Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.063288

ARTICLE

A Pedestrian Sensitive Training Algorithm for False Positives Suppression in Two-Stage CNN Detection Methods

Qiang Guo[1,2,*], Rubo Zhang[1] and Bingbing Zhang[3]

[1]College of Mechanical and Electronic Engineering, Dalian Minzu University, Dalian, 116650, China
[2]Dalian University of Technology and Postdoctoral Workstation of Dalian Everyday Good Electronic Co., Ltd., Dalian, 116650, China
[3]School of Computer Science and Engineering, Dalian Minzu University, Dalian, 116650, China

*Corresponding Author: Qiang Guo. Email: guoqiang01486@dlnu.edu.cn

Received: 10 January 2025; Accepted: 07 April 2025; Published: 09 June 2025

ABSTRACT: Pedestrian detection has been a hot spot in computer vision over the past decades due to the wide spectrum of promising applications, and the major challenge is false positives that occur during pedestrian detection. The emergence of various Convolutional Neural Network-based detection strategies substantially enhances pedestrian detection accuracy but still does not solve this problem well. This paper deeply analyzes the detection framework of the two-stage CNN detection methods and finds out false positives in detection results are due to its training strategy misclassifying some false proposals, thus weakening the classification capability of the following subnetwork and hardly suppressing false ones. To solve this problem, this paper proposes a pedestrian-sensitive training algorithm to help two-stage CNN detection methods effectively learn to distinguish the pedestrian and non-pedestrian samples and suppress the false positives in the final detection results. The core of the proposed algorithm is to redesign the training proposal generating scheme for the two-stage CNN detection methods, which can avoid a certain number of false ones that mislead its training process. With the help of the proposed algorithm, the detection accuracy of the MetroNext, a smaller and more accurate metro passenger detector, is further improved, which further decreases false ones in its metro passenger detection results. Based on various challenging benchmark datasets, experiment results have demonstrated that the feasibility of the proposed algorithm is effective in improving pedestrian detection accuracy by removing false positives. Compared with the existing state-of-the-art detection networks, PSTNet demonstrates better overall prediction performance in accuracy, total number of parameters, and inference time; thus, it can become a practical solution for hunting pedestrians on various hardware platforms, especially for mobile and edge devices.

KEYWORDS: Pedestrian detection; false positives; CNN; edge devices

1 Introduction

Pedestrian detection has always been fundamental in various artificial intelligence tasks, such as autonomous driving, pedestrian tracking, and abnormal behavior detection. It is indispensable in their successful applications and deployments [1]. However, compared to general object detection, pedestrian detection presents specific technical difficulties that make it one of the most challenging subfields of object detection. First, pedestrians are often found in complex and ever-changing scenes where other objects may appear similar, making it difficult for detectors to differentiate between them, even with advanced pedestrian feature extraction techniques. Second, the high intra-class variation in pedestrians due to differences in clothing, lighting, and pose requires the learned human features to be more semantically meaningful and robust to achieve accurate pedestrian recognition. This poses a significant challenge for feature extraction

and detection strategy design. Finally, pedestrians are frequently occluded by other objects or each other, leaving only partial human bodies visible to the detectors. This lack of complete pedestrian features hinders the detector's ability to segment and localize individual pedestrians from the crowd.

Over the past decades, a major methodology has enriched key pedestrian features for recognizing pedestrians in complex scenes [2,3]. Recently, deep learning, specifically Convolutional Neural Networks (CNN), has become the algorithm of choice for pedestrian detection. This is thanks to the significant improvements made in the realm of CNN [3]. However, False Positives (FPs) remain a notorious problem in pedestrian detection. While using CNN has enhanced the feature extraction capabilities of the entire detection framework, most CNN-based pedestrian detectors are adapted from general object detectors. These general detectors emphasize target position awareness and multiple object classification. In contrast, pedestrian detection is a binary classification problem requiring pedestrian awareness and the ability to classify pedestrian and non-pedestrian objects effectively. The issue of avoiding FPs is often overlooked in pedestrian detection, leading to the suboptimal performance of CNN-based detectors and making it difficult to improve their pedestrian detection capabilities further.

To address the issues above and improve detection accuracy, this paper conducts a deep analysis of the detection process of a CNN-based, accuracy-oriented, two-stage detection network. It highlights that the current training strategies are insufficient to enhance the network's classification capabilities, particularly in distinguishing between pedestrians and non-pedestrians, leading to many False Positives (FPs). To mitigate this issue, a novel Pedestrian Sensitive Training (PST) algorithm is proposed. This algorithm aims to strengthen the classification abilities of this detection network, thereby reducing FPs. Moreover, the recognition of pedestrians is often required on edge devices, which necessitates that any proposed solution improves the FP removal capability of the base detector without incurring significant additional computational costs. This is crucial to meet the hardware constraints of target platforms. Given these requirements, this paper selects a compact two-stage detector, MetroNext [4], and integrates it with the PST algorithm to create a small, fast, and accurate pedestrian detector called PSTNet. The effectiveness of PSTNet is validated using real-life metro station datasets, specifically SY-Metro, and on an embedded platform to assess its capability to quickly and accurately detect metro passengers, potentially replacing human surveillance. In summary, this paper makes three contributions:

1) This paper deeply analyses the detection pipeline of the two-stage CNN-based detection network. It points out that its training strategies can't help the whole detection network have a stronger classification ability to distinguish pedestrians and non-pedestrians. Thus, a novel PST algorithm is proposed, which can effectively guide the training process of the two-stage CNN-based detection network and promote its classification capability to wipe out non-pedestrian predictions, achieving zero-cost accuracy improvements.

2) Various experiments have been conducted on challenging benchmark datasets and a real-life metro station dataset: SY-Metro. The ablation and benchmark experiments on these datasets demonstrate that the PST algorithm has the general ability and adaptability to improve the prediction accuracy of plain models and effectively suppresses false positives even for the small model. The metro scene dataset tests the newly-built PTSNet's ability to accurately detect pedestrians in crowded metro scenes, confirming its effectiveness in delivering pedestrian detection results accurately in such scenarios.

3) In order to accurately measure the detection latency and power consumption of the PSTNet, the inference speed and power usage analysis experiments are conducted on workstations and embedded platforms. The experiment results demonstrate that PSTNet achieves faster inference speed and lower power consumption compared to other competitors. This makes PSTNet a suitable pedestrian detection solution on embedded platforms.

2 Related Work

Over the past decades, the main solutions have been divided into traditional and emerging deep learning-based methods, which are briefly summarized below.

2.1 The Traditional Methods

Vision-based pedestrian detection methods obtain detection results through two key image processing steps: feature extraction and proposal classification. The traditional techniques design various handcrafted filters to extract pedestrian features and further process them via the following subnetwork to output detection results [5–7]. Due to the weak feature extraction capability of the handcrafted filters, the extracted features are poorly semantic in foreground (pedestrian object) and background (non-pedestrian object) information, making it difficult for the top classification network to distinguish between pedestrians and background images using these features, resulting in some false ones in the detection results. References [8–10] have proposed different tricks to suppress these FPs, among them, reference [8] creatively proposed a multi-resolution infrared vision pedestrian detection system, which designed a series of matched filters to avoid several FPs, and the efficient of this pedestrian detection system had been proved in various situations with lower false-positive rates. However, the traditional methods have limited learning ability and can't be adapted to the considerable intra-class variation of backgrounds and pedestrians, thus preventing further improvements in detection accuracy.

2.2 The Deep Learning-Based Methods

To compensate for the limitations of manually designed filters in traditional methods for pedestrian feature representation, reference [11] first represented pedestrian attributes using pedestrian features generated in CNN and then used these features to train SVM classifiers to recognize human objects from input images. Since then, references [12–15] continuously improved the pedestrian detection accuracy on popular benchmark datasets. For false positive detection results, researchers have also proposed different technical routes to deal with this problem.

