ITC 2/55
Information Technology and Control
Vol. 55 / No. 2/ 2026
pp. 858-879
DOI 10.5755/j01.itc.55.2.43457

Driver Behavior Detection Method Based on Improved YOLOv8

Received 2025/10/29 | Accepted after revision 2026/04/22

HOW TO CITE: Sui, G., Song, M., Sun, H., Sha, J., Zhang, M., Yang, Q., Xu, Z. (2026) Driver Behavior Detection Method Based on Improved YOLOv8. *Information Technology and Control*, 55(2), 858-879. https://doi.org/10.5755/j01.itc.55.2.43457

Driver Behavior Detection Method Based on Improved YOLOv8

Guozhu Sui*, Meixia Song, Haiyun Sun, Junlin Sha, Mingzhen Zhang, Qingyi Yang, Zhao Xu

School of Traffic and Electrical Engineering, Dalian University of Science and Technology, Dalian, 116052, China

Corresponding author: GuozhugzSui@outlook.com

As a core interaction in the human-vehicle-road system, automated and refined detection of driving behavior has emerged as a crucial research direction in intelligent transportation systems and advanced driver assistance systems. Traditional post-event monitoring models that rely on manual or sensor-based methods are no longer able to meet the requirements of real-time and accurate risk identification. Therefore, this study proposes the You Only Look Once - Lightweight - BiFPN - ECA (YOLO-LBE) detection method. By integrating ghost convolution and GhostC2f modules to diminish computational complexity, the study employs a weighted bidirectional feature pyramid network, and further embed an ECA module to significantly enhance the precision and stability of driver behavior detection. Experimental findings demonstrate that the improved YOLOv8 model improves mAP@0.5 by 5.3%, FPS by 32.4%, Params by 34.4%, and FLOPs by 33.3% compared to YOLOv5s. This research method outperforms existing mainstream models in terms of accuracy, efficiency, and interference tolerance, providing reliable technical support for real-time driving behavior monitoring.

KEYWORDS: YOLOv8; ECA; Driver behavior detection; Phantom convolution; BiFPN

1. Introduction

Recently, road traffic safety has become a focus of social attention [1]. In actual traffic scenarios, dangerous DB is often a precursor to traffic accidents. As a pivotal component in the human-vehicle-road system, the safety, standardization and real-time nature of DB directly affect the operating efficiency and public safety of the road traffic system [24]. Therefore, achieving accurate and real-time detection and recognition of driver DB has evolved into a crucial area of research within the realm of intelligent transportation and autonomous driving [23]. To solve the problem that existing fatigue driving detection is in-

effective in complex scenarios, Li et al. introduced a global behavior detection method that embeds spatial and channel reconstruction convolution, global attention mechanism and integrates cross-stage partial connection-fast spatial pyramid pooling module in YOLOv8. The findings revealed that the mAP50 of this method was improved by 1-42.7% [15]. Uddin et al. proposed a hybrid framework that integrates random forest and ResNet50/EfficientNetB6 [26]. The findings demonstrated that the accuracy of this framework was better than the existing technologies [26]. Gao et al. introduced a multi-scale (MS) spatio-temporal Transformer framework. The findings demonstrated that the framework had low inference cost on data [10]. To solve the problem that traditional DB monitoring systems are expensive and susceptible to environmental interference, Fan et al. introduced a lightweight solution based on forearm electromyography signals. The findings demonstrated that the average accuracy of this scheme after training with gated recurrent units reached 93.94% [8]. Du et al. [7] introduced a lightweight YOLOv5 improved model that integrates phantom convolution and bidirectional feature pyramid network. The findings demonstrated that the model had a mAP of 91.8%, which was better than the YOLO series of lightweight models [7]. To solve the problem that abnormal driving data is difficult to obtain and shallow models depend heavily on extensive annotations, J. Hu et al. introduced a deep feature learning framework based on stacked sparse autoencoders. Experiments revealed that the detection performance of this scheme was better than the existing methods on self-collected datasets, and it was the first time to achieve deep representation of abnormal DBs with autoencoders [12].

As an efficient target detection model, YOLO series algorithms have shown great potential in real-time behavior detection. A. Mujahid et al. proposed a lightweight gesture recognition model based on YOLO v3 and DarkNet-53 convolutional neural network (CNN) for gesture detection in complex environments. The results showed that the accuracy of the proposed model on the gesture tagging dataset reached 97.68% [21]. Wang et al. designed a multi fusion block using convolution and visual transformers to solve the problem of real-time dangerous driving behavior detection, and proposed a multi fusion YOLO lightweight object detection model [27]. The results indicate that the average accuracy (mAP) of the model is 91.4%, which is 6.4% higher than YOLOv5n [27]. Li et al. [16] optimized the structure of YOLOv7 and proposed a YOLO based driver eye state detection algorithm that integrates multi-scale features using Convolutional Block Attention Mechanism (CBAM). The results indicate that the algorithm achieved a mAP value of 99.0% on the CEW public dataset [16]. Zhang et al. [32] proposed a lightweight YOLOv8 solution that directly measures the head pitch angle in side view. The findings demonstrated that the frame rate of this method was increased to 37–53FPS, and mAP50 was increased by 0.4–0.5%. The actual vehicle verification could accurately assess fatigue [32]. Wei et al. [29] proposed a spatial channel collaborative attention YOLO network to improve the safety of autonomous driving, and integrated Ghost module to promote the lightweight characteristics of the network. The results indicate that the network has demonstrated certain effectiveness on public datasets [29]. Efficient Channel Attention (ECA) is a lightweight attention mechanism designed to boost the model's focus on crucial features. Debsi et al. [5] proposed a driver behavior detection model that combines a multi head self-attention module, ECA, and YOLOv8 to achieve real-time detection of fatigue driving behavior. And the effectiveness of the model in real-time scenarios was demonstrated through test results on self-checking datasets [5]. Fang et al. [9] proposed an improved Mask R-CNN combined with ResNeXt network to improve the accuracy of multi-target detection in complex traffic scenes, and added ECA to the backbone feature extraction network to optimize the advanced low resolution semantic information map. The results showed that the method achieved 62.62% mAP on the CityScapes autonomous driving dataset [9]. To address the need for accurate localization of coronary artery stenosis, Duan et al. proposed a deep residual U-Net fused with ECA. This method achieved an Intersection-over-Union (IoU) of 94.29% on the test set [6]. Postupaiev et al. [20] used the YOLOv8 model for accurate real-time operator instance segmentation to eliminate potential visual interference. The results indicated that the YOLOv8 model could achieve an accuracy of 95.5% and a recall rate of 92.7% in low capacity environments [20].

Existing research still faces limitations such as excessive model complexity and susceptibility to in-

terference from redundant background information. To handle these issues, this study proposes a DB detection method, You Only Look Once - Lightweight - BiFPN - ECA (YOLO-LBE), that combines an improved YOLOv8 and ECA. This study proposes a clear design assumption: in vehicle visual behavior detection, a model structure that simultaneously satisfies lightweight, MS perception, and channel selective attention can more effectively overcome core challenges such as limited computing resources, variable target scales, and background noise interference. To this end, this study systematically combines Ghost lightweight convolution, Bidirectional Feature Pyramid Network (BiFPN) MS fusion, and ECA channel attention mechanism. This combination is not a simple module stacking, its inherent logic lies in Ghost Convolution (GhostConv) achieving computational efficiency through low rank approximation, but may sacrifice high-frequency details and complex spatial dependencies. BiFPN reconstructs MS spatial semantics through weighted cross scale connections to compensate for structural information loss. ECA strengthens the channel features most relevant to behavior and suppresses background noise without dimensionality reduction. This study strictly defines driver behavior detection as an end-to-end visual object detection task. The model directly predicts the bounding box coordinates of driver behavior instances in the image and their corresponding behavior categories, and all evaluation metrics are calculated based on the detection paradigm. For some of the classification metrics used in the comparative experiments, such as accuracy and F1 score, they are calculated by extracting the highest confidence category label from the detection output, which is used to verify the performance of the model in multiple dimensions. However, the architecture and loss function of the model itself are completely designed for object detection tasks.

The innovation of this study lies in: (1) the design of a lightweight GhostC2f module that embeds GhostBottleneck into the C2f structure of YOLOv8, reducing the number of parameters while preserving gradient flow. (2) We have constructed a YOLO-LBE model structure for driver behavior detection, which combines GhostConv, BiFPN, and ECA modules to form a collaborative inference path of lightweight feature extraction, multi-scale weighted fusion, and channel attention refinement. (3) Introduce decoupling detection head and WIoU loss function to optimize the independent convergence of classification and regression tasks and the robustness of bounding box regression, respectively.

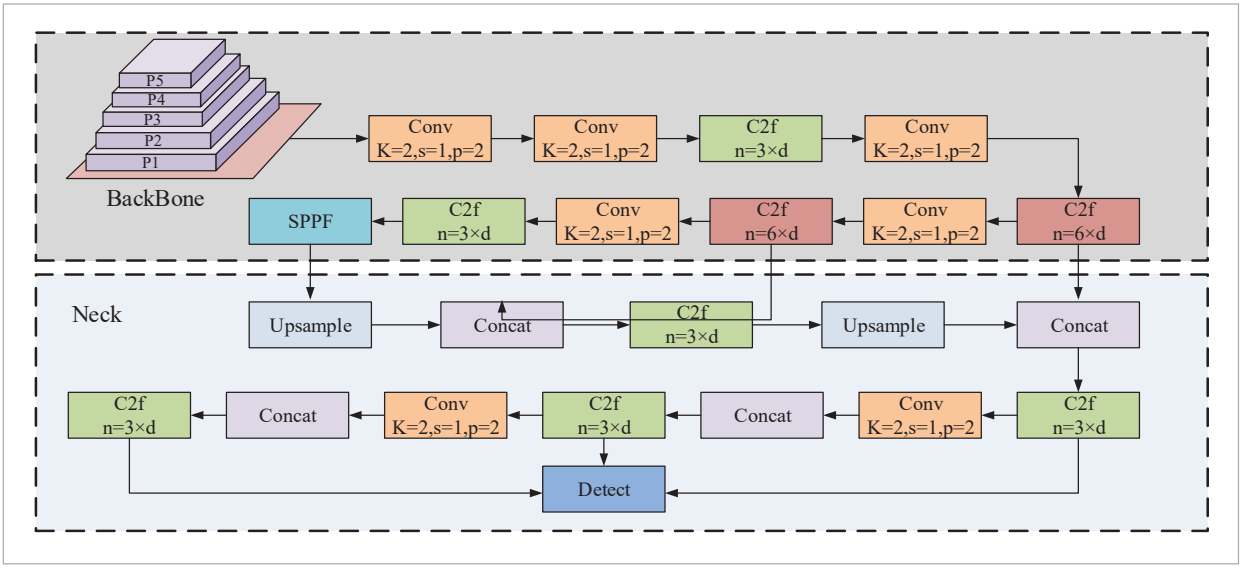
The main contribution of this study lies in: (1) revealing the functional synergy between GhostConv's lightweight feature extraction and ECA channel attention mechanism in driver behavior detection tasks. The former reduces computational redundancy, while the latter compensates for key detail responses that may be lost by the former. This discovery provides a new combination approach for balancing accuracy and efficiency in lightweight model design. (2) A lightweight detection head and loss function adaptation scheme (decoupling head+WIoU) has been constructed, providing a reference solution for the engineering challenge of improving the accuracy of fuzzy boundary sample localization under limited computing power conditions. (3) The detection stability of the model under multi-scale targets and fuzzy interference was evaluated on public datasets, demonstrating the potential of the proposed model for transfer to real-time monitoring scenarios in vehicles.

