DISTRACTED DRIVING BEHAVIOR DETECTION BASED ON CCLA-YOLOv8

Weihua LIU[1]*, Lele LV[2], Wei SHENG[1]

With the acceleration of urbanization and the continuous growth of vehicle ownership, traffic safety issues caused by distracted driving behaviors have become increasingly prominent. To address this problem, this paper focuses on the research and design of a driver behavior detection system based on the YOLOv8 algorithm. By introducing the Compact Inverted Block (CIB) and the Large Separable Kernel Attention (LSKA) module, the original YOLOv8 model was enhanced. The improved model was subsequently trained using the State Farm dataset from the Kaggle platform. Experimental results demonstrate that the optimized model achieved significant lightweighting and performance improvements: parameter count decreased from 5.95M to 4.59M, the computational load reduced from 8.9 GFLOPs to 6.4 GFLOPs, and the precision increased to 98.4%. Compared with the original model, the enhanced YOLOv8 exhibits notable advantages in detecting distracted driving behaviors. Finally, a driver behavior detection system was designed to meet practical application requirements, aiming to monitor and regulate distracted behaviors in real-time during driving, thereby reducing traffic accident rates.

Keywords: distracted driving behavior; Compact Inverted Block (CIB); YOLOv8; Large Separable Kernel Attention (LSKA)

1. Introduction

By the end of June 2025, China had become the world's largest automotive market, with national motor vehicle ownership reaching 460 million, including 359 million automobiles (36.89 million new-energy vehicles), there were 550 million motor vehicle drivers, among which 515 million were automobile drivers. This growth drives economic development but also brings severe traffic safety challenges. In 2021, there were 273,000 road accidents in China, over 60% of which were vehicle - related collisions [1], causing 62,000 fatalities and ¥1.45 billion in property losses. Research shows distracted driving (e.g., smoking or using a phone) is a major factor. The YOLOv8 algorithm, with its object - detection advantages, is an optimal solution for driver behavior monitoring [2].

Driver behavior detection, a core intelligent transportation technology, has seen global progress in algorithm innovation and scenario - specific applications [3]. Balamurugan M's team (2021) embedded regularization modules in

[1] School of Artificial Intelligence, Gansu University of Political Science and Law, Lanzhou 730070, China, e-mail: burengn32@163.com
[2] Anning Sub - branch, Lanzhou Branch, Industrial and Commercial Bank of China, Lanzhou 730070, China

convolutional networks with dynamic feature distillation, raising transient behavior recognition (e.g., hands - off steering wheel) completeness to 91.4% [4]. Zakaria Boucetta et al. (2022) combined Local Weber Patterns (LWP) and Gradient Patterns (LGP) for a multimodal network, using optimization algorithms to adjust LSTM parameters and achieving 93.6% classification accuracy in complex backgrounds [5]. Domestically, there are also breakthroughs. Li Shaofan et al. (2023) designed a YOLO - Ghost architecture with MobileNetv3 for multimodal decision - making, cutting false alarms in rain/fog to 6.2% [6]. Zheng Meirong's team (2024) combined AHP, K - means clustering, and BP neural networks to create a weighted scoring model for driving behavior features [7].

This study aims to develop a real - time driving behavior monitoring system integrating an optimized YOLOv8 model with a PyQt5 interface for high accuracy and user-friendliness [8]. It offers technical support to reduce traffic accident risks through real-time monitoring and analysis of driver behaviors. Algorithmic optimization of YOLOv8 is based on driving scenario characteristics. The Compact Inverted Block (CIB) replaces the original Bottleneck in the C2f module, creating a new C2fCIB module [9]. Also, the Large Separable Kernel Attention (LSKA) module is integrated to enhance accuracy and reduce parameters, enabling the model to adapt to in - vehicle edge computing.

2. YOLOv8 model

2.1 The YOLOv8 model

The network architecture of the YOLOv8 model consists of multiple modules. The YOLOv8 Head generally refers to the PANet network backbone, and the Backbone is composed of modules such as C2f, Conv, and SPPF [10].

2.1.1 PANet Network Backbone

The network structure of PANet (Fig. 1) has five core modules: (a) is an FPN (Feature Pyramid Network), (b) is a bottom - up feature fusion layer added by PANet, (c) is an adaptive feature pooling layer, (d) is the bounding box prediction head of PANet, and (e) is a fully - connected fusion layer for mask prediction. PANet's key contribution is the bottom - up path, transferring high - resolution location info from shallow layers to deep - layer features. Its bidirectional feature fusion enhances info interaction between different - level features, helping the model handle various - scale targets in object detection, boosting accuracy and robustness. It excels in small - target and precise - location tasks, which is why it's used in the YOLOv8 model.

