Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5544194, 14 pages
https://doi.org/10.1155/2021/5544194

Research Article

Feature-Enhanced Occlusion Perception Object Detection for Smart Cities

Jie Xu,[1] Hanyuan Wang,[1] Mingzhu Xu,[1] Fan Yang,[1] Yifei Zhou,[2] and Xiaolong Yang[3]

[1]*School of Information and Communication Engineering, University of Electronic Science and Technology of China, 611731, China*
[2]*State Grid Sichuan Electric Power Corporation Metering Center, 610045, China*
[3]*School of Computer and Communication Engineering, University of Science and Technology Beijing, 100083, China*

Correspondence should be addressed to Jie Xu; xuj@uestc.edu.cn

Received 26 February 2021; Revised 8 March 2021; Accepted 10 March 2021; Published 30 March 2021

Academic Editor: Xin Liu

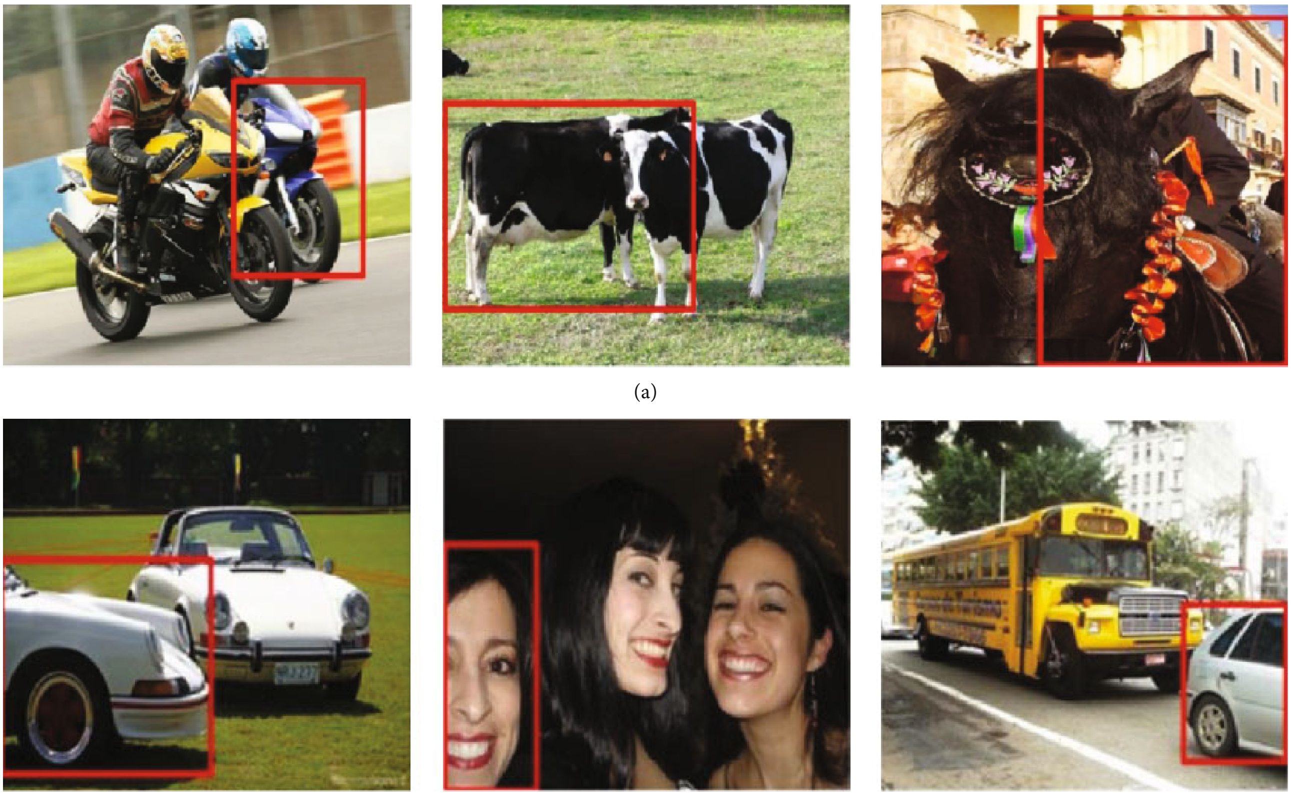
Object detection is used widely in smart cities including safety monitoring, traffic control, and car driving. However, in the smart city scenario, many objects will have occlusion problems. Moreover, most popular object detectors are often sensitive to various real-world occlusions. This paper proposes a feature-enhanced occlusion perception object detector by simultaneously detecting occluded objects and fully utilizing spatial information. To generate hard examples with occlusions, a mask generator localizes and masks discriminated regions with weakly supervised methods. To obtain enriched feature representation, we design a multiscale representation fusion module to combine hierarchical feature maps. Moreover, this method exploits contextual information by heaping up representations from different regions in feature maps. The model is trained end-to-end learning by minimizing the multitask loss. Our model obtains superior performance compared to previous object detectors, 77.4% mAP and 74.3% mAP on PASCAL VOC 2007 and PASCAL VOC 2012, respectively. It also achieves 24.6% mAP on MS COCO. Experiments demonstrate that the proposed method is useful to improve the effectiveness of object detection, making it highly suitable for smart cities application that need to discover key objects with occlusions.

1. Introduction

The development of smart cities is inseparable from the two key technologies of the Internet of Things (IoT) and artificial intelligence (AI). Although the IoT technology [1–3] has developed well in recent years, the effect needs to be improved. Therefore, the effective combination of IoT [4–6] and AI technology has become a major challenge today. Object detection based on neurocomputing, which is one of the tasks of smart cities, has been well studied in recent years, since it is a biologically inspired AI application. The goal of object detection is to localize an object of a predefined category. Recent state-of-the-art object detectors could be split into two main categories: the region-based detectors [7–9] and the regression-based detectors [10, 11]. These models have made a big contribution to object detection development.

Nevertheless, the robustness of object detection is still worth study. In practical application for smart cities, the network needs to detect some perturbed images. We can classify these images into two categories: (1) some parts of object are occluded (Figure 1(a)) and (2) object is beyond the picture boundary (Figure 1(b)). The occlusion occurs commonly in multiple object image, and the foreground objects always mask some features of object behind it. For another images, some object features are lost since the object is beyond the picture boundary. These images are called hard examples in this paper.

Because the hard examples have stronger transfer ability, the network is difficult to learn discriminative features for detection. Therefore, it is necessary to enhance the network's ability of mining useful information from perturbed images. However, it is difficult to train a robust model only given normal dataset. A solution to solve this problem is to integrate hard example mining in the training stage [12, 13], but it cannot solve the essential problem. An effective method is to generate hard examples from the detection



(a)

(b)

Figure 1: Illustration of hard examples in real situation.

dataset. Some works [14–18] have been devoted to addressing example generation problems. One useful solution is to generate realistic looking images by generative adversarial networks [14–17]. Another way is to generate masks on the original images directly. For instance, [18] generates hard occlusion examples for object detection during training.

In this paper, we propose a novel approach that better addresses the above issues. Our goal is to tackle the lack of hard examples and exploit more feature representation for the object with occlusions as far as possible.

