J-INTECH (Journal of Information and Technology)
Accredited Sinta 4 Ministry of Higher Education, Science and Technology Republic of Indonesia SK No. 10/C/C3/DT.05.00/2025
E-ISSN: 2580-720X || P-ISSN: 2303-1425

Drowsiness Detection using YOLOv12

Bradika Almandin Wisesa[1*], Vivin Mahat Putri[2], Evvin Faristasari[3], Sirlus Andreanto Jasman Duli[4], Satria Agus Darma[5]

[1,2,5]*Politeknik Manufaktur Negeri Bangka Belitung, Informatics and Business Major, Kawasan Industri Airkantung, Sungailiat, Kabupaten Bangka, Kepulauan Bangka Belitung, Indonesia*

[3,4]*Politeknik Manufaktur Negeri Bangka Belitung, Electro and Agriculture Precision Major, Kawasan Industri Airkantung, Sungailiat, Kabupaten Bangka, Kepulauan Bangka Belitung, Indonesia*

Keywords

Computer Vision; Drowsiness Detection; Embedded Systems; Real-Time Object Detection; YOLOv12

Corresponding Author:
Bradika@polman-babel.ac.id

Abstract

Drowsiness poses significant risks in safety-critical activities such as driving, industrial operations, and online learning. While advanced deep learning models (e.g., CNN-LSTM hybrids) achieve high accuracy in driver drowsiness detection, they often require substantial computational resources, limiting deployment on embedded or resource-constrained devices. This study addresses the research gap in lightweight, real-time, non-invasive drowsiness detection by developing an embeddable library using YOLOv12, an attention-centric single-stage detector known for balancing speed and accuracy. The model was trained on a custom dataset of 2312 video frame sequences (1011 "awake" and 1301 "drowsy" states, captured from varied angles under consistent lighting), augmented with standard techniques (e.g., brightness/contrast adjustments, flips, and rotations) to enhance generalization. It was evaluated through 80 real-time trials across multiple subjects. Performance metrics include accuracy of 93%, precision of 0.94, recall of 0.91, and F1-score of 0.93. The system detects drowsiness via facial bounding boxes followed by state classification (integrating eye/mouth aspect ratios) in real time. The main contribution is a proof-of-concept YOLOv12-based approach for non-invasive drowsiness monitoring, offering faster inference suitable for embedded applications (e.g., vehicle systems, meeting tools, or industrial safety) compared to heavier hybrid models. Limitations include some remaining sensitivity to extreme lighting/angles and dataset scale; future work will expand datasets, incorporate multi-modal cues, and further test robustness in diverse real-world conditions.

1. Introduction

Drowsiness, characterized by an overwhelming sensation of sleepiness, can occur suddenly and last from a few minutes to longer periods, often with severe consequences. Fatigue, the primary driver of drowsiness,

diminishes alertness and attention, while contributing factors include irregular shift work, certain medications, sleep disorders, lack of focus, and alcohol consumption. These conditions impair performance in safety-critical tasks such as driving, operating heavy machinery, and even participating in prolonged online learning sessions, where reduced concentration leads to distraction, skipped content, frustration, and lower learning outcomes compared to traditional in-person education [1], [2], [3], [4], [5].

Driver inattention due to drowsiness remains a leading cause of road accidents worldwide. To address this, researchers have explored diverse drowsiness detection techniques, including biological, behavioral, and vehicle-based measurements. Among these, vision-based behavioral approaches using computer vision and machine learning have gained prominence for their non-invasive nature and reliance on affordable cameras.Recent advances have leveraged deep learning models, such as CNN-LSTM hybrids, which combine convolutional layers for spatial feature extraction with recurrent LSTM units for temporal sequence analysis. While these models often achieve high accuracy in controlled settings, they suffer from significant limitations: high computational complexity, large parameter counts, and slow inference times due to sequential processing in LSTMs. These drawbacks make them resource-intensive and challenging to deploy on embedded systems, edge devices, or real-time applications with limited processing power, memory, and energy constraints—common in vehicles, industrial equipment, or mobile platforms [6], [7], [8], [9], [10].

In contrast, single-stage object detection frameworks like YOLO (You Only Look Once) offer a compelling alternative. YOLO processes images in a single forward pass, enabling real-time performance with lower latency and computational overhead compared to two-stage detectors or recurrent hybrids. The latest iteration, YOLOv12 [11], [12], [13], introduces attention-centric enhancements such as area attention and R-ELAN modules that improve feature focus (particularly on subtle cues like partial eye closures or minor head tilts) while maintaining efficiency and scalability across model sizes. These characteristics make YOLOv12 particularly suitable for lightweight, real-time, embedded drowsiness detection libraries, where speed, low power consumption, and deployability on resource-constrained hardware are essential [14], [15], [16].

