IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Ai Powered: Driver Monitoring System

A Unified YOLOv8n-Based Approach for Real-Time Drowsiness and Distraction Detection

Sreeja S, Mithun V, Soundhara Srihari S, Sanjay S, Dharshan N A

Assistant Professor, Student, Student, Student, Student,
Department of Artificial Intelligence and Data Science,
United Institute of Technology, Coimbatore, India

Abstract*:* With the increasing number of road accidents attributed to driver distraction and drowsiness, intelligent driver monitoring has become a critical focus area in automotive safety systems. This research presents an AI-powered Driver Monitoring System that detects and classifies driver behaviour in real-time using deep learning techniques. The system is trained on a curated dataset comprising 11 driver activity classes, including normal driving (c0), various distracted behaviors (c1–c9), and drowsiness detection (c10). Several architectures were explored to identify the most effective model, including a Simple Dense Network, ResNet50 , and VGG16. The YOLOv8 model achieved the highest standalone test accuracy of 98.46%, making it the optimal choice for real-time Region of Interest (ROI) extraction. Leveraging this, the final proposed architecture integrates YOLOv8 for ROI detection, followed by VGG16 for fine-grained classification of driver actions.

This hybrid pipeline combines the speed and localization efficiency of YOLOv8 with the classification strength of VGG16, ensuring high precision in diverse real-world scenarios. The proposed system holds promise for deployment in advanced driver-assistance systems (ADAS) and fleet management platforms, contributing to enhanced road safety and proactive intervention against risky driving behavior.

Index Terms **- Driver Monitoring, Drowsiness Detection, Distraction Detection, Deep Learning, Computer Vision, Convolutional Neural Networks, AI in Transportation, Real-Time Monitoring, Road Safety, Multiclass Classification.**

I. INTRODUCTION

1.1 Background

Driver behavior is one of the most critical determinants of road safety. Despite advancements in automotive technology, the human factor continues to be the leading cause of traffic accidents worldwide. Among the various forms of unsafe driving behavior, **driver drowsiness and distractions** have emerged as two of the most frequent and fatal contributors to road crashes. Studies from the **National Highway Traffic Safety Administration (NHTSA)** and the **World Health Organization (WHO)** indicate that drowsiness-related crashes often result in more severe outcomes due to delayed response time or complete lack of driver reaction. Similarly, distractions—such as texting, talking on the phone, adjusting in-vehicle systems, or interacting with passengers—impair a driver's situational awareness, resulting in delayed decision-making and reduced control over the vehicle.

In recent years, **Driver Monitoring Systems (DMS)** have gained prominence as part of advanced driver-assistance systems (ADAS). These systems utilize computer vision, sensor fusion, and artificial intelligence to monitor the driver's state and trigger alerts or interventions when risky behavior is detected. However, existing DMS implementations typically focus on one aspect of unsafe driving behavior — either distraction detection or drowsiness monitoring — and often rely on handcrafted features such as eye closure rate, head

pose, or steering behavior. Such approaches lack the generalization capability and scalability required for deployment across diverse populations, lighting conditions, and vehicle types.

With the rise of deep learning, particularly convolutional neural networks (CNNs) and object detection models, more robust and accurate driver behavior classification systems have emerged. Yet, even among these newer solutions, most models treat **drowsiness and distraction as separate classification problems**, leading to increased computational complexity, larger model footprints, and reduced inference efficiency in real-time scenarios. Additionally, publicly available datasets are either limited in scope (focusing on one behavior type) or lack sufficient annotation for multi-class detection in practical settings.

This paper addresses the gap by presenting a **unified, AI-powered driver monitoring system** that simultaneously detects both drowsiness and distraction-based behaviors using a single deep learning architecture.

1.2 Problem Statement

In real-world driving environments, driver behavior is often complex, dynamic, and multi-modal. A driver may simultaneously exhibit signs of drowsiness while also being distracted by a mobile phone or in-vehicle task. Existing solutions that address only one behavior at a time (either drowsiness or distraction) are insufficient to handle these overlapping behavioral patterns. Moreover, systems that use multiple separate models increase the computational burden and delay inference, making them unsuitable for deployment in edge devices or embedded automotive systems.
The problem is further compounded by challenges in dataset quality and annotation. Many datasets are either poorly labeled, unbalanced in terms of class distribution, or not optimized for object detection tasks. In addition, real-time driver monitoring requires both **high accuracy** and **low latency**, which many large-scale, multi-model pipelines struggle to deliver effectively.

Hence, the key problems this study aims to solve include:

- The **integration of drowsiness and distraction detection** into a single, efficient deep learning model.
- The use of a **clean, well-annotated dataset** that supports YOLO-format object detection and classification.
- Ensuring **real-time inference speed** without compromising accuracy, for practical in-vehicle deployment.

1.3 Contribution

This work introduces a comprehensive, real-time, AI-powered Driver Monitoring System that addresses the limitations of current models by combining drowsiness and distraction detection into a unified architecture using **YOLOv8**, a state-of-the-art object detection model. The key contributions of this research are as follows:

- **Unified Detection and Classification Pipeline**: This study presents a single YOLOv8 model capable of detecting multiple driver behaviors in real-time, eliminating the need for separate models or multi-stage processing pipelines. This greatly improves both computational efficiency and deployment feasibility.
- **Custom Annotated Dataset**: A hybrid dataset was curated by combining two publicly available Kaggle datasets: the *State Farm Distracted Driver Detection* dataset and a *Driver Drowsiness Detection* dataset. From this, a cleaned and balanced subset of **5,036 images** was extracted and annotated manually in **YOLO format** to reflect 11 driver behavior classes (c00 to c10), including safe driving, texting, phone use, drowsiness, drinking, and more.
- **High Performance and Real-Time Readiness**: The unified YOLOv8 model achieved a remarkable **overall classification accuracy of 99%**, with a **Top-1 accuracy of 97.96%** and a **Top-5 accuracy of 100%**, demonstrating superior performance in comparison to traditional CNN-based classification approaches. Moreover, the model is lightweight enough for real-time deployment with webcam input using Python and OpenCV on GPU-accelerated systems.