3D convolution operation Differing from 2D convolution operation, the 3D convolution operation can slide over a 3D volume, so it can directly process 3D perception data and help the detector to describe the spatiotemporal relationship of objects in the 3D space. Reference [16] proposed to combine 2D deep learning-based pedestrian detectors with a 3D CNN to wipe out false ones, where the 2D deep learning-based pedestrian detector is responsible for providing potential pedestrian proposals. In contrast, 3D CNN determines whether these proposals are true positives. Similarly, Reference [17] adopted the same detection pipeline to reject FPs and demonstrated that this method has better general capability than the CNN-based detection methods. Reference [18] proposed a novel 3D feature-pulling strategy to achieve 2D to 3D feature transformation, which demonstrates fewer false positives. These 3D convolution FPs suppression strategies increase the computational complexity of the whole detection system when plugging a 3D network to process 3D features. Thus, the application scenarios for this method are relatively limited, and it can't be deployed in the application scenarios with limited hardware resources.

Multimodal fusion The multimodal fusion effectively reduces false positives by integrating various types of input data. Meanwhile, this approach also involves developing recognition models and algorithms specifically tailored to process the fused data. Reference [19] combines the multimodal object detector with Kernel Extreme Learning Machine and Hybrid Salp Swarm Optimization algorithm to build the IPDC-HMODL model, which adopts an image processing techniques to process two kinds of input data: image frames and its ground truth images to help the IPDC-HMODL model to suppress the false ones in detection results. Reference [20] proposed a CNN-based LiDAR-camera fusion mechanism and then

designed a neural network to confirm the recognition results to reduce FPs. The comparison study has highlighted the performance gains of these methods. However, the integration of multimodal information faces several challenges. Firstly, the acquisition of multiple sensors can significantly increase its development costs. Secondly, the alignment of information across various sensors requires additional processing steps, which promote the system's complexity.

Non-maximum suppression Choosing the ideal NMS threshold for pedestrian detection remains a challenging task, and the higher threshold brings more FPs. At the same time, a lower one causes a higher miss rate. Reference [21] proposed an adaptive NMS strategy that can compute a suitable suppression threshold according to the object density, but this approach fails when accurate object density information is unavailable. Unlike standard NMS strategy, Reference [22] design a novel Representative Region NMS strategy to calculate the Intersection over Union (IoU) of two objects, where representative region boxes of two objects are used to compute this value, and a Paired-Box Model is proposed to responsible for predicting pedestrian's representative region boxes. However, the problem of this method is how to ensure that the PBM has strong generalization capabilities that can provide accurate predictions of pedestrian's representative region boxes in dynamically changing real-life scenarios. Moreover, it brings more human annotation work to label human visible bodies when establishing training datasets. Reference [23] had a similar technical route to optimize the NMS algorithm, and the false positives in the pedestrian detector have been reduced to a certain extent. However, the problem above has still not been solved.

Feature enhancement The Feature enhancement method constructs various image processing modules to strengthen and enrich pedestrian features to help the detector recognize hard samples. Reference [24] adopted two key components: attention modules and reverse fusion blocks to build a semantic attention fusion mechanism to increase the classification capability of the detector. Reference [25] designed a Pose-Embedding Network that combines human pose information with visual description and used the pedestrian pose information to re-evaluate the confidence scores of pedestrian proposals and eliminate the false ones with high confidence scores. However, the main drawback of these methods is that the constructed feature information extraction module further increases the computational cost of the deep learning methods, making them more demanding on the computational resources of the target platforms, which hinders their widespread adoption on low-end computing devices.

In summary, although several research works have been carried out to effectively remove pedestrian false positives, to the best of our knowledge, there is still a lack of a suitable method for deployment on embedded devices with good pedestrian false positives suppression with no extra computational cost. Therefore, this paper has devoted our efforts to researching this direction by proposing a training algorithm to help the two-stage CNN pedestrian detection method achieve this goal.

3 Methodology

3.1 The Training Strategy of the Two-Stage CNN-Based Detection Method

As is shown in Fig. 1, in the training stage, the backbone network extracts deep convolutional features from the input images. Then, the several convolutional layers of the Region Proposal Network (RPN) utilize these features to obtain coordinate offsets and scores of predefined anchors and send them to the following proposal layer to produce pedestrian proposals. Where the Intersection over Union (IoU) strategy is utilized to classify pedestrian proposals. However, the IoU strategy uses a threshold to classify pedestrian proposals, which will misclassify some ones containing human body information as negative ones, and these misclassified proposals are mixed into the training samples, which will directly influence the training

process of the subnetwork to weaken its pedestrian/background distinguishing ability, and inevitably lead to some FPs.

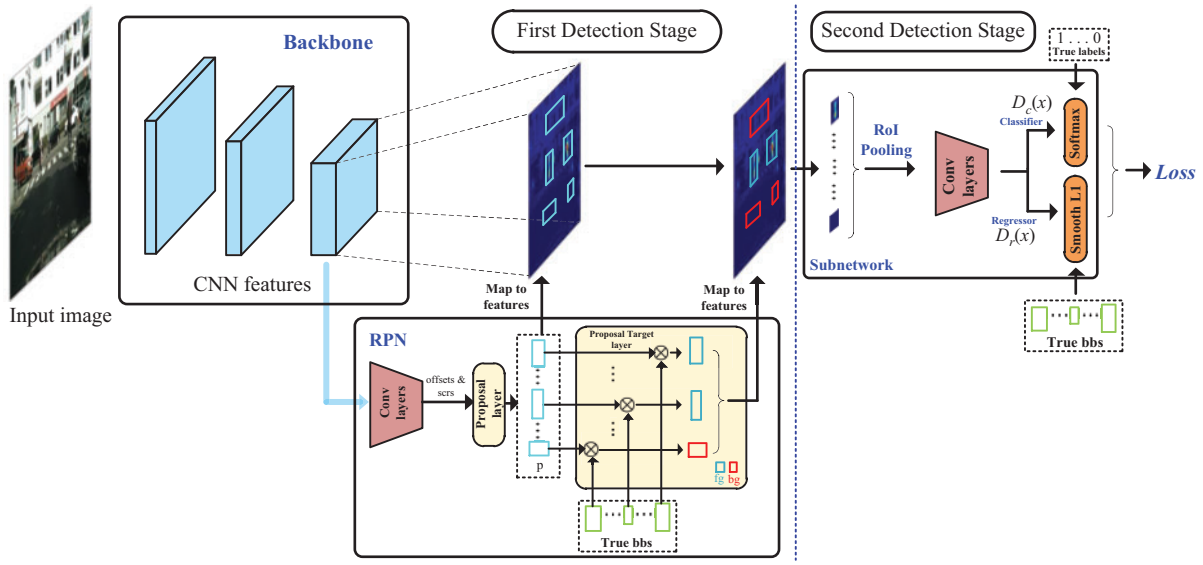


Figure 1: The training strategy of two-stage CNN-based pedestrian detection paradigm. Where "offsets & scrs" denotes the coordinate offsets and scores of predetermined anchors. "bbs" represents the bounding boxes. "p" denotes the proposals and their foreground and background are represented by "fg,bg". "RoI Pooling" denotes the RoI Pooling layer. "$D_c(x), D_r(x)$" denote the classifier and regressor of the subnetwork. $\otimes$ denotes the operator for calculating IoU values

3.2 The Drawback of IoU Strategy

IoU strategy is a common object detection metric used to measure the matching degree of two bounding boxes. The higher the IoU value of the two objects, the better the match between them. Generally, an IoU threshold is predetermined to evaluate the hit rate of the predictions to the ground truth. When the IoU value exceeds the threshold, the predictions hit the GT (considered in the foreground). Otherwise, they don't hit it (considering the background). Therefore, the IoU threshold is a very important hyperparameter, and choosing an appropriate IoU threshold has always been a tricky problem [26]. In addition, since the predefined IoU threshold can't be changed adaptively, it's unsuitable for dealing with object detection problems with large-scale variations such as pedestrian detection. This can be observed from its computational formula.