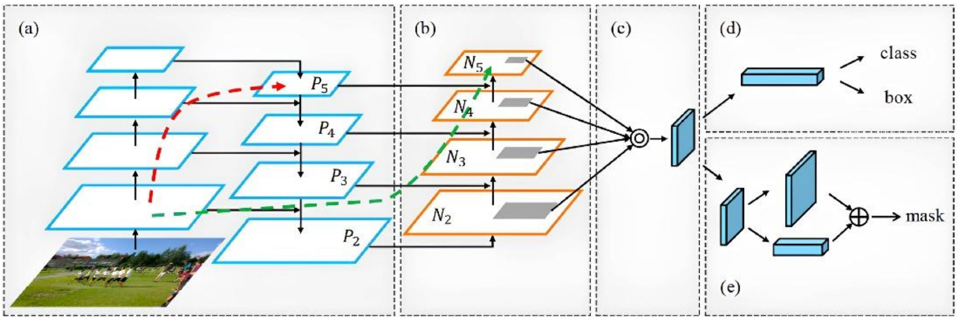


Fig. 1. PANet Network Backbone

2.1.2 Conv Module

The Conv module serves as the core computational unit in deep neural networks. Its standard forward propagation path involves a three - stage linkage mechanism of "convolutional layer → batch normalization → dynamic activation". The collaborative working process of the two paths is shown in Fig. 2.

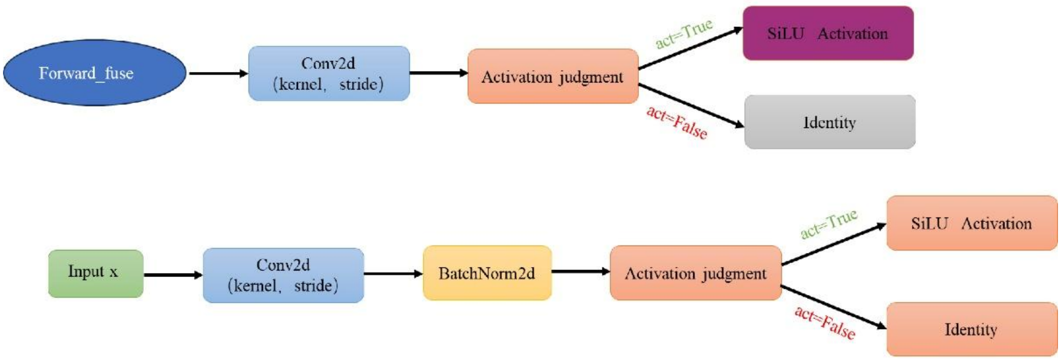


Fig. 2. Flowchart of the working mode of the Conv module

This module extracts spatial features through the Conv2d layer to capture image information. The kernel parameter determines the convolution kernel size, and the stride parameter controls the feature map down - sampling rate. The BatchNorm2d layer normalizes the output, stabilizing training, accelerating convergence, and enhancing model stability. The activation function is chosen according to the act parameter. When act = True, the SiLU function adds non - linearity; when act = False, it uses identity mapping. The autopad function calculates padding to keep input and output feature map sizes the same.

For efficient deployment, the Conv module has a forward fuse path, skipping BatchNorm2d calculations to reduce model complexity. This dual - mode design, along with the SiLU's continuous differentiability, ensures learning and stability during training and meets real - time inference requirements in applications like YOLOv8 [11].

2.1.3 C2f Module

As shown in Fig. 3, the C2f module includes a convolutional layer, bottleneck structure, feature concatenation, and compression. First, Conv1 (usually 1x1 convolution) doubles input channels to enhance feature representation [12]. Then, multiple Bottleneck modules with configurable residual connections extract deep - level features. Finally, outputs of each Bottleneck are concatenated with original features, and the channel number is compressed via Conv2 for the target output. Through staged processing, it combines residual connections and feature fusion, balancing flexibility and efficiency.

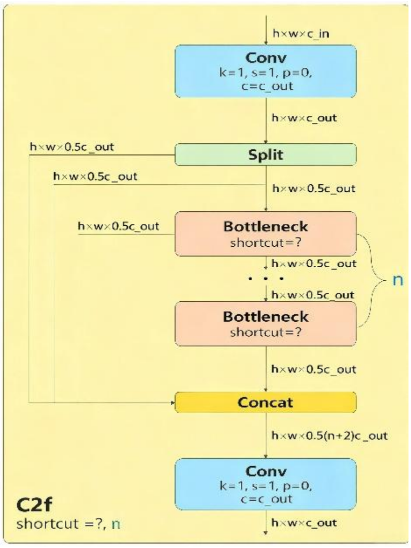


Fig. 3. Flowchart of the C2f module

The C2f module's core function is multi - scale feature aggregation and model lightweighting. It concatenates outputs of different - depth Bottlenecks with original features, integrating local details and global semantics. Two convolutions (for expansion and compression) optimize computational efficiency, reducing parameters while preserving representation ability. This design enhances the model's pattern - capturing ability and adapts to various tasks via channel adjustment.

2.1.4 SPPF Module

The SPPF (Spatial Pyramid Pooling - Fast) in YOLOv8 is designed to perform pooling on different sizes. Specifically, a 1x1 convolution is used to reduce the dimension to half of the original size [13]. Then, three consecutive nn. MaxPool2d operations with a kernel size of 5 are applied. After that, the output feature maps are concatenated along the channel dimension. Finally, a convolution layer is used to fuse the features of different scales and generate the final output feature map. Structural diagram of the SPPF module is shown in Fig.4.