2. Methods and Materials

2.1. Construction of Driver DB Detection Model Based on Improved YOLOv8

In the task of detecting DB, the model must simultaneously meet the dual requirements of lightweight and high accuracy. On the one hand, the limited computing resources of on-board equipment require the model to have a low parameter count and computational complexity to achieve real-time inference and efficient deployment. On the other hand, DBs are diverse and the details of the movements are complex, so the model must have strong feature extraction and MS perception capabilities to accurately identify key distracting behaviors. Although the YOLOv8 model performs well in the field of general object detection, when directly applied to DB detection, it suffers from large parameter count, high computational complexity, and insufficient ability to extract MS behavior features [19, 3]. Therefore, this study proposes a Driver Action Detection Model Based on Improved YOLOv8 (YOLO-DAD). The YOLOv8 network structure is presented in Figure 1.

Figure 1
YOLOv8 structure.



In Figure 1, the backbone network sequentially transmits feature maps (FMs), extracts features at different levels through multiple convolution operations and modules, and finally fuses MS features through the Spatial Pyramid Pooling Fast (SPPF) module [17]. The neck part fuses and further processes the features of each scale through upsampling and feature concatenation operations, combined with the C2f module, and finally outputs FMs for detection tasks. Model lightweighting and key feature enhancement are the core directions for optimizing YOLOv8 in this study. In the selection of lightweight convolution modules, GhostConv analyzes the redundancy of FMs in the output of standard convolutions and generates "phantom" FMs using linear transformations, which can simulate the rich feature expression ability of standard convolutions with fewer parameters. Its structure is more concise than the depthwise separable convolution (DS-Conv) in the MobileNet series, and it is more suitable for achieving efficient real-time inference in vehicle environments with limited computing resources. Therefore, this study introduced GhostConv into the YOLOv8 network and replaced FPN and PAN with a weighted BiFPN to improve the model's feature fusion (FF) and perception capabilities for DBs at different scales, effectively coping with the problem of variable target scales [18]. The standard convolutional layer requires a large number of parameters to generate rich FMs, while the GhostConv module significantly reduces the computational burden without losing feature expression capabilities. Therefore, the study replaces the original standard convolution layer in YOLOv8n with the GhostConv module. Assuming the input FM is $X \in R^{C_{in} \times H \times W}$, GhostConv first uses some convolution kernels to generate the intrinsic FM, as shown in Equation (1).

$$Y' = X * f', f' \in R^{k \times k \times C_{in} \times \frac{C_{out}}{r}}. \tag{1}$$

In Equation (1), f' is the convolution kernel. * is the convolution operation. r represents the compression ratio. Y' is the intrinsic FM. C_{out} is the final desired number of output channels. Subsequently, Y' is transformed through a simple linear transformation to generate additional s Ghost FMs, as presented in Equation (2).

$$\begin{cases} y_{ij} = \Phi_{i,j}(y'_i) \\ \forall i = 1, \ldots, \dfrac{C_{out}}{r} \\ \forall j = 1, \ldots, s \end{cases}. \tag{2}$$

In Equation (2), $\Phi_{i,j}$ represents the j-th cheap linear transformation of the i-th channel. y'_i is the i-th channel of Y'. Finally, the intrinsic FM and the Ghost FM are spliced in the channel dimension to obtain the output FM, as shown in Equation (3).

$$Y = [y'_{11}, y'_{12}, \ldots, y'_{\frac{C_{out}}{r} s}]. \quad (3)$$

In Equation (3), y'_{ij} represents the single-channel FM after the i-th intrinsic channel undergoes the j-th cheap transformation. This process lowers the theoretical computational demands and the parameter count while maintaining the approximate feature expression capability. This process essentially simulates the output distribution of the standard convolution with fewer parameters by first compressing and then expanding. For the DB detection task, GhostConv can reduce model complexity and improve inference speed without losing the ability to perceive key action features such as "holding a mobile phone". The C2f module is a core component in YOLOv8 for enhancing gradient flow and feature reuse, but its internal stacked multiple Bottleneck structures have a high computational cost when the quantity of channels is high. To this end, the study proposed the GhostC2f module, which replaces the traditional Bottleneck with GhostBottleneck to achieve dual optimization of gradient information retention and computational efficiency. The structural comparison of the C2f and GhostC2f modules is presented in Figure 2.

Figure 2 compares the structures of the C2f and GhostC2f modules. Figure 2(a) shows the C2f module structure. After the input passes through Bottleneck, it passes through two CBS modules and FF before entering another Bottleneck. Meanwhile, some branches undergo Split and repeat Bottleneck fusion n times to enhance feature extraction. Figure 2(b) introduces the GhostBottleneck module, which divides the input features into multiple branches through the Split operation, performs lightweight feature extraction in parallel, and finally fuses them. This reduces the amount of computation and parameter requirements while maintaining sufficient capture of feature details, improving computational efficiency and parameter utilization. Assuming the input feature is X, the feature splitting process in the forward propagation of the GhostC2f module is presented in Equation (4).

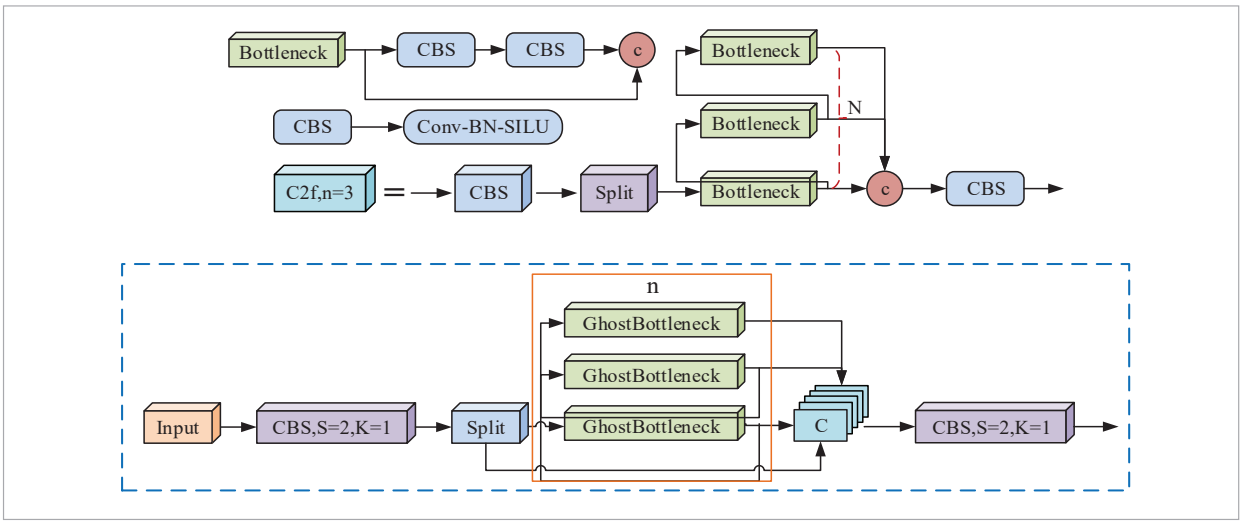
$$X_1, X_2 = \text{Split}(X). \quad (4)$$

In Equation (4), Split represents the average division along the channel dimension. X_1 and X_2 represent the two sub-FMs after the average division. This operation can reduce the single computational effort of the module and improve model efficiency. Furthermore, a lightweight transformation operation is used to superimpose n GhostBottlenecks only on the X_2 branch to generate more discriminative high-order features, as shown in Equation (5).

$$X_2 = \text{Stacked\_GhostBottlenecks}(X_2). \quad (5)$$

Figure 2

A structural comparison diagram of C2f and GhostC2f modules.



In Equation (5), Stacked_GhostBottlenecks represents the continuous stacking of multiple Ghost Bottleneck modules. This process allows the model to capture high-frequency details of subtle DBs while reducing the number of parameters by half. Finally, the transformed X_2 is concatenated with the original X_1 in the channel dimension, and the dual-branch information is integrated through convolution to obtain the final output, as presented in Equation (6).

$$Y = \mathrm{Conv}_{1\times1}(\mathrm{Concat}(X_1, X_2)). \tag{6}$$

In Equation (6), Concat represents channel-dimensional splicing. $\mathrm{Conv}_{1\times1}$ represents 1×1 convolution. GhostC2f uses the lighter GhostBottleneck to significantly minimize module parameter count and computational load while preserving comprehensive gradient details. The study replaced the original FPN+PAN structure with BiFPN. BiFPN employs adjustable weights to signify the significance of various input features and introduces cross-scale jump connections to achieve more efficient FF [25]. The fusion process is presented in Equation (7).

$$P_i^{out} = \mathrm{Conv}\left(\sum_j \frac{w_j \cdot P_j^{in}}{\epsilon + \sum_k w_k}\right). \tag{7}$$

In Equation (7), P_j^{in} is the input feature. w_j and w_k are the learnable weights corresponding to each input feature. represents a minimum value to prevent numerical instability. This mechanism enables the model to adaptively highlight the scale information most relevant to the current behavior when fusing features from different resolutions. A structural comparison between BiFPN and PAN is presented in Figure 3.

Figure 3 shows a structural comparison diagram of BiFPN and PAN. Figure 3(a) shows the PAN structure, which adopts a bottom-up and top-down single-path structure and realizes FF through simple upsampling and downsampling operations. It has a simple structure and is suitable for lightweight applications [4]. Figure 3(b) shows the BiFPN structure, which is optimized based on PAN and adopts multi-path parallel and fully connected network topology to enhance the diversity and interactivity of feature flow and improve the effect of FF, especially in complex scenes. The classification and regression tasks of the traditional YOLO detection head (DH) share the same network layer, which easily leads to feature conflicts between tasks, especially in tasks such as DB with few categories but fuzzy boundaries. The study uses a decoupled DH to separate classification and bounding box (BB) regression into two independent branches, optimize them separately, and improve convergence stability and detection accuracy. The decoupled DH structure is presented in Figure 4.

Figure 4

Decoupling DH structure.

