applied sciences

MDPI

Article

Multiscale Object Detection in Infrared Streetscape Images Based on Deep Learning and Instance Level Data Augmentation

Hao Qu *, Lilian Zhang, Xuesong Wu, Xiaofeng He, Xiaoping Hu and Xudong Wen

College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China; lilianzhang@nudt.edu.cn (L.Z.); wuxuesong10@nudt.edu.cn (X.W.); hexiaofeng@nudt.edu.cn (X.H.); xphu@nudt.edu.cn (X.H.); wenxudong13@163.com (X.W.)

* Correspondence: quhao199541@163.com; Tel.: +86-181-9978-1192

Received: 24 December 2018; Accepted: 30 January 2019; Published: 8 February 2019

Abstract: The development of object detection in infrared images has attracted more attention in recent years. However, there are few studies on multi-scale object detection in infrared street scene images. Additionally, the lack of high-quality infrared datasets hinders research into such algorithms. In order to solve these issues, we firstly make a series of modifications based on Faster Region-Convolutional Neural Network (R-CNN). In this paper, a double-layer region proposal network (RPN) is proposed to predict proposals of different scales on both fine and coarse feature maps. Secondly, a multi-scale pooling module is introduced into the backbone of the network to explore the response of objects on different scales. Furthermore, the inception4 module and the position sensitive region of interest (ROI) align (PSalign) pooling layer are utilized to explore richer features of the objects. Thirdly, this paper proposes instance level data augmentation, which takes into account the imbalance between categories while enlarging dataset. In the training stage, the online hard example mining method is utilized to further improve the robustness of the algorithm in complex environments. The experimental results show that, compared with baseline, our detection method has state-of-the-art performance.

Keywords: infrared streetscape images; multiscale object detection; Faster R-CNN; instance level data augmentation

1. Introduction

With the development of infrared sensor technology, object detection in infrared images has attracted more attention in the fields of face recognition and pedestrian detection. Infrared images can display the temperature information of objects, which has obvious advantages in environments without sufficient illumination, for example at night. However, there are many challenges in using infrared images for object detection. Firstly, existing detection methods for infrared images are limited to single category detection. For street scene images containing different categories and multi-scale objects, related research is still scarce. In addition, infrared images have low resolution and contrast. There are fewer high-quality infrared street scene datasets compared with visible images, which also impedes the development of related detection algorithms.

In the field of traditional methods, for solving multi-scale issues, the mainstream methods at present are to use multi-scale sliding windows and image pyramids. In References [1,2], sliding windows were used to scan the whole image to generate thousands of proposals. The features in the proposals were then extracted by histogram of oriented gradient and classified by support vector machine. The number and size of sliding windows were determined by image size. However,

these methods produce too many proposals, which is not conducive to promoting detection efficiency. In References [3,4], researchers combined prior information such as image intensity, target motion, and gradient to reduce the number of proposals. Compared with the sliding window method, image pyramid can capture more information [5]. In Reference [6], researchers proposed the construction of a fast feature pyramid and used fixed-size sliding windows on each layer. Shi et al. introduced aggregated feature channels, which combined integral channel features and fast pyramids, which further improves robustness and real-time performance [7]. However, these methods only use the low-level feature of images to predict and filter proposals. When temperature and light change dramatically, these algorithms cannot work well.

The object detection methods based on deep learning utilize multi-layer convolution networks to extract more abstract semantic information of images. Their performance in complex environments is more robust than that of traditional methods [8,9]. Some researchers have tried to apply existing frames such as Faster Region-Convolutional Neural Network (R-CNN) to infrared images. In Reference [10], two kinds of network structures were designed to detect pedestrians in multispectral images. Compared with traditional methods, the miss rate was significantly reduced. Liu et al. [11] designed four network fusion structures based on Faster R-CNN to combine infrared and visible information and find out the optimal fusion strategy. Michelle et al. used two region of interest (ROI), pooling at the middle and bottom layers of the feature exactor, to obtain information at different scales. The results showed that this technique could effectively reduce the miss rate of pedestrian detection [12]. However, in Faster R-CNN, the region proposal network (RPN) only uses the last layer of the feature extraction network to generate proposals, which is not suitable for multi-scale object detection. Although more anchors with different scales can be set up to compensate, the cost of computation is highly increased. Moreover, Faster R-CNN only uses the coarsest feature map to classify, which is not capable of capturing the representation of small targets. In order to improve the performance of Faster R-CNN, there have been many efforts in the generic object detection pipeline. Some researchers integrate the low-level features with the high-level ones to makes the network more sensitive to small objects [13,14]. SSD [15], RSSD [16], and DSSD [17] generate proposals and predict objects on each layer of backbone to improve the prediction accuracy of multi-scale objects. FPN [18], RON [19], and Mask R-CNN [20] combine coarse to fine features to capture richer multi-scale representation. However, these methods inevitably enlarge the feature dimension, which leads to a rise of computational cost. In order to balance efficiency and performance, some researchers [21–23] have employed multi-scale ROI pooling to combine the output of different hierarchies. The network can then obtain multi-scale features simultaneously. Other studies have focused on changing the receptive field of feature maps to obtain multi-scale information without significant increase in computational complexity. DetNet [24] and RFBNet [25] add dilated convolution into the feature extraction network to increase the density of the feature map. The experiment results show that the performance of these methods is competitive with the mainstream ones.