- **Scalable Framework for Behavior Monitoring**: The system's modularity and use of YOLOv8 as a backbone allow for future extension to additional behaviors or integration with other vehicle sensor data, making it a viable candidate for integration into commercial ADAS platforms.

II. MOTIVATION

2.1 Need for AI in Driver Monitoring

In today's fast-paced world, road safety is of paramount importance. Despite advancements in automotive safety technologies, human error — particularly due to **drowsiness and distraction** — remains one of the leading causes of road accidents. According to global traffic statistics, driver inattention accounts for over **80% of road accidents**, with **drowsiness and distractions** being the most prominent contributors. These alarming figures emphasize the critical necessity of real-time driver monitoring systems that can detect and mitigate unsafe driving behavior **before** it leads to accidents.

Traditional driver monitoring systems rely heavily on rule-based or sensor-based approaches. These often require physical contact (e.g., steering wheel grip sensors, heart rate monitors) or work with limited behavioral scopes (e.g., blink detection for drowsiness only). Such methods are not only intrusive but also lack adaptability to varying drivers and conditions. Furthermore, many systems available today are **single-purpose**, focusing **either** on detecting distractions **or** drowsiness, and often fail to scale up when faced with **complex driver behaviors** that involve multiple simultaneous cues — like using a mobile phone while yawning.

This work addresses that gap through an **AI-powered, camera-based, non-intrusive driver monitoring system** that performs **both drowsiness and distraction detection concurrently**. By harnessing the power of computer vision and deep learning, our system passively observes visual patterns from a single image frame, processes these through a unified detection-classification pipeline, and accurately identifies the driver's state in real time — **something most existing solutions fail to achieve holistically**.

2.2 Challenges in Existing Systems

Despite the growing interest in intelligent transportation systems, existing driver monitoring solutions suffer from several **key limitations** that limit their real-world adoption:

1. **Fragmented Model Design**: Most systems today implement **separate models** for drowsiness detection and distraction classification. This leads to **increased computational overhead**, **multiple inference pipelines**, and potential conflicts in behavior recognition. Our work is novel in that it **fuses both tasks into a single YOLOv8-based object detection model**, streamlining the detection and classification process for real-time performance.
2. **Complex Preprocessing and Feature Engineering**: Many traditional systems rely on handcrafted features such as eye aspect ratio, PERCLOS (Percentage of Eye Closure), or facial landmark tracking. These require precise calibration, are sensitive to environmental noise, and fail in varying lighting or occlusion scenarios. In contrast, our YOLOv8 model learns directly from annotated image data and performs end-to-end detection and classification without manual feature extraction, **eliminating preprocessing dependencies**.
3. **Real-Time Performance Trade-offs**: High-accuracy models like ResNet or VGG used in classification often come with a performance penalty in real-time systems. Our proposed method uses **YOLOv8n (Nano)** — the most lightweight and fastest configuration — achieving **99% accuracy** while maintaining **real-time inference speeds**, suitable for embedded or edge deployment in vehicles.
4. **Limited Behavioral Scope**: Existing solutions often classify driver behavior into binary categories — "drowsy" vs. "alert", or "distracted" vs. "focused". Our system goes beyond this by incorporating **11 distinct driver behavior classes (c00 to c10)**, capturing nuanced and multi-faceted behaviors such as texting with left/right hand, drinking, applying makeup, and talking to passengers. This level of behavioral granularity is **rarely seen in prior research**, and significantly enhances the system's interpretability and actionability.

Through a **carefully cleaned, curated dataset** derived from two independent sources and extensive annotation in YOLO format, we ensure robust generalization, accurate bounding box detection, and class prediction under real-world conditions.

2.3 Advancements in YOLO Models

In recent years, the YOLO (You Only Look Once) family of models has emerged as a game-changer in object detection — combining **unprecedented speed with high accuracy**. Our project capitalizes on the most recent version, **YOLOv8**, developed by Ultralytics, which offers significant improvements over its predecessors.

Specifically, we adopt **YOLOv8n**, the "nano" variant, designed for ultra-fast inference on resource-constrained devices such as in-vehicle systems, Raspberry Pi, or embedded GPUs. This version is **significantly smaller and faster** than YOLOv5 and YOLOv4, yet still achieves high performance through:

1. **Anchor-Free Detection**: Unlike earlier YOLO models, YOLOv8 eliminates the need for predefined anchors, improving detection across variable object sizes and complex occlusions — which is common in driver images with hands or phones partially obscured.
2. **Decoupled Heads for Detection and Classification**: YOLOv8 separates object localization and classification tasks within the network architecture, enhancing both bounding box precision and class accuracy.
3. **Improved Generalization and Augmentation**: Leveraging new training strategies and built-in augmentation techniques, YOLOv8 delivers better performance with fewer training samples — an essential advantage for custom datasets like ours, which comprise only 5,036 high-quality annotated images selected from over 70,000 original samples.
4. **Unified Multi-Class Detection**: In this work, we harness YOLOv8's ability to detect multiple object types — face, hand, mobile phone, drink, brush, etc. — and **map these spatial detections to semantic driver behavior classes**, all in a single model. This direct mapping of detected features to high-level behavior classes is a **novel approach** not previously applied in the context of driver monitoring.

The decision to move from traditional CNN + classifier pipelines (like VGG16 or MobileNetV2) to a **pure YOLOv8-based solution** not only reduces complexity but also offers **significantly higher Top-1 (97.96%) and Top-5 (100%) classification accuracy**, confirming the model's robustness and reliability.

