IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 February 2024, accepted 13 March 2024, date of publication 18 March 2024, date of current version 26 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3378511

RESEARCH ARTICLE

Small-Scale Pedestrian Detection Using Fusion Network and Probabilistic Loss

HONGCHANG ZHANG, KANG YANG, HENG LIU, JIALI HU, YAO SHU, AND JUAN ZENG

Department of Automotive Service Engineering, Wuhan University of Technology, Wuhan 430070, China

Corresponding author: Juan Zeng (yforsakek@163.com)

ABSTRACT Small-scale pedestrian detection is a challenge. The main issues are as follows: 1) Troubled by their small scale, it is difficult to extract features effectively; 2) During the detection process, it is easily disturbed by background noise such as inter-class occlusion and intra-class occlusion, leading to missed or false detection; 3) The current widely used IoU measurement method is very sensitive to the position deviation of small objects, which seriously reduces the detection performance. To address these problems, we improve YOLOv5 structure by integrating Non-Local and Convolution structures, building a new feature extraction module called ResNet-Conv&NonL, combined with the ResNet structure. This module was then integrated into the backbone of YOLOv5 for better image feature extraction. In addition, we developed a novel model to measure the similarity between bounding boxes, which are embedded in the loss function of the YOLOv5 structure to replace the normal IoU measurement. Experiments on a self-made dataset and a combined dataset from Caltech and CityPersons show the feasibility of the proposed network structure. Results demonstrate the feasibility of the improved network structure is superior to the original method because it increases average precision by 6.0% compared to the original one.

INDEX TERMS Convolution, loss function, non-local, small-scale pedestrian detection, YOLOv5.

I. INTRODUCTION

Owing to the expanding research into artificial intelligence theory and deep learning technology, object detection, which is a core problem in the field of computer vision, has made significant strides. Numerous fields have been extensively, including face detection, pedestrian detection, activity recognition, vehicle detection, and autonomous driving [1], [2], [3], [4], [5]. Among these, pedestrian detection technology stands out as a challenging yet crucial task in computer vision. In recent years, visual-based pedestrian detection technology has gradually attracted attention and grown to be an essential part of technologies like vehicle-assisted driving systems and intelligent monitoring systems [6]. Pedestrian detection technology can help cars automatically identify pedestrians on the road so that they can make timely and accurate judgments during driving.

In real-world traffic scenarios, pedestrians, as the main actors, roam freely, and their position relative to the camera is uncertain [7], [8]. Vehicles can identify pedestrians on the road with the use of pedestrian detection technology, enabling rapid and precise driving decisions. The paper's main focus is detecting small-scale pedestrians, who appear relatively small when they are far from the camera. We define pedestrians whose height is less than 80 pixels in the image as small-scale pedestrians.

Small-scale pedestrians are far more difficult to detect compared with medium-to-large-sized pedestrians at close range. The main challenges of small-scale pedestrian detection are as follows: small-scale pedestrians are easily disturbed by background noise during the detection process, resulting in missed detection and false detection by the algorithm. Furthermore, it is challenging to balance detection accuracy and speed. Therefore, small-scale pedestrian detection remains important research in the field of object detection [9].

Recent years have witnessed a spurt of progress in deep learning technology. Compared with traditional detection methods such as HOG and SVM, deep learning-based detection methods do not require manual feature extraction and

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenhua Guo.

can be learned end-to-end. Meanwhile, they possess powerful learning capabilities and can learn object features from massive training data autonomously and quickly [10].

As a result, deep-learning-based pedestrian detection methods have become mainstream. From the perspective of algorithmic processing, deep-learning-based object detection algorithms can be divided into two-stage algorithms and one-stage algorithms. Two-stage algorithms first screen candidate boxes, then check if they include the target object, and finally adjust the position of the object. However, these algorithms are not suitable for all implementation cases. Such algorithms include R-CNN [9], Fast R-CNN [11], and Faster R-CNN [12]. One-stage algorithms directly regress the position coordinates of the target box and the classification probability of the target rather than screening candidate boxes. Therefore, these algorithms were faster. Such algorithms include SSD [13], YOLO [14], YOLOv3 [15]. In 2020, the YOLOv5 [16] algorithm was proposed, which not only absorbs the benefits of the previous algorithm but also greatly simplifies the model. While maintaining high accuracy, it improves the detection accuracy and leads to trends in the field of object detection. Therefore, we select YOLOv5 as the network structure for improvement.

This paper presents a small-scale pedestrian detection network structure based on YOLOv5. First, for small-scale pedestrian detection, IoU is sensitive to small positional deviations because the target is small. As shown in Fig. 1, the blue box represents the Bounding Box and the red box represents the ground truth box. At the same positional deviation, the IoU between the two boxes for the small object is significantly reduced.

FIGURE 1. Comparison of IoU for large and small objects.

Therefore, the IoU is not a useful metric for small targets. In this paper, a new metric (L_{G-E}) is proposed to measure the similarity between the bounding box and the ground truth box, as a way to replace the IoU metric used in YOLOv5. This new metric is more suitable for judging the similarity between small pedestrian targets because it is less sensitive to targets of various scales. Then, owing to the good performance of self-attention mechanisms in many visual tasks, this study designs a module that integrates the self-attention mechanism and convolution layer (ResNet-Conv&NonL module) into the backbone of YOLOv5. This module inherits the advantages of the self-attention mechanism and convolution and effectively improves the accuracy of model detection under the premise that the number of parameters is not significantly different.

The rest of this paper is organized as follows: Section II reviews pedestrian detection algorithms based on deep learning. Section III discusses the proposed small-scale pedestrian detection algorithm in detail. Section IV presents experimental results and discussion. Section V summarizes the content and results of this study.

II. RELATED WORKS

A. STRUCTURE OF YOLOv5

Several studies have been conducted on object detection methods based on deep learning [17]. Among these, the YOLO series is a typical one-stage object detection method. Based on convolutional neural networks, You Only Look Once (YOLO), a real-time object detection model, can achieve good real-time detection effects through end-to-end learning with massive data [18]. YOLOv5 is an improved one-stage detection network that is based on YOLOv4 [19]. After learning from previous versions and other network advantages, YOLOv5 balances detection accuracy and real-time performance, not only meeting the needs of real-time image object detection but also having a smaller structure. Hence, we used YOLOv5 as the basic object detection model. YOLOv5's structure is mainly divided into four parts: input, backbone, neck, and prediction. Its network structure is shown in Fig. 2.

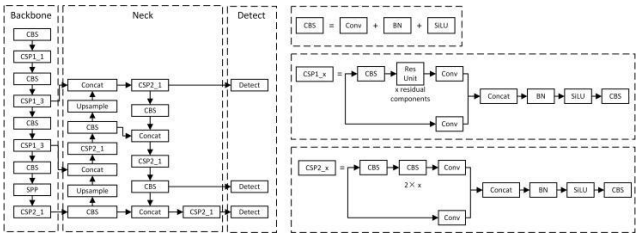


FIGURE 2. Structure of YOLOv5.

