Copyright ⓒ 2022 KSII

A Lightweight Pedestrian Intrusion Detection and Warning Method for Intelligent Traffic Security

Xinyun Yan[1,2], Zhengran He[3,*], Youxiang Huang[3], Xiaohu Xu[3], Jie Wang[4], Xiaofeng Zhou[1], Chishe Wang[4], and Zhiyi Lu[5]

[1] College of Computer and Information, Hohai University, Nanjing, 211100, China
[2] Engineering School of Networks and Telecommunications, Jinling Institute of Technology, Nanjing, 211169, China
[3]College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
[4]Engineering School of Networks and Telecommunications, Jinling Institute of Technology, Nanjing, 211169, China
[5]AI Innovation Center, Nanjing Great Information Co., Ltd., Nanjing, 210046, China
[e-mail : 1320017313@njupt.edu.cn]
* Corresponding Author: Zhengran He

Received October 25, 2022; revised November 18, 2022; accepted December 5, 2022; published December 31, 2022

Abstract

As a research hotspot, pedestrian detection has a wide range of applications in the field of computer vision in recent years. However, current pedestrian detection methods have problems such as insufficient detection accuracy and large models that are not suitable for large-scale deployment. In view of these problems mentioned above, a lightweight pedestrian detection and early warning method using a new model called you only look once (Yolov5) is proposed in this paper, which utilizing advantages of Yolov5s model to achieve accurate and fast pedestrian recognition. In addition, this paper also optimizes the loss function of the batch normalization (BN) layer. After sparsification, pruning and fine-tuning, got a lot of optimization, the size of the model on the edge of the computing power is lower equipment can be deployed. Finally, from the experimental data presented in this paper, under the training of the road pedestrian dataset that we collected and processed independently, the Yolov5s model has certain advantages in terms of precision and other indicators compared with traditional single shot multiBox detector (SSD) model and fast region-convolutional neural network (Fast R-CNN) model.

After pruning and lightweight, the size of training model is greatly reduced without a significant reduction in accuracy, and the final precision reaches 87%, while the model size is reduced to 7,723 KB.

Keywords: Traffic safety, pedestrian detection, computer vision, YOLOv5, lightweight.

1. Introduction

The fast development of wireless communications and internet of things illuminates many intelligent applications for secure and convenient daily life [1-5]. There is a lot of space for pedestrian detection in all fields [6-9]. For example, in the autonomous areas, accurate and timely target detection can make autonomous driving technology a reality [10-13]. From the perspective of traffic safety, pedestrian detection can be used to avoid traffic accidents caused by pedestrian intrusion [14-16]. In the aspect of safety protection, pedestrian detection and face recognition can control the entry and exit of suspicious persons [17]. According to the management of public places, pedestrian detection can also be used to count people flow data, thereby optimizing the allocation of human and material resources. Therefore, the necessity of pedestrian detection is increasing.

Pedestrian detection is a branch of target detection. The main detection idea is consistent with the traditional target detection method, and it includes three stages. First generate the target region proposal, then extract the features in each proposal, and finally classify according to the features. However, considering the prediction accuracy and speed of the prediction, traditional detection algorithms have been unable to meet today's various needs. In the field of computer vision and signal processing, deep learning has high accuracy and fast training speed, which makes it gradually become the mainstream of detection methods [18-26]. At this stage, most pedestrian detection algorithms with high performance are basically trained by deep learning. The method of extracting image features using depth learning model for the first time is region convolution neural network (R-CNN), with a detection accuracy of 49.6% [27]. The problem of computational redundancy still exists, so a fast R-CNN emerges as the times require. It introduces a simplified spatial pyramid pooling (SPP) layer for merging model training and testing processes [28]. Later, Faster R-CNN proposes the Regional Proposal Network (RPN), which is utilized to achieve targeted regional proposals [33]. When original image of any pixel is input, a batch of rectangular regions can be output, and each rectangular region corresponds to the coordinate information and confidence of a target. In summary, from the earliest R-CNN model to the Faster model, deep learning method integrates the three stages of traditional target detection into one network. This is also the advantage of deep learning algorithms over traditional detection methods [31-33].

With the advent of detection models based on regression algorithms, the field of object detection has reached a new level. Among them, YOLO and SSD methods, which are one-stage detection, can make real-time detection possible [34-38]. However, there are still problems with the above training method detection model using deep learning need to be solved urgently. First of all, in terms of accuracy, the current pedestrian detection accuracy is not high, and it cannot be applied to the field of road safety detection with low detection accuracy tolerance. In addition, the training models of these methods are also memory-intensive and not suitable for widespread deployment in edge devices. Aiming at the above problems, a lightweight pedestrian detection and early warning method based on YOLOv5s is proposed. After training the detection model optimized by sparsification and pruning, significantly reduces the memory consumption of our model, to meet the requirements of the edge equipment deployment.

We summarize the main contributions of this paper into the following three points.

1) To solve the problem of the less practical dataset of pedestrian on the road, we generate an autonomously labeled road pedestrian dataset for pedestrian intrusion detection.

2) Based on the generated dataset, we propose a road pedestrian intrusion detection method using the Yolov5s model. The detailed algorithm is stated to show the proposed detection method.
3) Aiming at the problem that model training occupies a large amount of memory and is difficult to deploy widely, a lightweight model compression method via batch normalization (BN) layer pruning is proposed, which significantly reduces the size of the Yolov5 model and facilitates edge device deployment.

2. Related Works

Here we draw a figure to show the development of target detection algorithm, as shown in **Fig. 1**. Taking 2012 as the dividing line, before it, it is based on the traditional detection algorithm, while after it, it is the deep learning detection algorithm.

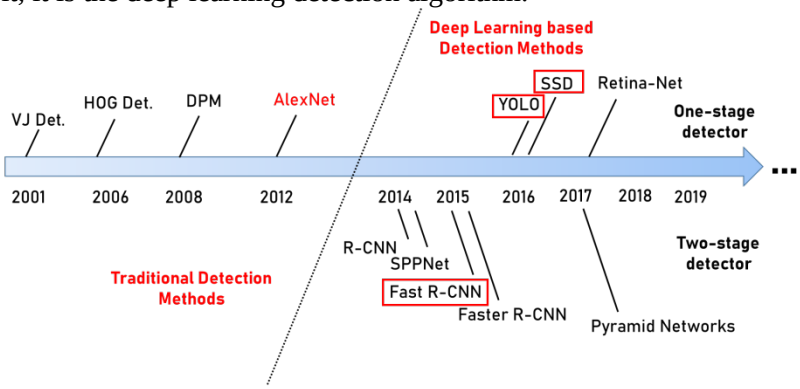


Fig. 1. The development of human detection methods.

In 2001, P. Viola and M. Jones proposed the VJ detector and realized face detection for the first time. This detector was a milestone algorithm in the development of face detection technology [39,40]. The algorithm mainly used sliding window detection, image fusion, feature selection and detection cascade technology, these new technologies greatly improved the detection speed. In 2005, N. Dalal proposed a directional gradient histogram (HOG) as a feature descriptor [41]. And this HOG detector has been an important foundation for computer vision applications for many years. In 2008, P. Felzenszwalb *et al.* [42] proposed deformable part model (DPM) on the basis of HOG detector, and won the championship of VOC 07, 08 and 09 for three years, reaching the peak of traditional detectors at that time. After that, R. Girshick also made various improvements to the method [43,44]. The AlexNet algorithm was born in 2012, and since then, deep learning methods were gradually emerging, and convolutional neural network was the most representative one [45]. Since deep convolutional networks could learn robust and high-level feature representations for images, they were also extremely used in target detection algorithms.