There are three motivations for our study. Firstly, consider improving the network robustness in a balanced dataset, a deep mask generator is proposed in our approach to generate hard positive examples. Secondly, the weakly supervised object localization is crucial for a mask generator to obtain mask accurately. Locating in a discriminative region can let the generator know which region of object is possibly occluded in real life. Thirdly, the richness of the feature map is important for a mask generator to obtain hard examples. Therefore, the multiscale feature map can contain far more information and detail. To sum up, our main contributions are as follows:

(1) We introduce an end-to-end approach which can improve the robustness of the object detector and achieve competitive performance in object detection task

(2) We proposed a mask generator which uses weakly supervised location to generate pixel-wise masks and show that the mask generator is helpful to obtain more realistic hard examples during training

(3) We design a multiscale feature fusion module and context-aware information module to exploit abundant spatial information and demonstrate that these can improve the richness of feature map

2. Related Work

In recent years, many object detectors based on region perform classification and bounding box regression on each proposal region. Compared with the regression-based detectors, the detection accuracy and location accuracy of the region-based detectors are superior. Following the pioneer region-based object detector R-CNN, Fast R-CNN [7] increases model's accuracy by adding a RoI-pooling layer. In Faster R-CNN [9], region proposal network (RPN) generated more precise proposals than selective search. Our work builds on Faster R-CNN, which is a remarkable end-to-end method.

2.1. Multiscale Representation Concatenation. Recently, many significant works presented that multiscale feature concatenation is vital for object detection [19, 20]. For example, ION [21] extracts some feature descriptors and combines them after RoI-pooling layer [8]. HON [22] aggregates high-

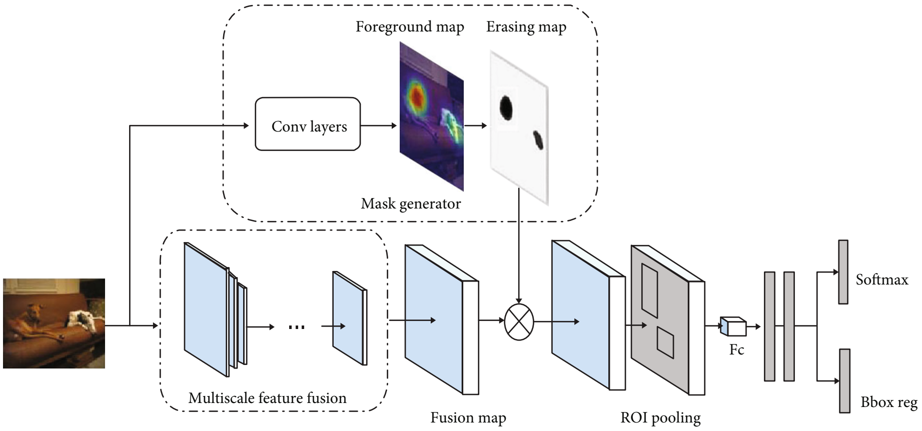


Figure 2: The overall architecture of our method. Mask generators are added before RoI-pooling.

level semantic features and shallow detail features through reverse connections. HyperNet [23] incorporates hierarchical feature maps and compresses them into a fixed-size space. In order to perform detection at multiple scales, RON [24] uses reverse connection to predicts objects at different layers, and FPN [25] presented a clean and simple framework for building feature pyramids inside ConvNets; it archived good a result and trained using the COCO trainval35k dataset. EfficientDet [26] proposes a weighted bidirectional feature pyramid network (BiFPN), which allows easy and fast multiscale feature fusion. AugFPN [27] incorporates consistent supervision, residual feature augmentation, and Soft RoI selection, which can significantly improve the baseline approach on challenging MS COCO datasets. Chu et al. [28] used an ensemble object detection system to combine the relationships between objects and context features based on global scenes. Res2Net [29] construct a hierarchical residual-like connections which represents multiscale features at a granular level. [30] proposes a multiconnection module to fuse multigrained information to enhance feature representation. VDNets [31] uses a feature fusion method based on the attention mechanism to make full use of multiscale feature information. In summary, the multiscale feature concatenation is to improve the richness of the feature map, so as to improve the detection accuracy of the detector. Different from these works, our work uses a new convolution layer with higher semantic information and trained using the trainval dataset for COCO.

2.2. Hard Example Mining. Some works focus on how to better use data for improve the performance of the model. [32] enhances the representations for small objects using perceptual GAN. One direction is to insert hard examples in the training stage. For example, inspired by bootstrapping, OHEM [13] improves the capacity of object detection by reranking and training hard examples. This work is further extended by Wang et al. [18], which generates hard examples by adversarial learning for object detection. AOFD [33] pays attention to increase the capacity of face detection by generating occlusion-like face features and proposes a multitask training method. [34] studies online selection of hard examples for minibatch SGD methods. [35] independently selects the positive and negative examples with a stochastic strategy of the training set, and [36] uses a ranking loss function to find hard negative patches from a large set. C-RPNs [37] adopt multiple stages to mine hard samples.

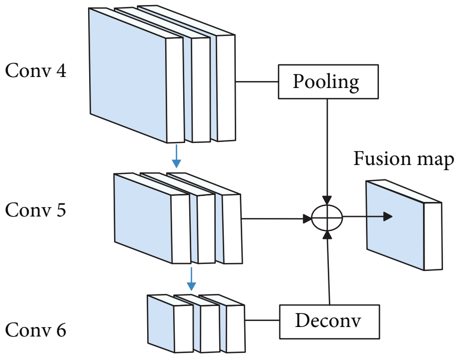


Figure 3: The architecture of multiscale feature fusion module.

Similar to these works, our method improves the capacity of detector by adding hard examples during training. However, these works may have collapsed and instable problems. Therefore, our model generates hard examples by masking the discriminative parts of object, rather than adversarial learning. These discriminative parts are accurately located by weakly supervised learning.

2.3. Weakly Supervised Object Localization. CNNs are proved to have great performance on object location. Recently, there are many works exploring weakly supervised localization (WSL) using CNNs. Huang et al. [38] pays attention to improve the quality of initialized object locations. [39] proposes a self-taught model to localize the stronger responsive regions when artificially masking. [40] contributes a multifold multiple instance learning procedure to localize objects with CNN features. To solve the WSL problem and improve the detection ability, [41] proposes a deep self-taught approach to localize more positive samples by retrain itself. [42] proposes to integrate the feature pyramid network (FPN) with convolutional neural network (CNN) for weakly supervised object localization. Hide-CAM [43] utilizes a hide

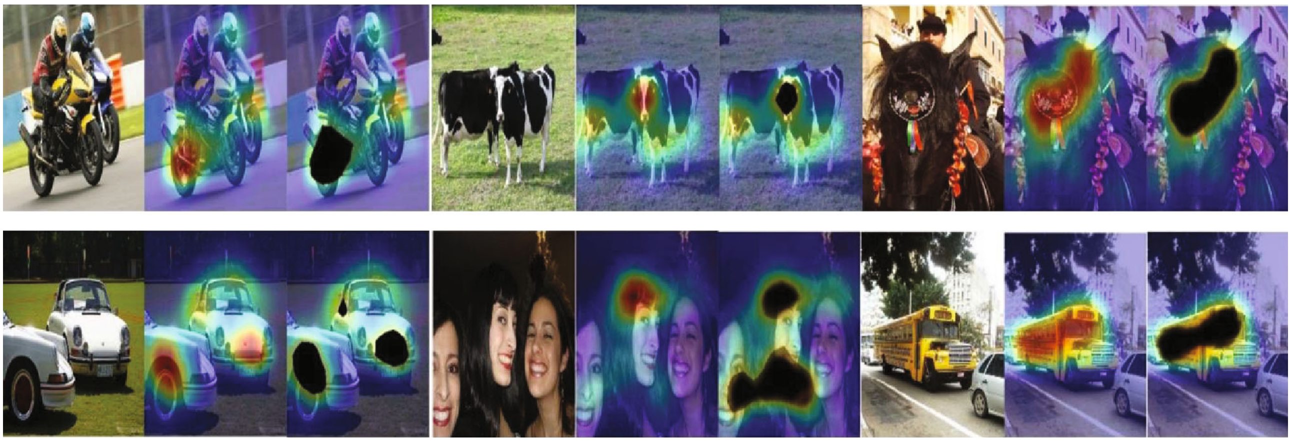


Figure 4: Selected examples of the discriminative object mined by weakly supervised object location and generate hard examples.