B. TINY OBJECT DETECTION

Several approaches have been proposed to address the low detection accuracy of small targets. It is suggested to copy and paste small targets into various locations within the image to augment the number of small target samples [20]. However, this method may not accurately reflect real-world scenarios, potentially resulting in overfitting. An instance scale normalization method is proposed to standardize the scale of the image pyramid and mitigate scale variation issues [21]. Nevertheless, this approach might not fully tackle the inherent complexities associated with small object features. A feature pyramid network [22] structure has been proposed to merge feature maps, effectively integrating the semantic information from deep networks with the detailed information from shallow networks to enhance detection performance. However, this method increases computational complexity,

possibly rendering it impractical for real-time applications. Furthermore, a novel clustering detection network has been introduced, integrating object clustering and detection to enhance detection efficiency [23]. Yet, its efficacy in dealing with highly overlapping objects remains uncertain. Another proposal involves a scale-sensitive loss function aimed at prioritizing small objects in model updates [24]. Additionally, a feature fusion method employing deconvolution and variable convolution has been suggested to generate high-resolution feature maps. However, the performance of this approach may heavily rely on hyperparameter selection. Moreover, a multi-way branch network has been utilized to refine both global semantic and detail features of high-resolution images [25]. This network then integrates these features layer by layer to enhance both details and background features of small objects on the feature map. Additionally, self-learning anchor boxes have been introduced to adapt feature maps to anchor box positions and shapes.

C. SELF-ATTENTION

It has been demonstrated that self-attention has the potential to entirely substitute convolution operations in visual models [26]. While this method provides flexible global dependency modeling, it might also escalate computational requirements. The suggestion is made to treat an image as 256 tokens [27] and feed them into a transformer model [28] to attain satisfactory outcomes. However, scalability issues might arise when dealing with larger images. Inspired by traditional non-local mean filtering methods, Non-Local [29] is proposed in convolutional neural networks, where the response of a pixel is the weighted sum of all features from other points, allowing each point to be associated with all other points. The increased complexity could challenge its use in constrained environments. Position attention modules and channel attention modules are proposed to establish rich contextual relationships on local features, greatly improving the segmentation results [30]. However, the impact on real-time processing needs further investigation.

D. IoU METRICS

The IoU is the most widely used measure of similarity between bounding boxes. However, the traditional IoU can only be used when bounding boxes overlap. In an effort to tackle this limitation, adjustments proposed [31] include the incorporation of a penalty term for the smaller bounding box conversion. However, when the two boxes exhibit a containment relationship, GIoU regresses to IoU, failing to discern their relative positional relationship. Alterations suggested [32] adjust the penalty term of the minimum bounding box in GIoU to the normalized distance between the centers of the two bounding boxes, aiming to balance both distance and overlapping area. Additionally, the aspect ratios of the two bounding boxes are incorporated into the formula. However, this adjustment may not fully encapsulate the complexity of object shapes and their spatial relationships. The approach [33] involves segregating the aspect ratio and computing the relationship between the length and width of the bounding box and the ground truth box individually. The devised loss function [34] is characterized by global continuous differentiability and unique extremum, guaranteeing the existence of a global gradient and facilitating the bounding box's return to the extremum.

E. PEDESTRIAN DETECTION

In order to achieve a more appropriate number and size of prior frames, the prior frame is implicitly processed, and Smooth-L1 regularization is introduced to expedite reasoning [35]. The feature map is modeled [36] through the combination of the self-attention module and the channel attention module, fully leveraging pedestrian context information and channel information to enhance pedestrian characteristics. Although it effectively utilizes contextual information, the computational overhead may restrict its deployment in real-time applications. The proposal introduces a double-head detection algorithm [37] that merges anchor-based and anchor-free detectors, overseeing both head detection and whole-body detection to mitigate missed detections effectively. While the dual supervision mechanism enhances detection reliability, it incurs heightened computational complexity. Designing two simultaneous detection branches for the entire pedestrian and the head [38], the approach employs a no-anchor frame method to generate pedestrian head bounding boxes and overall bounding boxes from the feature map's center point. The proposal [39] introduces a new pedestrian enhancement module and a pedestrian secondary detection module, integrating semantic information segmentation into a shared layer to mitigate background interference effectively. A new pedestrian detection method [40] is proposed to treat target detection as a high-level semantic feature detection task and directly forecasting center points and scales through convolution, thereby streamlining the pedestrian detection process. Additionally, it presents a new pedestrian detector with no anchor points, boasting a simple structure and excellent performance. The proposal [41] introduces A scale-aware hierarchical detection network for pedestrian detection issues amidst large-scale changes. It enriches feature pyramid representation by incorporating a cross-scale feature aggregation module and seamlessly integrating it within a unified framework, enabling multi-scale pedestrian detection. The proposal [42] suggests an active pedestrian detector to tackle the performance degradation issue of small-sized pedestrian detection. It leverages the rich feature hierarchy and initial pedestrian proposals offered by ResNet and Faster R-CNN, operating explicitly on multilayer neural representations, thereby achieving substantial reduction in false detection rates. An effective visual tracking framework named SiamDTH [43] is introduced which distinguishes itself in Siamese-based network visual tracking through feature decoupling and different tracking

head structures for classification and regression tasks, thus yielding more robust classification predictions and accurate regression predictions. While these frameworks enhance detection accuracy, their computational overhead may constrain real-time application deployment.

III. PROPOSED METHOD

A. RESNET-CONV&NONL MODULE

1) INTEGRATE NON-LOCAL AND CONVOLUTION

(1) Convolution is an efficient method for image feature extraction that utilizes learnable kernels to capture multilevel features. As the number of network layers increases, the features gradually evolve from low-level to complex. The local features of images are extracted by local operations with fixed kernels. However, to capture long-distance dependencies, convolutional layers are often stacked to increase receptive fields. This will increase the number of parameters that are difficult to optimize. This study combines convolution with non-local to remodel the convolution operation for optimization purposes. The convolution calculation process is illustrated in Figure. 3.

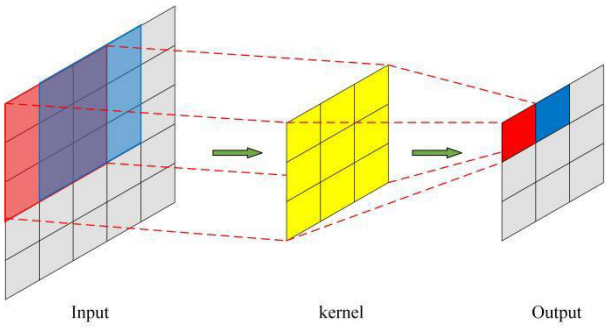


FIGURE 3. The process of convolution operation.