Given two bounding boxes $p = \{x_1^p, y_1^p, x_2^p, y_2^p\}$ and $tb = \{x_1^{tb}, y_1^{tb}, x_2^{tb}, y_2^{tb}\}$, their IoU value is computed

$$IoU = \left| \frac{p \cap tb}{p \cup tb} \right| \tag{1}$$

where $p \cap tb$ and $p \cup tb$ represent the area of interaction and union region of p and tb, respectively.

According to Eq. (1), if the denominator is much larger than the numerator, a lower IoU score below the IoU threshold will be obtained, and the corresponding proposal will be regarded as a negative training sample, which may occur in pedestrian detection. As shown in Fig. 2, although several proposals contain human bodies, their IoU values after IoU computation with true bounding boxes are lower than the IoU threshold. Thus, they are misclassified as negative samples, which directly affects the training process of

the two-stage CNN-based pedestrian detection paradigm, so it can't effectively classify pedestrians and backgrounds, which leads to some FPs. A direct solution is to lower the IoU threshold, but only adjusting the IoU threshold during the training stage can't effectively solve classification issues between positive and negative ones. For this reason, this paper proposes the PST algorithm to meet this demand.

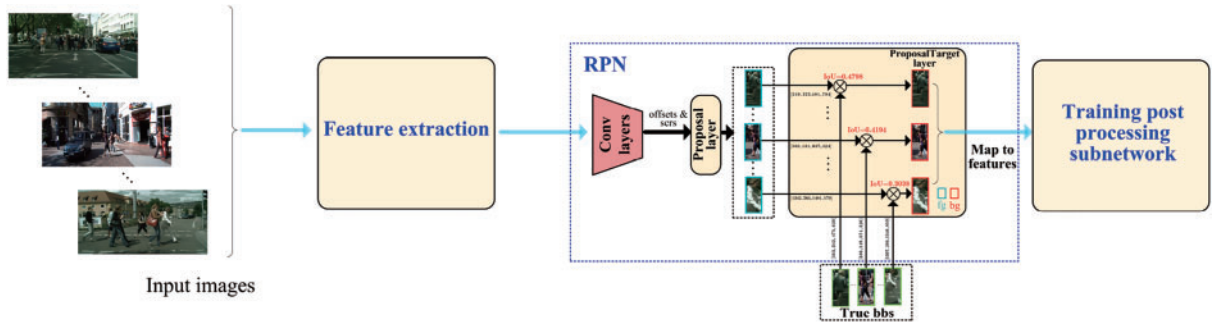


Figure 2: Some training samples are misclassified to negative ones using the IoU strategy

3.3 The PST Algorithm

As mentioned above, false positives occur because the IoU strategy misclassifies the training samples, thereby affecting the training process of the subnetwork and weakening its classification ability. The solution is to accurately classify positive and negative samples to the network and guide it to train effectively. During the training process of the two-stage pedestrian detection paradigm, the network must share weights between building accurate pedestrian localization and classification capabilities, making it difficult to power the network to build strong classification capabilities [27]. Therefore, this paper adopts the cascading strategy to design the PST algorithm to enhance this paradigm's pedestrian/background classification capability, which empowers this paradigm with pedestrian sensitivity and helps it accurately differentiate pedestrians/backgrounds during training. The theory of the PST algorithm is described below.

Overall processing pipeline The PST algorithm brings the step of evaluating the pedestrian information quantity of each proposal and changes the allocation process of positive and negative proposals in the proposal target layer, which is plugged into the two-stage CNN-based pedestrian detection paradigm (demonstrated in Fig. 3), in which a series of proposals $\boldsymbol{P}$ generated from the proposal layer can be denoted as

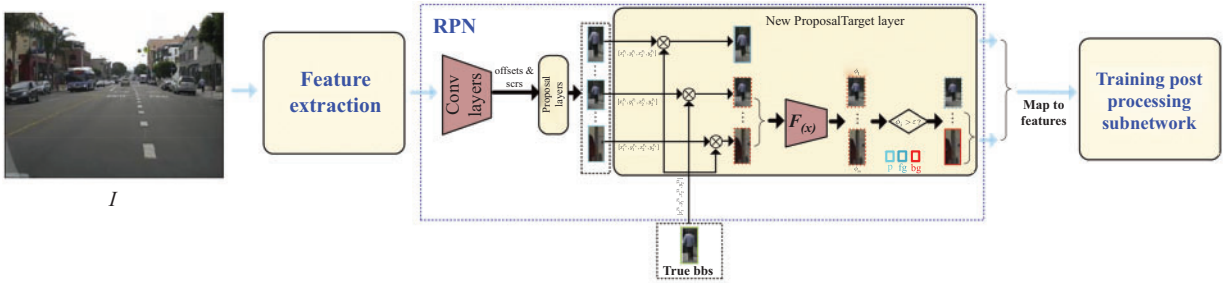


Figure 3: The processing pipeline of the PST algorithm, where $\boldsymbol{I}$ represents the input image. $F(x)$ denotes the pedestrian sensitive classifier. ϕ_i represents the pedestrian confidence of the proposal. ε represents the confidence threshold for pedestrians. All dotted bounding boxes indicate these proposal are not used for training subnetwork

$$\boldsymbol{P} = \{p_1, \cdots, p_n\} \tag{2}$$

where ith proposal is represented as $p_i = \{x_1^{p_i}, y_1^{p_i}, x_2^{p_i}, y_2^{p_i}\}$, $1 \le i \le n$. These proposals are computed with the true bounding box tb to obtain their IoU scores.

$$\boldsymbol{S}_{IoU} = \left\{ \left| \frac{p_1 \cap tb}{p_1 \cup tb} \right|, \cdots, \left| \frac{p_n \cap tb}{p_n \cup tb} \right| \right\} \tag{3}$$

where the IoU score of i-th proposal is denoted as $IoU_i = \left| \frac{p_i \cap tb}{p_i \cup tb} \right|$. Comparing each IoU score with the threshold ε_{IoU}, the corresponding proposals can be divided into positive and negative training samples, as is

$$\begin{aligned} \boldsymbol{P}^+ &= \{p_i, IoU_i \ge \varepsilon_{IoU}, 1 \le i \le n\} \\ \boldsymbol{P}^- &= \{p_j, IoU_j < \varepsilon_{IoU}, 1 \le j \le n\} \end{aligned} \tag{4}$$

In the redesigned proposal target layer, the $\boldsymbol{P}^-$ can't be seen as the true negative samples and require a new processing procedure to reevaluate the human information of each proposal and suppress the proposals with rich human body information. To this end, a new-designed pedestrian-sensitive classifier is adopted to finish this job, which processes the $\boldsymbol{P}^-$ corresponding input image slices (represented as $\boldsymbol{I}_{\boldsymbol{P}^-}$) and output their human body information scores.

$$\boldsymbol{\Phi} = F(\boldsymbol{I}_{\boldsymbol{P}^-}), \boldsymbol{\Phi} = (\phi_1, \ldots, \phi_m) \tag{5}$$

When scores in the $\boldsymbol{\Phi}$ higher than the threshold, the corresponding proposals will be omitted from $\boldsymbol{P}^-$, which solves the problem that these samples containing rich human torso information mislead the training process of the subnetwork. In addition, these proposals are not used as positive ones to train the subnetwork because the positioning accuracy of these proposals is poor, which requires larger coordinate offsets and affects the convergence of the training process of the network. Thus, the new negative samples are

$$\boldsymbol{P}_n^- = \{p_i, \phi_i < \varepsilon, 1 \le i \le m\} \tag{6}$$

Finally, the $\boldsymbol{P}^+$ and $\boldsymbol{P}_n^-$ is merged to a new training proposal to effectively guide the training process of the subnetwork to construct a stronger pedestrian/background classification capability.

$$\boldsymbol{P}_t = \boldsymbol{P}^+ \cup \boldsymbol{P}_n^- \tag{7}$$

Pedestrian-sensitive classifier The key component of the PST algorithm is the pedestrian-sensitive classifier, which helps the PST algorithm accurately classify the pedestrian/background proposals. Moreover, the classifier must have a low computational burden so as not to slow down the paradigm's training speed in each training epoch. As a result, the newly designed classifier needs to have a performance balance between stronger classification capabilities and fewer computational resources. This paper uses the following methods to construct the pedestrian-sensitive classifier to achieve this goal.