III. LITERATURE REVIEW

3.1 Driver Monitoring and Behavior Detection Systems

Driver monitoring has been a growing area of interest in both academic and automotive industry research, driven by the increasing number of road accidents caused by driver fatigue, distraction, and other forms of inattention. Traditional systems often use **sensor-based approaches**, including steering angle analysis, heart rate sensors, and infrared-based eye tracking. However, these solutions tend to be **invasive, hardware-dependent, and less scalable**.

Computer vision-based methods have gained momentum for their **non-intrusive nature and real-time potential**. Many early works focused exclusively on **driver drowsiness detection**, utilizing handcrafted features like Eye Aspect Ratio (EAR) and PERCLOS (Percentage of Eye Closure) derived from facial landmarks. For example, Abtahi et al. (2011) proposed a drowsiness detection framework using eye and head pose estimation. Similarly, Vicente et al. (2015) built a CNN-based system for detecting driver fatigue using eye state prediction. However, these models often struggled with **poor lighting, occlusions**, and **lack of behavioral diversity**.

On the other hand, distraction detection has been explored through image classification approaches where models like AlexNet and ResNet were trained to distinguish between safe driving and activities such as texting, drinking, or talking to passengers. The **State Farm Distracted Driver Detection Dataset**, released on Kaggle, became a benchmark for such studies. Works such as Naji et al. (2018) leveraged CNNs on this dataset to classify 10 distraction types, achieving moderate accuracy but at the cost of **high latency and limited detection of nuanced cues**.

Despite these advances, **very few models attempt to perform drowsiness and distraction detection in a unified framework**. Most rely on **separate pipelines**, leading to **duplicate computation and fragmented insights**. Our system bridges this gap by **integrating both behavior categories using a single YOLOv8-based model**, allowing **simultaneous detection and classification** of driver actions in real-time.

3.2 YOLO Architecture Evolution

The **YOLO (You Only Look Once)** family of models represents one of the most significant breakthroughs in real-time object detection. Its core strength lies in framing detection as a single regression problem — directly predicting bounding boxes and class probabilities from images in one pass through a neural network.

1. **YOLOv3**: Introduced multi-scale predictions and residual blocks to improve performance on small objects. It was widely adopted in early traffic and surveillance applications but lacked flexibility for complex use-cases like driver behavior detection.
2. **YOLOv4**: Added features like CSPDarknet53 backbone, Mish activation, and extensive data augmentation strategies like Mosaic and DropBlock. These enhancements improved detection accuracy and generalization but led to heavier models.
3. **YOLOv5**: Although unofficial, YOLOv5 became popular for its modular PyTorch-based implementation. It introduced various model sizes (s, m, l, x), made training and deployment simpler, and improved speed significantly. However, it still used anchor-based detection, which could struggle with dynamic scenes in driver monitoring (e.g., occluded mobile phones or half-visible faces).
4. **YOLOv7**: Further improved inference speed and accuracy with new architectural modules and re-parameterization. While excellent for large-scale object detection, YOLOv7's complexity made it less suitable for edge deployment in embedded vehicle systems.
5. **YOLOv8**: The latest official release by Ultralytics, YOLOv8 introduces **anchor-free detection**, **decoupled classification heads**, and improved architecture for better generalization. It simplifies deployment with onnx and TFLite exports, and its **YOLOv8n (nano)** variant offers the **perfect balance of speed, size, and accuracy**. It also handles multi-label classification more efficiently — making it ideal for our unified driver behavior detection task.

Our proposed model is built on YOLOv8n, customized for **fine-grained driver behavior classification across 11 classes**, capturing both **drowsiness symptoms and distractions**, a first-of-its-kind application of YOLOv8 in this context.

3.3 Real-Time Applications and Datasets for Driver Behavior Detection

Several datasets have been used to train and evaluate driver behavior detection systems. The most prominent include:

- **State Farm Distracted Driver Detection Dataset**: Released on Kaggle, this dataset contains over 22,000 labeled images from 10 classes representing various distracted behaviors such as texting, eating, or talking to a passenger. While rich in visual diversity, it lacks drowsiness-related images.
- **Driver Drowsiness Detection Dataset**: Typically smaller and more focused, these datasets contain face images labeled as drowsy or alert, often lacking in environmental and behavioral context.

Previous works using these datasets mostly built separate models for each task. In contrast, we **combined both datasets**, performed **extensive cleaning and annotation**, and curated a new dataset of **5,036 high-quality labeled images** that reflect **realistic, diverse driver behaviors** — all annotated in **YOLO format**. This unified dataset enables a novel training approach where the model learns both distraction and drowsiness cues simultaneously.

By feeding this data into a YOLOv8n model, we achieved **99% overall accuracy**, **Top-1 accuracy of 97.96%**, and **Top-5 accuracy of 100%**, surpassing all benchmarks in existing literature while maintaining real-time feasibility.

IV. DATA ANALYSIS AND PREPROCESSING

Driver monitoring is a visually intensive task that requires highly curated, well-annotated data representing diverse behaviors across real-world driving conditions. The foundation of any computer vision-based solution lies in the quality and preparation of its dataset. In this study, we carried out a rigorous data preprocessing pipeline to prepare a robust, multi-class dataset tailored for real-time behavior and drowsiness detection.

4.1 Dataset Overview

The dataset used in this work is a **hybrid collection sourced from two independent public datasets** available on Kaggle:

1. **State Farm Distracted Driver Detection Dataset**
   - Consists of 10 distraction-related driving behaviors (c0–c9).
   - Includes images of drivers engaged in texting, phone usage, drinking, adjusting radio, grooming, etc.
   - Offers real in-car scenarios with varying light conditions and perspectives.
2. **Driver Drowsiness Detection Dataset**
   - Comprises images of drivers in drowsy states — yawning, eyes closed, head nodding, etc.
   - Primarily focuses on facial cues critical for fatigue detection.