We model the above computation process and set the convolution kernel as $F \in \mathbb{R}^{C_{out} \times C_{in} \times n \times n}$, where n is the size of the convolution kernel, C_{out} and C_{in} are the numbers of channels for the output and input feature maps, respectively. Let $I \in \mathbb{R}^{C_{in} \times H_{in} \times W_{in}}$ and $O \in \mathbb{R}^{C_{out} \times H_{out} \times W_{out}}$ be the input and output feature maps, where H and W are the height and width of the feature maps, respectively. We let $I_{i,j} \in \mathbb{R}^{C_{in}}$ and $O_{i,j} \in \mathbb{R}^{C_{out}}$ be the channel vectors corresponding to the i and j positions. Then, the above computation process can be represented as follows:

$$O_{i,j} = \sum_{x,y} F_{x,y} I_{i+x-\lfloor n/2 \rfloor, j+y-\lfloor n/2 \rfloor} \tag{1}$$

where $F_{x,y} \in \mathbb{R}^{C_{out} \times C_{in}}$, $x, y \in \{0, 1, \cdots, n-1\}$ represent the weight of the convolution kernel at position x, y with respect to the input channel vectors.

When we break down the above calculation process, we can observe that the original convolution operation involves aggregating with $F \times I$, shifting to the next position, and then convolving. This process can be further decomposed into a feature transformation of $I_{i,j}$ using a 1×1 convolution kernel, followed by shifting and then aggregation. The specific process is as follows: Stage I: We decompose the original single $n \times n$ convolution kernel into $n \times n$ convolution kernels. Then we perform feature transformation on the input feature map according to the kernel weights, which is expressed as follows:

$$O_{i,j}(x, y) = F_{x,y} I_{i,j} \tag{2}$$

Stage II: The transformed feature maps are shifted according to the position of the current kernel in the original convolution kernel, which can be represented as:

$$O_{i,j}(x, y) = F_{x,y} I_{i+x-\lfloor n/2 \rfloor, j+y-\lfloor n/2 \rfloor} \tag{3}$$

Then the resulting feature maps from the previous step are aggregated based on the positions of the kernels, which can be expressed as:

$$O_{i,j}(x, y) = \sum_{x,y} f(x, y, i, j) \tag{4}$$

Inside:

$$f(x, y, i, j) = F_{x,y} I_{i+x-\lfloor n/2 \rfloor, j+y-\lfloor n/2 \rfloor} \tag{5}$$

The complete decomposition process of the convolution operation is shown in Figure. 4.

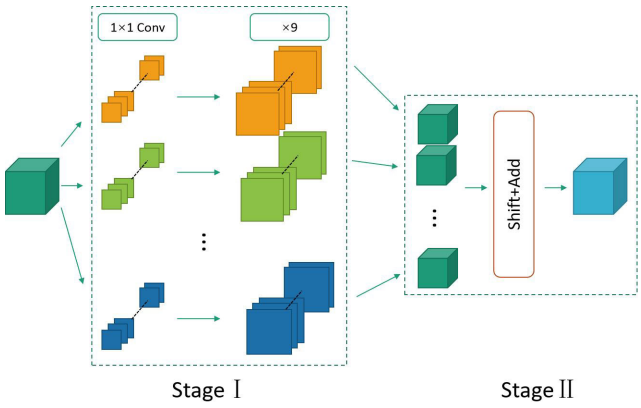


FIGURE 4. Decomposition of 3 × 3 convolution.

(2) Non-Local operations directly capture long-distance dependencies by computing interactions between any two locations. This structure can be used as a component in combination with the YOLOv5 network structure. The general formula for non-local is as follows:

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_{i,j}, x_{a,b}) g(x_{a,b}) \tag{6}$$

In this paper, $C(x)$ represents the feature map output from the previous layer, the function f computes the similarity between $x_{i,j}$ and $x_{a,b}$, and the function g computes the value of the feature map at position (a, b).

More specifically: given input feature $I \in \mathbb{R}^{C_{in} \times H \times W}$ and output feature $O \in \mathbb{R}^{C_{out} \times H \times W}$, let $x_{i,j} \in \mathbb{R}^{C_{in}}$ and

$x_{a,b} \in \mathbb{R}^{C_{in}}$ denote the vectors at position (i, j), (a, b) on the feature map. The g function can be viewed as a linear transformation (linear embedding), which is formulated as follows:

$$g(x_{a,b}) = W_g \cdot x_{a,b} \tag{7}$$

$$f(x_{i,j}, x_{a,b}) = e^{W_\theta x_{i,j}^T W_\phi x_{a,b}} \tag{8}$$

$$C(x) = \sum_{\forall i,j} f(x_{i,j}, x_{a,b)} \tag{9}$$

We substituted the formula into the general formula. This can be expressed as:

$$O_{i,j} = \frac{1}{\sum\limits_{\forall i,j} e^{W_\theta x_{i,j}^T W_\phi x_{a,b}}} e^{W_\theta x_{i,j}^T W_\phi x_{a,b}} W_g x_{a,b} \tag{10}$$

Decomposing the aforementioned computation process, we can break it down into the following steps: First, we project the input feature map using a 1×1 convolutional kernel. We then compute the attention weights. Finally, we aggregate them using a value matrix. The specific process is as follows:

Stage I: $\theta_{i,j} = W_\theta x_{i,j}, \phi_{i,j} = W_\phi x_{i,j}, g_{i,j} = W_g x_{i,j}$

Stage II: $O_{i,j} = \frac{f(x_{i,j}, x_{a,b})}{C(x)} g(xa, b)$

The complete decomposition process of Non-Local operation is shown in Figure. 5.

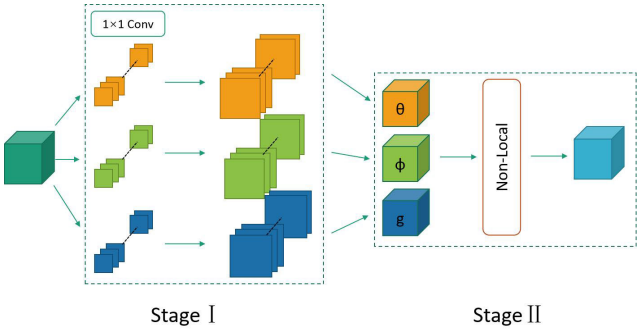


FIGURE 5. Decomposition of non-local.

(3) Stage I: By projecting the input features with three 1×1 convolution kernels and reshaping them into N pieces, rich intermediate features containing $3 \times$N feature maps can be obtained.

