Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 15, Issue. 1, January 2026, pg.73 – 80

A Comparative Study of YOLO-Family Models for Driver Eye-State Detection under Mask and Glasses Wearing Conditions

In Cheol Park[1]; Taek Lee[2]

[1,2]Division of Computer Science and Engineering, Sun Moon University, Republic of Korea
[1] incheol2250@sunmoon.ac.kr; [2] comtaek76@sunmoon.ac.kr
(Corresponding Author: Taek Lee)

DOI: https://doi.org/10.47760/ijcsmc.2026.v15i01.007

Abstract: Driver drowsiness is a major contributing factor to traffic accidents, and accurate and timely detection is crucial for enhancing road safety. With recent progress in deep learning-based object detection, YOLO-based models have been widely adopted for real-time driver condition monitoring using in-vehicle cameras. This paper does not directly classify driver drowsiness; instead, it focuses on robust eye-state (open/closed) detection, a key indicator for drowsiness assessment. To detect the driver's eye state under diverse conditions (e.g., mask wearing, glasses wearing, and variations in facial angle), several YOLO-based models were comparatively evaluated under identical experimental conditions. YOLOv5, YOLOv7, YOLOv8, and YOLO-Former were employed and evaluated on publicly available image datasets. The trade-off between detection accuracy and real-time performance was systematically analyzed using mAP, precision, and recall. Experimental results indicated that performance was strongly affected by environmental complexity, and that lightweight models could achieve competitive accuracy while enabling faster inference. These findings provide practical insights into selecting an appropriate YOLO architecture for real-time driver drowsiness detection systems.

Keywords: Driver eye state (open/closed) detection, YOLO-based models, performance comparison, object detection

© 2026, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

I. INTRODUCTION

Drowsy driving remains a significant traffic safety issue worldwide. Previous studies, based on statistical analyses of Korean driver populations, have reported that fatigue and drowsiness are closely related to accident frequency and severity, and that accidents occurring in a drowsy state tend to have higher fatality rates than general traffic accidents[1], [2]. These results support the need for driver monitoring systems that can detect drowsiness early and issue warnings in traffic environment.

In the context of drowsy driving detection, numerous studies have consistently demonstrated that eye-related features such as eye state (open/closed), eye closure ratio, eye closure duration, and blink frequency serve as reliable indicators of drivers' drowsiness and fatigue levels[3], [4]. In particular, eye state information represented as a binary classification (open/closed) is regarded as a relatively simple yet highly sensitive measure of changes in drowsiness. Building on this evidence, many researchers have proposed lightweight drowsiness detection models that primarily focus on the eye region rather than analyzing the entire face.

Meanwhile, early drowsy driving detection systems primarily relied on traditional image processing and machine learning techniques, such as Haar-cascade and HOG (Histogram of Oriented Gradients) features combined with SVM (Support Vector Machine) classifiers. However, prior studies have reported that these methods are vulnerable to variability in real-world driving conditions such as wearing glasses, sunglasses, or masks, and changes in facial angle and tilt thereby increasing missed detections and false positive rates[5]. In other words, while these methods may function under limited conditions, concerns have been raised regarding the difficulty of ensuring reliable performance in real-world driving environments where multiple environmental factors interact in complex ways.

To address these limitations, recent studies have increasingly adopted CNN (Convolutional Neural Network)-based object detection techniques for driver state recognition. In particular, YOLO (You Only Look Once) family models have gained attention as a promising option for DMS (Driver Monitoring Systems) that require real-time performance, as their one-stage architecture enables both fast inference and strong detection accuracy. Previous studies have reported that YOLO-based models can be applied to various driver state recognition tasks such as fatigue/drowsiness detection, mobile phone use detection, and seatbelt non-use detection achieving both real-time processing and high detection accuracy[6]. These findings support the adoption of YOLO models in real-time driver monitoring systems.