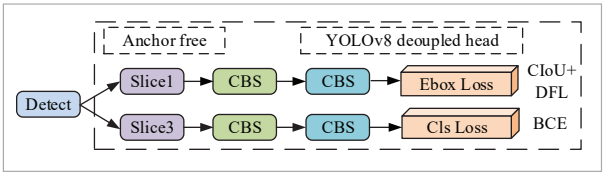


Figure 3

A structural comparison diagram of BiFPN and PAN.

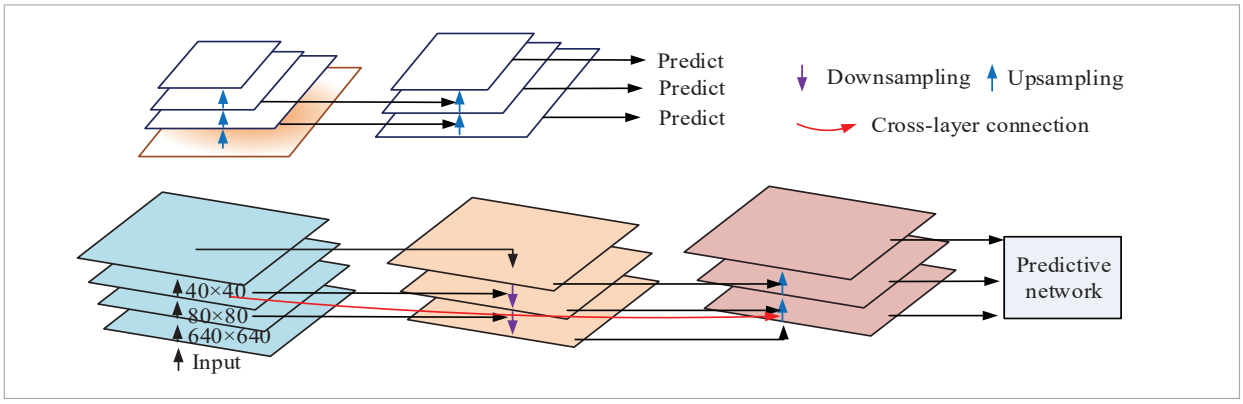


Figure 4 indicates the structure of the improved decoupled DH of YOLOv8. After the detection module outputs the FM, it is divided into two branches through a slicing operation. Branch 1 is connected to Ebox Loss after two CBS modules to predict the BB; branch 2 is connected to Cls Loss after two CBS modules to predict the target category. This structure achieves the decoupling of the detection task, allowing BB regression and category classification to be optimized independently, improving the flexibility and accuracy of the model. For the BB regression loss, Wise-IoU (WIoU) v3 is used. It effectively alleviates the interference of low-quality anchor boxes on training [31]. The WIoU loss is calculated as shown in Equation (8).

$$L_{WIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{W_g^2 + H_g^2} + \alpha v . \quad (8)$$

In Equation (8), ρ is the Euclidean distance. b and b^{gt} are the center points of the predicted box and the ground-truth box. W_g and H_g represent the width and height of the minimum bounding rectangle, respectively. α represents the weight function. v represents the consistency of the aspect ratio. represents IoU is the IoU between the predicted box and the ground-truth box. In the DB detection dataset, the proportion of medium quality anchor boxes generated by partial occlusion, sudden changes in lighting, etc. is relatively high, while simple positive samples and difficult negative samples are relatively few. The dynamic non-monotonic focusing mechanism of WIoU v3 evaluates and adjusts the attention weights for each anchor box in real-time through an IoU-based gradient gain term. For a large number of medium quality samples, this mechanism can adaptively allocate moderate gradient gains, avoiding simple suppression due to low IoU values and preventing excessive attention from damaging training stability. This enables the model to continuously optimize BB regression for such critical but fuzzy behavior instances, thereby improving the localization robustness of the model in actual complex driving environments. The overall LF of the model is presented in Equation (9).

$$L_{total} = \lambda_{cls} L_{cls} + \lambda_{obj} L_{obj} + \lambda_{box} L_{WIoU} \quad (9)$$

In Equation (9), L_{cls} is the BCE of the classification branch. L_{obj} is the BCE of the confidence branch. L_{WIoU} is the WIoU loss of the box regression branch. λ_{cls} is the weight of the classification loss. λ_{obj} is the weight of the confidence loss. λ_{box} is the weight of the WIoU loss. In summary, this study constructs the complete YOLO-DAD model, and its overall flow chart is presented in Figure 5.

Figure 5
Overall flowchart of YOLO-DAD.

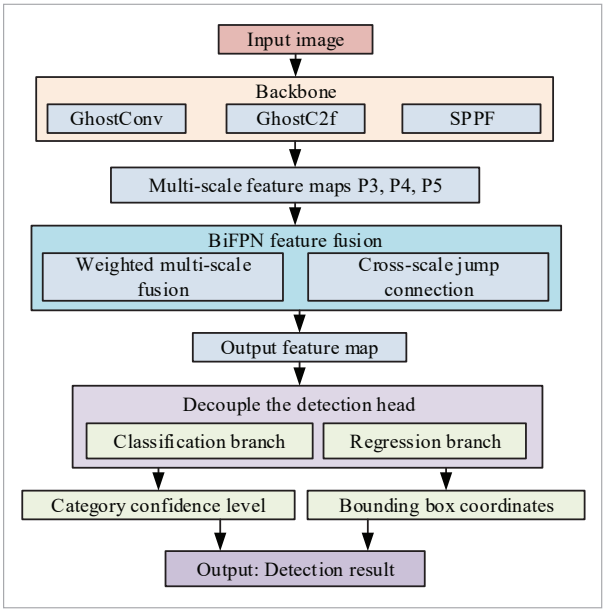


Figure 5 indicates the flow of the YOLO-DAD model. The initial step involves the input image being processed for feature extraction by the backbone network. The GhostConv module is employed to replace standard convolution to cunt down the number of parameters. The GhostC2f module is employed to optimize gradient flow and feature reuse, and MS features are finally fused via the SPPF module. Subsequently, the extracted MS FMs P3, P4, and P5 are fed into the neck network, which uses a weighted bidirectional feature pyramid network to replace the original FPN+PAN structure, achieving efficient FF through learnable weights and cross-scale skip connections. Finally, the fused FMs are fed into the decoupled DH, which separates the classification and BB regression tasks into two independent branches, outputting category confidence and BB coordinates, respectively. The training process is optimized using the WIoU LF, ultimately outputting the detection results.

2.2. Construction of DB Detection Model Based on YOLO-LBE

Although YOLO-DAD's lightweight design significantly reduces computational overhead, it may also lose some subtle features. Especially in the in-car environment, factors such as lighting changes, object interference, and camera angle offset can easily introduce background noise, affecting the judgment of the driver's true behavior. The ECA module directly captures cross channel interactions by avoiding one-dimensional convolution that reduces channel dimensionality. In theory, it can enhance the focus on key feature channels at extremely low computational overhead. Compared with composite attention mechanisms such as CBAM that include spatial and channel dimensions, its structure is lighter and more in line with the requirement of channel level feature filtering to suppress complex background noise in vehicles in this task. To this end, the study introduced the ECA module in the neck network to enhance the model's selective attention to behavior-related channel features at a very low computational cost. The ECA module is an extremely lightweight channel attention (CA) mechanism that avoids complex dimensionality reduction operations and efficiently captures cross-channel interactions through one-dimensional convolution (1D-Conv) [33, 11]. In DB detection tasks, channel information related to hand position, handheld objects, and facial orientation is crucial. These features are typically characterized by multiple channels and are sensitive to spatial context. The dimensionality reduction operation of modules such as SE will compress the nonlinear relationships between channels, which may weaken the ability to express the composite features mentioned above. ECA adopts a local cross-channel interaction strategy, where its adaptive 1D Conv kernel directly captures channel groups strongly correlated with key behavior areas within the same dimensional space. This method more fully preserves and enhances task-specific features, while simultaneously avoiding the risk of information loss caused by dimensionality reduction. As a result, ECA is more suitable for computationally sensitive real-time detection scenarios in vehicles. First, a global average pooling is performed on the given input features $U \in R^{C\times H\times W}$ to generate a channel statistics vector, as shown in Equation (10).

$$z_c = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} u_c(i,j)\ ,\ z \in R^C . \tag{10}$$

In Equation (10), $u_c(i,j)$ is the pixel value of the c-th channel at position (i,j). H and W are the height and width. C is the quantity of channels. Next, a one-dimensional convolution kernel size k with adaptive determination is used to capture local channel dependencies, and its calculation rule is shown in Equation (11).

$$k = \psi(C) = \left|\frac{log_2(C)}{\gamma} + \frac{b}{\gamma}\right|_{odd} . \tag{11}$$

In Equation (11), γ and b are hyperparameters set to 2 and 1, respectively, by standard convention. $|\cdot|_{odd}$ represents taking the nearest odd integer. The kernel size k is thus dynamically scaled with the channel dimension C. Afterwards, 1D-Conv and Sigmoid activation are performed to obtain the final CA weight, as shown in Equation (12).

$$\omega = \sigma(\mathrm{Conv1D_k}(z)) . \tag{12}$$

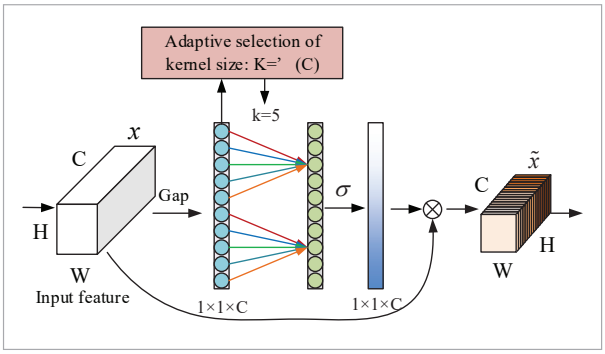
In Equation (12), $\mathrm{Conv1D_k}$ represents a 1D-Conv operation with the adaptively computed kernel size k. σ represents the Sigmoid function. Finally, the weights are multiplied by the original features channel by channel to obtain the refined feature output, as shown in Equation (13).

$$\tilde{U}_c = \omega_c \cdot U_c . \tag{13}$$

In Equation (13), ω_c represents the attention weight of the c-th channel. U_c is the two-dimensional FM of the c-th channel of the original feature. The ECA module avoids dimensionality reduction and directly models local correlations between channels. In DB detection, it can effectively enhance the response of key channels and suppress the influence of irrelevant regions. The structure of the ECA module is presented in Figure 6.

Figure 6 shows the ECA mechanism module. The input features are first compressed into a scalar per channel through a global average pooling module. Then, the convolution kernel size is adaptively selected through a learnable module. Next, the selected convolution ker-

Figure 6
ECA module structure diagram.



