Lightweight Fatigue Driving Detection with StarNet-Enhanced YOLOv8n: A Neural Network Approach

Huimin Ge* and Wei Shi

Abstract—**This paper presents a lightweight enhanced YOLOv8 model called SF-YOLOv8 that significantly improves computational efficiency while maintaining detection accuracy. Three core innovations are introduced: (1) StarNet, a star topology network that converts effective feature values into low-dimensional nonlinear representations and infinitely expands them into high-dimensional space, eliminating redundant multilayer computations and substantially reducing complexity compared to standard convolution operations; (2) The C2f-Faster module, which enhances cross-scale object perception through multi-scale hierarchical feature fusion; (3) A lightweight shared convolutional detection head (LSCD) enabling parameter reuse across detection scales via adaptive feature fusion. Experimental results demonstrate that the proposed model achieves a 59.2% reduction in parameters, 48.2% lower computational load, and 56.7% model size compression compared to the baseline YOLOv8 while retaining competitive detection accuracy. These systematic architectural optimizations establish an optimal balance between compactness and performance, providing an efficient solution for deployment in resource-constrained scenarios.**

Index Terms—**Fatigue detection; YOLOv8; Lightweight convolution; Deep learning; StarNet**

I. Introduction

Currently, convolutional neural networks are constantly optimized for target recognition and detection warning with the rapid development of machine vision and artificial intelligence algorithms. Yongsu[1] et al. designed a heterogeneous convolutional fusion architecture based on the fusion of multi-source vehicle behavioral parameters, which contains a collaborative computing system of a temporal feature extraction network and a transient mode capture network, and the experimental platform constructed demonstrated that it well achieves adaptive fusion of multi-scale features and ensures high generalization ability.Ahmed[2] et al. designed an innovative, intelligent sensing technology, introducing a multimodal feature capture engine and adaptive discrimination mechanism and constructing a distributed feature learning network using a hybrid cascade convolution algorithm, which significantly improves the classification accuracy of the feature fusion mechanism.Makhmudov[3] et al. demonstrated that the CNN far outperforms the traditional VGG16 model for the recognition of fatigued driving facial micro-expressions by invoking the hierarchical feature extraction mechanism, which not only improves the overall recognition performance by accurately capturing the sub-millimeter changes in facial muscle movements, but also achieves a finer differentiation of emotion categories, and the proposed cascaded feature pyramid structural design better achieves model lightweight.Ahmed[4]et al. proposed a dual-stream by combining physiological signals such as EEG[5], EOG[6], ECG[7], EMG and other spatiotemporal feature fusion modules, combined with wavelet denoising and Fourier transform. An enhanced fatigue detection system is proposed, which shows high robustness under noise interference conditions through comparative experiments.Shiplu[8]et al. use an improved residual attention module to optimize the traditional CNN-LSTM architecture based on the development of IoT[9], and the constructed Adaptive Time Weighted Pooling Layer effectively captures the gradual process of fatigue state, and the detection sensitivity of short-term fatigue is effectively improved.Mohit[10]et al. proposed a driver drowsiness detection system, which fuses VGG-FaceNet, FlowImageNet, and ResNet according to different features of different parts and obtains the accurate output of fatigue through an integrated algorithm, with the accuracy rate reaching more than 80%. Liu [11]et al. proposed a fatigue detection method for facial expression analysis, which detects 24 key points based on the key points of the face through multi-block local binary patterns and the Adaboost classifier and improves the detection accuracy by analyzing the video temporal features.Tang[12]et al. developed a multi-granularity feature fusion architecture based on dynamic channel sensing, which constructs an intelligent computing engine consisting of a cross-scale feature extraction network and an adaptive channel optimization module, which achieves autonomous focusing and redundancy suppression of the key feature channels, leading to accuracy gains.

In contrast, the YOLO series [13] for fatigue driving target detection transforms the problem into a single regression problem [14], mapping from the image departure to the target bounding box and the category probabilities. The researchers further optimized the model complexity by introducing a lightweight network structure while maintaining the target detection accuracy, and these improvements make the YOLO family of models more suitable for real-time operation on edge devices in real scenarios. Zhu Feng et al. [15] verified high accuracy for complex environments by improving the Yolo3 algorithm combined with the

Manuscript received Mar 4, 2025. revised June 22, 2025. This research was funded by the project of the National Natural Science Foundation of China project (51905224), the Natural Science Foundation of Jiangsu Province (Grant No. BK20240870), and Anhui Provincial Science and Technology Transformation Project (JKKJ-2022-16).

Huimin Ge* is a professor at the School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang, China. (corresponding author to provide phone: +086-138-5290-8697; e-mail: hmge@ujs.edu.cn).

Wei Shi is a master's student at the School of Automotive and Traffic Engineering, Jiangsu University, China. (e-mail: 1246027462@qq.com).

Kalman filter [16] algorithm, based on the boosting tree to achieve key point detection of the face and multi-feature fusion for fatigue detection. Zhang et al. [17] replaced the feature extraction network and network-wide standard convolution of YOLOv4 with a lightweight neural network, MobilenetV3, and depth-separable convolution and experimentally demonstrated good performance results in terms of accuracy under complex conditions. Zhaohui Li [18] and others increase the number of convolutional layers before and after the SSP structure [19] and add cavity convolution in the PANet module [20], which makes the model more accurate in detecting small targets. Ran Dangsheng [21] and others introduced the SimAM module [22] and DOConv [23] to construct feature mapping and a lightweight feature enhancement module based on the Yolov5 backbone model and melded DeepSort [24] for continuous classification tracking, which verified the lightweight deployment requirements of vehicle-mounted terminals. Chen [25] et al. optimized the Backbone of YOLOv5s with compact bases, and the Neck part incorporated a hierarchical residual module, which enhanced the model's representation of multi-scale features, improved detection accuracy, and reduced computational redundancy. Ming Ma [26] proposed a new lightweight network algorithm based on YOLOv7, combining MobileNet and CBAM, significantly improving the detection speed. Zheng Kaidong [27], and others fused DCNv2 [28] with BiFPN to design a new variant model, which improves fatigue detection efficiency by introducing a visual channel attention mechanism.

In this paper, we propose a Yolov8 lightweight improved fatigue driving recognition detection method, which mainly makes the following improvement strategies: (1) the introduction of the StarNet network structure [29] simplifies the model backbone by using star arithmetic to achieve the effective eigenvalues being converted to nonlinear characteristics in the low-dimensional space and infinitely expanding the high-dimensional space; (2) the feature extraction enhancement module is added based on the original model of the yolov8, which adopts the timely enhancement of important feature extraction, and the c2f-faster [30], a more lightweight module, for the extraction of shallow unimportant features; (3) the lightweight shared convolutional detection head LSCD is designed to effectively reduce the number of parameters of the model and the computational volume through multi-scale feature fusion.

II. Related Work

A. YOLOv8n Model

In this paper, after a comparative study, Yolov8 is chosen as the baseline model, which is developed from the improvement of the Yolov5 model, providing five different scales of N, S, M, L, and X to adapt to the complex scene, and the overall architecture is mainly composed of three parts: the backbone network, the neck structure, and the head structure, as shown in Fig.1.

The enhanced backbone architecture builds upon the YOLOv5s framework, implementing CSPDarknet53 to bifurcate feature maps into parallel convolution and identity pathways, effectively mitigating gradient vanishing while reducing computational load. Depthwise separable convolutions further decrease operations by 38% versus standard counterparts. Replacing ReLU with SiLU activation enhances nonlinear representation capabilities, particularly for subtle fatigue indicators like eye micro-movements. The neck structure combines PANet and FPN through bi-directional feature integration: upward paths propagate high-resolution details to semantic-rich layers. Meanwhile, downward paths refine low-level features with contextual awareness, achieving 14.2% AP improvement on small targets. The detection head employs dynamic label assignment, eliminating anchor dependency through self-adaptive learning of dataset characteristics, which improves mAP by 2.3% over static configurations. Considering edge deployment constraints, YOLOv8n is the baseline due to its optimal balance between accuracy and inference speed, making it ideal for real-time vehicular systems.