Furthermore, prior studies have demonstrated that even within the same YOLO family, performance and computational complexity can vary significantly depending on the version (v3, v5, v7, v8, etc.), backbone and neck architecture, lightweight model design strategies, and whether attention modules are incorporated[6], [7]. For example, some studies reported that introducing a lightweight backbone and an improved FPN (Feature Pyramid Networks) to the basic YOLOv8 architecture resulted in significant improvements in mAP, precision, and recall compared to existing models in driver fatigue detection[6]. This suggests that even when addressing the same eye state (open/closed)-based drowsiness detection problem, performance differences can arise depending on which YOLO-based model is selected and how its architecture is adjusted.

With respect to environmental factors, prior studies indicate that wearing facial accessories or coverings (e.g., masks and glasses) is an important variable affecting face/eye detection and drowsiness detection performance. Some studies have reported that drowsiness classification accuracy decreases markedly depending on whether masks or sunglasses are worn, highlighting that reliable recognition of key regions such as the eyes and mouth remains challenging when the face is partially occluded[7]. Other studies have also shown that facial occlusions, including glasses wearing, hinder eye state recognition and contribute to performance degradation in deep learning–based drowsiness detection models[4], [5]. Taken together, accessory wearing can be considered a critical factor that influences how accurately YOLO-based models detect the eye region.

In this paper, we systematically compared multiple YOLO-based models under identical conditions and analyzed eye-detection performance across accessory-wearing conditions by using the dataset from the AI-Hub platform (https://aihub.or.kr/). The dataset covers diverse recording scenarios including real-road driving, semi-controlled, and controlled environments and includes variations in mask/glasses wearing and facial orientation. As such, it provides a valuable basis for quantitatively evaluating architecture-dependent characteristics and performance differences attributable to accessory wearing.

Considering these trends in prior research and the available data, there is a need to compare the eye-detection performance of YOLO-based models on driver-state video data that reflect real-world road conditions, and to empirically analyze how performance varies under accessory-wearing conditions. In particular, since prior studies have repeatedly identified facial occlusion as a key factor that degrades performance[4], [5], [7], and have reported architecture-dependent performance differences, this paper formulates the following research hypotheses:

- *Hypothesis 1*: Even when trained on the same dataset, eye-detection performance differs significantly among YOLO-family models (e.g., YOLOv5, YOLOv7, YOLOv8, YOLOv5-Former).

© 2026, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

- *Hypothesis 2*: Eye-detection performance of YOLO-family models differs significantly depending on accessory-wearing conditions (mask/glasses wearing).

This paper aims to provide empirical evidence on, firstly, which YOLO-family model should be selected when designing a driver monitoring system that reflects real-world road conditions and, secondly, how performance varies under accessory-wearing conditions. To this end, the above hypotheses are tested using the AI-Hub driver-state video dataset.

This paper systematically compared eye-state (open/closed) detection performance across YOLO-family models under different accessory-wearing conditions. Practical applicability was assessed using only the AI-Hub real-road driving data, and the impact of a Transformer-based head on small-object (eye) detection was empirically analyzed.

II. RELATED RESEARCH

A. *Background of Study*

Driver Monitoring Systems (DMS) have become an essential technology in traffic safety. In particular, drowsy driving during prolonged driving is a major risk factor for severe accidents, highlighting the growing need for AI-based systems that can monitor a driver's state in real-time and provide timely warnings.

In drowsy driving detection research, the eye state (open/closed) is one of the most fundamental yet important cues. Earlier approaches largely relied on traditional computer-vision pipelines such as Haar-cascades and HOG features with SVM classifiers; however, they are susceptible to real-world variations and occlusions, including glasses wearing.

To address these issues, lightweight object detection models from the YOLO family have been widely adopted in recent studies. YOLO provides an effective balance between speed and accuracy, making it well suited for real-time applications.

B. *Experimental Setup*

1) YOLO-Family Models Compared: This paper compared the performance of four lightweight YOLO-family models on the same driver-state video dataset. The models included YOLOv5n (the nano variant of YOLOv5 in the Ultralytics implementation), YOLOv7n, YOLOv8n, and YOLO-Former, which incorporates a Transformer-based architecture. All four models are nano/lightweight variants designed to meet the computational constraints of real-time processing. This section compares their detection performance and inference speed under identical training conditions.