R. Girshick *et al.* first proposed region with CNN (RCNN) for object detection in 2014 [27]. After that, objection detection developed rapidly. After the rise of deep learning, target detection is divided into two categories. One is *one-stage detection*, the other is *two-stage detection*. *Two-stage detection* constructs detection as a process of coarse to fine, while *one-stage detection* achieves detection as a process of complete in one step [46]. In 2014, K. He [47] proposed a method called proposed spatial pyramid pool (SPPNet). SPPNet was 20 times faster than R-CNN without loss of accuracy, but it still suffered from the disadvantage of multi-stage training. In 2015, R Girshick further improved R-CNN and SPPNet, thus proposed

a fast R-CNN detector [28]. Shortly thereafter, in 2015, S. Ren proposed an end-to-end real-time depth learning detector, which was called Faster RCNN detector [48]. In 2017, feature pyramid network (FPN) was proposed on the basis of faster RCNN [52]. FPN was a top-down structure with side chains, which can effectively detect targets of various scales.

In one-stage detection, R. Joseph proposed YOLO in 2015 [50]. This algorithm was the first one-stage detector after the rise of deep learning, and it divided the image into different regions according to the algorithm, and used neural networks to predict the prediction box and class probability of each region at the same time. In 2015, W. Liu proposed the Single Shot Multi Box Detector (SSD) [51]. It adopted multi reference and multi-resolution detection technologies, which could greatly improve the final accuracy of small detection targets. Finally in 2017, T. Y. Lin *et al.* proposed RetinaNet, which introduced a new loss function Focal Loss in RetinaNet [52]. By changing the standard cross-entropy loss, the detector can focus on difficult and misclassified data during training.

3. System Model and Problem Formulation

3.1 Datasets

At the current stage, the dataset of pedestrian detection is not comprehensive and specific, so in this paper, we use our own labeled road camera dataset to detect the performance of the algorithm. We have collected a large number of road camera data sets and used labelimg software to label different targets. Our annotated dataset includes five types of labels, namely pedestrians, construction workers, roadblocks, fences, and others. Others represent objects that are easily confused with pedestrian features at different angles. Aiming at achieving more accurate algorithm recognition, we also mark it. The specific data categories and tag numbers can be found in **Table 1**.

Table 1. Pedestrian detection label data

Labels	Number of Tags
Construction workers	3,498
Pedestrian	1,389
Roadblock	2,422
Fence	610
Others	262

3.2 Pedestrian Detection Model

The following **Fig. 2** is the detection process of the pedestrian warning method. As shown in this figure, images captured by the camera along the road will be processed by image transmission. After preprocessing, the high-level image features can be extracted using deep neural networks. Finally, the desired pedestrian or object can be detected on the original image by visualization.

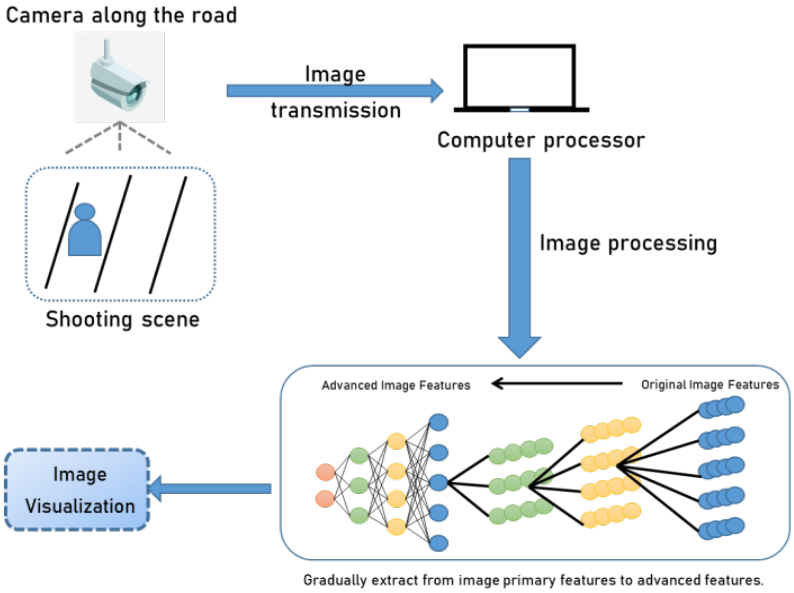


Fig. 2. The flow of pedestrian detection method.

To sum up, the core step in this early warning method is to use deep learning to extract advanced image features, which is the pedestrian detection algorithm. Pedestrian detection algorithms are mainly divided into two categories, as mentioned above, the Two-stage algorithm first generates a pre-selection box that may contain objects to be detected, that is, region proposal, and then the convolution neural network is used for classification. While the One-step detection algorithm is different from it. Instead of using region proposal, it first uses the network to extract features and then directly uses them to predict the classification and location of objects.

The main Two-stage detection algorithms include R-CNN, Fast R-CNN, etc., while One-stage algorithms include YOLO, SSD, and RetinaNet. This paper mainly compares Fast R-CNN, YOLOv5s6 and SSD algorithms, and analyzes and compares the differences in the performance of their algorithm models in the field of pedestrian detection.

3.2.1 Fast R-CNN

We take more classic algorithm fast R-CNN in Two-stage as an example to introduce the network features and algorithms. The following **Fig. 3** shows its algorithm network structure. Fast R-CNN uses a new method to extract candidate frames, namely selective search. After searching, CNN is used for feature extraction [28]. However, unlike R-CNN, Fast R-CNN uses the Region of Interest (RoI) pool layer to replace the support vector machine (SVM), and uses this layer to extract corresponding features of each RoI on the complete image features. The RoI pool layer will calculate the specific position, shape and size of the output feature map corresponding to each RoI according to the position and size of the original image corresponding to the previous RoI, and finally unify them. Continue to pass through two fully connected layers to obtain features, then through the new full connection layer, and finally connect respective loss functions to obtain classification and bounding boxes.

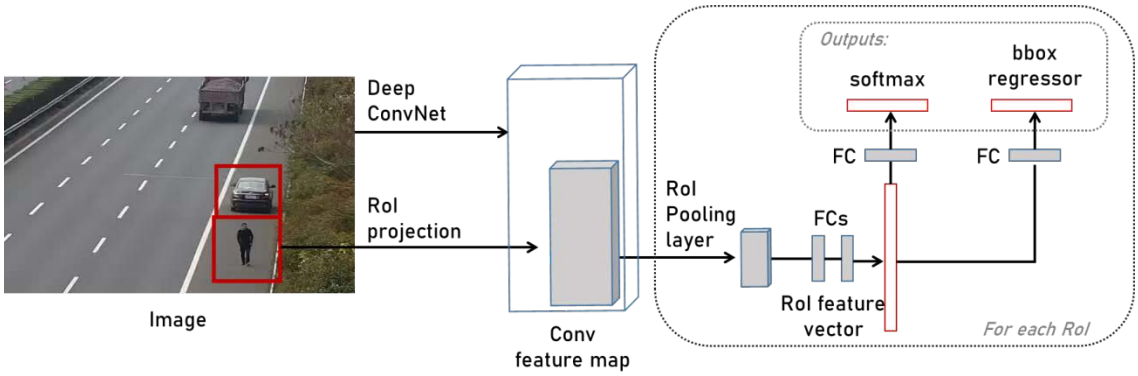


Fig. 3. Network structure of Fast R-CNN algorithm.

