Driver Drowsiness Detection System

1st Pham Hoang Phuc SE192874 2nd Pham Quoc Huy SE180478 3rd Vo Tran Ngoc Trung SE192927 4th Dang Tien Dung SE193537 5th Nguyen Phuong Vu SE190840

***Abstract*—This report presents the core modeling and evaluation phase of a real-time driver drowsiness detection system. Multiple facial processing pipelines are compared, deep learning models for eye and mouth state classification are evaluated, and a hybrid yawn detection method combining geometric features (MAR) and CNN predictions is proposed. Three face detection methods—Haar Cascade, MediaPipe Face Mesh, and YOLOv8 Face—are assessed based on precision, recall, and frames per second (FPS). MediaPipe was selected as the foundation due to its superior balance of accuracy and real-time performance. A custom lightweight CNN achieved 98.70% accuracy in eye state classification, outperforming YOLOv11-Classifier (99.10%) in terms of inference speed. For yawning, the combined Mouth Aspect Ratio (MAR) and CNN method significantly reduces false alarms through dual verification. The final modular pipeline ensures robust, real-time performance on standard hardware, with a clear roadmap for embedded deployment.**

I. Introduction

This research develops a camera-based, real-time driver drowsiness detection system using prolonged eye closure and yawning behavior as key behavioral indicators. Leveraging a comprehensively prepared dataset—consisting of labeled eye images (MRL Eyes) and augmented mouth images (YawDD, processed in Roboflow with 224×224 normalized resolution, normalization, and augmentation)—this phase focuses on model architecture design, performance evaluation, and system integration.

The modules for detection, tracking, landmark extraction, region classification, and temporal decision logic are rigorously evaluated through quantitative metrics and trade-off analysis, with the goal of constructing an effective and practically deployable pipeline.

II. Problem & Context

Drowsiness is widely recognized as a significant risk factor for road accidents. While reported statistics may vary across studies and methodologies, the overall evidence base consistently highlights the need for effective, timely drowsiness warnings. This project addresses that need with an accessible, camera-based solution

III. Technological Solution and Project Significance

Proposed Solution (Computer Vision): A camera-based driver monitoring approach that analyzes facial behavior.

–Eye State Analysis: Monitor blink frequency, continuous eye closure (micro-sleeps), and eye openness. –Head Pose Recognition: Identify nodding or head-lowering patterns associated with fatigue.

Project Significance:

–Urgency: Contributes directly to reducing accidents and protecting lives and property. –Technological Feasibility: Leverages AI and open-source tools (e.g., OpenCV, MediaPipe) to build a low-cost, flexible system that runs on common laptops or smartphones. –Broad Application Potential: Extensible to night-shift worker monitoring, online learning, or control rooms to enhance productivity and safety.

IV. Objectives and Scope

A. Project Objectives

Develop a real-time system that detects key drowsiness indicators (e.g., eye closure, yawning) from webcam video with accuracy $\geq 85\%$ Provide instant audio and visual alerts when drowsiness is detected. Complete a live demonstration and comprehensive report within 10 weeks. Report model evaluation with confusion matrix, accuracy, and recall (and real-time metrics if available).

B. Scope and Limitations

Scope: Real-time detection from a single-driver webcam stream, focusing on prolonged eye closure and yawning behaviors. Limitations:

- Out of Scope: Multi-person detection, vehicle control, or the use of physiological sensors.
- Extreme Conditions: Performance is not guaranteed under very low light, glare, or extended facial occlusion.
- Assumption: The driver's face remains within the camera's field of view.

V. Methodology and Implementation Plan

The system follows a data-processing pipeline suitable for real-time operation.

A. ROI Extraction (Region of Interest)

Main Tool: MediaPipe (Face Mesh Landmarks) to locate eye and mouth regions.
Region of Interest: Crop eye and mouth ROIs from each frame to support downstream analysis and annotation.
Fallback Mechanism: If MediaPipe loses face tracking, switch to a HaarCascade-based fallback to maintain continuity.

B. Model (Detection & Classification)

Regarding the models evaluated, the system compared face detection and tracking methods including Haar Cascade, MediaPipe, and YOLOv8 Face. For eye state classification, a Custom CNN and the YOLOv11 Classifier were considered. Yawn detection specifically utilized a hybrid method combining MAR geometric features and a CNN.
The primary tasks of the model include face detection and tracking, followed by the classification of eye states (Closed/Open) and the detection of yawning behavior (Yawning/Normal).