2) YOLOv5n: YOLOv5[8] is a PyTorch-based one-stage object detector that inherits the grid-based multi-scale prediction paradigm of YOLOv3 and YOLO4, while incorporating a CSPNet-based backbone, a PANet-based neck, and various data augmentation and training techniques[8]. In the experiments, a single lightweight variant from the YOLOv5 family was selected and evaluated.

The overall architecture of YOLOv5 comprises three components: a backbone, neck, and head[8]. The backbone is built from Conv-BN-SiLU blocks and C3 modules, which progressively down-sample the input image (640×640×3). Through this process, feature maps are extracted at three scales: 80×80 (P3), 40×40 (P4), and 20×20 (P5). An SPPF (Spatial Pyramid Pooling-Fast) module is applied to the lowest-level 20×20 feature map to incorporate contextual information with a larger receptive field[8].

The neck employs a CSP-PAN architecture to fuse the P3, P4, and P5 feature maps via up-sampling, down-sampling, and concatenation, enabling information sharing across the three resolutions and supporting objects of various sizes—from small to large[8]. Finally, the head applies 1×1 convolutions to the fused 80×80, 40×40, and 20×20 feature maps to simultaneously predict anchor-based bounding box coordinates, objectness scores, and class probabilities. The final detections are obtained by applying a confidence threshold followed by non-maximum suppression(NMS)[8].

3) YOLOv7n: YOLOv7 is a real-time object detection architecture with a three-part design consisting of a backbone, neck, and head. It progressively extracts features from the input image, fuses them across multiple scales, and ultimately outputs object localization and classification results. The network first generates multi-scale features (P3, P4, and P5) in the backbone, fuses them in the neck via top-down and bottom-up pathways, and then performs bounding box regression and classification in the head using the feature maps at each scale. This overall pipeline helps improve accuracy while preserving the high-speed detection capability of the YOLO family [9].

The backbone is built around Extended Efficient Layer Aggregation Network (E-ELAN) blocks, a core component of YOLOv7, which progressively down-sampling the input image (640×640) across multiple stages to produce multi-scale feature maps at P3, P4, and P5. The E-ELAN design splits and merges features through multiple branches and employs Cross-Stage Partial (CSP) connections to facilitate stable gradient propagation through deeper layers. Additionally, by combining 1x1 and 3x3 convolutions to capture both local and global patterns, it maximizes representational capacity without increasing the number of parameters. This backbone structure provides a foundational feature representation for recognizing objects of various sizes [10].

© 2026, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

The neck adopts a structure that combines a Feature Pyramid Network (FPN) and a Path Aggregation Network (PANet), repeatedly fusing multi-scale features (P3, P4, and P5) from the backbone through top-down and bottom-up pathways. Higher-level semantic features are up-sampling and combined with lower-level spatial details, while lower-level features are down-sampling and merged back into higher-level representations. Additionally, the ELANBlock within the neck enhances representational capacity by rearranging and expanding fused features. As a result, the neck produces multi-scale feature maps that enable robust detection of objects across a wide range of sizes, from small to large [11].

The head is designed in an Anchor-free manner and takes the three-scale feature maps (P3, P4, and P5) from the neck as input to jointly predict bounding box regression (localization), objectness, and class probabilities. One of the key components of the YOLOv7 head is the RepConv/RepVGG module. During training, it employs a multi-branch structure to enhance feature learning, whereas during inference it is re-parameterized into a single 3×3 convolution, substantially improving inference speed. In addition, ImplicitA and ImplicitM modules are incorporated into the head to learn latent information related to class-specific biases and regression characteristics, further enhancing the model's overall representational capacity. Finally, candidate boxes generated at each scale are filtered by non-maximum suppression (NMS) to produce the final detections[9].