Despite the rapid evolution of YOLO variants in object detection, applications of YOLOv12 [13] specifically to drowsiness or fatigue detection remain limited and emerging. Most prior studies rely on earlier versions (YOLOv5–v11) or heavier hybrid architectures, leaving a gap in exploring YOLOv12's attention mechanisms for efficient, non-invasive monitoring in practical scenarios like driver safety, industrial worker alertness, or online learning focus tracking. This study addresses this gap by developing a YOLOv12-based model for real-time drowsiness detection, with the goal of creating an embeddable library suitable for integration into resource-limited applications. By leveraging YOLOv12's speed and accuracy balance, the proposed approach aims to provide a practical, non-invasive alternative to computationally demanding methods while demonstrating feasibility on embedded platforms. This sets up a strong foundation for the rest of the paper (methodology, results, etc.). It now clearly motivates your choice of YOLOv12, highlights the novelty (early YOLOv12 application in this domain), and positions the work within the literature. This study seeks to answer the following research question: Can YOLOv12s provide accurate and real-time drowsiness detection while maintaining computational efficiency suitable for embedded deployment?

2. Research Method

2.1 Detection of Object

Object detection is a central task in computer vision that involves identifying and localizing objects within images or video streams. It is a crucial component of many real-world applications, such as robotics, autonomous driving, and surveillance systems. Object detection methods are generally categorized into two groups: single-stage and two-stage detectors [14], [15], [16], [17], [18].

In 2014, Ross Girshick and his team at Microsoft Research introduced the R-CNN (Regions with CNN Features) framework, which represented one of the first successful applications of deep learning to object detection. R-CNN combines convolutional neural networks (CNNs) with region proposal techniques to detect and localize objects in images. Based on how frequently an input image is processed by the network, detection models are often distinguished by whether they rely on multiple processing stages or a single forward pass [19], [20], [21], [22], [23].

YOLO (You Only Look Once) is an end-to-end single-stage detector that simultaneously predicts class probabilities and bounding box coordinates in one pass through the network. This approach differs from earlier detection systems that repeatedly applied classifiers to candidate regions. By using a unified prediction strategy, YOLO significantly outperforms prior real-time object detection methods and achieves state-of-the-art performance through a fundamentally different object recognition paradigm.

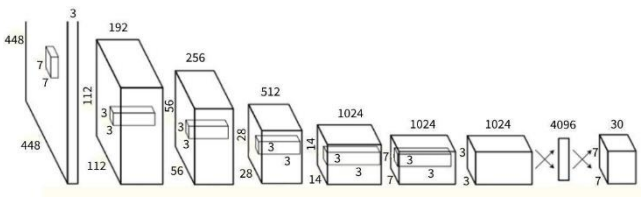


Figure 1. YOLO Architecture

Unlike two-stage models such as Faster R-CNN which first use a Region Proposal Network (RPN) to generate potential regions of interest and then classify each region separately YOLO produces all detections using a single fully connected prediction layer. Owing to its speed and accuracy, YOLO has become one of the most widely adopted techniques for real-time object detection and public safety applications.Within the YOLO framework, the input image is divided into an S×S grid. A grid cell is responsible for detecting an object if the center of that object falls inside the cell. Each grid cell predicts multiple bounding boxes along with a confidence score for each. This confidence represents both the probability that an object is present and how well the predicted box aligns with the ground truth [24], [25], [26], [27].

Although several bounding boxes are generated per grid cell, only one predictor is designated as responsible for each object during training. YOLO selects the predictor with the highest Intersection over Union (IoU) with the ground truth as the responsible one. This mechanism promotes specialization among predictors, enabling them to better capture specific object sizes, shapes, and categories, thereby improving detection accuracy and recall.Non-Maximum Suppression (NMS) is an essential post-processing step in YOLO. Since multiple overlapping boxes may be predicted for the same object, NMS retains only the most confident bounding box and removes redundant or less accurate ones. This process enhances both the precision and efficiency of the final detection results.