C. Temporal Module

Purpose: Analyze behavior over time and reduce false alarms.
Approach: Compute temporal indicators such as PERCLOS (percentage of eye closure), blink frequency, and yawn count within a fixed time window.

D. Decision Logic

Alert if:

- Eyes remain closed for > 2 seconds, or
- ≥ 3 yawns per minute are observed.

Thresholds will be tuned to balance missed detections and false alarms.

E. Alerting System

Audio: Audible beep.
Visual: On-screen popup.
Extension (future): Haptic alerts (e.g., seat/steering vibration) in real-world integration.

VI. Expected Outcomes

A functional real-time drowsiness detection system with accuracy $\geq 85\%$ (per the defined test protocol).
A live demonstration and a detailed project report (including confusion matrix, accuracy, recall).
A solid foundation for subsequent development or deployment in broader contexts.

VII. Dataset Description and Preparation

A. Data Collection

In this study, the team used two main datasets to train two independent classification models, serving the driver drowsiness detection system based on facial features.

1) MRL Eyes Dataset: Extracted from the paper "A Survey on Drowsiness Detection – Modern Applications and Methods" [1]. The dataset comprises a total of 84.9K eye images, divided into two classes:

- `open_eyes`: approximately 43.0K images
- `closed_eyes`: approximately 41.9K images

The images were captured under various lighting conditions and viewing angles, which increases the diversity of the dataset. The MRL Eyes dataset is an open-source resource commonly used in Eye State Classification and Drowsiness Detection research [2].

2) YawDD Dataset (Yawning Detection Dataset): Based on the paper "Real-Time Driver-Drowsiness Detection System Using Facial Features" [3], this dataset consists of videos recording participants' faces during simulated driving. It includes three behaviors:

- Yawning
- Non-Yawning
- Talking

The data was downloaded directly from the official YawDD website [4] and is organized into two main folders:

- **Dash:** records the face from a frontal view.
- **Mirror:** records the face via the rearview mirror.

Using both viewing angles helps increase the model's generalization capability when deployed in real driving conditions.

3) Kaggle Yawn Dataset (Optional Dataset for Mouth State Detection): As an alternative or supplementary dataset for mouth state classification, the team also considered the Kaggle Yawn Dataset [5]

B. Data Labeling

1) MRL Eyes: The data was already labeled with the two states, `open_eyes` and `closed_eyes`, provided directly in the original set. Therefore, the team did not need to perform the labeling process again, only conducting random checks on samples to confirm label accuracy and consistency.

2) YawDD: Unlike MRL Eyes, the YawDD data only consists of raw, unannotated videos. The team therefore built a labeled image dataset through the following process.

a) Frame Extraction:: Using the OpenCV library, the team extracted individual frames from the videos in both the `Dash` and `Mirror` folders. The code used is as follows:

b) Filtering & Cleaning:: Blurred, duplicated, or ambiguous images (especially frames at the start of videos) were manually removed.

c) Classification and Label Assignment:: After inspection, the team obtained 1,448 valid images, equally distributed into two classes:

- `yawn`: 723 images
- `no_yawn`: 725 images

The labeling process was performed manually combined with internal cross-checking, ensuring high accuracy and reliability of the input dataset.

C. Data Splitting

1) MRL Eyes Dataset: The MRL Eyes dataset was uploaded and processed directly on the Roboflow platform, which automatically split the data into three sets with the ratio 70% for training, 15% for validation, and 15% for testing. This ensured no data leakage between sets.

2) YawDD Dataset: After cleaning, the YawDD dataset was split on Roboflow using the same ratio (70% train, 15% validation, 15% test). The platform ensured balanced and randomized splitting, consistent with later preprocessing and augmentation steps.

TABLE I
YAWDD DATASET STRUCTURE AFTER AUGMENTATION.

Dataset	Number of Images
Train Set	3,042
Validation Set	217
Test Set	217
Total	**3,476**

D. Data Preprocessing

After uploading and splitting on Roboflow, the team standardized the input format for the YOLOv11 Classification model [5]. The main steps included:

- **Resizing:** All images were resized to 224×224 pixels.
- **Normalization:** Pixel values normalized to [0, 1].
- **Format Conversion:** All images saved in `.jpg` format.