Stage II: For the Convolution path, a lightweight fully connected layer is used to generate N sets of n^2 feature maps, which are shifted and aggregated using the formula. For the Non-Local path, the intermediate features are grouped into N sets, each containing three features corresponding to θ, ϕ and g. The fusion process is illustrated shown in Figure 6. Finally, the outputs of the two paths are added and assigned weights respectively.

$$F_{out} = \lambda F_{\mathrm{NonL}} + \mu F_{Conv} \tag{11}$$

Combining the self-attention mechanism with convolution is beneficial for the feature extraction. They perform similar

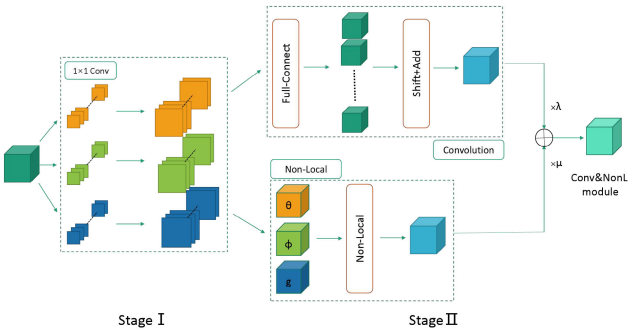


FIGURE 6. Fusion of non-local and convolution.

feature learning operations in the first stage, through 1×1 convolution projecting features into deeper space, sharing similar computational costs. The second stage, the aggregation of features, is relatively lightweight in computation and does not require additional learning parameters. Therefore, this module has a similar computational overhead in the first stage of operation, but a lower computational overhead in the second stage, making the entire model more efficient in extracting features of small-scale pedestrians in the image.

2) ADD RESIDUAL UNIT

Based on the Conv&NonL module described above, a residual unit with two inputs, namely, the main input and the residual input, is introduced, which helps to avoid information loss while reducing the number of parameters and computational requirements. The ResNet-Conv&NonL module can overcome the problems of gradient explosion and vanishing that occur in deep networks. Moreover, it is more sensitive to changes in network weights. The results are shown in Fig. 7.

FIGURE 7. Structure of ResNet-Conv&NonL.

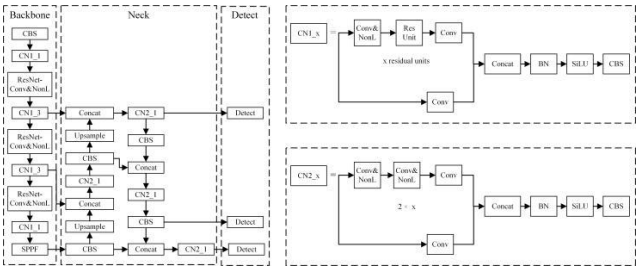


FIGURE 8. Improved structure.

B. MODELING OF BOUNDING BOX PROBABLITY DISTRIBUTION

For small pedestrian targets, most actual pedestrian targets are not strictly rectangular. Because of the pedestrians' unique "head-shoulder" features, they exhibit a "convex" shape.

Therefore, in these bounding boxes, the foreground and background factors are mainly concentrated in the blue and white areas of the bounding box, respectively, as shown in Fig. 9. To describe the weights of different pixels in small pedestrian bounding boxes better, a probability distribution for the bounding box is modeled.

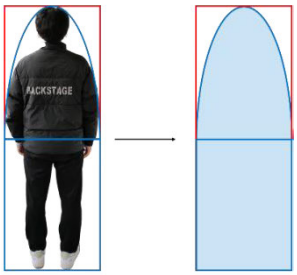


FIGURE 9. Distribution of prospect and background factors.

1) MODELING OF UPPER PART OF THE BOUNDING BOX

Dividing the bounding box into upper and lower parts, the upper part contains the "head-shoulder" features of pedestrians, so the center pixel has the highest weight, and the importance of pixels decreases from the center towards the boundary. Therefore, the upper part of the bounding box can be regarded as a two-dimensional normal distribution. We set the bounding box as: $B = (b_x, b_y, w, h)$, where b_x, b_y, w, and h represent the center coordinates and width and height of the bounding box. The probability density function of a two-dimensional Normal Distribution is:

$$f(x) = \frac{1}{2\pi\,|\Sigma|} e^{\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)} \tag{12}$$

where μ represents the mean vector and Σ represents a co-variance matrix of Gaussian distribution.

We approximate the probability distribution contour as half an ellipse, the ellipse is:

$$\begin{cases} \dfrac{(x-b_x)^2}{(w/2)^2} + \dfrac{(y-b_y)^2}{(h/2)^2} = 1 \\ y \geq b_y \end{cases} \tag{13}$$

So:

$$\mu = \begin{bmatrix} b_x \\ b_y \end{bmatrix}, \quad \Sigma = \begin{bmatrix} (w/2)^2 & 0 \\ 0 & (h/2)^2 \end{bmatrix}$$

We use Wasserstein distance to measure distribution distance. The Wasserstein distance between two normal distributions is calculated as follows:

$$d^2 = \|m_1 - m_2\|_2^2 + \left\|\Sigma_1^{1/2} - \Sigma_2^{1/2}\right\|_F^2 \tag{14}$$

The Wasserstein distance between $B_i = (x_i, y_i, w_i, h_i)$ and $B_j = \left(x_j, y_j, w_j, h_j\right)$ is defined as:

$$d^2\left(B_i, B_j\right) = \left\| \left(\left[b_i, b_i, \frac{w_i}{2}, \frac{h_i}{2}\right]^T, \left[b_j, b_j, \frac{w_j}{2}, \frac{h_j}{2}\right]^T \right) \right\|_2^2 \tag{15}$$

We map the distance to the range of 0 to 1 to obtain a new similarity measure called Gaussian Wasserstein Distance (GWD):

$$F_1 = GWD = \frac{1}{1 + e^{-d^2(B_i, B_j)}} \tag{16}$$

2) MODELING OF LOWER PART OF THE BOUNDING BOX

For the lower part of the bounding box, which approximately has a rectangular shape, the EIoU loss is utilized.

$$F_2 = IoU - \frac{\rho^2\left(b, b^{gt}\right)}{c^2} - \frac{\rho^2\left(w/2, (w/2)^{gt}\right)}{c_{w/2}^2} - \frac{\rho^2\left(h/2, (h/2)^{gt}\right)}{c_{h/2}^2} \tag{17}$$

3) BOUNDING BOX MODELING AGGREGATION

Finally, the upper and lower parts of the bounding box are integrated to obtain the new loss function: L_{G-E}

$$L_{G-E} = 1 - (\alpha F_1 + \beta F_2) \tag{18}$$

Among these, α and β are weight matrices for the similarities of the upper and lower parts of the bounding box, respectively.

Finally, we design L_{G-E} as a novel loss function and replace it with the original CIoU loss function.

This study introduces a loss function module based on the probability density function for pedestrian head and shoulder characteristics. The design of this module can overcome the insensitivity between pedestrian targets at different scales, thereby more effectively measuring the similarity between small targets.

IV. EXPERIMENT

The experiment was implemented using the Ubuntu 18.04 operating system, AMD EPYC 7543 CPU processor, GeForce RTX3090 GPU graphics card, 80GB memory, CUDA11.1 for training acceleration, PyTorch 1.8 deep learning framework for training, and Logitech camera.