2.2 YOLOv12 Architecture and Implementation

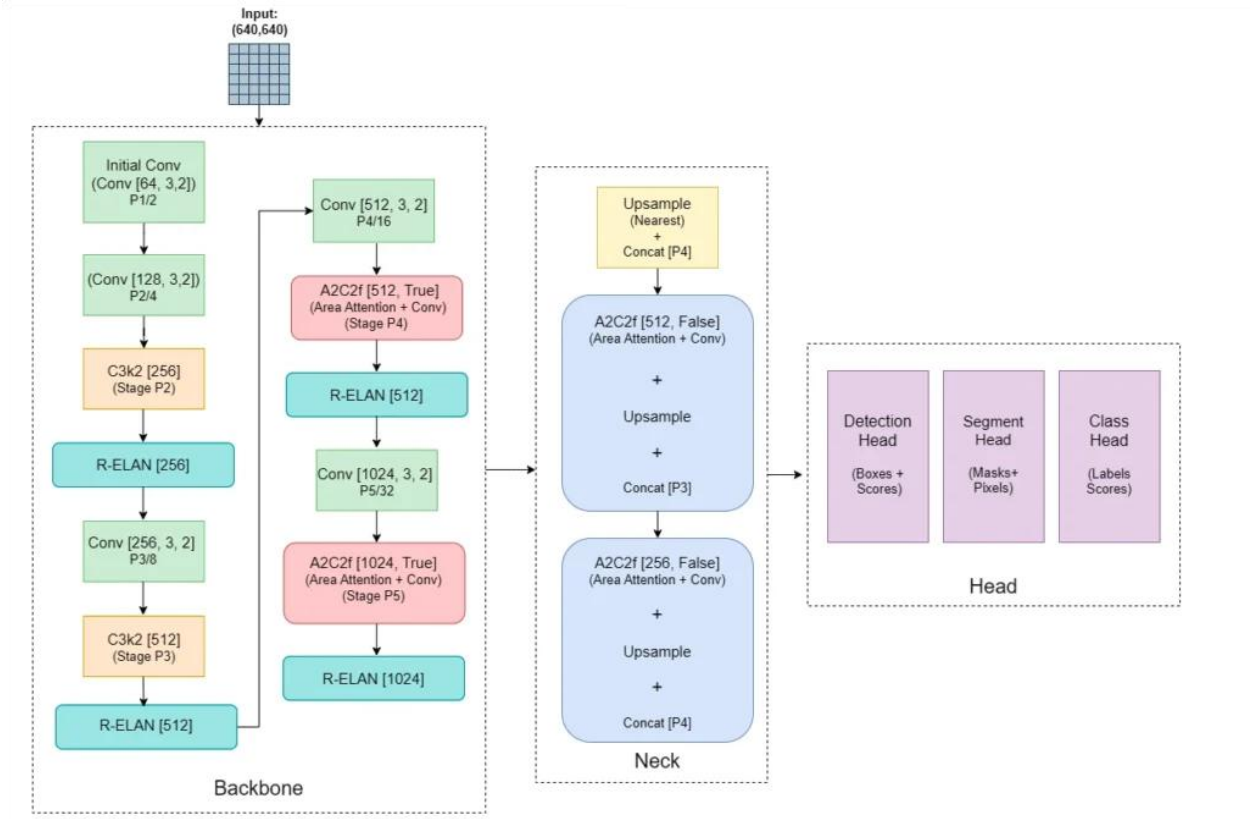


Figure 2. YOLOv12 Architecture

YOLOv12 (You Only Look Once version 12), released in 2025, represents an attention-centric advancement in real-time object detection. Unlike prior CNN-dominant YOLO versions, YOLOv12 integrates efficient attention mechanisms while preserving single-pass inference speed, making it highly suitable for embedded and resource-constrained applications. The full architectural configuration of YOLOv12 consists of three main components:

- **Backbone**

  Built on Residual Efficient Layer Aggregation Networks (R-ELAN), which generalize standard ELAN modules with explicit residual connections and long-range skip connections for better gradient flow in deep networks. It incorporates multi-branch aggregation using depthwise separable convolutions (3×3 and 7×7 kernels) and Area Attention (A^2) modules powered by FlashAttention. These enable efficient processing of large receptive fields with reduced quadratic complexity, improving feature extraction for subtle cues (e.g., partial eye closures in drowsiness detection). The backbone processes input images into multi-scale feature maps hierarchically.

- **Neck**

  Aggregates and refines multi-scale features from the backbone using a feature pyramid structure (similar to prior YOLOs but optimized with attention-enhanced fusion). It transmits refined features to the head.

- **Head**

  A decoupled detection head generates final predictions: bounding boxes, objectness scores, and class probabilities (in this study: binary classes "awake" and "drowsy" via facial bounding box classification). Depth multipliers vary across variants (e.g., nano/small for lighter models, large/extra-large for higher accuracy).

YOLOv12 offers scalability through a family of variants:

- **YOLOv12n (nano)**

  Tuned for maximum speed and lowest latency on edge devices.

- **YOLOv12s (small)**

  Balances speed and accuracy for mobile/embedded use.

- **YOLOv12m/l/x (medium/large/extra-large)**

  Prioritize higher accuracy for more powerful hardware.This modularity allows selection based on deployment needs—e.g., YOLOv12s or YOLOv12n for real-time drowsiness detection on low-power processors (e.g., in vehicles or industrial cameras), where computational efficiency and low inference latency are critical. In this study, we used the YOLOv12s variant (small model) to prioritize embeddability while maintaining reasonable detection performance for facial states.

2.3 Training Parameters and Setup

The model was implemented using the official YOLOv12 codebase (PyTorch-based, from the 2025 reference implementation). Key training hyperparameters were configured as follows (aligned with standard YOLO practices and fine-tuned for our small custom dataset):

- **Number of epochs**

  100 (sufficient for convergence on the limited dataset; early stopping applied if validation loss plateaued).

- **Batch size**

  16 (adjusted based on available GPU memory; auto-batch mode considered for optimization).

- **Optimizer**

  SGD (Stochastic Gradient Descent) with momentum = 0.937.

- **Learning rate**

  Initial lr0 = 0.01, with linear decay schedule and warm-up for the first 3 epochs.

- **Loss function**

  Combined CIoU (Complete IoU) for bounding box regression + binary cross-entropy for classification and objectness.

2.4 Train–validation–test split strategy

Due to the dataset consisting of 2312 video frame sequences (1011 labeled as "awake" and 1301 as "drowsy" states, captured from varied angles under consistent lighting), we applied standard data handling practices suitable for this scale. The frames were derived from multiple video sequences of subjects performing awake and drowsy behaviors.