The 224×224 resolution ensures compatibility with YOLOv11 and most modern image classification models.

E. Imbalanced Data Handling

Both MRL Eyes and YawDD datasets are nearly balanced:

- MRL Eyes: open_eyes vs. closed_eyes ≈ 1:1.
- YawDD: 723 yawn vs. 725 no_yawn.

No oversampling or class weighting was required.

F. Data Augmentation

1) Rationale: Only the training set was augmented. **MRL Eyes:** No augmentation applied (dataset already large and diverse). **YawDD:** Small dataset (≈ 1,448 images), so enhancement was necessary to improve model generalization.

2) Platform: Roboflow: Roboflow provides integrated tools for image augmentation (based on Albumentations) and automatic splitting of data versions.

3) Techniques Used:

- Rotation: –15° to +15°
- Brightness: –30% to +30%
- Horizontal Flip
- Light Blur (1 px)

Each training image generated 2–3 variations, greatly expanding the dataset.

4) Dataset After Augmentation: Before augmentation, the YawDD split was:

- Train Set: 1,000 images
- Validation Set: 217 images
- Test Set: 217 images

After augmentation, Roboflow automatically re-split the data (88% Train – 6% Validation – 6% Test):

After augmentation, the total image count more than doubled, increasing diversity and reducing overfitting risk.

G. Data Cleaning

MRL Eyes: Already preprocessed and labeled; no cleaning required. **YawDD:** Manual review removed images that were:

- Unclear in face or mouth regions.
- Did not show yawning (normal/talking states).
- Duplicated or motion-blurred.

Resulting dataset was clean and consistent for model training.

H. Missing Data Handling

Both datasets contained no missing data after preprocessing; no imputation or removal was required.

I. Data Overview Summary

MRL Eyes: 84.9K images, no augmentation, 70–15–15 split.
YawDD: 1.45K original images, augmented to ≈13.5K, same split ratio. Augmentation expanded YawDD over twofold, improving YOLOv11-Classification model performance.

J. Conclusion

The team's data preparation process strictly followed the Data Science Pipeline:

- Verified data sources and legitimate collection.
- Accurate labeling with quality control.
- Proper train/validation/test split.
- Selective augmentation for generalization.

As a result, two high-quality datasets were built for the driver drowsiness detection system:

- Eye State Recognition: open/closed (MRL Eyes)
- Mouth State Recognition: yawn/no_yawn (YawDD)

These datasets provide a strong foundation for YOLOv11-Classification model training and evaluation, ensuring reliability of the experimental results.

VIII. EVALUATION METHODOLOGY

To ensure a fair and objective comparison among different face detection models, three representative algorithms—*Haar Cascade*, *MediaPipe Face Mesh*, and *YOLOv8 Face Detector*—were evaluated on the same labeled test dataset. The dataset was uniformly annotated following YOLO format to maintain consistency.

Each model was quantitatively evaluated using three metrics: Precision, Recall, and Frames Per Second (FPS). These metrics were computed through an automated evaluation pipeline implemented in Python using OpenCV, MediaPipe, and Ultralytics YOLOv8 (PyTorch).

A. Evaluation Metrics

1) Precision (Accuracy): Precision measures the proportion of correctly detected faces among all detected faces:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

where TP (True Positive) represents correctly detected faces, and FP (False Positive) represents incorrectly detected regions.

2) Recall (Sensitivity): Recall quantifies the model's ability to detect all faces present in the images:

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

where FN (False Negative) denotes missed face detections.

3) Frames Per Second (Processing Speed): FPS measures real-time performance by calculating the number of frames processed per second:

$$FPS = \frac{N_{images}}{T_{total}} \tag{3}$$

where N_{images} is the number of images in the test set and T_{total} is the total processing time.

B. Bounding Box Evaluation with IoU

For both Haar Cascade and deep learning-based detectors, prediction accuracy was validated by comparing the predicted bounding boxes with ground truth boxes using the Intersection over Union (IoU) metric:

$$IoU = \frac{Area(B_{pred} \cap B_{gt})}{Area(B_{pred} \cup B_{gt})} \tag{4}$$

A detection is considered valid (*True Positive*) if:

$$IoU \geq 0.5 \tag{5}$$

C. Data Aggregation

After determining TP, FP, and FN across all test samples, overall Precision and Recall values were computed for each model. These aggregated results were then used to compare model robustness, detection reliability, and computational efficiency under varying lighting and pose conditions.