Classifier MacroArchitecture The pedestrian-sensitive classifier is a CNN-based classifier, which can directly process the images and extract the image features to classify the object. The architecture of the classifier can be expressed as

$$F(x) = (L_1 \circ, \cdots, \circ L_i \circ, \cdots, \circ L_k)(\boldsymbol{I}) \tag{8}$$

where L_i is a network layer and k means the total layers of the classifier. $\circ$ denotes the layer connection.

To select the total number of layers, the resolution of the input image must be considered first. The chosen image resolution should not be excessively high, as this would increase the training time cost. Therefore, balancing image resolution and computational efficiency is essential to ensure optimal model performance. In pedestrian detection datasets, most pedestrians are observed at a scale of 30 to 80 pixels [28]. Therefore, a 64×64 pixel image size is chosen as the input image size, which is suitable for pedestrian classification. This resolution effectively captures the necessary features of pedestrians while maintaining reasonable computational requirements.

Secondly, the trade-offs between receptive field and classification accuracy are critical considerations in designing neural network architectures. A larger receptive field allows the model to capture more global context, while a smaller receptive field emphasizes image patches to extract local features. The pedestrian proposals generated in the proposal layer often encompass parts of the human torso. Hence, this paper selects a moderate-sized receptive field to ensure that the model focuses on relevant local features while reducing computational complexity, thereby decreasing the training load.

The equation for calculating the receptive field of CNN is as follows [29].

$$r_n = r_{n-1} + (f_n - 1) \prod_{i=1}^{n-1} s_i \tag{9}$$

where r_n, r_{n-1} represent the receptive field of nth layer and $(n-1)$th layer, respectively.

Using Eq. (9), in a nine-layer CNN, the top layer has a receptive field of more than a quarter of the 64×64 image, enabling the classifier to extract image features effectively.

Finally, the architecture design does not utilize the residual and multi-scale feature extraction blocks. This decision is based on several considerations. First, a shallow network helps avoid the vanishing gradient problem, which is crucial for maintaining stable training. Second, each pedestrian proposal typically contains only one pedestrian or parts of their body, reducing the need for multi-scale feature extraction. Since the model does not need to handle significant variations in pedestrian size, these blocks are omitted. As a result, the network becomes more lightweight and faster, which is beneficial for speeding up the training process.

Thus, the Eq. (8) can be expressed as

$$F(x) = (L_1 \circ, \cdots, \circ L_i \circ, \cdots, \circ L_9)(\boldsymbol{I}) \tag{10}$$

Classifier MicroArchitecture Convolutional neural network has many types of network layers. This paper combines the convolutional layer, pooling layer, plus fully connected layer to construct the classifier, in which the convolutional layer is used to extract the image features and the pooling layer is used to compress the features information, and the fully connected layer is adopted to map extracted features to the label space. Then, the $F(x)$ can be as

$$F(x) = (L_1^c \circ L_1^p, \cdots, \circ L_i^c \circ L_i^p \circ, \cdots, \circ L_9^{FC})(\boldsymbol{I}) \tag{11}$$

where L_i^c, L_i^p denote i th convolutional and pooling layer, respectively. L_9^{FC} stands for the last fully connected layer. Let $\boldsymbol{W}_i^c, \boldsymbol{W}_i^p$ represent sampling matrix of convolutional and pooling layers, and $\boldsymbol{W}_9^{FC}$ stands for the fully connected weights. $\boldsymbol{X}_i \in \boldsymbol{R}^{H \times W \times C}$ denotes input features of different layers. The Eq. (11) can be rewritten as

$$F(x) = \{\{\boldsymbol{W}_8^c * \{\cdots\{\boldsymbol{W}_2^c * [(\boldsymbol{W}_1^c * \boldsymbol{I}) \downarrow \mathbf{W}_1^p]\} \downarrow \mathbf{W}_2^p \cdots\}\} \downarrow \mathbf{W}_8^p\} \boldsymbol{W}_9^{FC} \tag{12}$$

$$\mathrm{M}_{Conv} = f^2 C_i C_f W' H' \tag{13}$$

$$H' = [(H - f + 2p)/s] + 1$$
$$W' = [(W - f + 2p)/s] + 1 \tag{14}$$

where M_{Conv} stands for the computation complexity of the convolution layer. f is the filter size. C_i, C_f represents the channel number in the input image and convolution filters. p, s are the padding and stride parameters, respectively. According to Eqs. (13) and (14), the computational complex of the convolution layer is controlled by H, W, p, s, f, C_i, C_f, among which only the f, C_i, C_f has optimized selection space, so this paper need to optimize these parameters to reduce the computational complexity of the network.

For the fully connected layer, its total parameters are computed by

$$N_{FC} = IJ \tag{15}$$

where I, J are its input and output vectors' size. Since I is determined by the feature map shape of previous layers, this paper can reduce J to decrease the number of network parameters.

This paper utilizes 3 × 3 convolutional filter size to balance the computational burden and pedestrian feature extraction. The 3 × 3 convolutional filter size is effective at capturing local features. Besides, by stacking multiple 3 × 3 filters, the network can effectively capture more complex and hierarchical features.

In summary, this paper utilizes the following strategies to design the pedestrian-sensitive classifier, whose structure is illustrated in Fig. 4.

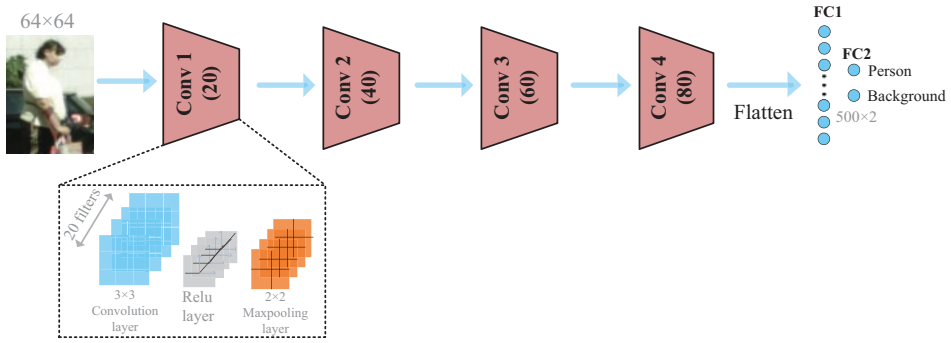


Figure 4: The architecture of the pedestrian-sensitive classifier. "Conv" denotes its convolutional block and the number in the bracket means the number of convolutional filters. "FC" stands for the Fully Connected layer. No Maxpooling layer in Conv 4 block

1) The 9-layer network structure ensures that the top layer of the classifier has a reasonable receptive field to cover the input image.

2) The residual and multi-scale feature extraction blocks are omitted in the network design to make a lightweight and fast classifier suitable for efficient training.

3) This paper adopts a 3 × 3 convolution filter and the shorter output vector length of fully connected weights to cut down the computation burden of the classifier.

To demonstrate the PST algorithm in detail, its pseudo-code is listed in Algorithm 1.