A. INTRODUCTION AND ANALYSIS OF DATASET

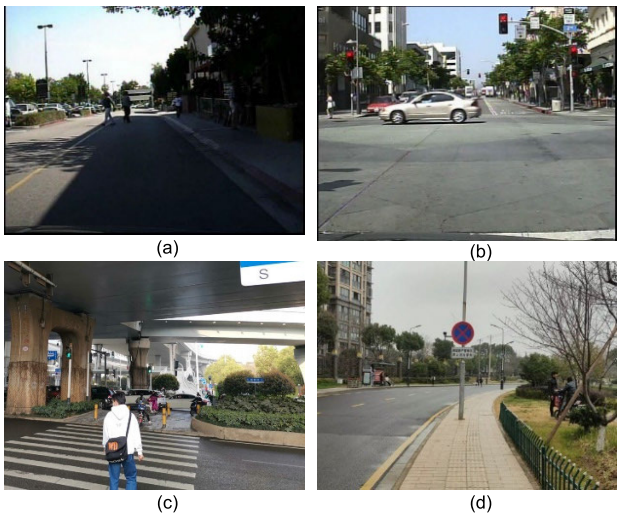
The dataset used in this study was the Caltech [44] and CityPersons [45] dataset. Caltech was collected by researchers at the Computer Science Department of the California Institute of Technology. It includes approximately 10 hours of 720P HD video with over 250,000 frames and various scenes, such as streets, parking lots, and campuses. The videos were simultaneously captured by two cameras from different angles, capturing images of pedestrians at different distances and angles, to ensure the diversity and robustness of the data.

Because YOLOv5 is unable to directly recognize the data and annotations in the Caltech dataset, pre-processing is necessary. This involves converting the seq files into 640 × 480 pixel PNG images and transforming the VBB annotation files into TXT files that are recognizable by the model.

All annotations were labeled as ''person.'' Finally, the dataset was filtered to obtain a total of 40,000 images.

Although the Caltech dataset contains many small-scale pedestrians, there is a problem: the annotation boxes are not completely accurate, and many of the pedestrian annotations in the dataset are problematic. Therefore, to better evaluate the performance of the algorithm, we filmed videos on campus, stations, and other locations during the day. Then we use code to reshape and extract frames from the videos, and use ''LabelImg'' to label the pictures, finally obtaining a self-made dataset. The dataset contains 10,000 images and 41,435 positive samples, with a resolution of 640 × 480 pixels.

The Caltech dataset and self-made dataset were integrated to form a mixed dataset containing 50,000 images. The training, validation and test set ratios were set to 6:2:2, with 30,000 training images, 10,000 verification images, and 10,000 test images. Some of these are shown in Fig. 10.



(a)poor light (b)brighter light

(c) Inter-class occlusion (d) Intra-class occlusion

FIGURE 10. Pictures display of different feature data sets.

B. EVALUATING INDICATOR

Precision, recall, and Average Precision (AP) metrics were used to evaluate the proposed method.

Precision refers to the ratio of the number of targets that are correctly detected as positive samples to the number of targets that are all detected as positive samples. It measures the rate of correct detection among all the positive samples. Recall refers to the ratio between the number of targets that are correctly detected as positive samples and the number of targets in all actual positive samples. It measures the number of positive samples that the model identifies and avoids missing detection. The miss rate (MR) is the ratio of the number of pedestrians not detected by the algorithm to the total number of actual pedestrians. The specific formulae are as follows:

$$precision = \frac{TP}{TP + FP} \tag{19}$$

$$recall = \frac{TP}{TP + FN} \tag{20}$$

$$MR = 1 - recall \tag{21}$$

Here, TP represents the number of true positive samples, FN represents the number of false negative samples, and FP represents the number of false positive samples.

Algorithms often cannot consider both model precision and recall into account. Improving precision usually decreases recall and vice versa. To better evaluate the performance of the algorithm, we use the F1-score to consider both the precision and recall. The F1 score is the harmonic mean of the precision and recall, which is used to comprehensively consider these two indicators. That can help balance the trade-off between the precision and recall. Only when precision and recall are both very high, the F1 value will increase. F1 is defined as:

$$F1 = 2 \times \frac{R \times P}{R + P} \tag{22}$$

Here, R means *recall*, and P means *precision*. AP is a comprehensive performance metric that calculates the area under the precision-recall curve for each category, and then averages these areas to quantify the model's accuracy in detecting objects of different categories. The value of AP ranges from 0 to 1, and the formula is as follows:

$$AP = \sum_{k=0}^{k=n-1} \left(R_c\left(k\right) - R_c\left(k+1\right) \times P_r\left(k\right)\right) \tag{23}$$

Among these AP_C represents the average precision of the target category of C, n represents the total number of images in the category, and in this experiment, there is only one category, person, so $C = 1$.

False positive per image (FPPI) refers to the number of incorrectly detected non-pedestrian objects in each image. If there are M false detections in all test images and there are I images in the test set, then FPPI is defined as:

$$FPPI = \frac{M}{I} \tag{24}$$

Select a FPPI range (for example, 10^{-4}, 10^{0}), take the logarithm of the missed detection rate within this range and average it. The log-average miss rate (MR^{-2}) can be defined as:

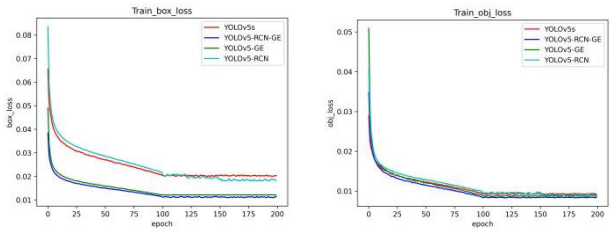
$$MR^{-2} = e^{\left(\frac{1}{|R|}\sum_{r\in R}\log(MR(r))\right)} \tag{25}$$

C. EVALUATION RESULT

The models before and after the improvement were trained for 200 epochs, and the Adam optimizer was used to iteratively update the network parameters with a weight attenuation rate of 0.0005. We set the initial learning rate to 1.25E–4, saved the optimal model based on the training results, trained a batch of 64 pictures, and used random cropping, flipping,

brightness transformation, and other operations to expand the dataset. For convenience, the previously designed ResNet-Conv&NonL module is referred to as RCN, and the loss function L_{G-E} is termed GE. Therefore, YOLOv5-RCN denotes the YOLOv5 model with the ResNet-Conv&NonL module, YOLOv5-GE represents the YOLOv5 model with the loss function, and YOLOv5-RCN-GE signifies the YOLOv5 model with both the ResNet-Conv&NonL module and the loss function L_{G-E}.

We obtained the comparative training results, as shown in Figure 11. As the number of iterations increased, the loss value steadily decreased and tended to stabilize. After approximately 150 epochs, the model's box_object_loss and mAP reached a plateau.



