The Journal of Supercomputing (2025) 81:622
https://doi.org/10.1007/s11227-025-07129-6

Tp-yolov8: a lightweight and accurate model for traffic accident recognition

Zhaole Ning[1] · Tianze Zhang[2] · Xin Li[1] · Aiying Wu[1] · Gang Shi[1]

Accepted: 25 February 2025

Abstract
Traffic accident detection is an important part of road safety, impacting the lives of those involved and others on the road. Using surveillance cameras on traffic poles to detect accidents poses unique challenges, such as incomplete dataset categories, small-sized detection objects, and the need for lightweight models. Current traffic accident recognition algorithms, while effective in detection, often require extensive resources, making deployment on edge devices difficult. This paper proposes a more accurate and lightweight traffic accident recognition model based on YOLOv8, optimized for traffic pole monitoring and deployment on edge devices. To improve small object detection, we made improvements to the neck. We modified the neck by adding a detection layer for small-sized objects using large-scale feature maps, along with a dedicated small object detection head (SODL-SODH). Additionally, we design a lightweight cross-scale feature fusion module (LCSFFM) to optimize the PAN-FPN structure, reducing model parameters and computational complexity while enhancing small-target detection. In the downsampling layer, we incorporate the squeeze-excited aggregate spatial attention module (SEASAM) into the C2F module to help the network focus on essential image information, with minimal impact on model parameters and computational complexity. To address dataset limitations, we built the traffic accident-type (TAT) dataset for training and evaluation, and validated it against other advanced methods. Experimental results show that our model outperforms the baseline on the TAT dataset, improving the mAP0.5 by 1% and reducing parameters by 25.9%. On the BDD-IW dataset, our TP-YOLOv8s outperforms other methods in terms of accuracy. Compared with the best other methods, it improves the mAP0.5 index by 1.4% and reduces the number of parameters by 84.1%.

Keywords Traffic accident recognition · SODL-SODH · LCSFFM · SEASAM · TAT

Z. Ning, T. Zhang: These authors contributed equally to this work.

Extended author information available on the last page of the article

Published online: 14 March 2025

1 Introduction

With globalization and urbanization, the rapid growth of transportation systems has brought significant social and economic benefits. However, along with the continuous increase in the number of motor vehicles, the number of traffic accidents continues to increase. Traffic accidents have gradually become a major public safety issue worldwide. Traffic accidents not only cause a large number of casualties and property losses, but also bring huge economic burden to society. In order to effectively reduce the occurrence of traffic accidents and improve the level of road traffic safety, intelligent transportation systems (ITS) have gradually become a hot research area. In ITS, rapid identification and emergency response of traffic accidents are particularly important.

Traditional methods for identifying traffic accidents mainly include reporting by the parties involved, manual inspections, traffic video surveillance inspections, and identification methods based on traditional machine learning. The way for the parties to report traffic accidents mainly relies on the accident vehicle driver or witnesses to actively report the accident. Accurate accident location information may not be provided, and serious traffic accidents cannot be reported and handled in a timely manner. Manual inspections are based on regularly scheduled routine checks. It is difficult to take photographs and collect evidence at specific locations when traffic accidents occur on special sections of road, and the efficiency is low. Traffic accidents on remote and empty sections of road are difficult to detect in a timely manner. The traffic video surveillance inspection method mainly involves installing a large number of surveillance cameras in traffic-intensive areas such as cities and highways. These cameras provide remote video surveillance by monitoring road conditions in real time, assisting traffic police patrols, but continuous processing and analysis of traffic videos may require a lot of time and human resources. Such a method has certain human errors and low efficiency. Detection methods based on traditional machine learning have achieved significant improvements in detection speed and accuracy. Xiao et al. [1] proposed an ensemble learning method to address the problem of traffic accident robustness. They trained individual SVM [2] and KNN [3] models and then combined them. This ensemble learning strategy improved the robustness of individual models. Kumeda et al. [4] studied the classification of road traffic accident data and discussed six algorithms with high accuracy and optimal classification performance. Experiments showed that the fuzzy-FARCHD algorithm can effectively classify the data set with an accuracy of 85.94%. However, it requires manual feature extraction, and the extraction of a large number of features will slow down the detection process.

With the development of deep learning, artificial intelligence technology has rapidly emerged in recent years, and detection algorithms based on deep learning have been continuously applied to the field of traffic accident detection. Chakraborty et al. [5] used a YOLOv3 [6] classifier to identify traffic event trajectories from cameras when studying early detection events to reduce traffic-related congestion. The enhanced YOLOv5l [7] detection model proposed by Xia et al. [8] removes the EfficientNet [9] structure of the SE [10] layer, greatly reducing the number of model

parameters and the amount of computation. In order to improve the detection accuracy of the model, the fusion attention technique is used at the prediction end. It can be placed on the vehicle side for real-time detection of road traffic incidents. In order to more efficiently detect and handle traffic accidents, Gour et al. [11] proposed the optimized-YOLO algorithm, which focuses on optimizing the YOLO algorithm so that it can detect accidents in real time and can also run on CPU-based devices and aims to create smaller and faster detection models. Pillai et al. [12] proposed a deep learning model Mini-YOLO trained using knowledge distillation to develop a reliable and computationally inexpensive real-time automatic accident detection system that can be deployed with minimal hardware requirements. Lee et al. [13] used the YOLO algorithm to detect abnormalities on the road and only processed frontal collisions. Ghahremannezhad et al. [14] proposed a new framework for intersection accident detection for traffic monitoring. Based on the efficient and accurate target detection of YOLOv4, the experimental results on actual traffic video data demonstrate the feasibility of this method in real-time applications of traffic monitoring. In order to improve road traffic safety, Ahmed et al. [15] proposed a real-time traffic incident detection and alarm system based on the YOLOv5 algorithm. The model is able to accurately detect and classify the severity of an accident and, if a serious accident occurs, will immediately send an alert message to the nearest hospital.

Although the above methods have made great progress, there are still many problems to be solved in the field of traffic accident detection: (1) Traffic accident-type datasets are not comprehensive. They are either small scale, not from traffic surveillance cameras, not open source, or have very single scenarios. (2) In the specific perspective of the traffic pole traffic monitoring camera, large targets and small targets exist at the same time. Detecting small targets has always been a challenge for the YOLO series of target detection algorithms. (3) Existing lightweight detection models have greatly reduced the number of parameters in the model itself, but on edge devices with limited computing power, they are still difficult to deploy and cannot detect targets due to the large model and computational complexity. (4) How to make the model itself pay more attention to features and improve feature extraction capabilities.

In order to solve the above problems, this paper proposes the TP-YOLOv8 traffic accident recognition model, among which TP-YOLOv8n is the lightest. The main contributions of this paper are as follows:

(1) An improved four-layer PAN-FPN [16, 17] structure is designed, which mainly adds a small-size target detection layer based on large-scale feature maps and attaches a small-size target detection head (SODL-SODH), which significantly enhances the ability of the algorithm to detect small-sized targets.
(2) A lightweight cross-scale feature fusion module (LCSFFM) is proposed to improve the PAN-FPN structure, which improves the small-size target detection accuracy while reducing the number of model parameters and computational complexity.
(3) The squeeze-excited aggregate spatial attention module (SEASAM) is proposed. The multi-branch design improves feature representation and integrates global

information. It enables the network to learn input data features more effectively, taking into account the dependencies between channels. Then, the spatial attention part further enhances the model's perception of key features, thereby improving the performance and robustness of the model. It is worth noting that the parameters and computational complexity of the SEASAM module itself are extremely small.

(4) We constructed a new dataset, TAT, to address the issues of existing traffic accident datasets, which are limited in categories, lack traffic pole monitoring data, are small in scale, and are mostly not publicly available.

2 Related Work

2.1 YOLOv8 network structure

On January 10, 2023, ultralytics open-sourced the next major update of YOLOv5, YOLOv8 [18], which is built on the previous successful YOLO version and has many improvements in model architecture, detection accuracy, inference speed, etc. In order to completely eliminate the dependence on predefined anchor boxes, the training and tuning process of the model is simplified, while enhancing the flexibility of the model in dealing with objects of different scales. YOLOv8 abandons the Anchor [6, 1923] mechanism that the previous YOLO model relied on and uses an Anchor-free [2426] design instead. This improvement simplifies the model training process, reduces the hyperparameters related to the anchor boxes (such as the number, scale, and aspect ratio of anchor boxes), and reduces the complexity of model tuning. Since it no longer relies on anchor boxes, the model shows better generalization capabilities when facing targets of different scales and shapes, especially in small-target detection and complex target detection. Since it is no longer necessary to regress and classify each anchor box, the computational overhead of the model is reduced and the reasoning speed of the model is greatly improved. In order to improve the smoothness and fluidity of the gradient and enhance the stability and effect of model training, SiLU [27] was introduced as the activation function to replace the traditional ReLU [28].

In order to enable the model to extract and fuse multi-scale features more efficiently. YOLOv8's Backbone uses an improved version of the CSPNet [29] structure and adjusts the depth and width. CSPNet can effectively separate and fuse features, thereby reducing the amount of computation and improving the feature extraction capability of the model. In the Neck part, an improved PAN-FPN structure is used to better fuse feature maps from different scales. In this way, the model can maintain high detection accuracy when dealing with both small and large objects. The head part has undergone major changes, replacing it with the current mainstream decoupled head structure, separating the classification and detection heads. The model architecture is shown in Fig. 1.