D. Face Detection Performance

Three representative face detection algorithms were evaluated: Haar Cascade, MediaPipe Face Mesh, and YOLOv8 Face Detector. Table II presents the comparison on a unified labeled test set using common metrics: Precision, Recall, and Frames Per Second (FPS).

TABLE II
PERFORMANCE COMPARISON OF FACE DETECTION METHODS

Method	Precision	Recall	FPS (CPU)	FPS (GPU)
Haar Cascade	0.607	0.433	39.5	N/A
MediaPipe	0.962	0.975	359	N/A
YOLOv8 Face	0.987	0.994	23.4	119.1

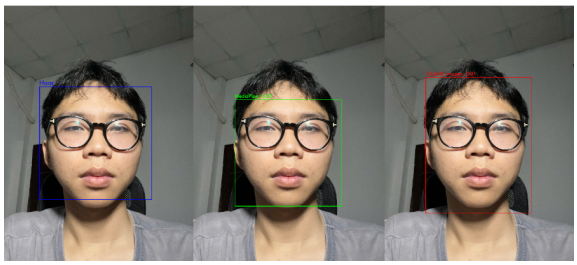

Fig. 1. Example of detected faces using Haar Cascade, MediaPipe, and YOLOv8.

YOLOv8 achieved the highest accuracy but required GPU acceleration to maintain real-time performance. MediaPipe provided the best trade-off between precision and speed, operating efficiently on CPU hardware. Haar Cascade remained as a baseline reference but exhibited poor robustness under varying lighting and pose conditions.

E. Facial Landmark Visualization

Facial landmarks are crucial for defining regions of interest such as eyes and mouth. Both Dlib and MediaPipe were compared in terms of landmark precision and stability. MediaPipe provided denser and more consistent landmark outputs, enhancing subsequent eye and mouth analysis.

Fig. 2. Visualization of landmarks from Dlib vs. MediaPipe.

F. Eye State Classification Performance

Two deep learning models were trained for binary eye-state classification (*Open* vs. *Closed*):

- **Custom CNN:** A lightweight 5-layer CNN [3] achieving 98.70% accuracy on the test set.

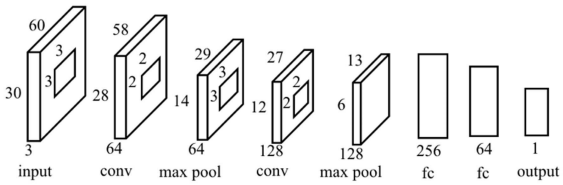

Fig. 3. Diagram of CNN architecture for eye classification.

TABLE III
CLASSIFICATION REPORT FOR EYE STATE CNN MODEL

Class	Precision	Recall	F1-score	Support
Close-Eyes	0.9888	0.9849	0.9869	6292
Open-Eyes	0.9853	0.9891	0.9872	6443
Accuracy			0.9870	12735
Macro avg	0.9871	0.9870	0.9870	12735
Weighted avg	0.9871	0.9870	0.9870	12735

- **YOLOv11 Classifier:** Achieved 99.10% accuracy, slightly higher but computationally heavier.

Confusion matrix results demonstrated that both models maintained excellent separation between the two classes. The custom CNN was preferred for deployment due to its lower inference latency and minimal hardware requirements.

TABLE IV
CLASSIFICATION REPORT FOR EYE STATE YOLOV11 CLASSIFIER

Class	Precision	Recall	F1-Score	Support
Close-Eyes	0.9911	0.9906	0.9909	6292
Open-Eyes	0.9908	0.9913	0.9911	6443
Accuracy			**0.9910**	12735
Macro Avg	0.9910	0.9910	0.9910	12735
Weighted Avg	0.9910	0.9910	0.9910	12735

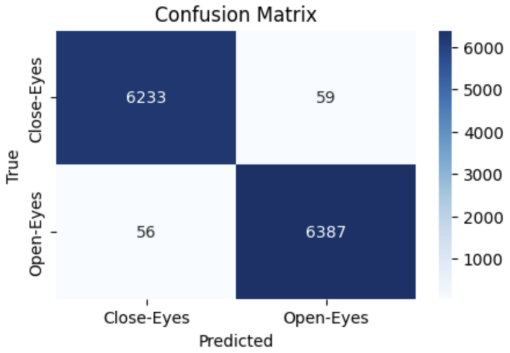

Fig. 4. Confusion matrix for eye-state classification model.