For the bounding boxes, using bounding box regression to correct the target bounding box, we use the smooth L1-loss paradigm as the loss function, as shown in (1) and (2). While the loss function for classification is the softmax function, as shown in (3).

$$\mathrm{L}_{loc}(t, t_*) = \sum_{i \in x,y,w,h} smooth\ L_1(t_i, t_i^*), \tag{1}$$

$$smooth\ L_1(X) = \begin{cases} 0.5X^2, & |X| \le 1 \\ |X| - 0.5, & |X| > 1' \end{cases} \tag{2}$$

$$\mathrm{L}_{cls} = -\log(p_u)\,, \tag{3}$$

where x, y, w, h are coordinates of region, t_i represents predicted value, and t_i^* represents the ground truth coordinates. The X in smooth $L_1(X)$ is $t_i - t_i^*$, which is the difference between the corresponding coordinates. In the loss function of classification, $p = (p_0, p_1, \ldots, p_k)$ is the classification probability prediction for each ROI area (K is the total number of categories), while p_u represents the probability that u belongs to the correct category.

3.2.2 YOLOv5s

In the One-stage algorithm, we mainly introduce the algorithm network structure of YOLOv5s, as shown in the following **Fig. 4**. As shown in the figure, the YOLOv5s network mainly includes four parts, namely Input part, Backbone part, Neck part and finally Head part.

In input part, the training of YOLOv5 model uses Mosaic data enhancement method. The method uses four images that are scaled, cropped, and aligned to maximize the dataset, improve the training speed, and finally reduce the size of the model. In addition, this part can also perform adaptive image scaling. Firstly, the scaling ratio is calculated according to the size of the original image and the size of the input image, and then the size of the scaled image is calculated. Finally, the size of the black boundary generated at both ends of the scaled image is calculated and filled, which greatly improves the reasoning speed and versatility of the algorithm.

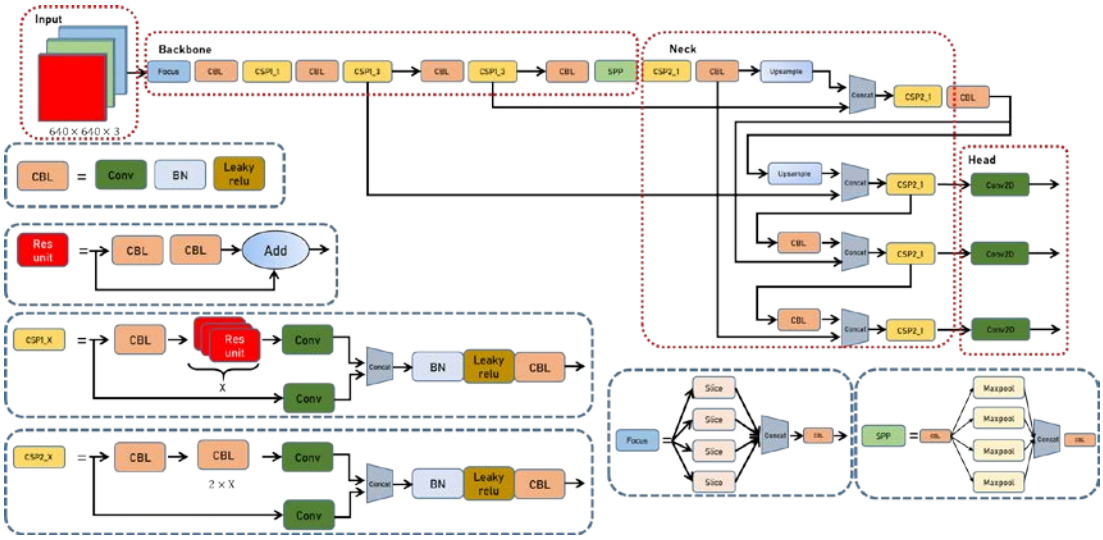
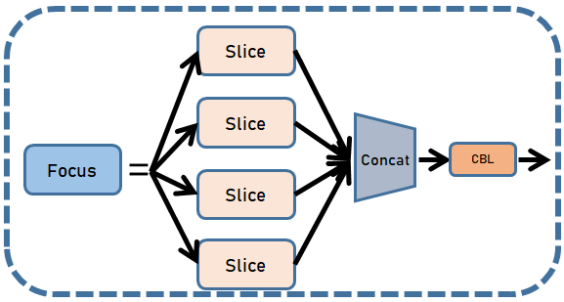
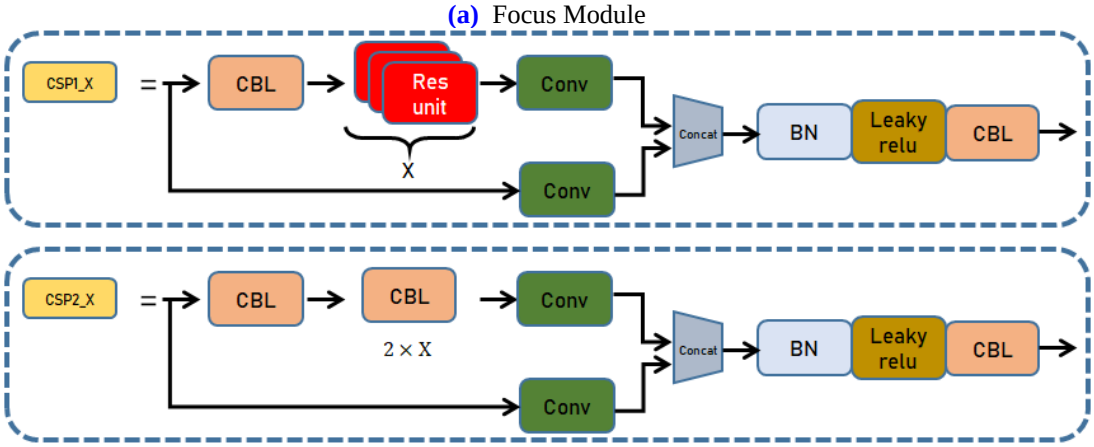


Fig. 4. Network structure of YOLOv5s algorithm.

As shown in **Fig. 5**, in the backbone part, the YOLOv5 network mainly adopts the focus structure and the CSP structure, as shown in the **Fig. 5(a)** and **Fig. 5(b)**. The Focus structure mainly clips the input image by slicing operation. Suppose the size of the original input image is $2N \times 2N \times 3$. After we do the Slice and Concat operations, a $N \times N \times 12$ feature map is output; then a 32-channel convolutional (Conv) layer, the number of channels is similar to YOLOv5s structure, but other structures will be changed accordingly, and the final output is $N \times N \times 32$ feature maps.



(a) Focus Module



(b) CSP Module

Fig. 5. The structure of backbone part.

The CSP structure in YOLOv5 is divided into two types: CSP1_X for backbone and CSP2_X for neck. The difference is the use of a different middle layer. One uses the residual network and the other uses Conv+BN+Leakyrelu (CBL), as shown in the figure. CSP module can be the characteristics of the base layer is divided into two parts, first stage again by across hierarchy combine two parts. On the one hand, the calculation amount is reduced, on the other hand, the accuracy is also guaranteed.