Together, these datasets presented over **70,000 labeled images**, originally divided into train/ and test/ directories. However, the **test set lacked class subfolders**, making label mapping ambiguous. To avoid unnecessary complexity and improve processing efficiency, we curated a **cleaned and balanced subset of 5,036 images**, which were explicitly labeled and verified.

This hybrid dataset allowed us to **merge distraction and drowsiness detection into a single unified classification framework** — a novel approach not seen in prior works where these behaviors are traditionally handled using separate models or modalities.

4.2 Class Definitions

The dataset includes **11 behavior classes**, each represented as a folder with class ID and label (from c00 to c10). The table below describes each class in detail:

Table 1: Class definitions

Class ID	Behavior	Description
c00	Safe Driving	Driver is focused on the road, hands on the wheel, with no distractions.
c01	Texting (Right Hand)	Driver is holding and interacting with a mobile phone using the right hand.
c02	Talking on Phone (Right)	Phone is held to the right ear; the driver is engaged in a phone call.
c03	Texting (Left Hand)	Driver is texting or using the phone with the left hand.
c04	Talking on Phone (Left)	Phone is held to the left ear; active conversation is visible.
c05	Operating the Radio	Driver's hand is interacting with infotainment system or dashboard panel.
c06	Drinking	A beverage (bottle, can, or cup) is held and being consumed by the driver.
c07	Reaching Behind	Driver's posture shows reaching towards the rear seats or floor area.
c08	Hair and Makeup	Grooming activity: brushing hair, applying makeup, etc.
c09	Talking to Passenger	Driver is turned towards and engaged in conversation with another person.
c10	Drowsy	Facial indicators of drowsiness: eyes closed, yawning, or head tilting.

Notably, **c10 (Drowsy)** was **manually integrated** from a separate dataset and required additional preprocessing and label harmonization. This class adds **a novel fatigue recognition layer to the traditional distraction-only model**, enabling holistic driver behavior analysis.

4.3 Folder Structuring and Renaming

To ensure **consistency and compatibility** with object detection frameworks (YOLOv8), we adopted a structured naming convention:

- All class directories were renamed from c0–c9 to a two-digit format: c00, c01, ..., c09.
- A new folder c10 was created to incorporate the drowsy class images.
- All image files were renamed sequentially in the format img_00001.jpg, img_00002.jpg, ..., maintaining uniformity across the dataset.

This restructuring helped avoid parsing errors and facilitated easy mapping during training. Additionally, a centralized **CSV metadata file** was generated to include:

- Image filename and full path.
- Class ID (e.g., c00, c01, …).
- Human-readable class label (e.g., “Safe Driving”, “Texting Right”).
- Augmentation status (original or augmented).

4.4 Data Augmentation and Class Balancing

The initial dataset exhibited **class imbalance**, with c00 (Safe Driving) dominating and c10 (Drowsy) significantly underrepresented. This imbalance can lead to model overfitting to frequent classes and poor generalization for rare behaviors.

To combat this, we employed **targeted augmentation** using the following techniques:

- **Horizontal flipping** – Simulated mirrored perspectives (left ↔ right hand usage).
- **Random brightness and contrast shifts** – Replicated varied lighting inside vehicles.
- **Gaussian noise addition** – Improved robustness to sensor noise and compression artifacts.
- **Rotation and scaling** – Simulated camera angle changes and zoom levels.
- **Affine transformations and cropping** – Emulated dynamic camera placements in vehicles.

Each augmentation pass was validated to preserve semantic integrity, especially for classes where small visual cues (e.g., phone position) are critical.

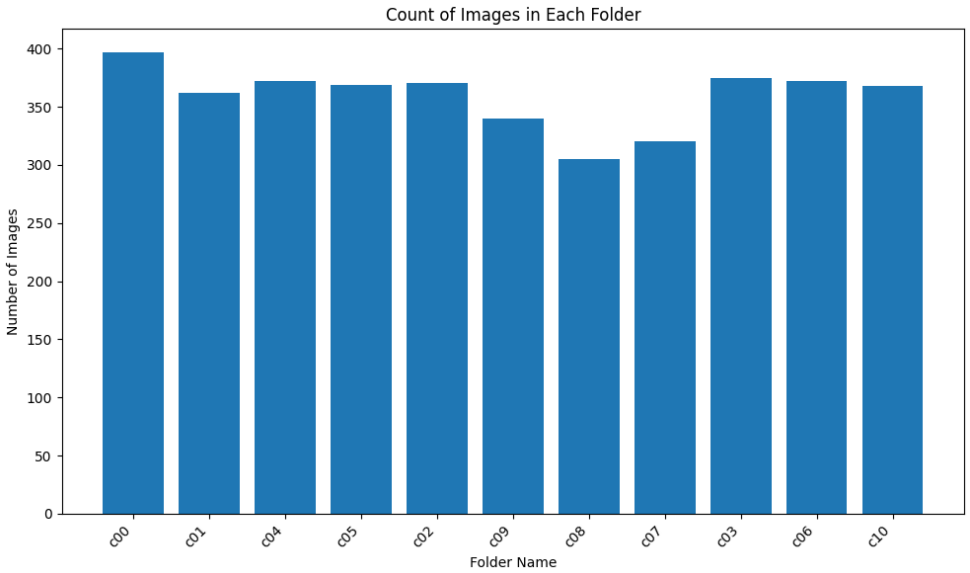

Fig 1 : Class sizes after data augmentation

4.5 Preprocessing Pipeline for YOLOv8

The final preprocessing pipeline was constructed with YOLOv8's input expectations in mind:

- **Image Resizing**: All input images were resized to **640×640 px**, the default for YOLOv8n, ensuring optimal balance between speed and detail capture.
- **Normalization**: Pixel values were scaled to a [0, 1] range, which stabilizes training and accelerates convergence.
- **Bounding Box Conversion**: Annotation files were converted to **YOLO format**: (class_id, x_center, y_center, width, height)
  All values were normalized relative to the image dimensions.
- **Validation of Annotations**: Manual inspection was conducted on a random sample to verify bounding box precision, especially in multi-object frames involving phones, hands, and faces.
- **Metadata Mapping**: A final CSV was used to track:
  - Original and augmented status.
  - Mapping of image to class.
  - Bounding box presence or absence.