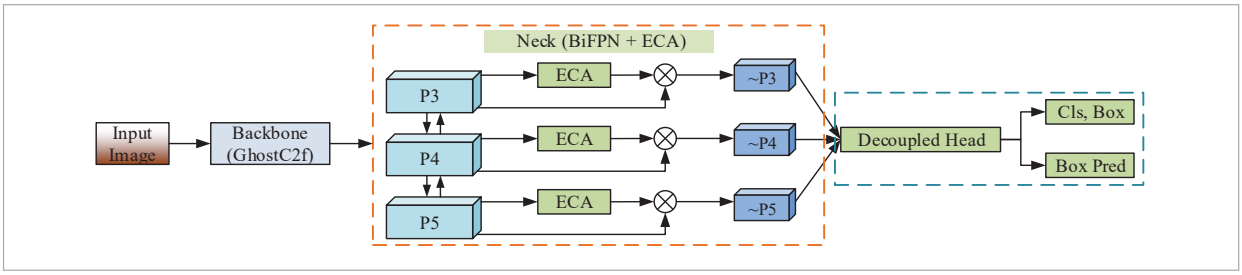
nel weights are generated through an activation function and used to perform weighted fusion of the input features. Ultimately, the size of the output features remains consistent with the original input features, but the expressiveness of the features is enhanced. This mechanism improves the model's capability to capture features and flexibility by dynamically modifying the convolution kernel size. To highlight key features related to DB in complex backgrounds, this study embeds the ECA module after each output feature layer of the neck network. Specifically, an ECA module is immediately connected after the BiFPN output FM P_i, and its output is presented in Equation (14).

$$P_i' = ECA(P_i) \otimes P_i . \tag{14}$$

In Equation (14), P_i represents the fused FM. $\otimes$ represents channel-by-channel multiplication. This design allows the model to immediately perform channel-level feature selection after completing MS FF, enhancing informative features and suppressing irrelevant features. The linear transformation used by GhostConv is essentially a low rank approximation, which may mainly lose two key features. One is high-frequency detail features (such as fingertip movements, small object edge textures), and the other is long-range spatial dependency features with complex nonlinear relationships. The loss of these features may weaken the model's ability to distinguish subtle or occluded DBs. The collaborative work of BiFPN and ECA modules can compensate for the aforementioned information loss. BiFPN, through its weighted MS fusion mechanism, can reconstruct more discriminative spatial structural information from features of different scales, alleviating the loss of long-range dependent features. Subsequently, the ECA module dynamically enhances the feature channels related to behavior through lightweight channel attention, while suppressing background noise, thereby strengthening the focus and utilization of high-frequency detail features. This design allows the model to maintain sensitivity to key behavioral features while being lightweight. A schematic diagram of the integrated location of the ECA module in the neck network is presented in Figure 7.

Figure 7 indicates the embedded position of the ECA module in the YOLO-LBE model. The MS features (P3, P4, P5) fused by the BiFPN network are sent to independent ECA modules for CA weight calculation. The weights are then multiplied channel by channel with the original features to obtain refined features (~P3, ~P4, ~P5), which are finally sent to the decoupled DH for prediction. This design ensures that after the model efficiently fuses MS features, it can further focus on the key channel information related to distracted DB. After integrating all components, the YOLO-LBE model has a clear multi-stage pipeline for its complete forward propagation process. Its lightweight feature extraction and MS FF operations are shown in Equation (15).

Figure 7
Schematic diagram of the integrated position of the ECA module in the neck network.



$$\begin{cases} F = \text{Backbone}_{\text{GhostC2f}}(I) \\ P = \text{Neck}_{\text{BiFPN}}(F) \end{cases}. \quad (15)$$

In Equation (15), I represents the input image. F represents the multi-level FM output by the backbone network. P represents the MS feature pyramid output by the BiFPN network. Through these two steps, the primary to high-level semantic features of the image can be efficiently captured, ensuring that the model has consistent perception capabilities for objects of different scales [22]. CA feature refinement and decoupled prediction are then performed, as shown in Equation (16).

$$\begin{cases} \tilde{P} = \text{ECA}_{\text{Block}}(P) \\ C_{pred}, B_{pred} = \text{Head}_{\text{Decoupled}}(\tilde{P}) \end{cases}. \quad (16)$$

In Equation (16), $\tilde{P}$ represents the refined features after attention weight modulation. B_{pred} represents the predicted BB coordinates. C_{pred} represents the predicted category confidence. CA feature refinement suppresses the channel response related to background noise in the FM and strengthens the channels related to the key features of distracted DB, thereby improving the discriminability of the features [28]. The decoupled design effectively alleviates the conflict between classification and regression tasks. Finally, end-to-end training is performed using the gradient descent algorithm to uniformly optimize all parameters, as shown in Equation (17) [30].

$$\Theta_{t+1} = \Theta_t - \eta \cdot \nabla_{\Theta} L_{total}(\Theta_t). \quad (17)$$

In Equation (17), η represents the learning rate. ∇_{Θ} represents the gradient operator. Θ_t represents the trainable parameters of the entire network at the t-th iteration. In summary, through the synergy of Ghost lightweight convolution, BiFPN MS fusion, and ECA mechanism, this model improves the detection precision and dependability of MS DBs in complex in-vehicle environments while preserving real-time operational efficiency. The overall flow chart of YOLO-LBE is presented in Figure 8.

Figure 8 indicates the flow of the YOLO-LBE model. The input image is first processed by a lightweight backbone network, which uses the GhostConv module to lower the computational burden and the GhostC2f module to enhance feature reuse. Finally, the SPPF module achieves MS FF. The extracted FM is then fed into the improved neck network, which uses a BiFPN architecture instead of the traditional FPN+PAN architecture. This architecture utilizes a learnable weight mechanism to achieve efficient cross-scale FF. An ECA module is embedded after each BiFPN output layer. The refined FM enters the decoupled DH. The WIoUv3 LF is employed to optimize the BB regression process, improving the precision and dependability of the model for detecting MS distracted DBs in complex in-vehicle environments. The pseudocode of the proposed YOLO-LBE model is shown in Table 1.

Figure 8
The overall flowchart of YOLO-LBE.

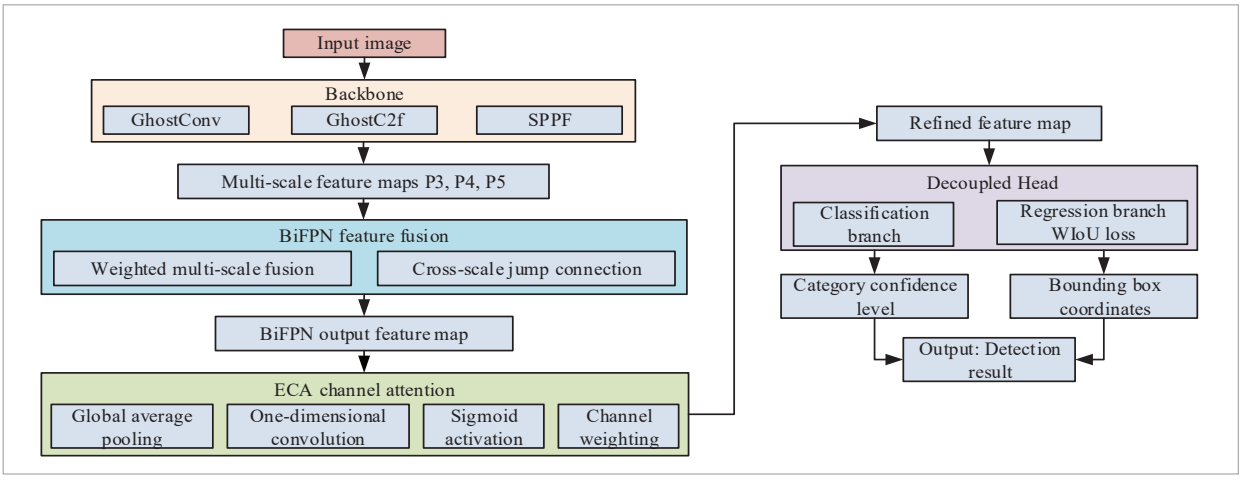


Table 1
Pseudo code of YOLO-LBE model.

Algorithm: YOLO-LBE (Driver Behavior Detection)

Input:
images // input image tensor of shape [B, C, H, W]
model // 'inference' or 'training'
ground_truth // (if mode='training') list of bounding boxes and class labels
Output:
if mode='inference':
boxes: predicted bounding box coordinates [B, N, 4]
scores: predicted class confidence scores [B, N, num_classes]
if mode='training':
loss: scalar total loss
BEGIN
// Step 1: Lightweight Feature Extraction (Backbone)
features_0 ← GhostConv(input_image)
features_1 ← GhostC2f(features_0)
P3, P4, P5 ← SPPF_and_Scaling(features_1) // Multi-scale feature maps
// Stepe 2: Multi-Scale Feature Fusion (BiFPN Neck)
fused_features ← BiFPN(P3, P4, P5) // Weighted cross-scale fusion
// Stepe 3: Channel Attention Refinement (ECA Module)
for each scale s in fused_features **do**
channel_stats ← global average pooling (fused_features) // [C]
kernel size k ← adaptive_kernel_size(C) // Equation (11)
attention_weights w ← Sigmoid(Conv1D_k(z)) // Equation (12)
refined_features ← channel-wise weighting // Equation (13)
end for
// Stepe 4: Decoupled Detection Head
for each refined feature map refined_features **do**
class_scores ← classification branch (refined_features)
bounding_boxes ← regression branch ((refined_features)
end for
// Stepe 5: Post-processing (Inference) or Loss Computation (Training)
if mode = 'inference' **then**
boxes, scores ← NMS_and_Decode(reg_box, cls_score)
return boxes, scores
else if mode = 'training' **then**
L_cls ← BinaryCrossEntropy(cls_score, ground_truth.cls)
L_obj ← BinaryCrossEntropy(obj_score, ground_truth.obj)
L_box ← WIoUv3(reg_box, ground_truth.box) // Equation (8)
total_loss // Equation (9)
// Back-propagation
Update all trainable parameters using gradient descent with learning rate // Equation (17)
return total_loss
end if
END

The input image size of the model is fixed at 640 × 640. The first layer of the backbone network adopts a standard convolutional layer (3 × 3, step size 2), followed by four GhostC2f modules connected in series. The number of stacked GhostBottlenecks is 3, 6, 6, and 3, respectively, and the output channel numbers are 64, 128, 256, and 512, respectively. Finally, the SPPF module outputs MS FMs P3, P4, and P5. The neck network uses weighted BiFPN for cross scale fusion of input features, and immediately embeds an ECA module. Its one-dimensional convolution kernel size is adaptively set to 3 to recalibrate the features of each channel output by BiFPN. The detection head adopts a decoupled design, with independent classification and BB regression branches, each containing two convolutional layers. The training adopts SGD optimizer with momentum of 0.937, weight decay of $5e^{-4}$, initial learning rate of 0.01, supplemented by cosine annealing scheduling, batch size of 32, and total training rounds of 300. In the loss function, the classification branch uses binary cross entropy, and the regression branch uses WIoU v3, with weights set to 0.5 and 0.05, respectively.