4) YOLOv8n: The YOLOv8 model used in this paper is an object detection architecture released by Ultralytics[12], which substantially redesigns the backbone and head to address limitations of earlier YOLO versions. A key component of YOLOv8 is the C2f (Cross Stage Partial bottleneck with two convolutions) module-integrates the ELAN (Efficient Layer Aggregation Network) design principles into the conventional C3 module. This module splits the input feature map into two branches: one branch is forwarded directly, while the other is processed through multiple bottleneck blocks, after which the two branches are concatenated. This allows the model to remain lightweight while improving training efficiency by facilitating gradient flow through deeper layers of the network. Additionally, the detection head adopts an Anchor-free approach that directly predicts object center points and boundary distances without relying on predefined anchor boxes, thereby reducing design complexity. It also employs a Decoupled Head that handles classification and regression in separate branches, mitigating task interference between the two objectives. In particular, by combining the DFL (Distribution Focal Loss) and CIoU (Complete IoU) loss functions for localization regression, the model can achieve precise localization while accounting for uncertainty in object boundaries. Based on the proposed architecture, the model was trained for 50 epochs and showed favorable convergence behavior and accuracy. In the final evaluation, mAP@0.5 (mAP50), a core metric for object detection, reached 0.934, while the stricter mAP@0.5:0.95 (mAP50–95) was 0.665. These results suggest that the C2f module and the decoupled head contribute to precise and robust detection even under complex visual conditions.

5) YOLO-Former: YOLO-Former, originally proposed for object detection in drone-captured scenarios[13], is an enhanced detector built upon the YOLOv5 architecture. To improve detection performance, it integrates a Transformer encoder block, CBAM (Convolutional Block Attention Module), and an additional prediction head.

The backbone is based on the CSPDarknet53 architecture used in YOLOv5, with a Transformer encoder block added at the final stage to enhance CNN-based feature extraction. The Transformer encoder consists of multi-head attention and a feed-forward network, learning long-range dependencies within feature maps to capture global context that CNNs may struggle to model.

The neck is based on the PANet (Path Aggregation Network) structure of YOLOv5, integrating multi-resolution features via up-sampling and down-sampling to effectively propagate information across object scales. In addition, CBAM (Convolutional Block Attention Module) and a Transformer encoder block are incorporated into the original PANet, enabling more stable feature fusion even for small objects and complex backgrounds.

The head is a key component of YOLO-Former and adopts a Transformer Prediction Head (TPH), which improves upon the convolution-based head in YOLOv5. By leveraging global contextual information more effectively than convolutional heads, TPH is advantageous for detecting visually similar targets and small objects. In addition to the standard three detection heads, an extra head is introduced to improve small-object detection, resulting in a four-head design. Although replacing the entire backbone with Transformer layers would substantially increase computational cost, incorporating Transformer modules only in the head provides global context with a limited increase in computation.

III. DATA PREPROCESSING

This study utilized the AI-Hub "Driver State Information Video for Drowsy Driving Prevention" dataset, which was constructed with funding from the Ministry of Science and ICT (MSIT) and support from the National Information Society Agency (NIA). The dataset is publicly available through the AI-Hub platform.

To build a stable and generalizable object detection model, a systematic data preprocessing pipeline is essential. This section outlines the preprocessing steps performed in our study, including data structure analysis, label format standardization and directory organization, as well as data normalization and format conversion.

© 2026, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

A. Data Structure

The AI-Hub driver state information dataset was initially released in June 2021, and comprises three scenarios depending on the recording environment:

- Real-road driving data,
- Semi-controlled environment data,
- Controlled environment data.

This study utilized only real-road driving data to evaluate the model's adaptability to real-world driving environments. The dataset includes various factors such as mask/glasses wearing and the driver's eye state (open/closed). The raw data are provided as frame-level images extracted from videos, along with JSON files containing per-frame annotations.

B. Annotation Format Conversion

The AI-Hub dataset provides object location information in a JSON structure; therefore, it must be converted into YOLO-format .txt labels to train YOLO-based models. In this paper, the coordinates provided by AI-Hub were transformed into normalized YOLO labels consisting of the center coordinates and the width/height, as follows.

- x, y, w, h: Original bounding box coordinates (top-left x, y, width, height)
- W, H: Normalized YOLO
- x', y', w', h': Normalized YOLO coordinates

The normalized YOLO coordinates were computed as follows: $x' = (x + w / 2) / W$, $y' = (y + h / 2) / H$, $w' = w / W$, $h' = h / H$.