Algorithm 1: PST Algorithm

procedure PST(Input image: $\boldsymbol{I}$)
//Initialize, define the proposal set $\boldsymbol{P}$, the IoU scores set $\mathbf{S}_{IoU}$, positive and negative training samples $\boldsymbol{P}^{+}$, $\boldsymbol{P}^{-}$, pedestrian confidence scores for image slices set $\boldsymbol{\Phi}$, the refined negative training samples set $\boldsymbol{P}_{n}^{-}$ and the refined training sample set $\boldsymbol{P}_{t}$. Give the confidence score and IoU threshold $\varepsilon_{IoU}, \varepsilon$.
1: **init** $\boldsymbol{P}, \mathbf{S}_{IoU}, \boldsymbol{P}^{+}, \boldsymbol{P}^{-}, \boldsymbol{\Phi}, \boldsymbol{P}_{n}^{-}, \boldsymbol{P}_{t}, \varepsilon_{IoU}, \varepsilon$
2. Obtain feature maps $\mathbf{X}_{\text{backbone}}$ by processing $\boldsymbol{I}$.
3. Obtain proposals $\boldsymbol{P}$ by using RPN to process $\mathbf{X}_{\text{backbone}}$.
4. Obtain $\boldsymbol{P}^{+}$, $\boldsymbol{P}^{-}$ using the IoU algorithm.
    4.1 **for** $i \leftarrow 1$ *to* $i \leftarrow n$ **do**
    4.2 $IoU_i = \left|\frac{p_i \cap tb}{p_i \cup tb}\right|$
    4.3 **if** $IoU_i \geq \varepsilon_{IoU} \boldsymbol{P}^{+} \cup \{p_i\}$ **else** $\boldsymbol{P}^{-} \cup \{p_i\}$
5. Obtain $\boldsymbol{I}_{\boldsymbol{P}^{-}}$ based on the coordinates of $\boldsymbol{P}^{-}$
6. Obtain refined negative proposals $\boldsymbol{P}_{n}^{-}$ using the classifier *F(x)* to process $\boldsymbol{I}_{\boldsymbol{P}^{-}}$
    6.1 **for** $i \leftarrow 1$ *to* $i \leftarrow m$ **do**
    6.2 $\phi_i = F(\boldsymbol{I}_{p_i})$
    6.3 **if** $\phi_i < \varepsilon \boldsymbol{P}_{n}^{-} \cup \{p_i\}$
7. Obtain refined training proposals $\boldsymbol{P}_{t} = \boldsymbol{P}^{+} \cup \boldsymbol{P}_{n}^{-}$ to guide training process of the subnetwork.

As illustrated in Step 5 in this table, removing negative training samples relies on advanced feature extraction and contextual understanding of the proposed classifier, which is independent of the choice of the IoU threshold. By decoupling the IoU threshold from the training proposals generating process, the algorithm avoids the potential biases introduced by IoU threshold selection, especially in crowded pedestrian scenes [22]. This makes the algorithm particularly effective at handling highly occluded pedestrians and reduces false positives in dense crowds. The experimental results on the benchmark datasets further validate this approach, demonstrating its efficacy in improving pedestrian recognition accuracy in challenging environments.

Finally, the proposed methods have two strengths compared to other algorithms:

1) Enhanced detection capabilities without extra computing cost. During the training stage, the PST algorithm guides the model in classifying foreground and background to form stronger pedestrian detection capability and suppress false positives. Unlike other methods that require the addition of extra computational modules (e.g., feature enhancement methods), this method enhances the model's pedestrian detection capability without additional computational cost. This is advantageous for applications on resource-constrained embedded devices.
2) Redesigned proposal generating mechanism. The proposed method redesigns the proposal generation mechanism by adding a step to evaluate pedestrian feature information, then combining it with an IoU strategy that can provide high-quality pedestrian proposals for the subnetwork.

4 Experiments and Results

4.1 System Setting

Experimental platform. The evaluation experiments of the proposed methods have been done on both a workstation and an embedded platform, whose specifications are outlined in Table 1. Two hardware devices are used on the embedded platform: the Jetson Nano and the RK1808 AI compute stick, the latter of which integrates an NPU chip to offer performance up to 3TOPS, which can effectively speed up the inference of the model on the embedded platform. The two embedded devices with different computing power can comprehensively test the application potential of the model in embedded scenarios.

Table 1: The detailed hardware and software specifications of the experimental platforms

Software & Hardware	Platforms		
	Workstation	**Jetson nano**	**RK1808 AI compute stick**
CPU	Intel Core i7-6950x	ARM Cortex-A53	RK1808
MEMORY	64 G	4 G	1 G
GPU	NVIDIA TITAN X Pascal	NVIDIA Maxwell	–
Operating system	Ubuntu 18.04		–

Evaluation metrics. This paper employs total parameters, inference time, precision, recall, and miss rate as key metrics in the following experiments. These evaluation metrics provide a holistic assessment of memory usage, computational complexity, and detection accuracy of all detectors in pedestrian detection. By testing these aspects, the strengths of the PST algorithm in enhancing pedestrian detection without adding extra computational overhead are highlighted, and the strengths and weaknesses of all detectors on different platforms are reflected. In addition, all experimental setups adopt an Intersection over Union (IoU) threshold of 0.5 as the default value.

4.2 Datasets

SY-Metro dataset The images are collected from the metro on-board cameras in Shenyang, China, with 1503 images, including passenger images at varying times from different metro lines to reflect the metro scene. All the passengers in the images have been manually annotated for the metro passenger detection experiments.

The images in the SY-Metro dataset are shown in Fig. 5. During the dataset-building process, this paper collected images that covered a variety of metro scenarios to guarantee the model's ability to detect passengers in complex and diverse real-life scenarios.

Figure 5: Various metro scenes in SY-Metro dataset

This paper utilizes the benchmark datasets and the SY-Metro dataset to validate the feasibility of the proposed methods. The benchmark datasets include Caltech [28], CUHK_Occ [30], and CityPersons [31] datasets, which have been widely adopted in research works and can be used to reflect the effectiveness of the PST algorithm to solve the pedestrian detection, demonstrating the comprehensive performance of the plain detector aided by the PST algorithm in hunting persons. The metro scene dataset is used for the pedestrian detection experiments in this paper because the metro scene is the main application field of the proposed method, and this dataset can test the effectiveness of the proposed method for passenger detection in this scene. In addition, since the benchmark dataset lacks metro scene images, this dataset further validates the generalization capability of the proposed method for pedestrian detection in cross-scene applications.

4.3 Baseline Detectors

Extensive popular baseline detectors, including FasterRCNN [32], SSD [33], Tiny YOLOV3 [34], FPN [35], Pelee [36], EfficientDet [37] and Swin Transformer [38] are employed for comprehensive comparison with the proposed method in pedestrian detection, enabling a thorough validation of the PST algorithm's capability to enhance the detection accuracy of the plain detection network without extra computational cost. By comparing with these established state-of-the-art detectors, the strengths and weaknesses of the detection performance of the PST-augmented detector can be witnessed.

4.4 The Ablation Experiment

To verify the feasibility of the PST algorithm in improving the pedestrian detection accuracy of two-stage CNN-based detectors, this paper chooses the widely adopted FasterRCNN as the backbone detector and evaluates it on the CityPersons dataset. Detailed experimental results are shown in Table 2, where "MR" signifies the log average miss rate (lower is better), "Params" stands for the total number of parameters ("M" indicating millions), and "GPU" refers to the time spent per frame in milliseconds (ms/frame). Values in parentheses show the variation of corresponding performance metrics compared to the plain detector.

Table 2: Results of CityPersons dataset from FasterRCNN using the PST algorithm

Detector @ CityPersons	Backbone	MR (%)	GPU (ms/frame)	Params (M)
FasterRCNN-PST	VGG16	73.63 (−0.8)	72 (+0)	137.1 (+0)

The CityPersons dataset's experiment results, as illustrated in Table 2, exhibit a diminution in miss rate by 0.8% accomplished without any extra computational expense. This improvement in the detection accuracy of the plain detector for pedestrian detection without any additional computational burdens demonstrates the effectiveness of the PST algorithm in promoting the detector's accuracy while maintaining computational efficiency.