strategy to locate the most discriminative and the complementary object regions of the object. Oquab et al. [44] proposes global max pooling and demonstrate that object localization can be completed using the output of CNNs. Different from [44, 45] localizes the discriminative image regions base on the global average pooling. [46] trains a weakly supervised framework and to mine the entire region of object by randomly hiding patches. [46] is closely related to the adversarial erasing method [47]; this method also localizes dense regions by erasing some discriminative parts iteratively.

Similar to [45, 46], our work localizes the regions of discriminative features and masks these discriminative features to generate hard examples, which does not need any bounding boxes annotations to locate regions of features.

3. Network Architecture

As is shown in Figure 2, we use a multiscale feature concatenation architecture in feature exaction network. A mask generation branch is added after input layer, followed by an element-wise product layer. The aim of the branch is to let the network know which region of object is possibly occluded in real life. After that, the generated feature representation input into the RoI-pooling layer. Two fully connected (Fc) layers process each descriptor and produce two outputs: a class prediction and bounding box.

In this section, we first introduce feature extraction network which can extract richer information. Then, we describe the mask generator that products mask in a weakly supervised way.

3.1. Multi-Scale Representation Concatenation. For the feature extraction stage, multiscale representation concatenation is necessary when you want to get a more detailed feature map. Observing hierarchical feature map has different characteristics in convolutional neural networks (CNNs), we take a different use for these feature maps.

In our model, convolution layers 4, 5, and 6 are combined to obtain more feature detail (see Figure 3). After pooling and convolving convolution layer 5, layer 6 is obtained. The size of convolution layer 4 is double of the layer 5, and layer 6 is a half of convolution layer 5. To maintain the same resolution of multilevel maps, layer 4 resize to the same as layer 5 through a max pooling layer, and layer 6 resize to the same as layer 5 through a deconvolution layer for upsampling operation. For unify amplitudes from various levels, our method uses L2 normalize to different feature representation. Because the amplitudes from different levels are various, L2 normalize is important to representation concatenation. Therefore, the scale of final representation is 1/16 of the origin image scale, which is suitable for RoI-pooling layer.

Our method uses VGG-16 [48] as the pretrain network. To meet the output shape of the RoI-pooling layer, the final map's scale should be 7×7 pixels with 512 channels. In addition, it is the formal input for the next detection network (fc6). Therefore, the representation map will input to the RoI-pooling layer without any special operation; this process can guarantee feature map have more details.

3.2. Weakly Supervised Mask Generator. Even in large-scale datasets, it is difficult to sample all possible hard examples. We take a flexible approach to generate hard examples, rather than relying on data augmentation. The mask generator is used to find some distinguish areas by weakly supervised object location and generate various realistic masks. More specifically, it is only mask the discriminative part of the object in the training dataset. This effective way forces the model to learn feature which look like object even if object is incomplete. Note that we only apply the mask generator during training but not during testing.

3.3. Weakly Supervised Object Location. To localize these discriminative image regions, we use the network of Zhou et al. [45] to generate a class activation map (CAM). After learning a classification network, the CAM can represent the discriminative image regions for a particular class. In general, the classification network is initialized based on the AlexNet [49], GoogLeNet [50], and VGGNet [49]. In our work, we learn a classification network base on VGG-16 architecture. To generate a CAM for an image, global average pooling (GAP) is performed on the last convolutional feature maps. For a class, the GAP's output is the average of the last convolutional feature map in each unit. A weight corresponds to a class, CAM is the weight sum of these output values. Our final output is generated from the top 5 predicted categories for the input image.

Given an image I, denote $f_i(x, y)$ to be the last convolutional activation of unit i at spatial location (x, y). When perform global average pooling, the output F_k is

$$F_k = \sum_{x,y} f_i(x, y). \tag{1}$$

For class c, w_i^c is the weight of the last classification layer, which corresponds to unit i. w_i^c can consist of a $N \times M$ weight matrix of the classification layer, where M is the number of feature maps in the last convolution layer, and N is the number of categories. Thus, the class score is

$$S_c = \sum_{i}^{M} w_i^c F_c = \sum_{i}^{M} w_i^c \sum_{x,y} f_i(x, y) = \sum_{x,y} \sum_{i}^{M} w_i^c f_i(x, y). \tag{2}$$

It is obvious that w_i^c reflects the importance of F_k for class c. Then, the class activation map for class c is

$$CAM(c, I) = \sum_{i}^{M} w_i^c f_i(x, y). \tag{3}$$

Hence, for an image to c class, $\mathrm{CAM}(c, I)$ indicates the importance of the activation at spatial location (x, y).

Some examples of weakly supervised object location are shown in Figure 4.

3.4. Masking Strategy. We generate mask map X for an image with size $w \times h \times c$, where w, h are the length and width of the concatenation map, and c is the number of channels. $X_{x,y}$ is the pixel value for location (x, y) of the mask map, and each pixel value of X is compress to 0 or 1. Then, the values of the mask map are obtained by applying a hard threshold O to the CAM. If the pixel value of activation map M_i^c is greater than O, this location (x, y) belongs to discriminative region. Thus, $X_{x,y} = 0$, the value of corresponding spatial location will be drop out in all channels. To the contrary, the feature values of general region will be retained. Our strategy is to mine some strongly responsive areas in feature map and mask these distinguish regions precisely. This strategy is more accurate than dropping pixels randomly. Occluded samples will become the hard examples for training. Some examples are shown in Figure 4.

Now, we explicate our mask generator more formally. Let $D = (I_i, Y_i)_{i=1}^{N}$ be a training set including N images, and P_i is the mask regions for image i. Denote M_i^c is the activation map of image i for class c, which is generated by $\mathrm{CAM}(c, I_i)$, note that the class $c \in Y_i$. The $p_{i,x,y}$ is the pixel value for location (x, y) of the activation map M_i^c. Once the value of $p_{i,x,y}$ is greater than hard threshold O_m, the region of location (x, y) will be mined. Then, we mask the mined region and the new training data set D' is obtained. The procedure is detailed in Algorithm 1.

```
Input: training data $D = \{(I_i, Y_i)\}_{i=1}^{N}$, hard threshold $O_m$.
    Initialization: $I' = \varnothing, P_i = \varnothing (i = 1, \cdots, N)$.
    for $I_i$ in $D$ do
        Calculate $M_i^c$ by $\mathrm{CAM}(c, I_i)$ [29];
        for $p_{i,j}$ in $M_i^c$ do
            while $p_{i,x,y} > O_m$ do
                $P_i = P_i \cup \{p_{i,x,y}\}$;
            end while
        end for
Mask the mined regions from training image $I_i' = I_i \setminus P_i$;
end for
Output: a set of new training data $D' = \{(I_i', Y_i)\}_{i=1}^{N}$
```

Algorithm 1: Weakly supervised mask generator.