Another issue that affects the accuracy of object detection on infrared images is the lack of high-quality datasets. The common solution is data augmentation, for example, flipping, changing the contrast, blurring, cropping, and mirroring. However, in the field of object detection, there is little research on it. In Reference [15], the Pascal VOC dataset was randomly cropped, mirrored, and scaled to further improve the mean average precision of SSD for small objects. In Reference [14], an algorithm randomly selected angles, then the images and annotations were rotated accordingly. In Reference [26], an algorithm randomly selected a rectangular region in the image, and replaced its pixels with random values. Multi-scale occlusion was then generated. To summarize, these existing data augmentation methods for object detection have the following shortcomings. Firstly, these algorithms randomly transform every image without considering the proportions of different objects in the dataset. It is not helpful to solve non-uniform class distribution. Secondly, since these methods take the whole image to rotate, clip, scale, or erase, they are not able to solve non-uniform class distribution when different kinds of objects exist in the image. Thirdly, in the process of augmentation, these algorithms

do not take into account the spatial relationship between the objects in the image. It is easy to lead to the loss of image information. In the field of image classification, Liu et al. put forward the method of under-sampling, which can effectively expand the amount of minor category data without losing the original data [27]. However, these methods have difficulty balancing the proportions of different categories while expanding the object detection dataset, especially when the proportions of categories vary greatly.

In this paper, our first motivation is to modify the structure of Faster R-CNN to improve the performance of multi-scale object detection in infrared images. Our second motivation is to develop a data augmentation method that can solve the issue of non-uniform class distribution while augmenting a dataset. The general structure of the network is shown in Figure 1. Our main contributions are summarized as follows:

(1) We designed a double-layer RPN pyramid, in which the receptive field of the top feature map is more suitable for the prediction of large objects, while the receptive field of the bottom is denser and the resolution is higher for the prediction of small objects.

(2) We introduce a multi-scale pooling method plus the inception4 module to project proposals on outputs of different scales simultaneously. A 1×1 convolution is then utilized to synthesize coarse and fine features. In this way, the network can capture more abundant multi-scale information.

(3) By adding the inception4 module after the backbone, the receptive fields of feature maps generated by different size convolution kernels are various. This makes the extractor more sensitive to objects with different sizes. At the same time, the shortcut makes network training smoother.

(4) Because of the background noise and the scarcity of foreground details, we used PSalign pooling to extract the local features of objects. It can fully explore the foreground information of objects.

(5) Since the existing data augmentation methods cannot deal well with the imbalance of categories, we elaborately designed an instance-level data augmentation method. Taking account of the proportion of different categories and the spatial relationships between objects, the algorithm ensures that the original data is not damaged while expanding the dataset.

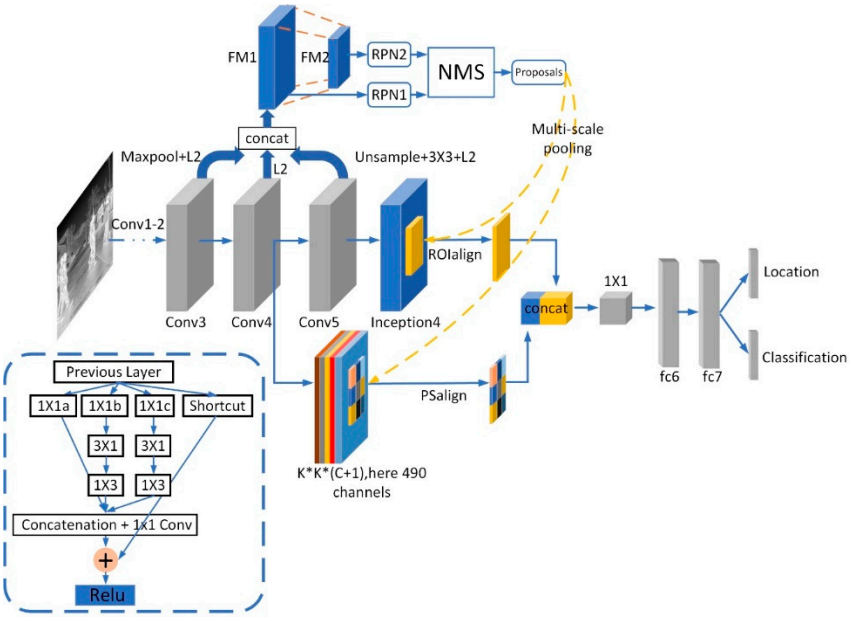


Figure 1. The complete structure of our frame.

The structure of this paper is as follows: Sections 2 and 3 introduce the details of the network modification. Section 4 describes our data augmentation method. Section 5 presents the experimental results and analysis. We draw the conclusion in Section 6.

2. Double-Layer RPN Pyramid

In Faster R-CNN, the regional proposal network uses the last layer of a feature extraction network, such as VGG16, to predict candidate boxes. However, the receptive field of the last layer is too large and the resolution is too low, making it unsuitable for small object detection. Hypernet [13] and other structures combine the output of multiscale layers to construct a fusion feature map, which contains high-level semantic features and low-level object details. This has been proven to improve the detection of small objects. However, in street scene images where multiple targets exist simultaneously, our attention should not be limited to small objects. Inspired by Hypernet, we redesigned RPN to further explore the detection potential of RPN for multi-scale objects. The details are shown in the following.

As shown in Figure 1, similar to Hypernet, we used max pooling to resize the third layer output into the dimension of a fourth layer. Unlike the other methods mentioned in References [14,22], which use deconvolution layers or bilinear up-sampling layers, we used the nearest neighbor interpolation to unify the output of the fifth layer to the fourth layer dimension, and add a convolutional layer (kernel size 3) to eliminate the overlap of pixels caused by the up-sampling. Since the properties of the feature map and the number of feature channels from different layers were different, we used the L2 normalize method to normalize the feature channel before the combination layer. A convolutional layer (kernel size 1) was then used to reduce the channels to 512. After adding a distillation convolution layer (kernel size 3, rate 2), the receptive field of the RPN can be further enlarged. After that, we used a max pooling layer to reduce the dimension of the feature map, followed by a convolution layer (kernel size 3) to further expand the receptive field. RPN1 and RPN2 with the same structure could predict the proposals on feature map 1(FM1) and feature map 2(FM2), respectively. Due to the different scale of receptive field, RPN1 is more suitable for predicting small targets and RPN2 is more suitable for predicting large targets. According to the size of object in the dataset, we set the same base size and ratio of anchor (base size: 64^2, 128^2, 256^2, 512^2, ratio: 0.5, 1, 2) on RPN1 and RPN2. At the same time, in order to control the number of proposals, we used non maximum suppression (NMS) to reduce the number of boxes from 12000 to 2000. The threshold of intersection over-union (IOU) was 0.7.

