Research Article

Multiscale Anchor-Free Region Proposal Network for Pedestrian Detection

Zhiwei Cao,1 Huihua Yang,1 Weijin Xu,1 Juan Zhao,2 Lingqiao Li,3 and Xipeng Pan3

1School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China
2China Mobile Research Institute, Beijing 100053, China
3School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China

Correspondence should be addressed to Huihua Yang; yhh@bupt.edu.cn

Received 27 February 2021; Revised 19 March 2021; Accepted 5 April 2021; Published 26 April 2021

Academic Editor: Yingjie Wang

Copyright © 2021 Zhiwei Cao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Pedestrian detection based on visual sensors has made significant progress, in which region proposal is the key step. There are two mainstream methods to generate region proposals: anchor-based and anchor-free. However, anchor-based methods need more hyperparameters related to anchors for training compared with anchor-free methods. In this paper, we propose a novel multiscale anchor-free (MSAF) region proposal network to obtain proposals, especially for small-scale pedestrians. It usually has several branches to predict proposals and assigns ground truth according to the height of pedestrian. Each branch consists of two components: one is feature extraction, and the other is detection head. Adapted channel feature fusion (ACFF) is proposed to select features at different levels of the backbone to effectively extract features. The detection head is used to predict the pedestrian center location, center offsets, and height to get bounding boxes. With our classifier, the detection performance can be further improved, especially for small-scale pedestrians. The experiments on the Caltech and CityPersons demonstrate that the MSAF can significantly boost the pedestrian detection performance and the log-average miss rate (MR) on the reasonable setting is 3.97% and 9.5%, respectively. If proposals are reclassified with our classifier, MR is 3.38% and 8.4%. The detection performance can be further improved, especially for small-scale pedestrians.

1. Introduction

Pedestrian detection played an important role in self-driving vehicle tasks by assisting drivers to judge whether there are pedestrians in the front of the driving area. Therefore, pedestrian detection performance directly affects pedestrian safety [1–4]. In recent years, with the research and development of convolutional neural networks (CNN), pedestrian detection methods based on CNN have shown rapid progress. According to the regression starting status, pedestrian detection can be divided into anchor-based [5–9] and anchor-free [10–16] detection methods.

Based on the number of the detection stages, anchor-based methods can be divided into two-stage [9, 17–20] and one-stage [6, 21, 22] detection methods. In two-stage detection, region proposals are generated firstly, and then, the proposals are classified by a classifier. In one-stage detection, the final detection results can be obtained via only one step; pedestrian detection can skip the classification stage and predict bounding boxes with confidence scores directly.

The most impressive anchor-based method is the region proposal network (RPN) [7], which was first proposed in Fast-RCNN [7]. The regression starting status of RPN is predefined by a set of anchor boxes with multiple scales and ratios, and then, the anchors are transformed according to the learning parameters into proposals. Multiscale anchors can avoid the problem of scale imbalance [23] caused by the width and height range of ground truth. Although RPN can achieve excellent performance, it needs to design anchor boxes manually, which will affect the generalization ability of the model.

Different from anchor-based methods, anchor-free start regression from a point and do not require hyperparameters
about anchors. At the same time, with the help of focal loss [24] to solve the problem of the imbalance between positive and negative samples in training, many remarkable anchor-free detection methods have been developed, such as CSPNet [12], FCOS [14], and FSAF [25]. Among these methods is CSPNet, a single-scale anchor-free detector which is efficient on pedestrian detection datasets. However, as Figure 1 presents the height and the area of the pedestrians in the Caltech and CityPersons datasets, we observe most of the pedestrians in the dataset are small, which leads to scale imbalance [23]. CSPNet is insufficient for handling with scale imbalance, because it is a single-scale detection head and only concatenation is used to fuse these multiscale feature maps on different stages.

Inspired by RPN and CSPNet, we design a multiscale anchor-free detection head (MSAF) on adaptive channel feature fusion (ACFF) to generate proposals at different scales of features. The deeper the network is, the more difficult it is to detect small pedestrians. As we all know, different feature layers have different receptive field sizes. If multiscale regression is trained on the same feature layer, the size of ground truth and actual receptive fields do not match. Fortunately, within the feature extract module, the predict boxes do not necessarily need to correspond to the actual receptive fields of each layer. We design the multiscale detection head so that specific feature maps learn to be responsive to the particular scale of the pedestrians.

The main contributions of this work are summarized as follows: first, we propose an effective approach, named adapted channel feature fusion, to extract channel features at different levels so that only useful channel features are kept for fusion. Second, we propose a multiscale anchor-free method to replace the anchor-based method. It is used to reduce the hyperparameters that exist in anchor-based and address the scale imbalance problem. Third, a RCNN classifier is proposed to further improve the detection performance, especially small-scale pedestrian detection. Fourth, our detection method achieves state-of-the-art performance on the Caltech database [26] and competitive performance on the CityPersons [27] pedestrian benchmark.