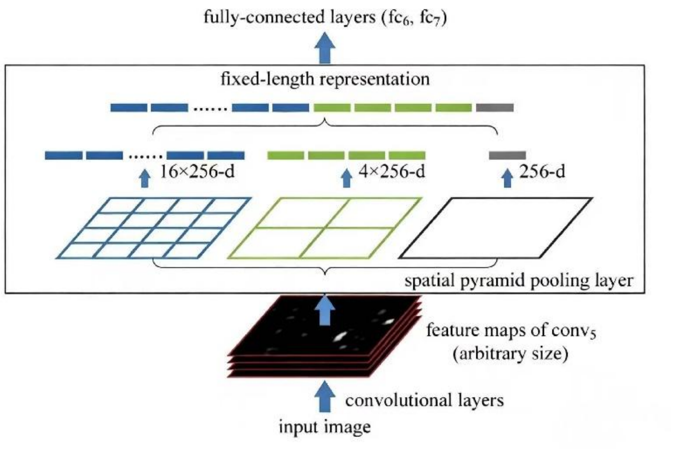


Fig.4. Structural diagram of the SPPF module

3. Improvement of the YOLOv8 Model

3.1 C2fCIB Module

The Compact Inverted Block (CIB) structure is incorporated into YOLOv8 to substitute the original Bottleneck in the C2f module. Its position and structure in YOLOv8 are presented in Fig. 5.

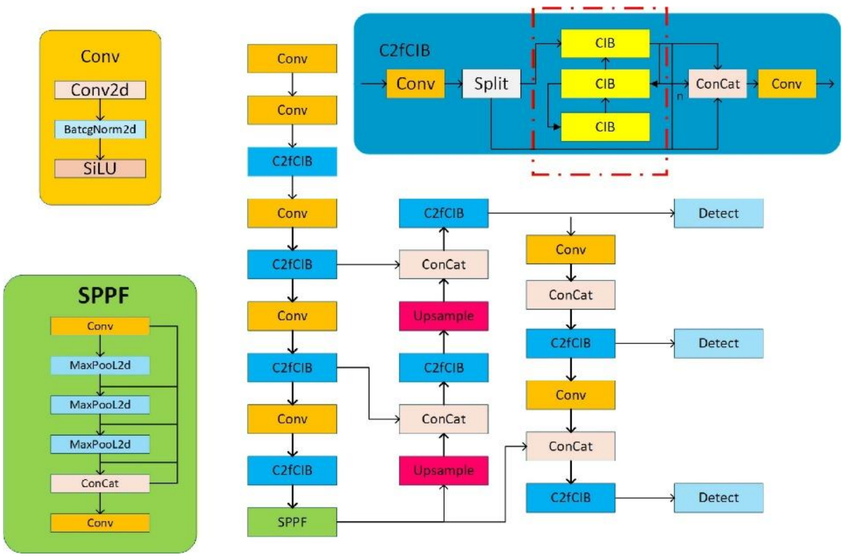


Fig. 5. The position and structural diagram of C2fCIB in YOLOv8

The CIB structure cuts the computational burden and redundancy using depthwise separable convolutions for spatial mixing and pointwise convolutions for channel mixing [14]. In the C2fCIB module, multiple CIB blocks are stacked within the C2f framework, maintaining feature extraction ability while reducing parameter count and computational load. Regarding the number of parameters and computational load, YOLOv8 has 8.9 GFLOPs and 5.95 million parameters, while YOLOv8-C2fCIB has 6.5 GFLOPs (27.0% decrease) and 4.59 million parameters (22.9% decrease).

The lightweight classification head replaces standard convolutions with depthwise separable convolutions, reducing the computational load of the classification head (FLOPs are reduced by 60%). For the spatial - channel decoupled downsampling, the number of channels is first adjusted through pointwise convolutions, and then downsampling is performed using depthwise convolutions.

3.2 LSKA Module

In the YOLOv8 object detection model, to significantly reduce the computational complexity and memory usage while improving the detection accuracy, the Large Separable Kernel Attention (LSKA) module is introduced. Its model structure is shown in Fig. 6.

The LSKA module cleverly decomposes the convolution kernel, splitting the two - dimensional depthwise convolution kernel into a one - dimensional convolution kernel in the horizontal direction (1×K) and a one - dimensional convolution kernel in the vertical direction (K×1). This approach effectively reduces the number of parameters and the computational load of the model [15]. The module supports various configurations of ultra - large - sized convolution kernels, such as 7×7, 23×23, 53×53, etc., and expands the receptive field through dilated convolutions.