G. Mouth (Yawn) Detection Performance

Yawning behavior is an important physiological indicator for assessing driver fatigue. In this study, two independent approaches were implemented for mouth region analysis: (1) a geometric method based on the *Height-to-Width Ratio (H/W)*, and (2) a CNN-based classifier trained to detect yawning states.

H. A. Geometric Method: Height-to-Width Ratio (MAR)

Following the approach adopted in the paper 'Driver Drowsiness Detection using Machine Learning and Deep Learning' [6], the degree of mouth opening is represented using the *Mouth Aspect Ratio (MAR)*, defined as the ratio between the vertical height (H) and the horizontal width (W) of the mouth region.

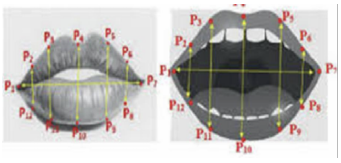

Fig. 5. Illustrates the measurement of mouth height and width.

Facial landmarks were extracted to locate mouth keypoints. Let p_1 and p_5 denote the left and right mouth corners, defining the horizontal width W. The vertical height H is computed as the average distance between three pairs of upper and lower lip landmarks:

$$(p_2, p_8), (p_3, p_7), (p_4, p_6)$$

The MAR is calculated as:

$$MAR = \frac{||p_2 - p_8|| + ||p_3 - p_7|| + ||p_4 - p_6||}{2 \times ||p_1 - p_5||} \quad (6)$$

The MAR value increases when the mouth opens and decreases when it closes. A mouth is considered open when:

$$MAR > \tau \quad (7)$$

where τ is a predefined threshold empirically determined from the dataset. If the threshold is exceeded continuously for 2–3 seconds, the system confirms a yawning event.

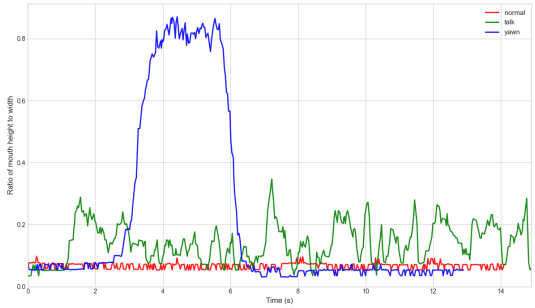

Fig. 6. Mouth H/W Ratio Comparison for Different Actions.

I. Deep Learning Method: CNN-Based Mouth Classifier

To enhance detection robustness under varying lighting and facial expressions, a lightweight CNN model was trained to classify cropped mouth regions into two categories:

- **Yawn:** Mouth open, representing a yawning state.
- **Non-Yawn:** Includes talking, smiling, or closed mouth.

The model was trained on grayscale mouth images resized to 30×60 pixels. Its architecture follows the same lightweight CNN structure used in the Eye Region Model, optimized for binary classification with minimal parameters for real-time deployment.

TABLE V
CLASSIFICATION REPORT FOR MOUTH STATE (YAWN / NO-YAWN)

Class	Precision	Recall	F1-Score	Support
no_yawn	0.9842	0.9614	0.9727	389
yawn	0.9613	0.9842	0.9726	379
Accuracy			0.9727	
Macro avg	0.9728	0.9728	0.9727	768
Weighted avg	0.9729	0.9727	0.9727	768

J. Combined MAR–CNN Hybrid Strategy

To minimize false positives (e.g., talking or smiling mistaken as yawns), a hybrid decision rule was applied. A yawn is confirmed only when both the geometric MAR threshold and CNN classifier output indicate a yawning state simultaneously. This combination significantly improves reliability and robustness in real-world driving conditions.