In addition, YOLOv8 uses multi-task loss functions, including classification loss [30, 31], bounding box regression loss [32, 33], and IoU [3436] loss. These loss functions are designed to better balance different tasks and improve the overall

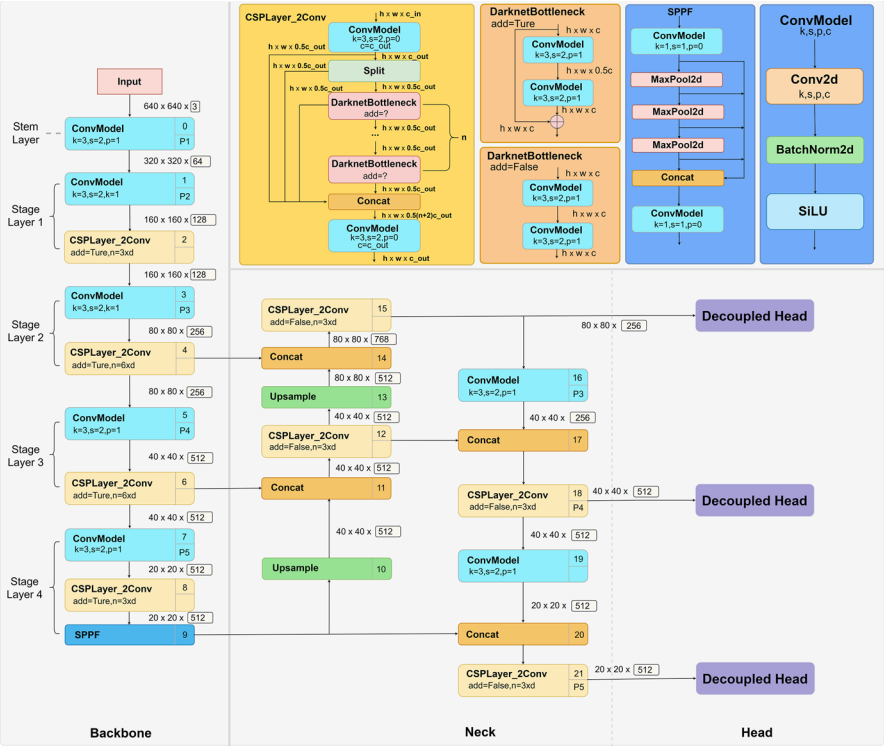

Fig. 1 YOLOv8 network structure

performance of the model. YOLOv8 has greatly improved its accuracy compared to YOLOv5, but the corresponding number of parameters and computational complexity of the N/S/M model have increased a lot, and its deployment cost on edge devices has further increased. Its generalization on custom data sets still needs more verification.

2.2 Dataset for model generalization validation

BDD100K [37] is a large-scale, diverse dataset released by the University of California, Berkeley, suitable for research in computer vision tasks such as object detection, semantic segmentation, instance segmentation, and road scene understanding. It includes 70,000 training samples, 10,000 validation samples, and 20,000 test samples. The weather conditions cover sunny, overcast, rainy days, as well as various times of day and night. The dataset features ten main label categories, including car, truck, bus, traffic light, traffic sign, pedestrian, bicyclist, motorcycle, motorcyclist, and train. The ratio of the training set to the validation set is 7:1.The DBB-IW [38] dataset is a subset of BDD100K, comprising one-fifth of the original dataset, with 14,619 training samples and 2,007 validation samples, also maintaining a ratio of

approximately 7:1. Due to the limited instances of trains and motorcycles, these categories are excluded. Bicyclists and motorcyclists are grouped into the same category as pedestrians. Thus, the optimized BDD-IW dataset includes seven categories: person, car, bus, truck, bicycle, traffic signal, and traffic sign. This dataset was created to study the differences between models under adverse weather conditions, incorporating fog processing on the basis of sunny, rainy, and snowy weather. This dataset is particularly suitable for studying the generalization of the proposed model. The dataset can be publicly accessed at https://github.com/ZhaoHe1023/Improved-YOLOv4.

The COCO2017 [39] dataset was released in 2017. It is an important standard dataset in the field of computer vision and is widely used in tasks such as image recognition, object detection, segmentation, and image description. This dataset was launched by Microsoft and contains a large number of high-quality images and their annotations. It aims to promote the research of machine learning and artificial intelligence in image understanding. Specifically, the COCO2017 dataset contains about 330,000 images, covering 80 categories, and provides the location and category labels of objects in each image for the object detection task. All annotations are done manually and the annotation quality is high. The COCO dataset provides rich task support and is an important benchmark dataset for evaluating image understanding models.

The LISA [40] dataset is a visual dataset specifically used for autonomous driving and vehicle safety research, especially for traffic sign recognition and traffic scene analysis. The dataset was released by the LISA Lab at the University of California, San Diego, and aims to promote research on tasks such as traffic sign detection, classification, and position estimation. The main feature of the LISA dataset is that it focuses on traffic signs in urban road scenes, providing high-quality annotations for the research of autonomous driving systems, and is particularly suitable for the development of traffic sign recognition systems. The LISA dataset contains more than 2,000 images, covering 47 traffic sign categories, including but not limited to stop signs, speed limit signs, and turn warning signs. It has become one of the important benchmark datasets in the field of traffic sign recognition. As one of the important benchmark datasets in the field of traffic sign recognition, the LISA dataset provides rich data support for research in fields such as autonomous driving, computer vision, and deep learning.

2.3 Lightweight design of network structure

Zhao et al. [41] designed an efficient hybrid encoder that increases intra-scale interaction and inter-scale fusion. The introduction of multi-scale features not only accelerates training convergence, but also significantly improves performance. However, although the deformable attention mechanism reduces the computational cost, the rapidly growing sequence length still makes the encoder a computational bottleneck. As pointed out by Lin et al. [42], the encoder computation accounts for 49% of the total GFLOPs but only 11% of the AP in Deformable-DETR. To address this problem, the researchers first analyzed the computational redundancy in the multi-scale

transformer encoder. Intuitively, high-level features contain rich semantic information that is extracted from low-level features, so performing feature interactions on cascaded multi-scale features appears redundant.

Therefore, the researchers designed a set of variants of different types of encoders to verify the parallelism of the encoder. The researchers rethought the structure of the encoder and proposed an efficient hybrid encoder. The encoder consists of two core modules: an attention-based intra-scale feature interaction module (AIFI) and a neural network-based cross-scale feature fusion module (CCFF). Among them, the CCFF module is optimized based on cross-scale fusion, and several fusion blocks composed of convolutional layers are inserted into the fusion path. The role of these fusion blocks is to fuse two adjacent scale features into a new feature, as shown in Fig. 2. The structure of the fusion block includes two 1x1 convolutions for adjusting the number of channels, and N RepBlocks composed of RepConv [43] to achieve feature fusion. Finally, the outputs of the two paths are fused by element-by-element addition.

2.4 Attention mechanism

It is well known that attention plays a crucial role in regulating human perception [4446]. A striking property of the human visual system is that it does not attempt to process an entire scene at once. In contrast, the human visual system selectively focuses on salient areas through a series of rapid, localized fixations [47].

Narayanan [48] introduces a novel aggregated multilayer perceptron, a multi-branch dense layer designed with squeeze-excited residual modules to surpass the performance of existing architectures. This module combines two key technologies: squeezing and excitation. The characteristic of the squeezing module is that it uses the fully connected (FC) layer to compress the input features. Specifically, the output of the convolutional layer passes through the global average pooling layer to generate the input in the channel dimension, which is then passed to the FC layer and reduced in size. In contrast, the excitation component restores the input to its original size through the FC layer and does not involve size reduction. After completing the FC layer operation, the excitation part combines the output with the feature map through channel-by-channel multiplication. The final output is rescaled to

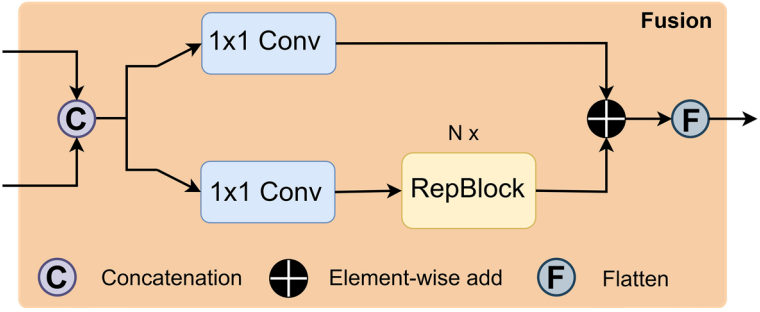

Fig. 2 Fusion block in neural network-based cross-scale feature fusion Model

align with the initial shape. The scaled output is concatenated with the input in the residual module to further enhance the feature expression.

Woo et. al., [49] mentioned that in the attention mechanism, spatial attention plays a key role in determining "where attention should be focused," as shown in [50]. Different from channel attention, spatial attention focuses on the information of "where" and is an effective supplement to channel attention. To compute the spatial attention, they first apply average pooling and max pooling operations along the channel axis and combine them to generate an effective feature descriptor. Studies have shown that pooling operations along the channel axis are more effective in highlighting key information areas [51]. In order to better determine the spatially important regions, a larger receptive field is required.

In addition, the researchers found that for a given input image, the two modules, channel attention and spatial attention, respectively, calculate complementary attention, the former focusing on "what" needs attention and the latter focusing on "where" needs attention. At this point, the two modules can be combined in a parallel or sequential manner. The researchers' experiments show that the sequential arrangement performs better than the parallel arrangement. In the sequential processing arrangement, the experimental results further show that the channel-priority arrangement has a slight advantage over the space-priority arrangement.

3 Method

3.1 Traffic accident-type dataset

In the field of traffic accident detection, commonly used datasets include the CADP dataset [52] and the TAD dataset [53]. The CADP dataset is a dataset constructed by collecting traffic accident videos on YouTube, namely the Automobile Accident Detection and Prediction Dataset, which is used for multiple purposes: time segmentation, object detection, tracking, vehicle collision, accident detection and prediction. As an image dataset for traffic accident analysis, it is mainly used for accident detection and prediction-type analysis, solving the problem of lack of public data for studying road traffic safety. It contains 1,416 traffic accident clips from fixed traffic camera views, and 205 segments with HD quality are selected to annotate spatiotemporal data for object detection, tracking, and collision detection. Its annotated categories are "person," "car" (including minivans), "bus," "two-wheeler" (including cyclists, motorcycles), "three-wheeler," and "other" (objects that do not belong to other categories). The majority of the dataset is occupied by small objects. The TAD dataset constructs a large-scale traffic accident dataset from the monitoring perspective of various scenarios. There are 333 videos in total, and 24,810 labeled images of four main accident types are randomly extracted from the 333 videos. It includes collisions between multiple vehicles, collisions between vehicles and bicycles/motorcycles, collisions between vehicles and inanimate entities, and rollovers.

The CADP dataset has been used by many researchers for traffic accident prediction; however, the annotation of CADP is performed at the time-space (time segmentation and dense spatiotemporal annotation) level rather than on the accident

type. Although the TAD dataset has annotated the main accident types, only a small portion of the unlabeled data mentioned in the paper is currently available. Other existing traffic accident datasets are either small scale, not from surveillance cameras, not open source, or have very single scenes. Therefore, we select suitable traffic accident images from CADP and TAD, and integrate them together to give full play to the advantages of the two datasets and form richer traffic scenes. Secondly, we collected traffic accident images of multiple regions, environments, and types through websites such as roboflow and CSDN and online traffic accident videos. The images of traffic accidents mainly occur in road scenes, including urban roads, highways, rural roads, etc. Accidents in different scenes are also accompanied by different environmental factors such as lighting and weather. Further enrich vehicle accident instances, supplement with fewer instances of two-wheeled vehicles and personnel in traffic accidents, and increase vehicle fire instances to make the data more diverse and comprehensive. The data images include first-person perspective, fixed traffic surveillance camera perspective and vertical perspective to meet the multi-angle requirements of traffic pole monitoring. Finally, we strictly screen the collected traffic accident images and eliminate duplicate images and images without traffic accidents. To ensure the overall image quality in the dataset, we removed images that were too blurry, overexposed, and had extremely low resolution to meet the requirement that the dataset can be used as a benchmark for training and evaluating traffic accident object detection algorithms.