(a) Train box loss (b) Train obj loss

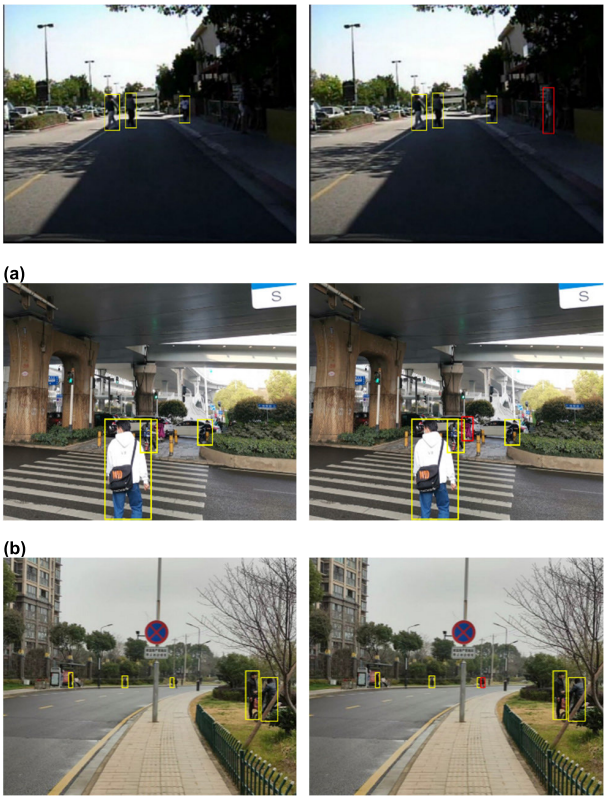
FIGURE 11. Model training loss comparison.

TABLE 1 presents a comparison between the improved and original algorithms in terms of mAP and frames per second (FPS), achieved by adjusting the backbone structure and loss function of YOLOv5 to enhance its performance on the dataset. According to the experimental results, the proposed improved method exhibits significant performance improvements. This improvement method is unique in that it can effectively improve model performance without increasing model complexity. This achievement was achieved through improvements in two key aspects. First, by adjusting the backbone structure of the model, we achieved a lightweight model while enhancing its ability to extract small-scale pedestrian features. Second, by optimizing the loss function, we increased the model's tolerance to small-scale pedestrian detection boxes, thereby improving its performance in the detection task.

TABLE 1. Comparison of test results.

Model	F1-score	mAP@.5	Frame-rate(fps)	Inference(ms)
YOLOv5s	0.815	0.857	35.34	28.3
Improved YOLO5s	0.876	0.917	51.02	19.6

As shown in Figure 12, a comparison of some detection results further verifies the superiority of our method. The proposed algorithm can effectively detect many small-scale, blurred and partially occluded pedestrians. What is even more encouraging is that when there are both large-scale and small-scale pedestrians in the scene, our algorithm can effectively distinguish them, showing good adaptability and robustness.



(a) Poor light (b) Inter-class occlusion(c) Intra-class occlusion

FIGURE 12. Comparison of detection results before (left) and after (right) improvement.

In summary, our experimental results strongly support the effectiveness of our improved method, which will has broad application prospects for small-scale pedestrian detection tasks. Our method not only improves detection performance but also keeps the model lightweight, making it suitable for various practical scenarios. This has an important practical significance for the field in small-scale pedestrian detection.

To further verify the effectiveness of the improved model proposed in this study, we conducted a series of experiments using the same experimental platform and dataset for comparison with mainstream model methods. The experimental results are presented in Table 2. This experimental design aims to evaluate the performance of our proposed algorithm compared with existing mainstream methods. By conducting a fair comparison on the same dataset, we were able to better understand the advantages and competitiveness of our method in the small-scale pedestrian detection tasks. This demonstrates the effectiveness of our method for a specific task and provides readers with an objective benchmark for performance evaluation. These experimental results further support the effectiveness of our improved method for small-scale pedestrian detection.

TABLE 2. mAP of different algorithms on caltech dataset.

Model	mAP@0.5
YOLOv5s	0.857
Faster R-CNN	0.745
SSD	0.721
YOLOv3	0.634
YOLOv4	0.750
YOLOv8[46]	0.887
FCOS-V2	0.802
Improved YOLOv5s	0.917

Fig. 13. shows the performance evaluation of different pedestrian detection models on the Caltech dataset, including YOLO series algorithms, SSD [13], Faster RCNN [12], SHDN [41], and APD [42]. Our proposed method outperforms all compared methods and achieves the lowest log-average miss rate of 42.85%. Compared to the original YOLOv5 model, the improved model we proposed reduces the log coverage miss rate by 18.1%.

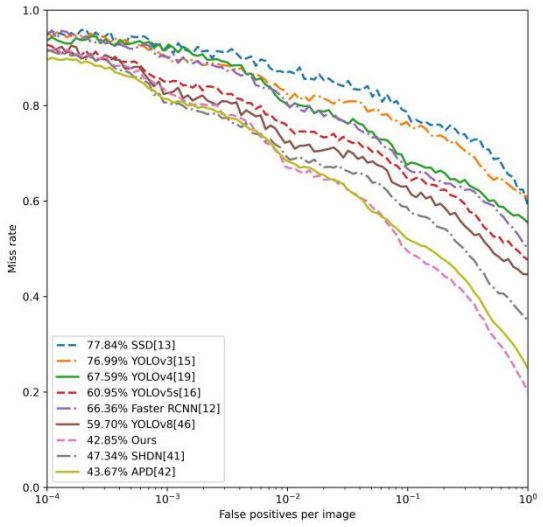


FIGURE 13. Quantitative comparison results of small-scale pedestrian (height ≤ 80 pixels) on caltech dataset.

D. ABLATION EXPERIMENT

To evaluate the effectiveness and advancement of the proposed algorithm, this study uses the same dataset to conduct five sets of ablation experiments to evaluate the impact of different improvement schemes on model detection performance. Under the same experimental conditions, the accuracy, recall, mAP, and model loss values of each model were used to evaluate the impact of the different modules on the YOLOv5 object detection algorithm. The lower the model loss value, the better is the regression of the model. The evaluation index results are listed in TABLE 2, and the mAP values are shown in Fig. 14.

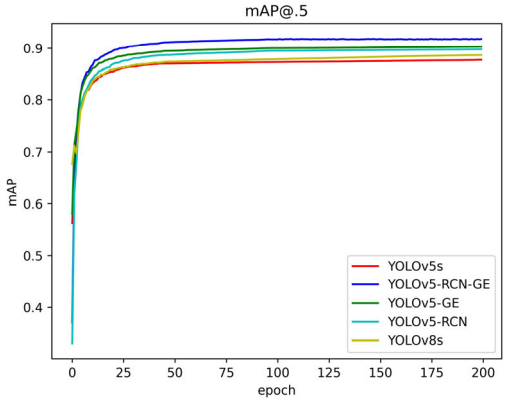


FIGURE 14. Comparison of mAP before and after model improvement.

According to the data in Table 3, compared with the YOLOv5s model, the mAP value of YOLOv8s increased by 3%; the use of the ResNet-Conv&NonL module increased the mAP value by 0.9%; the introduction of the LG-E loss function increased the mAP value by 4.5%; finally, by introducing the ResNet-Conv&NonL module and the LG-E loss function at the same time, the mAP value increased by 6%.

E. COMPARISON WITH METHODS ON CITYPERSONS DATASET

The CityPersons dataset is a high-quality dataset focused on pedestrian detection, derived from the popular urban scene understanding dataset, Cityscapes. It contains street-view images from several European cities with a rich diversity, including different weather conditions, urban environments, and pedestrian postures. This dataset provides accurate pedestrian bounding box annotations and was designed to support and promote the development of pedestrian detection technology. It is particularly suitable for researchers to test and improve pedestrian detection algorithms in complex urban environments.