3. Multi-Scale Pooling with Inception4 Module and PSalign

We believe that the responses of objects with various sizes on the feature map are also different. However, in Faster R-CNN, the proposals generated by the RPN use the ROI (Region of Interest) pooling layer to extract the corresponding response directly on the output from conv5, in which the size of the receptive field for each neuron is the same. Therefore, the network is not robust to multiscale object detection. In order to solve this issue, we decided to obtain feature maps of corresponding proposals from different hierarchies (conv5, conv4). We then introduced an inception4 [28] module to further explore multi-scale features. As shown in Figure 1, after conv5, we added three convolution layers (kernel size 1) to divide the channels of the feature map into 256, 128, and 128. After 1×1b, we employed a 3×3 convolution layer. In order to speed up training, we replaced it with two 1×3 and 3×1 convolution layers. Using the same strategy, 1×5 and 5×1 convolution layers were added after 1×1c to substitute 5×5 convolution layers. After that, the outputs of three stacks were concatenated by a concat layer, and the number of channels was adjusted by a 1×1 convolution layer to 512. Moreover, we applied a shortcut module, and a ReLu layer was used. Eventually, we used a ROI align pooling layer to project the region of the proposals predicted by RPN to the size of 7×7. Using an ROI align pooling layer overcomes the misalignment problem of the ROI pooling layer. This can locate the position of proposals more accurately on the feature map by means of quadratic difference, which is very helpful for the detection accuracy of small objects [20].

In addition, due to the lack of background information in infrared images, training of the network should focus on the foreground information. In this paper, we added a 1×1 convolution layer after conv4, which outputs a feature map with $K^2(C+1)$ channels. This means that we divided the proposal into K^2 regions, and each was encoded by (C+1) channels in the feature map. C represents the number of categories [29]. Similar to Light head R-FCN [30], we directly set the category number to 9, increasing the number of channels of feature map to enrich the appearance. We then used a position sensitive ROI align (PSalign) pooling layer to project the corresponding feature map to $7 \times 7 \times 10$ channels. Furthermore, we used a concat layer to merge the output of the PSalign pooling layer and the ROI align pooling layer. After that, a 1×1 convolution layer was added to reduce the channels of the feature map to 512. Additionally, we applied two 4096 full-connection layers to encode the information of all feature maps. Ultimately, we employed two full-connection layers for location regression and classification.

4. Instance Level Data Augmentation

In this section, we propose a novel instance level data augmentation method. For different categories and different scales, different strategies have been designed.

We used the FILR dataset for our experiment, which includes four categories: person, car, bicycle, dog, and other vehicle. We choose three of them for the experiment: person, car, and bicycle. Among them, the number of cars was 46,692, the number of people was 28,151, and the number of bicycles was 4457. Obviously, the number of bicycles was much less than that of the other two categories.

Training of a network tends to favor categories of a larger proportion, which, to some extent, suppresses minor ones [14]. Inspired by the ideas in References [27,31], we down-sampled the objects of major categories and up-sampled the objects of minor categories. Since the number of cars in the dataset was much larger than that of the other categories, we only down-sampled the number of cars, and intentionally expanded other two categories. Since transforming small-scale targets will lead to distortion, we only transformed large-scale targets by a series of strategies. All operations were performed on single object. We outline the steps of the algorithm in Figure 2. The specific procedures are as follows:

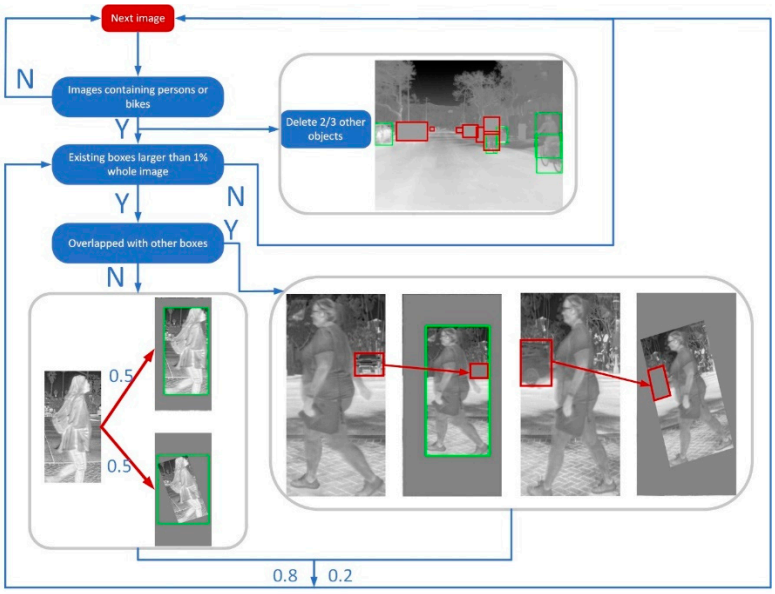


Figure 2. Flow chart of data augmentation algorithm. Green lines indicate annotations, red lines indicate deleted boxes.

Firstly, we picked out images containing people or bicycles. If there were objects (person or bicycle) larger than 1% of the whole image, they were considered to be large-scale objects. If there were

no large-scale objects, two-thirds of the cars in the image were randomly deducted. That is, the pixels in boxes were replaced by the mean of image pixels, and their annotations were deleted in the xml file. At the same time, we kept unselected objects intact. The next step was to determine whether those large-scale objects overlapped with other objects. If false, the algorithm scaled or rotated the large-scale objects randomly, and the probability of each choice was 0.5. The first choice was to zoom out the content inside and randomly move its center within a certain range. The ratio of zooming out was 0.6 to 0.8. If there was overlap with other objects, the overlapped objects were deducted. The deletion operation was the same as above. We then created a new image in which the previously deleted objects were retained and the large-scale object was deleted. The purpose of this step is to keep the completeness of the original dataset. If random rotation was selected, the algorithm rotated the frame at a small angle. Rotation angle was 10 to 20 degrees. After processing one object, the algorithm processed the next large-scale object or jumped to next image with a probability of 0.8 or 0.2. The above procedures were performed on images of people and bicycle respectively. Compared with random clipping [26], our method can produce richer variation during data augmentation.