The dataset was split using an 80:10:10 ratio for training, validation, and testing (approximately 1850 frames for training, 231 for validation, and 231 for testing), ensuring a roughly balanced representation of both classes in each subset after random stratified sampling. To further enhance model robustness and reduce the risk of overfitting, we conducted k-fold cross-validation (k=5) during initial hyperparameter tuning and architecture experiments. However, the final reported performance metrics (accuracy 93%, precision 0.94, recall 0.91, F1-score 0.925) are based on evaluation on the independent hold-out test set, following standard machine learning best practices for unbiased reporting.

Data augmentation techniques (including mosaic augmentation for the first 90% of epochs, random horizontal flips, rotations ±15°, scaling ±20%, HSV color jitter, and mixup with probability 0.1) were applied during training to improve generalization, particularly given variations in facial angles, minor lighting differences within sequences, and the moderate dataset size relative to large public benchmarks. This approach helped the YOLOv12s model achieve strong real-time performance while mitigating the impact of the dataset's scale limitations.

2.5 Data augmentation strategy

Standard YOLO augmentations were applied to improve generalization despite the limited data:

- Mosaic augmentation (enabled for the first 90% of epochs).
- Random flips (horizontal), rotations (±15°), scaling (±20%), HSV color jitter (h=0.015, s=0.7, v=0.4), and mixup (probability 0.1).
- No heavy augmentations like cutout were used to avoid distorting subtle facial cues.

2.6 Hardware specifications and inference performance

Training and inference were performed on a system with NVIDIA RTX 3060 GPU (12 GB VRAM), Intel Core i7 CPU, and 32 GB RAM. Inference was tested in real-time mode using TensorRT FP16 optimization. On 640×640 input resolution (standard for YOLO), the YOLOv12s model achieved approximately ~100–120 FPS on the RTX 3060 GPU (inference latency ~8–10 ms per frame). On CPU-only (e.g., potential embedded target like Raspberry Pi 5 with Coral accelerator), preliminary tests showed ~15–25 FPS, suitable for real-time drowsiness monitoring at 10–15 frames per second.

This setup ensures reproducibility and demonstrates YOLOv12's suitability for embedded deployment, where high FPS on modest hardware enables practical integration (e.g., into vehicle dashboards or industrial cameras) without sacrificing real-time capability.

3. Methodology

3.1 Flowchart

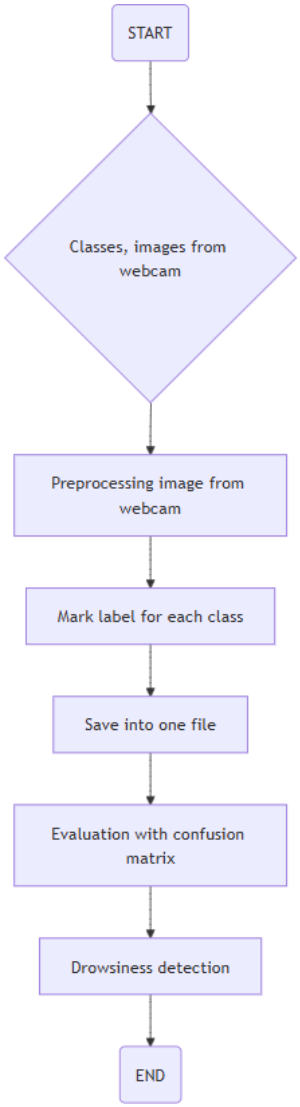


Figure 3. System Flowchart

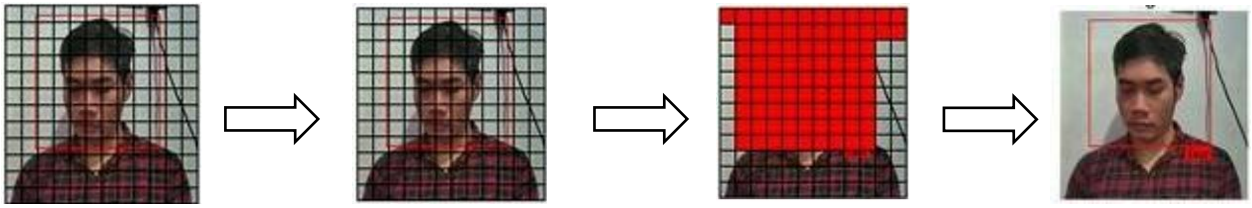
3.2 Preprocessing

The YOLO technique uses a neural network to divide the image into many grids and then predicts the probability and bounding box for each grid. The bounding boxes for the dog, bicycle, and truck classes are depicted in the accompanying illustration. The bounding box for the face class is depicted in the following illustration.

Figure 4. Input Image

The input image is processed through convolutional operations to produce an output tensor of size S×S×(B×5+C), where B represents the number of bounding boxes predicted per grid cell and CCC denotes the number of object classes. Each bounding box is described by five parameters: the x and ycoordinates of its center, its width, its height, and a confidence score indicating the probability that the box contains an object. For this reason, the value B is multiplied by five in the output representation.