3. Results

3.1. DB Detection and Verification Based on the YOLO-DAD Model

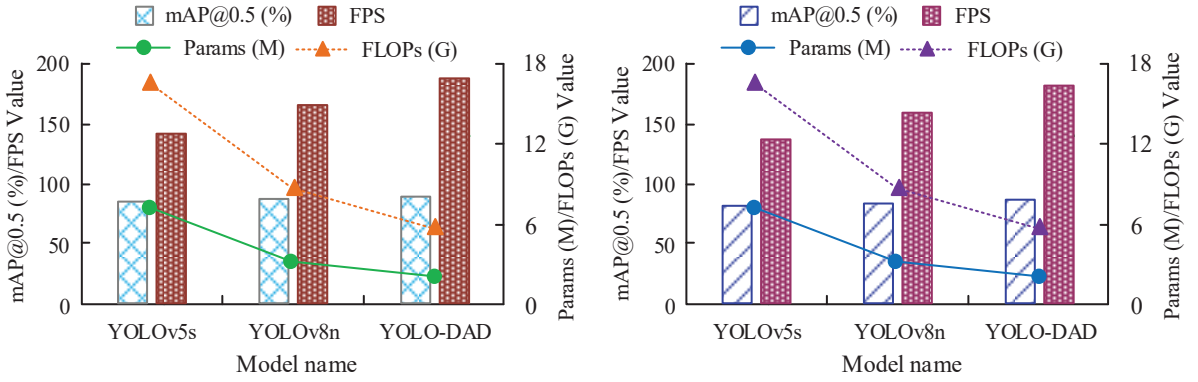
This study further conducted experiments to validate the performance of YOLO-DAD-based DB detection. The experimental environment configuration remained consistent to ensure the dependability of the experimental outcomes. The experimental environment refers to the hardware configuration and software parameters. Specific parameters are shown in Table 2.

The experimental environment was configured according to the parameters in Table 2. The study then introduced two datasets: the State Farm Distracted Driver Detection DatasetSF (SFD) and the AUC Distracted Driver Dataset (AUCD). SFD, released by State Farm Insurance, covers typical DBs, such as driving safely, texting, making calls, tuning the radio, sipping water, retrieving items from the back seat, grooming, and conversing with passengers. With over 22,000 training images, each sized at 640x480 pixels, it serves as an ideal benchmark

Table 2
Hardware and software configuration parameters table.

Category	Item	Specification / Version
Hardware	CPU	Intel Core i7-10700K
	GPU	NVIDIA GeForce RTX 3080 (10GB VRAM)
	RAM	16 GB DDR4 3200MHz
	Storage	256 GB SSD + 1 TB HDD
	Camera	1080p, 30 fps, autofocus
Software	Operating system	Ubuntu 20.04 LTS
	Deep learning framework	PyTorch 1.9.0 + CUDA 11.1
	Programming language	Python 3.8
	Data processing	OpenCV 4.5, NumPy 1.21, Pandas 1.3
	Model evaluation	Scikit-learn 1.0, TensorBoard 2.7
	Visualization	Matplotlib 3.5, Seaborn 0.11
	Annotation tool	LabelImg 1.8.6

Figure 9
Performance comparison of various models on datasets.



(a) Performance comparison on the SFD dataset

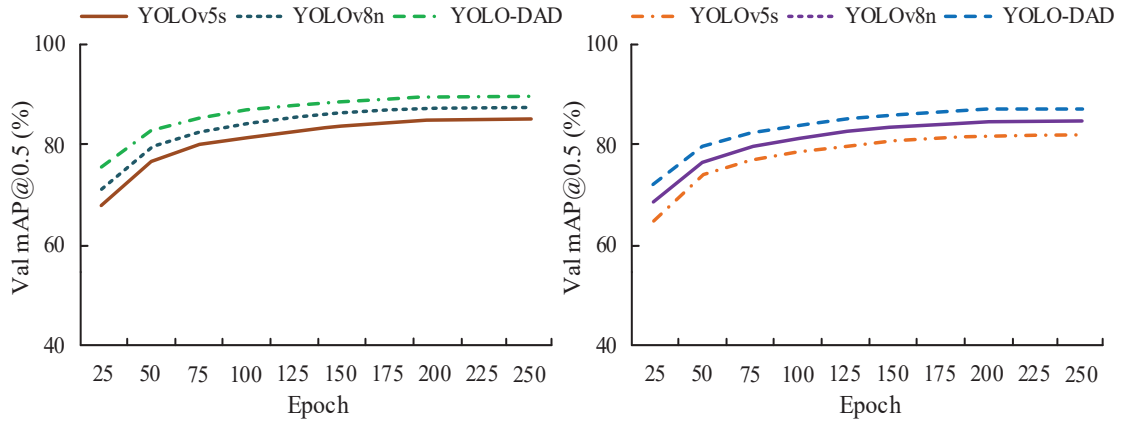
(b) Performance comparison on the AUCD dataset

dataset for validating the model's basic classification performance. AUCD, released by the American University in Cairo, Egypt, covers typical DBs, including safe driving, talking on the phone, texting, operating the radio/center console, drinking, retrieving items from the back seat, grooming, and conversing with passengers. It contains approximately 17,000 images and 74 video clips from 31 different drivers. To ensure the reproducibility of experimental results and no data leakage, this study adopted strict partitioning and sampling strategies for the SFD and AUCD datasets. The SFD dataset adopts driver level segmentation, dividing 26 drivers into a training set (18 people, about 15400 images), a validation set (5 people, about 4400 images), and a test set (3 people, about 2600 images) in a ratio of approximately 7:2:1. The same driver image belongs only to a single subset to prevent identity overlap across sets. The AUCD dataset contains 74 videos of 31 drivers, divided in equal proportions by driver (22 in the training set, 6 in the validation set, and 3 in the testing set). The video frame sampling adopts an interval extraction strategy, retaining 1 frame every 10 frames to avoid redundancy and leakage risks caused by adjacent frames in time. After partitioning, the training, validation, and testing subsets contain approximately 9800, 2800, and 1400 images, respectively. To verify the comprehensive performance of different models, key indicators were compared, as presented in Figure 9.

Figure 9 indicates the performance of the three models on two datasets. In Figure 9(a) on the SFD dataset, the YOLO-DAD model improved mAP@0.5 by 5.3% and FPS by 32.4% compared to YOLOv5s; and improved mAP@0.5 by 2.5% and FPS by 13.9% compared to YOLOv8n [13]. In Figure 9(b) on the AUCD dataset, the YOLO-DAD model improved mAP@0.5 by 6.3% and FPS by 31.9% compared to YOLOv5s; and improved mAP@0.5 by 2.8% and FPS by 13.8% compared to YOLOv8n. Compared to YOLOv5s, the YOLO-DAD model reduced Params by 34.4% and FLOPs by 33.3%. Compared to YOLOv8n, the YOLO-DAD model reduced Params by 70.8% and FLOPs by 64.8%. Overall, YOLO-DAD improved detection accuracy and inference speed while maintaining low Params and FLOPs, demonstrating superior overall performance. To analyze the performance of different models during training, the study compared the mAP@0.5 trend curves over training to demonstrate the differences in model convergence speed and final performance. This is presented in Figure 10.

Figure 10 compares the mAP@0.5 performance of various models on different datasets over time. On the SFD dataset (Figure 10(a)), the average mAP@0.5 of the YOLO-DAD model improved by 6.6% compared to YOLOv5s and by 3.2% compared to YOLOv8n. On the AUCD dataset (Figure 10(b)), the average mAP@0.5 of the YOLO-DAD model improved by 7.1% compared to YOLOv5s and by 3.3% compared to YOLOv8n. This demonstrates that YOLO-DAD optimizes faster

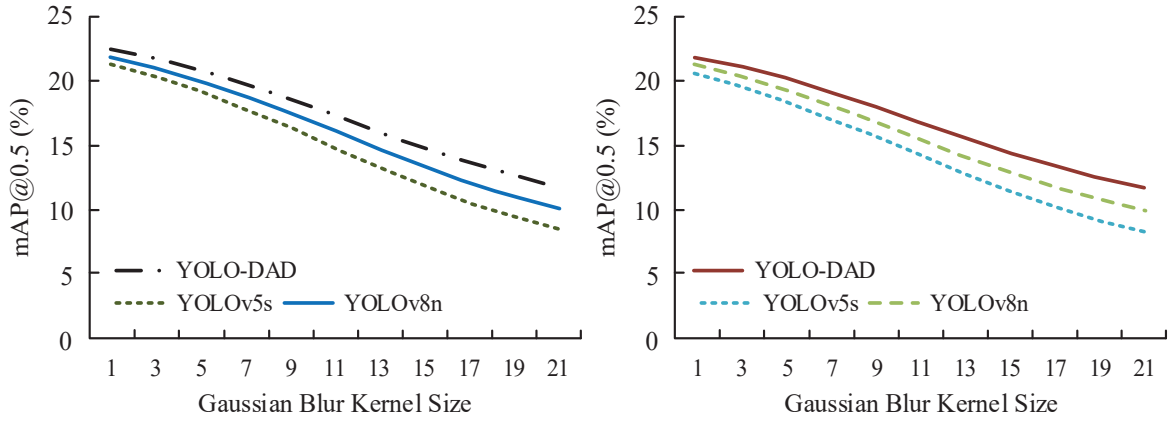
Figure 10
mAP@0.5 performance during training on validation sets.



(a) Comparison of mAP@0.5 trends on the SFD validation set

(b) Comparison of mAP@0.5 trends on the AUCD validation set

Figure 11
Comparison of robustness of different models under Gaussian blur on datasets.



(a) Comparison of robustness of SFD datasets under Gaussian Blur

(b) Comparison of robustness of AUCD datasets under Gaussian Blur

during training and achieves higher final accuracy. Gaussian blur, a common image perturbation, can affect the recognition accuracy of object detection models. To explore the adaptability of different models to blurry scenes, this study conducted robustness tests with varying Gaussian blur kernel size. This is presented in Figure 11.

Figure 11 compares the robustness of different models across datasets as the Gaussian blur kernel size varies. In Figure 11(a) on the SFD dataset, YOLO-DAD's average mAP@0.5 rose by 16.0% compared to YOLOv5s and 7.3% compared to YOLOv8n. In Figure 11(b) on the AUCD dataset, YOLO-DAD's average mAP@0.5 rose by 17.4% compared to YOLOv5s and 8.1% compared to YOLOv8n. This demonstrated that YOLO-DAD exhibited greater robustness and superior performance in the face of Gaussian blur interference.

3.2. DB Detection and Verification Based on YOLO-LBE Model

To verify the combined effect of each improved module, ablation experiments were conducted on the SFD dataset. Using YOLOv8n as the baseline, gradually add GhostC2f, BiFPN, decoupling head, WIoU, and ECA. The results of the ablation experiment are shown in Table 3.