2. Related Work

In this section, we mainly introduce anchor-based and anchor-free pedestrian detection methods based on the feature extraction and detection head. In the step of pedestrian classification, the description of how to select backbone and design classifier to address various problems is also highlighted.

2.1. Anchor-Based Methods. Anchor-based methods need a set of predefined anchors with different scales and ratios for regression training, and then, the anchors are transformed according to the training parameters into proposals. In two-stage pedestrian detection, generating high-quality proposal boxes is the first key step; then region proposals are classified by a classifier. The most representative method is RPN which is first introduced in Fast-RCNN [7]. RPN [19] takes a smooth L1 loss for regression training and is implemented on the final high-level feature layers. MS-CNN [17] sets RPN modules on different level layers of backbone to pay more attention to small object detection. FPN [9] uses the top-down feature fusion method to build a feature pyramid and generate bounding boxes with RPN on different levels. We can also find that RPN based on feature fusion can significantly improve the detection performance. SSD-RCNN [28] applies semantic segmentation to RPN and RCNN to boost pedestrian detection accuracy. SSA-CNN [29] proposes a self-attention mechanism to connect the RPN and RCNN stages to improve pedestrian detection performance. AR-Ped [5] utilizes a stackable decoder-encoder module consisting of top-down and bottom-up pathways for feature fusion to improve the precision of the RPN stage. Repulsion [30] and aggregation [31] loss are designed on the RPN to tackle occluded pedestrians in crowded scenes.

In one-stage pedestrian detection, bounding boxes are predicted with only one step. SSD [6] predicts the detect results at different levels of features with a prior anchor. YOLOv3 [22] and YOLOv4 [21] predict the object on three different scale branches, and feature fusion architecture like FPN is used to detect small-scale objects. RetinaNet [24] also takes the feature fusion architecture like FPN to object detection, and focal loss is proposed to address the foreground-background class imbalance.

2.2. Anchor-Free Methods. There are two ways to find objects in anchor-free detection. The first way is to use the center point or region of the pedestrian to predict the length from the bounding box boundary. YOLOv1 [13] predicts pedestrians on the final layer of backbone and detect objects in a grid cell if the center of pedestrian falls into. UnitBox [32] takes Intersection over Union (IoU) loss as detection head to predict proposals and avoid the box-level scale imbalance and optimizes the L2 loss in DenseBox [33]. CSPNet [12] extracts multiscale features with concatenation on different stages, and pedestrian detection is simplified as a straightforward center and scale prediction task through convolutions. Wang [15] appends some adaptations on CSPNet to improve the robustness of the method. CSID [16] proposes a pedestrian detector with a novel identity-and-density-aware non-maximum suppression (NMS) algorithm to refine the detection results. FCOS [14] selects FPN as feature fusion architecture and defines the inside of bounding box as positive.

The second way is to predict key points on heatmaps as detection head. CornerNet [11] takes an hourglass network for multiscale feature fusion and detects a pedestrian as a pair of key points on heat maps. CenterNet [10] adapts CornerNet’s detection head as a triplet of keypoints to predict bounding boxes. ExtremeNet [34] also uses an hourglass network to extract features and predict four extreme points and one center point for each pedestrian.

2.3. Classifier. In the step of pedestrian classification, different types of classifiers have been designed to address various problems. In order to get better classification accuracy, we design different convolutional neural network architectures according to the application scenarios, such as MobileNet
[28], VGGNet [36], GoolgeNet [36], ResNet [37], and DenseNet [38]. In order to ensure that for any size of input regions, it can always produce the same size region features, RoIPooling [19] and RoIAlign [39] are designed, and the shared features are directly classified according to the Region of Interest (RoI).

To detect the small objects, RPN+BF [20] uses the cascaded Boosted Forest for pedestrian classification to mine hard negative examples and handle the small number of instances. In another approach, scale-aware [8] weights are predicted and a large-scale subnetwork and a small-scale subnetwork are combined into a unified framework to solve the multiscale pedestrian classification problem and achieve state-of-the-art performance on the Caltech dataset. In order to improve detection performance with feature fusion, BCN [28] combines semantic segmentation and classification together to perform pedestrian classification. SA-RCNN [29] uses self-attention to perform feature extraction for pedestrian classification and achieves good results. A previous study [40] designs a hyperlearner, which is a new type of feature fusion framework, to extract features, and uses additional pedestrian features to improve the detection performance. To handle the occlusion problem in pedestrian detection, a new partial occlusion-aware pooling unit [31] is used in classification. To address the IoU distribution imbalance, in Cascade R-CNN [41], several RCNN networks are cascaded based on different IoU thresholds, and the detection results are continuously optimized to improve the detection performance.

3. Baseline Method

CSPNet [12] is a single-scale anchor-free pedestrian detector. It can directly obtain the detection results by predicting the center location, the height of bounding boxes, and center offsets with single scale. The architecture consists of two modules: the feature extraction and the detection head.