All converted labels were saved as text files with the same filenames as the image files, making them directly usable by YOLO-based models. During this process, samples lacking eye-region annotation information were excluded from training.

C. Class Definition and Unified Labeling

In this study, the eye state was defined using binary classes: eyes closed (0) and eyes open (1). Mask/glasses wearing was not a detection or classification target but an external factor that may affect performance; therefore, the labels were standardized to keep a consistent class scheme across all conditions.

IV. METHODOLOGY

In this paper, YOLO-family models (YOLOv5n, YOLOv7n, YOLOv8n) and a Transformer-augmented model (based on the TPH-YOLOv5 architecture) were trained on the pre-processed driver eye-state (open/closed) dataset. In Fig. 1, the model development process was structured as follows:

- Defining the model architecture,
- Configuring the training hyperparameters,
- Initializing the model weights,
- Performing model training.

Each model was designed to maintain identical experimental conditions, enabling a fair comparison.

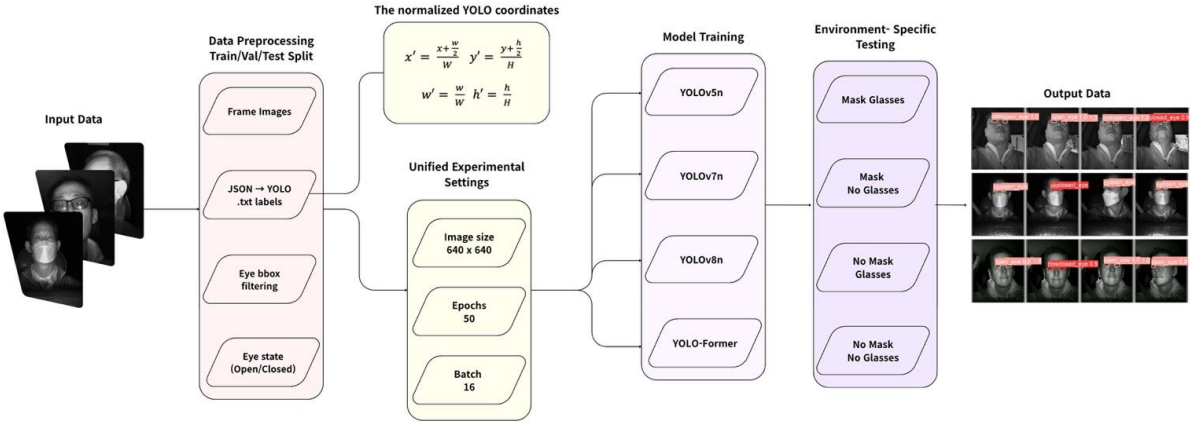

Fig. 1 The overall experimental pipeline

A. Model Architecture Design

1) Base Architectures of YOLOv5, YOLOv7, YOLOv8: The YOLOv5n, YOLOv7n, and YOLOv8n models utilize the base architecture released by Ultralytics as-is, while maintaining the input resolution (640×640) and applying the same augmentation strategy to ensure suitability for detecting ultra-small objects, such as eye blinks.

© 2026, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

2) Creating the YOLO-Former Model (TPH-YOLOv5-based): YOLO-Former is an architecture based on YOLOv5s that inserts a Transformer encoder block into the head and incorporates attention mechanisms to strengthen visual context. In our study, the head architecture was modified as follows, following the TPH-YOLOv5 paper[13]:

- Replace some CNN blocks in the head with a Transformer encoder block,
- Strengthen global-context learning via Self-Attention,
- Increase the use of high-resolution features in the Feature Pyramid to improve small-object detection (eyes),
- Optionally insert CBAM (Convolutional Block Attention Module) after the neck to suppress noise.

These modifications are expected to improve small-object detection precision compared to YOLOv5 and enhance robustness to environmental variations.