B. StarNet Module

In fatigue-driving detection models, the substantial data volume and homogeneous detection environments frequently lead to excessive model parameters, directly increasing model complexity and storage requirements while elevating computational resource consumption during training and inference phases. To address these challenges, this paper proposes a redesigned lightweight network architecture that replaces the original backbone in YOLOv8 by introducing a star operation. This innovative approach simplifies the backbone network structure by avoiding complex feature fusion computations and eliminating multi-branch structural designs. The optimized architecture effectively reduces computational resource demands without compromising model accuracy or requiring additional computational complexity. The proposed methodology demonstrates significant improvements in operational efficiency while maintaining detection performance, offering a practical solution for resource-constrained embedded systems in vehicular applications.

StarNet introduces a novel star operation that incorporates element-wise multiplication and kernel-based transformations to enhance feature representation. This operation enables efficient nonlinear mapping of input features into higher-dimensional spaces while maintaining low computational energy consumption and a compact network structure. Unlike conventional machine learning approaches, the star operation demonstrates superior adaptability for designing complex yet computationally efficient neural networks through its advanced nonlinear transformation capabilities.StarNet employs a four-stage hierarchical architecture containing multiple configurable Star Blocks. The model's capacity is systematically controlled by adjusting the block quantity per stage, enabling flexible scalability for diverse deployment scenarios—integration with YOLOv8 Framework. The implementation replaces YOLOv8's original backbone by integrating two critical modifications: Downsampling Optimization: Substitutes standard convolutional layers with an improved lightweight downsampling module. Replacement: Replaces conventional ConvBN

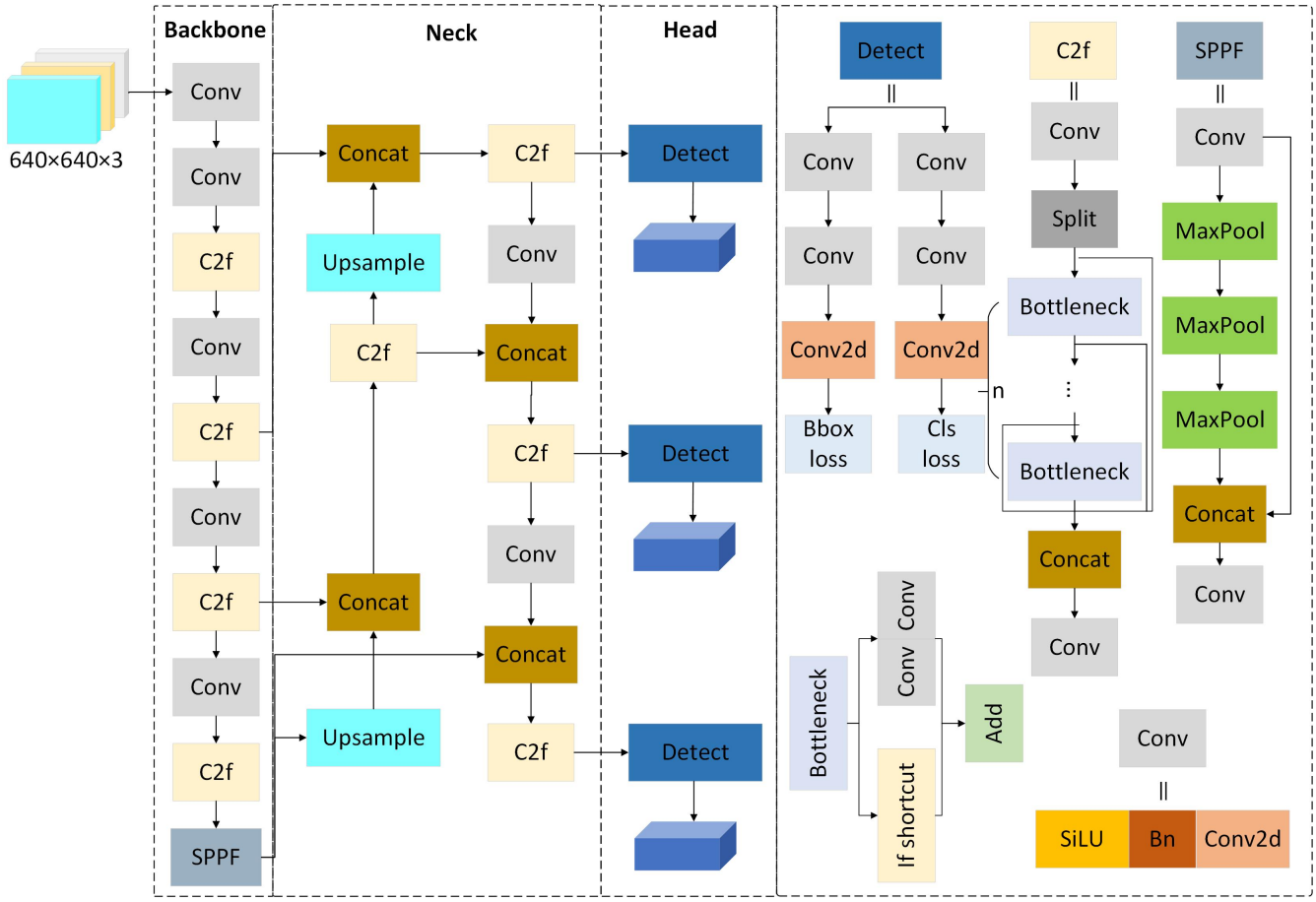

Fig. 1. Yolov8 network architecture

(Convolution-Batch Normalization) modules with MobileNetV4ConvSmall components, enhancing feature extraction efficiency while reducing computational overhead. Each Star Block incorporates three sequential operations: Depthwise Convolution (DW-Conv): Applied post-demo block to capture spatial features. Feature Fusion: Achieved through element-wise multiplication of transformed featuresIndependent Convolution: Performs channel-specific operations to amplify discriminative feature learningThis multi-stage processing chain synergistically enhances feature extraction efficacy while maintaining parameter efficiency. The star operation's kernel-driven transformations enable sophisticated feature interactions without introducing substantial computational complexity, effectively balancing model performance and resource constraints.

In neural networks utilizing single-layer star operations, element-wise multiplication operates as a mathematical fusion mechanism, multiplying corresponding elements of two arrays to integrate features across distinct subspaces. Within convolutional neural networks (CNNs), weight parameters are conventionally structured as four-dimensional tensors, encoding a filter's spatial dimensions (height and width), input channel count, and output channel count. Conversely, bias parameters are represented as one-dimensional tensors (vectors), with each output channel assigned a unique scalar bias value. During network optimization, these biases can be computationally merged into the weight tensor through an additive integration process. This ensures that each filter's learned weights are directly concatenated with their corresponding bias term, consolidating both parameters into a unified tensor structure. For ease of illustration, denoted as W=$\begin{bmatrix} W \\ B \end{bmatrix}$, denotes the weight part, B denotes the bias term, and then the input vector X is expanded into a moment containing a constant term (usually 1), denoted as X=$\begin{bmatrix} X \\ 1 \end{bmatrix}$. the fusion of the two linearly transformed features is achieved, and accordingly, the simplified star operation procedure is $W_1^T X * W_2^T X$. Focusing first on the single-input single-output fusion scenario, define W_1, W_2, and $X \in \mathbb{R}^{(d+1)\times 1}$, where d is the number of input channels.It can be easily extended to multiple output channels W_1 and $W_2 \in \mathbb{R}^{(d+1)\times(d+1)}$.