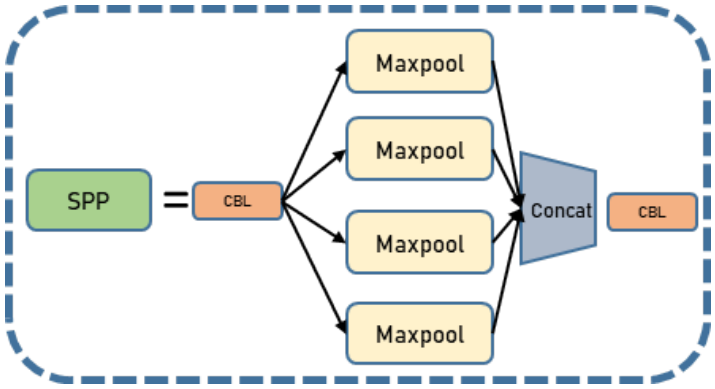


Fig. 6. The structure of SPP module.

In the Neck part, YOLOv5 mainly uses the SPP module and the feature pyramid network (FPN) and path aggregation network (PAN) structure [29]. SPP module obtains a robust feature representation by fusing different-sized max-pooling layers, as shown in **Fig. 6**.

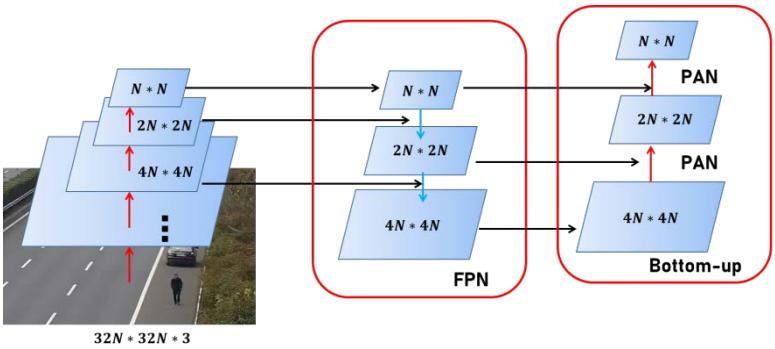


Fig. 7. The structure of FPN and PAN.

In **Fig. 7**, FPN can better solve the problem of mesoscale target detection by building a pyramid on the feature map. While PAN is a bottom-up structure, and two PAN structures are added to FPN, $N \times N$, $2N \times 2N$ and $4N \times 4N$ are the three feature maps for the final prediction. Different from the convolution operation in YOLOv4, the CSP2 structure was used to enhance the feature fusion ability of YOLOv5 network.

The last part is the Head, whose main function is to predict image features, generate classified objects and bounding boxes. In general, the loss function for object detection is a combination of a classification loss and a regression loss. The regression loss function used by YOLOv5s is generalized intersection over union (GIoU). Assuming that A is the ground truth box and B is the prediction box, the calculation formula of IoU is given as

$$\mathrm{IoU} = \frac{|\mathrm{A} \cap \mathrm{B}|}{|\mathrm{A} \cup \mathrm{B}|}, \tag{4}$$

for A and B to find the smallest closed convex object C. Then, the calculation formula of GIoU can be obtained as,

$$\mathrm{GIoUloss} = 1 - \mathrm{IoU} + \frac{\left|\frac{\mathrm{C}}{\mathrm{A}} \cup \mathrm{B}\right|}{|\mathrm{C}|}. \tag{5}$$

3.2.3 Model Pruning

Although the YOLOv5s network is the most lightweight detection network among various YOLOv5 networks, in the context of pedestrian detection in this paper, it still has a large model size, which is not convenient for the deployment of edge devices. Simply reducing the input dimension, such as changing the input 640×640 dimension to 320×320 dimension, will seriously affect the performance and cause a great loss of recognition accuracy. Network pruning can significantly reduce the size of the algorithm model without modifying the network input, thus meeting the deployment requirements of edge devices.

This paper refers to the literature [53-54], and constrains the BN layer coefficients by adding L1 regularity to make the coefficients sparse. After sparse training, pruning is performed. Since the layers with small sparse results have small corresponding activation functions, pruning these layers has little effect on the subsequent layers. By iterating this process, and finally a very compact model can be obtained.

The calculation of the BN layer is given as

$$\begin{aligned} \hat{z} &= \frac{z_{in} - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \\ z_{out} &= \gamma^{\hat{z}} + \beta. \end{aligned} \tag{6}$$

The input and output of BN layer are z_{in} and z_{out} respectively, B is the current mini batch. μ_B represents the mean value of input activation quantity, and σ_B represents its standard deviation, γ and β represent scale and displacement respectively, which are affine transformation parameters that can be trained.

Since the z_{out} of each batch is positively related to the coefficient γ, γ can be pruned, and removing the value of γ close to 0 will not affect the size of the final activation value. However, since the BN layer coefficients are normally distributed, γ tends to 0 in very few cases, which cannot be effectively pruned. Therefore, L1 regularity constraints can be added to make pruning feasible when γ is equal to 0. The L1 regularity constraint is expressed as

$$\mathrm{L} = \sum_{(x,y)} l(f(x, W), y) + \lambda \sum_{\gamma \in \Gamma} g(\gamma), \tag{7}$$

where (x, y) represents the input and target of the training, respectively, W represents the training weight, the first term represents the loss function of normal training. $g(\cdot)$ is the sparsity penalty for the scale factor, λ is the regularization coefficient, and Γ represents the range of the coefficient γ. Using the L1 regularity constraint, we can also perform sparse training on parameters first, that is, change the loss function in backpropagation as follows:

$$\begin{aligned} \mathrm{L}' &= \sum l' + \lambda \sum g'(\gamma) \\ &= \sum l' + \lambda \sum |\gamma|' \\ &= \sum l' + \lambda \sum \gamma sign(\gamma), \end{aligned} \tag{8}$$

hence, one can find that L1 regularization can balance the value of the BN's scaling factor towards zero, enabling the identification of unimportant channels (or neurons) and facilitating channel-level pruning in the next steps. After sparsification and network pruning, the resulting

network is smaller than the initial network in model size, and better in memory footprint and computation operation at runtime. The above process is repeated several times to obtain a simplified multi-channel network scheme, which makes the network more compact.

Algorithm 1: The proposed model training method.

Input : The original image after preprocessing, the dimension is $\boldsymbol{I} = (32N \times 32N \times 3)$; the ground truth B^g bounding box coordinate, $B^g = (x_1^g, y_1^g, x_2^g, y_2^g)$.

Output: Object label and three prediction feature maps with dimensions $(N \times N \times 33)$, $(2N \times 2N \times 33)$, $(4N \times 4N \times 33)$.