B. *Hyperparameter Configuration*

1) Common Training Settings: To ensure a fair comparison between models, the training settings were standardized across all four models. In all experiments, the number of training epochs was fixed at 50, the batch size was set to 16, and the input image resolution was uniformly set to 640×640. The same training, validation, and test data splits were used, and data augmentation techniques were applied as identically as possible. This approach aimed to focus the analysis on the impact of differences in the model architecture itself on performance. Other hyperparameters (e.g., optimizer, initial learning rate, and loss function) followed the default settings of each model implementation, while comparison-sensitive factors-namely the number of epochs, batch size, and input size-were kept identical across all four models.

V. EXPERIMENTAL RESULTS

Model evaluation was conducted on the held-out test set from the AI-Hub drowsy driver video dataset, which contains independent combinations of environmental conditions (MG, NG, NN, and MN). To compare the performance of the YOLO-family models and YOLO-Former, the following evaluation metrics were used.

1) Evaluation Metrics: This paper evaluated object detection performance using four metrics: Precision, Recall, mAP50, and mAP@0.5:0.95. Precision is the proportion of true positives among the bounding boxes predicted by the model, and Recall is the proportion of ground-truth objects that are successfully detected. mAP50 denotes mean average precision at an IoU threshold of 0.5, reflecting performance under a relatively relaxed matching criterion. mAP@0.5:0.95 denotes mean average precision averaged over IoU thresholds from 0.5 to 0.95, providing a stricter, comprehensive measure that reflects both detection accuracy and bounding-box localization quality. In particular, mAP@0.5:0.95 was used as the primary metric to assess the overall detection capability of the models.

2) Environment-Specific Testing Method: The test set consisted of four environment combinations (MG: Mask On, Glasses On; NG: No Mask, Glasses On; NN: No Mask, No Glasses; MN: Mask On, No Glasses). However, because the default YOLOv5 evaluation pipeline assumes a single-directory input, a filename-based categorization scheme was employed to conduct environment-specific evaluations.

3) Training Results: The training results are summarized in Table 1, and the environment-specific test results by mask–glasses combinations are presented in Table 2. In terms of mAP@0.5:0.95, YOLO-Former achieved 0.75386, which is markedly higher than the other models (YOLOv5n, YOLOv7n, and YOLOv8n; approximately 0.60–0.66), indicating the best overall performance. In addition, YOLO-Former maintained balanced performance, with both Precision and Recall exceeding 0.92, suggesting that it is not merely overfitted to specific environments but is the most stable model in terms of overall detection performance and generalization capability. Although YOLO-Former is structurally more complex, it was included as a baseline for comparison because it achieves improved detection performance while remaining within the feasible range for real-time processing.

TABLE I
TRAINING RESULTS FOR EACH MODEL

Model	Precision	Recall	mAP-50	mAP-50:95
YOLOv5n	0.896	0.898	0.934	0.662
YOLOv7n	0.884	0.896	0.930	0.622
YOLOv8n	0.888	0.897	0.932	0.663
YOLO-Former	0.932	0.923	0.964	0.753

TABLE II
LEARNING OUTCOMES BY ENVIRONMENT (MAP-50:95)

Environments	YOLOv5n	YOLOv7n	YOLOv8n	YOLO-Former
MG	0.395	0.352	0.407	0.651
MN	0.460	0.428	0.509	0.695
NG	0.702	0.680	0.712	0.744
NN	0.779	0.746	0.800	0.760

© 2026, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

Table 2 summarizes mAP@0.5:0.95 across different environments. All four models consistently achieved the highest mAP@0.5:0.95 in the NN condition (no mask, no glasses), while the lowest performance was observed in the MG condition (mask and glasses worn simultaneously). This suggests that increased facial occlusion reduces visual information around the eyes, making detection more challenging. Taken together with the overall results in Table 1, YOLO-Former showed the most stable and best performance across most conditions, with particularly strong robustness under occluded settings.

VI. RESULT

This paper compared YOLO-family models with YOLO-Former, which incorporates a Transformer-based prediction module, to accurately identify the driver's eye state (open/closed) from driver-state videos captured in real-world driving environments. The AI-Hub driver-state dataset used in the study includes diverse conditions such as mask and glasses wearing, providing an appropriate setting for evaluating the generalization performance of small-object detection models. Training stability was ensured through data preprocessing, including label format unification, directory reorganization, image normalization, and format conversion. Under identical conditions, YOLOv5n, YOLOv7n, YOLOv8n, and YOLO-Former were compared. Experimental results show that YOLO-Former with an integrated Transformer module demonstrated superior mAP performance for the small eye region and exhibited enhanced stability compared to YOLOv5n, YOLOv7n, and YOLOv8n across environments involving mask and glasses wearing.