The feature extraction module uses CNN to extract feature maps for pedestrian detection. In this paper, ResNet-50 is used as the backbone of feature extraction; its convolution layers can be divided into five stages according to the pooling stride. Given an input image of size $H \times W$, the feature resolution of stage $i$ is $(H/2^i) \times (W/2^i)$. The experiment results show that the best detection performance can be obtained by deconvolution the features of stages 3, 4, and 5 into the same resolution $(H/4) \times (W/4)$ before concatenation.

The detector head is used to generate the bounding box, which contains three branches: the first branch is to predict the classification score of the bounding box and determine the center location of the proposal. The second branch is used to predict the height of the bounding box and then use the aspect ratio to get the bounding box. The third branch is used to predict the center offsets of the bounding box and adjust the center location. The center offsets are defined as $(\Delta x, \Delta y) = (|x_k|/r - |x_t|/r, |y_k|/r - |y_t|/r)$. The details are provided in Figure 2(b). In the training process, the modified focal loss is used as the loss function in the classification task, and smooth L1 is used as the regression loss function in the height and center offsets prediction task.

4. Our Approach

In this section, we introduce an adaptive channel feature fusion method to extract channel features at different levels and propose a multiscale anchor-free region proposal network to generate proposals. The proposed network has fewer hyperparameters than anchor-based methods and can significantly boost detection performance, especially for small objects.

4.1. Adapted Channel Feature Fusion for Feature Extraction

In CSPNet, only concatenation is used to fuse the features on different levels. Currently, the common feature fusion methods are element-wise sum (SUM) or concatenation. As
we all know, different feature layers have distinct abilities, and the low-level feature maps can provide more precise localization information while the high-level maps contain more semantic information. Therefore, here, we introduce ACFF, which can not only adaptively select channel features on different scales for fusion but also boost the feature discrimination. The detail of ACFF can be found in Figure 3.

Two steps are needed to implement the ACFF. In the first step, the features of different levels are scaled to the same resolution and then concatenated on the channel dimension. If you want to get the fused feature of the $i$-th level, you need to scale the features of the other two adjacent different levels to the same resolution as the $j$-th level and then get the concatenated feature maps, because the features at three

**Figure 2:** An illustration and comparison of different methods for generating proposals. Our MSAF is used to extract features based on ACFF and predict bounding boxes on different scales; after that, NMS is applied to generate final proposals with IoU threshold of 0.5.
levels in detect backbone have different resolutions as well as different numbers of channels. The concatenated feature maps can be presented as

\[
X = \text{Concat}(X^{i-1}, X^i, X^{i+1}),
\]

where \(X^i\) is defined as the feature of the \(i\)-th level and \(X^i \in \mathbb{R}^{(H/r) \times (W/r) \times C} \), \(r\) is stride, and \(r = 2^i\).

For example, let us assume that channel feature fusion is performed at the fourth level. If the resolution of the feature to be fused is smaller than that of the target feature, deconvolution is used to enlarge the feature, and then \(1 \times 1\) convolution layer is used to compress the channel to 256. If the resolution of the feature to be fused is greater than that of the target feature, we use a \(3 \times 3\) convolution layer with a step size of 2 to reduce the feature resolution and channel dimension.

In the second step, the global average pooling (GAP) is used to generate channel feature vectors \(F^1\). The \(c\)-th channel element from GAP is calculated by the following formula:

\[
F^1_c = \text{GAP}(X_c).
\]

Then, a new compact feature \(F^2\) is created to adaptively learn the fusion weights of different level features. This is achieved by a fully connected (FC) layer with the lower dimension:

\[
z = F^2 = FC(F^1),
\]

where \(F^1 \in \mathbb{R}^C\), \(F^2 \in \mathbb{R}^{C'}\), and \(C' = \max(C/r, L)\) is a typical setting in our experiment.

Further, softmax is used for normalization, and the learned weights \(\alpha_c, \beta_c\), and \(\gamma_c\) are used to select the corresponding level features for final fusion \(F_c\). Note that \(\alpha_c, \beta_c,\) and \(\gamma_c\) is simple a scale value at channel \(c\) and \(\alpha_c, \beta_c, \gamma_c \in [0, 1]\).

\[
\begin{align*}
\alpha_c &= \frac{e^{\lambda_c}}{e^{\lambda_c} + e^{\beta_c} + e^{\gamma_c}}, \\
F_c &= \alpha_c \cdot X^{i-1}_c + \beta_c \cdot X^i_c + \gamma_c \cdot X^{i+1}_c,
\end{align*}
\]

where \(F_c \in \mathbb{R}^{(H/r) \times (W/r)}\), \(\alpha_c + \beta_c + \gamma_c = 1\), and \(A, B, C \in \mathbb{R}^{C \times C'}\).