After processing all the images, we counted the number of the three categories in the processed dataset, and found that the algorithm did not change the number of bicycles to the same order of magnitude as it did the number of other two categories. Therefore, the random rotation algorithm was used to further expand the dataset. For bicycles, we wanted a further four times the data, so we randomly sampled four angles in (0,360) for each image containing bicycles. The algorithm then rotated the image and its corresponding annotation using the following Equation (1).

$$\begin{Bmatrix} x_i' \\ y_i' \end{Bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad \mathrm{i} = 1, 2, 3, 4 \tag{1}$$

Compared with the previous method, we selected the midpoints on the four sides of the box after rotation, then used Equation (2) to get the coordinates of the four corners. The goal was to reduce the useless content in the new box.

$$\begin{cases} x_{\min} = \min\begin{pmatrix} x_1^{mid} & x_2^{mid} & x_3^{mid} & x_4^{mid} \end{pmatrix} \\ y_{\min} = \min\begin{pmatrix} y_1^{mid} & y_2^{mid} & y_3^{mid} & y_4^{mid} \end{pmatrix} \\ x_{\max} = \max\begin{pmatrix} x_1^{mid} & x_2^{mid} & x_3^{mid} & x_4^{mid} \end{pmatrix} \\ y_{\max} = \max\begin{pmatrix} y_1^{mid} & y_2^{mid} & y_3^{mid} & y_4^{mid} \end{pmatrix} \end{cases} \tag{2}$$

The results of the data augmentation algorithm are shown in Figure 3. We can see that the algorithm can effectively suppress the increase of cars and transform large-scale objects without introducing noise as much as possible. Some implementation details are supplied in the next section.

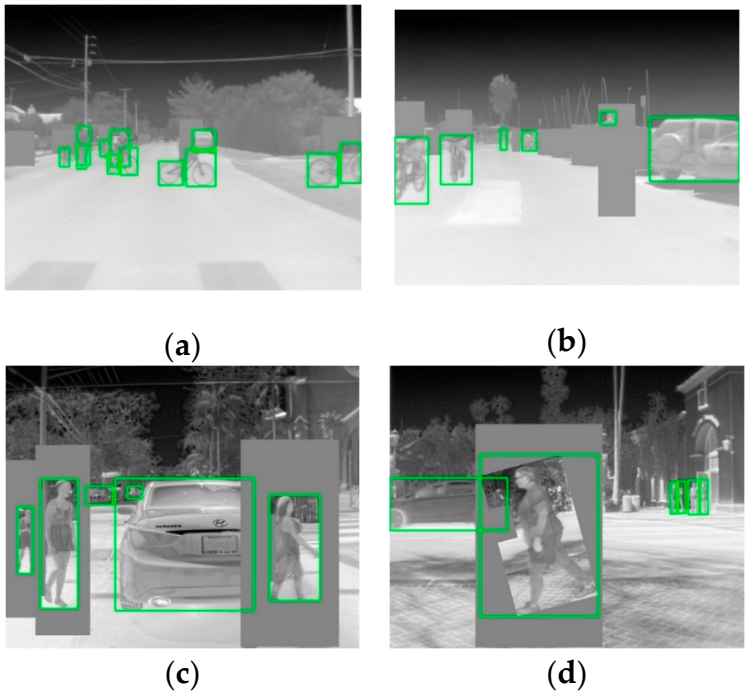


Figure 3. Examples of the results of instance level data augmentation before rotation. Green lines indicate annotations after transformation. (**a**) and (**b**) are augmentation examples for bicycles, (**c**) and (**d**) are augmentation examples for persons.

5. Experiments and Results

5.1. Implementation Steps

We use TensorFlow to build, train, and test the network. All the experiments were implemented in the workshop with Intel i7-8700 K CPU, NVIDIA GTX1080ti GPU, and 32 GB RAM. Our network is based on Faster R-CNN, in which the feature extraction network is VGG16. The parameters from the first layer to the full connection layer were pre-trained on the ImageNet dataset. Other layers were randomly initialized by the Xavier method. We used datasets provided by FILR for training and testing. Additionally, the number of official training images was 8862. However, in our experiment, only person, car, and bicycle were selected for use in the experiment; there were 7856 pictures left after deleting the pictures without the above categories. We merged truck, van, and tram in other vehicle into car, and person sitting into person. The number of test sets was 1360. All images were subtracted by mean values (102.9801, 115.9465, 122.7717) before training and testing. We trained the RPN and Fast R-CNN jointly. For all the networks involved in this paper, we trained for 8 epochs. The learning rate of the first three epochs was 0.001, and that of the last five epochs was 0.0001. Using the stochastic gradient descent (SGD) training method, the momentum was 0.9 and the weight decay weight was 0.0001. The batch size was 1. We resized the shortest edge of each image to 600 during training and testing.

5.2. Implement Details of Instance Level Data Augmentation

In order to make the description of the algorithm more intuitive, the object that algorithm currently operates on is called the current box/object, and the object around it is called another box/object.

Since the overlap of objects in street scene images is common, when transforming a single box, it will somehow affect the content of the surrounding boxes. For example, if the current box is transformed, and the overlapped area with another box is too large, it will inevitably damage content in another box. When another box is completely in the current box, its annotations will be invalidated. Of course, we can simultaneously transform another box, but this will make programming

more complicated. Because of the complex spatial relationship between objects, the results of the augmentation will be affected to some extent. Therefore, it is necessary to quantitatively measure the spatial relationship among objects in algorithm. In this paper, two ratios (IOU1, IOU2) are proposed to take different roles in algorithm.