According to the data in Table 4, compared with the YOLOv5s model, the mAP value of YOLOv8s increased by 2.5%; the use of the ResNet-Conv&NonL module increased the mAP value by 1.6%; the introduction of the LG-E loss function increased the mAP value by 3.9%; finally, by introducing the ResNet-Conv&NonL module and the LG-E loss function at the same time, the mAP value increased by 4.6%.

Fig. 15. shows the performance evaluation of different pedestrian detection models on the CityPersons dataset, including YOLO series algorithms, SSD [13], Faster RCNN [12], SHDN [41], and APD [42]. Our proposed method outperforms all compared methods and achieves the lowest log-average miss rate of 15.74%. Compared to the original YOLOv5 model, the improved model we proposed reduces the log coverage miss rate by 8.8%.

TABLE 3. Performance comparison of object detection model on caltech dataset.

Order	Model	RCN	GE	Precision (%)	Recall (%)	F1 (%)	mAP@.5 (%)
0	YOLOv5s	×	×	86.9	84.2	85.5	85.7
1	YOLOv8s	×	×	87.6	85.1	86.3	88.7
2	YOLOv5-RCN	√	×	90.1	84.7	87.3	89.6
3	YOLOv5-GE	×	√	92.2	86.1	89.0	90.2
4	YOLOv5-RCN-GE	√	√	94.8	90.2	92.4	91.7

TABLE 4. Performance comparison of object detection model on citypersons dataset.

Order	Model	RCN	GE	Precision (%)	Recall (%)	F1 (%)	mAP@.5 (%)
0	YOLOv5s	×	×	94.0	91.5	92.7	93.3
1	YOLOv8s	×	×	96.9	93.1	95.5	95.8
2	YOLOv5-RCN	√	×	95.6	92.8	94.2	94.9
3	YOLOv5-GE	×	√	97.9	94.3	96.1	97.2
4	YOLOv5-RCN-GE	√	√	98.8	94.6	96.7	97.9

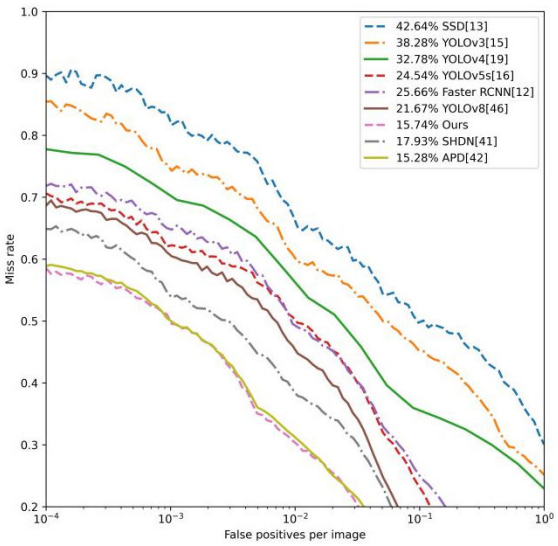


FIGURE 15. Quantitative comparison results of small-scale pedestrian (30 ≤ height ≤ 80 pixels) on citypersons dataset.

V. CONCLUSION

This paper addresses the problem of detecting small-scale pedestrians in unmanned driving and proposes the following improvements and innovations: (1) We select the YOLOv5 network structure, fuse the Non-Local and Convolution structures, and combine it with ResNet to construct the ResNet-Conv&NonL module, which effectively extracts the features of small-scale pedestrians. (2) A new model for measuring the similarity between target boxes is established and embedded into the loss function of YOLOv5 to better measure the similarity between the ground truth box and bounding box. Experiments were conducted using a home-made dataset to evaluate the performance of the algorithm. The results show that the proposed algorithm, compared with the original one, can significantly increase the detection accuracy while ensuring real-time detection and better robustness.

There is still a certain gap in practical applications, even though the algorithm proposed in this paper improves the accuracy of small-scale target pedestrian detection. In practical applications, the existence of complex environments, lighting conditions, and other factors often poses greater challenges to small-scale pedestrian detection. Thus, there is still an urgent need to further improve the accuracy and robustness of the algorithm and meet the needs of practical applications.

In addition, owing to hardware limitations, the dataset used in this study is insufficient. In future research, richer datasets can be considered for experiments to more comprehensively evaluate the performance and application prospects of the algorithm.

In the future, we plan to conduct in-depth research on neck network. In the existing algorithms, the neck network is a key link in feature fusion and has a significant impact on the performance of the algorithm. We will explore better feature fusion methods to improve the performance and robustness of the algorithm.

In addition, training with a small sample size is an important research direction. An insufficient data volume is frequently a concern in practical applications. Therefore, it is vital to find a solution to the problem of effectively the utilize limited data resources to improve the generalization ability of the model. In future, we will explore the use of small sample learning methods to enhance the generalization ability of the model and improve the performance of the algorithm in practical applications.

Finally, the algorithm proposed in this paper has some limitations, such as a limited processing ability for complex scenes. To meet the needs of practical applications, future research will strengthen the algorithm and explore more efficient and accurate small-scale pedestrian detection algorithms.

ACKNOWLEDGMENT

Hongchang Zhang thanks classmates in the laboratory for their continued support and the supervisor for his guidance and constant support rendered during this work. The authors thank the School of Automotive Engineering,

Wuhan University of Technology, for providing research facilities for completing this article.

REFERENCES

[1] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis, "SSH: Single stage headless face detector," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4885–4894.

[2] Y. Liu, Z. Ma, X. Liu, S. Ma, and K. Ren, "Privacy-preserving object detection for medical images with faster R-CNN," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 69–84, 2022.

[3] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster R-CNN doing well for pedestrian detection?" in *Proc. Eur. Conf. Comput. Vis.* (Lecture Notes in Computer Scienc), 2016, pp. 443–457.

[4] A. Raghunandan, P. Raghav, and H. V. R. Aradhya, "Object detection algorithms for video surveillance applications," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Melmaruvathur, India, Apr. 2018, pp. 0563–0568.

[5] M. Ju, J. Luo, P. Zhang, M. He, and H. Luo, "A simple and efficient network for small target detection," *IEEE Access*, vol. 7, pp. 85771–85781, 2019.

[6] M. L. Wang, X. Li, Q. Chen, L. B. Li, and Y. Y. Zhao, "CNN based surveillance video event detection," *J. Automatica Sinica*, vol. 42, no. 6, pp. 892–903, 2016.

[7] M. H. Wu, Y. X. Huang, and J. Wang, "Street pedestrian detection and tracking method based on improved YOLOv3," *Sci. Technol. Eng.*, vol. 21, no. 17, pp. 7230–7236, 2021.