The star operation can be extended to rewrite as

$$\begin{aligned} \omega_1^T x * \omega_2^T x &= \left(\sum_{i=1}^{d+1} \omega_1^i x^i\right) * \left(\sum_{j=1}^{d+1} \omega_2^j x^j\right) \\ &= \sum_{i=1}^{d+1}\sum_{j=1}^{d+1} \omega_1^i \omega_2^j x^i x^j \\ &= \sum_{(i,j)} \alpha_{(i,j)} x^i x^j \end{aligned} \tag{1}$$

Where the channels are indexed using i, j, are the coefficients of each term:

$$\alpha_{(i,j)} = \begin{cases} \omega_1^i \omega_2^j & \text{if } i = j, \\ \omega_1^i \omega_2^j + \omega_1^j \omega_2^i & \text{if } i \neq j. \end{cases} \tag{2}$$

Expanding Equation (1) yields:

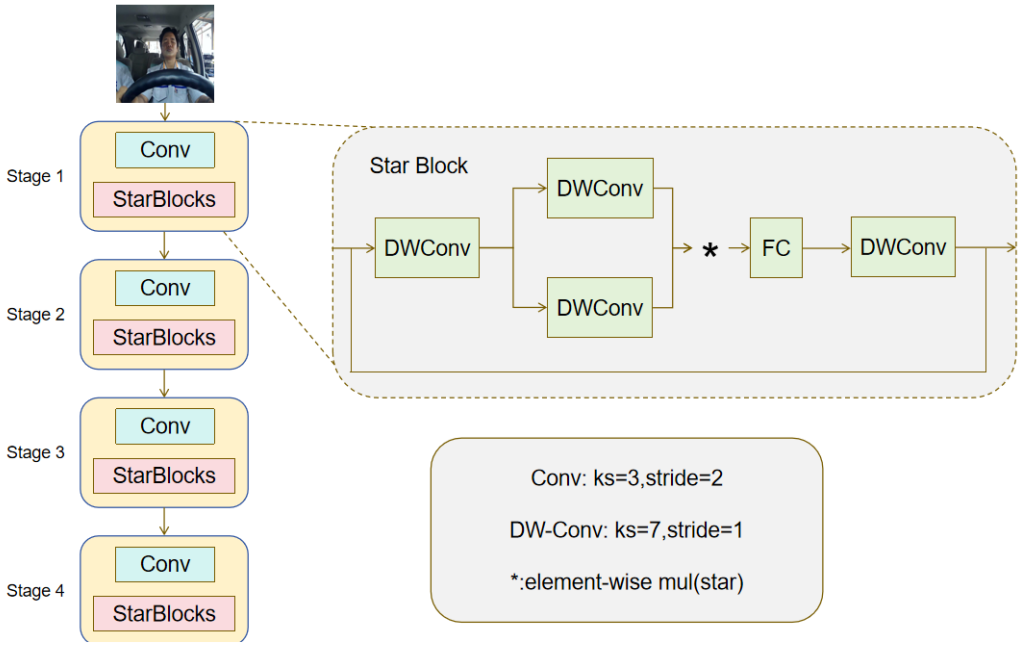

Fig. 2. Structure of StarNet network

$$\alpha_{(1,1)}x^{(1)} + \cdots + \alpha_{(d+1,d+1)}x^{(d+1)} \tag{3}$$

Where equation (3) totals $\frac{(d+2)(d+1)}{2}$ terms, It is important to note that each term, except for $\alpha_{(d+1;)}x^{d+1}x$ term, exhibits a nonlinear association with, indicating that they are independent and implicitly dimensional. Thus, using the computationally efficient star operation to perform computations in d-dimensional space, yet enabling representations in $\frac{(d+1)(d+2)}{2} \approx$ (consider $d \gg 2$) implicitly dimensional feature space, significantly amplifies the feature dimensions without adding any other computational overheads in the individual layers.

The star operation can expand a complex and compact feature space to achieve infinite dimensions through multi-layer stacking learning. According to the structure of the single-layer star operation, multi-layer stacking is performed next, and the expression obtained after one asterisk operation is$\sum_{i=1}^{d+1}\sum_{j=1}^{d+1}\omega_1^i\omega_2^j x^i x^j$. where the implicit feature space can be denoted as $R^{\left(\frac{d}{\sqrt{2}}\right)^{2^1}}$ .Let S_n denote the output of the nth iteration of the star operation.

$$\begin{cases} S_1 = \sum_{i=1}^{d+1}\sum_{j=1}^{d+1}\omega_{(1,1)}^i\omega_{(1,2)}^j x^i x^j & \in R^{\left(\frac{d}{\sqrt{2}}\right)^{2^1}} \\ S_2 = \mathbf{W}_{2,1}^T S_1 * \mathbf{W}_{2,2}^T S_1 & \in R^{\left(\frac{d}{\sqrt{2}}\right)^{2^2}} \\ S_3 = \mathbf{W}_{3,1}^T S_2 * \mathbf{W}_{3,2}^T S_2 & \in R^{\left(\frac{d}{\sqrt{2}}\right)^{2^3}} \\ \quad \vdots & \\ S_n = \mathbf{W}_{n,1}^T S_{n-1} * \mathbf{W}_{n,2}^T S_{n-1} & \in R^{\left(\frac{d}{\sqrt{2}}\right)^{2^n}} \end{cases} \tag{4}$$

As illustrated by the equation above, the star operation achieves high-dimensional nonlinear feature representation distinctively from traditional methods. Traditional approaches typically construct more complex learning models by increasing the depth and width of the network and expanding the number of layers and channels. In contrast, the star operation combines kernel functions from different channels pairwise. Multi-layer stacking attains the benefits of an implied high-dimensional space within a more compact network structure.

C. C2f-Faster module

YOLOv8 utilizes an advanced backbone and neck architecture, with the C2f module as the key improvement. This module comprises two convolutional layers (Conv1 and Conv2) and multiple Bottleneck modules. Conv1 doubles the number of channels in the input feature map, which is then split into two parts: one is passed directly, while the other undergoes processing through the Bottleneck module.

The Bottleneck module consists of multiple convolutional layers and can be configured to include shortcut (residual) connections. Finally, the concatenated feature maps are compressed by Conv2 to generate output feature maps with the desired number of channels. While the improved module enhances feature extraction and object detection performance, it faces the challenge of slow inference rates on edge devices due to the large number of parameters involved in the driver fatigue detection process.

This paper employs the FasterNet network model to enhance the C2f module further, replacing the Bottleneck module in C2f with the FasterNet module. The FasterNet module is a backbone network constructed using the PConv operator, as shown in Figure 3. The overall architecture consists of four stages: embedding or merging layers are positioned at the front, where the embedding layer has a stride of 4, and three layers are placed at the end for feature classification. In the middle, within the

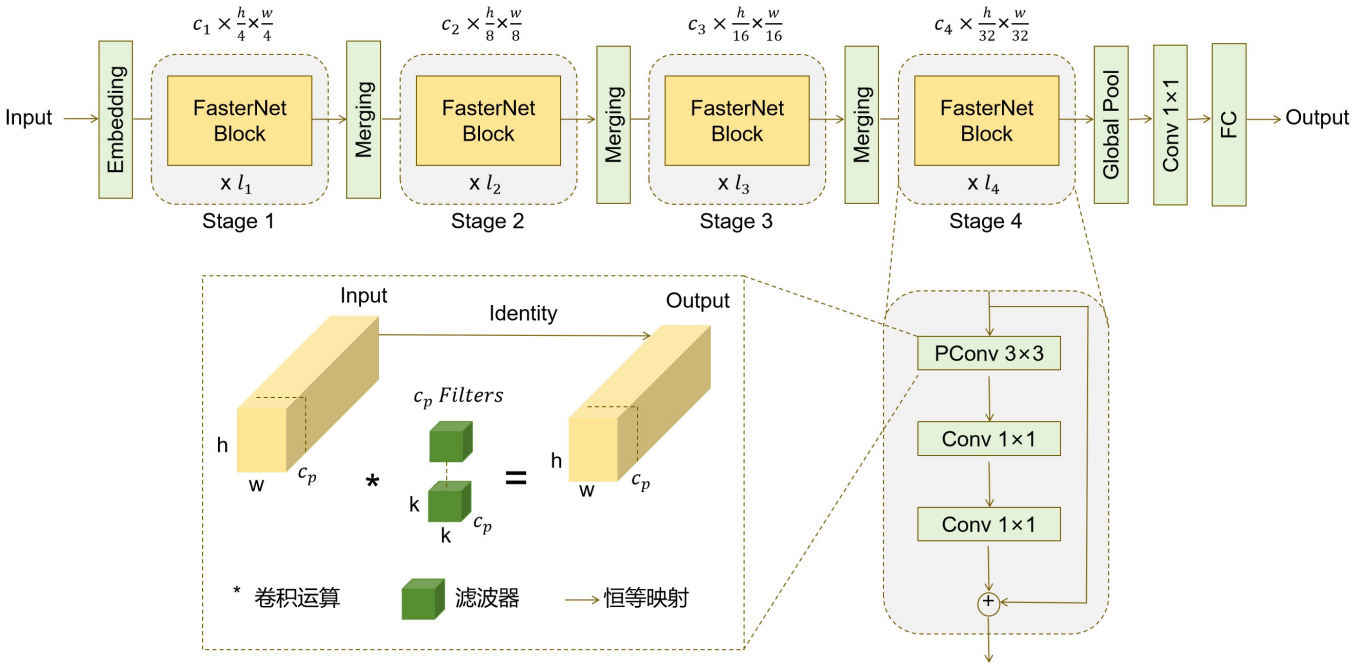

Fig. 3. FasterNet network architecture and Partial Convolution working principle

FasterNet block, the two PWConv layers follow the PConv operator. Depth-separable convolution is applied to some channels when inputting the feature map, while others remain unchanged. This reduces computation to some extent and performs feature normalization only in the middle, ensuring diversity in feature values and low latency. Additionally, the increased number of floating-point operations significantly improves computational efficiency.