4.2. Multiscale Anchor-Free Detection Head. Scale imbalance occurs in the pedestrian dataset because certain sizes of the objects or input bounding boxes are overrepresented [23]. Taking the Caltech [26] and CityPersons [27] datasets as examples, the height of the pedestrians ranges from 30 px to 350 px and 35 px to 965 px; the distribution of the pedestrian heights at different scales is imbalance. Approximately 80% of the pedestrians in the Caltech and 64% of the pedestrians in the CityPersons are less than 112 px in height. The detailed statistical information is provided in Figure 1. The scale imbalance problem suggests that a single scale of visual processing is not sufficient for detecting objects at different scales. If single-scale regression is used to predict the height, the constraint range is too large and may cause a deviation in the prediction results, like CSPNet in Figure 2(d). Two popular approaches have been used for multiscale predictions. The first approach is the prediction on the same layer. For example, in RPN [19], as shown in Figure 2(a), the detection head is on the end of the backbone with different aspect ratios and scales to train. The second approach is to predict on different level layers at multiple scales as shown in Figures 2(b) and 2(c). The detection head is different for each feature layer. For example, in MS-CNN [17], FPN [9], and FCOS [14], detection head is attached to each level on the feature pyramid to obtain proposals.

To address shortcoming of scale imbalance, we propose MSAF to assign the ground truth in different scale-spaces for forward and backward as shown in Figure 2(e). The MSAF is based on a convolutional network that produces bounding boxes with scores followed by an NMS step to produce the final detection. Three branches are attached to the final feature maps \(F\) to predict the pedestrian location, height, and center offsets at each scale. The width can be calculated with an aspect ratio and height. According to the statistics of pedestrian detection bounding box annotations [26, 27], the aspect ratio is generally set to 0.41. In this detection head, we attach a \(3 \times 3\) convolution layer on the fusion

**Figure 3:** Illustration of adapted channel feature fusion. The first stage of ACFF is to concatenate features with equivalent scales along channel dimension. Then, the second stage uses learning weights to aggregate features in an adaptive way.
feature $F$, and then three head map layers are appended to predict location, height and offset with $1 \times 1$ convolution kernel.

During training, we must determine how to assign the ground truth to the corresponding scale. To handle different object scales, we refer to the formula in FPN [9]. If the ground truth height is $h$, we assign it to the scale $k$ according to

$$k = k_0 + \log_2 \left( \frac{h}{224} \right), \quad k_0 = 4.$$  

Taking the Caltech as example, the ground truth bounding boxes are assigned into three scales: 56~112 px, 112~224 px, and 224 px~. An illustration example of assigning pedestrians to different levels according to different scales of pedestrians is depicted in Figure 4. The red ground truth is assigned to the low-level features, and the green ground truth is assigned to the high-level features for prediction.

To predict the center location, we define the positive region, ignore region, and negative region for center ground truth. If the point falls into the center region of the pedestrian, it is assigned to positive samples for training. The center region is the area with a pedestrian center as the center within a radius of 2. The ignore region is the location where the ground truth bounding box is not assigned in this scale $k$ and ground truth bounding box excluding positive samples. If the positive and ignore regions are excluded in the image, the rest are negative regions. The whole illustration can be found in Figure 4.

The modified focal loss is used as the loss function to predict center location as follows:

$$L_{loc} = -\frac{1}{P} \sum_{i=1}^{W} \sum_{j=1}^{H} a_{ij} \left(1 - p_{ij}\right)^\gamma \log \left(\hat{p}_{ij}\right),$$

$$a_{ij} = \begin{cases} 1, & y_{ij} = 1, \\ \left(1 - M_{ij}\right)^\beta, & \text{otherwise}. \end{cases}$$

In the above, $P$ is the number of positive samples. $\gamma$ and $\beta$ are the focusing hyperparameters, and we experimentally set $\gamma = 2$ and $\beta = 4$ as suggested in [12]. $M_{ij}$ is a 2D Gaussian mask centered at the location $(i,j)$, and the mask is proportional to the height and width of the individual objects. If $y_{ij}$ is equal to 1, $\hat{p}_{ij}$ is set to $p_{ij}$, and $\hat{p}_{ij}$ is set to $1 - p_{ij}$ otherwise.

To predict pedestrian height and center offsets, the pedestrian height in scale $k$ is redesigned as $h_k = \log \left(\frac{(h - 28(k - 2))}{(7 \times 2^{k - 1})}\right)$. We select the smooth L1 loss for height and offset prediction:

$$L_{box} = \frac{1}{P} \sum_{i=1}^{P} \text{Smooth L1} \left(b, \hat{b}\right),$$

where $b = (\Delta x, \Delta y, h_k)$ and $\hat{b} = (\Delta x, \Delta y, \hat{h}_k)$ represent the offset and height from ground truth and prediction of positive samples, respectively.