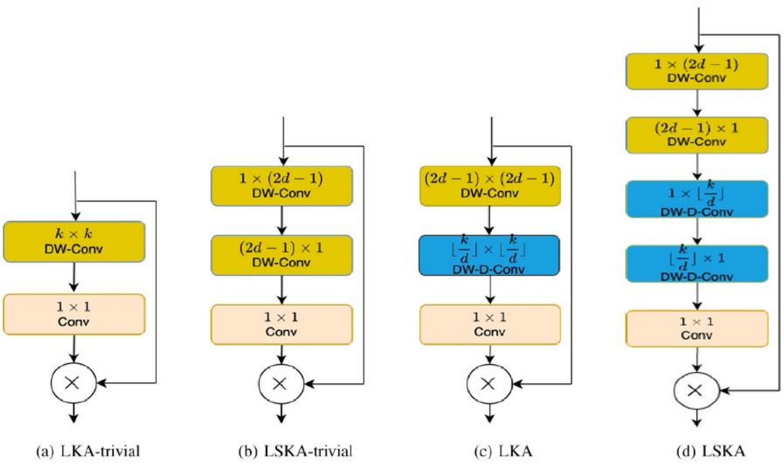


Fig. 6. Structure of the LSKA module

In its attention structure, an activation function (ReLU), a spatial gating unit (i.e., the LSKA module itself), and a residual connection are integrated. By replacing the original C2f module with C2f_LSKA_Attention and introducing the

LSKA module in the Backbone and Head parts, both the computational load (GFLOPs) and memory usage of the model are significantly reduced.

By introducing the C2fCIB module and the Large Separable Kernel Attention (LSKA) module, the C2fCIB - LSKA - YOLOV8 (CCLA - YOLOv8) model is formed.

4. Detection of driving behavior based on the YOLOv8 algorithm

4.1 Experimental Environment and Parameter Configuration

The main hardware configurations and software environment for the experiment are shown in Table 1 and Table 2.

Table 1

Hardware configurations

Component	Model/ Specifications
CPU	Intel Core i7-12700K
GPU	NVIDIA GeForce RTX 4090（24GB graphics memory ）
Camera	Logitech C920 Pro HD（1080p,30fps）

Table 2

Software environment

Software/Library	Version
CUDA	V11.8
cuDNN	V8.6
OpenCV	V4.1.2
PyTorch	V2.2.2

In order to ensure an adequate amount of training data while having sufficient data for model validation, the experimental dataset was randomly divided into a training set and a validation set at a ratio of 8:2. The C2fCIB module and the LSKA module were added to the YOLOv8 model, and 100 epochs of training were conducted to achieve the optimal operating effect. The training parameter configuration is shown in Table 3.

Table 3

Parameter configuration

parameter names	parameter configuration
Batch_Size	16
Epochs	100
weights	YOLOv8s.pt
learning rate	0.001
optimizer	Adam

4.2 Introduction to the dataset

The dataset used in this experiment is the State Farm Distracted Driver Detection dataset, which was released by State Farm on the Kaggle platform. The dataset contains a total of 10,760 photos, with approximately 1,000 photos in each category. A partial display of the dataset is shown in Fig. 7.

Fig. 7. Partial Dataset Illustration

In this dataset, c0 represents normal driving, while the remaining nine categories are dangerous driving behaviors, specifically as follows:c1: Making a phone call with the right hand;c2: Typing with the right hand;c3: Making a phone call with the left hand;c4: Typing with the left hand;c5: Tuning the radio;c6: Drinking something;c7: Talking to other passengers;c8: Reaching for items in the back row;c9: Doing hair and makeup. This dataset exhibits rich category diversity, which can well reflect the diversity of the data and meet the detection requirements.

4.3 Model evaluation

In the evaluation of algorithm performance, commonly used indicators include Precision, Recall, F1-Score, and mAP (mean Average Precision) [16,17].

Precision measures the proportion of samples that are actually positive among those predicted as positive by the model. The higher the precision value, the fewer misjudgments the model makes when predicting positive examples. The calculation formula is shown in Equation (1):

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

Recall measures the proportion of positive samples that are correctly predicted as positive by the model among all actual positive samples. The higher the recall value, the lower the probability that the model fails to detect positive samples. The calculation formula is shown in Equation (2):

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

The F1-Score is calculated based on precision and recall and represents the harmonic mean of these two key indicators. This score ranges from 0 to 1. when the F1-Score reaches 1, it indicates that the performance of the model is in the most ideal state; conversely, if the score is 0, it reflects the worst - performing state of the model. The calculation formula is shown in Equation (3):

$$F1 - Score = \frac{2 \times precision \times recall}{precision + recall} \tag{3}$$

mAP (mean Average Precision) is the average of the Average Precision (AP) for each category. It is used to evaluate the performance of multi - class object detection or classification tasks. mAP comprehensively takes into account the precision performance of the model at different recall rates and is the core indicator for object detection tasks. The calculation formula is shown in Equation (4):

$$mAP = \frac{1}{N} \times (AP_r(0) + AP_r(0.1) + \cdots + AP_r(1.0)) \tag{4}$$

GFLOPs (Giga FLoating-point OPerations) is a core indicator for measuring the computational complexity of neural network models [18]. It reflects the total amount of floating-point operations (in billions) required for a single inference or training iteration of the model. Its calculation formula is shown in Equation 5:

$$GFLOPs = \frac{Total\ FLOPs\ of\ the\ model}{10^9} \tag{5}$$

Among them, FLOPs (Floating - Point Operations) refers to the number of floating - point operations required for the model to complete one forward propagation [19].