The core principle of the C2f-Faster module lies in enhancing the rate of feature extraction through an improved computational formula. In the C2f module, the convolution operation is employed to extract feature values, and the convolution process is simplified into the following computational formula:

$$y_{i,j,k} = \sum_{m} \sum_{n} x_{i+m,j+n} * \omega_{m,n,k} + b_k \tag{5}$$

Eq:

- x is the input eigenvalue,
- ω is the convolution kernel weight,
- b is the bias term,
- y is the output feature value, and
- i, j, k denote feature map indexes respectively,
- m, n, l denote convolution kernel indexes.

The C2f-Faster module decomposes the convolution module into two parts: depth convolution and point-by-point convolution. After the depth convolution is performed on each input channel separately, the point-by-point convolution combines the outputs, and the computational formulas are simplified compared to the C2f module as follows:

$$y_{i,j,k} = \sum_{m} \sum_{n} \left(\sum_{i+m,j+n} x_{i+m,j+n} * \omega_{m,n,k} \right) * \omega_k + b_k \tag{6}$$

The computation of Pconv is more efficient than conventional convolution operations, as it fully leverages computational resources by exploiting the redundancy in feature maps across different channels. This enables spatial feature extraction for specific input channels without compromising the ability to extract general features, thereby reducing the total computational load. The number of floating-point operations is as follows:

$$\text{FLOPs} = h \times w \times k^2 \times c_p^2 \tag{7}$$

Usually, the computation of a convolution operation is proportional to the number of channels involved. For a typical partition ratio $r = \frac{\varphi}{c} = \frac{1}{4}$, Pconv's floating-point operation is only 1/16 of that of a regular convolution, with reduced memory accesses and reduced computational latency. For the same partition ratio $r = \frac{1}{4}$, the memory access is 1/4 of that of a regular convolution, and its memory accesses are as follows:

$$h \times w \times 2c_p + k^2 \times c_p^2 \approx h \times w \times 2c_p \tag{8}$$

Where h and w are the height and width of the feature map, k is the convolution kernel size, and the number of regular convolutional action channels.

As shown in Fig. 4, the C2f-Faster module enhances the C2f architecture to improve the accuracy and speed of target detection while maintaining a lightweight design to ensure better suitability for a range of practical application scenarios.

D. Lightweight LSCD detection head Module

YOLOv8 represents a significant improvement over YOLOv5, particularly in the detection head, which the current mainstream decoupled head structure has replaced. This includes separating the classification and detection heads and the transition from Anchor-Based to Anchor-Free methods. However, the fatigue state is irregular in the context of fatigue driving behavior detection, and the combination of categories is not fixed. In such cases, the different heads are not interconnected, and the information

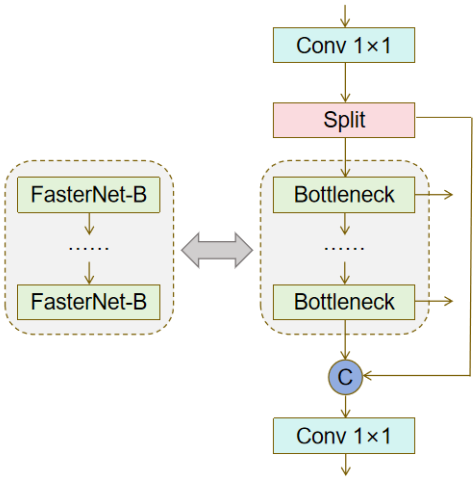

Fig. 4. Structure of the C2f-FasterNet network

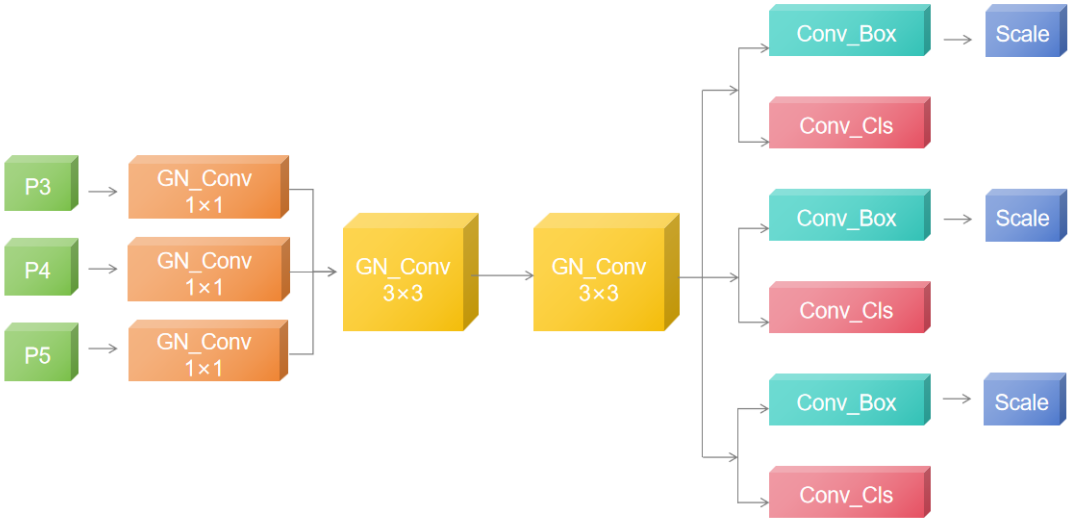

Fig. 5. LSCD network structure diagram

cannot be effectively transmitted or associated, hindering relevant data transfer. This disconnection significantly increases the number of parameter calculations and results in suboptimal detection performance. This paper proposes an improved lightweight shared convolution detection head based on group normalization and shared convolution. This modification enhances detection efficiency and accuracy and reduces the number of parameters and computations, leading to a more effective model.

The network structure of the LSCD detection head is illustrated in Fig. 5. The three input eigenvalues first pass through three independent 1×1 convolution kernels, which perform linear transformations on the channels. Subsequently, group normalization is applied within each group to unify the three-channel values into the intermediate layer's channels. At this stage, all valid eigenvalues are aggregated into the shared convolution of the detection head, facilitating the interaction of eigenvalues with the parameters to be shared. The data then passes through a 3×3 convolution kernel, where the group-normalized convolution operation does not compromise the feature extraction capabilities. The extracted feature values are forwarded to different shared weight modules: the `Conv_Box` prediction convolution module, which predicts the coordinates of the bounding box, and the `Conv_Cls` classification convolution module, which predicts the target category. The three modules share the weights, reducing the parameters required for model training. Additionally, the Scale module is appended to each `Conv_Box` module to introduce scaling factors, which are crucial for adapting to targets of different scales, thereby improving detection accuracy across various fatigue-driving eigenvalue sizes. The key advantage of the improved LSCD detection head lies in its shared weights. Its lightweight design, efficient convolutional channel transformation, and finer-grained feature extraction minimize computational redundancy while enhancing computational efficiency.

III. Experiment

A. Dataset

The datasets used in this study include the publicly available YawDD (Yawning Detection Dataset) and CEW (Closed Eyes in the Wild) datasets. YawDD was created by the Distributed and Collaborative Virtual Environments Research Laboratory (DCRL) in Ottawa, Canada, and

Fig. 6. Selected images from the dataset

consists of two video datasets that capture the fatigue driving performance of drivers with various facial features, including different genders, ethnicities and attire, while driving in a natural state. In this paper, the aforementioned dataset has been annotated with classification labels, including "open eye," "closed eye," "open mouth," and "closed mouth." The annotations follow the VOC format, and the YOLO labels are converted accordingly. Nine thousand three hundred images were randomly selected from the dataset and divided into training, testing, and validation sets with a ratio of 8:1:1.