[Model Training]:
Randomly initialize the YOLOv5s network.
for $i = 0, 1, \cdots, M$ **do**
  $\boldsymbol{I} \xrightarrow{Slice} \boldsymbol{I}_1 = (16N \times 16N \times 12)$;
  $\boldsymbol{I}_1 \xrightarrow{\text{Conv. layer (CBL)}} \boldsymbol{I}_2 = (16N \times 16N \times H)$, here H represents the number of convolutional layer channels;
  The CSP and CBL modules are repeatedly calculated, and finally three prediction feature maps and prediction $B^p = (x_1^p, y_1^p, x_2^p, y_2^p)$ are obtained through the FAN and PAN module;
  For the predicted box B^p, ensuring $x_2^p > x_1^p$ and $y_2^p > y_1^p$:
  $\hat{x}_1^p = \min(x_1^p, x_2^p), \hat{x}_2^p = \max(x_1^p, x_2^p)$,
  $\hat{y}_1^p = \min(y_1^p, y_2^p), \hat{y}_2^p = \max(y_1^p, y_2^p)$.
  Get the area of B^g : $Aera^g = (x_2^g - x_1^g) \times (y_2^g - y_1^g)$.
  Get the area of B^p : $Aera^p = (x_2^p - x_1^p) \times (y_2^p - y_1^p)$.
  Calculating intersection I between B^p and B^g:
  $x_1^I = \max(\hat{x}_1^p, x_1^g), x_2^I = \min(\hat{x}_2^p, x_2^g)$,
  $y_1^I = \max(\hat{y}_1^p, y_1^g), y_2^I = \min(\hat{y}_2^p, y_2^g)$.
  $I = \begin{cases} (x_2^I - x_1^I) \times (y_2^I - y_1^I), & \text{if } x_2^I > x_1^I, y_2^I > y_1^I \\ 0, & \text{othewise.} \end{cases}$
  Search coordinate of smallest enclosing box B^c:
  $x_1^c = \min(\hat{x}_1^p, x_1^g), x_2^c = \max(\hat{x}_2^p, x_2^g)$,
  $y_1^c = \min(\hat{y}_1^p, y_1^g), y_2^c = \max(\hat{y}_2^p, y_2^g)$.
  Get the area of B^c : $Area^c = (x_2^c - x_1^c) \times (y_2^c - y_1^c)$.
  $IoU = I/U$, where $U = Area^p + Area^g - I$.
  $GIoU = IoU - Area^c - U/Area^c$.
  $L_{IoU} = 1 - IoU, \quad L_{GIoU} = 1 - GIoU$.
  Calculate IoU and GIoU loss functions, reverse update, iterate network parameters. Output three types of predicted feature maps of different sizes according to the size of the object to be tested.
end
[Model Testing]:
for $e = 1, 2, \cdots, E$ **do**
  Save the model according to the algorithm in the training phase, verify the performance of the model on the test set, and finally output the classification status and prediction box.
end

3.3 Model Training Algorithm

The training and testing process of the model is shown in **Algorithm 1**. As shown in the algorithm flow chart, we adopt the overall framework of the algorithm of YOLOv5s model, and start training by extracting the characteristics of the target to be measured in the picture and the location box of the target to be measured. The input is the original image preprocessed in the training set and the labeled detection box. After dimensional transformation, the size of

the picture entering the YOLOv5 network is $(32N \times 32N \times 3)$, and N is 20 in this paper. In response to objects to be detected of different sizes, the final output of the model is three prediction frame sizes, $N \times N \times 33$, $2N \times 2N \times 33$, and $4N \times 4N \times 33$ respectively. These Heads of different scales are used to detect objects of different sizes, of which 33 means (6 categories to be tested + 1 probability + 4 prediction box coordinates) × 3 anchor boxes = 33.

During the training process, as shown in the flowchart in **Fig. 4**, after the input image dimension passes through the Backbone and neck modules in the YOLOv5s network, the label of the object to be tested and the size of the predicted box are detected according to regression and clustering. Relying on the formula in the algorithm flow chart, the final GIoU loss function can be calculated by the predicted prediction frame B^p and the detection frame B^g in the actual real scene.

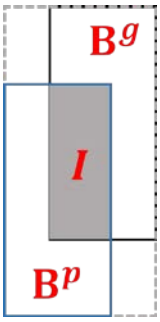


Fig. 8. Schematic diagram of GIoU calculation.

The calculation formula of GIoU is shown in formula (4)-(5), and the schematic diagram of the image is shown in **Fig. 8**. The blue box represents the position and size of the prediction box, and the black box is the detection box in the real scene. I represents the area of the shaded part where the two overlap on the way, and U is the total area of the two minus the area of the shaded part. The last $Area^c$ is the minimum external area of the two rectangles, which is the total area of the dotted box shown in the figure.

4. Experimental Results

The YOLOv5s algorithm model used in this paper and the simulation of subsequent pruning and comparison algorithms are based on pytorch. It does this with a GeForce GTX 3090. We split the dataset into training set, validation set, and test set in a ratio of 8:1:1. The following simulation results are the results of the test set.

4.1 Comparison of Different Algorithms

The more commonly used pedestrian detection algorithms at this stage are mostly deep learning algorithms. As described in related works, they are divided into one-stage method and two-stage method. The detection algorithm we use in this paper is the YOLOv5s algorithm model, which is compared with the SSD algorithm model of the one-stage method and the Fast R-CNN algorithm model of the two-stage method.

Fig. 9 shows the training results using the YOLOv5s model. As shown in the figure, it can be seen that when 300 epochs are selected, the final loss curves of the training set and the validation set have tended to converge. And the evaluation indicators such as precision, recall and mean average precision (mAP) can be considered to reach the optimal value.

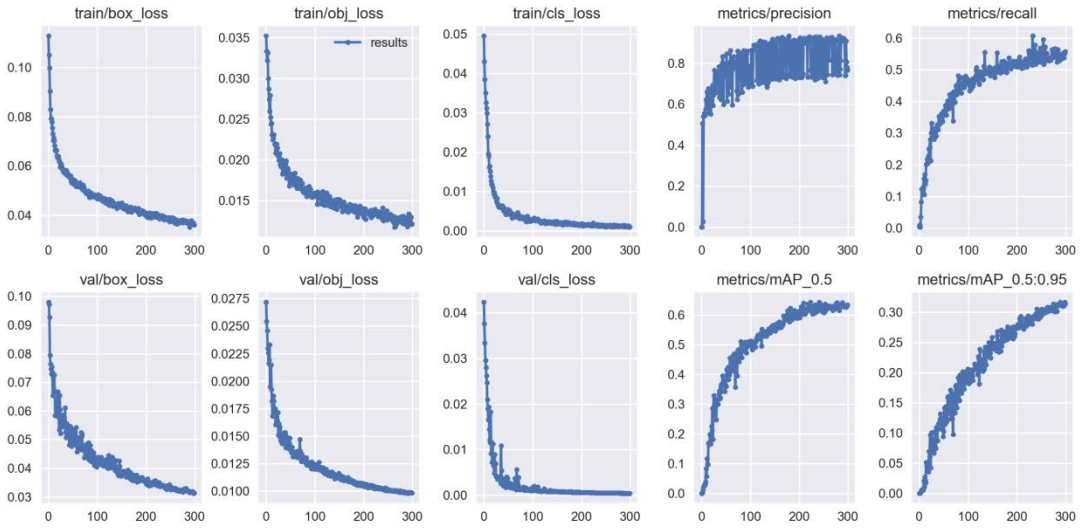
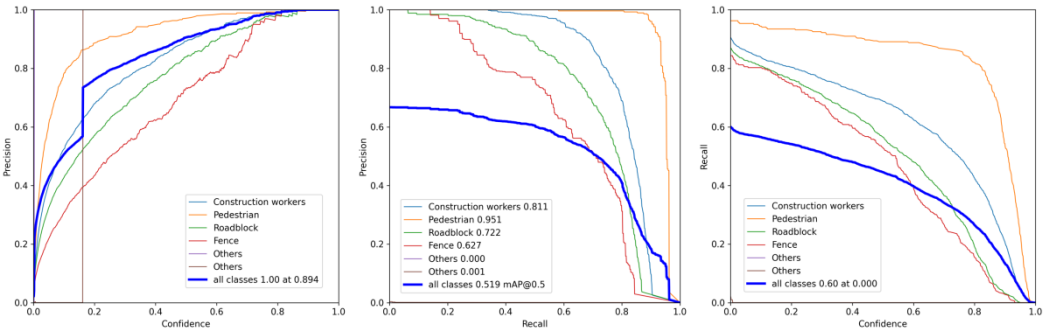


Fig. 9. The training results of pedestrian detection method.