To sum up, the optimization objective int scale $k$ is

$$L_k = \alpha_{k1} L_{loc} + \alpha_{k2} L_{box}.$$
4.3. RCNN Classifier. The RCNN classifier is used to classify the proposals generated by the MSAF as pedestrian or non-pedestrian. We take the object classifiers from [5, 20, 28, 29] as references to construct the RCNN classifier. We resize the object to a fixed resolution and then use it as the input of the classifier to determine whether it is a pedestrian based on the final score. As shown in Figure 1, the height of most pedestrians is less than 112 px. If the image is resized to 224 × 224 px as the input, the information of the image will be distorted, degrading the classification performance. To alleviate this problem, we cropped the object from the image, added 25% padding, and resized it to 112 × 112 px as the input. VGG-16 [32] without the pool5 layer is chosen as the backbone since the size of the receptive field of VGG-16 is the same as that of the pedestrian. ACFF is used to extract features to improve the discrimination ability of the model; the detail can be found in Section 4.1.

5. Experiments

In this section, we first introduce the implementation details, evaluation metrics, and dataset information. Then, the ablation studies about the MSAR and ACFF are reported. Finally, we also give a detailed description of the benchmark comparison experiments.

5.1. Training, Inference, and Implementation Details. The whole detect framework is implemented on Keras. In MSAF stage, the ResNet-50 is used as backbone to predict the bounding boxes. Specifically, our MSAF is trained using the adaptive moment estimation (Adam) algorithm for 100 K iterations with an initial learning rate of 0.0001 and a learning policy for the steps. Each minibatch is constructed from an N = 16 image. It is trained with multiscale input image in the scale between 0.6 and 1.5. Whole image is taken as input to predict height, offset, and locations. We first select bounding boxes with score above 0.01 and then use NMS with threshold 0.5 for final processing. In RCNN stage, VGG-16 is selected as backbone and is trained with the stochastic gradient descent (SGD) algorithm for 120 K iterations with an initial learning rate of 0.001 and a learning policy for the steps. After 60 K iterations, the learning rate is set to 0.0001. The weight decay and momentum are set as 0.0005 and 0.9, respectively. No more than 20 proposals are selected from MSAF for each SGD minimatch, and these are selected according to the scores in a descending order. To carry out the experiments, an Intel Xeon E5-2620 @ 2.1 GHz CPU server with 48 GB of memory and two TITAN RTX (24 GB) GPUs are used.

5.2. Evaluation Metrics. To evaluate MSAF, two benchmark datasets, Caltech [26] and CityPersons [27], are selected for the experiment and comparison. The log-average miss rate over False Positive Per Image (FPII) ranging in [10⁻², 10⁰] (denoted as MR⁻²) [26] is used to evaluate the pedestrian detection performance. A lower miss rate indicates better detection performance. The evaluation settings are from Caltech and CityPersons, respectively. Generally speaking, we need to focus on the height in the reasonable setting is greater than 50 pixels and in the all setting is greater than 20 pixels.

5.2.1. Caltech. The Caltech pedestrian dataset [26] consists of approximately 10 hours of 640 × 480 30 Hz video taken from a vehicle driving through regular traffic in an urban environment. Approximately 250,000 frames with a total of 350,000 bounding boxes and 2300 unique pedestrians were annotated. We extract one out of every 4 frames from the raw videos (acquiring a total of 42782 images) to form the training set and one frame out of every 30 frames from the raw videos (acquiring a total of 4024 images) to form the test

Table 1: Comparison of different feature fusion methods on the Caltech with only single-scale detect head.

| Method       | VGG-16 IoU = 0.5 | VGG-16 IoU = 0.75 | ResNet-50 IoU = 0.5 | ResNet-50 IoU = 0.75 |
|--------------|------------------|------------------|---------------------|---------------------|
| Concat       | 4.54             | 25.76            | 4.66                | 30.87               |
| SUM          | 4.66             | 30.87            | 4.66                | 30.87               |
| ACFF         | 4.13             | 26.23            | 4.13                | 26.23               |

Table 2: Comparison of different region proposal methods on the Caltech dataset under the reasonable and all setting.

| Method          | Reasonable | All | Reasonable | All |
|-----------------|------------|-----|------------|-----|
| RPN [20]        | —          | —   | 10.67      | 64.22 |
| MS-CNN [17]     | —          | —   | 9.47       | 63.59 |
| SDS-RPN [28]    | —          | —   | 8.17       | 61.29 |
| SSA-RPN [29]    | —          | —   | 8.30       | 61.77 |
| AR-RPN [5]      | —          | —   | 7.16       | 59.98 |
| CSPNet [12]     | 4.54       | 56.94| 5.29       | 60.06 |
| FPN [9]         | 4.32       | 56.16| 5.09       | 59.98 |
| MSAF (ours)     | 3.97       | 55.93| 4.91       | 58.86 |

The final objective loss function is a multitask loss in different scales to be optimized as follows:

\[ L = \sum_{k=1}^{C_1} L_k, \] (9)

where \( C_1 \) corresponds to the max scale of pedestrian height in function (5).