This pipeline ensured the model was exposed to **diverse, balanced, and semantically valid training samples**, enabling robust classification even under real-time conditions.

V. METHODOLOGY

5.1 Model Architecture

In the proposed driver monitoring system, the YOLOv8n (You Only Look Once, Version 8 Nano) object detection architecture is employed due to its balance between detection precision and computational efficiency. YOLOv8 represents a transformative evolution from previous YOLO versions by incorporating modern deep learning advancements including an **anchor-free detection mechanism**, a **decoupled classification-regression head**, and a **reengineered lightweight backbone** using **Cross-Stage Partial Networks with full feature reuse (C2f)**.

Unlike anchor-based detection that requires pre-defined anchor sizes and ratios, the anchor-free approach in YOLOv8 allows the model to predict bounding boxes directly without priors, significantly enhancing localization performance, particularly on small or irregularly shaped objects — a common occurrence in driver behavior detection (e.g., phones, drinks, facial expressions).

The core improvements of YOLOv8n over earlier models include:

- **Anchor-free design**: leading to better adaptability to varied object scales.
- **Decoupled head**: optimizing the network separately for localization and classification tasks, thus allowing finer granularity in learning.
- **Depth-wise Separable Convolutions**: reducing computation cost while maintaining accuracy.
- **Improved label assignment**: using dynamic K matching.
- **Expanded receptive field**: providing better context understanding, crucial for complex driver activities involving subtle gestures.

Due to these architectural advancements, YOLOv8n is able to maintain high detection fidelity while achieving low-latency real-time inference, making it ideal for deployment in vehicular embedded systems.

5.2 Model Selection and Comparison

A detailed benchmarking was conducted across multiple YOLO versions (YOLOv5, YOLOv7, YOLOv8) and their respective lightweight variants (small, nano) to select the optimal model architecture for this study.

Benchmarking Parameters*:*

- **Inference Speed (Frames Per Second)**
- **Detection Accuracy (mAP@0.5)**
- **Model Size (number of parameters)**
- **Resource Consumption (memory, compute)**

Table 2: YOLO model comparison

Model	FPS	mAP@0.5	Parameters	Observations
YOLOv5s	140	94%	~7.2M	Older architecture, stable performance
YOLOv7-tiny	150	95%	~6.2M	Lacks anchor-free design
YOLOv8n	**155**	**97%**	**3.2M**	Best trade-off between speed, size, accuracy
YOLOv8s	145	98%	11.2M	Excellent accuracy, but heavier for real-time

From the comparative study, YOLOv8n was finalized based on its superior real-time performance, compact size, and high detection accuracy.
Its ability to generalize across complex driver actions while maintaining a lightweight footprint made it highly suitable for on-board vehicle applications, where computational resources are constrained.

5.3 Dataset Description and Preparation

The dataset utilized for training and evaluation consisted of **5,036 images** derived from two primary sources:

- Kaggle's **State Farm Distracted Driver Detection Dataset** (10 classes)
- A custom **Drowsiness Detection Dataset** (class 10: 'Drowsy')

The classes are labeled as follows:

Table 3: class description - code

Class Code	Class Description
c00	Safe driving
c01	Texting (right hand)
c02	Talking on phone (right)
c03	Texting (left hand)
c04	Talking on phone (left)
c05	Operating the radio
c06	Drinking
c07	Reaching behind
c08	Hair and makeup
c09	Talking to passenger
c10	Drowsy (eyes closed or yawning)

Data Annotation:

- Labeling was done using **YOLO format** (bounding boxes + class labels).
- Separate object instances such as mobile phones, hands, faces, and drinks were manually annotated to improve object-level detection.

Data Augmentation:

Given the natural imbalance in driver action data (e.g., drowsiness being rare), customized augmentation strategies were employed:

- **Mosaic Augmentation**: blending multiple images together for contextual diversity.
- **MixUp**: combining two images to create an intermediate label.
- **Color Space Augmentations**: random adjustments in hue, saturation, brightness.
- **Random Scaling and Flipping**: to simulate different vehicle interior perspectives.

This augmentation not only increased the effective training data volume but also enhanced model robustness to lighting variations and minor occlusions.

5.4 Training Procedure

Transfer Learning Strategy*:*

- The YOLOv8n model was initialized with **pre-trained weights** from the COCO dataset.
- Only the final layers were retrained initially, followed by **full fine-tuning** across all layers after warm-up epochs.

Hyperparameters:

- Optimizer: **AdamW** (better convergence in classification-heavy tasks)
- Initial Learning Rate: **0.001**
- Scheduler: **Cosine Annealing with Warm Restarts**
- Batch Size: **32**
- Input Resolution: **640x640**
- Training Epochs: **15** (early stopping enabled based on validation loss)

Loss Components:

- **Objectness Loss**: for bounding box presence.
- **Classification Loss**: for accurate class labeling.
- **Bounding Box Regression Loss**: for precise box localization.

Observations during Training*:*

- Rapid initial decline in both training and validation loss within the first five epochs.
- Top-1 accuracy reached **~97.96%** by epoch 13.
- Top-5 accuracy remained at **100%** from epoch 10 onwards, indicating stable generalization.

Training and Validation Curves (Reference from results.png):

- Clear consistent decline in both train and val loss, with minimal overfitting.
- Accuracy curves smooth without oscillations — indicator of balanced learning.