Table 4 shows the results of the ablation experiment. By introducing the CIB module and the LSKA module, the CCLA - YOLOv8 model is formed. The comparison of indicators (mAP50, recall, mAP50-95, GFLOPs, precision) demonstrates that while maintaining the detection accuracy basically unchanged (mAP50 slightly increased by 0.00076), the computational load of this model is significantly reduced to 6.4 GFLOPs (a 28.1% reduction compared to the original YOLOv8).

Table 4

Evaluation Table of Indicators for Dangerous Driving Models

Models	mAP50	Recall	mAP50-95	GFLOPs	Parameters(M)	Precision
YOLOv8	0.98385	0.97663	0.80426	8.9	5.95	0.98386

YOLOv8+C2fCIB	0.98331	0.9817	0.80424	6.5	4.59	0.98527
YOLOv8+LSKA	0.98254	0.97892	0.80034	7.4	5.37	0.98127
CCLA-YOLOv8	0.98461	0.9827	0.80329	6.4	4.59	0.98401

Fig. 8 depicts the variation curves of three types of training loss and validation loss: Training/Validation Box Loss, Training/Validation Classification Loss, and Training/Validation DFL Loss. As can be seen from the Fig. 8, in the initial stage of training, specifically within the first 30 epochs, the training losses of all models across the board decline rapidly. After that, the rate of decline slows down, and the losses gradually converge. The optimal result is achieved at the 90th epoch.

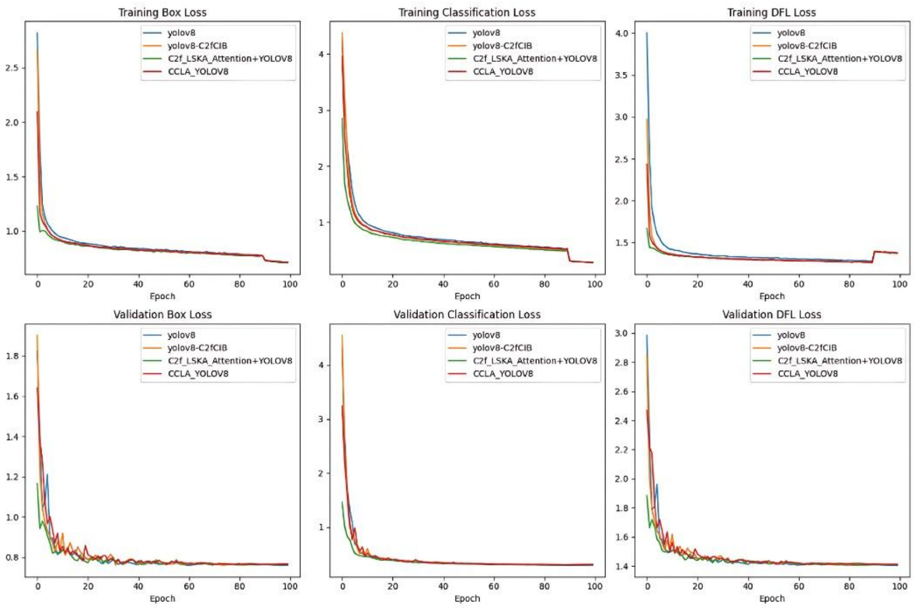
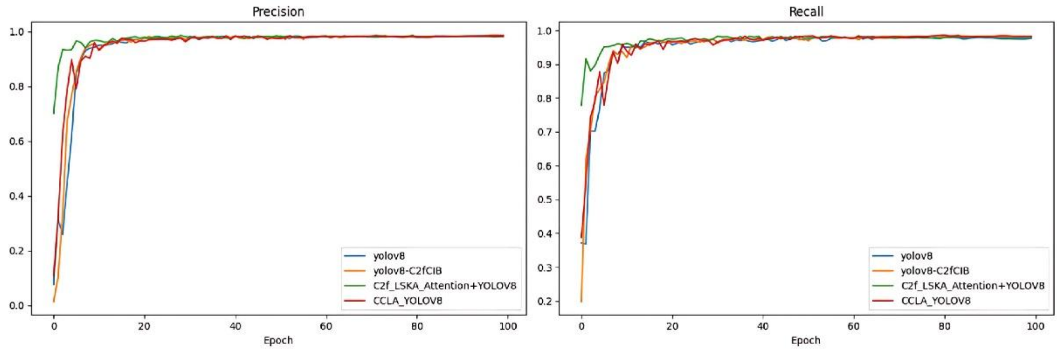


Fig. 8. Comparison chart of Training loss and Validation loss

Fig. 9 shows the precision (Fig. 9 a) and recall (Fig. 9 b) of four models, among these four models, C2f_LSKA_Attention + YOLOv8 and CCLA + YOLOv8 perform best in terms of precision and recall. Especially in the later stage of training, they can maintain a relatively high-performance level. This indicates that introducing specific attention mechanisms into the model can significantly improve the performance of object detection tasks.



(a) Precision (b) Recall

Fig. 9. Precision and Recall

The experimental results comparison of mAP50 - 95 for the four models is shown in Fig. 10. The Fig. 10 show that in the early stage of training (Epoch 0 - 20), the mAP of all models rises rapidly. The green curve (C2f_LSKA_Attention + YOLOv8) rises the fastest and reaches the highest around Epoch 20. From Epoch 20 to Epoch 40, the mAP of each model is still increasing, but the speed slows down. The red curve (CCLA+YOLOv8) gradually approaches the performance of the green curve in this stage. After Epoch 40, the mAP of each model tends to be stable. The green and red curves remain at a high level, while the mAP of the blue curve (yolov8) and the orange curve (yolov8 - C2fCIB) is slightly lower.