According to Table 3, the introduction of GhostC2f module achieved a slight improvement in accuracy while achieving lightweighting. The addition of BiFPN module enabled mAP@0.5 to improve by 3.29% compared to YOLOv8n+GhostC2f. The decoupling head separates classification and regression tasks, which caused mAP@0.5 to further increase by 1.05%. The highest mAP@0.5 value of the complete YOLO-LBE model was 91.34%, and the parameter count (2.14M) only slightly increased, with minimal FPS impact, proving its efficiency. To quantify the independent contributions of each improvement module to model performance, this study conducted phased ablation experiments with YOLOv8n as the baseline and added each component separately. The results of the staged ablation experiment are shown in Table 4.

From Table 4, it can be seen that the GhostConv module not only reduces the parameter count by 0.66M, but also mAP@0.5 Increased by 0.17%, verifying the effectiveness of lightweight convolution in maintaining feature expression ability. The GhostC2f module further compresses the parameters to 1.95M, mAP@0.5 Increased to 84.53%. After introducing the BiFPN module, mAP@0.5 Improved by 3.29%, indicating that weighted multi-scale fusion plays a crucial role in detecting multi-scale and variable targets in driver behavior. The ECA module has increased by 1.11% on the basis of YOLOv8n+BiFPN mAP@0.5 The parameter increase of only 0.02M confirms the effectiveness of lightweight channel attention in enhancing key behavioral features. Decoupling head module enables mAP@0.5 Compared to YOLOv8n+BiFPN, it has increased by 0.74%, and the loss function of WIoU v3 has further improved mAP@0.5 Push to 89.10%. The sum of the independent gains of each module is basically consistent with the total gain of the complete model, confirming the synergistic effect between components rather than functional redundancy. To fully verify the selection advantages of GhostConv and ECA modules in driving behavior detection tasks, this study designed a specialized ablation experiment to compare the combined performance of different lightweight convolution modules and attention mechanisms. Using YOLOv8n as the unified baseline structure, the convolution modules in its backbone network were replaced with GhostConv and MobileNetV3's DS-Conv, and ECA, CBAM, and SE attention modules were embedded after the neck network. All models used the same training settings. Input size was 640 × 640, SGD optimizer, trained for 300 rounds. The ablation experimental results of the lightweight convolution and attention modules are shown in Table 5.

According to Table 5, under the same attention mechanism, the model using GhostConv had a lower number of parameters compared to the model using MobileNetV3 DS-Conv mAP@0.5 Higher, faster reasoning speed. The ECA module exhibited the best accuracy improvement effect and the lowest additional

Table 3
Results of ablation experiment.

Model Configuration	mAP@0.5/%	FPS	Parameter count/M
YOLOv8n	84.19	142	3.01
YOLOv8n+GhostC2f	84.53	165	1.95
YOLOv8n+GhostC2f+ BiFPN	87.82	155	2.10
YOLOv8n+GhostC2f+ BiFPN+decoupled head	88.87	151	2.12
YOLOv8n+GhostC2f+ BiFPN+decoupled head+WIoU v3	90.10	151	2.12
YOLO-LBE	91.34	148	2.14

Table 4
Results of staged ablation experiment.

Model Configuration	mAP@0.5/%	FPS	Parameter count/M
YOLOv8n	84.19	142	3.01
YOLOv8n+GhostConv	84.36	168	2.35
YOLOv8n+GhostC2f	84.53	165	1.95
YOLOv8n+BiFPN	87.82	155	2.10
YOLOv8n+ECA (after BiFPN outputs)	88.93	152	2.12
YOLOv8n+Decoupled Head (after BiFPN outputs)	88.56	153	2.11
YOLOv8n+WIoU v3 (after BiFPN outputs)	89.10	154	2.10

Note: The ECA embedding position depends on the output layer structure of BiFPN, and the decoupling head and WIoU need to be evaluated on the multi-scale features provided by BiFPN. Therefore, all three are based on YOLOv8n+BiFPN.

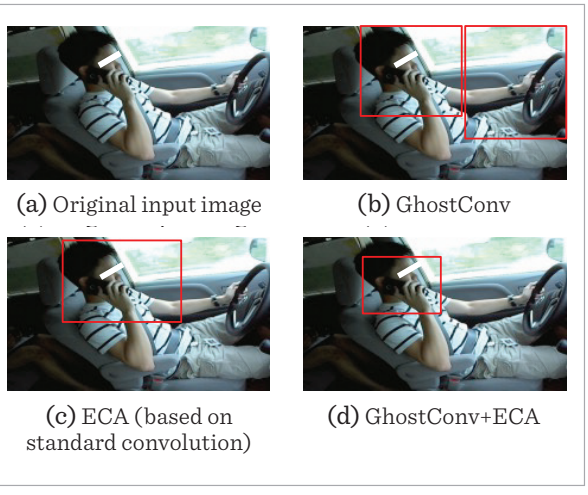
Table 5
Experimental results of ablation using lightweight convolution and attention modules.

Convolution module	Attention module	Parameter/M	mAP@0.5/%	FPS
DS-Conv	SE	2.31	84.88	152
DS-Conv	ECA	2.30	85.54	150
DS-Conv	CBAM	2.35	85.20	145
GhostConv	SE	1.98	87.06	157
GhostConv	ECA	2.00	88.93	155
GhostConv	CBAM	2.02	87.77	149

computational overhead in all convolutional backbones. The parameter count for the combination of GhostConv and ECA was 2.00M, mAP@0.5 The value was 88.93%, and the inference speed was 155 FPS, which was better than other combinations. It could effectively meet the multiple requirements of high accuracy, low latency, and low resource consumption for driving behavior detection, providing a reliable basis for the actual deployment of the model. To further verify the synergistic effect between ECA and GhostConv, visual analysis was conducted using c2 (making phone calls with the right hand) in the SFD dataset as an example. The study used GhostConv, ECA (based on standard convolution), and GhostConv combined with ECA for recognition. The recognition results of the three methods are shown in Figure 12.

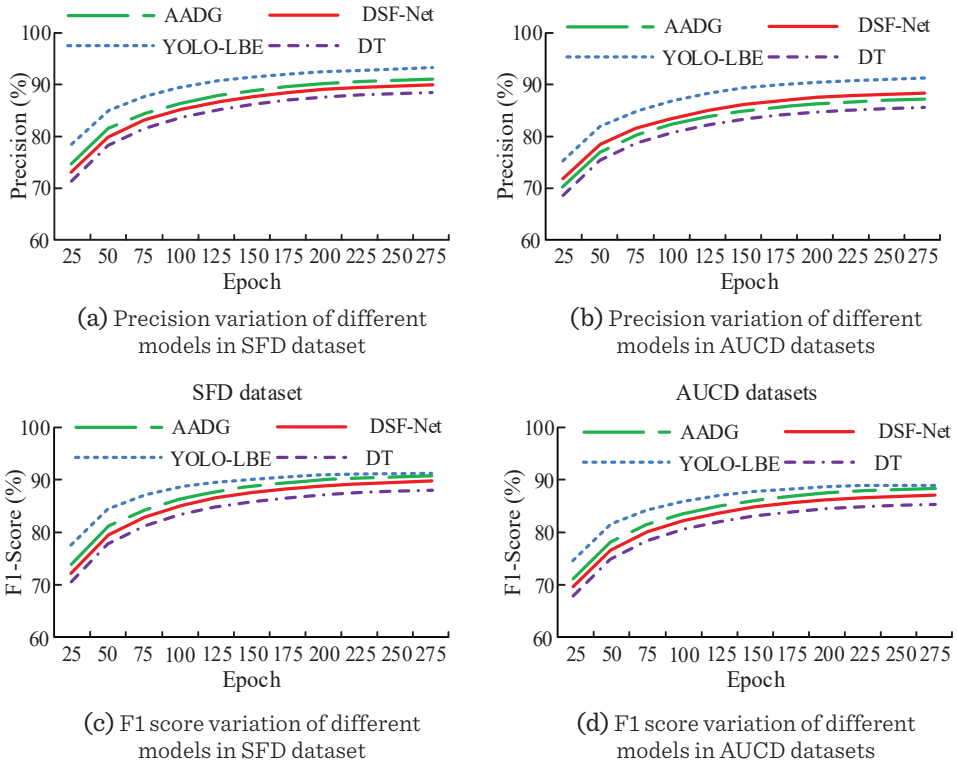
From Figure 12(b), when only GhostConv was used, the feature activation was relatively diffuse and re-

Figure 12
Identification results of three methods (Image from: photo by the author).



(a) Original input image (b) GhostConv (c) ECA (based on standard convolution) (d) GhostConv+ECA

Figure 13
Comparison of the changes in accuracy during the training process on various model datasets.



(a) Precision variation of different models in SFD dataset

(b) Precision variation of different models in AUCD datasets

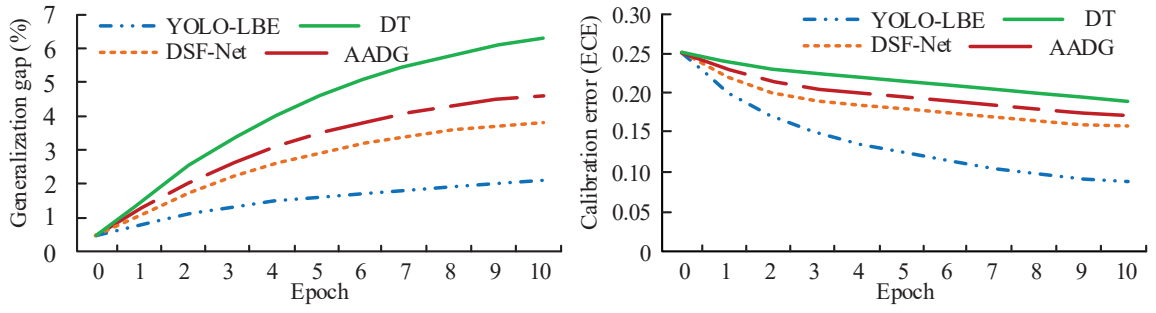
(c) F1 score variation of different models in SFD dataset

(d) F1 score variation of different models in AUCD datasets

sponsive to the target area (hand, phone), but the background noise activation was significant. From Figure 12(c), when using ECA alone, the activation was relatively concentrated, but still contains some irrelevant details due to the feature redundancy of standard convolution. From Figure 12(d), the combination of GhostConv and ECA could make the activation more concentrated and further enhance the noise suppression effect. To assess the stability and convergence of different models during training, the four models were compared on the dataset, with their precision and F1 score trends changing over time. This is presented in Figure 13.