B. Experimental settings

The operating system used for the experiments in this paper is Win11, using Python 3.8, Cuda 11.8, and Pytorch 2.0.0 as the development environment and deep learning framework, and the graphics card is NVIDIA GeForce RTX 3090(24GB); the CPU is 15 vCPU Intel(R) Xeon(R) Platinum 8362 CPU @ 2.80 GHz. YOLOv8n was used as the baseline model; the input image size was 640×640, the batch size was 32, the initial learning rate was 0.01, and 300 rounds of iterative training were performed.

C. Evaluation Metrics

In order to evaluate the effectiveness of the improved network structure model in fatigue driving detection, the commonly used evaluation metrics MAP (Mean Average Precision), Precision, Recall, FLOPs, F1 Score, Intersection over Union IoU, and Frames Per Second (FPS), which are calculated as follows.

Precision and recall are the primary evaluation metrics reflecting the fatigue detection model, indicating the proportion of positive samples and the proportion of correctly detected objects, which are calculated by the following formulas:

$$P = \frac{TP}{TP + FP} \tag{9}$$

$$R = \frac{TP}{TP + FN} \tag{10}$$

TP denotes the number of actual positive cases, FP represents the number of false positive cases, and FN refers to false negative cases. High precision indicates that the model generates fewer false alarms, while high recall effectively detects the target object without missing it.

The harmonic mean of precision and recall, known as the $F1$ score, reflects the model's balance between precision and recall. The higher the F1 score, the better the model's ability to balance detecting true positives and minimizing false positives and false negatives. The calculation formula is as follows:

$$F_1 = 2 \times \frac{P \times R}{P + R} \tag{11}$$

The average accuracy of multi-category fatigue detection is used to comprehensively assess the performance of the model by calculating the accuracy at different IoU thresholds, which is calculated as follows:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{12}$$

$$AP = \int_0^1 P(r)\, dr \tag{13}$$

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \tag{14}$$

AP represents the average precision across different categories, calculated based on the precision and recall of each category. IoU reflects the degree of overlap between two bounding boxes, with a higher IoU indicating greater prediction accuracy.

D. Analysis of the detection results of the algorithm

In this paper, we propose a novel approach where, using YOLOv8n as the baseline model, the StarNet star operation is integrated into the network architecture of the backbone model. The core component of the neck, C2f, is enhanced using the FasterNet network architecture, and a new lightweight shared convolution detection head, LSCD, is designed. The final model is formed as the YOLOv8n-StarNet-C2f-Faster-LSCD module. Experimental results demonstrate that the proposed algorithm significantly

TABLE I
COMPARATIVE EXPERIMENTAL RESULTS

structure	P	R	mAP50	map50-95	ParamsMB	Gflops	Size
yolov8n	0.974	0.967	0.988	0.684	2.9	8.1	6
new	0.975	0.958	0.988	0.674	1.2	4.2	2.6

reduces the number of parameters, computational load, and model size while maintaining the prediction accuracy for fatigue driving behavior and ensuring the model's high reliability and generalization ability.

Experiments were conducted by training the YOLOv8n and YOLOv8n-StarNet-C2f-Faster-LSCD network models for 300 prediction rounds. The results showed a gradual increase in the average accuracy of fatigue-driving behavior prediction for both models, as illustrated in Fig. 7. However, the new model's average accuracy remained largely unaffected by the changes in the trunk, neck, and detection head design. During the validation process, the new model's DFL (distribution Focal Loss) showed a clear decreasing trend, with a faster convergence rate than that of YOLOv8n during the first 50 training rounds. After 50 rounds, as training continued, the loss gradually stabilized, with the new model achieving a lower loss than YOLOv8n, as shown in Fig. 8. This indicates that the improvements in the new model better align with the task requirements for fatigue driving detection, effectively addressing the category imbalance issue in the detection process and enhancing the model's generalization ability and accuracy.

Additionally, by comparing the training loss with the validation loss, the data suggest that the loss in the validation set converges more effectively than in the training set, indicating that the new model performs better in predicting the unknown detection dataset. This demonstrates the model's generalization ability and resilience to overfitting.

Comparing the results of YOLOv8n with the proposed model in terms of efficiency and complexity features, as shown in Table 1, it is evident that the detection accuracy remains unaffected mainly during the training and prediction process of the improved algorithm. The mAP@50-95 fluctuates within a range of only 0.01. Additionally, the training parameters of the new model are reduced by more than half, which can be attributed to the integration of star operations into the backbone of the YOLOv8 model, enhancing the model's ability to capture fine-grained features when processing complex data. The number of floating-point operations (FLOPs) is improved by 48% as a result of replacing the conventional convolution network with FasterNet and incorporating partial convolution. This modification results in the FLOPs of PConv being lower than regular Conv and higher than DWConv/GConv. The design of PConv reduces redundant computations, thereby improving the computational speed and efficiency of the new model. Furthermore, the model size is reduced by more than half, from 6MB in YOLOv8 to 2.6MB, due to the lightweight improvements in the detection head. These include the simplified backbone network design, GroupNorm instead of BatchNorm, the shared convolution mechanism that reuses convolution kernel parameters, and the reduction of unnecessary parameters during training and prediction in the new model.

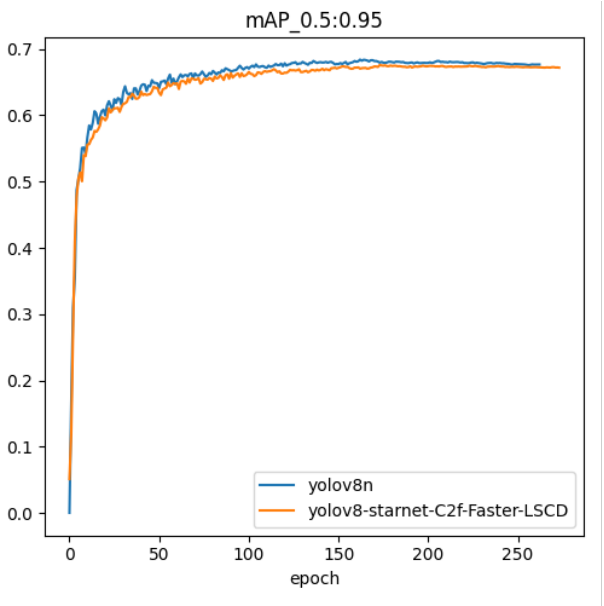

Fig. 7. mAP 0.5:0.95 comparison curve

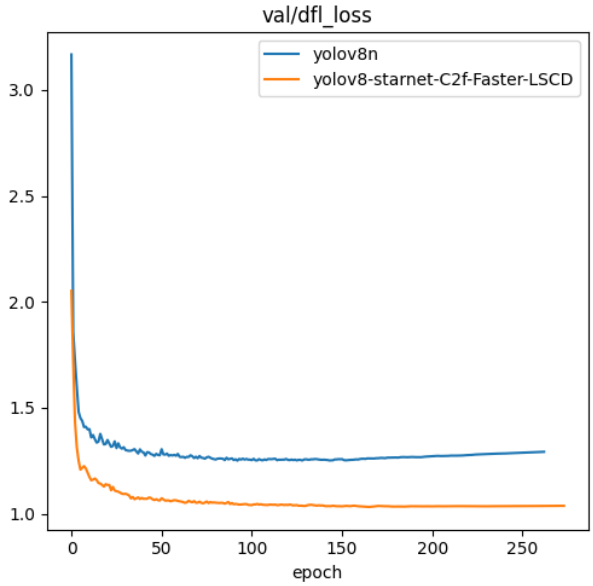

Fig. 8. DFL loss comparison curve

E. Improvement of backbone analysis

This section conducts a comparative experiment involving various backbone improvements to evaluate the impact of StarNet operation on the lightweight enhancement of the YOLOv8 backbone network. Several effective, lightweight models, with YOLOv8 as the baseline, have been proposed recently and are selected for comparison. To ensure the fairness and reliability of the experiments, all models are

TABLE II
COMPARISON EXPERIMENT RESULTS OF BACKBONE NETWORK

structure	P	R	mAP50	map50-95	ParamsMB	Gflops	Size
YOLOv8n	0.974	0.967	0.988	0.684	3.8	8.1	6
YOLOv8n + EfficientViT	0.973	0.961	0.987	0.678	4	9.4	8.4
YOLOv8n + Fasternet	0.971	0.966	0.988	0.679	5.7	22.5	8.2
YOLOv8n + Mobilenetv4	0.98	0.967	0.988	0.68	2.4	6.9	11.2
YOLOv8n + HGNetV2	0.972	0.968	0.989	0.681	2.4	6.9	4.8
YOLOv8n + GhostHGNetV2	0.976	0.972	0.988	0.681	2.3	6.8	4.7
YOLOnv8n + StarNet	**0.977**	**0.969**	**0.987**	**0.674**	**2.2**	**6.5**	**4.5**

trained and evaluated under identical conditions. The detailed experimental results are presented in Table 2.