Firstly, IOU1 is the ratio of the common part of two boxes to the current box, as seen in Figure 4a. For the current object, if the content of another object occupies a large proportion (IOU1 > 0.4) of its box, deleting the common part will lead to a huge loss of current box content, and, to a certain extent, false positive examples will be introduced. Therefore, we don't delete the common part, as seen in Figure 4b. If IOU1 is less than 0.4, deleting the common part is equivalent to a cropping operation. Secondly, IOU2 is the ratio of the area of the common part to the area of another box, as seen in Figure 4c. For another box, if most of its content is in the current box (IOU2 > 0.35), the transformation of the current box will inevitably lead to serious loss of another box's content. Therefore, the algorithm needs to delete another object, otherwise false positive examples will be introduced into the dataset, as seen in Figure 4d. If the proportion is less than 0.35, it is equivalent to the cropping another object. The setting of relevant parameters was based on experiment experience.

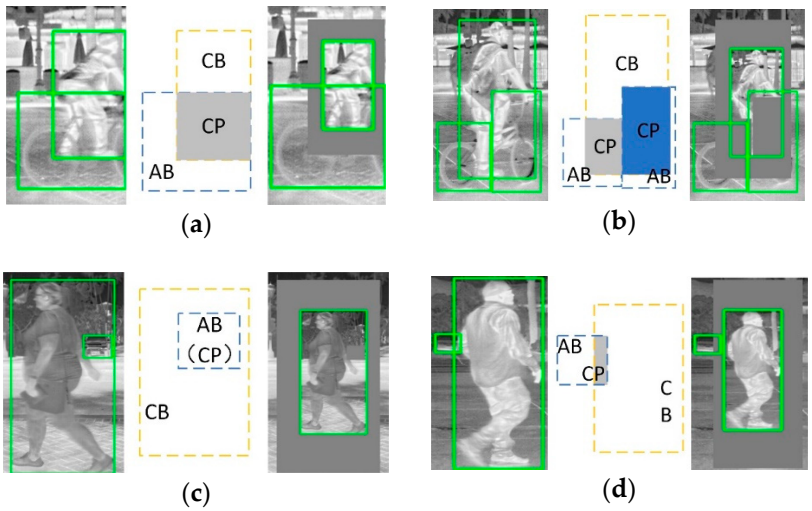


Figure 4. Examples of two ratios. CP: common part, CB: current box, AB: another box.

5.3. Online Hard Example Mining

In infrared images, the scarcity of image details makes it difficult to distinguish the background from the foreground. Faster R-CNN samples the amount of foreground and background information after NMS at a ratio of 1:3. However, most of the background information is easy to distinguish. Excessively easy samples are not conducive to convergence of training. Therefore, it is necessary to focus on learning more difficult samples to enhance the robustness of networks. After 2000 candidate boxes were generated by two RPNs, they were brought into the subsequent layers to calculate their loss values. The algorithm then selected 256 mini-batches with the largest loss value and applied back propagation to them.

5.4. Comparative Experiment

In this section, in order to verify that our network is competitive with other mainstream Faster R-CNN based frames, we strictly controlled the experiment conditions. The network structures of all the experiments listed in Table 1 were the same as those designed in original papers. For convenience, all experiments adopted an end-to-end training strategy. In order to prove the superiority of our network structure, we did not use Online Hard Example Mining (OHEM) to pick out 256 proposals generated by a double-layer RPN pyramid. In all the experiments shown in Table 1, we directly selected 256 proposals as a mini-batch for training, where the ratio of positive and negative samples

was 1:3. Relevant training strategies were the same as those stated in Section 5.1. We used the original FILR dataset to train all frames in the experiment, using flipping as a data augmentation strategy.

Table 1. The results of different frames on the FILR dataset. Bold numbers indicate the highest indicator values of all frames. IOU threshold is 0.3.

Detention Method	AP (%)			mAP (%)	Test Time (s)
	Car	Person	Bicycle		
Faster R-CNN (vgg16)	74.05	62.02	43.98	60.02	0.068
Faster R-CNN(Res101)	77.80	65.73	46.06	63.20	0.157
HyperNet	78.10	72.42	47.61	66.04	0.113
CMS-R-CNN [32]	77.39	60.59	42.50	60.16	0.085
R-FCN	76.80	63.62	43.87	61.43	**0.065**
Light head R-FCN [30]	77.77	64.49	48.78	63.68	0.119
Our method	**78.96**	**74.12**	**52.49**	**68.52**	0.108

Compared with Faster R-CNN (vgg16), using a deeper residual network as a backbone can extract more abstract features of the object, resulting in a performance improvement of 3.18%. However, since only the last layer of backbone is used as the input of RPN, the detection performance was not improved much. At the same time, the computational complexity has increased dramatically. Hypernet fuses the output of the first, third, and fifth layers of backbone to get the fused feature map for proposal generation. Additionally, it reduces the channels of the fused feature map in order to avoid an increase of computation. These improvements have positive effects, but the computational cost is still expensive. The purpose of performing a CMS-R-CNN experiment is to verify the role of context in infrared images. We set the area of context to 1.3 times that of the proposal. However, the experimental results show that the effect of context information on detection is very slight, which also verifies that the background information in infrared image is monotonous. Meanwhile, the scale parameter is also difficult to be set up. Both Light head R-FCN and R-FCN use a PSalign pooling layer to extract the local features of objects. The former uses a full connection layer to locate and recognize objects, while the latter uses a voting mechanism. The speed of R-FCN is noteworthy, however, the experimental results show that local features alone are still inadequate for multi-scale object detection. A double-layer RPN pyramid module, PSalign pooling layer, and inception4 module were used in our method. Relevant experiments show that the performance of detecting cars, persons and bicycles has been improved remarkably, while the detection efficiency was maintained at a middle level.