4.5 Benchmark Experiments

To further validate the PST algorithm's effectiveness and versatility in improving pedestrian detection accuracy across diverse real-world scenarios, mainly when applied to a compact detection network. This study adopts the compact two-stage pedestrian detector, MetroNext, as the baseline detector to build PSTNet, and then it's fully evaluated against other widely used detectors across benchmark datasets. A summary of the experimental results is presented in Table 3, which offers insights into the algorithm's capability to promote

the detector's overall detection performance under realistic conditions. From the data in this table, you can see that:

Table 3: Experimental results of PSTNet and baseline detectors on the benchmark datasets

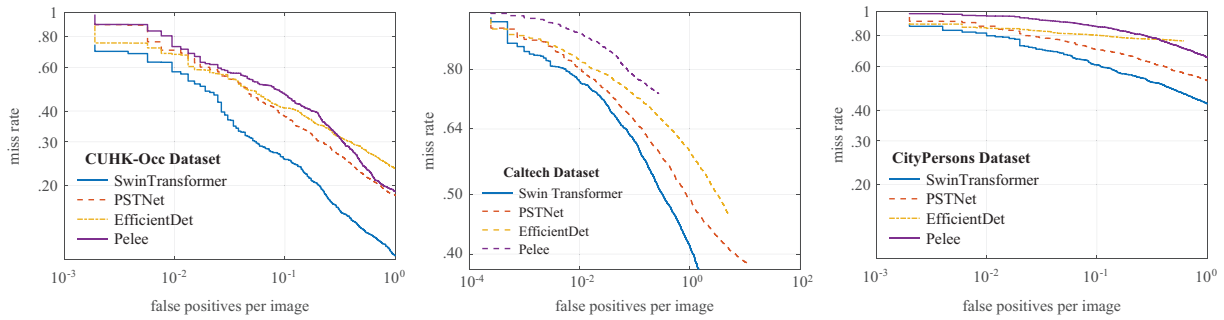
Detectors	Params (M)	CUHK-Occ		Caltech		CityPersons	
		MR (%)	GPU (ms/frame)	MR (%)	GPU (ms/frame)	MR (%)	GPU (ms/frame)
Swin Transformer	45.31	24.80	105	58.72	106	60.22	106
FPN	42.12	28.01	182	59.68	180	64.28	181
FRCNN VGG16	136.69	36.59	70	64.77	69	73.63	72
SSD512	23.75	35.81	50	69.26	50	79.41	51
Tiny YOLOV3	8.66	53.33	3	77.63	3	86.82	3
Pelee	5.29	41.28	16	74.42	16	83.96	16
EfficientDet	3.90	40.62	32	71.49	32	80.49	33
MetroNext	4.56	37.81	25	64.82	25	71.87	26
PSTNet	4.56	37.00	25	64.63	25	69.56	26

1) Empowered by the PST algorithm, the newly designed PSTNet witnesses a boost in its ability to hunt pedestrians across various scales within benchmark datasets. Specifically, in the CityPersons dataset, The 2.31% improvement in detection accuracy highlights the algorithm's ability to guide the detector to distinguish between true and false detections during training. This enhancement in accuracy demonstrates the feasibility of incorporating the proposed PST algorithm into the learning phase of the CNN-based pedestrian detection network. Thus, it can be a practical strategy tailored to CNN-based pedestrian detection networks to hunt pedestrians.
2) The PSTNet has demonstrated outstanding detection capabilities, showing competitive performance compared to other baseline detectors. In particular, despite MetroNext having a tiny model size and a smaller number of network parameters, it has better MR results within benchmark datasets, achieving 37.81%, 64.82%, and 71.87%, respectively. The integration of the PST algorithm further boosts the MetroNext detection accuracy, reducing the miss rate by up to 2% compared to the plain model without an extra increase in total parameters and inference time. This means that the PST algorithm is effective in improving the detection accuracy of the model, even for smaller detection networks.
3) Compared to other deep CNNs, while they may be excellent in pedestrian detection accuracy, they also suffer from two significant drawbacks: a large number of network parameters and relatively slow inference speed. For instance, while the Swin Transformer achieves higher accuracy, it comes with a larger parameter count of 45.31 M and an inference speed of about 100 ms/frame, the overhead of which poses a challenge for deployment in resource-constrained embedded platforms. Therefore, the drawbacks of these detectors, including a large number of network parameters and slower inference speeds, make them unsuitable for deployment on embedded platforms. Besides, compared to the EfficientDet, which is specifically oriented to the resource-constrained platform, the PSTNet strikes an optimal balance between accuracy, total parameter, and inference speed. Thus, the proposed network is more suitable for an embedded environment.

4) When the detectors are evaluated on the Caltech and CityPersons datasets, the pedestrians in these datasets are heavily occluded, which poses a significant challenge to the most powerful detection models. All detectors have higher MR results compared to their performance on the CUHK-Occ dataset. However, the established PSTNet achieves better detection accuracy, demonstrating the general capability of the PST algorithm to enhance the detection accuracy of small pedestrian detectors in complex real-world urban environments.

Discussion All detectors are trained and evaluated using a uniform dataset partition. The proposed methods achieve performance gains on benchmark datasets, showing the PST algorithm's robustness in effectively guiding the learning phase of the plain detector to distinguish pedestrians and backgrounds. Aided by the PST algorithm, The established PSTNet excels in terms of total parameters and inference speed yet lags behind larger models in accuracy. This is due to the smaller model scale of the plain detector plus the compact pedestrian-sensitive classifier in the PST algorithm, which can't store enough semantic features to aid the detector in accurately recognizing pedestrians. Consequently, PSTNet is a competitive choice for pedestrian detection in resource-constrained edge devices.

Fig. 6 depicts the graphical representation of miss rates against false positives per images for Swin Transformer, PSTNet, EfficientDet, and Pelee across benchmark datasets. These curves span miss rate intervals from 0.1 to 1 (adjusted to 0.4 to 1 for the Caltech dataset) and false positives per image ranging from 10^{-3} to 10^{0} (from 10^{-4} to 10^{2} for the Caltech dataset). A lower curve signifies superior detection performance, and the legend illustrates these detectors' log average miss rates.

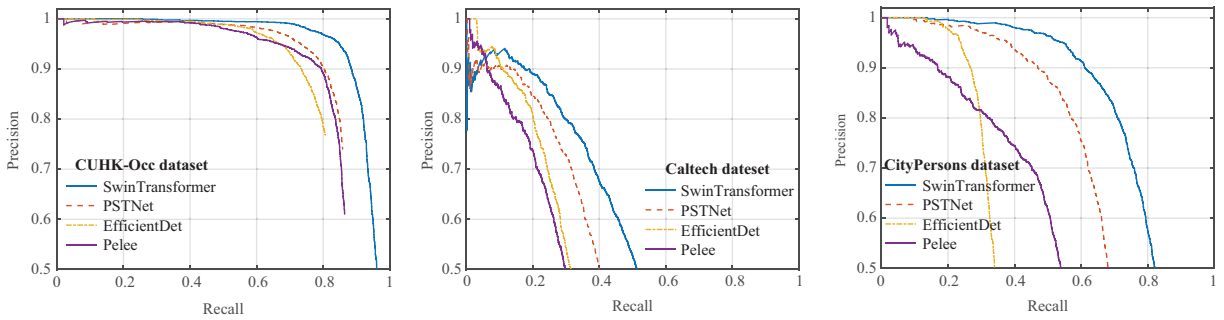


(a) CUHK-Occ dataset (b) Caltech dataset (c) CityPersons dataset

Figure 6: MR curves of PSTNet and baseline detectors on benchmark datasets: (a) CUHK-Occ dataset, (b) Caltech dataset, (c) CityPersons dataset

Fig. 7 shows the precision-recall curves for these detectors evaluated on the benchmark dataset, corresponding to the precision and recall ranges of $[0.5, 1.0]$ and $[0, 1.0]$, respectively. These curves represent the value pairs of precision and recall of detectors and highlight the ability of each detector to identify positive samples while minimizing false positives accurately. The higher curve indicates superior detection performance.