3.5. Context-Aware Information. Work [21] uses a recurrent neural network (RNN) to extract contextual information. To connect features from different contextual regions, Zeng et al. [51] through the gated bidirectional network for feature expression. Influenced by [52], this method extracts contextual information from different regions of the feature map. The difference is extract from the feature representation with richer information.

After RoI-pooling layer, we stack feature from regions of object feature and contextual information. At the first, the context region is default as one and a half times of region of interest (RoIs). The context region of fusion map feed to RoI-pooling layer and generate a fixed-length feature descriptor of size $7 \times 7 \times 512$. After that, the object region's descriptor is obtained. Our method combines these two descriptors through adding corresponding value at pixel level. This method omits additional layers to decrease dimension, so that improving model efficiency and reducing extra runtime.

3.6. Detection and Training

3.6.1. Region Generating. For region generating, the region proposal network (RPN) [9] is used to generate various boxes. In order to scan to various sized objects, we use 3 scales and 3 aspect ratios to generate various sized boxes. However, the RPN always generate many redundant region proposals. To reduce redundancies, nonmaximum suppression (NMS) is used to filter proposals. For a proposal, when the value of intersection-over-union (IoU) greater than threshold, the proposal will be deleted. Our method defines threshold as 0.7, the top rank three hundred proposals will be used in the next stage.

3.6.2. Object Detection. After generating proposed regions, the detection module needs to classify proposed regions into $K + 1$ categories ($K = 20$in PASCAL VOC database) and achieve bounding box regression. The previous module outputs an abundant pooled feature map. We make the maximum use of the pooled feature by two fully connected layers, then compute the per-class score with Softmax, and output an adjustment to the bounding box.

Table 1: Detection results from PASCAL VOC 2007 test task.

Method	M	C	H	mAP	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	Tv
FRCN [9]				73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
OHEM [13]				74.6	77.7	81.2	74.1	64.2	50.2	86.2	83.8	88.1	55.2	80.9	73.8	85.1	82.6	77.8	74.9	43.7	76.1	74.2	82.3	79.6
RON [24]				75.4	78.0	82.4	76.7	67.1	56.9	85.3	84.3	86.1	55.5	80.6	71.4	84.7	84.8	82.4	76.2	47.9	75.3	74.1	83.8	74.5
ION [21]				75.6	79.2	83.1	77.6	65.6	54.9	85.4	85.1	87.0	54.4	80.6	73.8	85.3	82.2	82.2	74.4	47.1	75.8	72.7	84.2	80.4
HyperNet [23]				76.3	77.4	83.3	75.0	69.1	62.4	83.1	87.4	87.4	57.1	79.8	71.4	85.1	85.1	80.0	79.1	51.2	79.1	75.7	80.9	76.5
Ours	✓			75.4	78.1	82.0	75.1	66.2	60.4	83.1	87.7	87.9	55.4	81.4	68.2	86.7	84.9	78.3	79.0	49.2	77.1	71.2	82.0	73.1
Ours			✓	76.9	77.5	84.1	75.0	67.6	60.9	85.4	87.8	88.1	59.5	84.1	72.8	87.9	87.5	78.3	78.9	49.0	77.1	75.3	85.1	76.0
Ours	✓		✓	77.0	77.5	83.3	76.2	67.5	60.9	87.0	88.3	87.2	61.0	83.2	73.9	85.3	85.8	80.6	79.3	49.5	76.2	77.5	84.4	75.2
Ours		✓	✓	77.2	78.6	80.9	78.5	67.0	63.8	86.4	87.9	87.7	62.1	83.0	73.9	85.3	86.8	77.8	79.2	50.2	76.7	76.3	85.5	76.1
Ours	✓	✓	✓	77.4	78.3	82.9	76.9	68.5	62.3	87.2	88.6	88.0	58.9	85.0	72.6	86.3	87.1	79.1	79.2	50.8	77.8	77.9	84.4	76.2

Legend: M: presentation fusion, C: context-information, H: mask generator.

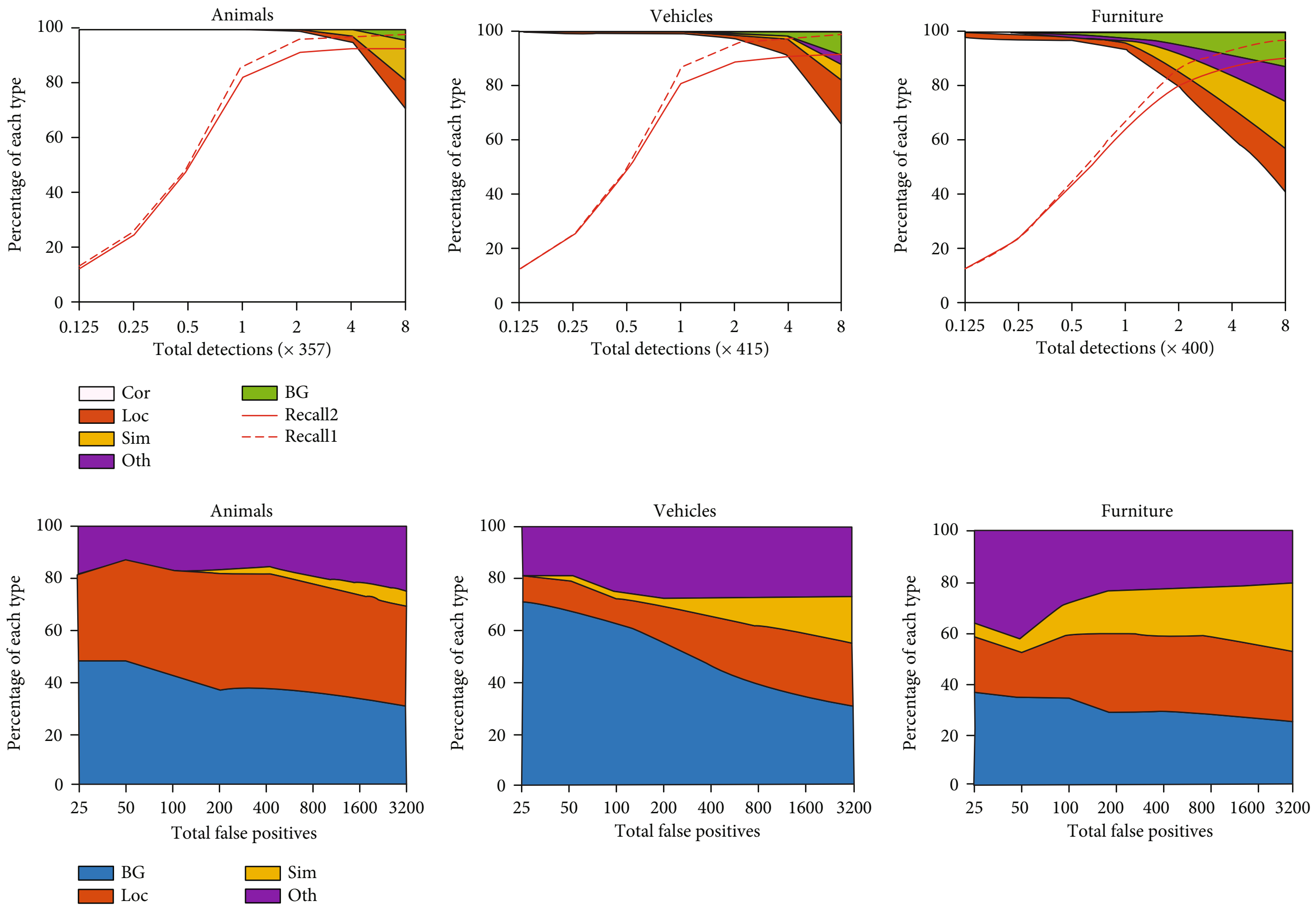


Figure 5: Visualization of the performance of our model on animals, vehicles, and furniture classes in the PASCAL VOC 2007 test.

3.6.3. Joint Training. For training way, this paper adopts an end-to-end way to jointly optimize the loss function. During training, the detection network and RPN are combined into one network. For per iteration of training, RPN generate a set of region proposals for detection network to predict classification scores and regress locations. This process is the precompute of forward propagation. In the RPN stage, we give positive label to a box which intersection-over-union (IoU) higher than 0.7 or highest IoU with a ground-truth box. On the contrary, box which IoU lowers than 0.3 will be given negative label. In backward propagation, loss of two networks generate gradient signal. To achieve this process, the multitask loss function is defined as

$$L(p_i, t_i) = \frac{1}{N_{\text{cls}}} \sum_i L_{\text{cls}}(p_i, p_i^*) + \lambda \frac{1}{N_{\text{reg}}} p_i^* L_{\text{reg}}(t_i, t_i^*), \tag{4}$$

where p_i is the predicted probability of positive box. The value of p_i^* indicates the ground truth label of anchor i. So, $L_{\text{cls}}(p_i, p_i^*)$ is classification loss function, and $L_{\text{reg}}(t_i, t_i^*)$ is regression loss function:

$$L_{\text{cls}}(p_i, p_i^*) = -\log\left[p_i^* p_i + (1 - p_i^*)(1 - p_i)\right], \tag{5}$$

$$L_{\text{reg}}(t_i, t_i^*) = R(t_i - t_i^*), \tag{6}$$

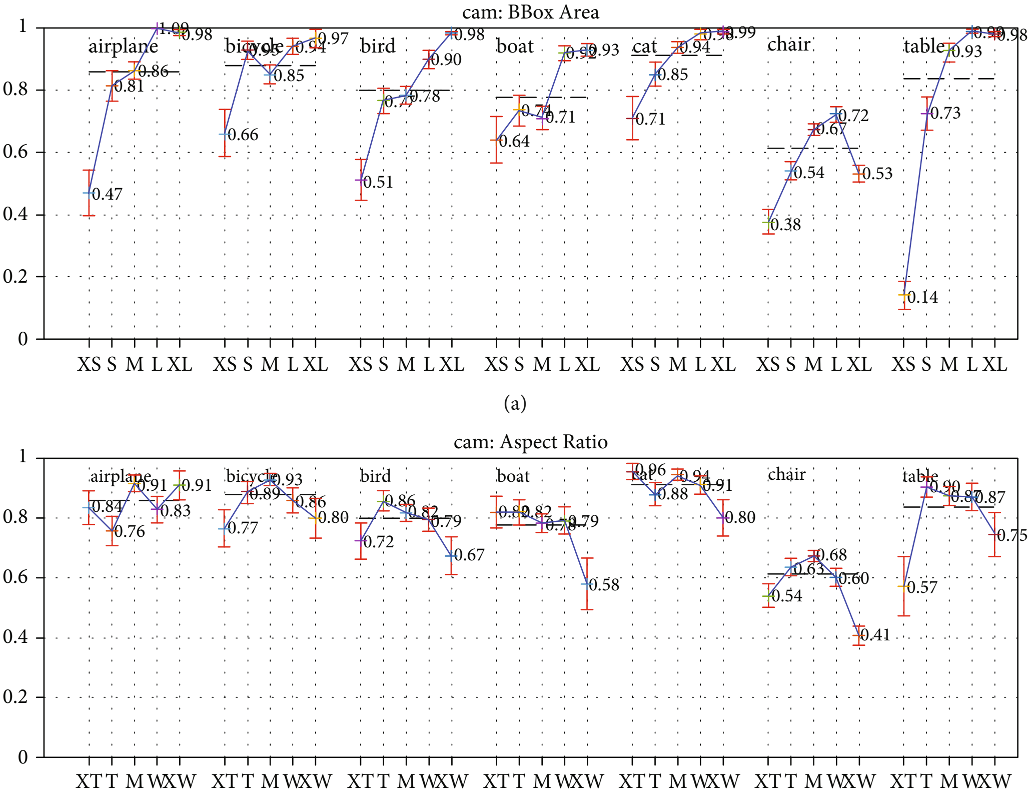
where R is the robust loss function smooth L1 [2]:

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2, |x| < 1, \\ |x| - 0.5, \text{otherwise}. \end{cases} \tag{7}$$

4. Experiment

We conduct experiments based on three detection datasets: PASCAL VOC 2007, PASCAL VOC 2012 [53], and COCO [54]. For PASCAL VOC, the union set of PASCAL VOC 2007 trainval and 2012 trainval is used to train all networks, and PASCAL VOC 2007 and PASCAL VOC 2012 are used to verify different networks, respectively. For MS COCO, we trained networks on the trainval set and test on the test-dev evaluation server. The results are measured by mean average precision (mAP).

4.1. Experimental Setup. Our networks are design based on the VGG-16 framework [48] and Fast R-CNN baseline. The max size of the longest side is 1024 pixels. The test image scale is the same size as train image. For solver parameter, stochastic gradient descent (SGD) as the iterative method used to optimize objective function. We set the initial learning rate to 0.001, and decreased by a factor of 10 times after every 50,000 iterations. The weight decay is set to 0.0005



(a)

(b)

Figure 6: Sensitivity and impact of different object characteristics on the PASCAL VOC 2007 test set.

and momentum to 0.9, so that the learning rate is 0.001 for the first 50 k minibatches and 0.0001 for the next 20 k. VGG-16 is the pretrained model, which was firstly pretrained on the ImageNet benchmark; after that, it was fine-tuned on detection benchmark.

4.2. PASCAL VOC 2007 Test Set. For the PASCAL VOC 2007 detection task, we compare our models with the state-of-the-art detectors (see Table 1). All parameters are set as Faster RCNN except for the image size. Our full model with all three modules improves the performance to 77.4%; the final result gives 4.2% boost upon Faster R-CNN. The bounding box voting [55] is also a useful mechanism to improve detection performance.

To understand the performance of our model in detail, we use the detection analysis tool from [56]. As shown in Figure 5, the top row shows that our model can detect various object categories with high quality (large white area). The majority of its confident detections are correct. The solid red line and dashed red line reflect the change of recall with strong and weak criteria, respectively. The bottom row shows the distribution of the top-ranked false positive types. Figure 6 demonstrates that our model is robust to different object sizes and aspect ratios. Compare with other state-of-the-art detectors, our model achieves better performance at three aspects: (1) The location error (Loc) of our model is less; this means that our model can localize objects better. (2) Our model with less false positives caused by confusion with similar categories, because it can exploit more object's feature by training with hard examples. (3) Our model with less false positives caused by confusion with background, since our model can learn more richness object's feature through multiscale feature fusion.

4.3. PASCAL VOC 2012 Test Set. We also test our networks on PASCAL VOC 2012 and submit results to the public evaluation server (anonymous URL: http://host.robots.ox.ac.uk:8080/anonymous/NG67QK.html.). Our models are trained on set of VOC 2007+2012, but without VOC 2007 test set. Table 2 show our network obtains 74.3% mAP.

4.4. MS COCO Test Set. In addition to PASCAL VOC, we present more results on the Microsoft COCO and got reports from the public evaluation server (anonymous URL: https://competitions.codalab.org/my/competition/submission/461101/stdout.txt). As shown in Table 3, our network achieves 24.6% mAP, which is greater than Faster R-CNN. It is noted that when IoU is 0.5 : 0.95, the mAP of our network is poorer than DSSD321, SSD300, and ION, but when the area is small, the result is better. So, our network is good at detection of small object, due to using the multiscale feature fusion module. Note that the feature in DSSD321 is extracted by Residual-101, but our network is by VGG-16.

5. Ablation Analysis

To study the impact of multiscale representation and context-aware, we conduct some comparative experiments with

Table 2: Detection results from PASCAL VOC 2012 test task.

Method	M	C	H	mAP	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	Tv
FRCN [9]				70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
OHEM [13]				71.9	83.0	81.3	72.5	55.6	49.0	78.9	74.7	89.5	52.3	75.0	61.0	87.9	80.9	82.4	76.3	47.1	72.5	67.3	80.6	71.2
RON [24]				73.0	85.4	80.6	71.9	56.3	49.8	80.6	76.8	88.2	53.6	78.1	60.4	86.4	81.5	83.8	79.4	48.6	77.4	67.7	83.4	69.5
HyperNet [23]				71.4	84.2	78.5	73.6	55.6	53.7	78.7	79.8	87.7	49.6	74.9	52.1	86.0	81.7	83.3	81.8	48.6	73.5	59.4	79.9	65.7
VDNets [31]				73.2	85.1	82.4	73.6	57.7	61.2	79.2	77.1	85.5	54.9	79.8	61.4	87.1	83.6	81.7	77.9	45.6	74.1	64.9	80.3	73.1
Ours	✓			71.5	84.7	79.9	74.0	57.1	53.3	77.9	78.9	89.3	50.7	74.9	53.0	87.6	79.5	80.5	81.9	48.1	74.9	58.3	80.8	64.9
Ours			✓	72.9	84.9	80.2	73.9	58.1	54.7	80.0	78.8	89.5	51.9	78.7	56.7	88.4	83.5	83.3	82.7	48.4	76.4	60.7	81.0	68.1
Ours	✓		✓	72.8	84.9	81.4	72.4	58.1	55.6	79.9	79.1	88.9	51.8	78.4	56.8	87.7	82.4	81.5	82.2	50.2	74.7	61.1	82.1	65.9
Ours		✓	✓	74.3	85.6	82.4	74.5	60.2	55.8	81.5	80.8	90.4	53.3	80.4	58.4	89.4	84.5	84.7	82.4	49.0	77.7	64.1	82.3	68.2
Ours	✓	✓	✓	73.2	84.5	81.4	73.8	59.5	54.4	80.6	79.5	88.5	51.9	79.0	59.3	87.9	82.6	82.4	82.2	50.3	75.4	61.2	82.1	67.0

Legend: M: presentation fusion, C: context-information, H:mask generator.

Table 3: Detection results from MS COCO test task.

Method	Training data	mAP, IoU:			mAP, area:		
		O.5 : 0.95	0.5	0.75	S	M	L
FRCN	trainval	21.9	42.7	—	—	—	—
OHEM	trainval	22.6	42.5	22.2	5.0	23.7	37.9
ION	train	23.6	43.2	23.6	6.4	24.1	38.3
SSD300	trainval35k	25.1	43.1	25.8	6.6	25.9	41.4
DSSD321	trainval135K	28.0	46.1	29.2	7.4	28.1	47.6
Ours	trainval	24.6	46.4	23.5	9.3	27.9	36.4

Table 4: Ablation analysis for three modules.

Module	Our model					
M: multiscale feature concatenation	✓		✓			✓
C: context-aware information	✓	✓			✓	
H: hard example generation	✓	✓	✓	✓		
mAP (%)	77.4	77.2	77.0	76.9	75.0	75.4

Faster R-CNN when we remove one addition one after another. As shown in Table 4, our method improves baseline from 73.2% to 76.9% when adding a mask generator. Multiscale representation and context-aware further improves the Faster R-CNN mAP to 77.0% and 77.2%, respectively. But, more importantly, our model achieves a mAP of 77.4% with the three additions. Note that we use same parameter settings and image size to guarantee a fair comparison. All models are trained on the union set of VOC 2007 and VOC 2012 and tested on VOC 2007 test set.

5.1. Analysis for Hard Examples Generation

5.1.1. What Is the Best Value of Hard Threshold O? We propose a hard threshold O to measure the mask degree. With suitable O, only the pixel of the discriminative region is selected and masked, thus, the hard threshold O is crucial. We find that the detector would be obstructed by mask if the mask region is too large; this happens because the network saw few discriminative pixels. Oppositely, it would be useless if it too small. To find a suitable hard threshold O, we conduct a series of experiments only with a mask generator branch.

Table 5 gives a brief result of our experiments. Let R is the value to 256 colors in class activation map, region with high value of R will be highlight as the discrimination. For a location in class activation map, the greater the value of R, the greater the response of the class. The pixel will be masked, provided that R is greater than O. Thus, setting a high threshold means a lower degree of mask. When O is 170, our detector achieves competitive results (76.9% mAP). But, when we set O to 160 or 180, the results are not very competitive.

The reason is that the mask generator cannot product useful hard examples with a too high value of O. Nevertheless, it would be break detector if the value of O is too low. According to results, two keys can be summarized: (1) The hard threshold O is vital to generate useful mask. (2) Occluding one-third area ($O = 170$) of feature map can product a reasonable mask.