In order to verify the usefulness of each network structure improvement and data augmentation, we have carefully designed the following experiments. It is noteworthy that the baseline of the following experiments is Faster R-CNN with VGG16 as backbone and pooling as ROI align. Obviously, using ROI align pooling, we can get more precise locations of proposals. Compared to the original Faster R-CNN with ROI pooling, mean average precision (mAP) was increased by 6.18%. We listed experiment results in Table 2.

Ours1 was set to verify the improvement of double-layer RPN pyramid (DR). Ours1 uses two RPNs to generate different scale proposals on coarse and fine feature maps in DR. Using NMS to filter redundancy, DR can provide more precise proposals. The results show that DR gives Ours1 a 1.39% improvement over baseline.

In order to effectively capture multi-scale features, Ours2 obtains the appearance of corresponding objects on different levels (conv5, conv4). Combining inception4 and PSalign pooling layer, the network can explore more potentials of the multi-scale pooling (MSP) module. The results show that MSP leads to a 0.93% improvement over Ours1.

Table 2. The results of comparative experiments of the different parts in our frame. Bold numbers indicate the highest indicator values of all frames. IOU threshold is 0.3.

Detection Method	DR	MSP	ILDA	OHEM	AP (%)			mAP (%)
					Car	Person	Bicycle	
baseline	x	x	x	x	78.59	67.29	52.73	66.20
Ours1	√	x	x	x	78.66	71.71	52.40	67.59
Ours2	√	√	x	x	78.96	74.12	52.49	68.52
Ours3	√	√	√	x	79.10	73.96	**55.60**	69.55
Ours4	√	√	√	√	**79.58**	**74.95**	55.48	**70.00**
Ours5	√	x	√	√	78.87	72.20	55.49	68.85
Ours6	√	√	x	√	79.29	74.72	53.68	69.23
Ours7	x	√	√	√	79.46	73.46	52.02	68.31

In these experiments, the default method of data augmentation was flipping. The purpose of Ours2 and Ours3 was to verify the alleviation of non-uniform class distribution by instance level data augmentation (ILDA). From the results, we can see that our method improved the detection of a minor category (bicycle) by 3.11%, and also improved the detection of cars by 0.14%, but the detection of people showed a slight degradation by 0.16%.

DR selected out 2000 proposals using NMS. Ours3 directly selected positive and negative examples of 1:3 to form 256 mini-batches for training. However, there were many false positive and false negative examples among them. Therefore, training should pay more attention to them. Using loss value as a criterion to measure batch difficulty, we chose 256 batches with the highest loss value for training. The results show that, compared with Ours3, Ours4 had a low-level improvement of 0.45%.

In order to further verify the impact of each factor in our frame, we conducted three experiments from Ours5 to Ours7. The results showed that all parts of our frame are indispensable. Figure 5 shows the experimental results of Ours3 to Ours7 compared with Ours4.

Figure 5. Comparative results (**a**) between Ours6 and Ours4, (**b**) between Ours5 and Ours4, (**c**) between Ours3 and Ours4, (**d**) between Ours7 and Ours4. Yellow lines indicate correctly-detected objects, red lines indicate misjudged objects, and black lines indicate incorrectly-detected objects.

From Figure 5a, we can see that, without abundant and diverse bicycle data, the frame is not robust to bicycle detection. From Figure 5b, without multi-scale pooling module, the network is not able to adequately capture multi-scale representations, and the detection results are poor in complex environments. From Figure 5c, without OHEM to strengthen the training, the detection will be disturbed in complex environments and noise will be easily misidentified as an object. From Figure 5d,

without a double-layer RPN pyramid, the network is not able to produce multi-scale proposals in the RPN stage, which will damage the detection results.

To further verify the validity of the inception module (INCEP) and PSalign (PS) layers in multi-scale pooling, we conducted three experiments based on Ours4. Tthe results are shown in Table 3. In Ours4a, DR-generated proposals were projected directly on conv4 and conv5 to obtain corresponding features. In Ours4b, after conv5, an inception 4 module was added. The network used ROI align to obtain corresponding features after inception 4 and conv4, respectively. In Ours4c, a convolution layer with 490 channels was added after conv4. No additional modules were added after conv5. The corresponding features were then obtained using PSalign and ROI align respectively. We listed experiment results in Table 3.

Table 3. The results of comparative experiments of different parts in multi-scale pooling. Bold numbers indicate the highest indicator values of all frames. IOU threshold is 0.3. All experiments utilize double-layer RPN pyramid (DR), instance level data augmentation (ILDA), and Online Hard Example Mining (OHEM).

Detention Method	INCEP	PS	AP (%)			mAP (%)
			Car	Person	Bicycle	
Ours4a	x	x	79.32	75.07	53.78	69.39
Ours4b	√	x	79.17	72.29	**55.72**	69.06
Ours4c	x	√	79.53	**75.18**	53.27	69.33
Ours4	√	√	**79.58**	74.95	55.48	**70.00**

Compared with Ours5 in Table 2, we can see that in Ours4a the positive effect of multi-scale pooling module was obvious. Although the mAP of ours4b and ours4c were lower than that of Ours4a, they actually had some advantages in detecting bicycles and people. It is noteworthy that network design is a systematic work. As shown in Ours4, different modules can combine to explore each other's potential. Therefore, we can demonstrate that each modification in the multi-scale pooling module is useful and indispensable.