Based on the high-quality characteristics of the aforementioned public datasets, the training and prediction accuracy of the different backbone models generally remains between 0.97 and 0.98, with no more than a 1% difference. The mAP@50-95 fluctuates between 0.674 and 0.684, showing no more than 1.5% variance. All of the improved backbone models maintain high detection accuracy. Among them, MobileNetV4 has the highest number of parameters, leading to increased model complexity and a significant burden on the inference and prediction process. FastENet and EfficientViT require considerable computational resources due to their large number of parameters, but their detection accuracy does not show a notable improvement over the baseline YOLOv8n. In contrast, HGNetV2 and the model presented in this paper reduce the computational parameter count, resulting in a 16% decrease in computational load compared to the original model. The F1 curves for YOLOv8n and the model in this paper are shown in Fig. 9. In conclusion, the model proposed in this study demonstrates superior prediction performance, simplifies the network model's complexity while ensuring higher detection accuracy, maximizes the backbone network's detection capabilities, and increases feature extraction efficiency. Furthermore, the model size is reduced by 25%, achieving lightweight optimization without compromising performance.

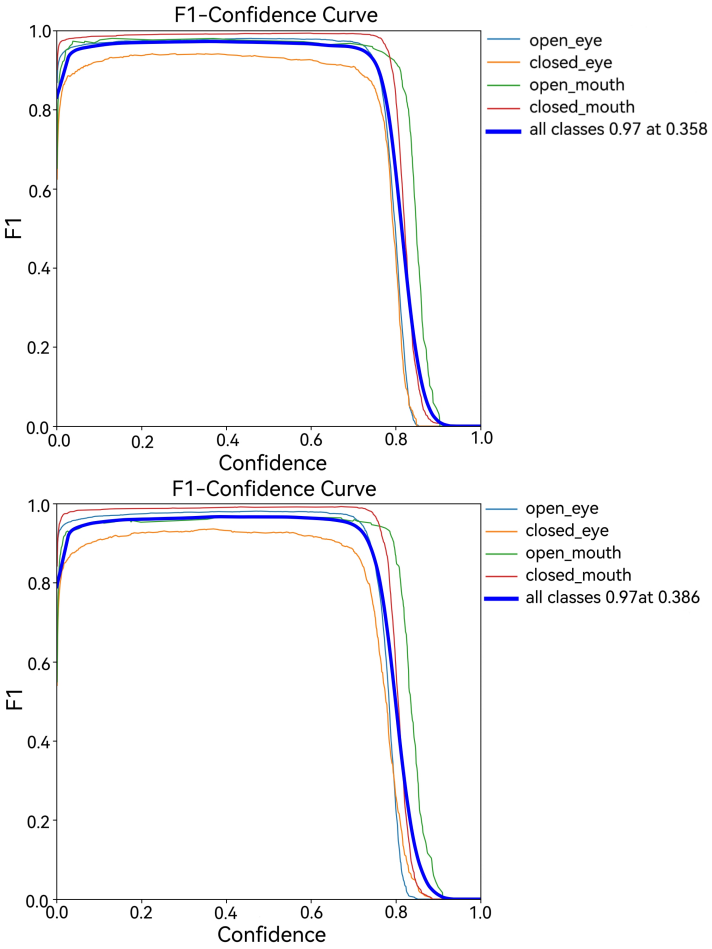

Fig. 9. Comparison of F1 curves of YOLOv8n and YOLOv8-Starnet

F. Ablation experiments

To evaluate the effectiveness of each improvement module on the primary algorithmic model presented in this paper, YOLOv8n is used as the baseline model. Ablation experiments are conducted by incrementally adding or removing various improvement strategies to assess the impact of each module on the experimental performance. Eight test configurations are carried out, starting with the original YOLOv8n model and progressively incorporating the StarNet, C2f-Faster, and LSCD modules. The experimental results are presented in Table 3.

The experimental results demonstrate that the introduction of each improvement module enhances the model's efficiency and complexity characteristics to varying degrees. Experiment 2 introduces the StarNet operation to the backbone network model, achieving high-dimensional nonlinear mapping. This operation balances computational complexity and performance, reducing the model's parameter count from 2.9M to 2.1M and decreasing the computational load by 20%. In Experiment 3, the Bottleneck module in the C2f module is replaced with the FasterNet module. Although accuracy is slightly reduced by only 1.3%, the model's computation is reduced by 22.2%, and its size is reduced by 23.3%, demonstrating that C2f-Faster enhances computational efficiency by optimizing depth-separated convolutions for more efficient memory access. Experiment 4 introduces the LSCD module, which decreases the accuracy and significantly reduces the number of parameters and computational load, thus improving computational efficiency. Experiment 6 verifies the combined effectiveness of C2f-Faster and LSCD, resulting in a 0.2 percentage

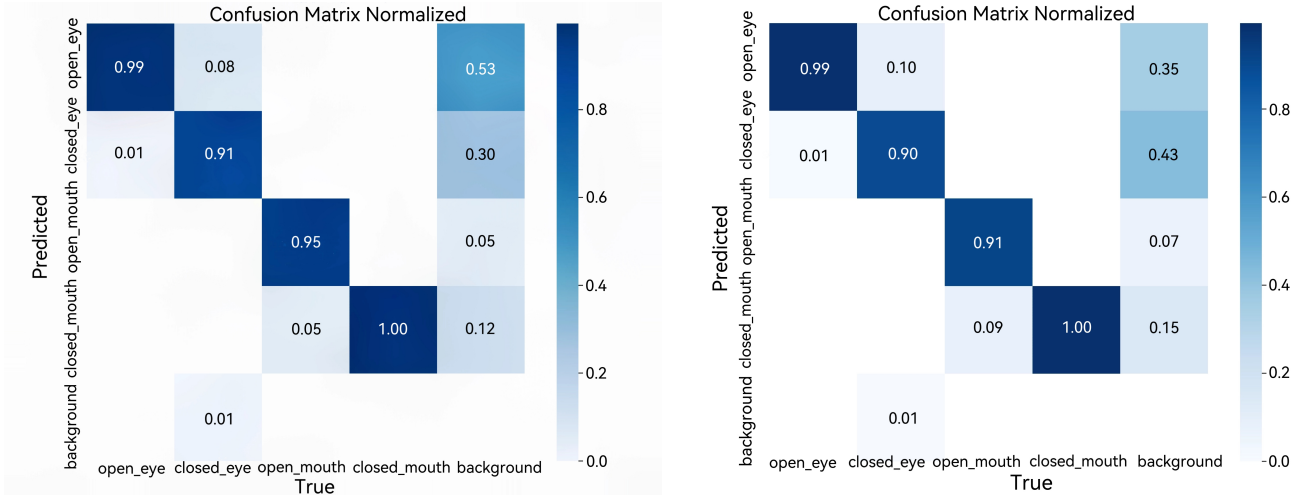

Fig. 10. Comparison of confusion matrices: YOLOv8n model (left) and the model in this paper (right).

point improvement in accuracy, compensating for the slight accuracy loss incurred when used individually. Additionally, the number of parameters is reduced by 44.8%, the computational load by 40.7%, and the model size is halved. In Experiment 7, an improved loss function compensates for the minor accuracy loss introduced by LSCD. This modification results in a nearly halved parameter count and significant reductions in model size. Finally, the improved model achieves a 0.1% increase in accuracy with a 1.5 percentage point reduction in mAP50-95. Furthermore, the number of parameters, computational load, and model size are reduced by 58.6%, 48.1%, and 56.7%, respectively. The confusion matrix of YOLOv8n and the model proposed in this paper is shown in Fig. 10. In contrast, the training results for the proposed model are displayed in Fig. 11. Overall, the new model improves performance, running speed and storage requirements, while maintaining high detection accuracy.