Fig. 10 shows the evaluation indicators of the YOLOv5s network under 300 epochs training. (The label other has two curves. Because there is less data, and the final recognition effect task is irrelevant, it will not be discussed) Among them, Precision represents the fraction of predicted positive samples that are actually positive, and recall refers to all positive samples that are actually predicted as positive. The size of the area under the PR curve, to a certain extent, characterizes the relative "double high" ratio of the algorithm in terms of precision and recall. mAP represents the area under the PR curve. The better the classifier used, the higher the AP value.



(a) P curve **(b) PR curve** **(c) R curve**

Fig. 10. The training results of pedestrian detection method.

The calculation of each indicator is as follows.

$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FP} \tag{9}$$
$$AP = \frac{1}{N}\sum_{i=1}^{N} P_i$$

$$mAP = \frac{1}{K}\sum_{i=1}^{K} AP_i,$$

here TP is a true example, TN is a true negative example, FP is a false positive example, and FN is a false negative example. And N represents the number of dataset labels for pedestrian detection (6 in this paper), and K represents the number of IoUs for the threshold.

Table 2. Comparison of test results of three algorithms

Algorithms	Precision	Recall	F1score	mAP/0.5
YOLOv5s6	82.02%	84.78%	56.00%	88.65%
SSD	72.73%	16.33%	27.00%	42.61%
Fast R-CNN	21.22%	26.36%	24.00%	5.48%

Table 2 shows the performance comparison of YOLOv5s algorithm, SSD algorithm and Fast R-CNN algorithm using the same dataset and training parameters. It can be seen that the Fast R-CNN algorithm using two-stage detection has poor performance, while SSD and YOLOv5 using single-stage detection can achieve better recognition accuracy. Among them, YOLOv5 has the best performance.

4.2 Performance Comparision and Final Detection Result

Although the YOLOv5s model outperforms the other three algorithms in model size, aftqer training with the pedestrian detection dataset marked by itself, the size of the final trained network model also reaches 55 M. It is not suitable for deployment to edge devices, so we use the BN layer pruning method in system model to prune the trained model. As the process in the algorithm flow chart, we first conduct sparse training for the best model trained by YOLOv5s, then perform L1 regular optimization on the BN layer, and cut 60% of the network structure when γ is close to 0 according to the formula (7), and finally fine-tune the network. The performance results after performing the various steps are shown in **Table 3**.

Table 3. Algorithm comparison of performance during pruning

Models	Images	Labels	P (%)	R (%)	mAP@0.5 (%)
YOLOv5s	1020	3000	0.820	0.848	0.887
YOLOv5s-sparsification	1020	3000	0.824	0.788	0.843
YOLOv5s-prune	1020	3000	0.828	0.789	0.844
YOLOv5s-finetune_prune	1020	3000	0.874	0.756	0.834

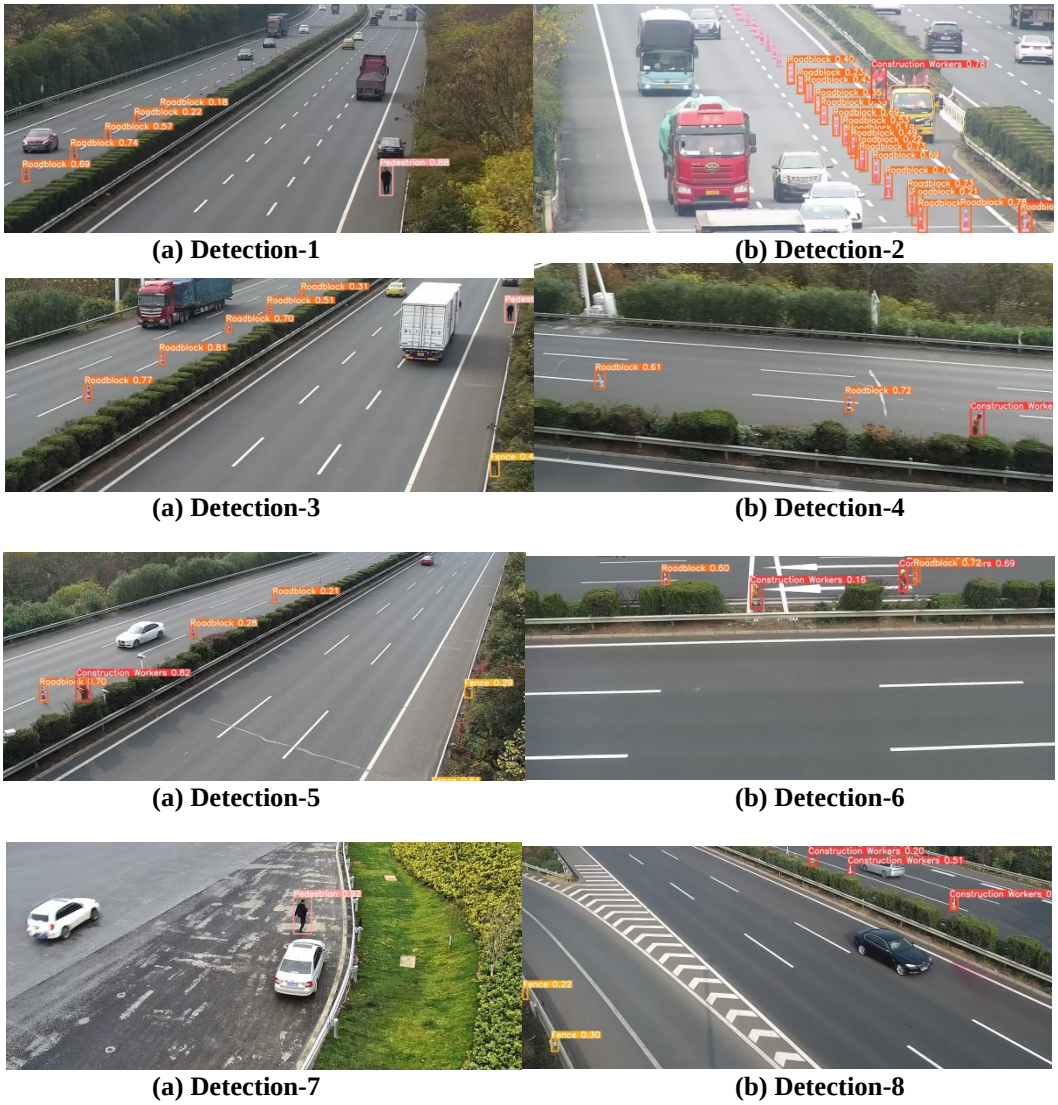
We mainly focus on the precision of the model and mAP performance indicators. As can be seen from **Table 3**, the precision of the original YOLOv5s model is 82.02%, and the mAP is as high as 88.65%. After sparsification, the precision rate did not change much, about 82%, while mAP decreased by 4 percentage points; after pruning, the precision rate remained at 82.8%, and mAP remained at 84.4%; after fine-tuning, the precision rate increased to 87%, while mAP declined, but also above 83%.