The dataset has five main label categories, including multi-vehicle collision accidents (car-crash), four-wheel vehicle accidents (car-acc), two-wheel vehicle accidents (bic-acc), vehicle fire accidents (fire), and personnel accidents (per-acc), as shown in Fig. 3. The calibration standards for these five categories are as follows: For the multi-vehicle collision accident category, two or more vehicles

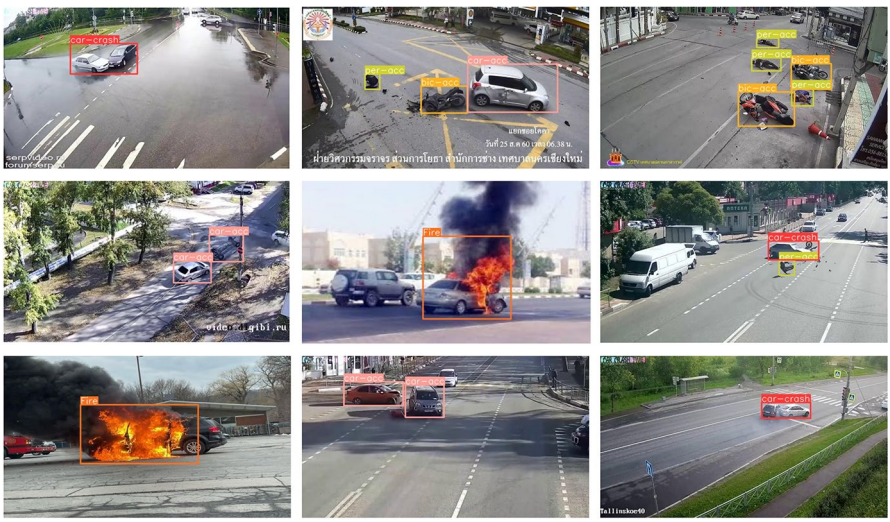

Fig. 3 Examples of annotated categories in the TAT dataset

collide and do not separate after the collision, which is marked as a multi-vehicle collision accident. The vehicles separate after the collision. Four-wheeled vehicles are marked as four-wheeled vehicle accidents, and two-wheeled vehicles are marked as two-wheeled vehicle accidents. Four-wheeled vehicle accidents mainly refer to large trucks, buses, and family cars that show abnormal conditions such as rollover, deformation, and collision with obstacles; two-wheeled vehicle accidents refer to abnormal situations such as rollover, deformation, and collision with obstacles of motorcycles, electric vehicles, and bicycles; Vehicle fire accidents refer to situations where open flames or thick smoke occur in all types of vehicles; personnel accident category: The personnel involved in the traffic accident. This dataset is named traffic accident-type dataset (TAT).

The dataset consists of 8392 images, and the training set and test set are divided in a ratio of 8:2. There are 6713 images in the training set and 1679 images in the test set, with a total of 14551 accident labels of various types. As shown in Fig. 4, the number of labels for each category in the training set is as follows: the number of labels for multi-vehicle collision accidents is 2694, the number of labels for four-wheeled vehicle accidents is 7203, the number of labels for two-wheeled vehicle accidents is 741, the number of labels for vehicle fire accidents is 456, and the number of labels for personal traffic accidents is 584; the number of labels for each category in the test set is as follows: the number of labels for multi-vehicle collision accidents is 687, the number of labels for four-wheeled vehicle accidents is 1745, the number of labels for two-wheeled vehicle accidents is 188, the number of labels for vehicle fire accidents is 112, and the number of labels for personal traffic accidents is 141. The overall ratio of the

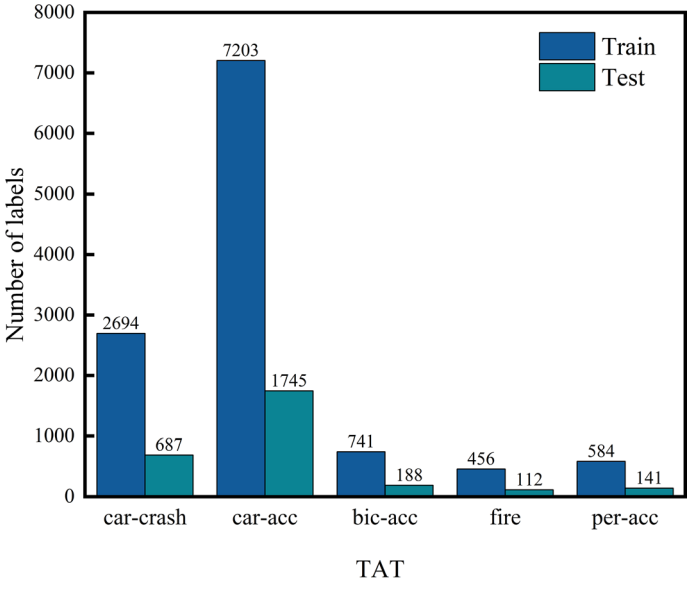

Fig. 4 Comparison of label quantities between training and testing sets

number of labels in the training set and the test set is also 8:2. The subsequent datasets are publicly available at https://github.com/Ningdashuai/TAT Dataset.

3.2 TP-YOLOv8 model

In order to overcome the difficulty of traffic accident identification based on the traffic pole monitoring perspective, further enhance the algorithm's ability to detect small-sized targets, and reduce the overall number of model parameters, this paper makes the following improvements to the YOLOv8 network. In the neck part, we expanded the improved path aggregation network-feature pyramid network (PAN-FPN) structure to four layers, mainly adding a detection layer that uses large-scale feature maps to tailor-made for small-sized targets and added a small-sized target detection head (SODL-SODH) significantly enhances the algorithm's ability to detect small-sized targets. The addition of the P2 detection layer greatly increases the number of parameters and computational complexity of our model. Therefore, before adding the P2 detection layer, we designed a lightweight cross-scale feature fusion module (LCSFFM) to make lightweight improvements to the entire improved PAN-FPN structure. The features output by backbone are adjusted to a fixed value of 256 through a 1x1 convolution module, and then output to the improved four-layer PAN-FPN structure. Under the premise of ensuring that the target detection accuracy does not decrease, the number of channels in all modules is adjusted to a fixed value of 256, which significantly reduces the amount of model parameters and computational complexity while maintaining the richness of features as much as possible. Finally, we integrate SEASAM into the CSPLayer_2Conv module in the downsampling layer to further enhance the model's perception of key features and thus improve model performance and robustness. The network model architecture of TP-YOLOv8 is shown in Fig. 5.

This paper further improves the depth and width of the TP-YOLOv8 network model and obtains the TP-YOLOv8 network series, namely TP-YOLOv8n, TP-YOLOv8s and TP-YOLOv8l. Among them, the TP-YOLOv8n model has the fewest parameters but achieves relatively high accuracy, making it suitable for edge device deployment. This network model series achieved good detection results on the BDD-IW and TAT datasets.

3.2.1 Small-scale object detection layer and small-scale object detection head (SODL-SODH)

In the object detection task, the detection of small-sized objects has always been a difficult problem. YOLOv8 has much better detection accuracy than other YOLO algorithms, but considering that the objects in the images we need to identify are relatively small, if we follow the three-layer PAN-FPN structure of YOLOv8, it will limit the detection of tiny objects in large-size images. Typically, object detectors need to process objects of different scales, and large-scale feature maps (i.e., high-resolution feature maps) can provide more detailed information for small-size objects. Therefore, using large-scale feature maps to specifically detect

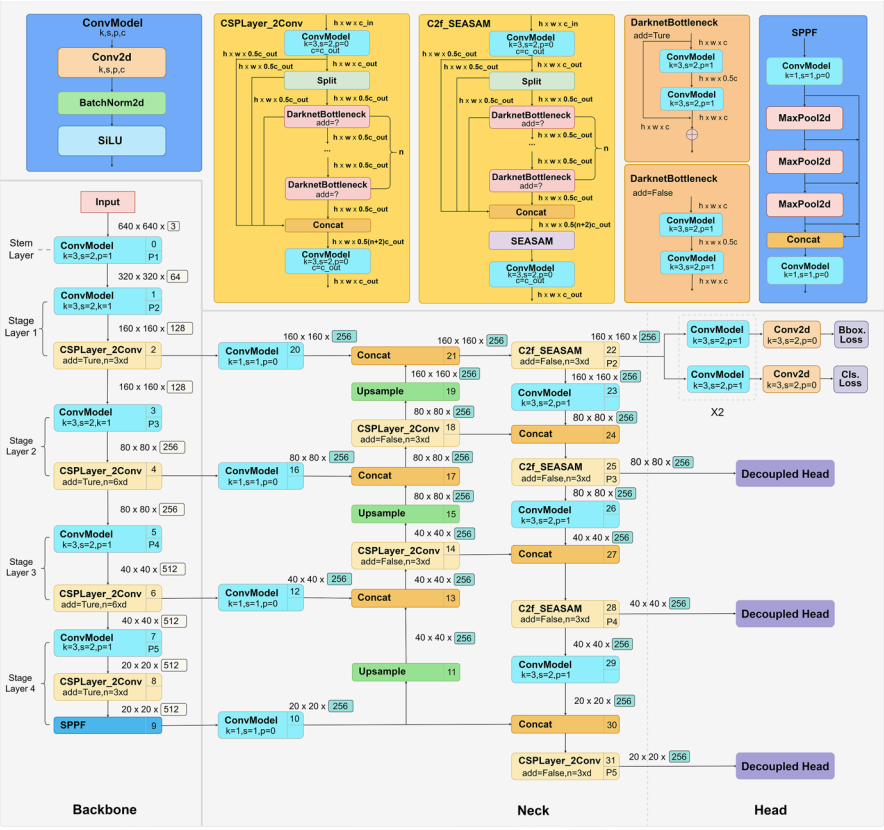

Fig. 5 TP-YOLOv8 network structure

small-sized objects is an effective strategy. We add the P2 layer to expand the PAN-FPN structure to a four-layer structure, and use feature maps of different resolutions such as P2, P3, P4, and P5 to detect small-, medium-, and large-sized targets as shown in Fig. 6.

To better process the high-resolution feature maps of the P2 layer, our small object detection head extracts features of small-sized objects through further convolution operations. Use a decoupled head consisting of a classification branch and a regression branch. In the classification problem of the target detection task, there is a difference between the probability distribution predicted by the model and the probability distribution of the true label. The classification branch uses BCE loss to effectively deal with the problem of category imbalance, and optimizes model parameters through gradient descent method to improve classification accuracy. This is achieved by calculating formula (1), where N is the total number of samples. y_i is the true label of the i sample, usually 0 or 1. p_i is the probability that the model predicts that the i sample is a positive class (label 1).

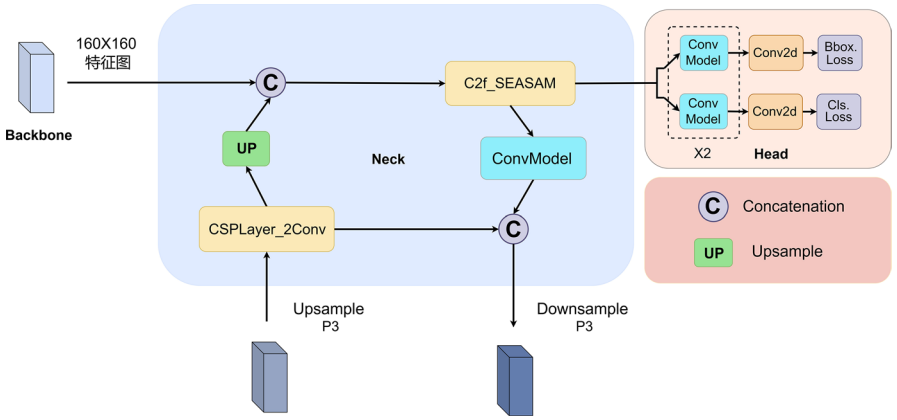

Fig. 6 SODL-SODH structure

$$\text{BCEloss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right] \tag{1}$$

In traffic accidents, our detection objects often have unclear boundaries or occlusions, so we use distribution focal loss in the regression branch. It fully considers the distribution characteristics of the bounding box coordinates instead of treating them as a single fixed value, which can improve the model's prediction accuracy for the bounding box location. This is calculated using formula (2), where S_i and S_{i+1} are adjacent predicted probability values, y_i and y_{i+1} are the bounding box coordinate values associated with S_i and S_{i+1}, and y is the true target value.

$$\text{DFL}(S_i, S_{i+1}) = -\left((y_{i+1} - y)\log(S_i) + (y - y_i)\log(S_{i+1})\right) \tag{2}$$

Since the objects detected in this paper have large changes in shape and size when a traffic accident occurs, CIoU loss is introduced, which takes into account the alignment and scale of the bounding box and helps to predict the bounding box more accurately. It is calculated by formula (3), where IoU represents the ratio of the intersection and union between the predicted box and the true box. d represents the Euclidean distance between the center point of the predicted box and the center point of the true box. c represents the diagonal distance of the minimum enclosing rectangle. v is a correction factor to account for aspect ratio consistency. The calculation method is (4) where w and h are the width and height of the predicted box, respectively, and w_{gt} and h_{gt} are the width and height of the real box, respectively. α is a weight factor used to balance the influence of v, calculated as (5).

$$\text{CIoU} = 1 - \text{IoU} + \frac{d^2}{c^2} + \alpha v \tag{3}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w_{gt}}{h_{gt}} - \arctan\frac{w}{h}\right)^2 \tag{4}$$

$$\alpha = \frac{v}{(1 - \mathrm{IoU}) + v} \tag{5}$$

The introduction of the Anchor-Free model no longer uses predefined anchor boxes to predict targets, making the model more concise. Since it is no longer necessary to calculate the regression of multiple anchor boxes for each grid point, the amount of calculation is reduced. There is no need to manually adjust the number and size of anchor boxes, which makes the model more robust on different datasets.

3.2.2 Lightweight cross-scale feature fusion module (LCSFFM)

In order to better detect small-sized objects, the P2 detection layer is added, but this leads to an increase in model parameters and an increase in computational burden. Therefore, this paper starts from the source and proposes an innovative method to improve the entire PAN-FPN structure by adjusting the number of input feature channels and fixing the number of output channels. As shown in Table 1, experiments have demonstrated that when the number of channels is set to 256, the number of model parameters and complexity are significantly reduced, while the richness of features is maintained as much as possible. As shown in Fig. 7, we output the features of different sizes extracted by the backbone network from Stage Layer 1, Stage Layer 2, Stage Layer 3 and SPPF to the neck part from top to bottom. First, the number of channels of the input features is adjusted to 256 through a 1x1 convolution module that integrates Conv2d, BatchNorm2d and SiLU. The 1x1 convolution performs point-by-point convolution on each feature point to compress those redundant or unimportant features without affecting the spatial dimension and only adjusting the number of channels. Compressing the number of channels to 256 can significantly reduce the number of parameters that need to be calculated in subsequent layers and reduce the computational complexity of the model. Then, adding BatchNorm2d helps to normalize the feature distribution of the convolution output. The smoothness and nonlinearity of the SiLU activation function can better capture complex feature patterns and improve the expressiveness of the model. The processed multi-scale feature maps are output to the P2, P3, P4, and P5 layers for upsampling and downsampling for cross-scale feature fusion. Adjusting the number of channels of all feature maps to 256 can make these feature maps no longer

Table 1 Comparison experiment of fixed channel numbers for LCSFFM

Number of channels	mAP0.5(%)	Params/10^6	GFLOPs
128	80	1.68	9.6
256	80.6	2.23	17.3
512	81.8	4.24	45.3

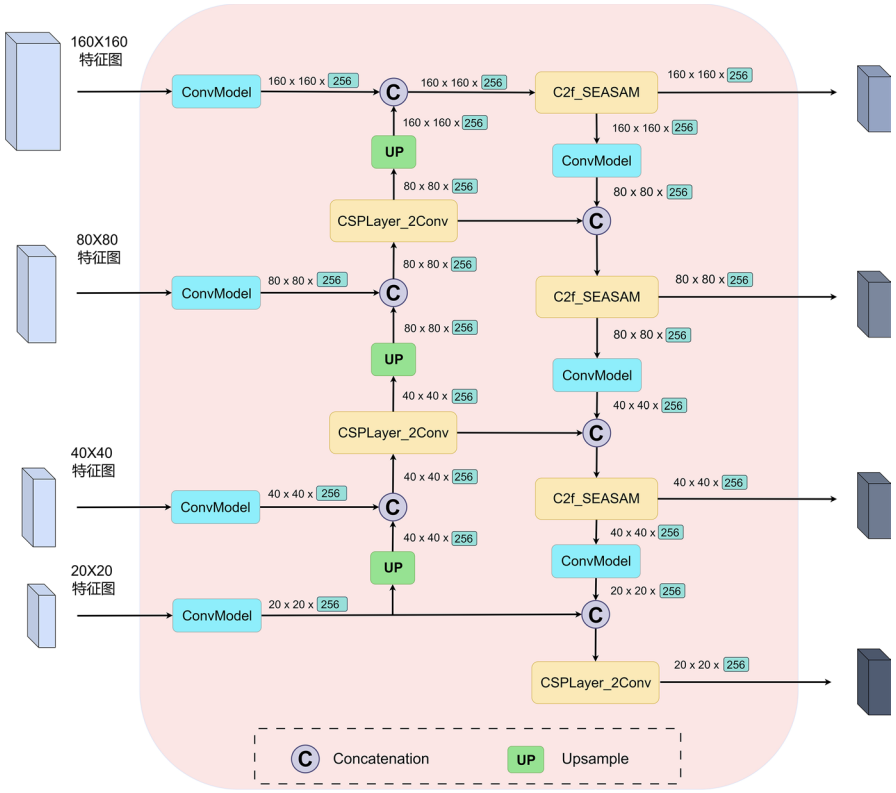

Fig. 7 LCSFFM structure

require additional channel alignment operations when they are fused, thus simplifying the feature fusion process. With a unified number of channels, the network can more easily integrate feature information from different scales, thereby improving the ability to detect objects of different sizes.

3.2.3 Squeeze-excited aggregated spatial attention module (SEASAM)

In the feature fusion process, since the low-level features need to be fused with the high-level features, the low-level high-resolution features are downsampled or convolved, which reduces the spatial resolution of the low-level features, resulting in blurred position information and loss of detail information. At the same time, the characteristics of small targets are easily masked by high-level semantic information. This paper designs a squeeze-excited aggregate spatial attention module (SEASAM) that combines channel attention and spatial attention when extracting image features. Weighting features from the channel dimension and spatial dimension allows the model to better focus on important features,

significantly enhancing the model's expressive ability in the feature fusion stage, thereby improving detection accuracy and robustness.

This part of the model improvement is analyzed, as shown in Fig. 8, (a) the visualized heat map of the baseline model, and (b) the visualized heat map of the improved model. The baseline model can identify the overturned accident vehicle, but pays less attention to the accident vehicle on the left and fails to capture the detailed accident features of the vehicle. The baseline model of the overturned vehicle on the right also only focuses on a part of the target area, reflecting that the model has limited feature extraction. In contrast, the improved model successfully detected the accident vehicle on the left and focused on the damaged area, with a significantly increased focus on the target area and more edge and texture features extracted. Reduced focus on the background, more precise positioning of the target area.

The specific implementation method is shown in Fig. 9. The spatial information of the feature map is compressed into the global statistical information of each channel through global average pooling, and the information of the high-dimensional feature map is condensed into the global representation of each channel to reduce the computational complexity. The channels of the feature map are then divided into four groups, each group of channels shares the same weight while preserving the correlation between the channels. The number of parameters in each channel group is reduced, thereby reducing the computational overhead. A ReLU activation function is added to each group after a 1x1 convolution, and a nonlinear activation function is introduced to improve the expressiveness of the model. The model can learn different global representations through multiple branches, and the outputs of these branches are finally concatenated together to enhance the feature representation. Then, a fully connected layer is used to integrate the information. This approach not only enhances the representation capabilities of the model, but also does not significantly increase the number of parameters of the model since the outputs of the branches are integrated through a lightweight compression layer after aggregation. Then, the sigmoid activation function is used to recalibrate the channel features, and finally the calculated weights are multiplied by the original feature map channel by channel to complete the re-weighting of the feature map.

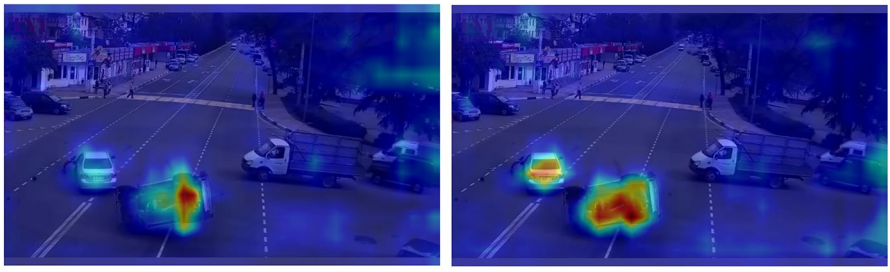

Fig. 8 Heatmap comparison. (a) Baseline model visualization. (b) lmproved model visualization

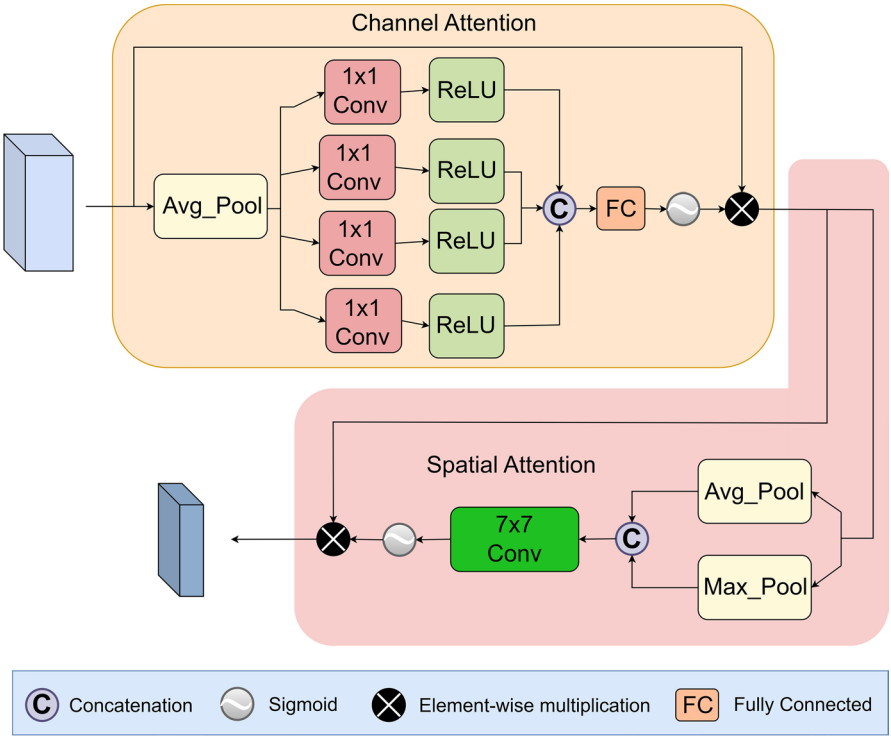

Fig. 9 SEASAM structure

Then, global maximum pooling and global average pooling are performed on the feature map in the channel dimension to obtain two single-channel feature maps, namely the global information and local salient information of the entire feature map. They are spliced together to perform a 7×7 convolution operation to generate a spatial attention map, and the generated spatial attention map is multiplied element-by-element with the input feature map to obtain a weighted feature map, which improves the model's ability to capture local areas and global structures. Combined with the previous channel attention, multi-dimensional feature selection of feature maps is achieved.

4 Experiment

4.1 Experimental environment and parameters

The model training in this article was performed using the Ubuntu 22.04.4 LTS operating system equipped with an NVIDIA A40 computing card with 48GB of GDDR6 video memory. The model in this experiment is built using the PyTorch 1.13.1 deep learning framework, with training acceleration provided by Cuda 11.6 and Python version 3.9.18.During the experiment, we set the input image size to 640×640, selected an

initial learning rate of 0.01, set the momentum to 0.937, adopted the stochastic gradient descent optimizer, set the batch size to 16, and used two threads to speed up data loading. All models do not use pre-trained weights. To ensure that the model can converge sufficiently, the number of iterations is set to 300 rounds. To prevent the model from overfitting, if the accuracy does not improve within 50 iterations, the model will stop training.

4.2 Evaluation metrics

To better evaluate the effectiveness of the model, we use four common indicators: category accuracy (AP), average accuracy (mAP), parameters (Params), and FLOPs. To objectively evaluate the detection performance of the proposed model, the interpretation of these indicators is as follows:

Precision (P) refers to how many of the samples judged as positive by the model actually belong to the positive class. It is calculated by formula (6), where *TP* represents the number of samples correctly predicted as positive, and *FP* represents the number of samples incorrectly predicted as positive.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

Recall (R) refers to the proportion of samples that are successfully predicted as positive among all samples that are actually positive. It is calculated by formula (7), where *FN* is the number of samples that are actually positive but are mistakenly predicted as negative.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

The precision (AP) is obtained by calculating the average of the precisions under different recall rates. It measures the accuracy performance of the model under a given recall rate and is calculated using formula (8):

$$\text{AP} = \int_0^1 P(R)\, dR \tag{8}$$

Mean average precision (mAP) is obtained by calculating the average precision (AP) for each class and then averaging the *AP* for all classes. Calculated by formula (9), our average precision indicator here is mAP0.5, which means that when the IoU between the bounding box predicted by the model and the true bounding box reaches 0.5, the prediction is considered correct.

$$\text{mAP} = \frac{1}{k} \sum_{i=1}^{k} AP_i \tag{9}$$

4.3 Comparison methods

In order to objectively and fully verify the actual target recognition performance and generalization of TP-YOLOv8, we selected 13 SOTA methods and compared them. Comparative experiments were conducted on the BDD-IW and TAT datasets under the same settings with mainstream two-stage and one-stage object detection algorithms. The methods compared include faster R-CNN [54], SSD [55], YOLOv3 [6], YOLOv4 [19], YOLOv5 [7], YOLOv6 [25], YOLOX [24], YOLOv7 [23], TPH-YOLOv5 [56], PPYOLOE [26], improved YOLOv4 [38], and YOLOv8 [18].

4.4 Comparative experiments

The input image size is fixed to 640x640, and the test is performed on the DBB-IW dataset to obtain the comparative experimental results shown in Table 2. AP0.5 is used as the metric for evaluation results for each object category, and mAP0.5 is used as the metric for evaluation results for all categories of objects. The TP-YOLOv8l proposed in this paper achieved the best results in mAP0.5, and the AP0.5 of seven different categories of object recognition was the best. Compared to improved YOLOv4, previously the best on this dataset, TP-YOLOv8s improved mAP0.5 by 1.4% and TP-YOLOv8l by 5.7%. TP-YOLOv8n improved mAP0.5 by 6.7% over the baseline, demonstrating the strong performance of the TP-YOLOv8 series.

On the DBB-IW dataset, TP-YOLOv8n and TP-YOLOv8s have notably small parameters compared to existing methods. Even for TP-YOLOv8l, the number of model parameters is reduced by 32.2% compared with the original best method on the dataset. For easily detected categories of objects, good recognition results are maintained, while recognition of small, hard-to-detect targets like bicycles improved significantly.

The input image size is fixed to 640x640, and the test is performed on the TAT dataset to obtain the comparative experimental results shown in Table 3. It can be found that other advanced methods have also achieved good detection results, verifying the effectiveness of the proposed traffic accident category dataset. Compared with the baseline method, our TP-YOLOv8n mAP0.5 value increased by 1%, while the number of parameters was reduced by 25.9%.

In order to strengthen the evaluation and further prove the effectiveness of the TP-YOLOv8 model improvement, we compare the performance of the model on the COCO val2017 dataset. The input image size is fixed to 640x640 and tested on this dataset. The comparative experimental results are shown in Table 4. The TP-YOLOv8 models with different parameter sizes proposed in this paper achieved the best results in mAP0.5. Compared to the baseline model, TP-YOLOv8n improved mAP0.5 by 1.9%, TP-YOLOv8s improved it by 0.69%, and TP-YOLOv8l improved it by 1.2%, fully demonstrating the effectiveness of the improvements made to the TP-YOLOv8 model.

Table 2 Experimental comparisons on the DBB-IW dataset

Method	Person (%)	Car (%)	Bus (%)	Truck (%)	Bike (%)	Traffic light (%)	Traffic sign (%)	mAP0.5 (%)	Params/10^6	GFLOPs
Faster R-CNN	-	-	-	-	-	-	-	55.6	136.77	369.8
SSD	-	-	-	-	-	-	-	34.7	24.14	175.9
YOLOv3	47.8	69.0	44.1	51.0	26.0	46.9	48.9	47.7	103.69	283.0
YOLOv4	55.2	73.4	49.4	56.4	34.0	55.4	57.8	54.5	52.5	119.8
YOLOv5	54.5	73.2	52.3	57.2	32.7	55.0	56.1	54.4	76.8	111.4
YOLOv6	-	-	-	-	-	-	-	49.5	59.6	150.7
YOLOX	58.5	76.7	54.5	59.5	31.6	62.0	60.4	57.6	54.2	155.6
TPH-YOLOv5	51.0	71.8	45.4	52.9	26.1	55.2	57.1	51.4	41.5	160.6
PPYOLOE	52.9	72.7	46.4	53.5	28.8	56.5	58.9	52.7	52.2	110.0
improved YOLOv4	62.0	78.8	57.6	60.5	33.6	64.9	64.9	60.3	51.09	91.0
YOLOv8n	50.6	73.5	47.3	53.3	21.4	47.4	50.8	49.2	3.01	**8.2**
YOLOv8s	57.9	77.3	53.6	57.7	30.4	55.0	59.5	55.9	11.14	28.7
YOLOv8l	62.9	80.2	59.5	61.5	40.3	60.5	65.4	61.5	43.64	165.4
TP-YOLOv8n(ours)	57.6	78.1	50.3	56.2	26.6	60.1	62.2	55.9	**2.23**	17.3
TP-YOLOv8s(ours)	63.0	81.4	55.8	61.1	35.3	66.6	68.5	61.7	8.10	55.1
TP-YOLOv8l(ours)	**68.5**	**83.5**	**60.2**	**64.3**	**44.0**	**68.7**	**72.5**	**66.0**	34.61	291.9

Bold values highlight the best experimental metrics

Table 3 Experimental comparisons on the TAT dataset

Method	Car-crash	Car-acc (%)	Bic-acc (%)	Fire (%)	Per-acc (%)	mAP0.5 (%)	Params/10^6	GFLOPs
Faster R-CNN	88.8	87.3	78.0	56.2	33.0	68.7	136.77	369.8
SSD	84.3	87.9	75.8	**61.3**	36.6	69.2	24.14	175.9
YOLOv3-tiny	90.8	91.1	80.9	62.8	58.2	76.8	12.13	19.0
YOLOv4-tiny	79.3	80.0	63.8	37.2	45.2	61.1	5.88	16.2
YOLOv5n	94.2	92.8	81.9	68.1	57.7	78.9	2.51	**7.2**
YOLOv6n	**95.1**	93.5	83.0	**71.3**	55. 6	79.7	4.23	11.9
YOLOX-s	92.2	91.4	**84.9**	69.2	**63.6**	80.2	9.0	26.8
YOLOv7-tiny	92.1	91.3	82.6	67.7	63.3	79.4	6.03	13.2
YOLOv8n	94.5	93.9	83.7	66.4	59.7	79.6	3.01	8.2
TP-YOLOv8n(ours)	94.2	**94**	83.9	68.4	62.5	**80.6**	**2.23**	17.3

Bold values highlight the best experimental metrics

Table 4 Experimental comparisons on the COCO val2017 dataset

Method	Precision(%)	Recall(%)	mAP0.5(%)	Params/10^6	GFLOPs
YOLOv5s	67.2	51.5	55.5	9.15	24.2
YOLOv8n	57.9	46.8	47.6	3.01	8.2
YOLOv8s	66.7	52.8	57.7	11.14	28.7
YOLOv8l	71.7	60.6	66.5	43.64	165.4
TP-YOLOv8n(ours)	58.5	47.3	48.5	2.23	17.3
TP-YOLOv8s(ours)	67.1	53.2	58.1	8.10	55. 1
TP-YOLOv8l(ours)	72.2	61.1	67.3	34.61	291.9

4.5 Ablation experiment

This paper conducts experiments on DBB-IW and TAT datasets, and validates each module through ablation experiments. To assess each module's impact on YOLOv8, we strictly controlled the variables during the experiment and kept all hyperparameters unchanged. In the ablation experiment of the DBB-IW dataset, the YOLOv8s model is selected as the baseline method for comparison. In the ablation experiment of the TAT dataset, the YOLOv8n model is selected as the baseline method for comparison. We propose a lightweight cross-scale feature fusion module (LCSFFM), add a small-scale object detection layer based on a large-scale feature map and attach a small-size object detection head (SODL-SODH), and design a squeeze-excited aggregate spatial attention module (SEASAM). After the above three stages of improvements, we proposed the TP-YOLOv8 model.

As can be seen from Table 5, after we improved LCSFFM to the baseline model, we reduced the number of model parameters by 34.8% and the computational complexity by 19.2% while maintaining mAP0.5, laying a foundation for subsequent

Table 5 Ablation experiments on DBB-IW

YOLOv8s(Base)	LCSFFM	SODL-SODH	SEASAM	mAP0.5(%)	Params/10^6	GFLOPs
✓				55.9	11.14	28.7
✓	✓			55.9	7.26	23.2
✓	✓	✓		61.3	8.08	55.1
✓	✓	✓	✓	61.7	8.10	55.1

The check mark indicates that the module has been added to the baseline model

improvements. After adding SODL-SODH, the model's detection accuracy for small targets has been greatly improved, mAP0.5 has increased by 9.7%, and the number of parameters has been reduced by 27.5%. After integrating SEASAM into the model, mAP0.5 increased by 10.4% and the number of parameters was reduced by 27.3%. Figure 10 intuitively shows the ablation experiment comparison results measured by four indicators: precision, recall, mAP0.5, and mAP0.5−0.95 after adding various improved modules to the baseline model.

As can be seen from Table 6, after we improve LCSFFM to the baseline model, the mAP0.5 increases by 0.25%, the number of model parameters decreases by 34.9%, and the computational complexity decreases by 18.3%, laying a foundation for subsequent improvements; after continuing to add SODL-SODH, the model's detection accuracy for small targets is greatly improved, with mAP0.5 increased by 0.87% and the number of parameters reduced by 26.2%; after

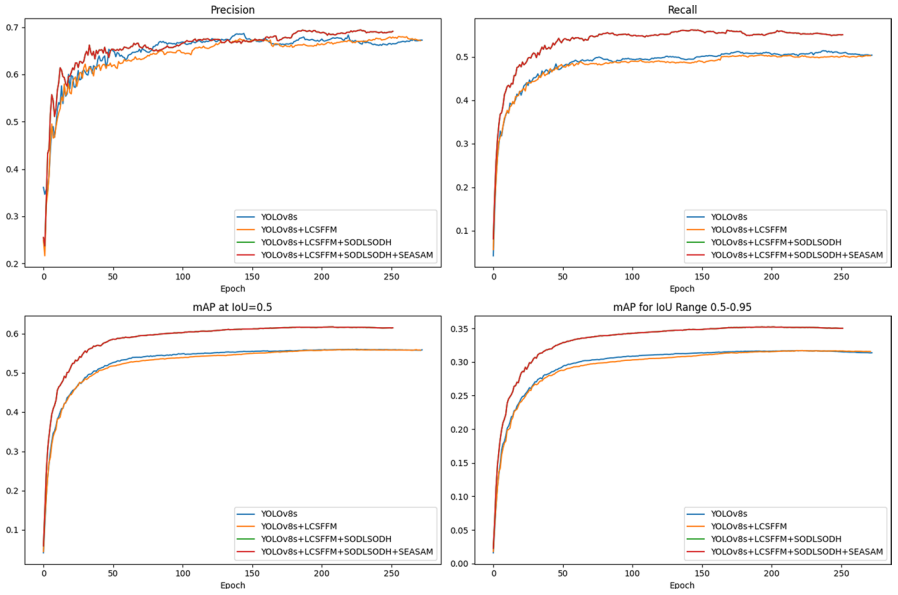

Fig. 10 Visualization of ablation experiments on the DBB-IW dataset after adding each improved module

Table 6 Ablation experiments on TAT

YOLOv8s(Base)	LCSFFM	SODL-SODH	SEASAM	mAP0.5(%)	Params/10^6	GFLOPs
✓				79.6	3.01	8.2
✓	✓			79.8	1.96	6.7
✓	✓	✓		80.3	2.22	17.3
✓	✓	✓	✓	80.6	2.23	17.3

The check mark indicates that the module has been added to the baseline model

integrating SEASAM into the model, the mAP0.5 increases by 1.26% and the number of parameters decreases by 25.9%. Figure 11 intuitively shows the ablation experiment comparison results measured by four indicators: precision, recall, mAP0.5, and mAP0.5−0.95 after adding various improved modules to the baseline model.

As shown in Fig. 12, the baseline model can identify the overturned accident vehicle, but pays less attention to the accident vehicle on the left and fails to capture the detailed accident features of the vehicle. The baseline model of the overturned vehicle on the right also only pays attention to a part of the target area. With the addition of the improved module, the model pays more attention to the target, pays less attention to the background, and locates the target area more accurately.

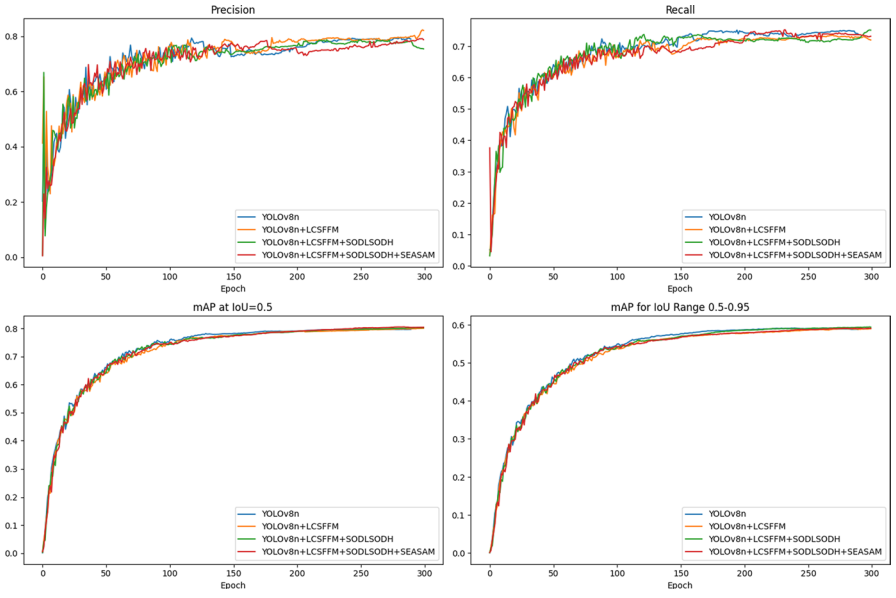

Fig. 11 Visualization of ablation experiments on the TAT dataset after adding each improved module

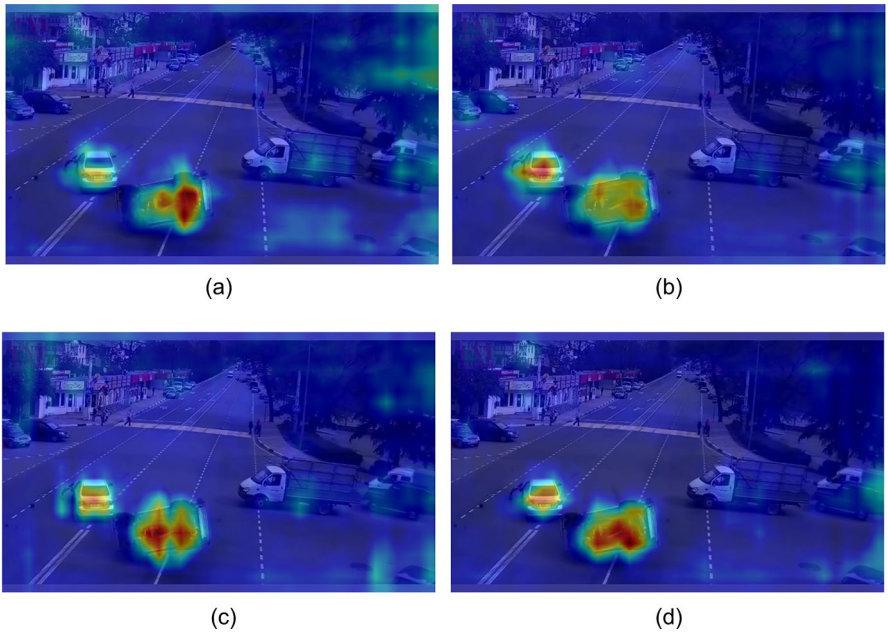

Fig. 12 Visualization heat map of model ablation experiment on TAT dataset. (a) Visualization heat map of the baseline model, (b) Visualization heat map of the baseline+LCSFFM model, (c) Visualization heat map of the baseline+LCSFFM+SODL-SODH model, (d) Visualization heat map of the baseline+LCSFFM+SODL-SODH+SEASAM model

4.6 Experimental results visualization

4.6.1 Display of traffic accident recognition effectiveness

To provide a more intuitive demonstration of the effectiveness of the proposed method in traffic accident recognition, we present the detection results on the TAT dataset. As shown in Fig. 13, our method shows excellent performance in traffic accident recognition from the perspective of a traffic pole surveillance camera.