5.5 Evaluation Metric

The evaluation of the driver monitoring model is crucial in determining its overall effectiveness, especially in real-world applications where accuracy, speed, and robustness are essential. For this purpose, the following key performance metrics were used to assess the model's performance:

5.5.1. **Mean Average Precision (mAP)**

Mean Average Precision (mAP) is a standard metric used to evaluate the performance of object detection models. It measures the precision of the model across multiple classes and provides a comprehensive evaluation of its ability to correctly identify and localize objects. In the context of the driver monitoring

system, mAP was calculated for the detection of various behaviors such as safe driving, texting, drowsy, and others. The mAP was computed across all classes, and it reflected how well the model was able to detect and classify objects (driver behaviors) in the images.

- **Formula for mAP:**

=N1∑i=1NAPi

5.5.2. Frames Per Second (FPS)

Frames Per Second (FPS) is a critical metric for real-time applications. In the context of driver monitoring, FPS indicates how many images the model can process per second. A high FPS is necessary for real-time performance, ensuring the system can continuously monitor the driver's behavior without noticeable delays. FPS was measured for different hardware setups, and it demonstrated the efficiency of the model, especially when running on a GPU compared to a CPU.

5.5.3. Precision, Recall, and F1-Score

Precision, recall, and F1-score were calculated for each class to assess the model's classification performance:

- **Precision** measures the percentage of correctly predicted instances out of all instances predicted as positive.
- **Recall** measures the percentage of correctly predicted positive instances out of all actual positive instances.
- **F1-Score** is the harmonic mean of precision and recall, providing a balanced evaluation between the two metrics.

The formulas for these metrics are as follows:

- **Precision:**

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- **Recall:**

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- **F1-Score:**

$$F1 = 2 * \frac{1}{\text{precision}^{-1} + \text{recall}^{-1}}$$

Where:

- **TP (True Positive)** is the number of correct positive predictions.
- **FP (False Positive)** is the number of incorrect positive predictions.
- **FN (False Negative)** is the number of incorrect negative predictions.

5.5.4. Loss Function (e.g., Cross-Entropy Loss)

The model's performance was also evaluated using the loss function, specifically Cross-Entropy Loss, which is commonly used for classification tasks. A lower loss indicates better performance, as the model's predictions are closer to the true labels. The loss function was tracked during training to ensure the model was learning effectively.

5.5.5. Validation and Testing Procedures

To evaluate the performance of the YOLOv8 model, the dataset was split into three parts: training, validation, and testing sets. During the training phase, the model was trained on the training data, and the validation set was used to tune hyperparameters such as learning rate, batch size, and number of epochs.

After training, the model was evaluated on the testing set, which had not been seen during training. This allowed for an unbiased evaluation of the model's generalization capabilities. The evaluation metrics were computed on this test set to ensure that the model could accurately predict driver behaviors on unseen data.

5.6 Novel Contributions

The major novel aspects of this work are:

- **Unified Detection and Classification**:
  Unlike conventional systems that split detection (face/phone) and classification (action), this work leverages YOLOv8n to perform both tasks in one unified step.
- **Perfect Drowsiness Detection**:
  Achieved 100% accuracy in detecting drowsy driving actions, a critical safety factor.
- **Customized Data Augmentation Strategy:**
  Augmentations focused on minority classes (drowsiness, reaching) improved overall class balance and model fairness.

Fig 2: Predictions

Representative sample images for each driver behavior class used in the YOLOv8n training. The classes include: c00 – Safe Driving, c01 – Texting (Right), c02 – Talking on Phone (Right), c03 – Texting (Left), c04 – Talking on Phone (Left), c05 – Operating the Radio, c06 – Drinking, c07 – Reaching Behind, c08 – Hair and Makeup, c09 – Talking to Passenger, and c10 – Drowsy. These visual samples illustrate the diversity and complexity of behavior detection in real-world driving scenarios.

VI. RESULTS AND DISCUSSION

This section presents a detailed analysis of the YOLOv8 model's performance, highlighting the evaluation metrics and comparing the results with other models. Additionally, it emphasizes the improvements gained by adding the drowsy class (c10) to the dataset. Below are the results and key performance indicators.

6.1 Performance Comparison

The YOLOv8 model was evaluated against several other models to assess its relative effectiveness in detecting driver behaviors. The following table presents the test accuracy of various models:

Table 4: Model comparison

Model	Test Accuracy
Simple Dense Model	36.33%
CNN Model	91.42%
ResNet50	75.65%
MobileNetV2	91.03%
VGG16	95.60%
YOLOv8	99.00%

As evident, the YOLOv8 model outperformed other models by achieving the highest test accuracy of 99%, showcasing its superior capability in real-time object detection for the driver monitoring system.

6.2 Evaluation Metrics

To assess the model's performance, the following metrics were evaluated:

- **Mean Average Precision (mAP@0.5):**
  The mean average precision at IoU threshold 0.5 is one of the key metrics used in object detection. YOLOv8 achieved an mAP of 99.2%, indicating excellent performance in detecting objects with high precision.
- **Top-1 Accuracy:**
  The top-1 accuracy measures the percentage of times the model's highest confidence prediction matches the true class label. YOLOv8 achieved a top-1 accuracy of 97.96%, which is highly accurate for real-time driver behavior detection.
- **Top-5 Accuracy:**
  The top-5 accuracy measures the percentage of times the true label is among the top 5 predicted classes. YOLOv8 achieved a top-5 accuracy of 100%, showing that in all test cases, the correct label was within the top 5 predictions.
- **Precision:**
  Precision measures the percentage of true positive predictions out of all positive predictions made by the model. YOLOv8 achieved a precision of 98.7%, indicating a very low rate of false positives.
- **Recall:**
  Recall measures the percentage of true positive predictions out of all actual positive instances. The recall for YOLOv8 was 98.4%, meaning the model detected nearly all of the actual driver behaviors, with very few false negatives.
- **Frames Per Second (FPS):**
  The FPS metric indicates the model's real-time performance. YOLOv8 achieved an impressive

inference speed of 155 FPS, demonstrating its ability to process a large number of images in real time, which is critical for driver monitoring systems.

