International Journal of Machine Learning and Cybernetics (2019) 10:3155–3166
https://doi.org/10.1007/s13042-019-01006-4

ORIGINAL ARTICLE

Enhance the recognition ability to occlusions and small objects with Robust Faster R-CNN

Tao Zhou[1] · **Zhixin Li**[1] · **Canlong Zhang**[1]

Received: 29 March 2019 / Accepted: 21 August 2019 / Published online: 26 August 2019

Abstract
Recognizing objects with vastly different size scales and objects with occlusions is a fundamental challenge in computer vision. This paper addresses this issue by proposing a novel approach denoted as Robust Faster R-CNN for detecting objects in multi-label images. Robust Faster R-CNN employs a cascaded network structure based on the Faster R-CNN architecture to extract features from objects with different size scales. However, the proposed design provides greater robustness than Faster R-CNN by replacing the RoIPooling operation with RoIAligns to eliminate the harsh quantization conducted by RoIPooling, and we design a multi-scale RoIAligns operation by adding multiple pool sizes for adapting the detection ability of the network to objects with different sizes. Furthermore, we combine an adversarial network with the proposed network to generate training samples with occlusions significantly affecting the classification ability of the model, which improves its robustness to occlusions. Experimental results for the PASCAL VOC 2012 and 2007 datasets demonstrate the superiority of the proposed object detection approach relative to several state-of-the-art approaches.

Keywords Object detection · Robust Faster R-CNN · Multi-cascaded network · Adversarial network · Feature fusion

1 Introduction

Object detection is one of the fundamental problems in computer vision that has been substantially addressed due to the great advances in deep learning over the past few years. It is well known that prevalent object detectors mostly regard detection as a problem of classifying candidate boxes[4, 5, 17]. This has led to the increasingly successful of the application of CNN(convolutional neural networks) in image recognition tasks [18, 25–27]. As a result, an increasing number of novel object detection methods based on CNNs [2, 10, 19] have been proposed. These structurally diverse frameworks have improved the accuracy of object detection to a certain degree, and many have achieved real-time performance for many benchmark datasets. However, images typically contain occlusions and small objects to which most current object detection methods are not sensitive. Insensitivity to these objects will inevitably restrict the accuracy of object detection. Therefore, the development of detection methods that are sensitive to occlusions and small objects in images is a key problem that must be addressed to provide more robust object detection.

In general, the problem associated with small object detection is actually a problem involving the detection of objects with vastly different size scales, which is a very common problem in object detection. Hence detection of small objects become more challenging. As such, current object detection methods accommodate the detection of small objects by generating feature representations of different scales. A number of empirical studies [13, 14, 17] have suggested that feature representations generated by multi-scale feature maps are very helpful for detecting small objects, especially large-scale feature maps. This indicates that multi-scale feature extraction methods can be expected to enhance the detection of small objects. We are thus motivated to attempt to design a multi-scale feature extraction method and integrate it into our model. The problem of ensuring sensitivity to occlusions is generally addressed by capturing a large number of variations in visual features within a large dataset. However, capturing all possible occlusions within a dataset is not possible, and occlusions with low probabilities will be absent from even very large datasets. So, how can

✉ Zhixin Li
lizx@gxnu.edu.cn

[1] Guangxi Key Lab of Multi-source Information Mining and Security, Guangxi Normal University, Guilin 541004, China

we get these rare occlusions? Moreover, an effort to address this issue by collecting larger data sets is highly inefficient. Therefore we consider trying to use an adversarial network to generate the occlusions what we need.

This paper addresses these problems by proposing an improved approach denoted as Robust Faster R-CNN. The novel design employs a cascaded network structure based on the Faster R-CNN architecture to extract features from objects with different scales in multi-label data. In addition, we train the adversarial network to generate training samples with occlusions significantly affecting the classification ability of the model, which improves its robustness to occlusions. Furthermore, we design a multi-scale RoIAlign operation by adding multiple pool sizes for adapting the detection ability of the network to objects with different sizes. Experimental results for the PASCAL VOC 2012 and 2007 datasets, which are widely used benchmarks for evaluating object detection performance, demonstrate that our approach performs more effectively and more accurately than several state-of-the-art approaches.

2 Related work

In the past few years, many works have been carried out on various object detection models. These models are usually based on two types of frameworks: One kind of object detection method rely on region proposal. These region-based methods divide the object detection task into two stages. In the first stage, a dedicated region proposal generation network(RPN) is grafted onto a deep convolutional neural networks (CNNs) to extract features from the proposed regions, and thereby generate high quality candidate boxes. Then a region-wise sub-network is designed to classify and refine these candidate boxes in the second stage. And another region-free methods divide the object detection task into one stage.