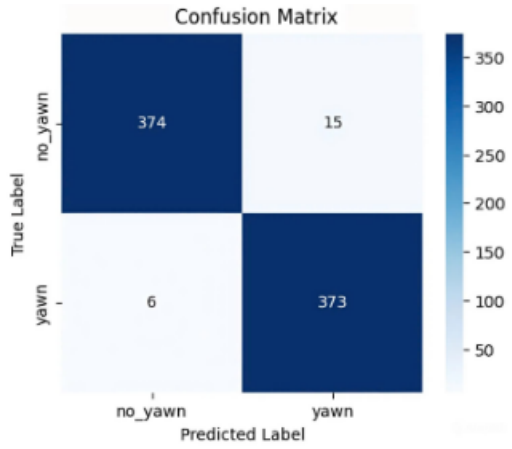

Fig. 7. Confusion matrix for the CNN-based mouth classification model.

IX. Discussion

The experimental results validate the modular design choices and provide clear insights into the trade-offs inherent in the pipeline.

Regarding the pre-processing component, the evaluation confirmed a distinct trade-off between accuracy and real-time performance. As hypothesized, YOLOv8 provided the highest precision in face and landmark detection. However, its computational cost constrained performance on standard CPUs. In contrast, MediaPipe delivered the optimal balance for the system, maintaining stable detection and integrated tracking at a high efficiency suitable for real-time applications.

In the eye classification module, the custom-designed lightweight CNN demonstrated its effectiveness. It achieved competitive accuracy when benchmarked against the YOLOv11-Classifier baseline, but did so with significantly lower latency. This result supports the conclusion that a carefully designed, task-specific small CNN can outperform larger, more general-purpose architectures for this specific classification task.

For yawn detection, the proposed hybrid MAR–CNN fusion method successfully enhanced reliability. By combining geometric features (MAR) with a CNN classifier, the system demonstrated improved robustness and a reduction in false alerts previously triggered by non-drowsy actions, such as speech or expression changes.

However, significant limitations were identified. System performance was observed to degrade noticeably under poor illumination and when subjects wore heavy occlusions (e.g., sunglasses or face masks). These findings align with the planned future work, emphasizing the necessity of addressing these challenges through data augmentation or, more robustly, through the integration of infrared (IR) sensing.

X. Conclusion

This report presents the design, implementation, and evaluation methodology for a driver drowsiness detection system based on a modular architecture. We have constructed a complete pipeline that allows for flexible experimentation and interchange of core components, including:

- Pre-processing: A comparison of three face and landmark detection methods (Haar, MediaPipe, and YOLOv8) to select the optimal solution for the speed/accuracy trade-off.
- Eye Classification: Implementation of an efficient custom CNN model and its comparison against a baseline model (YOLOv11-Classifier).
- Yawn Detection: Proposal of a hybrid (fusion) method that combines geometric features (MAR) and a CNN model to enhance reliability.

The experimental results from this system (to be detailed in the final report) are expected to demonstrate that a hybrid pipeline, utilizing MediaPipe for pre-processing and lightweight, task-specific CNN models, can achieve an optimal balance between accuracy and real-time performance.

Future work will focus directly on addressing the identified limitations. The most prominent directions include collecting additional data in low-light conditions and utilizing infrared (IR) cameras to mitigate issues with lighting and occlusions (such as sunglasses), while also optimizing the models for deployment on embedded devices.

References

[1] B. Fu, F. Boutros, C.-T. Lin, and N. Damer, "A Survey on Drowsiness Detection – Modern Applications and Methods," 2024, *arXiv:2408.12990v1* [cs.CV]. [Online]. Available: https://arxiv.org/abs/2408.12990v1. To be published in *IEEE Trans. Intell. Veh.*

[2] E. Sojka, J. Gaura, and R. Fusek, "MRL Eye Dataset," Media Research Lab, VSB - Technical University of Ostrava, 2018. [Online]. Available: http://mrl.cs.vsb.cz/eyedataset.html.

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

[4] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, "YawDD: A Yawning Detection Dataset," in *Proc. ACM SIGMM Conf. Multimedia Syst. (MMSys)*, Singapore, Singapore, 2014, pp. 24–28, doi: 10.1145/2557642.2563678.

[5] D. Vazquez, "Yawn Dataset," Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/davidvazquezcic/yawn-dataset.

[6] K. Srinivas, G. V. V. M. U. V. Karthik, I. Suhas, *et al.*, "Driver Drowsiness Detection using Machine Learning and Deep Learning," *Int. J. Adv. Res. Sci., Commun. Technol.*, vol. 4, no. 4, p. 234, Apr. 2024, doi: 10.48175/IJARSCT-17442.