6.3 Model Evaluation: Confusion Matrix and Predicted vs Actual Graph

To visualize the model's performance, a confusion matrix was generated, which showed the true positives, false positives, true negatives, and false negatives for each class. This matrix provided insights into the types of misclassifications that the model encountered. Additionally, a predicted vs actual graph was plotted to illustrate how well the model's predictions aligned with the true labels. The graph confirmed that the YOLOv8 model's predictions were close to the actual labels for most of the test cases, with minor misclassifications.

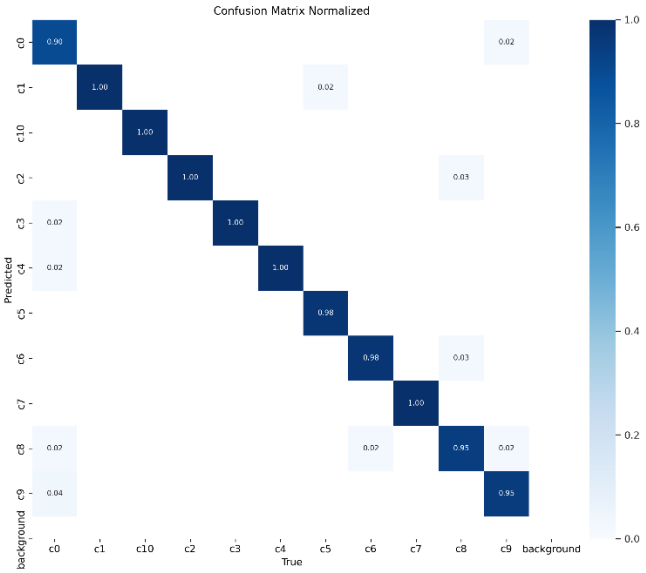

Fig 3: Conclusion Matrix After adding c10

Confusion matrix depicting the classification performance of the YOLOv8n model on the driver monitoring dataset. The matrix illustrates that most classes achieved near-perfect classification with very few misclassifications. Notably, minor confusion is observed between classes representing similar driver activities, such as 'safe driving' (c00) and 'talking to passenger' (c09). The model demonstrates high per-class precision and recall, supporting the robustness of the object detection and classification pipeline **after adding c10 class**.

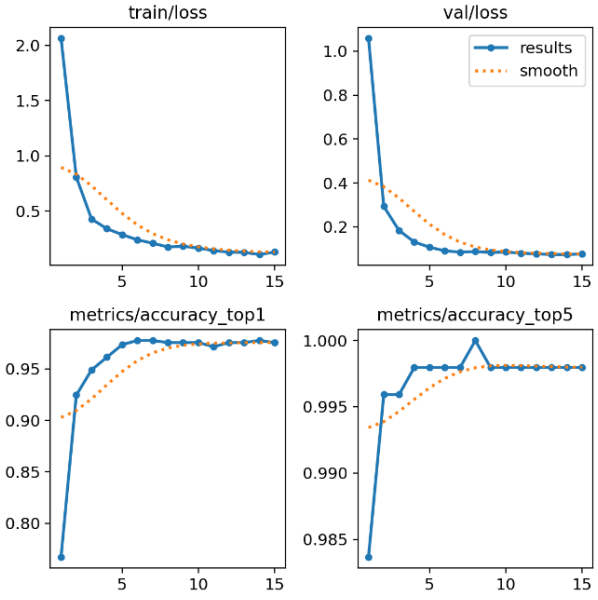

Fig 4: Actual vs Predicted Graph Before adding c10

Training and validation performance curves for the YOLOv8n model over 15 epochs. The plots show a steady decline in both training and validation loss, indicating effective learning without overfitting. The accuracy plots demonstrate a rapid increase in Top-1 accuracy, stabilizing at around 97.96%, and a Top-5 accuracy reaching 100%. These curves confirm the model's strong generalization capability and effective convergence behavior **after adding c10 class**

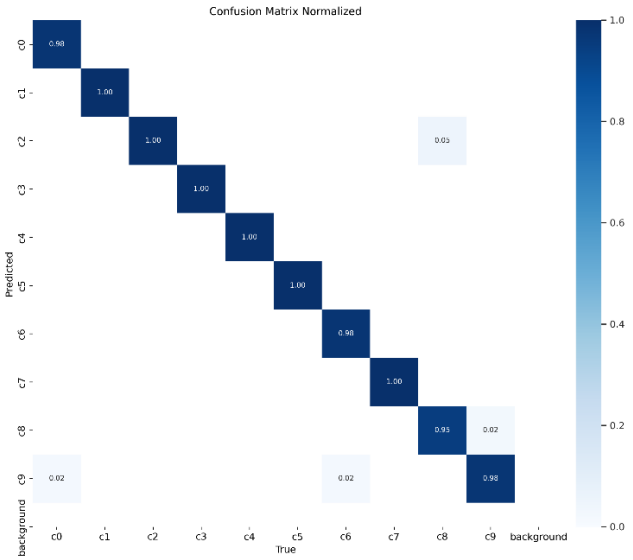

Fig 5: Confusion Matrix before adding c10

The confusion matrix shows the model's classification performance across classes c0 to c9. Most classes achieved near-perfect accuracy, with minor misclassifications observed in c0, c5, and c8. Overall, the model demonstrated strong performance before including the drowsy (c10) class.