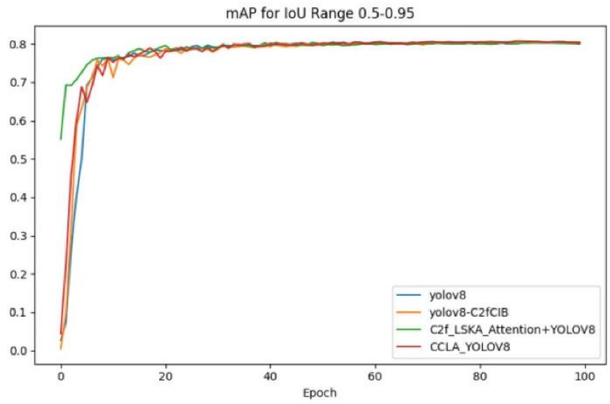


Fig. 10. Comparison Chart of mAP

Fig. 11 shows the F1 - curve of the improved CCLA - YOLOv8 model. By observing the data in the Fig., the F1 - values of all classes are close to 1.0, indicating that when the current model is highly confident in the prediction results, the precision and recall of classification are highly balanced, with almost no misjudgments or missed detections. The legend is marked as "all classes 0.98 at 0.490", which means that when the confidence threshold is set to 0.49, the average

F1 - value of all classes reaches 0.98, which is the optimal balance point between precision and recall, demonstrating that the model performs well.

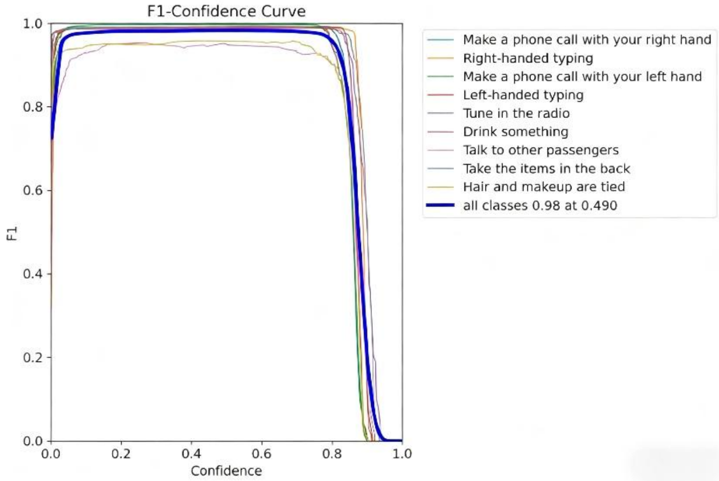


Fig. 11. F1_curve of CCLA - YOLOv8

Through comparative analysis of the above performance indicators, it is fully verified that the improved algorithm has enhanced the detection performance. The experiment proves that the improvement of the YOLOv8 algorithm in this study meets the detection requirements.

5. Visualization experiment design

5.1 Framework design

The system interface uses PyQt5 to build an efficient driver dangerous behavior detection system. This system can monitor and identify potential dangerous behaviors of drivers in real - time and provide an intuitive user interface [20]. It can effectively detect data inputs such as pictures, videos, and camera feeds, process the received data in real - time, display the detection results, issue early warnings for distracted driving behaviors, and record real - time footage [21], which is convenient for subsequent determination of accident liability and helps people pay more attention to safe driving. The modular functional structure of this system is shown in Fig. 12.

The results of the static data test are shown in Fig. 13. The system detects that the person in the image is holding a mobile phone [22]. The confidence level of facial detection is 0.90, and the confidence level of detecting the act of using the right hand to operate the phone is also 0.90, indicating that the system has identified that the driver might be using their right hand to use the phone [23].

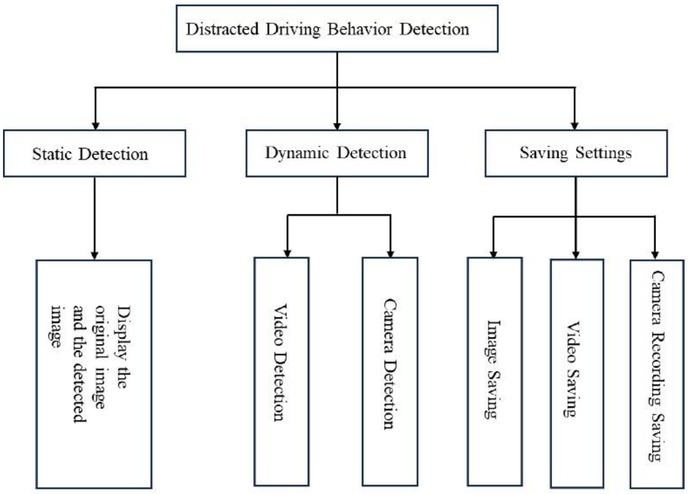


Fig. 12. System framework diagram

Fig. 13. Static detection results

The dynamic data testing module is mainly used for real - time detection and analysis of video files and camera footage. For video detection, the target detection algorithm is executed frame by frame, and the analysis results are displayed in real - time, as shown in Fig. 14.