To sum up, after sparsification, pruning and fine-tuning, the recognition precision of the pedestrian recognition model here is improved by 5% compared with the initial model, although mAP is partially reduced, according to the model size in **Table 4**, it can be seen that the model is optimized by the algorithm in this paper, and the final size is only 7,723 KB, which is almost one-eighth of the original model size.

Table 4. Size of model

Models	Size of model.pt
YOLOv5s	55,510 KB
YOLOv5s-sparsification	27,944 KB
YOLOv5s-prune	15,021 KB
YOLOv5s-finetune_prune	7,723 KB

Although the optimized model loses some performance, it is greatly optimized in terms of model size and can be effectively applied to the deployment of various edge devices, so a certain degree of performance degradation can be accepted. After the performance comparison of the above network algorithms, as well as the final sparsification, pruning and fine-tuning optimization, this paper adopts YOLOv5s fine_tuning model to realize the final early warning system of pedestrian detection. Some of the visualized results are shown below in **Fig. 11**.



(a) Detection-1 **(b) Detection-2**

(a) Detection-3 **(b) Detection-4**

(a) Detection-5 **(b) Detection-6**

(a) Detection-7 **(b) Detection-8**

Fig. 11. Pedestrian detection visualization results using the pruned-optimized YOLOv5s network.

5. Conclusion

This paper proposes a computer vision-based pedestrian intrusion detection and early warning method, which utilizes the high detection efficiency and high recognition accuracy of the YOLOv5s network in computer vision to significantly improve the recognition accuracy of the final task. Due to the problem of large size and difficult edge device deployment, this paper performs BN layer pruning optimization, and the optimized model is only one-eighth the size of the original model with acceptable performance degradation. After conducting experiments, the final results show that the YOLOv5s model proposed and optimized in this paper is superior to other detection methods in detection performance, and achieves an accuracy of more than 87% in Precision. And the model size is only 7 M, which is suitable for the deployment of edge devices.

References

[1] DC. Nguyen, M. Ding, PN. Pathirana, A. Seneviratne, J. Li, D. Niyato, O. Dobre, H. Vincent Poor, "6G internet of things: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 359–383, Jan. 2022. Article (CrossRef Link)

[2] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi, "6G: Opening new horizons for integration of comfort, security and intelligence," *IEEE Wireless Communications Magazine*, vol. 27, no. 5, pp. 126–132, Oct. 2020. Article (CrossRef Link)

[3] F. Tang, B. Mao, N. Kato, and G. Gui, "Comprehensive survey on machine learning in vehicular network: technology, applications and challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 2027–2057, thirdquarter, 2021. Article (CrossRef Link)

[4] Y. Lin, M. Wang, X. Zhou, G. Ding, and S. Mao, "Dynamic Spectrum Interaction of UAV Flight Formation Communication With Priority: A Deep Reinforcement Learning Approach," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 3, pp. 892-903, Sept. 2020. Article (CrossRef Link)

[5] Y. Lin, Y. Tu, Z. Dou, L. Chen, and S. Mao, "Contour Stella Image and Deep Learning for Signal Recognition in the Physical Layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 34-46, March 2021. Article (CrossRef Link)

[6] G. Zheng, and Y. Chen, "A review on vision-based pedestrian detection," in *Proc. of IEEE Global High Tech Congress on Electronics*, pp. 49–54, 2012. Article (CrossRef Link).

[7] X. Chen, E. Li, J. Li, S. Yang, S. Zhang, and Z. Wang, "Research on pedestrian intrusion detection in static scenes," in *Proc. of Prognostics and Health Management Conference (PHM)*, pp. 444–448, 2022. Article (CrossRef Link)

[8] B. Han, Y. Wang, Z. Yang, and X. Gao, "Small-scale pedestrian detection based on deep neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 3046–3055, Jul. 2020. Article (CrossRef Link)

[9] B. Pottier, L. Rasolofondraibe, and S. Kerroumi, "Pedestrian detection strategy in urban area: capacitance probes and pedestrians' signature," *IEEE Sensors Journal*, vol. 17, no. 17, pp. 5663–5668, Sep. 2017. Article (CrossRef Link)

[10] H. Song, "The application of computer vision in responding to the emergencies of autonomous driving," in *Proc. of International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, pp. 1–5, 2020. Article (CrossRef Link)

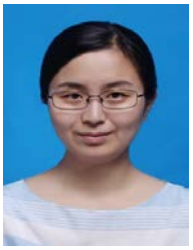
[11] G. Akyol, A. Kantarcı, A. E. Çelik, and A. Cihan Ak, "Deep learning based, real-time object detection for autonomous driving," in *Proc. of 28th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2020. Article (CrossRef Link)

[12] G. Ros, S. Ramos, M. Granados, A. Bakhtiary, D. Vazquez, and A. M. Lopez, "Vision-based offline-online perception paradigm for autonomous driving," in *Proc. of IEEE Winter Conference on Applications of Computer Vision*, pp. 231–238, 2015. Article (CrossRef Link)

[13] B. Kanchana, R. Peiris, D. Perera, D. Jayasinghe, and D. Kasthurirathna, "Computer vision for autonomous driving," in *Proc. of International Conference on Advancements in Computing (ICAC)*, pp. 175–180, 2021. Article (CrossRef Link)
[14] J. Chen, and W. Pan, "Study on airport runway incursion monitoring based on computer vision," in *IEEE International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, pp. 1328–1333, 2021. Article (CrossRef Link)
[15] B. Qi, W. Zhao, H. Zhang, Z. Jin, X. Wang, and T. Runge, "Automated traffic volume analytics at road intersections using computer vision techniques," in *Proc. of International Conference on Transportation Information and Safety (ICTIS)*, pp. 161–169, 2019. Article (CrossRef Link)
[16] A. Lad, P. Kanaujia, Soumya and Y. Solanki, "Computer vision enabled adaptive speed limit control for vehicle safety," in *Proc. of International Conference on Artificial Intelligence and Machine Vision (AIMV)*, pp. 1–5, 2021. Article (CrossRef Link)
[17] H. He, "Yolo target detection algorithm in road scene based on computer vision," in *Proc. of IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, pp. 1111–1114, 2022. Article (CrossRef Link)
[18] H. He, Z. Yan, Z. Geng, and X. Liu, "Research on pedestrian tracking algorithm based on deep learning," in *Proc. of International Conference on Computer Information Science and Artificial Intelligence (CISAI)*, pp. 487–490, 2021. Article (CrossRef Link).
[19] Z. Ahmed, R. Iniyavan, and M. M. P., "Enhanced vulnerable pedestrian detection using deep learning," in *Proc. of International Conference on Communication and Signal Processing (ICCSP)*, pp. 0971–0974, 2019. Article (CrossRef Link)
[20] H. Song, I. K. Choi, M. S. Ko, J. Bae, S. Kwak, and J. Yoo, "Vulnerable pedestrian detection and tracking using deep learning," in *Proc. of International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–2, 2018. Article (CrossRef Link)
[21] C. B. Hou, G. W. Liu, Q. Tian, Z. C. Zhou, L. J. Hua, and Y. Lin, "Multi-signal modulation classification using sliding window detection and complex convolutional network in frequency domain," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 19438-19449, 2022. Article (CrossRef Link)
[22] X.-X. Zhang, B. Adebisi, H. Gacanin, and F. Adachi, "NAS-AMR: neural architecture search based automatic modulation recognition method for integrating sensing and communication system," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 3, pp. 1374–1386, Sept. 2022. Article (CrossRef Link)
[23] B. Dong, B. Adebisi, H. Gacanin, et al., "A lightweight decentralized learning-based automatic modulation classification method for resource-constrained edge devices," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 24708-24720, 2022. Article (CrossRef Link)
[24] G. Gui, J. Wang, J. Yang, M. Liu, and J.-L. Sun, "Frequency division duplex massive multiple-input multiple-output downlink channel state information acquisition techniques based on deep learning," *Journal of Data Acquisition and Processing*, vol. 37, no. 3, pp. 502–511, May 2022.
[25] J. Yang, Y. Wang, et al., "MobileNet and knowledge distillation based automatic scenario recognition method in vehicle-to-vehicle systems," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 11006-11016, 2022. Article (CrossRef Link)
[26] H. Huang, G. Haris, et al., "Unsupervised learning-inspired power control methods for energy-efficient wireless networks over fading channels," *IEEE Transactions on Wireless Communications*, vol. 21, no. 11, pp. 9892-9905, 2022. Article (CrossRef Link)
[27] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014. Article (CrossRef Link)
[28] R. Girshick, "Fast R-CNN," in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015. Article (CrossRef Link)
[29] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, 2018. Article (CrossRef Link)