A comparison of these curves highlights the differences in performance between these detectors. Considering both computational efficiency and detection accuracy, the proposed PSTNet demonstrates an excellent balance in detection performance. PSTNet not only achieves higher precision and recall but also reduces the computational burden, ensuring that PSTNet can deliver robust detection results without compromising speed or resource utilization, making it an efficient solution for real-world applications.



(a) CUHK-Occ dataset (b) Caltech dataset (c) CityPersons dataset

Figure 7: The precision-recall curves of Swin Transformer, PSTNet, EfficientDet, and Pelee on the benchmark datasets: (a) CUHK-Occ dataset, (b) Caltech dataset, (c) CityPersons dataset

4.6 The SY-Metro Experiment

In pursuit of validating the PST algorithm's universal effectiveness and exploring its capacity to guide compact detection models in distinguishing pedestrians from background across various real-life scenarios. The PSTNet and baseline detectors are validated on the SY-Metro dataset, and the detailed experimental results are listed in Table 4, which reveal several insights:

Table 4: Experimental results of PSTNet and baseline detectors on SY-Metro datasets

Detectors	Params (M)	SY-Metro	
		MR (%)	GPU (ms/frame)
Swin Transformer	45.31	18.11	106
FPN	42.12	21.33	183
FRCNN VGG16	136.69	29.99	71
SSD512	23.75	26.16	51
Tiny YOLOV3	8.66	42.29	3
Pelee	5.29	32.25	16
EfficientDet	3.90	27.27	32
MetroNext	4.56	26.70	26
PSTNet	4.56	24.68	26

1) With the help of the PST algorithm, the PSTNet achieves competitive detection performance. Apart from the Swin Transformer and FPN, PSTNet boasts a notably lower miss rate of 24.68% with a few parameters and considerable inference time. This achievement further supports the PST algorithm's versatility in boosting the pedestrian detection accuracy of the plain detector. In addition, considering the limited hardware resources of the embedded system for deploying online passenger detectors in metro stations, the comprehensive performance of the PSTNet becomes a more viable option compared to other detectors.
2) Metro stations are settled scenes compared to complicated outdoor scenes in other benchmark datasets, and MetroNext has achieved a lower miss rate. On this basis, the use of the PST algorithm can effectively guide the network training process to form an accurate pedestrian classification capability, thus further improving the pedestrian detection accuracy of the plain detector (the MR value is reduced by up to 2%),

which demonstrates that the PST algorithm is better at steadily improving the plain detector detection accuracy in settled scenes compared to its performance gains in complicated outdoor scenes.

Fig. 8 draws the miss rate vs. false positives per image of all models on the SY-Metro dataset, which clearly illustrates the detection prowess of each model in identifying metro passengers. As shown in Fig. 8, our model achieves a competitive metro passenger detection ability compared to competitors.

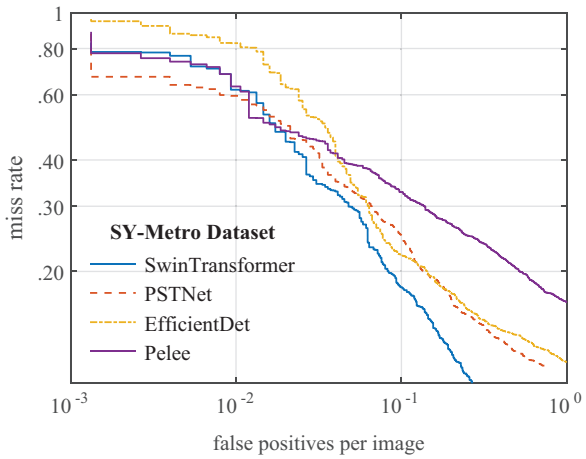


Figure 8: MR curves of PSTNet and baseline detectors on SY-Metro datasets

Fig. 9 compares the detection results of MetroNext and PSTNet on metro videos, demonstrating that PSTNet outperforms MetroNext in wiping out false positives.

4.7 The Experiment on the Embedded Development Board

The experiments on the workstation have shown that the proposed classifier has the potential to be deployed on the embedded platforms. To accurately estimate its inference speed on the embedded development board with limited hardware resources, this paper writes our model's forward inference program for metro passenger detection. Then, the inference speed of the PSTNet on the SY-Metro dataset will be tested. The specific experimental results are shown in Table 5, showing that:

1) Our model demonstrates a fast inference speed of 358 ms when processing an image on Jetson Nano, which means that it can quickly deliver the pedestrian detection results in the carriage to the metro video surveillance system. Due to the relatively little movement of passengers in metro carriages over a short period, Jetson Nano's processing speed is sufficient for video surveillance in this situation.
2) Aided by the stronger computing power of NPU, the inference speed of the proposed detector on the RK1808 AI computing stick is further improved, processing an image in only 120 ms, which can satisfy the time-sensitive task's requirements.

Figure 9: Comparing detection results of MetroNext and PSTNet on metro videos. "TPs" and "TPs+FPs" denote the True Positives and True Positives plus False Positives detection results, respectively. False Positives are drawn in red

Table 5: The inference speed of our model on Jetson Nano and RK1808 AI computing stick

PSTNet	Jeston Nano	RK1808
CPU (ms/frame)	358	120

4.8 Power Consumption Analysis Experiments

In this paper, the power consumption of PSTNet and its competitors is further analyzed to reflect the operation of these models on embedded devices, especially for battery-powered embedded devices. The applicability of the proposed PSTNet is validated in embedded environments to test whether it can better handle pedestrian detection tasks with high power consumption constraints. The experiments measure the

GPU power consumption of these networks in the inference stage with a time interval of 0.01 s, and the specific experimental results are shown in Table 6:

Table 6: The power usage of PSTNet and other competitors

Models	Energy consumption		
	Max (W)	Min (W)	Ave (W)
Swin Transformer	328	54	165
PSTNet	86	50	64
EfficientDet	82	55	63
Pelee	73	50	60

1) The average power consumption of PSTNet is 64 W, comparable to that of other small models such as EfficientDet and Pelee, and is much lower than Swin Transformer's power consumption of 165 W. This is because the PST algorithm guides the model to be effectively trained and improves its detection capability without increasing the computational cost. Therefore, PSTNet has better applicability for embedded platforms and can meet its demand for strict power consumption.
2) Transformer-based models like the Swin Transformer deployed on embedded devices can lead to increased energy consumption due to the computationally intensive matrix multiplications required by self-attention and feed-forward operations. This leads to more frequent processor use and memory accesses, making the operations computationally intensive and power-intensive. In addition, higher processor activity and memory bandwidth requirements lead to increased heat and latency, which brings a big challenge for energy-constrained devices.

5 Conclusion

In this paper, a well-designed Pedestrian Sensitive Training algorithm has been proposed to improve the pedestrian detection accuracy of the CNN-based detection method by removing FPs. Ablation experiments have shown that the proposed algorithm is feasible and practicable to enhance the detection performance of mainstream detection networks on the prevailing benchmark dataset. Combining the PST algorithm with MetroNext, the PSTNet is established and then validated on benchmark datasets. The experiment results have demonstrated that the PSTNet is a more competitive pedestrian detector. Besides, inference speed and power consumption experiments also support the fact that the PSTNet has a fast detection speed and lower power usage. In summary, the PST algorithm can effectively improve the detection accuracy of the model without adding extra computational burden, and the PSTNet is a practical pedestrian detector tailored explicitly for embedded vision applications.

Acknowledgement: Not applicable.

Funding Statement: Not applicable.