These results suggest that leveraging global contextual information through self-attention can be effective for detecting small objects. In summary, this study presents a foundational model capable of reliably detecting and classifying eye-open/closed states in real-world driver monitoring environments. It demonstrates that combining YOLO-based lightweight models with Transformer-based architectures can enhance small-object detection performance, thereby contributing to improved detection accuracy.

However, our study has the following limitations. First, since eye opening/closing detection in the study relies on single-frame-based object detection results, it cannot directly model temporal changes across consecutive frames (e.g., blinking, prolonged eye closure). Second, the focus was on comparing eye opening/closing detection performance, and no final judgment on drowsiness status was performed. Third, separate performance analysis for lighting changes and night driving conditions was not included. Future research plans include establishing a quantitative judgment system that distinguishes between momentary blinks and drowsiness-related eye closures by accumulating and converting frame-level predictions into time-series metrics. This will be expanded to include environmental robustness evaluation under various illumination conditions and a final drowsiness judgment model.

REFERENCES

[1]. Jinmo Yang, Janghwan Kim, R. Young Chul Kim, Kidu Kim, "Learning Model for Avoiding Drowsy Driving with MoveNet and Dense Neural Network," The International Journal of Internet, Broadcasting and Communication, (2023), Vol.15, No.4, pp.142-148.

[2]. Sooyoung Kwon, Heejung Kim, Gwang Suk Kim, Eunhee Cho, "Fatigue and Poor Sleep Are Associated with Driving Risk among Korean Occupational Drivers," Journal of Transport and Health, (2019), Vol.14, 100572.

[3]. Łukasz Dziuda, Paulina Baran, Piotr Zieliński, Krzysztof Murawski, Mariusz Dziwosz, Mariusz Krej, Marcin Piotrowski, Roman Stablewski, Andrzej Wojdas, Włodzimierz Strus, Henryk Gasiul, Marcin Kosobudzki, Alicja Bortkiewicz, "Evaluation of a Fatigue Detector Using Eye Closure-Associated Indicators Acquired from Truck Drivers in a Simulator Study," Sensors, (2021), Vol.21, No.19, 6449.

[4]. Ebenezer Essel, Fred Lacy, Fatema Albalooshi, Wael Elmedany, Yasser Ismail, "Drowsiness Detection in Drivers Using Facial Feature Analysis," Applied Sciences, (2025), Vol.15, No.1, 20.

[5]. Dina Salem, Mohamed Waleed, "Drowsiness Detection in Real-Time via Convolutional Neural Networks and Transfer Learning," *Journal of Engineering and Applied Science*, (2024), Vol.71, 122.

[6]. Genchao Liu, Kun Wu, Wei Lan, Yunjie Wu, "YOLO-FDCL: Improved YOLOv8 for Driver Fatigue Detection in Complex Lighting Conditions," *Sensors*, (2025), Vol.25, No.15, 4832.

[7]. Jimin Lee, Soomin Woo, Changjoo Moon, "3D-CNN Method for Drowsy Driving Detection Based on Driving Pattern Recognition," *Electronics*, (2024), Vol.13, No.17, 3388.

[8]. Ultralytics, "YOLOv5 architecture description," accessed on [10-9-2025]. [Online].

[9]. Wong Kin-Yiu, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," arXiv:2207.02696, 2022.

[10]. Chien-Yao Wang and Hong-Yuan Mark Liao, "ELAN: Efficient Layer Aggregation Networks," Technical Report, 2022.

© 2026, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

[11]. Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv:2004.10934, 2020.

[12]. M. Yaseen, "What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector," *arXiv preprint arXiv:2408.15857*, 2024.

[13]. Zhu, X., Lyu, S., Wang, X., & Zhao, Q. (2021). TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 2778-2788).

© 2026, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden