Research on Deep Learning-Based Vehicle and Pedestrian Object Detection Algorithms

Xin Zhang [1], He Huang [1], Junxing Yang [1]*, Shan Jiang [1]

[1] School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, Beijing, China - zx2806404866@163.com, huanghe@bucea.edu.cn, yangjunxing@bucea.edu.cn, 2108570023131@stu.bucea.edu.cn

KEY WORDS: Intelligent Transportation Systems, Improved YOLOv8, Coordinate Attention, Vehicle and Pedestrian Dataset.

ABSTRACT:

As urbanization accelerates, traffic congestion and frequent accidents have become prominent issues, prompting the development of intelligent transportation systems. This paper focuses on the research of vehicle and pedestrian detection algorithms to improve detection accuracy in complex traffic environments. Considering the limitations of traditional object detection algorithms in complex situations, this study adopts the deep learning-based YOLOv8 algorithm and introduces the Coordinate Attention (CA) module to enhance the model's feature extraction and localization capabilities. Experimental results show that the improved YOLOv8 network achieves a 1.1% increase in detection accuracy while maintaining its original speed. Furthermore, this paper constructs a vehicle and pedestrian dataset suitable for Chinese traffic scenes, providing an effective solution for autonomous driving assistance systems. Overall, this study holds significant reference value for vehicle and pedestrian detection in the field of intelligent transportation.

1. INTRODUCTION

Object detection technology is crucial in computer vision, aiming to accurately identify and locate various targets in images and determine their categories. Since 2012, the rapid advancement of deep learning technology has significantly progressed research in this area, making object detection a focus of attention (Girshick et al., 2014; Ren et al., 2015). Currently, this technology is widely applied in fields such as autonomous driving, remote sensing, robot vision, and video surveillance, showing great application potential and commercial value.

In autonomous driving, object detection is key for autonomous vehicle driving. It enables the system to identify various road targets in real-time and accurately, providing important information for autonomous control and safe driving. Compared to traditional methods, deep learning-based algorithms have higher real-time performance, more accurate recognition capabilities, and lower false detection rates (Liu et al., 2016; Redmon et al., 2016).

Traditionally, object detection relied on algorithms suitable only for specific scenarios with weak generalization capabilities. These algorithms required extensive computation for extracting candidate regions, leading to high complexity and lower accuracy. The detection process mainly consisted of three stages: candidate region selection, feature extraction, and classifier classification (Lin et al., 2017).

In candidate region selection, the sliding window method was commonly used, leading to many overlapping boxes and increasing computational complexity. For feature extraction, manually designed features such as SIFT (Lowe, 2004), HOG (Dalal & Triggs, 2005), and SURF (Bay, Tuytelaars, & Van Gool, 2006) were used, but designing robust features was challenging. In the classifier classification stage, classifiers like SVM and AdaBoost were used, with higher requirements for speed and accuracy in multi-category detection.

The robustness of traditional methods was poor due to manually designed features in the feature extraction stage and sensitivity to environmental factors, resulting in suboptimal detection.

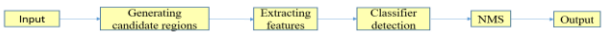


Figure 1. Steps of Traditional Object Detection Algorithms. The detection process of traditional object detection algorithms mainly consists of three stages: the candidate region selection stage, the feature extraction stage, and the classifier classification stage.

With the rapid development of deep learning, object detection methods are mainly divided into two categories: two-stage and one-stage detection algorithms, both making significant progress (Ren et al., 2015; Liu et al., 2016). Two-stage algorithms, like R-CNN, Fast R-CNN, and Faster R-CNN (Girshick et al., 2014; Ren et al., 2015), are known for their high accuracy but have high computational complexity. One-stage algorithms, like SSD and the YOLO series (Liu et al., 2016; Redmon et al., 2016), improve detection efficiency by transforming the task into a direct regression problem, suitable for real-time detection requirements but with generally lower accuracy.

One-stage and two-stage algorithms each have their advantages and limitations. One-stage algorithms are favored for their efficiency and real-time performance, while two-stage algorithms are renowned for their accuracy. Choosing the appropriate algorithm based on needs and scenario characteristics is crucial.

With continuous progress in deep learning, various datasets for object detection tasks have emerged. Publicly available datasets

* Corresponding author

include PASCAL VOC, ImageNet, MSCOCO, and Objects365 (Everingham et al., 2010; Deng et al., 2009; Lin et al., 2014; Shao et al., 2019). These datasets provide a foundation for deep learning model research, but there are differences compared to real-world scenarios. In real road scenarios, the environment is complex, and relying solely on general scene datasets, algorithms struggle to generalize, failing to meet the road detection requirements of intelligent vehicle vision perception algorithms. Therefore, high-quality datasets are crucial for ensuring algorithm performance and robustness. With continuous development, object detection and tracking algorithms have achieved a high level in general scenarios, but complex and variable environmental factors in real road scenarios still pose challenges to model performance, an urgent problem for researchers worldwide (Redmon and Farhadi, 2018; Bochkovskiy et al., 2020; Tan et al., 2020).

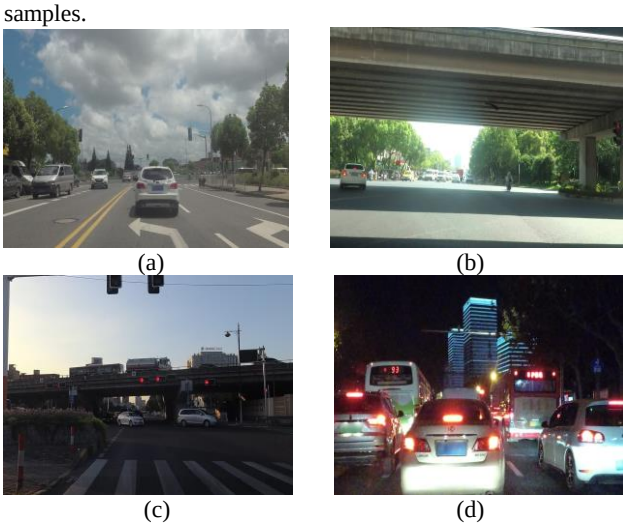
2. DATASET

Currently, datasets such as KITTI, BDD100K and Cityscapes have complex road scene changes, multiple categories of targets, and different weather conditions, which make them popular datasets among many researchers. For different detection tasks in traffic scenes, there are different requirements for the method of sample collection, the time period of collection, and the collection scene. The quality of dataset collection varies greatly, with differences in the number of samples, resolution, and image target scale, which also directly affect the testing effect and performance of deep learning models. At present, most publicly available datasets are open-sourced datasets from abroad, and there is a lack of publicly available datasets in China. Therefore, this paper uses a self-made dataset for pedestrian and vehicle target detection. The self-made dataset was captured using an HONOR 70 Pro, and it contains two categories: car and person.

2.1 Sample Acquisition of the Dataset

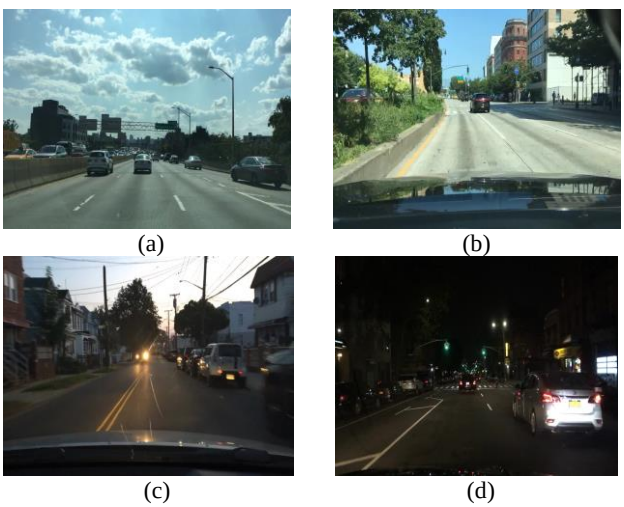
Currently, the main datasets used for evaluating intelligent vehicle environment perception algorithms are open-source datasets from abroad. Complex road scene datasets from abroad often use autonomous vehicles for collection, but this method incurs high costs, and the dataset collection process is fraught with difficulties. To align with the research on target detection and tracking under the real driving environment in China, a new complex road scene dataset has been constructed. Due to the diversity of complex scenes, occlusion scenes, and blurry scenes are the main scenes in complex road scenarios, so this paper mainly focuses on research under these situations. Firstly, the samples were obtained by placing the HONOR 70Pro smartphone inside the vehicle for shooting, with a video resolution of 4K ultra-clear and a frame rate of 30 frames per second. The complex road scene dataset constructed in this paper consists of two parts. First, videos were shot by using a smartphone during long drives in a real driving scenario in a certain area of Beijing. The videos include 15 segments, covering various traffic scenarios in the real world. The videos were manually frame-extracted at a fixed frame rate. Second, to further verify the model's generalization ability and robustness, some high-quality images from the NEXET-2017 dataset of the Kaggle competition were selected, and some sample images were obtained from websites such as Baidu and Google. Another part is the publicly available dashcam videos downloaded, which were shot with dashcams and smartphone cameras, including various weather conditions like sunny, foggy, rainy, and snowy days, and various driving road conditions like highways, congested roads, and sandy roads. To ensure the quality of the constructed dataset, the driving videos obtained by the first method were processed. First, video segments with poor quality during the collection process were filtered out. Then, each video segment was frame-extracted to convert the video into image samples, with a resolution of 640×640. The obtained images were then assigned to the corresponding annotators for strict annotation. The filtered-out videos include: videos with severe camera shake during driving, videos with no moving targets or few moving targets during long drives, and videos that are unclear due to strong light.

In summary, based on the two methods of sample acquisition, the first method collected 3200 samples, covering various complex road scenes, as shown in Figure 5. The second method collected 2300 samples, selecting representative samples from different time periods, as shown in Figure 6. The dataset is named BJCR, with a total of 5500 image samples. Finally, the first part of the dataset was expanded through image preprocessing methods, laying the foundation for improving the generalization ability of the algorithm model and stabilizing the training robustness while expanding the number of dataset samples.



(a) (b) (c) (d)

Figure 5. Examples of samples from different time periods in the first part. (a) Morning, (b) Noon, (c) Afternoon, (d) Evening.

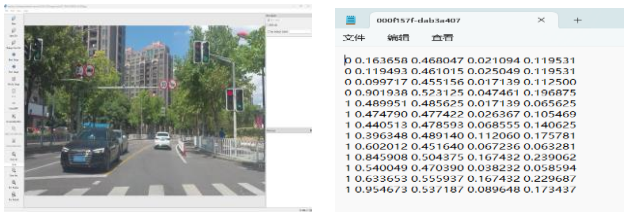


(a) (b) (c) (d)

Figure 6. Examples of samples from different time periods in the second part. (a) Morning, (b) Noon, (c) Afternoon, (d) Evening.

2.2 Classification Annotation Method

In the process of constructing a dataset, the collection of data samples should be of high quality, and at the same time, the annotation of data samples must be strictly carried out. In recent years, the technology related to object detection has been sufficiently developed, and annotation tools and platforms for dataset samples have emerged accordingly. The annotation labels can be used for classification, segmentation, object detection, and other visual tasks. Currently, annotation tools can be divided into manual annotation, semi-automatic annotation, and automatic annotation. Each of these three annotation methods has certain drawbacks. Given the needs of object detection datasets, mainstream image annotation software such as Labelme, LabelImg, LabelBee, LabelBox, RectLabel, VoTT, and Sprite Annotation Assistant have stood out. Different annotation tools should be chosen for different tasks, and the annotation results of different tools will have some differences. This study selects the LabelImg annotation software to annotate the complex road scene dataset accordingly. This software supports object detection, image segmentation, and other functions and can export labels in YOLO format. The following figure is a functional diagram of LabelImg and the YOLO format.



(a) (b)

Figure 7. LabelImg. (a) Functional diagram of LabelImg, (b) YOLO format.

3. METHOD

3.1 YOLOv8

The YOLO model has achieved great success in the field of computer vision. Based on this, researchers have improved the method and added new modules, proposing many classic models. YOLOv8 is an algorithm released by Ultralytics on January 10, 2023. Compared with previous excellent models in the YOLO series (such as YOLOv5 and YOLOv7), YOLOv8 is an advanced, cutting-edge model that provides higher detection accuracy and speed. The network structure of YOLOv8 mainly consists of a backbone, neck, and head (Wang et al., 2023). The backbone network is responsible for extracting features from the image, the neck network further processes the features, and the head network is responsible for the final object detection task, including classification and localization. See Figure 5 for details.

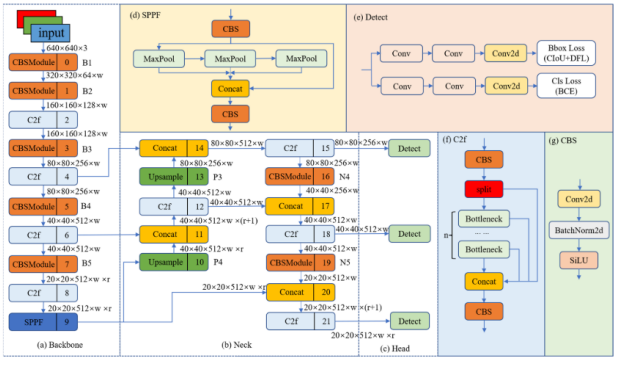


Figure 5. Structure diagram of YOLOv8. The positions of the key components are as follows: The backbone network (Backbone) is located in the (a) area, extracting basic features for the model; the neck (Neck) part is located in the (b) area, used for feature enhancement and fusion; the head (Head) is located in the (c) area, performing the final object detection task. Specifically, the (d) part corresponds to the Spatial Pyramid Pooling Fast (SPPF), (e) represents the Detection (Detect) module, (f) is the Cross Stage Partial Network (C2f), and (g) refers to the Convolution-Batch Normalization-Activation (CBS) unit.

3.1.1 Input : The input layer of YOLOv8 is not only responsible for the preprocessing of image data but also inherits and optimizes the mosaic data augmentation strategy from the YOLOv4 and YOLOv5 algorithms. This strategy is implemented by randomly selecting four images for scaling and then randomly stitching them together to expand the diversity of the training dataset. This method significantly enhances the richness of the dataset on one hand, effectively increasing the robustness of the model. On the other hand, it also helps reduce the occupancy of GPU storage space during training, thereby improving resource utilization efficiency.However, if the mosaic data augmentation strategy is continuously enabled throughout the entire training process, it may have adverse effects on the final training results. Based on this understanding, the mosaic data augmentation feature is turned off in the last 10 epochs of the training in the practice of YOLOv8. This adjustment aims to balance data diversity and model learning accuracy, ensuring that while enhancing robustness, more precise training results can also be obtained.

3.1.2 Backbone: YOLOv8 uses an improved CSPDarknet53 as its backbone network, performing downsampling five times on the input features to obtain five different scale features, denoted as B1-B5. The structure of the backbone network is shown in Figure 5(a). The original CSP (Cross Stage Partial) module in the backbone network is replaced with the C2f module. The C2f module adopts gradient parallel connections to enrich the information flow of the feature extraction network while maintaining a lightweight design. The CBS module performs convolution operations on the input information, then batch normalization, and finally activates the information flow using the SiLU activation function to obtain the output results. The backbone network uses the Spatial Pyramid Pooling Fast (SPPF) module to pool the input features into a fixed-size mapping for adaptive-sized output. Compared to the Spatial Pyramid Pooling (SPP) structure, SPPF reduces the computational load and has lower latency by sequentially connecting three max-pooling layers. The backbone network consists of the CBS module, C2f module, and SPPF module.

(1) CBS module: The CBS module includes convolution operations (Conv), batch normalization (BN), and the activation function (SiLU), which is an improvement over the CBL module. Compared to the LeakyReLU activation function in the CBL module, the smooth and non-monotonic characteristics of the SiLU function can provide better results in deep learning training.

(2) C2f module: YOLOv8 introduces a new module, the C2f module, to replace the original C3 module in the YOLOv5 network architecture. Combining the ideas of the C3 module and the ELAN module, the C2f module optimizes the module structure with gradient bifurcation connections, enriching the

information flow of the feature extraction network while maintaining a lightweight design, effectively improving the overall detection performance of the algorithm. The structure of the C2f module is as follows:

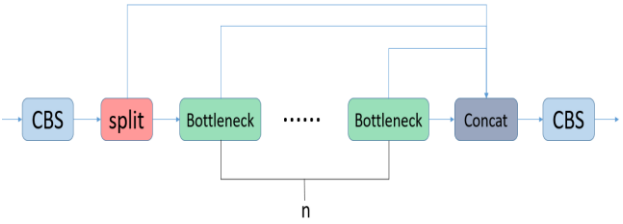


Figure 6.C2f structure diagram

(3) SPPF Module: The Spatial Pyramid Pooling Fast (SPPF) structure used in YOLOv8 is an optimization of the original SPP structure. The SPPF structure replaces the 13×13, 9×9, 5×5, and 1×1 convolution kernels in the original SPP structure with three 5×5 convolution kernels. By using multiple small convolution kernels in series, the computational load of the model is reduced, and the detection rate is improved while the detection accuracy remains close to the original structure. The structure diagram of the SPPF module is as follows:

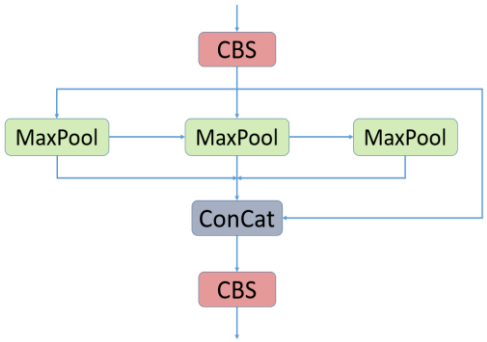


Figure 7.SPPF structure diagram

3.1.3 Neck: Inspired by PANet, YOLOv8 incorporates the PAN-FPN structure in its neck design, as shown in Figure 1b. Compared to the neck structure of YOLOv5 and YOLOv7 models, YOLOv8 removes the convolution operations after upsampling in the PAN structure, achieving model lightweight while maintaining original performance. We use P4-P5 and N4-N5 to represent two different scales of features in the PAN structure and FPN structure of the YOLOv8 model, respectively. The traditional FPN uses a top-down approach to transmit deep semantic information. FPN enhances the semantic information of features by fusing B4-P4 and B3-P3, but it may lose some target location information. To alleviate this problem, PAN-FPN adds PAN on the basis of FPN. PAN enhances the learning of position information by fusing P4-N4 and P5-N5, achieving top-down path enhancement. PAN-FPN constructs a top-down and bottom-up network structure, and through feature fusion, it achieves complementarity of shallow location information and deep semantic information, ensuring the diversity and completeness of features. The Neck layer of YOLOv8 still uses the PANet structure, composed of the Feature Pyramid Network (FPN) and Path Aggregation Network (PAN). It constructs a top-down and bottom-up network structure, effectively complementing shallow location information and deep semantic information through feature fusion, maintaining the integrity of feature information.

3.1.4 Head: The detection part of YOLOv8 adopts a decoupled head structure, as shown in Figure 1e. The decoupled head structure uses two independent branches for object classification and bounding box regression, with different loss functions for these two tasks. For the classification task, binary cross-entropy loss (BCE loss) is used. For the bounding box regression task, distribution focal loss (DFL) and CIoU are used. This detection structure can improve detection accuracy and accelerate model convergence. YOLOv8 is an anchor-free detection model, which can concisely specify positive and negative samples. It also uses a Task-Aligned Assigner to dynamically allocate samples, improving the model's detection accuracy and robustness.

(1) Decoupled Detection Head: The detection layer of YOLOv8 adopts a decoupled detection head structure, using two independent branches for object classification and bounding box regression, and different loss functions for the two tasks. On one hand, binary cross-entropy loss (BCE) is used for the classification task; on the other hand, distribution focal loss (DFL) and CIoU are used for the bounding box regression task. The decoupled head structure can improve detection accuracy and speed up model convergence.

(2) Anchor-Free: Traditional anchor-based methods rely on manually designed anchor frameworks. The size and aspect ratio of the anchor framework should be as close as possible to the real target, which has poor generality for different datasets with large differences in target sizes. The large number of anchor frames generated during the detection process not only increases the computational load but also reduces training and inference speed, and many anchor frames become negative samples because they do not achieve a certain IoU with the real frames. This leads to the problem of imbalance between positive and negative samples. The anchor-free method adopted by YOLOv8 can cope with multi-scale target changes and has good generalization ability. Since this method is not affected by the number and position of anchor frames, it has better robustness when dealing with occlusions and dense targets.

3.1.5 Label Assignment Strategy: Although YOLOv5 designed some functions for automatic clustering of candidate boxes, the clustering of candidate boxes depends on the dataset. If the dataset is not sufficient and cannot accurately reflect the distribution characteristics of the data itself, the clustered candidate boxes will also have a large disparity in size ratio compared to the real objects. YOLOv8 does not adopt the candidate box strategy, so the problem it solves is the multi-scale distribution of positive and negative sample matching. Unlike the SimOTA used by YOLOX, YOLOv8 adopts the same TOOD strategy as YOLOv6 for the label assignment problem, which is a dynamic label assignment strategy. YOLOv8 only uses $target_{bboxes}$ and $target_{scores}$, and does not include object presence prediction. Therefore, the loss of YOLOv8 mainly includes two parts: category loss and location loss. For YOLOv8, its classification loss is VFLLoss (Varifocal Loss), and its regression loss is in the form of CIoU Loss and DFL Loss. Varifocal Loss is defined as follows:

$$VFL(p,q)=\begin{cases}-q\left(qlog\left(p\right)+\left(1-q\right)log\left(1-p\right)\right) & q>0\\ -\alpha p^{\gamma}\ log\left(1-p\right) & q=0\end{cases} \quad (1)$$

In the formula, p is the predicted class score, $p\in[0,1]$.q is the predicted object score (if it is the true class, then q is the IoU

between the prediction and the ground truth; if it is another class, q is 0). VFL Loss uses asymmetric parameters to weight positive and negative samples, achieving unequal treatment of foreground and background contributions to the loss by only decaying negative samples. For positive samples, weighting is performed using q. If the GT_{IOU} of a positive sample is high, it contributes more to the loss, allowing the network to focus on high-quality samples, i.e., the training of high-quality positive examples contributes more to the improvement of AP than low-quality ones. For negative samples, down-weighting is applied by using p^{γ} , reducing the negative samples' contribution to the loss. This is because the predicted probability p becomes smaller when raised to a power greater than one, thus decreasing the overall impact of negative samples on the loss.

3.2 Coordinate Attention

Coordinate Attention (CA), presented by Qinbin Hou et al. at CVPR 2021, is a lightweight attention mechanism designed for mobile networks that considers both channel and spatial dimensions in parallel. CA addresses two issues: first, the SENet attention mechanism, while excellent, focuses only on channel-wise information without considering spatial positional information; second, the CBAM attention mechanism, despite addressing both channel and spatial dimensions, does not solve the long-range dependency issue in its spatial attention branch. CA improves upon channel-wise attention in SENet by incorporating positional information to capture spatial structure, making it a lightweight attention approach with lower module complexity than both SENet and CBAM. By embedding positional information into channel information, it enhances the feature representation of mobile networks. It mainly includes two steps: coordinate attention embedding and coordinate attention generation. The specific process of the CA attention mechanism is shown in Figure 8.

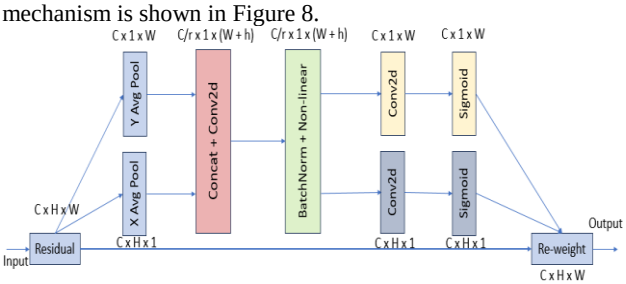


Figure 8. Flowchart of CA Attention Generation Process.

The CA attention mechanism primarily includes the following three operations:

(1) Coordinate Information Embedding: For a given input feature map, global average pooling is applied separately along the horizontal and vertical directions of the feature map to obtain two embedded information feature maps. See Figure 9 for an illustration.

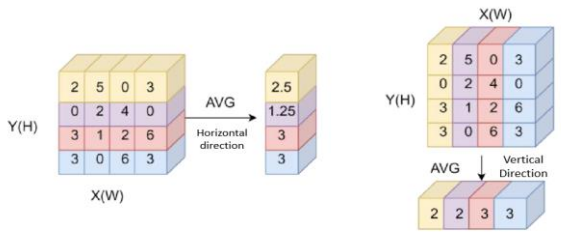


Figure 9. Diagram of CA Information Embedding Operation.

In the horizontal direction, also known as the X direction, an H ×1 pooling kernel is used to perform global average pooling on the $H \times W \times C$ input feature map to obtain an $H \times 1 \times C$ information feature map, as shown in Equation (2):

$$Z_c^W(h) = \frac{1}{H}\sum_{0\le j\le H} X_c(j,w), Z_c^w \in R^{C*1*W} \quad (2)$$

In the vertical direction, also known as the Y direction, a 1×W pooling kernel is used to perform global average pooling on the H×W×C input feature map to obtain a 1×W×C information feature map, as shown in Equation (3):

$$Z_c^W(h) = \frac{1}{H}\sum_{0\le j\le H} X_c(j,w), Z_c^w \in R^{C*1*W} \quad (3)$$

(2) Attention Generation Operation (Coordinate Attention Generation): The two information feature maps obtained from the previous step, Z_c^h and Z_c^W are concatenated along the spatial dimension. They are then passed through a 1×1 convolution operation and an activation function. Afterwards, the feature map is split along the spatial dimension to obtain two separate feature maps, which are then transformed and passed through an activation function individually to produce two attention vectors, g^h and g^w.This process is described by Equations (4)-(7).

$$f^h \in R^{\frac{C}{r}*H*1} \quad (4)$$

$$f^W \in R^{\frac{C}{r}*1*W} \quad (5)$$

$$g^h = \sigma(F_H(f^h)) \quad (6)$$

$$g^w = \sigma(F_W(f^w)) \quad (7)$$

(3) Feature Map Calibration Operation (Re-weight): The two attention vectors $g^h \in C \times H \times 1$ and $g^w \in C \times 1 \times W$ obtained from the previous operation are broadcast to match the dimensions of $C \times H \times W$,the channel dimension of the input feature map. They are then element-wise multiplied with the input feature map x_c that has gone through a residual operation. This process results in the final attention-modulated feature map. The operation is as described in Equation (8).

$$y_c = x_c \times g^h \times g^w \quad (8)$$

The CA (Coordinate Attention) mechanism addresses the long-range dependency problem through the attention generation operation where it concatenates the information feature maps Z_c^h and Z_c^w along the spatial dimension. By concatenating the global features from the horizontal direction with those from the vertical direction into an entire global feature representation, it effectively captures dependencies over longer spatial distances within the image, thus addressing the issue of long-range dependencies to a certain extent. This enables the model to better understand the overall context of the scene, which is particularly beneficial for tasks requiring an understanding of spatial relations and structures across the entire image.

3.3 Improved YOLOv8

To enhance the performance of YOLOv8 in vehicle and pedestrian detection tasks, improving detection and the accuracy of detecting small targets, while also boosting feature expression and generalization capabilities, we incorporated the CA (Coordinate Attention) mechanism into the backbone of YOLOv8 (Chi et al., 2023). Introducing the lightweight CA attention mechanism into the backbone network not only strengthens the model's perception of key features and contextual information in images but also reduces the model's computational complexity through fewer parameters and computational operations. This enables YOLOv8 to maintain high detection accuracy and also to achieve real-time detection on resource-constrained devices.

Incorporating the CA attention mechanism into the backbone of YOLOv8 brings several benefits. Firstly, the backbone network is responsible for extracting features from the input image in YOLOv8. Introducing the CA attention mechanism into the backbone can enhance the model's perception of key features and contextual information within images. The CA mechanism can adaptively adjust attention weights based on image content, focusing better on small objects and important features, thus improving the accuracy of small object detection. Secondly, the choice to add CA to the backbone is based on the pivotal role of the backbone network in feature extraction. Located at the front end of the model, the backbone network has a decisive impact on feature extraction. Introducing the CA attention mechanism into the backbone allows the model to utilize attention to model the relationship between targets and background at an early stage of feature extraction, helping to extract more discriminative feature representations. This early attention helps the model better differentiate between targets and background, increasing the accuracy of object detection. Additionally, the backbone is one of the most critical components in YOLOv8, playing a key role in the performance and speed of the entire detection model. By adding the CA attention mechanism into the backbone, the spatial and semantic information in images can be fully utilized, enhancing the representational power of targets during feature extraction. This helps improve the model's generalizability and robustness while reducing dependence on other stages.

Therefore, integrating the CA attention mechanism into the bac kbone of YOLOv8 can improve the model's detection precision for small objects and introduce attention mechanisms during the feature extraction phase, effectively utilizing key features and c ontextual information within images. This design choice can im prove the performance of object detection and provide more acc urate feature representations for subsequent processing stages, l eading to better detection results. Furthermore, the adoption of a lightweight CA attention mechanism design can further reduce the model's computational complexity and parameter count, all owing YOLOv8 to remain highly performant while also being more computationally efficient and compact in size, making it m ore suitable for deployment and application in resource-constrai ned environments.

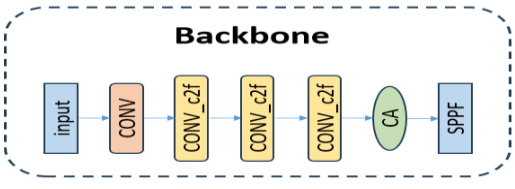


Figure 10. YOLOv8 with Added CA Attention Mechanism.

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Experimental Setup

The training and testing dataset selected for this paper was the BJCR dataset, as it specifically annotates vehicles and pedestrians, offering higher applicability compared to other datasets. The focus of this research is on the detection of vehicles and pedestrians, and other datasets do not provide detailed annotations in this regard, hence they do not meet the requirements of this study. The BJCR dataset was chosen for training and testing because it satisfies the needs of this experiment.

For the training and testing phases, the operating system used was Windows 10, and the graphics processing unit was an NVIDIA GTX1080Ti. The version of Python used was 3.8.18, utilizing the Torch 1.12.1 framework, with training and testing conducted in a cuda 11.3 accelerated environment.

The choice of operating system, graphics processor, Python version, and acceleration environment ensured the reliability of the experimental results.

4.2 Parameter Settings and Evaluation Metrics

The input image size was set to 1280×720 pixels, with the optimizer being Adam. The learning rate was set at 0.013, momentum at 0.937, and weight decay at 5e-4. The learning rate was adjusted using the cosine annealing algorithm, the batch size was set at 32, and the training duration was 100 epochs.

In terms of detection accuracy, the mean Average Precision (mAP) was used as the evaluation metric, with the IoU threshold set to greater than 0.5, i.e., mAP@0.5. mAP is the mean of the Average Precision (AP) across all object categories. The mAP value reflects the accuracy of the model on the dataset. The specific calculation method for mAP is shown in Equation (9):

$$mAP = \frac{1}{c}\sum_{(i=1)}^{c} AP_i \tag{9}$$

In the equation,c represents the number of object categories to be detected, and i is the index of each category. The AP (Average Precision) value is calculated as the area under the curve plotted with precision and recall values, ranging from 0 to 1. It provides a comprehensive measure of both precision and recall for a specific category. The definitions of precision and recall are given in Equations (10) and (11), respectively:

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

In the formulas,TP represents the number of true positives,FP represents the number of false positives, and FN represents the number of false negatives. Precision reflects the proportion of true positives in the positive samples predicted by the model, while recall reflects the proportion of positive samples correctly predicted by the model out of the total positive samples.

For evaluating detection speed, the number of parameters (Params) in the model and the number of frames processed per second (Frames Per Second, FPS) are used as metrics.

4.3 Ablation Study

To verify the effectiveness of the improvements made in our model, we designed two sets of ablation experiments on the BJCR dataset. The results of the ablation study are presented in Table 1.

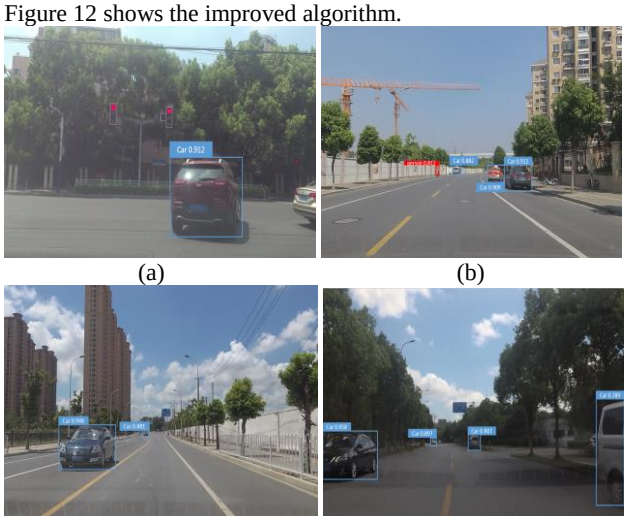
In the table, " √ " indicates that the component is included, while "" indicates it is not included. From the table above, it can be seen that in the experiment without the CA attention mechanism, the number of parameters is 30.18M, the precision is 0.885, the recall rate is 0.83, the average precision is 0.905, and the FPS reaches 760, with 8.2 GFLOPs, reflecting the consistent approach of the YOLOv8 algorithm, which is to improve real-time performance while ensuring a certain level of accuracy. From the second row of data, it is found that after integrating the CA attention mechanism, the precision and average precision increased by 1.7% and 1.1%, respectively, but this came with an increase in the number of parameters and a decrease in real-time performance, with FPS decreasing from 48.3 to 47.6. Considering all the data, the method proposed in this paper is effective, improving the average precision by 1.1%

Group	CA	Params(M)	Precision(%)	Recall(%)	mAP@0.5(%)	FPS	GFLOPs
1	\	30.06	0.881	0.813	0.905	48.3	8.2
2	√	30.09	0.898	0.830	0.916	47.6	8.2

Table 1. Ablation Study. Group1 is the original algorithm, and Group2 is the algorithm after adding the CA attention mechanism.

and the precision by 1.7% while ensuring real-time performance. The experimental results show that the improved algorithm can correctly detect vehicles and pedestrians during data inference (the data used for inference is randomly selected from the BD100K dataset, which did not participate in any training or testing phases), and can still correctly identify them under interference at different times, without false detections or missed detections.

Furthermore, the experimental results show that this improvement in accuracy is especially evident in the detection of small targets. Figure 11 shows the original algorithm, while Figure 12 shows the improved algorithm.



(a) (b)

(c) (d)

Figure 11.YOLOv8

(a) (b)

(c) (d)

Figure 12.Improved YOLOv8

Comparing Figures 11 and 12, it can be observed that the improved algorithm in Figure 12 demonstrates enhanced detection capabilities. Specifically, Figure 12.a successfully detects a car on the right rear side that is missed in Figure 11.a. In Figure 12.b, a car that is obscured in Figure 11.b is accurately detected. Additionally, Figure 12.c is able to detect a distant car that is not detected in Figure 11.c. Moreover, Figure 11.d fails to detect a distant car and an obscured car, whereas Figure 12.d successfully identifies both. These improvements indicate that the modifications made in the algorithm for Figure 12, such as the integration of the CA attention mechanism into YOLOv8, have significantly enhanced the model's ability to detect vehicles, especially in challenging scenarios involving occlusion and distance.

5. CONCLUSION

In response to the relatively low availability of datasets specifically tailored to the unique characteristics of Chinese traffic, this paper creates and annotates a vehicle and pedestrian dataset from a certain area in Beijing, named BJCP. This dataset encompasses images under real-world conditions and aligns with actual application scenarios, making it suitable for testing various object detection models. To address the challenges associated with detecting small objects in images, this paper proposes a small object detection algorithm based on the YOLOv8 algorithm, enhanced with the integration of the CA (Coordinate Attention) attention mechanism. By incorporating the CA attention mechanism into the backbone of the model, the feature extraction capability for small objects is enhanced, thereby improving detection accuracy. Experimental results demonstrate that the algorithm proposed in this paper outperforms the current state-of-the-art YOLOv8 algorithm in

vehicle and pedestrian detection. The algorithm significantly enhances detection accuracy while maintaining high-speed inference.

In summary, the BJCR dataset and the improved YOLOv8 algorithm proposed in this paper are of significant reference value for vehicle and pedestrian detection in the context of autonomous driving technology. They effectively enhance detection accuracy while preserving the advantage of high-speed inference, providing a reliable solution for vehicle and pedestrian detection in autonomous driving.

ACKNOWLEDGEMENTS

This research was funded by the National Natural Science Foundation of China (Grant Numbers 42201483) and the China Postdoctoral Science Foundation (Grant Numbers 2022M710332).

REFERENCES

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded Up Robust Features. In Proceedings of the 9th European Conference on Computer Vision - Volume Part I (ECCV '06) (pp. 404-417).

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

Chi, X., Huang, H., Yang, J., Zhao, J., and Zhang, X. (2023): D ataset and improved YOLOv7 for text-based traffic sign detecti on, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XL VIII-1/W2-2023, 881–888, https://doi.org/10.5194/isprs-archiv es-XLVIII-1-W2-2023-881-2023.

Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05) (Vol. 1, pp. 886-893).

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 248-255).

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, 88(2), 303-338.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

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).

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In European conference on computer vision (pp. 740-755).

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In European conference on computer vision (pp. 21-37).

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60(2), 91-110.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., & Sun, J. (2019). Objects365: A Large-Scale, High-Quality Dataset for Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (pp. 8430-8439).

Tan, M., Pang, R., & Le, Q. V. (2020). EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10781-10790).

Wang, G., Chen, Y., An, P., Hong, H., Hu, J., Huang, T. (2023): UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios. Sensors, 23, 7190. https://doi.org/10.3390/s23167190.