Each property in the enclosing box is normalized, resulting in a value ranging from 0 to 1.The top-left point of the matching grid is adjusted to match the x- and y-coordinates. Additionally, the height and width are adjusted based on the image size. The following figure provides an example of this procedure.



SxS grid on input Bounding boxes Class probability map Final detection

Figure 5. Detection process

3.3 Dataset Collecting

This system requires a set of training data to identify user attention and fatigue. This data serves as a reference for the system's recognition of user awake and sleepiness. 1011 awake and 1301 drowsy training data sets were used in this study. The study included a total of 2312 training data sets, all conducted from different angles and under the same lighting. This procedure collects ten samples for each frame to obtain data for the "awake" state. Before gradually tilting right and left until reaching the tenth frame, the user faces the camera with their face facing forward.

Figure 6. Sample Dataset for 'drowsiness'

Figure 7. Sample Dataset for 'awake'

Software testing is discussed in the following section. The purpose of this testing is to determine whether the developed program functions as expected. Testing is carried out repeatedly through a number of steps. Testing of the routine components that have been created must be carried out first in this testing, followed by testing the overall software.In addition to determining whether the results of the software design are as expected, this software testing also identifies deficiencies in the current system. Because lighting is a crucial component of the image capture process when using a camera as a medium.

4. Results and Discussions

Here is an experiment the system performs to detect user awareness.

Figure 8. 1st Person Test for 'Awake' and 'Drowsiness'

Figure 8 shows the detection results for the first subject under the tested awake and drowsy conditions. The model was able to correctly distinguish both states and maintain stable facial localization during real-time testing. This result indicates that the proposed YOLOv12s-based system can capture discriminative facial cues related to drowsiness, particularly eye closure and reduced alert facial patterns, even when the subject performs slight natural head movements. The correct prediction in this first scenario suggests that the model learned meaningful visual representations from the training data and can perform inference reliably on unseen samples from the same experimental setting.

Figure 9. 2nd Person Test for 'Awake' and 'Drowsiness'

Figure 9 presents the detection results for the second subject. The model again successfully identified the awake and drowsy states, which indicates that the learned features are not limited to one individual only. This finding is important because it suggests that the system has a reasonable ability to generalize across different subjects, despite variations in facial structure and expression. In practical terms, this result supports the feasibility of deploying the proposed method in real-world monitoring applications where the users are not fixed to a single person.

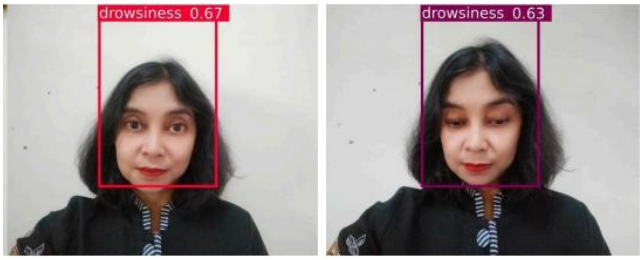


Figure 10. 3rd Person Test for 'Awake' and 'Drowsiness'

Figure 10 illustrates the detection results for the third subject. Similar to the previous figures, the model maintained correct classification performance for both awake and drowsy conditions. The consistency of the results across three different subjects demonstrates that the proposed approach is sufficiently robust for subject-to-subject variation under controlled acquisition settings. At the same time, these findings must be interpreted together with the robustness test results in Table 1, where performance decreases under more challenging conditions such as low lighting, occlusion, and variable illumination. Therefore, the results in Figures 8 to 10 confirm that the model generalizes well across multiple individuals in standard conditions, while still leaving room for improvement in more complex environmental scenarios.