5.5. False Alarms and Misjudgment

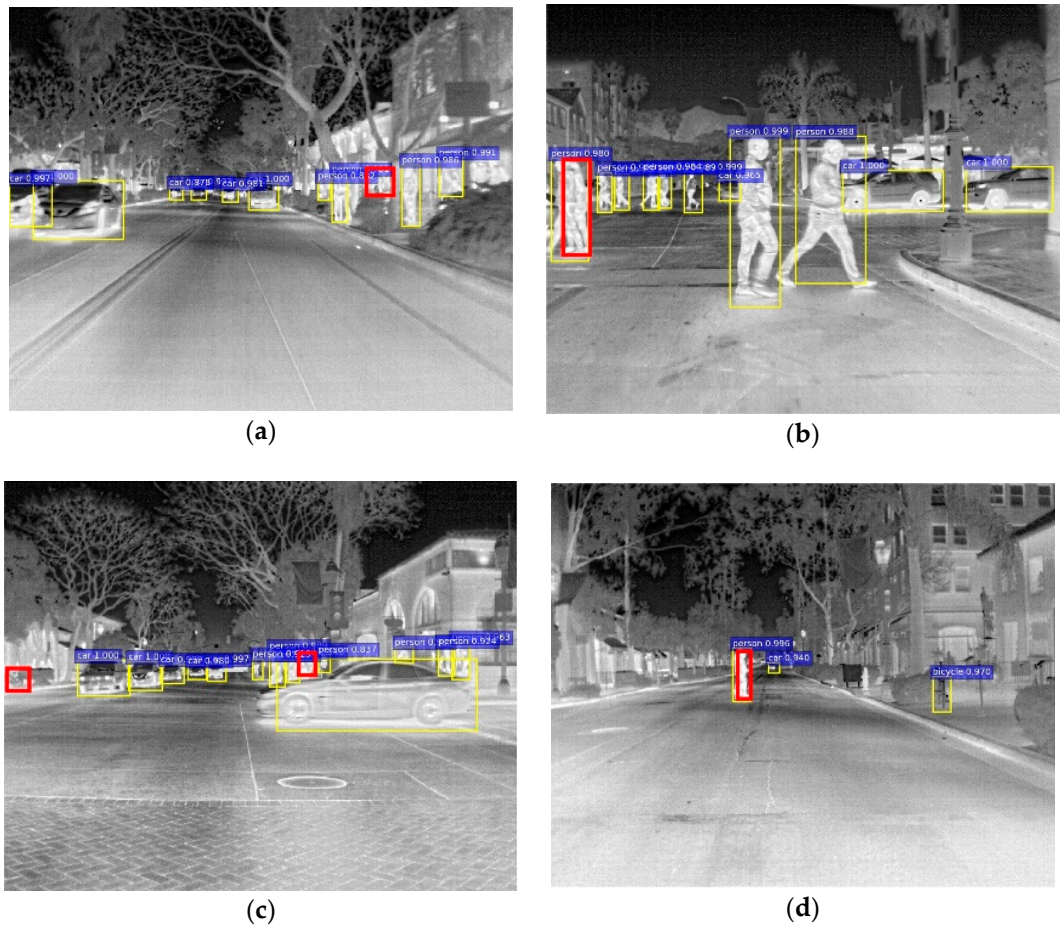
The validity of our frame has been fully confirmed by the above comparative experiments, especially for the detection of multi-scale objects in infrared images. However, it still can be seen from the table that the improvement of the algorithm for person and car detection is not stable. Additionally, there is still much room for improving bicycle detection. Observing the whole test process, we found that false alarm and misjudgment had great influence on the test results, as shown in Figures 6 and 7.

(**a**) (**b**)

Figure 6. *Cont.*

(**c**) (**d**)

Figure 6. False alarms occur under complex circumstances. Red lines indicate errors. Yellow lines indicate correct detections.



(**a**) (**b**)

(**c**) (**d**)

Figure 7. Misjudgments occur under the complex circumstance. Red lines indicate undetected objects. Yellow boxes indicate correct detections.

As can be seen from Figure 6a–c, most of the false alarms were errors in the recognition of small objects. This is because the infrared image resolution is too low, resulting in the scarcity of small-scale details. It is difficult to improve the discernment of the frame under complex circumstances by using OHEM for error-oriented training. It can be seen from Figure 6d that objects of similar shape may also be misidentified. Since the reality is that the infrared image lacks object details, it is possible to try to combine visible images in future work.

It can be seen from Figure 7a,c that for small objects with lower resolution, the network may also confuse them with the background, which makes them undetected. For cases where the objects in Figure 7b,d overlap each other, the network seems to be slightly incapable. In general, the performance of our frame for multi-scale objects is still worthy of attention.

6. Conclusions

In this paper, we started from two aspects to solve the problems of multi-scale object detection in infrared street scene images. Firstly, from the view of changing the receptive field, we designed a double-layer RPN pyramid for more accurate proposal prediction. The structure of the backbone was improved by introducing a multi-scale pooling module plus inception 4 to fully explore the properties of different scales. The local features obtained by the PSalign pooling layer were used to further improve the performance of the network, using OHEM to further explore the potential of the frame. Secondly, because of the scarcity of data augmentation for object detection, we designed an instance level data augmentation, in which the algorithm fully takes into account the spatial and quantitative relationship between categories. To some extent, it alleviates the issue of non-uniform class distribution in datasets. In order to fully demonstrate the superiority of our frame, we carried out several experiments, in which we compared the performance and efficiency with the current mainstream modifications based on Faster R-CNN. The experiments showed that our frame for multi-scale detection in infrared street scene images has reached state-of-the-art level, while the efficiency can be maintained at a reasonable level. In addition, we designed comparative experiments for each part of frame, which fully proved that each improvement was reasonable and effective. However, there are still some shortcomings in our method. For example, the effect of data augmentation is not particularly obvious. We speculate that the fixed parameters in the algorithm are not robust for variable situations. We will design more intelligent algorithms to automate parameter settings in future work. In addition, considering some detection errors appeared in the detection results, we will use a more complex backbone in future experiments to improve the network's ability to distinguish different objects.

Author Contributions: H.Q. and L.Z. discussed and designed the structure of the network. X.W. provided the dataset for experiment. X.H. verified the rationality of the experiment. X.H. provided experimental instruments. X.W. polished the manuscript. H.Q. and L.Z. revised the paper.

Funding: This research was funded by the National Natural Science Foundation of China under grants 61773394, 61573371, and 61503403.