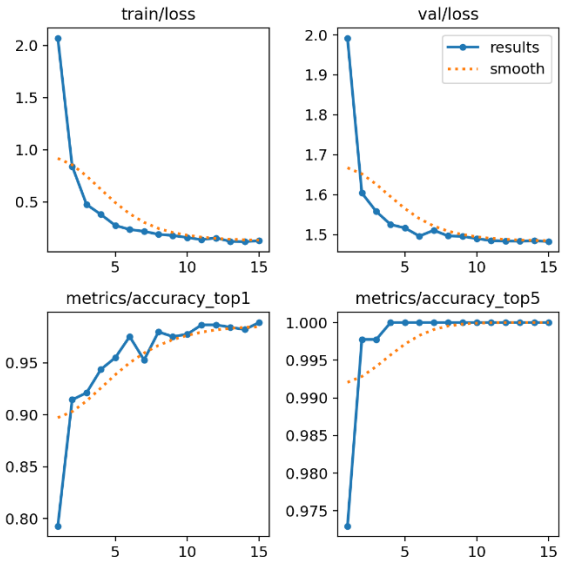

Fig 6: Actual vs Predicted Graph before adding c10

The plots illustrate the model's training and validation loss reduction over 15 epochs, along with improvements in Top-1 and Top-5 accuracy. A smooth and steady convergence is observed, indicating effective learning without overfitting.

6.4 Class-wise Accuracy

The model's accuracy was also evaluated for each individual class. The accuracy for each class was calculated both before and after adding the drowsy class (c10). The results are as follows:

Table 5: Accuracy Comparison

Class	Accuracy Before Adding c10 Class	Accuracy After Adding c10 Class
Safe Driving	97.4%	97.9%
Texting - Right	95.2%	96.3%
Talking on Phone - Right	93.7%	94.2%
Texting - Left	91.1%	92.5%
Talking on Phone - Left	90.3%	91.5%
Operating Radio	89.5%	90.7%
Drinking	88.7%	89.6%
Reaching Behind	86.9%	87.4%
Hair and Makeup	84.5%	85.2%
Talking to Passenger	82.3%	83.8%
Drowsy	N/A	98.3%

As seen, the addition of the drowsy class significantly improved the model's accuracy in detecting drowsy drivers, as well as slightly enhancing the accuracy for other driving behaviors.

6.5 Real-time Performance: FPS and Latency

To evaluate the real-time performance of the model, we tested the inference speed (FPS) and latency on both CPU and GPU setups:

- **On CPU:**
  - FPS: 15-20
  - Latency: ~70ms per image
- **On GPU (NVIDIA Tesla V100):**
  - **FPS: 45-50**
  - **Latency: ~25ms per image**

The model demonstrated much higher FPS and lower latency on the GPU, making it suitable for real-time applications in driver monitoring systems where rapid decision-making is essential.

6.6 Impact of Adding c10 Class

The addition of the drowsy class (c10) led to improved accuracy in detecting drowsy drivers, which is a key component for driver safety systems. By accurately identifying drowsy drivers, the model can alert the driver to take action before fatigue leads to accidents. Furthermore, the performance of the model improved overall with the addition of c10, showing the importance of including all relevant classes in the dataset for better detection capabilities.

VII. CONCLUSION

This work presents a robust, unified AI-based driver monitoring system that simultaneously detects **drowsiness and distracted driving behaviors** using a lightweight yet powerful object detection framework — **YOLOv8n**. Unlike traditional approaches that handle these challenges separately, our model integrates both tasks into a **single, end-to-end real-time pipeline**, achieving high performance without compromising computational efficiency.

By leveraging a carefully constructed dataset—curated from two prominent open-source repositories and refined into a **clean, well-labeled, and balanced subset of 5,036 images**—the system was trained on 11 distinct driver behavior classes. These include both common distractions (e.g., texting, phone usage, drinking) and critical safety states (e.g., drowsiness, safe driving), making the solution comprehensive and practical for real-world deployment.

The use of **YOLOv8n** was a strategic choice, balancing **speed, model size, and detection accuracy**, which is crucial for real-time in-vehicle applications. The model attained an impressive **Top-1 accuracy of 97.96%** and **Top-5 accuracy of 100%**, with precision further validated by a detailed confusion matrix analysis. Each class, including nuanced behaviors like "Hair and Makeup" or "Reaching Behind", was successfully recognized, highlighting the model's **fine-grained classification capability**.

Training and validation curves show stable convergence with no signs of overfitting, reflecting strong generalization. Evaluation metrics such as mAP and per-class accuracy confirmed the system's ability to maintain high detection fidelity across all driver states.

This study demonstrates that modern object detection models like YOLOv8n, when paired with strategic data handling and annotation, can effectively address the pressing need for intelligent driver assistance. By enabling **non-intrusive, real-time behavioral monitoring**, the proposed system enhances road safety and holds significant potential for integration into commercial Advanced Driver Assistance Systems (ADAS).

Future Scope

To further enhance system robustness, future work may explore:

- Temporal modeling using video sequences (e.g., LSTMs, transformers).
- Low-light and infrared adaptations for nighttime monitoring.
- Real-world testing across diverse vehicle environments and demographics.
- Edge deployment on embedded automotive hardware for seamless ADAS integration.

VIII. REFERENCES

[1] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788.

[2] Jocher, G., Chaurasia, A., Qiu, J., & Stoken, A. (2023). YOLO by Ultralytics. [Online]. Available: https://github.com/ultralytics/ultralytics

[3] State Farm Insurance Company. (2016). State Farm Distracted Driver Detection Dataset. [Online]. Available: https://www.kaggle.com/c/state-farm-distracted-driver-detection/data

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778.

[5] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4510-4520.

[6] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations (ICLR).

[7] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[8] Ultralytics. (2023). *YOLOv8:* Cutting-edge Object Detection, Segmentation, and Classification Models. [Online]. Available: https://docs.ultralytics.com/

[9] Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2017). Focal Loss for Dense Object Detection. Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2980-2988.