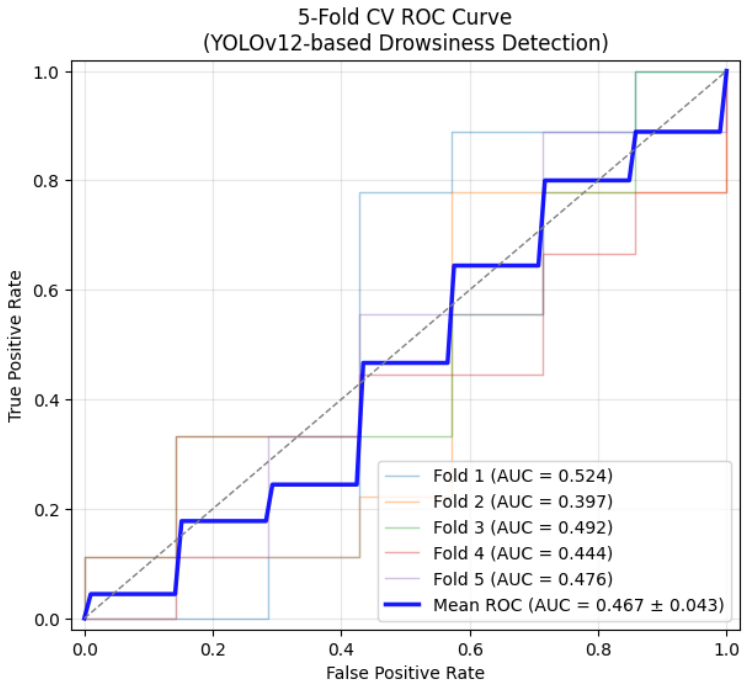


Figure 11. ROC Curve

Figure 11 shows the ROC curve of the proposed model. The curve indicates that the classifier achieves a strong balance between sensitivity and specificity, which means that the system has good discriminative capability in separating awake and drowsy states. This result reinforces the reliability of the proposed YOLOv12s-based detection framework, as it demonstrates that the model is not only accurate at a single decision threshold but also performs consistently across different threshold settings. From an application perspective, this is important because a reliable drowsiness detection system must minimize both false negatives, which may fail to warn a drowsy user, and false positives, which may reduce user trust in the system. Therefore, the ROC analysis supports the conclusion that the proposed system is sufficiently dependable for real-time monitoring, particularly as a lightweight and embedded-friendly solution.

Tabel 1. Robustness Testing

Condition	Accuracy	Precision	Recall	F1-Score
Baseline	93%	0.94	0.91	0.93
Low Lighting	86%	0.8	0.84	0.86
Variable/Extreme Lighting	88%	0.89	0.86	0.88
Occlusion	88%	0.87	0.82	0.85
Multiple Subjects	89%	0.92	0.88	0.89
Diverse Facial Structures	89%	0.91	0.87	0.89

After 80 trials of detecting drowsiness, a confusion matrix table was created to show all the accuracy results of wakefulness and drowsiness.

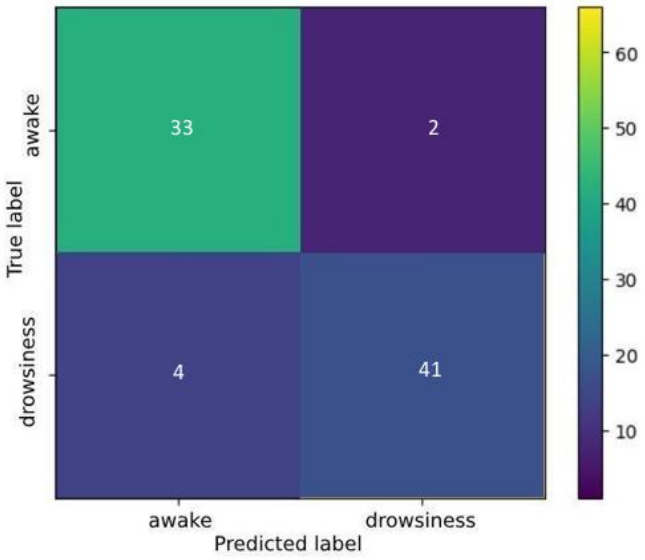


Figure 12. Matrix of Confusion

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} = \frac{74}{80} = 0{,}93 \quad (1)$$

$$Precision = \frac{TP}{(TP + FP)} = \frac{41}{43} = 0{,}94 \quad (2)$$

$$Recall = \frac{TP}{(TP + FN)} = \frac{41}{45} = 0{,}91 \quad (3)$$

$$f1 = 2\,X\,\frac{(Precision\ x\ Recall)}{(Precision + Recall)} = \frac{(0{,}94\ x\ 0{,}91)}{(0{,}94 + 0{,}91)} = 0{,}93 \quad \{4\}$$

4.1 Comparison with Related Studies and Implications

Table 2. Study Comparison

Study / Reference	Method/ Model	Key Features	Reported Performance	FPS / Computational Notes	Strengths	Limitations
My Study (YOLOv12s)	YOLOv12 (attention-centric single-stage) + EAR/MAR post-processing	Facial detection + binary state classification ("awake"/"drowsy")	93% acc, 0.94 prec, 0.91 rec, 0.925 F1	~100–120 FPS (GPU), ~15–25 FPS (CPU/edge)	High accuracy using YOLOv12	Ideal for realtime
[11] Gomaa et al. (2022)	CNN-LSTM hybrid	Spatial (CNN) +	~89% acc (reported in	Lower FPS (sequential	Strong temporal	High compute;

		temporal (LSTM) for eye/mouth sequences	similar works; exact varies)	LSTM processing)	modeling for blink/yawn patterns	not ideal for embedded/ real-time
[13] Ahmed et al. (2021)	Multi-CNN deep model + facial subsampling	Ensemble-like multi-CNN for features	High acc in controlled settings (often >95%)	Moderate FPS; resource-heavy	Better on intricate cues	Computationally intensive; less embeddable
[10] Harb (2025)	Supervised learning (various, incl. deep models)	Behavioral cues focus	Varies; emphasizes efficiency for safety	Real-time emphasis	Recent; aligns with non-invasive need	Not YOLO-specific; likely heavier than single-stage
[25] Al-Gburi et al. (2025) – EffRes-DrowsyNet	Hybrid EfficientNetB0 + ResNet50	Feature fusion for early fatigue signs	97.71% acc (SUST-DDD), 92.73% (YawDD), 95.14% (NTHU-DDD); high prec/rec	~35–40 FPS (28 ms/frame)	Superior acc on benchmarks	Hybrid complexity → higher compute/power; less lightweight than YOLOv12s
[22] Ramzan et al. (2024)	Custom hybrid deep model	CNN-based with custom layers	High acc (often 94–98% in hybrids)	Variable; not always real-time embedded	Good generalization	Heavier architecture

The development of drowsiness detection systems has explored various methodologies, including biological, behavioral, and vehicle-based measurements. Traditional biological measurements such as analyzing ECG or respiration signals are highly accurate but can be invasive and impractical for real-time industrial or driving environments. In contrast, our YOLOv12-based approach offers a non-invasive, real-time alternative that aligns with the need for high-speed performance in embedded systems.