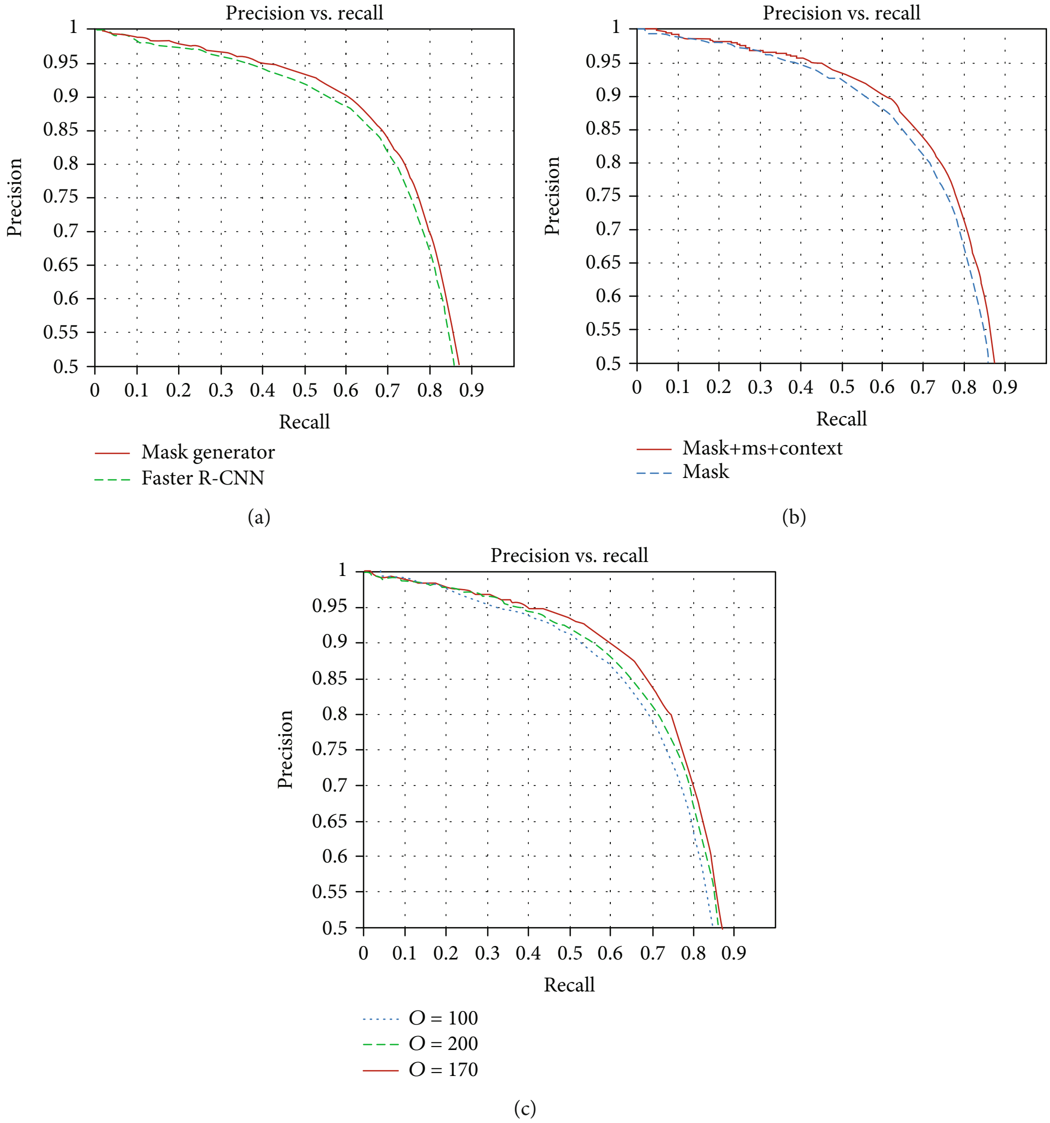
Table 5: Influence of different value of hard threshold O.

Hard threshold O	100	150	170	200
mAP(%)	74.0	75.6	76.9	76.3

5.1.2. Do Mask Generator Help? To prove that the mask generator is useful in the object detection network, we conduct a set of experiments which compare it with the baseline. As shown in Figure 7(a), our method achieves a better result. Furthermore, with multiscale representation concatenation and contextual information module, the performance becomes well (see Figure 7(b)). We also use three types of mask area to get different hard examples for training;, the performance for different mask area is shown in Figure 7(c). Our method performs better when the hard threshold O is 170.

5.2. Analysis for Multiscale Representation Concatenation. To validate the effectiveness of the multiscale representation concatenation, we design a series of experiments and study why the detection performance is affected by representation concatenation. To better understand the importance of multiscale feature fusion, we have removed the mask generator branch.

Our network obtains high-level semantic information through fusion higher convolution layer 6. However, does the convolution layer 6 really work? This paper designed a set of experiments to verify this issue. Firstly, we train a model detect from single layer 5, which achieved 70.0% mAP. Secondly, we trained model detect from layers 3, 4, and 5 and 4, 5, and 6, respectively. We evaluate the detection performance with different layers at the region proposal number is 100 in Table 6. Fusing layers 3, 4, and 5 gets 73.1% mAP, and fusing layer 4, 5 and 6 gets the best detection result (mAP = 75.4%). These detection results also show the



(a)

(b)

(c)

Figure 7: Precision versus recall for mask generator.

Table 6: Influence of different layer concatenation strategies and normalization.

Concatenation from layer				Normalization	
3	4	5	6	L2 normalization	Local response normalization
		✓		70.0	
✓	✓	✓		73.1	70.0
	✓	✓	✓	75.4	67.3

effectiveness of convolution layer 6. Therefore, the new convolution layer 6 is useful for the fusion feature map, since it has richer semantic information compared with layer 5.

As Table 6 has shown, we use different methods to normalize feature map; the first one is L2 normalize; the second one is local response normalization (LRN) [49]. The last model achieved 75.4% mAP with L2 normalization and 67.3% mAP with local response normalization (LRN). So, the L2 normalization is more effective.

5.3. Analysis for Context Information. The context information is very important for feature extraction. Therefore, we design a set of experiments to verify the necessary of context information. As shown in Figure 7, our model with contextual information achieved better result than baseline. There are two keys should be concluded: (1) embedding contextual information is a good way to improve detection performance and (2) the sum operation at pixel level is vital to embed operation.

6. Conclusion

This paper proposed a novel architecture to solve the object occluded problem for object detection. We aim to learn an object detector that is robust to different occlusions. To achieve this goal, we propose an end-to-end framework that generate hard examples during training and achieving competitive performance in the object detection task.

To learn object models that are invariant to occlusions, we proposed a mask generator which uses weakly supervised location to generate pixel-wise masks and show that mask generator is helpful to obtain more realistic hard examples during training. To exploit more spatial information and improve the richness of feature map, we design a novel multiscale representation concatenation model for the feature extraction stage and add the context-aware module in the region proposal network. Our method obtains comparable results, 77.4% mAP and 74.3% mAP on PASCAL VOC 2007 and VOC 2012, respectively. It also achieves 24.6% mAP on MS COCO. Our studies demonstrate that hard examples and rich spatial information is vital for object detection, promoting smart cities to solve the occlusion problem of object detection.

Data Availability

The data (PASCAL VOC 2007, PASCAL VOC 2012, and MS COCO) supporting this study are from previously reported studies and datasets, which have been cited. These prior studies (and datasets) are cited at relevant places within the text as references [46, 47].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of China (Grant Nos. 62071098, 61941113, and 61971033), Sichuan Science and Technology Program (Grant Nos. 2019YFG0191 and 2021YFG0307), and Chengdu Science and Technology Project (Grant No. 2019-YF05-01830-SN).

References

[1] X. Liu and X. Zhang, "NOMA-based resource allocation for cluster-based cognitive industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5379–5388, 2020.

[2] X. Liu, M. Jia, X. Zhang, and W. Lu, "A novel multichannel internet of things based on dynamic spectrum sharing in 5G communication," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 5962–5970, 2019.

[3] M. Jia, Z. Yin, Q. Guo, G. Liu, and X. Gu, "Downlink design for spectrum efficient IoT network," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3397–3404, 2018.

[4] X. Liu, X. Zhai, W. Lu, and C. Wu, "QoS-guarantee resource allocation for multibeam satellite industrial internet of things with NOMA," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2052–2061, 2021.

[5] J. Xu, L. Wang, Y. Shen et al., "Family-based big medical-level data acquisition system," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2321–2329, 2019.

[6] X. Liu and X. Zhang, "Rate and energy efficiency improvements for 5G-based IoT with simultaneous transfer," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 5971–5980, 2019.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, 2014.

[8] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Santiago, Chile, 2015.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[10] J. Redmon, S. Divvala, R. 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.