4.6.2 Demonstration of model generalization effectiveness

As shown in Fig. 14, in order to more intuitively prove that the method proposed in this paper has good generalization performance, we show the detection results on the DBB-IW dataset. Our method shows excellent performance under adverse weather conditions. The improved algorithm can successfully detect various types of targets under various lighting conditions and maintains good detection capabilities even in dimly lit scenes.

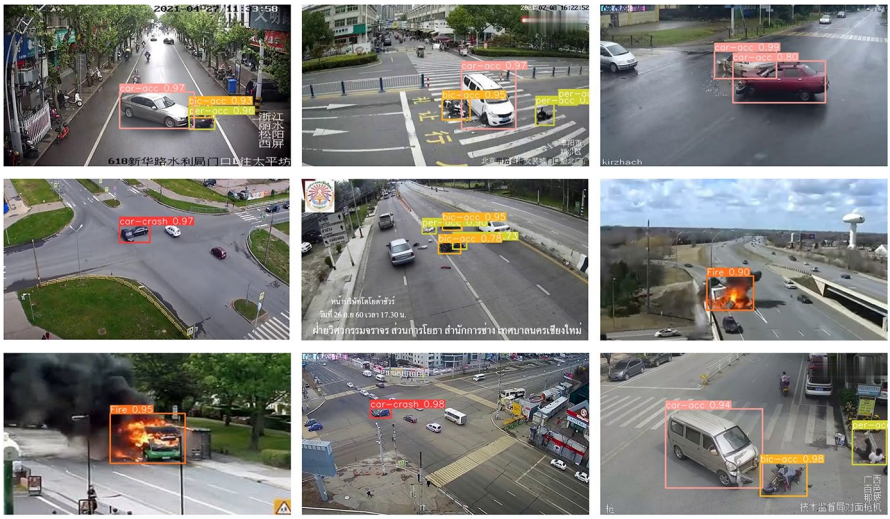

Fig. 13 Test results of TP-YOLOv8 on the TAT dataset

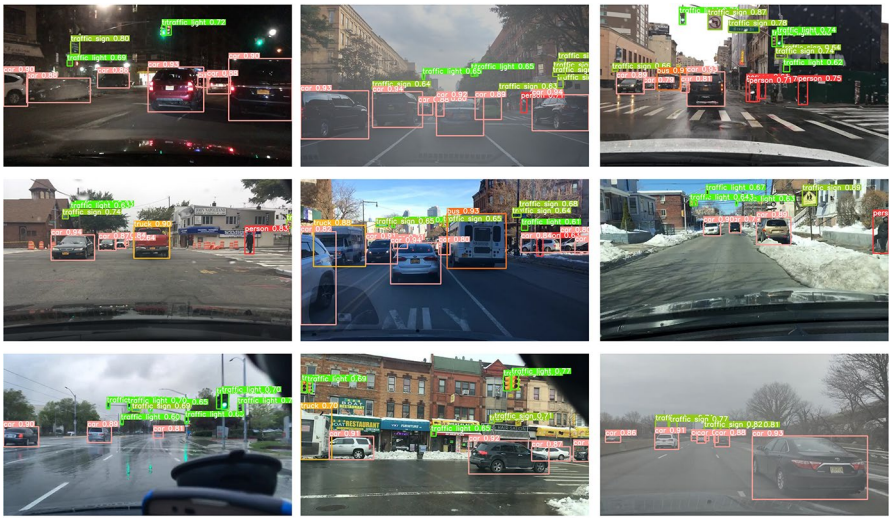

Fig. 14 Detection results of TP-YOLOv8 on the DBB-IW dataset

4.6.3 Edge device performance verification

To demonstrate the practicality of the improved model, we tested the TP-YOLOv8n model on the target detection task using the TAT dataset on the NVIDIA Jetson TX2 NX embedded platform, as shown in Figs. 15 and 16. The results show that the model has achieved the established detection goals. It is worth noting that the

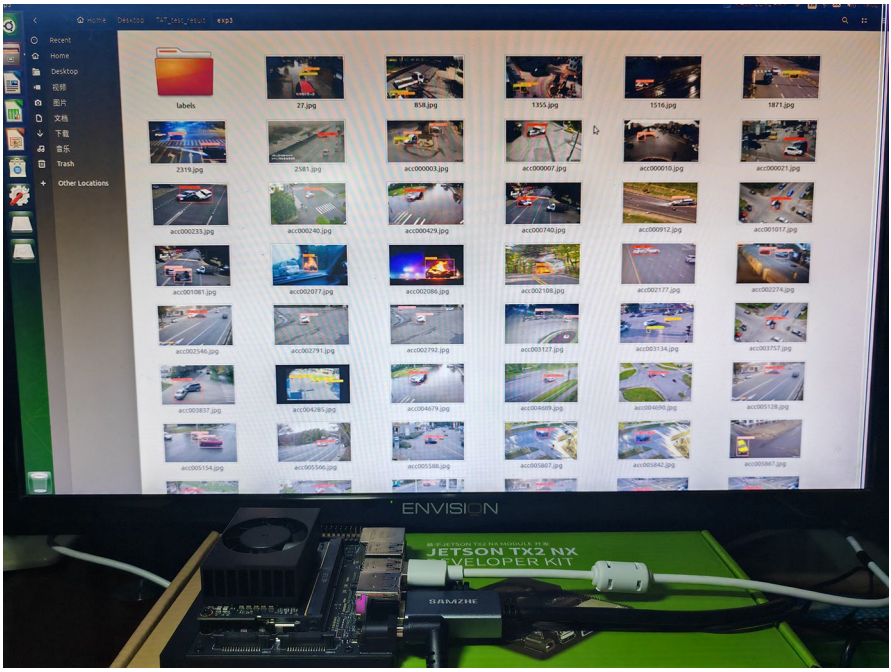

Fig. 15 NVIDIA Jetson TX2 NX Platform

Fig. 16 Test results of TP-YOLOv8n model using TAT dataset on NVIDIA Jetson TX2 NX platform

method we proposed includes a model system with different parameter scales, of which the lightweight version has been verified on the NVIDIA Jetson TX2 NX platform. The results show that even models with smaller parameter scales can still

Table 7 NVIDIA Jetson TX2 NX hardware specifications

Device Name	Power Consumption	CPU	GPU	TFLOPs
NVIDIA Jetson TX2 NX	15W	Dual-core NVIDIA Denver 2™ 64-bit CPU with Quad-core Arm® Cortex®-A57 MPcore	NVIDIA Pascal™ Architecture, Equipped with 256 NVIDIA® CUDA® Cores	1.33

maintain high-precision detection performance. Combined with the actual computing power analysis of the platform, as shown in Table 7, our method has achieved a good balance between accuracy and computing power requirements, fully proving that it is suitable for target detection tasks on the NVIDIA series of high-performance embedded platforms.

5 Conclusions

This paper proposes an improved TP-YOLOv8 model to address the challenges of accident detection in traffic monitoring scenarios, such as specific viewpoints and varying target scales. To solve issues like incomplete dataset categories and small dataset size, we constructed a dedicated dataset, TAT, covering typical accident types. Several innovative improvements were made to the YOLOv8 model: first, a lightweight cross-scale feature fusion module was designed to significantly enhance multi-scale detection capabilities while reducing model parameters; second, a small-target detection layer and corresponding detection head were added on top of the large-scale feature map to improve the ability to capture small accident targets; finally, a squeeze-incentive aggregation spatial attention module was proposed to strengthen the expression of key features during the feature fusion stage. Experimental results show that among TP-YOLOv8 models with different parameter sizes, the larger parameter version is suitable for desktop computing platforms, while the smaller parameter lightweight version achieves high-precision detection on the NVIDIA Jetson TX2 NX embedded platform. Although the model parameter size has been optimized, the computational complexity of the lightweight version still needs further reduction. In the future, we will focus on optimizing computational efficiency to adapt to embedded devices with lower performance.

Author Contributions Z.N. and T.Z. conceptualized the study; Z.N. curated and processed the data; Z.N. and T.Z. conducted formal analysis; G.S. acquired funding, provided resources, participated in the investigation, and supervised the project; Z.N. designed the methodology; A.W. and X.L. managed the project's progress and coordination; Z.N. and X.L. validated the results and created visualizations; Z.N. wrote the main manuscript text; A.W. and T.Z. reviewed and edited the manuscript.

Funding This research was funded by the Key R&D projects of Xinjiang Uygur Autonomous Region, grant number 2022B01006.

Data Availability The datasets used in the study are from the website and can be downloaded through the following links: BDD-IW dataset (https://github.com/ZhaoHe1023/Improved-YOLOv4), TAT dataset (https://github.com/Ningdashuai/TAT).

Declarations

Conflict of interest No potential conflict of interest was reported by the authors.