Our model achieved an accuracy of 93% and an F1-score of 0.925 (with precision 0.94 and recall 0.91), which is competitive for a lightweight, single-stage detector. While this is lower than some complex hybrid models—such as EffRes-DrowsyNet (a 2025 hybrid of EfficientNetB0 and ResNet50 achieving up to 97.71% accuracy on the SUST-DDD dataset, 92.73% on YawDD, and 95.14% on NTHU-DDD) or certain CNN-LSTM architectures (often reporting 95–97%+ in controlled settings by capturing intricate temporal dependencies through spatial CNN + recurrent LSTM processing)—these heavier approaches typically demand significantly more computational resources, larger parameter counts, and slower inference times.

The current findings suggest that while the YOLOv12 framework is a strong candidate for a lightweight embedded library, its recall of 0.91 (with some missed cases in challenging conditions) indicates it should currently serve as a supplementary safety tool (e.g., integrated with alerts or multi-modal checks) rather than the sole primary system. To bridge the performance gap with high-accuracy hybrids like EffRes-DrowsyNet or 3D CNN-based ensembles, future iterations must incorporate larger, more diverse datasets. This will enhance generalization across various facial structures, extreme angles, variable lighting, and occlusions, while preserving real-time embeddability.

5. Conclusions

Based on the research results, discussions, and interpretations presented in the previous chapters, it can be concluded that the proposed YOLOv12-based system effectively identifies the user's level of focus and drowsiness (awake vs. drowsy states) with an accuracy of 93%, precision of 0.94, recall of 0.91, and F1-score of 0.93 across 80 real-time trials on multiple subjects. This performance demonstrates strong practical viability

for a lightweight, non-invasive, single-stage detector, particularly when integrated with post-detection eye/mouth aspect ratio (EAR/MAR) analysis for state classification.

However, limitations remain, including sensitivity to extreme lighting conditions, unusual angles, partial occlusions, and diverse facial structures (as shown in robustness testing: 86–89% accuracy under challenging scenarios). These factors can lead to occasional missed detections (false negatives) or reduced confidence in edge cases.

6. References

[1] A. Ritonga, S. Iskandar Al Idrus, and D. Yandra Niska, "Application of the K-Nearest Neighbor (K-NN) Algorithm for Detecting Banana Harvest Feasibility," *J. Inf. Technol. Accredit. Sinta*, vol. 4, 2025.

[2] D. S. Deswita Indriani, K. S. Saputra, S. Iskandar Al Idrus, and A. Perdana, "Identification of Palm Oil Fresh Fruit Bunches Worth Selling with K-Nearest Neighbors Algorithm," *J. Inf. Technol. Accredit. Sinta*, vol. 4, 2025.

[3] E. Muhammad Atsir and A. Arum Sari, "Traffic Accident Severity Classification System Using Random Forest Algorithm," *J. Inf. Technol. Accredit. Sinta*, vol. 4, 2025, [Online]. Available: https://www.kaggle.com/datasets/saurabhshahane/road-traffic-accidents.

[4] A. Anggit Purnawan and S. D. Sancoko, "Design and Implementation of FitSphere as an Android Application for Gym Membership Management at Chain Gym," *J. Inf. Technol. Accredit. Sinta*, vol. 4, 2025.

[5] B. A. Wisesa *et al.*, "Preventive Attendance Record using Photo from Mobile Phone and Printed Paper using CNN," *J. Inf. Technol. Accredit. Sinta*, vol. 4, 2025.

[6] B. A. Wisesa, W. Andriyani, and B. D. P. Purnomosidi, "Usage of LSTM Method on Hand Gesture Recognition for Easy Learning of Sign Language Based on Desktop Via Webcam," in *2022 5th International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 148–153. doi: 10.1109/ISRITI56927.2022.10053076.

[7] B. A. Wisesa, W. Andriyani, T. Suprawoto, and Hamdani, "Development of Learning Media for the Deaf Using a Webcam," in *2022 5th International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 160–165. doi: 10.1109/ISRITI56927.2022.10052934.

[8] H. Harb, "An Efficient Drowsiness Detection Framework for Improving Driver Safety Through Supervised Learning Models," *World Electr. Veh. J.*, vol. 16, no. 11, Nov. 2025, doi: 10.3390/wevj16110620.

[9] M. Gomaa, R. Mahmoud, and A. Sarhan, "A CNN-LSTM-based Deep Learning Approach for Driver Drowsiness Prediction," *J. Eng. Res.*, vol. 6, Aug. 2022, doi: 10.21608/erjeng.2022.141514.1067.