Table 3: Comparisons on different RCNN classifiers with new annotations under the reasonable setting.

| Method          | VGG-16 | ResNet-50 |
|-----------------|--------|-----------|
| RPN-BF [20]     | 4.26   | 4.66      |
| SDS-RCNN [28]   | 3.84   | 4.08      |
| AR-Ped [5]      | 3.96   | 4.09      |
| RCNN            | 3.47   | 3.63      |
| RCNN+ACFF (ours)| 3.38   | 3.58      |

5.3. CSPNet

CSPNet [12] is used as the backbone for the RCNN classifier.

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Proc. 25th Int. Conf. Mach. Learn., 2012, pp. 1–8.

[2] J. Zhang, Z. Zhang, Y. Sun, and X. Tang, “Deep learning face representation with the combined multipleinstance and multi-view objectives,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 3116–3124.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2009, pp. 248–255.
set. The new annotations [42] from the Caltech dataset are used on the experiments.

5.2.2. CityPersons. The CityPersons [27] is built upon Cityscapes dataset. It is a large and diverse set of stereo video sequences and is collected in urban street scenes. The dataset contains a total of 5000 images, and the resolution is 2048 × 1024, more than 35 k person and 13 k ignore regions. The split of train, validation, and test subsets is the same as that of Cityscapes. The training subset has 2975 images and was recorded across 18 different cities in three different seasons and various weather conditions. The validation subset was created from 3 different cities and has 500 images. The test subset was collected from 6 different cities and has 1575 images.

5.3. Ablation Study. In this section, we conduct ablation experiments on Caltech to evaluate the performance of each component of the proposed method. For the proposals, we focus on analyzing the impact of the MSAF and ACFF. For the classifier, we evaluate the impact of classification on the overall performance.

5.3.1. ACFF for Region Proposals. To assess the importance of ACFF, we compare it with other feature fusion methods: SUM and concatenation [43]. ResNet-50 is taken as the backbone in these experiments. The three feature fusion methods contain the same detector head from CSPNet [12].

It can be observed from Table 1 that ACFF has the best performance when it is used to feature fusion and the performance of feature fusion using SUM and Concat is similar. ACFF can adaptively select different levels of channel features for fusion, which can improve the performance of pedestrian detection. If ResNet-50 is taken as the backbone, MR$^{-2}$ of ACFF on the Caltech is 0.41% higher than concatenation and 0.53 higher than SUM when IoU is 0.5.

5.3.2. Importance of MSAF. To highlight the excellent performance of MSAF, it is compared with RPN [19], SDS-RPN [28], SSA-RPN [29], AR-RPN [5], and CSPNet [12] on the Caltech dataset under the reasonable and all setting. The detailed results are given in Table 2.

Compared with other methods, MSAF is state-of-the-art as shown in Table 2, the MR$^{-2}$ is 3.97% under the reasonable setting, and the MR$^{-2}$ is 55.93% under the all setting when the backbone is ResNet-50. MSAF also gets the best performance when the backbone is VGG-16 and the MR$^{-2}$ is 4.91% under the reasonable setting and the MR$^{-2}$ is 58.86% under the all setting. Through the experimental comparison on two different backbones, we find that the region proposal methods on ResNet-50 are better than those on VGG-16. Compared with methods CSPNet, FPN (the same detection head as MSAF), and MSAF, it can be observed that the effect of multiscale is better than that of single scale. Multiscale regression can effectively improve the detection performance.

5.3.3. Importance of RCNN Classifier. To evaluate the influence of our classifier on the detection performance, MSAF is used to extract the proposals as the inputs and our RCNN classifier is compared with other classifiers from RPN+BF [20], SDS-RCNN [28], and AR-Ped [5]. The comparison experiments are performed on the Caltech dataset with different resolutions, and the results are given in Table 3.
Figure 6: Comparison with the state-of-the-arts on the Caltech using new annotations. (a) is the results for the experiment performed on the reasonable setting; (b) shows the results of the experiment performed on the all setting.
5.3.4. Small Object Detection. In order to further illustrate the effectiveness of our method MSAF in small object detection, we make a comparison with CSPNet, CSPNet-RCNN, and MSAF-RCNN at different pedestrian heights in Figure 5. As shown in the figure, small-scale pedestrian detection is difficult; the higher the pedestrian’s height is, the better the detection effect is. The performance improvement in CSPNet and MSAF between 30 and 50 pixels is about 5% and 0.5% between 30 and 50 pixels. When the height is between 70 and 90 pixels, the improvement is not obvious. The improvement gap on small object detection using our RCNN classifier is large when the height between 30 and 50 pixels, about 11% improvement over CSPNet-RCNN and 7% improvement over MSAF.