Acknowledgments: The author would like to thank all the teachers and colleagues who provided inspirations and equipment in the experiment. The author would like to thank all the anonymous reviewers for their meticulous comments and helpful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. O'Malley, R.; Jones, E.; Glavin, M. Detection of pedestrians in far-infrared automotive night vision using region-growing and clothing distortion compensation. *Infrared Phys. Technol.* **2010**, *53*, 439–449. [CrossRef]
2. Qi, B.; John, V.; Liu, Z.; Mita, S. Use of Sparse Representation for Pedestrian Detection in Thermal Images. In Proceedings of the Workshop on Perception Beyond the Visible Spectrum, Columbus, OH, USA, 23–28 June 2014; pp. 274–280.
3. Bertozzi, M.; Broggi, A.; Carletti, M.; Fascioli, A.; Graf, T.; Grisleri, P.; Meinecke, M. IR Pedestrian Detection for Advanced Driver Assistance Systems. In Proceedings of the Pattern Recognition, Dagm Symposium, Magdeburg, Germany, 10–12 September 2003; pp. 582–590.
4. Dai, C.; Zheng, Y.; Xin, L. Pedestrian detection and tracking in infrared imagery using shape and appearance. *Comput. Vis. Image Underst.* **2007**, *106*, 288–299. [CrossRef]
5. Piotr, D.; Ron, A.; Serge, B.; Pietro, P. Fast Feature Pyramids for Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1532–1545.

6. Biswas, S.K.; Milanfar, P. Linear Support Tensor Machine: Pedestrian Detection in Thermal Infrared Images. *IEEE Trans. Image Process.* **2016**. [CrossRef] [PubMed]
7. Shi, Y.B.; Zhang, Y. Algorithm for Infrared Pedestrian Detection Based on Aggregated Channel Features. *Infrared* **2018**, *39*, 44–50.
8. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
9. Girshick, R. Fast R-CNN. *arXiv* **2015**, arXiv:1504.08083.
10. Hou, Y.L.; Song, Y.; Hao, X.; Shen, Y.; Qian, M. Multispectral pedestrian detection based on deep convolutional neural networks. In Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing, Xiamen, China, 22–25 October 2017; pp. 1–4.
11. Liu, J.; Zhang, S.; Wang, S.; Metaxas, D.N. Multispectral deep neural networks for pedestrian detection. *arXiv* **2016**, arXiv:1611.02644.
12. Galarza-Bravo, M.A.; Flores-Calero, M.J. Pedestrian Detection at Night Based on Faster R-CNN and Far Infrared Images. In Proceedings of the International Conference on Intelligent Robotics and Applications, Newcastle, NSW, Australia, 9–11 August 2018; pp. 335–345.
13. Kong, T.; Yao, A.; Chen, Y.; Sun, F. HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 845–853.
14. Ren, Y.; Zhu, C.; Xiao, S. Small Object Detection in Optical Remote Sensing Images via Modified Faster R-CNN. *Appl. Sci.* **2018**, *8*, 813. [CrossRef]
15. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. *arXiv* **2015**, arXiv:1512.02325.
16. Jeong, J.; Park, H.; Kwak, N. Enhancement of SSD by concatenating feature maps for object detection. *arXiv* **2017**, arXiv:1705.09587.
17. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. DSSD: Deconvolutional Single Shot Detector. *arXiv* **2017**, arXiv:1701.06659.
18. Lin, T.-Y.; Dollár, P.; Girshick, R.B.; He, K.; Hariharan, B.; Belongie, S.J. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; p. 4.
19. Kong, T.; Sun, F.; Yao, A.; Liu, H.; Lu, M.; Chen, Y. Ron: Reverse connection with objectness prior networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; p. 2.
20. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
21. Zhong, Z.; Jin, L.; Zhang, S.; Feng, Z. Deeptext: A unified framework for text proposal generation and text detection in natural images. *arXiv* **2016**, arXiv:1605.07314.
22. Xu, D.; Ouyang, W.; Ricci, E.; Wang, X.; Sebe, N. Learning Cross-Modal Deep Representations for Robust Pedestrian Detection. *arXiv* **2017**, arXiv:1704.02431.
23. Yang, X.; Sun, H.; Fu, K.; Yang, J.; Sun, X.; Yan, M.; Guo, Z. Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks. *Remote Sens.* **2018**, *10*, 132. [CrossRef]
24. Li, Z.; Peng, C.; Yu, G.; Zhang, X.; Deng, Y.; Sun, J. DetNet: A Backbone network for Object Detection. *arXiv* **2018**, arXiv:1804.06215.
25. Liu, S.; Huang, D.; Wang, Y. Receptive Field Block Net for Accurate and Fast Object Detection. *arXiv* **2018**, arXiv:1711.07767.
26. Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y. Random Erasing Data Augmentation. *arXiv* **2017**, arXiv:1708.0489.
27. Liu, X.-Y.; Wu, J.; Zhou, Z.-H. Exploratory undersampling for class-imbalance learning. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2009**, *39*, 539–550.
28. Szegedy, C.; Ioffe, S.; Vanhoucke, V. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv* **2016**, arXiv:1602.07261.

29. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. *arXiv* **2016**, arXiv:1605.06409.
30. Li, Z.; Peng, C.; Yu, G.; Zhang, X.; Deng, Y.; Sun, J. Light-Head R-CNN: In Defense of Two-Stage Object Detector. *arXiv* **2017**, arXiv:1711.07264.
31. Shen, L.; Lin, Z.; Huang, Q. Relay Backpropagation for Effective Learning of Deep Convolutional Neural Networks. *Comput. Sci.* **2015**, *7214*, 467–482.
32. Zhu, C.; Zheng, Y.; Luu, K.; Savvides, M. CMS-RCNN: Contextual Multi-Scale Region-Based CNN for Unconstrained Face Detection. *arXiv* **2016**, arXiv:1606.05413.