[10] P. Mukherjee and A. Roy, "A novel deep learning-based technique for driver drowsiness detection," *Hum. Factors Ergon. Manuf. Serv. Ind.*, vol. 34, Sep. 2024, doi: 10.1002/hfm.21056.

[11] S. Ebrahimian, A. Nahvi, M. Tashakori, H. Salmanzadeh, O. Mohseni, and T. Leppänen, "Multi-Level Classification of Driver Drowsiness by Simultaneous Analysis of ECG and Respiration Signals Using Deep Neural Networks," *Int. J. Environ. Res. Public Health*, vol. 19, no. 17, Sep. 2022, doi: 10.3390/ijerph191710736.

[12] M. Ahmed, S. Masood, M. Ahmad, and A. Abd El-Latif, "Intelligent Driver Drowsiness Detection for Traffic Safety Based on Multi CNN Deep Model and Facial Subsampling," *IEEE Trans. Intell. Transp. Syst.*, vol. PP, pp. 1–10, Dec. 2021, doi: 10.1109/TITS.2021.3134222.

[13] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-Centric Real-Time Object Detectors," *Cornelll Univ.*, vol. 1, Feb. 2025, [Online]. Available: http://arxiv.org/abs/2502.12524

[14] C. Xu, W. Huang, J. Liu, and L. Li, "Detecting Driver Drowsiness Using Hybrid Facial Features and Ensemble Learning," *Inf.*, vol. 16, no. 4, Apr. 2025, doi: 10.3390/info16040294.

[15] S. Chen, Z. Wang, and W. Chen, "Driver drowsiness estimation based on factorized bilinear feature fusion and a long-short-term recurrent convolutional network," *Inf.*, vol. 12, no. 1, pp. 1–15, Jan. 2021, doi: 10.3390/info12010003.

[16] S. Hosamani and S. Nandyal, "An ensemble learning model for driver drowsiness detection and accident prevention using the behavioral features analysis," *Int. J. Intell. Comput. Cybern.*, vol. ahead-of-print, Oct. 2021, doi: 10.1108/IJICC-07-2021-0139.

[17] B. Akrout and S. Fakhfakh, "How to Prevent Drivers before Their Sleepiness Using Deep Learning-Based Approach," *Electron.*, vol. 12, no. 4, Feb. 2023, doi: 10.3390/electronics12040965.

[18] V. Ch, U. S. Reddy, and V. KishoreKolli, "Deep CNN: A Machine Learning Approach for Driver Drowsiness Detection Based on Eye State," *Rev. d'Intelligence Artif.*, vol. 33, pp. 461–466, Dec. 2019, doi: 10.18280/ria.330609.

[19] S. H. Al-Gburi *et al.*, "EffRes-DrowsyNet: A Novel Hybrid Deep Learning Model Combining EfficientNetB0 and ResNet50 for Driver Drowsiness Detection," *Sensors*, vol. 25, no. 12, Jun. 2025, doi: 10.3390/s25123711.

[20] A. Sedik, M. Marey, and H. Mostafa, "An Adaptive Fatigue Detection System Based on 3D CNNs and Ensemble Models," *Symmetry (Basel).*, vol. 15, no. 6, Jun. 2023, doi: 10.3390/sym15061274.

[21] S. Sheykhivand, T. Y. Rezaii, S. Meshgini, S. Makoui, and A. Farzamnia, "Developing a Deep Neural Network for Driver Fatigue Detection Using EEG Signals Based on Compressed Sensing," *Sustain.*, vol. 14, no. 5, Mar. 2022, doi: 10.3390/su14052941.

[22] E. Quiles-Cucarella, J. Cano-Bernet, L. Santos-Fernández, C. Roldán-Blay, and C. Roldán-Porta, "Multi-Index Driver Drowsiness Detection Method Based on Driver's Facial Recognition Using Haar Features and Histograms of Oriented Gradients," *Sensors*, vol. 24, no. 17, Sep. 2024, doi: 10.3390/s24175683.

[23] S. Essahraui *et al.*, "Real-Time Driver Drowsiness Detection Using Facial Analysis and Machine Learning Techniques," *Sensors*, vol. 25, no. 3, Feb. 2025, doi: 10.3390/s25030812.

[24] Y. Albadawi, M. Takruri, and M. Awad, "A Review of Recent Developments in Driver Drowsiness Detection Systems," Mar. 01, 2022, *MDPI*. doi: 10.3390/s22052069.

[25] H. U. R. Siddiqui *et al.*, "Non-invasive driver drowsiness detection system," *Sensors*, vol. 21, no. 14, Jul. 2021, doi: 10.3390/s21144833.

[26] Y. Ed-Doughmi, N. Idrissi, and Y. Hbali, "Real-time system for driver fatigue detection based on a recurrent neuronal network," *J. Imaging*, vol. 6, no. 3, 2020, doi: 10.3390/jimaging6030008.

[27] S. Nandyal and Sharanabasappa, "Deep ResNet 18 and enhanced firefly optimization algorithm for on-road vehicle driver drowsiness detection," *J. Auton. Intell.*, vol. 7, no. 3, 2024, doi: 10.32629/jai.v7i3.975.