G. Comparative experiments

In order to verify the effectiveness of the improved algorithm in this paper, this section will make experimental comparisons with the performance of mainstream detection models in recent years under the same parameter conditions and uniform training environment, mainly YOLOv3-tiny, YOLOv5n, YOLOv6n, YOLOv7-tiny, YOLOv9t, RT-DERT-r18, YOLOv9s, YOLOv10n, YOLOv11n, and the specific experimental results are shown in Table 4.

The SF-YOLOv8 (ours) model achieves the highest mAP50 of 0.988, comparable to the YOLOv8n, YOLOv8t, and other versions such as YOLOv4 and YOLOv5 in this comparison, which also show high detection accuracy in the 0.98 range. However, the mAP50-95 for SF-YOLOv8 is 0.674, which, while competitive, shows a slight reduction compared to the best-performing YOLOv8n (0.684) and YOLOv7-tiny (0.655). This suggests that while the SF-YOLOv8 model maintains high overall accuracy, it may exhibit slightly reduced performance in more challenging detection scenarios involving harder-to-detect objects (i.e., mAP50-95). One of the most significant advantages of the SF-YOLOv8 model lies in its parameter count and model size. It achieves an exceptionally low number of parameters (1.2 million) compared to other models in the comparison, especially when contrasted with the larger models like YOLOv5n (1.7M) and YOLOv4 (9.2M) or the much heavier RT-DETR-r18 (19M). This parameter reduction indicates that SF-YOLOv8 successfully achieves a lightweight model without sacrificing accuracy, making it suitable for real-time applications where model size is crucial. This is particularly important for deployment on resource-constrained devices such as embedded systems or mobile platforms. The SF-YOLOv8 model also excels in computational efficiency, with a GFLOPs value of 4.2,

TABLE III
RESULTS OF THE ABLATION EXPERIMENT

StarNet	C2f-Faster	LSCD	mAP@0.5	Params	GFLOPs	Size/M
	Baseline		0.988	2.9	8.1	6.0
✓			0.987	2.1	6.5	4.5
	✓		0.988	2.2	6.3	4.6
		✓	0.988	2.3	6.5	4.7
✓	✓		0.988	1.8	5.8	3.8
✓		✓	0.989	1.5	4.9	3.2
	✓	✓	0.987	1.6	4.8	3.4
✓	✓	✓	**0.988**	**1.2**	**4.2**	**2.6**

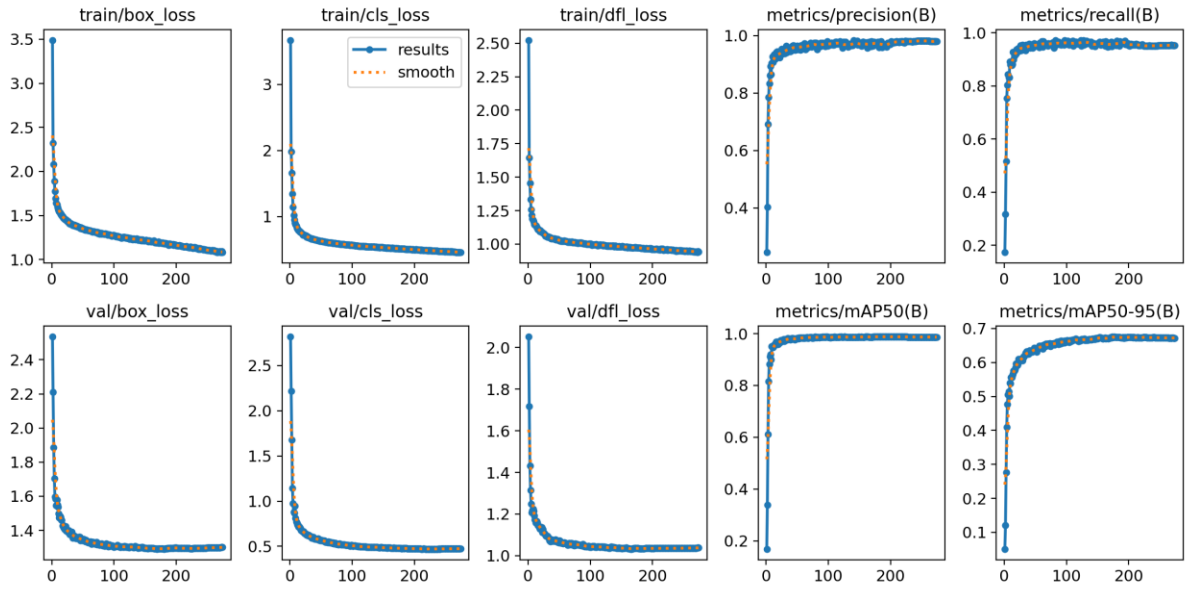

Fig. 11. Training results of ablation experiments for the model in this paper

TABLE IV
COMPARATIVE EXPERIMENTAL RESULTS

Model	mAP50	mAP50-95	Params	GFLOPs
yolov3-tiny	0.986	0.648	8.3	12.9
yolov5n	0.853	0.54	1.7	4.1
yolov6n	0.986	0.649	4.63	11.34
yolov7-tiny	0.988	0.655	5.7	13.0
yolov8n	0.988	0.684	2.9	8.1
yolov9t	0.988	0.679	2.5	11.7
yolov9s	0.988	0.685	9.2	38.7
yolov10n	0.987	0.683	2.2	6.5
yolov11n	0.987	0.68	2.5	6.3
RT-DETR-r18	0.965	0.649	19	57.0
SF-YOLOv8 (ours)	**0.988**	**0.674**	**1.2**	**4.2**

significantly lower than many other models. For instance, RT-DETR-r18 requires 57 GFLOPs, and even YOLOv8n needs 8.1 GFLOPs, indicating that SF-YOLOv8 is not only smaller in terms of parameters but also much more efficient regarding computational cost. This is critical for real-time applications, where processing speed and energy consumption are essential. When comparing SF-YOLOv8 to smaller models like YOLOv3-tiny and YOLOv5n, SF-YOLOv8 outperforms them in terms of both detection accuracy (mAP50) and computational efficiency (GFLOPs). For example, YOLOv3-tiny has a similar mAP50 (0.986) but requires significantly more computational resources (12.9 GFLOPs) and has a larger model size (8.3M parameters). Similarly, YOLOv5n performs well on mAP50 (0.853) but with considerably lower computational demand and a smaller parameter size.

H. Comparative experiment on heat map visualization

To more intuitively demonstrate the model's attention to different regions when making decisions and further verify the reliability of the detection results, this section presents a heat map visualization comparison experiment between the baseline and proposed models. The specific experimental results are shown in Fig. 12, which includes five groups. The first visualization map represents the original image, the second shows the detection results of YOLOv8n, and the third illustrates the detection results of the proposed model. The red and yellow regions in the heat map indicate areas that significantly contribute to the detection results, with the model focusing primarily on these regions. In contrast, the green and blue regions represent areas with minimal contribution to the detection results, where the model's attention is not focused during decision-making.

As shown in figure groups (a) and (b), under different lighting conditions, the attention area of YOLOv8n is less concentrated compared to the model proposed in this paper. The focusing range of our model is more effective in detecting the driver's facial features. Figure group (c) shows that YOLOv8n fails to capture the relevant focus area when detecting yawning. At the same time, the proposed model successfully handles the relationship between the eyes and mouth. In Figure groups (d) and (e), when detecting different angles of eyeglasses, YOLOv8n exhibits a vast and scattered detection area, leading to inaccurate face detection.

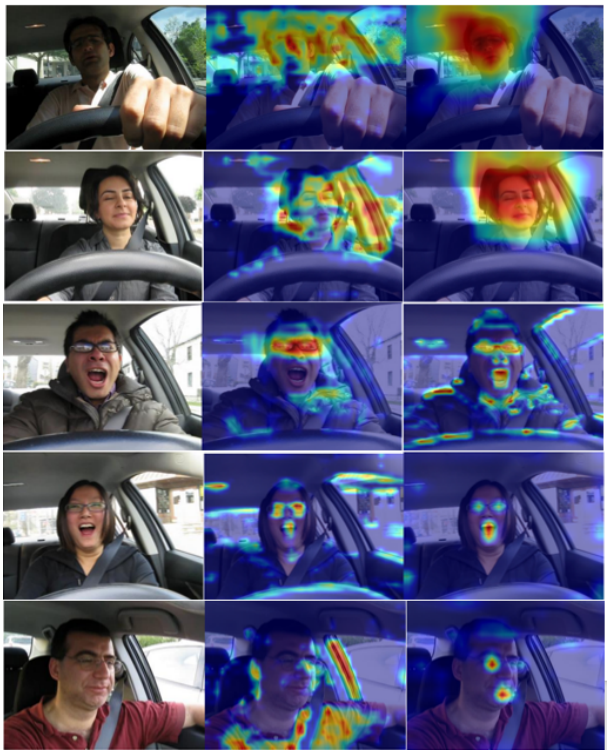

Fig. 12. Comparison results of heat map

In contrast, the proposed model demonstrates superior performance, with the detection area focusing precisely on the facial features. In conclusion, the improved model presented in this paper is better at concentrating on key facial features, effectively minimizing interference from irrelevant environmental factors, reducing computational redundancy, and maintaining high accuracy.