References

1. Xiao J (2019) Svm and knn ensemble learning for traffic incident detection. PhysA Stat Mech Appl 517:29–35. https://doi.org/10.1016/j.physa.2018.10.060
2. Tang Y (2013) Deep learning using linear support vector machines, https://doi.org/10.48550/arXiv.1306.0239. arXiv preprint arXiv:1306.0239
3. Abeywickrama T, Cheema MA, Taniar D (2016) K-nearest neighbors on road networks: a journey in experimentation and in-memory implementation, https://doi.org/10.48550/arXiv.1601.01549. arXiv preprint arXiv:1601.01549
4. Kumeda B, Zhang F, Zhou F, Hussain S, Almasri A, Assefa M (2019) Classification of road traffic accident data using machine learning algorithms. In: 2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN), IEEE, Chongqing, China, pp 682–687. https://doi.org/10.1109/ICCSN.2019.8905362
5. Chakraborty P, Sharma A, Hegde C (2018) Freeway traffic incident detection from cameras: a semi-supervised learning approach. In: Zhang W, Bayen AM, Medina JJS, Barth MJ (ed) 21st International Conference on Intelligent Transportation Systems, ITSC 2018, Maui, HI, USA, November 4-7, 2018, IEEE, Maui, HI, USA, pp 1840–1845. https://doi.org/10.1109/ITSC.2018.8569426
6. Farhadi A, Redmon J (2018) Yolov3: An incremental improvement. In: Computer Vision and Pattern Recognition, vol 1804, pp. 1–6 https://doi.org/10.48550/arXiv.1804.02767 . Springer Berlin/Heidelberg, Germany
7. Jocher G (2022) Yolov5 Release v7.0. Accessed: 2022. https://github.com/ultralytics/yolov5/tree/v7.0
8. Xia Z, Gong J, Long Y, Ren W, Wang J, Lan H (2022) Research on traffic accident detection based on vehicle perspective. In: 2022 4th International Conference on Robotics and Computer Vision (ICRCV), IEEE, Wuhan, China, pp 223–227. https://doi.org/10.1109/ICRCV55858.2022.9953179
9. Tan M (2019) Efficientnet: Rethinking model scaling for convolutional neural networks, 6105–6114. arXiv preprint arXiv:1905.11946
10. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 7132–7141
11. Gour D, Kanskar A (2019) Optimised yolo: algorithm for cpu to detect road traffic accident and alert system. Int J Eng Res Technol 8:160–163
12. Pillai MS, Chaudhary G, Khari M, Crespo RG (2021) Real-time image enhancement for an automatic automobile accident detection through cctv using deep learning. Soft Comput 25(18):11929–11940. https://doi.org/10.1007/S00500-021-05576-W
13. Lee C, Kim H, Oh S, Doo I (2021) A study on building a "real-time vehicle accident and road obstacle notification model'' using ai cctv. Appl Sci 11(17):8210. https://doi.org/10.3390/app11178210
14. Ghahremannezhad H, Shi H, Liu C (2022) Real-time accident detection in traffic surveillance using deep learning. In: IEEE International Conference on Imaging Systems and Techniques, IST 2022, Kaohsiung, Taiwan, June 21-23, 2022, IEEE, Kaohsiung, Taiwan, pp 1–6. https://doi.org/10.1109/IST55454.2022.9827736
15. Ahmed MIB, Zaghdoud R, Ahmed MS, Sendi R, Alsharif S, Alabdulkarim J, Saad BAA, Alsabt R, Rahman A, Krishnasamy G (2023) A real-time computer vision based approach to detection and classification of traffic incidents. Big Data Cogn. Comput. 7(1):22. https://doi.org/10.3390/BDCC7010022
16. Li H, Xiong P, An J, Wang L (2018) Pyramid attention network for semantic segmentation. arXiv preprint arXiv:1805.10180

17. Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2117–2125
18. Jocher G (2023) Yolov8. Accessed: 2023. https://github.com/ultralytics/ultralytics/tree/main
19. Bochkovskiy A, Wang C-Y, Liao H-YM (2020) Yolov4: Optimal speed and accuracy of object detection, https://doi.org/10.48550/arXiv.2004.10934. arXiv preprint arXiv:2004.10934
20. Huang X, Wang X, Lv W, Bai X, Long X, Deng K, Dang Q, Han S, Liu Q, Hu X, et al (2021) Pp-yolov2: A practical object detector, https://doi.org/10.48550/arXiv.2104.10419. arXiv preprint arXiv:2104.10419
21. Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 7263–7271. https://doi.org/10.1109/CVPR.2017.690
22. Wang C-Y, Bochkovskiy A, Liao H-YM (2021) Scaled-yolov4: scaling cross stage partial network. In: Proceedings of the IEEE/cvf Conference on Computer Vision and Pattern Recognition, pp 13029–13038. https://doi.org/10.48550/arXiv.2011.08036Focustolearnmore
23. Wang C-Y, Bochkovskiy A, Liao H-YM (2023) Yolov7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 7464–7475. https://doi.org/10.48550/arXiv.2207.02696
24. Ge Z (2021) Yolox: Exceeding yolo series in 2021, https://doi.org/10.48550/arXiv.2107.08430. arXiv preprint arXiv:2107.08430
25. Li C, Li L, Geng Y, Jiang H, Cheng M, Zhang B, Ke Z, Xu X, Chu X (2023) Yolov6 v3. 0: a full-scale reloading, https://doi.org/10.48550/arXiv.2301.05586. arXiv preprint arXiv:2301.05586
26. Xu S, Wang X, Lv W, Chang Q, Cui C, Deng K, Wang G, Dang Q, Wei S, Du Y, et al (2022) Pp-yoloe: an evolved version of yolo, https://doi.org/10.48550/arXiv.2203.16250. arXiv preprint arXiv:2203.16250
27. Elfwing S, Uchibe E, Doya K (2018) Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural Netw 107:3–11. https://doi.org/10.48550/arXiv.1702.03118
28. Agarap A (2018) Deep learning using rectified linear units (relu), https://doi.org/10.48550/arXiv.1803.08375. arXiv preprint arXiv:1803.08375
29. bibitemwang2020cspnet29 Wang C-Y, Liao H-YM, Wu Y-H, Chen P-Y, Hsieh J-W, Yeh I-H (2020) Cspnet: a new backbone that can enhance learning capability of cnn. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 390–391. https://doi.org/10.48550/arXiv.1911.11929
30. Mao A, Mohri M, Zhong Y (2023) Cross-entropy loss functions: theoretical analysis and applications. In: International Conference on Machine Learning, PMLR, pp 23803–23828. https://doi.org/10.48550/arXiv.2304.07288.
31. Li Q, Jia X, Zhou J, Shen L, Duan J (2024) Rediscovering bce loss for uniform classification, https://doi.org/10.48550/arXiv.2403.07289. arXiv preprint arXiv:2403.07289
32. Ren J, Zhang M, Yu C, Liu Z (2022) Balanced mse for imbalanced visual regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 7926–7935. https://doi.org/10.48550/arXiv.2203.16427
33. Li X, Wang W, Wu L, Chen S, Hu X, Li J, Tang J, Yang J (2020) Generalized focal loss: learning qualified and distributed bounding boxes for dense object detection. Adv Neural Inform Process Syst 33:21002–21012. https://doi.org/10.48550/arXiv.2006.04388
34. Yu J, Jiang Y, Wang Z, Cao Z, Huang T (2016) Unitbox: An advanced object detection network. In: Proceedings of the 24th ACM International Conference on Multimedia, pp 516–520. https://doi.org/10.1145/2964284.2967274
35. Rezatofighi H, Tsoi N, Gwak J, Sadeghian A, Reid I, Savarese S (2019) Generalized intersection over union: a metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 658–666. https://doi.org/10.1109/CVPR.2019.00075
36. Zheng Z, Wang P, Liu W, Li J, Ye R, Ren D (2020) Distance-iou loss: faster and better learning for bounding box regression. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 34, pp 12993–13000. https://doi.org/10.48550/arXiv.1911.08287
37. Yu F, Chen H, Wang X, Xian W, Chen Y, Liu F, Madhavan V, Darrell T (2020) Bdd100k: a diverse driving dataset for heterogeneous multitask learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 2636–2645. https://doi.org/10.48550/arXiv.1805.04687
38. Wang R, Zhao H, Xu Z, Ding Y, Li G, Zhang Y, Li H (2023) Real-time vehicle target detection in inclement weather conditions based on yolov4. Front Neurorobotics 17:1058723. https://doi.org/10.3389/fnbot.2023.1058723

39. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV). https://cocodataset.org
40. UCSD LISA Lab (2010) LISA Traffic Sign Dataset. University of California, San Diego. http://cvrr.ucsd.edu
41. Zhao Y, Lv W, Xu S, Wei J, Wang G, Dang Q, Liu Y, Chen J (2024) Detrs beat yolos on real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 16965–16974. https://doi.org/10.48550/arXiv.2304.08069
42. Lin J, Mao X, Chen Y, Xu L, He Y, Xue H (2022) D 2etr: Decoder-only detr with computationally efficient cross-scale attention, https://doi.org/10.48550/arXiv.2203.00860. arXiv preprint arXiv:2203.00860
43. Soudy M, Afify Y, Badr N (2022) Repconv: a novel architecture for image scene classification on intel scenes dataset. Int J Intell Comput Inform Sci 22(2):63–73. https://doi.org/10.21608/ijicis.2022.118834.1163
44. Itti L, Koch C, Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. IEEE Trans pattern Anal Mach Intell 20(11):1254–1259. https://doi.org/10.1109/34.730558
45. Rensink RA (2000) The dynamic representation of scenes. Vis Cognit 7(1–3):17–42. https://doi.org/10.1080/135062800394667
46. Corbetta M, Shulman GL (2002) Control of goal-directed and stimulus-driven attention in the brain. Nat Rev Neurosci 3(3):201–215. https://doi.org/10.1038/nrn755
47. Larochelle H, Hinton GE (2010) Learning to combine foveal glimpses with a third-order boltzmann machine. Adv Neural Inform Process Syst, 23
48. Narayanan M (2023) Senetv2: Aggregated dense layer for channelwise and global representations, https://doi.org/10.48550/arXiv.2311.10807. arXiv preprint arXiv:2311.10807
49. Woo S, Park J, Lee J-Y, Kweon IS (2018) Cbam: convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19. https://doi.org/10.48550/arXiv.1807.06521
50. Chen L, Zhang H, Xiao J, Nie L, Shao J, Liu W, Chua T-S (2017) Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 5659–5667. https://doi.org/10.48550/arXiv.1611.05594
51. Zagoruyko S, Komodakis N (2016) Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer, https://doi.org/10.48550/arXiv.1612.03928. arXiv preprint arXiv:1612.03928
52. Shah AP, Lamare J-B, Nguyen-Anh T, Hauptmann A (2018) Cadp: a novel dataset for cctv traffic camera based accident analysis. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp 1–9. https://doi.org/10.48550/arXiv.1809.05782. IEEE
53. Xu Y, Huang C, Nan Y, Lian S (2022) Tad: a large-scale benchmark for traffic accidents detection from video surveillance, https://doi.org/10.48550/arXiv.2209.12386. arXiv preprint arXiv:2209.12386
54. Ren S, He K, Girshick R, Sun J (2016) Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149. https://doi.org/10.48550/arXiv.1506.01497
55. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) Ssd: single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pp 21–37. https://doi.org/10.48550/arXiv.1512.02325 . Springer
56. Zhu X, Lyu S, Wang X, Zhao Q (2021) Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 2778–2788

Authors and Affiliations

Zhaole Ning[1] · Tianze Zhang[2] · Xin Li[1] · Aiying Wu[1] · Gang Shi[1]

✉ Gang Shi
shigang@xju.edu.cn

Zhaole Ning
107552204055@stu.xju.edu.cn

Tianze Zhang
zhangtianze.unimelb@gmail.com

Xin Li
107552204023@stu.xju.edu.cn

Aiying Wu
way@stu.xju.edu.cn

[1] School of Computer Science and Technology, Xinjiang University, Huarui Street, Urumqi 830000, Xinjiang, China

[2] Faculty of Science, The University of Melbourne, Grattan Street, Parkville, VIC 3010, Australia