[11] M. Najibi, M. Rastegari, and L. S. Davis, "G-cnn: an iterative grid based object detector," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2369–2377, Las Vegas, NV, USA, 2016.

[12] M. Jia, X. Gu, Q. Guo, W. Xiang, and N. Zhang, "Broadband hybrid satellite-terrestrial communication systems based on cognitive radio toward 5G," *IEEE Wireless Communications*, vol. 23, no. 6, pp. 96–106, 2016.

[13] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 761–769, Las Vegas, NV, USA, 2016.

[14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2, NIPS'14, MIT Press*, pp. 2672–2680, Cambridge, MA, USA, 2014.

[15] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, https://arxiv.org/abs/1710.10196.

[16] R. Huang, S. Zhang, T. Li, and R. He, "Beyond face rotation: global and local perception gan for photorealistic and identity preserving frontal view synthesis," in *2017 IEEE International Conference on Computer Vision (ICCV), IEEE Computer Society*, pp. 2458–2467, Los Alamitos, CA, USA, 2017.

[17] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society*, pp. 2242–2251, Los Alamitos, CA, USA, 2017.

[18] X. Wang, A. Shrivastava, and A. Gupta, "A-fast-rcnn: hard positive generation via adversary for object detection," 2017, https://arxiv.org/abs/1704.03414.

[19] J. Xu, Y. Tian, H. Wu, B. Luo, and J. Guo, "You only move once: an efficient convolutional neural network for face detection," *IEEE Access*, vol. 7, pp. 169528–169536, 2019.

[20] J. Xu, W. Wang, H. Wang, and J. Guo, "Multi-model ensemble with rich spatial information for object detection," *Pattern Recognition*, vol. 99, p. 107098, 2020.

[21] S. Bell, C. Zitnick, K. Bala, and R. Girshick, “Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society*, pp. 2874–2883, Los Alamitos, CA, USA, 2016.

[22] J. Wang, X. Tao, M. Xu, Y. Duan, and J. Lu, “Hierarchical objectness network for region proposal generation and object detection,” *Pattern Recognition*, vol. 83, pp. 260–272, 2018.

[23] T. Kong, A. Yao, Y. Chen, and F. Sun, “Hypernet: towards accurate region proposal generation and joint object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society*, pp. 845–853, Los Alamitos, CA, USA, 2016.

[24] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, “Ron: Reverse connection with objectness prior networks for object detection,” 2017, https://arxiv.org/abs/1707.01691.

[25] T.-Y. Lin, P. Dollr, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” 2016, https://arxiv.org/abs/1612.03144.

[26] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: scalable and efficient object detection,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10778–10787, Seattle, WA, USA, 2020.

[27] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan, “AugFPN: improving multi-scale feature learning for object detection,” in *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 12592–12601, Seattle, WA, USA, 2020.

[28] W. Chu and D. Cai, “Deep feature based contextual model for object detection,” 2016, https://arxiv.org/abs/1604.04048.

[29] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, “Res2net: a new multi-scale backbone architecture,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 652–662, 2021.

[30] Y. Li, H. Zheng, Z. Yan, and L. Chen, “Detail preservation and feature refinement for object detection,” *Neurocomputing*, vol. 359, pp. 209–218, 2019.

[31] M. K. Ebrahimpour, J. Li, Y.-Y. Yu et al., “Ventral-dorsal neural networks:object detection via selective attention,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, 2019.

[32] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, “Perceptual generative adversarial networks for small object detection,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society*, pp. 1951–1959, Los Alamitos, CA, USA, 2017.

[33] Y. Chen, L. Song, and R. He, “Adversarial occlusion-aware face detection,” 2017, https://arxiv.org/abs/1709.05188.

[34] I. Loshchilov and F. Hutter, “Online batch selection for faster training of neural networks,” 2015, https://arxiv.org/abs/1511.06343.

[35] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, and F. Moreno-Noguer, “Fracking deep convolutional image descriptors,” 2014, https://arxiv.org/abs/1412.6537.

[36] X. Wang and A. Gupta, “Unsupervised learning of visual representations using videos,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2794–2802, Santiago, Chile, 2015.

[37] D. Yang, Y. Zou, J. Zhang, and G. Li, “C-rpns: promoting object detection in real world via a cascade structure of region proposal networks,” *Neurocomputing*, vol. 367, pp. 20–30, 2019.

[38] D.-A. Huang, L. Fei-Fei, and J. C. Niebles, “Connectionist temporal modeling for weakly supervised action labeling,” 2016, https://arxiv.org/abs/1607.08584.

[39] L. Bazzani, A. Bergamo, D. Anguelov, and L. Torresani, “Self-taught object localization with deep networks,” in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE Computer Society*, pp. 1–9, Los Alamitos, CA, USA, 2016.

[40] R. G. Cinbis, J. Verbeek, and C. Schmid, “Weakly supervised object localization with multi-fold multiple instance learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 189–203, 2017.

[41] Z. Jie, Y. Wei, X. Jin, J. Feng, and W. Liu, “Deep self-taught learning for weakly supervised object localization,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4294–4302, Honolulu, HI, USA, 2017.

[42] X. Tang, Y. Song, and Y. Zhang, “Feature fusion for weakly supervised object localization,” in *2018 Chinese automation congress (CAC)*, pp. 2548–2553, Xi'an, China, 2018.

[43] J. Xu, S. Sheng, H. Wei, and J. Guo, “Hide-cam: finding multiple discriminative regions in weakly supervised location,” *IEEE Access*, vol. 7, pp. 130590–130598, 2019.

[44] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Is object localization for free?- weakly-supervised learning with convolutional neural networks,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 685–694, Boston, MA, USA, 2015.

[45] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, Las Vegas, NV, USA, 2016.

[46] K. K. Singh and Y. J. Lee, “Hide-and-seek: forcing a network to be meticulous for weakly-supervised object and action localization,” 2017, https://arxiv.org/abs/1704.04232.

[47] Y. Wei, J. Feng, X. Liang, M. Cheng, Y. Zhao, and S. Yan, “Object region mining with adversarial erasing: a simple classification to semantic segmentation approach,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6488–6496, Honolulu, HI, USA, 2017.

[48] K. Simonyan and A. Zisserman, “Very deep convolutional networks for largescale image recognition,” 2014, https://arxiv.org/abs/1409.1556.

[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 1, NIPS'12, Curran Associates Inc.*, pp. 1097–1105, USA, 2012.

[50] C. Szegedy, W. Liu, Y. Jia et al., “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, Boston, MA, USA, 2015.

[51] X. Zeng, W. Ouyang, B. Yang, J. Yan, and X. Wang, “Gated bidirectional cnn for object detection,” in *Computer Vision–ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., pp. 354–369, Springer International Publishing, Cham, 2016.

[52] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, “A unified multi-scale deep convolutional neural network for fast object detection,” 2016, https://arxiv.org/abs/1607.07155.

[53] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: a retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[54] T.-Y. Lin, M. Maire, S. Belongie et al., "Microsoft coco: common objects in context," 2014, https://arxiv.org/abs/1405.0312.

[55] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware cnn model," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1134–1142, Santiago, Chile, 2015.

[56] D. Hoiem, Y. Chodpathumwan, and Q. Dai, "Diagnosing error in object detectors," in *Proceedings of the 12th European Conference on Computer Vision-Volume Part III, ECCV'12, Springer-Verlag*, pp. 340–353, Berlin, Heidelberg, 2012.