Figure 13 indicates a comparison of the changes in precision and F1 scores of various models during training on the datasets. The blue dashed line in Figure 13 represents the metric results of the YOLO-LBE model, the purple dashed line represents the metric results of the DT model, the red solid line represents the metric results of the DSF Net model, and the green long dashed line represents the metric results of the AADG model. In Figure 13(a) on the SFD dataset, during training, YOLO-LBE's average accuracy improved by 6.6% compared to the Driven Transformer (DT) model, by 4.6% compared to the Dual-Stream Fusion Network (DSF-Net) model, and by 3.2% compared to the YOLO detector with Automatic Augmentation for Domain Generalization (AADG) using the domain generalization automatic augmentation strategy. In Figure 13(b) on the AUCD dataset, during training, YOLO-LBE's average accuracy improved by 7.4% compared to the DT model, by 5.4% compared to the DSF-Net model, and by 3.8% compared to the AADG model. In Figure 13(c) on the SFD dataset, YOLO-LBE's average F1 score rose by 5.7% compared to the DT model, by 3.6% compared to the DSF-Net model, and by 2.0% compared to the AADG model. On the AUCD dataset (Figure 13(d)), YOLO-LBE's average F1 score rose by 6.1% compared to the DT model, 4.0% compared to the DSF-Net model, and 2.3%

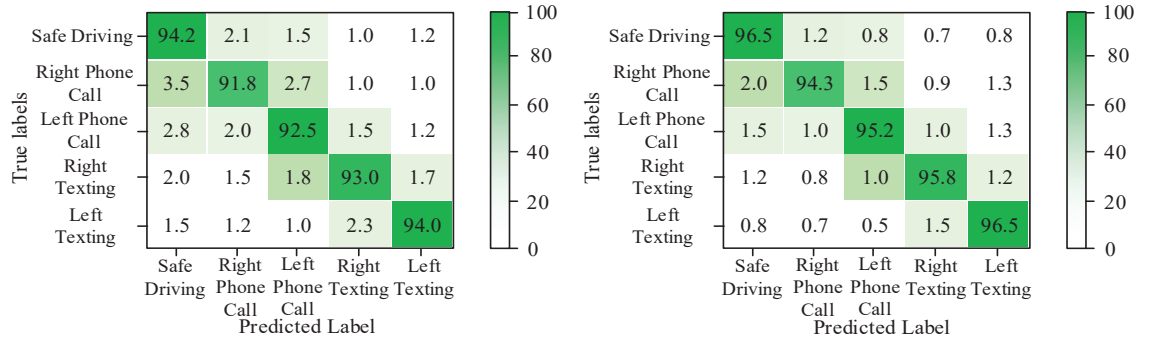
Figure 14
The generalization gap and expected calibration error of various models on the AUCD dataset.



(a) Comparison of generalization differences among different models on the AUCD dataset

(b) Comparison of expected calibration errors of different models on the AUCD dataset

Figure 15
Confusion matrix comparison of YOLOo-DAD and YOLOo-LBE in five typical categories.



(a) YOLO-DAD typical category confusion matrix

(b) YOLO-LBE typical category confusion matrix

compared to the AADG model. This demonstrated that YOLO-LBE surpassed the accuracy on both datasets, maintaining more robust and higher overall performance. To further assess the capability of various models in the DB detection task, the study chose generalization gap and calibration error as comparison metrics, as shown in Figure 14.

Figure 14 shows the generalization gap and expected calibration error of various models on the AUCD dataset. The blue dashed line in the Figure 14 represents the metric results of the YOLO-LBE model, the green solid line represents the metric results of the DT model, the orange dashed line represents the metric results of the DSF Net model, and the red long dashed line represents the metric results of the AADG model. In the generalization gap comparison in Figure 14(a), at the end of training, YOLO-LBE's generalization gap (2.1%) was reduced by 66.7%, 44.7%, and 54.3% compared to DT (6.3%), DSF-Net (3.8%), and AADG (4.6%), respectively. In the expected calibration error comparison in Figure 14(b), YOLO-LBE's expected calibration error (0.088) was reduced by 53.7%, 44.3%, and 48.8%, respectively. These results demonstrated that YOLO-LBE, through improvements such as the ECA mechanism and the WIoU LF, effectively mitigated overfitting, improved the reliability of model predictions, and provided strong support for its stability in practical deployment. To further analyze the model's recognition accuracy for specific DB categories, the study compared false positives and missed positives in the classification of key DBs, as shown in Figure 15.

Figure 15 compares the confusion matrices for the key behavior classifications of the two models studied. The data in Figure 15 (a) showed the classification accuracy of the YOLO-DAD model for various driving behaviors, while the data in Figure 15 (b) showed the classification accuracy of the YOLO-LBE model for various driving behaviors. In the safe driving category, YOLO-LBE's classification accuracy improved by 2.44% compared to YOLO-DAD. In the right-side phone call category, YOLO-LBE's classification accuracy improved by 2.72% compared to YOLO-DAD. In the left-side phone call category, YOLO-LBE's classification accuracy improved by 0.76% compared to YOLO-DAD. In the right-side texting category, YOLO-LBE's classification accuracy improved by 2.99% compared to YOLO-DAD. In the left-side texting category, YOLO-LBE's classification accuracy improved by 2.66% compared to YOLO-DAD. Overall, the YOLO-LBE model performed better in reducing cross-classification misclassification, demonstrating greater classification robustness and accuracy. To further validate the superiority of the proposed YOLO-LBE model, it was compared with three recent baseline models, namely Real Time Feature Enhanced Detection Transformer (RF-DETR), YOLOv12, and YOLO11m, on the AUCD dataset. The performance comparison results of the four models using accuracy, recall, and parameter quantity as evaluation indicators are shown in Table 6.

Table 6
Performance comparison results of four models.

Model	Accuracy/%	Recall/%	Parameter count/M
RF-DETR	89.5	88.2	32.1
YOLOv12	91.3	89.5	3.45
YOLO11m	90.6	88.9	38.8
YOLO-LBE	92.1	90.7	2.14

According to Table 6, the proposed YOLO-LBE model achieved comparable or even better results than the three advanced baseline models in terms of accuracy (92.1%), recall (90.7%), and parameter size (2.14M). The results indicated that the proposed YOLO-LBE model not only had good target recognition performance, but also achieved this goal with extremely low model complexity, demonstrating certain superiority. To comprehensively verify the stability of the proposed YOLO-LBE model in complex real-world vehicular environments, this study further evaluated its robustness under five common interferences: lighting changes, motion blur, target occlusion, compression artifacts, and viewpoint shift. The experiment is based on the AUCD test set and applies lighting changes (brightness factors set to 1.5 and 0.5 to simulate light and dark scenes), motion blur (kernel sizes set to 5 and 9 to simulate vehicle bumps), random occlusion (using black rectangles with an area of 10% to simulate occlusion), JPEG compression (quality factor set to 30 to simulate low bandwidth transmission), and perspective shift (image rotation 5° to simulate camera

Table 7
Models in complex environments mAP@0.5 Comparison results (%).

Perturbation Type	YOLOv5s	YOLOv8n	YOLO-DAD	YOLO-LBE
Original	84.52	84.19	89.67	91.34
Low brightness (0.5)	76.13	76.82	83.92	86.45
High brightness (1.5)	78.45	79.34	85.61	88.12
Motion blur (kernel 5)	72.64	73.28	81.77	84.79
Motion blur (kernel 9)	67.89	68.91	77.35	81.23
Random occlusion	70.32	71.45	80.14	83.67
JPEG compression	76.41	77.19	83.88	86.02
Viewpoint shift	74.76	75.68	82.59	84.93

installation deviation). Using YOLOv5s, YOLOv8n, and YOLO-DAD as comparative models, the performance of each model in complex environments is evaluated mAP@0.5 The comparison results are shown in Table 7.

As shown in Table 7, the proposed YOLO-LBE model was able to maintain the highest performance under all tested disturbance types mAP@0.5 Compared with YOLO-DAD and YOLOv8n, it has stable advantages. Especially in scenes with strong motion blur (kernel size 9), YOLO-LBE's mAP@0.5 (81.23%) were 3.32% and 3.88% higher than YOLOv8n and YOLO-DAD, respectively, verifying the synergistic suppression effect of BiFPN multi-scale fusion and ECA channel attention on deformation and local occlusion. The results show that the proposed YOLO-LBE model exhibits stronger robustness and detection stability against complex disturbances such as sudden changes in lighting, vibration blur, partial occlusion, and compression distortion in the vehicle environment, further supporting its feasibility for deployment on resource constrained vehicle platforms.

4. Discussion

With the rapid development of intelligent driving and in-vehicle monitoring systems, achieving high-precision, real-time DB recognition has become a critical issue in traffic safety. Traditional detection methods suffer from limitations such as low recognition accuracy, poor robustness, and difficulty in model deployment in complex driving environments. To address this, the YOLO-DAD and YOLO-LBE models were proposed. By introducing a lightweight GhostConv modules, GhostC2f modules, a cross-stage FF module, and an embedded ECA mechanism, the models aimed to improve detection precision. Compared with traditional methods, YOLO-DAD substantially improved detection precision and dependability while maintaining low computational overhead by incorporating GhostConv, BiFPN, and a decoupled DH. Experimental findings demonstrated that YOLO-DAD achieved high mAP@0.5 on the SFD and AUCD datasets, improving by 5.3% to 6.3% compared to YOLOv5s, increasing inference speed by 31.9% to 32.4%, and reducing the number of parameters by 34.4%. Especially in the Gaussian blur interference environment, YOLO-DAD still maintained strong performance, with mAP@0.5 improving by 7.3%~8.1% compared to YOLOv8n, showing excellent anti-interference ability. Compared with the driver identification method based on Siamese Temporal Convolutional Networks proposed by Azadani et al. [2], although it could realize identity recognition through steering wheel behavior characteristics, it relied on high-precision steering wheel angle sensors and did not cover fine-grained detection of typical distracting behaviors such as using mobile phones and drinking water. Although this method had a high recognition accuracy, its scope of application was limited and it was highly dependent on hardware. In contrast, YOLO-DAD only required monocular camera input to achieve end-to-end multi-class behavior detection, demonstrating a better balance between accuracy and efficiency on public datasets, providing support for its further application verification in driving scenarios.

After further introducing the ECA mechanism, the YOLO-LBE model's channel feature selection ability in complex in-vehicle environments was significantly enhanced. Its recognition accuracy in the five typical distraction behavior categories was further improved compared to YOLO-DAD, especially in the "texting on the right" and "safe driving" categories, which were improved by 2.99% and 2.44%, respectively, indicating that the ECA mechanism effectively suppressed background noise and strengthens key behavioral features. The generalization gap of YOLO-LBE was reduced by 66.7%, 44.7% and 54.3% compared to DT, DSF-Net and AADG respectively, and the expected calibration error was reduced by 53.7%, 44.3% and 48.8%, respectively. Compared with the EEG-based DB detection method mentioned in the review by Ju et al. [14], although this type of method could directly reflect the driver's neural state, it required wearing EEG acquisition equipment, which was uncomfortable, expensive, and easily affected by individual physiological differences, making it difficult to be widely used in actual vehicle environments. In contrast, YOLO-LBE was based on visual perception, making it more suitable for embedded deployment and real-time monitoring in real vehicles.