IV. Conclusions

To address the challenges of high complexity and poor real-time performance in driver fatigue detection algorithms, this paper proposes an improved lightweight algorithmic model, YOLOv8-Starnet-C2f-LSCD, based on the YOLOv8n baseline model. First, the latest StarNet operation replaces part of the backbone module, effectively expanding the high-dimensional feature space through multi-layer stacked learning without increasing network computational complexity. Second, further enhancements are made to the C2f module by integrating the FasterNet network model into its neck structure, which improves target detection efficiency and speed. Finally, a redesigned lightweight shared convolutional detection head, LSCD, is employed. This design ensures the effective fusion of feature information while maintaining detection accuracy and reducing computational and arithmetic complexity. Experimental results demonstrate that, compared to the original YOLOv8n baseline model, the proposed model reduces the number of parameters by 59.2%, computational load by 48.2%, and model size by 56.%, achieving a lighter model with higher detection accuracy while ensuring that the mAP50% reaches 0.988. However, the proposed method still requires further improvements under complex driving conditions. To address the current model's limitations, future research will focus on integrating multiple types of fatigue data, enhancing the model's generalization ability under complex scenarios, and exploring more advanced model architectures.

References

[1] Y. Jeon, B. Kim, and Y. Baek, "Ensemble cnn to detect drowsy driving with in-vehicle sensor data," *Sensors*, vol. 21, no. 7, p. 2372, 2021.

[2] A. Sedik, M. Marey, and H. Mostafa, "An adaptive fatigue detection system based on 3d cnns and ensemble models," *Symmetry*, vol. 15, no. 6, p. 1274, 2023.

[3] F. Makhmudov, D. Turimov, M. Xamidov, F. Nazarov, and Y.-I. Cho, "Real-time fatigue detection algorithms using machine learning for yawning and eye state," *Sensors*, vol. 24, no. 23, p. 7810, 2024.

[4] A. Sedik, M. Marey, and H. Mostafa, "Wft-fati-dec: enhanced fatigue detection ai system based on wavelet denoising and fourier transform," *Applied Sciences*, vol. 13, no. 5, p. 2785, 2023.

[5] G. Li, L. Zhang, Y. Zou, D. Ouyang, Y. Yuan, Q. Lian, W. Chu, and G. Guo, "Driver vigilance detection based on limited eeg signals," *IEEE Sensors Journal*, vol. 23, no. 12, pp. 13 387–13 398, 2023.

[6] W.-L. Zheng, K. Gao, G. Li, W. Liu, C. Liu, J.-Q. Liu, G. Wang, and B.-L. Lu, "Vigilance estimation using a wearable eog device in real driving environment," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 1, pp. 170–184, 2019.

[7] L. Wang, J. Li, and Y. Wang, "Modeling and recognition of driving fatigue state based on rr intervals of ecg data," *Ieee Access*, vol. 7, pp. 175 584–175 593, 2019.

[8] S. Das, S. Pratihar, B. Pradhan, R. H. Jhaveri, and F. Benedetto, "Iot-assisted automatic driver drowsiness detection through facial movement analysis using deep learning and a u-net-based architecture," *Information*, vol. 15, no. 1, p. 30, 2024.

[9] A. K. Biswal, D. Singh, B. K. Pattanayak, D. Samanta, and M.-H. Yang, "Iot-based smart alert system for drowsy driver detection," *Wireless communications and mobile computing*, vol. 2021, no. 1, p. 6627217, 2021.

[10] M. Dua, Shakshi, R. Singla, S. Raj, and A. Jangra, "Deep cnn models-based ensemble approach to driver drowsiness detection," *Neural Computing and Applications*, vol. 33, pp. 3155–3168, 2021.

[11] Z. Liu, Y. Peng, and W. Hu, "Driver fatigue detection based on deeply-learned facial expression representation," *Journal of Visual Communication and Image Representation*, vol. 71, p. 102723, 2020.

[12] J. Tang, W. Zhou, W. Zheng, Z. Zeng, J. Li, R. Su, T. Adili, W. Chen, C. Chen, and J. Luo, "Attention-guided multi-scale convolutional neural network for driving fatigue detection," *IEEE Sensors Journal*, 2024.

[13] Y. Zhou, "Iyolo-nl: An improved you only look once and none left object detector for real-time face mask detection," *Heliyon*, vol. 9, no. 8, 2023.

[14] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3," in *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*. IEEE, 2019, pp. 1–6.

[15] J. C. e. a. F. Zhu, J. Chen, "Driver fatigue detection based on improved yolov3," *Science, Technology and Engineering*, no. 022-008, 2022.

[16] T. Bao, Y. Zhao, S. A. R. Zaidi, S. Xie, P. Yang, and Z. Zhang, "A deep kalman filter network for hand kinematics estimation using semg," *Pattern Recognition Letters*, vol. 143, pp. 88–94, 2021.

[17] Z. Zhang, Y. Lu, Y. Zhao, Q. Pan, K. Jin, G. Xu, and Y. Hu, "Ts-yolo: an all-day and lightweight tea canopy shoots detection model," *Agronomy*, vol. 13, no. 5, p. 1411, 2023.

[18] W. Z. Z. Li, "Fatigue driving detection based on improved yolov4 algorithm," *Electronic Measurement Technology*, no. 044-013, 2021.

[19] B. Li, C. Wang, J. Wu *et al.*, "Improved yolov4 algorithm for surface defect detection on aero-engine components," *Laser Optoelectron. Prog.*, vol. 58, no. 14, pp. 414–423, 2021.

[20] Y. Hua, Z. ZHANG, S. Long, and Q. Zhang, "Remote sensing image target detection based on improved yolo algorithm," *Electronic Measurement Technology*, vol. 43, no. 24, pp. 87–92, 2020.

[21] S. S. e. a. D. Ran, S. He, "Tracking algorithm for fatigue driving detection by fusing feature enhancement and deepsort," *Overseas Electronic Measurement Technology*, vol. 42, no. 8, 2023.

[22] T. Zhang, J. Zhou, W. Liu, R. Yue, J. Shi, C. Zhou, and J. Hu, "Sn-cnn: A lightweight and accurate line extraction algorithm for seedling navigation in ridge-planted vegetables." *Agriculture; Basel*, vol. 14, no. 9, 2024.

[23] J. Cao, Y. Li, M. Sun, Y. Chen, D. Lischinski, D. Cohen-Or, B. Chen, and C. Tu, "Do-conv: Depthwise over-parameterized convolutional layer," *IEEE Transactions on Image Processing*, vol. 31, pp. 3726–3736, 2022.

[24] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime

tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.
[25] Q. Zhang, Q. Chen, W. Xu, L. Xu, and E. Lu, "Prediction of feed quantity for wheat combine harvester based on improved yolov5s and weight of single wheat plant without stubble," *Agriculture*, vol. 14, no. 8, p. 1251, 2024.
[26] M. Ma, "Fatigue driving state detection method based on lightweight yolov7-tmc network," *Computer and Information Technology*, vol. 32, no. 6, pp. 6–11, 2024.
[27] X. S. K. Zheng, "Improved yolov8 fatigue driving detection algorithm based on bfds," *Information Technology and Informatisation*, no. 2, pp. 213–216, 2024.
[28] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9308–9316.
[29] X. Ma, X. Dai, Y. Bai, Y. Wang, and Y. Fu, "Rewrite the stars," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5694–5703.
[30] J. Chen, S.-h. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H. G. Chan, "Run, don't walk: chasing higher flops for faster neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 12 021–12 031.