Fig. 14. Video detection results

The "Camera Detection" function uses a similar logical architecture. The difference lies in that the parameter to be called is changed to the camera device number (usually defaulting to 0)[24]. By activating the camera device, the real - time video stream is obtained for detection [25]. The detection results after clicking to turn on the camera are shown in Fig. 15.

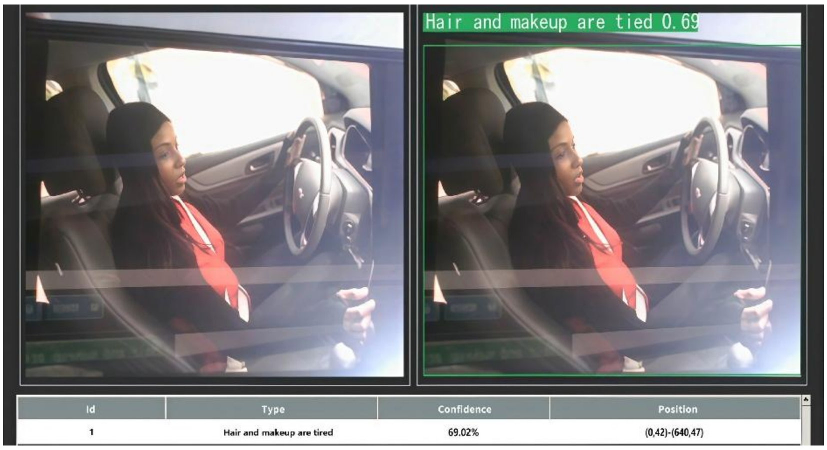


Fig. 15. Real-time camera detection results

During real - time driving behavior detection, whether in the static image detection mode or the dynamic video or camera footage detection mode, the system will meticulously record and intuitively present the detection results [26]. When an unfortunate traffic accident occurs, these detailed and accurate records can serve as crucial evidence for determining accident liability. Meanwhile, this traceability will also encourage drivers to attach more importance to safe driving and reduce the occurrence of dangerous driving behaviors [27,28].

6. Conclusions

This study develops a driver behavior detection system based on the YOLOv8 algorithm. Through real - time monitoring and accurate identification of drivers' potentially dangerous behaviors, this system significantly enhances road traffic safety. The system adopts an improved YOLOv8 object detection algorithm, with improvements made in two aspects. Firstly, the Compact Inverted Block (CIB) structure is introduced into YOLOv8 to replace the Bottleneck layer in the original C2f module, thus forming a new C2fCIB module. Secondly, the Large Separable Kernel Attention (LSKA) module is introduced. The improved model achieves significant lightweighting and performance enhancement: the number of parameters is reduced from 5.95 million to 4.59 million, the computational load decreases from 8.9 GFLOPs to 6.4 GFLOPs (a 28.1% reduction), while the precision increases to 98.4%. These improvements, combined with the proposed static and dynamic data detection framework, enable real-time monitoring of driving behaviors in vehicle-mounted systems, effectively reminding drivers to pay attention to safety.

Acknowledgement

This work was supported by the Gansu Provincial Department of Education University Teachers' Innovation Fund Project: Research on an Auxiliary Prediction Model for Postpartum Hemorrhage Volume Based on Machine Learning Methods, Project No. 2024A – 099; Gansu Provincial Higher Education Institutions' Scientific Research Project: Research on Behavior Recognition Algorithm Based on Feature Interaction, Project No.: 2022A-095.

REFERENCES

[1]. *Lan Zhangli, Fan Liang,Zhang Hong et al.,* Lightweight driving behavior detection network based on attitude assistance[J]. Journal of Chongqing Jiaotong University (Natural Science), 2025, 44(8): 75-82.

[2]. *Yingjie Du, Xiaofeng Liu, Yuwei Yi et al.,* Optimizing Road Safety: Advancements in Lightweight YOLOv8 Models and GhostC2f Design for Real-Time Distracted Driving Detection. Sensors. 2023, 23(21):8844.

[3]. *Gumaei A., Al-Rakhami M., Hassan M.M. et al.,* A deep learning-based driver distraction identification framework over edge cloud. Neural Computing and Applications. 2020: 1-16.

[4]. *Balamurugan M., Kalaiarasi R.*, Regularized Densely Connected Convolutional Network for Effective Real-Time Distracted Driver Detection. Journal of Computational and Theoretical Nanoscience. 2021, 18(1): 502-508.

[5]. *Boucetta, Z.; Djemil, D.E.; Hacine-Gharbi, A.; Ravier, P.,* Distracted Driver Detection Based on a Hand-Crafted Feature Fusion Network and Deep Learning, J. Imaging 2022, 8(7), 193.

[6]. *Li Shaofan, Gao Shangbing, Zhang Yingying et al.,* A Collaborative Detection System for Distracted Behaviors Based on Lightweight Network and Embedded Technology. Journal of Nanjing Normal University (Engineering and Technology Edition), 2023, 23(01): 25 - 32.