5. Conclusion

In summary, the proposed model significantly improved detection capabilities. It exhibited excellent stability under varying lighting and blurring conditions. This study validated the balance between accuracy and efficiency of the proposed method in a desktop experimental environment, providing a foundation for its further adaptation and testing on resource constrained platforms. However, this research still faces limitations, such as insufficient performance under extreme conditions and difficulty handling continuous actions. Future research will focus on integrating temporal contextual information and combining self-supervised learning strategies to boost the model's generalization capabilities.

Fundings

This work was supported by the Discipline Team Cultivation, Project of Dalian University of Science and Technology under Grant KYXK2025001.

References

1. Abba Haruna, A., Muhammad, L. J., Abubakar, M. Novel Thermal-Aware Green Scheduling in Grid Environment. Artificial Intelligence and Applications, 2022, 1(4), 244-251. https://doi.org/10.47852/bonviewAIA2202332
2. Azadani, M. N., Boukerche, A. Siamese Temporal Convolutional Networks for Driver Identification Using Driver Steering Behavior Analysis. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(10), 18076-18087. https://doi.org/10.1109/TITS.2022.3151264
3. Bhattacharjya, U., Sarma, K. K., Kakati, A., Medhi, J. P., Barman, G., Choudhury, B. K. Attention-Driven Enhancements in YOLOv8s For Improved Detection of Respiratory Conditions from HRCT Scans. IEEE Sensors Letters, 2025, 9(7), 1-4. https://doi.org/10.1109/LSENS.2025.3576896
4. Chao, Z., Xu, W., Liu, R., Cho, H., Jia, F. Surgical Action Detection Based on Path Aggregation Adaptive Spatial Network. Multimedia Tools and Applications, 2023, 82(17), 26971-26986. https://doi.org/10.1007/s11042-023-14990-1
5. Debsi, A., Ling, G., Al-Mahbashi, M., Al-Soswa, M., Abdullah, A. Driver Distraction and Fatigue Detection in Images Using ME-YOLOv8 Algorithm. IET Intelligent Transport Systems, 2024, 18(10), 1910-1930. https://doi.org/10.1049/itr2.12560
6. Duan, X., Sun, Y., Wang, J. ECA-UNet For Coronary Artery Segmentation and Three-Dimensional Reconstruction. Signal, Image and Video Processing, 2023, 17(3), 783-789. https://doi.org/10.1007/s11760-022-02288-y
7. Du, Y., Liu, X., Yi, Y., Wei, K. Incorporating Bidirectional Feature Pyramid Network and Lightweight Network: A YOLOv5-GBC Distracted Driving Behavior Detection Model. Neural Computing and Applications, 2024, 36(17), 9903-9917. https://doi.org/10.1007/s00521-023-09043-5
8. Fan, Y., Gu, F., Wang, J., Wang, J., Lu, K., Niu, J. SafeDriving: An Effective Abnormal Driving Behavior Detection System Based on EMG Signals. IEEE Internet Of Things Journal, 2022, 9(14), 12338-12350. https://doi.org/10.1109/JIOT.2021.3135512
9. Fang, S., Zhang, B., Hu, J. Improved Mask R-CNN Multi-Target Detection and Segmentation for Autonomous Driving in Complex Scenes. Sensors, 2023, 23(8), 3853. https://doi.org/10.3390/s23083853
10. Gao, J., Yi, J. G., Murphey, Y. L. Multi-Scale Space-Time Transformer for Driving Behavior Detection. Multimedia Tools and Applications, 2023, 82(16), 24289-24308. https://doi.org/10.1007/s11042-023-14499-7
11. Guo, Z., Ma, D., Luo, X. A Lightweight Semantic Segmentation Algorithm Integrating CA And ECA-Net Modules. Optoelectronics Letters, 2024, 20(9), 568-576. https://doi.org/10.1007/s11801-024-3241-z
12. Hu, J., Zhang, X., Maybank, S. Abnormal Driving Detection with Normalized Driving Behavior Data: A Deep Learning Approach. IEEE Transactions on Vehicular Technology, 2020, 69(7), 6943-6951. https://doi.org/10.1109/TVT.2020.2993247
13. Ji, B., Zhao, J., Tao, F., Zhang, J., Zhang, G., Wang, N., Zhang, P., Fan, H. A Novel Nectarine Fruit Maturity Detection and Classification Counting Model Based on YOLOv8n. IEEE Transactions on AgriFood Electronics, 2025, 3(1), 144-155. https://doi.org/10.1109/TAFE.2024.3488747
14. Ju, J., Li, H. A Survey Of EEG-Based Driver State and Behavior Detection for Intelligent Vehicles. IEEE Transactions on Biometrics, Behavior, And Identity Science, 2024, 6(3), 420-434. https://doi.org/10.1109/TBIOM.2024.3400866

15. Li, R. J., Yu, C. D., Qin, X. R., An, X., Zhao, J. P., Chuai, W. H., Liu, B. S. YOLO-SGC: A Dangerous Driving Behavior Detection Method with Multiscale Spatial-Channel Feature Aggregation. IEEE Sensors Journal, 2024, 24(21), 36044-36056. https://doi.org/10.1109/JSEN.2024.3457686

16. Li, X., Li, X., Shen, Z., Qian, G. Driver Fatigue Detection Based on Improved YOLOv7. Journal Of Real-Time Image Processing, 2024, 21(3), 75-86. https://doi.org/10.1007/s11554-024-01455-3

17. Li, Z., He, Q., Zhao, H., Yang, W. Doublem-Net: Multi-Scale Spatial Pyramid Pooling-Fast and Multi-Path Adaptive Feature Pyramid Network for UAV Detection. International Journal of Machine Learning and Cybernetics, 2024, 15(12), 5781-5805. https://doi.org/10.1007/s13042-024-02278-1

18. Liu, S., Yang, Y., Cao, T., Zhu, Y. A BiFPN-SECA Detection Network for Foreign Objects on Top of Railway Freight Vehicles. Signal, Image and Video Processing, 2024, 18(12), 9027-9035. https://doi.org/10.1007/s11760-024-03527-0

19. Lu, Y., Yu, J., Zhu, X., Zhang, B., Sun, Z. YOLOv8-Rice: A Rice Leaf Disease Detection Model Based on YOLOv8. Paddy And Water Environment, 2024, 22(4), 695-710. https://doi.org/10.1007/s10333-024-00990-w

20. Postupaiev, S., Damaševičius, R., Maskeliūnas, R. Real-Time Camera Operator Segmentation with YOLOv8 In Football Video Broadcasts. AI, 2024, 5(2), 842-872. https://doi.org/10.3390/ai5020042

21. Quang, N. H., Lee, H., Kim, N., Kim, G. Real-Time Flash Flood Detection Employing the YOLOv8 Model. Earth Science Informatics, 2024, 17(5), 4809-4829. https://doi.org/10.1007/s12145-024-01428-x

22. Shin, Y., Yu, C. SAR Ship Detection Based on Improved YOLOv5 And BiFPN. ICT Express, 2024, 10(1), 28-33. https://doi.org/10.1016/j.icte.2023.03.009

23. Sukumar, N., Sivashankar, B., Sumathi, P., Maurya, O. P. Physiological and Physical Sensors for Stress Level, Drowsiness Detection, And Behaviour Analysis. IEEE Transactions on Consumer Electronics, 2024, 70(1), 656-668. https://doi.org/10.1109/TCE.2024.3366988

24. Tan, D., Tian, W., Wang, C., Chen, L., Xiong, L. Driver Distraction Behavior Recognition for Autonomous Driving: Approaches, Datasets and Challenges. IEEE Transactions on Intelligent Vehicles, 2024, 9(12), 8000-8026. https://doi.org/10.1109/TIV.2024.3405990

25. Tao, H. Smoke Recognition in Satellite Imagery Via an Attention Pyramid Network with Bidirectional Multilevel Multigranularity Feature Aggregation and Gated Fusion. IEEE Internet Of Things Journal, 2024, 11(8), 14047-14057. https://doi.org/10.1109/JIOT.2023.3339476

26. Uddin, M. A., Hossain, N., Ahamed, A., Islam, M. M., Khraisat, A., Alazab, A., Ahamed, M. K. U., Talukder, M. A. Abnormal Driving Behavior Detection: A Machine and Deep Learning Based Hybrid Model. International Journal of Intelligent Transportation Systems Research, 2025, 23(1), 568-591. https://doi.org/10.1007/s13177-025-00471-2

27. Wang, C., Lin, M., Shao, L., Xiang, J. MF-YOLO: A Lightweight Method for Real-Time Dangerous Driving Behavior Detection. IEEE Transactions on Instrumentation and Measurement, 2024, 73, 1-13. https://doi.org/10.1109/TIM.2024.3472868

28. Wang, C., Yang, X., Jeon, G. Hybrid Attention Feature Refinement Network for Lightweight Image Super-Resolution In Metaverse Immersive Display. IEEE Transactions on Consumer Electronics, 2024, 70(1), 3232-3244. https://doi.org/10.1109/TCE.2023.3329813

29. Wei, F., Wang, W. SCCA-YOLO: A Spatial and Channel Collaborative Attention Enhanced YOLO Network for Highway Autonomous Driving Perception System. Scientific Reports, 2025, 15(1), 6459-6475. https://doi.org/10.1038/s41598-025-90743-4

30. Xu, P., Liu, M., Zhu, C., Zhang, X., Wang, X., Feng, Q., Han, J., Li, L. Design of Multichannel Triple-Band Reflective Metasurfaces with Large Frequency Ratio Supporting Spin Decoupling and Full-Polarization Decoupling. IEEE Transactions on Microwave Theory and Techniques, 2025, 73(7), 3683-3698. https://doi.org/10.1109/TMTT.2024.3519424

31. Yang, X., Liu, C., Han, J. Reparameterized Underwater Object Detection Network Improved by Cone-Rod Cell Module and WIoU Loss. Complex & Intelligent Systems, 2024, 10(5), 7183-7198. https://doi.org/10.1007/s40747-024-01533-w

32. Zhang, M., Zhang, F. Lightweight YOLOv8 Networks for Driver Profile Face Drowsiness Detection. International Journal of Automotive Technology, 2024, 25(6), 1331-1343. https://doi.org/10.1007/s12239-024-00103-w

33. Zhu, Y., Wang, Z., Zhang, W., Liu, Y., Wu, H. A Hybrid Model for Ultra-Short-Term PV Prediction Using SOM Clustering And ECA. Electrical Engineering, 2025, 107(3), 3349-3358. https://doi.org/10.1007/s00202-024-02710-3