5.4. Benchmark Comparison

From Table 3, compared with the other methods, our method has the highest detection accuracy and better robustness. We observe that the MR-2 in our RCNN classifier is 3.38% when the resolution is 112 × 112 px. The detection performance using 112 × 112 px as input is better than that using 224 × 224 px as input when VGG-16 is taken as backbone on the Caltech. If ACFF is used for feature fusion, the classification effect can be further improved from the method RCNN and RCNN+ACFF. In addition, we find that two-stage detection can significantly improve the performance with our classifier compared with one-stage detection.

| Method                  | Backbone       | Reasonable (%) | Heavy (%) | Partial (%) | Bare (%) | Small (%) | Medium (%) | Large (%) |
|-------------------------|----------------|----------------|-----------|-------------|----------|-----------|------------|-----------|
| Faster-RCNN [19]        | VGG-16         | 15.4           | —         | —           | 25.6     | 7.2       | 7.9        |
| RetinaNet [30]          | ResNet-50      | 15.6           | 49.98     | —           | —        | —         | —          |
| CornerNet [30]          | Hourglass-54   | 21.0           | 56.0      | —           | —        | —         | —          |
| Repulsion Loss [30]     | ResNet-50      | 13.2           | 56.9      | 16.8        | 7.6      | —         | —          |
| CSPNet [12]             | ResNet-50      | 11.0           | 56.9      | 10.4        | 7.3      | 16.0      | 3.7        | 6.5       |
| ACSP [15]               | ResNet-50      | 9.3            | 46.3      | 8.7         | 5.6      | —         | —          |
| CSID [16]               | ResNet-50      | 8.8            | 46.6      | 8.3         | 5.8      | —         | —          |
| MSAF (ours)             | ResNet-50      | 9.5            | 48.4      | 9.3         | 6.2      | 15.5      | 3.5        | 6.2       |
| MSAF-RCNN (ours)        | ResNet-50      | 8.4            | 46.9      | 8.6         | 5.5      | 15.1      | 3.3        | 6.4       |

The authors declare that they have no conflicts of interest.

6. Conclusions

To improve the pedestrian detection performance, a multiscale anchor-free region proposal network is proposed in this paper. ACFF is used to extract features firstly, and then MSAF detector head is used for training according to the height of pedestrians. Through experimental comparisons, we know that multiscale detection is easier to detect small-scale pedestrians than single-scale detection. In addition, the RCNN classifier is taken for further improvement. Compared with other detection methods, we find that the performance of two-stage detection is significantly better than that of one-stage detection. Overall, our detection method achieved state-of-the-art performance on Caltech with new annotations and obtains competitive performance on CityPersons.

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study. Requests for data, please send email to corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.
Acknowledgments

This research was supported in part by the National Key R&D Program (Grant No. 2018AAA0102600), the National Natural Science Foundation of China (Grant Nos. 62002082, 61866009, and 61906050), and Guangxi Natural Science Foundation (Grant Nos. 2019GXNSFAA245014 and 2020GXNSFBA238014).

References

[1] Z. Cai and Z. He, “Trading private range counting over big IoT data,” in 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp. 144–153, Dallas, TX, USA, 2019.

[2] Z. Cai, X. Zheng, and J. Yu, “A differential-private framework for urban traffic flows estimation via taxi companies,” IEEE Transactions on Industrial Informatics, vol. 15, pp. 6492–6499, 2019.

[3] D. Gerónimo, A. M. López, A. D. Sappa, and T. Graf, “Survey of pedestrian detection for advanced driver assistance systems,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, pp. 1239–1258, 2010.

[4] Y. Wang, Y. Gao, Y. Li, and X. Tong, “A worker-selection incentive mechanism for optimizing platform-centric mobile crowdsourcing systems,” Computer Networks, vol. 171, article 107144, 2020.

[5] G. Brazil and X. Liu, “Pedestrian detection with autoregressive network phases,” 2018, https://arxiv.org/abs/1812.00440.

[6] G. Cao, X. Xie, W. Yang, Q. Liao, G. Shi, and J. Wu, “Feature-fused SSD: fast detection for small objects,” 2017, https://arxiv.org/abs/1709.05054.

[7] R. B. Girshick, “Fast R-CNN,” in 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448, Santiago, Chile, 2015.

[8] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, “Scale-aware fast R-CNN for pedestrian detection,” IEEE Transactions on Multimedia, vol. 20, pp. 985–996, 2018.

[9] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, “Feature pyramid networks for object detection,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 936–944, Honolulu, HI, USA, 2017.

[10] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “Centernet: keypoint triplets for object detection,” 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 6568–6577, Seoul, Korea (South), 2019.

[11] H. Law and J. Deng, “Cornernet: detecting objects as paired keypoints,” in Computer Vision – ECCV 2018, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11218 of Lecture Notes in Computer Science, pp. 765–781, Springer, Cham, 2018.