[7]. *Zheng Meirong, Hu Jing.* Safety Assessment of Driving Behaviors Based on a Deep Learning Fusion Model. Journal of Yanbian University (Natural Science Edition), 2024, 50(04): 63 - 69.

[8]. *Zhao Xia, Li Chao, Fu Rui, et al.* Real - time Detection of Distracted Driving Behaviors Based on Deep Convolution - Token Dimensionality Reduction Optimized Vision Transformer. Automotive Engineering, 2023, 45(06): 974 - 988 + 1009.

[9]. *Bhakti Vilas Baheti; Sanjay Talbar; Suhas Gajre.* Towards Computationally Efficient and Realtime Distracted Driver Detection with MobileVGG Network. IEEE Transactions on Intelligent Vehicles. Volume PP, Issue 99. 2020. PP 1-1.

[10]. *Hu Lin, Bao Xingqian, Lin Miao, et al.* Research on risky driving behavior evaluation model based on CIDAS real data. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering. Volume 235, Issue 8. 2021. PP 2176-2187.

[11]. *Zhao Shuanfeng, Li Leping, Wang Maoquan, et al.* A Method for Detecting Drivers' Distracted Behaviors Based on Deep Learning. Electronic Measurement Technology, 2024, 47(09): 145 - 153.

[12]. *Zou Peng, Yang Kaijun, Liang Chen.* Real - time Detection of Lightweight Irregular Driving Behaviors with Improved YOLOv5. Computer Engineering and Applications, 2023, 59(13): 186 - 193.

[13]. *Zhong Kejia*. Research on Distracted Driving Behavior Detection Method Based on Deep Learning. Nanchang University, 2023.
[14]. *Jia L., Wei H., Hao L. et al.,* SLAFusion: Attention fusion based on SAX and LSTM for dangerous driving behavior detection. Information Sciences, 2023, 640.
[15]. *Biondi F.N, Horrey W.J., Donmez B.,* Preface to the Special Issue on Assessment and Effectiveness of Driver Monitoring Systems. Human factors. Volume 66, Issue 9. 2023. PP 187208231206073-187208231206073.
[16]. *Yong Peng, Xin Lou, Honggang Wang et al.*, Driving behavior in Hazardous situations: The interplay between risk scenarios and dimensional emotions. Transportation Research Part F: Psychology and Behaviour. Volume 107, Issue. 2024. PP 695-709.
[17]. *Gao Shangbing, Zhang Yingying, Wang Teng et al.* A Lightweight Detection Method for Drivers' Distracted Behaviors Based on BiViTNet. Journal of Chongqing Jiaotong University (Natural Science Edition), 2024, 43(02):57-64.
[18]. *Zhou Hongwei, Wang Wenbo, Wang Weiguang, et al.* Driving Distraction Behavior Detection Model Based on Improved YOLOv7 Algorithm. Automation Technology and Applications. 2024,12.
[19]. *Tian Jing, Cao Jinxuan, Wang Jianing, et al.* Recognition of Distracted Driving Behaviors Based on Improved Siamese Network Algorithm. Journal of People's Public Security University of China (Natural Science Edition), 2025, 31(01):46-53.
[20]. *Pan Jiandong, Fan Chaojie, Jiang Zhuoxi, et,al.* Research on Real - time Detection of Drivers' Distracted Behaviors Based on YOLOv8. Journal of Southwest University (Natural Science Edition).2025,47 (08):38-48.
[21]. *Sajid F., Javed A.R., Basharat A. et al.* An efficient deep learning framework for distracted driver detection[J]. IEEE Access, 2021, 9: 16927-16952.
[22]. *Poon Y S, Kao C.Y., Wang Y K, et al.* Driver distracted behavior detection technology with YOLO-based deep learning networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(11): 19950-19960.
[23]. *Zandamela F., Nicolls F., Kunene D.* Enhancing distracted driver detection with human body activity recognition using deep learning[J]. South African Journal of Industrial Engineering, 2023, 34(3): 142-156.
[24] *Khalil H.A., Hammad S.A., Munim H.E.A., et al.* Low-Cost Driver Monitoring System Using Deep Learning[J]. IEEE Access, 2025, 13: 14151-14164.
[25]. *Omerustaoglu F., Sakar C.O., Kar G.,* Distracted driver detection by combining in-vehicle and image data using deep learning[J]. Applied Soft Computing, 2020, 96: 106608.
[26]. *Ma Y, Du R, Abdelraouf A, et al.* Driver digital twin for online recognition of distracted driving behaviors[J]. IEEE Transactions on Intelligent Vehicles, 2024, 9(1): 2478-2491.
[27]. *Wei Y., Guo Z., Dai C, et al.* Distracted driver behavior detection based on an improved YOLOX framework[J]. IEEE Access, 2022, 10: 13642-13652.
[28]. *Ali F.M.A., Alnuaimi M.J., Alawadhi S.A. et al.,* Abnormal Driver Behavior Detection Using Deep Learning[C]. 2024 7th International Conference on Signal Processing and Information Security (ICSPIS). 2024: 1-4.