal S. K. (2021). "Drowsiness Detection System Using Deep Learning," Proc. 2021 IEEE 7th Int. Conf. Bio Signals, Images Instrumentation, ICBSII 2021, pp. 1–6, 2021, doi: 10.1109/ICBSII51839.2021.9445132.

[20] Talaat F. M., and ZainEldin H. (2023). "An improved fire detection approach based on YOLO-v8 for smart cities," Neural Comput. Appl., vol. 35, no. 28, pp. 20939–20954, 2023, doi: 10.1007/s00521-023-08809-1.

[21] Tanveer M. A., Khan M. J., Qureshi M. J., Naseer N., and Hong K. S., (2019). "Enhanced drowsiness detection using deep learning: An fNIRS Study," IEEE Access, vol. 7, pp. 137920–137929, doi: 10.1109/ACCESS.2019.2942838.

[22] Terven J., Córdova-Esparza D. M., and Romero-González J. A. (2023). "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," Mach. Learn. Knowl. Extr., vol. 5, no. 4, pp. 1680–1716, doi: 10.3390/make5040083.

[23] Yunidar Y., Melinda M., Khairani K., Irhamsyah M., and Basir N. (2023). "IoT-based Heart Signal Processing System for Driver Drowsiness Detection," Green Intell. Syst. Appl., vol. 3, no. 2, pp. 98–110, doi: 10.53623/gisa.v3i2.323.