[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017. Article (CrossRef Link)
[31] S. Akçali, and F. Erden, "Support of data augmentation with GAN on faster R-CNN based buried target detection," in *Proc. of Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2021. Article (CrossRef Link)
[32] Z. Hui, Z. Li, and A. Du, "Garbage classification system based on improved faster R-CNN with audio-visual combination," in *Proc. of International Conference on Intelligent Computing and Signal Processing (ICSP)*, pp. 1235–1239, 2022. Article (CrossRef Link)
[33] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun, "Object detection networks on convolutional feature maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1476–1481, Jul. 2017. Article (CrossRef Link)
[34] Y. Lu, L. Zhang, and W. Xie, "YOLO-compact: an efficient YOLO network for single category real-time object detection," in *Proc. of Chinese Control And Decision Conference (CCDC)*, pp. 1931–1936, 2020. Article (CrossRef Link)
[35] T. H. Wu, T. W. Wang, and Y. Q. Liu, "Real-time vehicle and distance detection based on improved Yolo v5 network," in *Proc. of World Symposium on Artificial Intelligence (WSAI)*, pp. 24–28, 2021. Article (CrossRef Link)
[36] T. T. Feng, and H. Y. Ge, "Pedestrian detection based on attention mechanism and feature enhancement with SSD," in *Proc. of 2020 5th International Conference on Communication, Image and Signal Processing (CCISP)*, pp. 145–148, 2020. Article (CrossRef Link)
[37] W. Cao, J. Zhang, C. Cai, Q. Chen, Y. Zhao, Y. Lou, and W. Jiang, "CNN-based intelligent safety surveillance in green IoT applications," *China Communications*, vol. 18, no. 1, pp. 108–119, Jan. 2021. Article (CrossRef Link)
[38] Y. Zhao, Y. Yin, and G. Gui, "Lightweight deep learning based intelligent edge surveillance techniques," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1146–1154, Apr. 2020. Article (CrossRef Link)
[39] P. Viola, and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1–1, 2001. Article (CrossRef Link)
[40] P. Viola, and M. Jones, "Robust real-time face detection," in *Proc. of Eighth IEEE International Conference on Computer Vision*, pp. 747–747, 2001. Article (CrossRef Link)
[41] N. Dalal, and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 886–893, 2005. Article (CrossRef Link)
[42] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008. Article (CrossRef Link)
[43] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *Proc. of 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2241–2248, 2010. Article (CrossRef Link).
[44] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010. Article (CrossRef Link)
[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.
[46] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: a survey," 2019. Article (CrossRef Link).
[47] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904-1916, 2015. Article (CrossRef Link)

[48] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017. Article (CrossRef Link)
[49] T. Y. Lin, P. Dollar, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–4, 2017. Article (CrossRef Link)
[50] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016. Article (CrossRef Link)
[51] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *Proc. of European Conference on Computer Vision*, pp. 21–37, 2016.
[52] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318-327, 2020. Article (CrossRef Link)
[53] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning Efficient Convolutional Networks through Network Slimming," in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pp. 2755–2763, 2017. Article (CrossRef Link)
[54] Y. Lin, Y. Tu, and Z. Dou, "An Improved Neural Network Pruning Technology for Automatic Modulation Classification in Edge Devices," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5703-5706, May 2020. Article (CrossRef Link)



Xinyun Yan received the M.S. degree from the Nanjing Normal University,Nanjing,China,in 2013.She is currently pursuing the Ph.D. degree in computer science and technology with Hohai University, Nanjing, China.Her research interests include computer vision and intelligent transportation.

Zhengran He graduated from Wuhan University with a bachelor's degree in Microelectronics Science and Engineering, and is currently studying for a master's degree in Electronic Information at Nanjing University of Posts and Telecommunications. At present, the main research direction is human behavior recognition based on computer vision.

Youxiang Huang graduated from the School of Physics and Electronic Information of Anhui Normal University, majoring in Communication engineering. Now he is studying at the School of Communication and Information Engineering of Nanjing University of Posts and Telecommunications, majoring in electronic information. His current research interests include deep learning, artificial intelligence, computer vision, Internet of Things technology.

Xiaohu Xu graduated from Anhui Jianzhu University with a bachelor's degree in Electronics and Information Engineering, and is currently studying for a master's degree in Electronic Information at Nanjing University of Posts and Telecommunications. At present, the main research direction is multi-modal signal identification and interpretability analysis.

Jie Wang received the the Ph.D. degree in signal and information processing from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2022. She is currently working as an Lecturer with the Engineering School of Networks and Telecommunications, Jinling Institute of Technology, Nanjing China. Her current research interests include deep learning, computer vision and their applications in intelligent transportation and intelligent wireless communication.

Xiaofeng Zhou received the B.S., M.S., and Ph.D. degrees in computer application from Hohai University,Nanjing, China. He is currently a Professor and the Director of the Institute of Computer Science and Technology. His current research interests include software reuse and Internet of Things.

Chishe Wang received the Ph.D. degree from the Anhui University, Hefei, China, in 2009. Since 2015, he has been a professor with Jinling Institute of Technology, Nanjing, China. His recent research interests include big data, artificial intelligence, machine learning and intelligent transportation. He is also the vice president of the Engineering School of Networks and Telecommunications, Jinling Institute of Technology, Nanjing China. And he is the talent cultivated by the outstanding young backbone teachers of Jiangsu Province's "Qinglan Project" and the third-level talent cultivated by the fourth phase of "333 Project" in Jiangsu Province.

Zhiyi Lu graduated from Nanjing University of Finance and Economics with a bachelor's degree in financial engineering. In 2018, she joined Nanjing Great Information Technology Co., Ltd. and engaged in project management of artificial intelligence and the Internet of Things.