[12] W. Liu, S. Liao, W. Ren, W. Hu, and Y. Yu, “High-level semantic feature detection: a new perspective for pedestrian detection,” 2019, https://arxiv.org/abs/1904.02948.

[13] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, Las Vegas, NV, USA, 2016.

[14] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: fully convolutional one-stage object detection,” in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9626–9635, Seoul, Korea (south), 2019.

[15] W. Wang, “Adapted center and scale prediction: more stable and more accurate,” 2020, https://arxiv.org/abs/2002.09053.

[16] J. Zhang, L. Lin, Y. Chen, Y. Hu, S. C. H. Hoi, and J. Zhu, “Attribute-aware pedestrian detection in a crowd,” 2019, http://arxiv.org/abs/1910.09188.

[17] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, “A unified multi-scale deep convolutional neural network for fast object detection,” in Computer Vision – ECCV 2016, B. Leibe, J. Matas, N. Sebe, and M. Wellin, Eds., vol. 9908 of Lecture Notes in Computer Science, pp. 354–370, Springer, Cham, 2016.

[18] Z. Cao, H. Yang, J. Zhao, X. Pan, L. Zhang, and Z. Liu, “A new region proposal network for far-infrared pedestrian detection,” IEEE Access, vol. 7, pp. 135023–135030, 2019.

[19] S. Ren, K. He, R. B. 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, pp. 1137–1149, 2017.

[20] L. Zhang, L. Lin, X. Liang, and K. He, “Is faster R-CNN doing well for pedestrian detection?” in Computer Vision – ECCV 2016, Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Wellin, Eds., pp. 443–457, Springer, Cham, 2016.

[21] A. Bochkovskiy, C. Wang, and H. M. Liao, “Yolov 4: optimal speed and accuracy of object detection,” 2020, https://arxiv.org/abs/2004.10934.

[22] J. Redmon and A. Farhadi, “Yolov 3: an incremental improvement,” 2018, https://arxiv.org/abs/1804.02767.

[23] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, “Imbalance problems in object detection: a review,” 2019, https://arxiv.org/abs/1909.00169.

[24] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999–3007, Venice, Italy, 2017.

[25] C. Zhu, Y. He, and M. Savvides, “Feature selective anchor-free module for single-shot object detection,” in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 840–849, Long Beach, CA, USA, 2019.

[26] P. Dollár, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: an evaluation of the state of the art,” IEEE trans., IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, pp. 743–761, 2012.

[27] S. Zhang, R. Benenson, and B. Schiele, “Citypersons: a diverse dataset for pedestrian detection,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4457–4465, Honolulu, HI, USA, 2017.

[28] G. Brazil, X. Yin, and X. Liu, “Illuminating pedestrians via simultaneous detection and segmentation,” in 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017.

[29] C. Zhou, M. Wu, and S. K. Lam, “Ssa-cnn: semantic self-attention cnn for pedestrian detection,” 2019, https://arxiv.org/abs/1902.09080.

[30] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, and C. Shen, “Repulsion loss: detecting pedestrians in a crowd,” 2017, https://arxiv.org/abs/1711.07752.

[31] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, “Occlusion-aware R-CNN: detecting pedestrians in a crowd,” in Computer...
[32] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. S. Huang, “Unitbox: an advanced object detection network,” in *Proceedings of the 24th ACM international conference on Multimedia*, pp. 516–520, Amsterdam, the Netherlands, 2016.

[33] L. Huang, Y. Yang, Y. Deng, and Y. Yu, “Densebox: unifying landmark localization with end to end object detection,” 2015, https://arxiv.org/abs/1509.04874.

[34] X. Zhou, J. Zhuo, and P. Krähenbühl, “Bottom-up object detection by grouping extreme and center points,” in 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 850–859, Long Beach, CA, USA, 2019b.

[35] A. G. Howard, M. Zhu, B. Chen et al., “Mobilenets: efficient convolutional neural networks for mobile vision applications,” 2017, https://arxiv.org/abs/1704.04861.

[36] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA, 2015.

[37] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.

[38] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in 2017 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, Honolulu, HI, USA, 2017.

[39] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, “Mask R-CNN,” in 2017 *IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, Venice, Italy, 2017.

[40] J. Mao, T. Xiao, Y. Jiang, and Z. Cao, “What can help pedestrian detection?,” in 2017 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6034–6043, Honolulu, HI, USA, 2017.

[41] Z. Cai and N. Vasconcelos, “Cascade R-CNN: high quality object detection and instance segmentation,” 2019, https://arxiv.org/abs/1906.09756.

[42] S. Zhang, R. Benenson, M. Omran, J. H. Hosang, and B. Schiele, “How far are we from solving pedestrian detection?,” in 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1259–1267, Las Vegas, NV, USA, 2016b.

[43] Z. Zou, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: a survey,” 2019, https://arxiv.org/abs/1905.05055.