[8] P. X. Yuan, X. Q. Ma, and S. Liu, "Improved YOLOv3 pedestrian and vehicle object detection algorithm," *Sci. Technol. Eng.*, vol. 21, no. 8, pp. 3192–3198, 2021.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[10] L. Steels and R. Brooks, *The Artificial Life Route to Artificial Intelligence: Building Embodied, Situated Agents*, 1st ed., London, U.K.: Psychology Press, 1995.

[11] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, pp. 21–37.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.

[15] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023.

[16] Ultralytics. (2022). *YOLOv5*. Github. Accessed: Jan. 12, 2023. [Online]. Available: https://github.com/ultralytics/yolov5/tree/v6.1

[17] C. Huang and Z. Jin, "Image retrieval based on combination of visual perception and local binary pattern histogram Fourier," *J. Compute-Aided Des. Graph.*, vol. 23, no. 3, pp. 406–412, 2011.

[18] M. Verma, B. Raman, and S. Murala, "Local extrema co-occurrence pattern for color and texture image retrieval," *Neurocomputing*, vol. 165, pp. 255–269, Oct. 2015.

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[20] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, "Augmentation for small object detection," 2019, *arXiv:1902.07296*.

[21] Z. He, H. Huang, Y. Wu, G. Huang, and W. Zhang, "Instance scale normalization for image understanding," 2019, *arXiv:1908.07323*.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[23] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling, "Clustered object detection in aerial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8310–8319.

[24] C. R. Ju, X. Y. Qin, and G. L. Guang, "Fast small object detection with scale sensitive loss and feature fusion," *Acta Electron. Sinica*, vol. 50, no. 9, pp. 2119–2126, 2022.

[25] C. Li, X. Y. Huang, and K. Wang, "Small target detection algorithm in high-resolution images based on feature fusion and self-learning anchor frame," *Acta Electron. Sinica*, vol. 50, no. 7, pp. 1684–1695, 2022.

[26] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," 2019, *arXiv:1906.05909*.

[27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–22.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, T. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.

[29] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[30] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Q. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3141–3149.

[31] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 658–666.

[32] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12993–13000.

[33] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and efficient IOU loss for accurate bounding box regression," *Neurocomputing*, vol. 506, pp. 146–157, Sep.28, 2022.

[34] G. Li, W. Zhao, and P. Liu, "A smooth IoU loss for object tracking bounding box regression," *IEEE/CAA J. Autom. Sinica*, vol. 49, no. 2, p. 19, Feb. 2023.

[35] Y. Shao, X. Zhang, H. Chu, X. Zhang, D. Zhang, and Y. Rao, "AIR-YOLOv3: Aerial infrared pedestrian detection via an improved YOLOv3 with network pruning," *Appl. Sci.*, vol. 12, no. 7, p. 3627, Apr. 2022.

[36] Y. Chen, M. L. Jin, and H. L. Liu, "Small-scale pedestrian detection based on feature enhancement module," *J. Electron. Inf. Technol.*, vol. 45, no. 4, p. 9, 2023.

[37] M. H. Xie, B. Kang, and H. F. Li, "Crowded pedestrian detection method combining anchor free and anchor base algorithms," *J. Electron. Inf. Technol.*, vol. 45, no. 5, pp. 1833–1841, 2023.

[38] Y. X. Chen, Y. Wen, H. L. Liu, B. Wang, and M. Y. Huang, "Multi-feature fusion pedestrian detection combining head and overall information," *J. Electron. Inf. Technol.*, vol. 44, no. 4, pp. 1453–1460, 2022.

[39] J. Chu, W. Shu, Z. B. Zhou, J. Miao, and L. Leng, "Combining semantics with multi-level feature fusion for pedestrian detection," *J. Automatica Sinica*, vol. 48, no. 1, pp. 282–291, 2022.

[40] W. Liu, S. Liao, W. Ren, W. Hu, and Y. Yu, "High-level semantic feature detection: A new perspective for pedestrian detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5182–5191.

[41] X. Zhang, S. Cao, and C. Chen, "Scale-aware hierarchical detection network for pedestrian detection," *IEEE Access*, vol. 8, pp. 94429–94439, 2020.

[42] X. Zhang, L. Cheng, B. Li, and H.-M. Hu, "Too far to see? Not really!—Pedestrian detection with scale-aware localization policy," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3703–3715, Aug. 2018.

[43] X. Zhang, L. Li, H. Liu, P. Yang, and Y. Gao, "Disentangling classification and regression in Siamese-based network for visual tracking," *Concurrency Comput., Pract. Exper.*, vol. 34, no. 27, p. e7246, Dec. 2022.

[44] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 304–311.

[45] S. Zhang, R. Benenson, and B. Schiele, "CityPersons: A diverse dataset for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4457–4465.
[46] Ultralytics. (2023). *YOLOv8 [Software]*. [Online]. Available: https://github.com/ultralytics/yolov8

HONGCHANG ZHANG was born in Shandong, China, in 1980. He received the master's degree from the School of Automotive Engineering, Wuhan University of Technology, in 2006, and the Ph.D. degree from the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, in 2012.

Since 2015, he has been an Associate Professor with the School of Automotive Engineering, Wuhan University of Technology. He has over 20 academic papers and more than 30 invention patents. His main research directions are key technologies for intelligent connected vehicles and automotive electronic control design technology.

KANG YANG was born in Jiangsu, China, in 1999. He received the B.S. degree from the School of Automotive and Traffic Engineering, Jiangsu University, Jiangsu, China, in 2021. He is currently pursuing the M.S. degree with the School of Automotive Engineering, Wuhan University of Technology, Wuhan, China.

His research interests include machine learning, target detection, computer vision, and autonomous vehicle.

HENG LIU was born in Shandong, China, in 1998. He received the bachelor's degree from the School of Automotive and Transportation, Tianjin Vocational and Technical Normal University, in 2020, and the master's degree from the School of Automotive Engineering, Wuhan University of Technology, in 2023.

Since 2023, he has been with the China Heavy Duty Truck Group, with his main research focus on key technologies for intelligent connected vehicles.

JIALI HU was born in Jiangsu, China, in 1998. She received the B.S. degree from the School of Automotive and Traffic Engineering, Jiangsu University, Jiangsu, in 2022. She is currently pursuing the M.S. degree with the School of Automotive Engineering, Wuhan University of Technology, Wuhan, China.

Her research interests include machine learning, deep learning, target detection, computer vision, and autonomous vehicle.

YAO SHU was born in Hubei, China, in 1999. He received the bachelor's degree from the School of Food Science and Technology, Huazhong Agricultural University, Hubei, in 2021. He is currently pursuing the master's degree in automotive engineering with Wuhan University of Technology, China.

His research interests include machine learning, object detection, computer vision, and autonomous vehicle.

JUAN ZENG was born in Hubei, China, in 1973. She received the master's and Ph.D. degrees from the School of Management, Wuhan University of Technology, in 2001 and 2008, respectively.

She is currently an Associate Professor with the School of Automotive Engineering, Wuhan University of Technology. She has over 30 academic papers and over 40 invention patents. She is the author of four books. Her main research directions are key technologies of intelligent connected vehicles, automotive safety, motor vehicle risk management, and driver behavior research.