Author Contributions: Qiang Guo: Writing—original draft preparation, Carrying out the experiments, Methodology, Data curation. Rubo Zhang: Supervision, Reviewing and editing. Bingbing Zhang: Contributing to experiments. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The benchmark datasets used in this paper come from these papers [28,30,31]. The SY-Metro dataset are not available due to commercial restrictions.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Abbreviations

CNN Convolutional Neural Network
PST Pedestrian Sensitive Training
FPs False Positives

References

1. Chen W, Zhu Y, Tian Z, Zhang F, Yao M. Occlusion and multi-scale pedestrian detection A review. Array. 2023;19(2):100318. doi:10.1016/j.array.2023.100318.
2. Benenson R, Omran M, Hosang J, Schiele B. Ten years of pedestrian detection, what have we learned? In: Computer Vision-ECCV 2014 Workshops. Zurich, Switzerland; 2014 Sep 6–7 and 12. Cham, Switzerland: Springer; 2015. p. 613–27.
3. Cao J, Pang Y, Xie J, Khan FS, Shao L. From handcrafted to deep features for pedestrian detection: a survey. IEEE Trans Pattern Anal Mach Intell. 2022;44(9):4913–34. doi:10.1109/TPAMI.2021.3076733.
4. Liu Q, Guo Q, Wang W, Zhang Y, Kang Q. An automatic detection algorithm of metro passenger boarding and alighting based on deep learning and optical flow. IEEE Trans Instrum Meas. 2021;70:1–13. doi:10.1109/TIM.2021.3118090.
5. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Piscataway, NJ, USA: IEEE; 2005. Vol. 1, p. 886–93. doi:10.1109/CVPR.2005.177.
6. Dollár P, Tu Z, Perona P, Belongie SJ. Integral channel features. In: British Machine Vision Conference, BMVC 2009. London, UK; 2009 Sep 7–10. doi:10.5244/C.23.
7. Schwartz WR, Kembhavi A, Harwood D, Davis LS. Human detection using partial least squares analysis. In: 2009 IEEE 12th International Conference on Computer Vision. Piscataway, NJ, USA: IEEE; 2009. p. 24–31.
8. Bertozzi M, Broggi A, Lasagni A, Rose M. Infrared stereo vision-based pedestrian detection. In: IEEE Proceedings. Intelligent Vehicles Symposium. Piscataway, NJ, USA: IEEE; 2005. p. 24–9.
9. Nam W, Dollár P, Han JH. Local decorrelation for improved pedestrian detection. In: Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 1. Cambridge, MA, USA: MIT Press; 2014. p. 424–32.
10. Rehman Y, Khan JA, Riaz I, Shin H. Chunks: the remedy for notorious false alarms in pedestrian detection. In: 2016 International Conference on Electronics, Information, and Communications (ICEIC). Piscataway, NJ, USA: IEEE; 2016. p. 1–4.
11. Szarvas M, Yoshizawa A, Yamamoto M, Ogata J. Pedestrian detection with convolutional neural networks. In: IEEE Proceedings. Intelligent Vehicles Symposium. Piscataway, NJ, USA: IEEE; 2005. p. 224–9.
12. Huang L, Wang Z, Fu X. Pedestrian detection using retinanet with multi-branch structure and double pooling attention mechanism. Multimed Tools Appl. 2024;93(2):6051–75. doi:10.1007/s11042-023-15862-4.
13. Althoupety A, Wang LY, Feng WC, Rekabdar B. Daff: dual attentive feature fusion for multispectral pedestrian detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2024; Piscataway, NJ, USA. p. 2997–3006.
14. Liu S, Cao L, Li Y. Lightweight pedestrian detection network for UAV remote sensing images based on strideless pooling. Remote Sens. 2024;16(13):2331. doi:10.3390/rs16132331.
15. Xu H, Huang S, Yang Y, Chen X, Hu S. Deep learning-based pedestrian detection using rgb images and sparse lidar point clouds. IEEE Trans Ind Inform. 2024;20(5):7149–61. doi:10.1109/TII.2024.3353845.
16. Gomez-Donoso F, Cruz E, Cazorla M, Worrall S, Nebot E. Using a 3D CNN for rejecting false positives on pedestrian detection. In: 2020 International Joint Conference on Neural Networks (IJCNN). Piscataway, NJ, USA: IEEE; 2020. p. 1–6.

17. Iftikhar S, Asim M, Zhang Z, El-Latif AAA. Advance generalization technique through 3D CNN to overcome the false positives pedestrian in autonomous vehicles. Telecommun Syst. 2022;80(4):545–57. doi:10.1007/s11235-022-00930-1.
18. Aung S, Park H, Jung H, Cho J. Enhancing multi-view pedestrian detection through generalized 3D feature pulling. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. Piscataway, NJ, USA: IEEE; 2024. p. 1196–1205.
19. Kolluri J, Das R. Intelligent multimodal pedestrian detection using hybrid metaheuristic optimization with deep learning model. Image Vis Comput. 2023;131(9):104628. doi:10.1016/j.imavis.2023.104628.
20. Alfred Daniel J, Chandru Vignesh C, Muthu BA, Senthil Kumar R, Sivaparthipan C, Marin CEM. Fully convolutional neural networks for lidar-camera fusion for pedestrian detection in autonomous vehicle. Multimed Tools Appl. 2023;82(16):25107–30. doi:10.1007/s11042-023-14417-x.
21. Liu S, Huang D, Wang Y. Adaptive nms: refining pedestrian detection in a crowd. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ, USA; 2019. p. 6452–61.
22. Huang X, Ge Z, Jie Z, Yoshie O. Nms by representative region: towards crowded pedestrian detection by proposal pairing. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ, USA; 2020. p. 10747–56.
23. Zhang J, Lin L, Zhu J, Li Y, Chen Y-C, Hu Y, Hoi SCH. Attribute-aware pedestrian detection in a crowd. IEEE Trans Multimedia. 2020;23:3085–97. doi:10.1109/TMM.2020.3020691.
24. Yu R, Wang S, Lu Y, Di H, Zhang L, Lu L. Saf: semantic attention fusion mechanism for pedestrian detection. In: PRICAI 2019: Trends in Artificial Intelligence. Cham: Springer International Publishing; 2019. p. 523–33.
25. Jiao Y, Yao H, Xu C. Pen: pose-embedding network for pedestrian detection. IEEE Trans Circuits Syst Video Technol. 2020;31(3):1150–62. doi:10.1109/TCSVT.2020.3000223.
26. Tang Y, Liu M, Li B, Wang Y, Ouyang W. OTP-NMS: towards optimal threshold prediction of NMS for crowded pedestrian detection. IEEE Trans Image Process. 2023;32:3176–87.
27. Luo Y, Xiao L. G-RCN: optimizing the gap between classification and localization tasks for object detection. arXiv:2012.03677. 2020.
28. Dollar P, Wojek C, Schiele B, Perona P. Pedestrian detection: an evaluation of the state of the art. IEEE Trans Pattern Anal Mach Intell. 2011;34(4):743–61. doi:10.1109/TPAMI.2011.155.
29. Gabbasov R, Paringer R. Influence of the receptive field size on accuracy and performance of a convolutional neural network. In: 2020 International Conference on Information Technology and Nanotechnology (ITNT). Samara, Russia; 2020. p. 1–4.
30. Ouyang W, Wang X. A discriminative deep model for pedestrian detection with occlusion handling. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. Providence, RI, USA; 2012. p. 3258–65.
31. Zhang S, Benenson R, Schiele B. CityPersons: a diverse dataset for pedestrian detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA; 2017. p. 3213–21.
32. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. Cambridge, MA, USA: MIT Press; 2015. Vol. 28.
33. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: single shot multibox detector. In: Leibe B, Matas J, Sebe N, Welling M, editors. Computer Vision—ECCV 2016. Cham, Switzerland: Springer International Publishing; 2016. p. 21–37.
34. Redmon J, Farhadi A. YOLOv3: an incremental improvement. arXiv:1804.02767. 2018.
35. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA; 2017. p. 936–44.
36. Wang RJ, Li X, Ling CX. Pelee: a real-time object detection system on mobile devices. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, ser. NIPS'18. Montréal, QC, Canada; 2018. p. 1967–76.

37. Tan M, Pang R, Le QV. EfficientDet: scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, WA, USA; 2020. p. 10781–90.
38. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z,et al. Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Montreal, QC, Canada; 2021. p. 10012–22.