With the rise of CNN, the two-stage methods has quickly become the mainstream of object detection in recent years. Such as R-CNN [5], Fast R-CNN [4], Faster R-CNN [17], SPPnet [6], R-FCN [2]. R-CNN [5] method extracted region proposals using the Selective Search method [23], and linear support vector machine (SVM) was adopted as a classifier for region proposals. However, for R-CNN, the process of generating region proposals was computationally slow. Accordingly, Fast R-CNN [4] was developed to increase the computational speed of the region proposal generation process by developing a novel RoIPooling (i.e., Spatial Pyramid-Pooling) that allowed the classification layers to reuse features computed over CNN feature maps. Then, Faster R-CNN [17] replaced the Selective Search method with a network of region proposal generation to further increase the computational speed of region proposal generation. Moreover the convolutional layers were shared with other components, which realized end-to-end training of the entire network. Faster R-CNN was elected as the state-of-the-art method in the ILSVRC and COCO 2015 competitions, and a performance of 69.9 was obtained for the PASCAL VOC 2007 dataset [3]. In addition, single-stage object detection methods, such as SSD [14], YOLO [16], and RON [11], have been developed in recent years. These methods directly estimate object candidates without a reliance on region proposal, and are therefore computationally faster than two-stage methods. While these methods have a great performance for salient and universal object, they are hard to recognize occlusions and small object.

The current success of object detection is closely related to the application of large-scale dataset. But for occlusion problem, some of the rare occlusions are not easy to find in large-scale dataset. However, it is inefficient to expand the dataset by adding rare occlusion samples. Therefore, instead of attempting to collect the dataset to find rare occlusions, we attempt to generate occlusions which will be rare occlusion samples. Hence, we do a lot of work about adversarial networks. As an alternative to relying on large-scale datasets to capture all possible variations of visual features, A-Fast-R-CNN [24] proposed the training of an Adversarial Spatial Dropout Network (ASDN) to generate low probablity adversarial examples in convolutional features. This approach has recently demonstrated good performance [22]. This inspire us and motivate us to find wonderful idea to enhance the ability of our model to solve the occlusion problem. Other methods have proposed the use of a cascaded network to recognize occluded or invisible key points [1]. In addition, a 1×1 convolutional layer has been employed to reduce the number of network parameters and thereby accelerate calculations [21]. Although these past developments have resulted in considerable improvements in object detection for images with small objects and occlusions, none of these methods can effectively solve both problems simultaneously with reasonable accuracy and computational speed.

In contrast, the proposed methods in this paper combine a highly effective network structure, multi-layer fusion, multi-scale pooling, and a more effective training strategy to take full advantage of CNNs for object detection, and extracts features with different size scales without substantially reducing the computational speed. In conjunction with the adversarial network, the proposed method can adapt to widely varying object characteristics in multi-label images.

3 Improved model

3.1 Multi-cascaded network

The different depth features of a CNN correspond to different levels of semantic features. In general, the features extracted

by a deep network contain a greater proportion of high-level semantic information, while features extracted by a shallow network contain more detailed features. Therefore, the feature map becomes increasingly abstract as the depth of the network increases, and the reduced proportion of detailed information results in a decreased recognition effect for small objects. The solution to this problem employed by nearly all current methods that have achieved good classification and object detection results is to adopt image pyramids, i.e., multi-scale training. However, this method is computationally intensive. This has led to efforts seeking to enhance the recognition of multi-scale objects by modifying the network structure.

The above-described effect of increasing network depth on network performance has been clearly demonstrated by the VGG16 model [20], which is illustrated in Fig. 1. As can be clearly seen in the figure, the convolution layers of the VGG16 model adopts multiple small 3×3 convolutional kernels in succession, which increases the depth of the network while reducing the number of parameters, and thereby reducing the computational complexity of the model. In addition, the use of a smaller core facilitates the use a greater number of filters compared with algorithms adopting large convolution kernels, such as AlexNet [12]. This in turn facilities the use of a greater number of activation functions, which will enhance the learning of more complex patterns and concepts. However, small convolutional kernels can provide less information regarding the scale, shape, and position of objects, particularly for small objects. Furthermore, extant filled edge features are counted several times, which increases the number of errors. In contrast, larger convolution kernels can capture more spatial context, which facilitates the recognition of objects with more spatial context, which facilitates the recognition of objects with different scales. However, it is noted that the effect of the number of convolutional kernels is equivalent to the effect of the number of parameters. Therefore the number of kernels mainly determined by the quantity of parameters in the cascade networks. If the convolutional layers brings lots of parameters to the network, this will undoubtedly limit performance. We must control the number of parameters while improving performance. Accordingly, an optimal tradeoff is required between the quality of feature representation and the computational performance. This is addressed in the present work by designing the multi-cascaded network structure of the improved Faster R-CNN model illustrated in Fig. 1. The structure adds two shallow networks to the original VGG16 model, where one layer contains five 5×5 convolution kernels and the other layer contains three 7×7 convolution kernels. In addition, the two shallow networks added to the original VGG16 model make the final output feature map size equivalent to that of the VGG16 model, but with more detailed object information owing to its higher resolution. Because of high resolution feature map has more information of objects but contains more information of objects. Each cascaded network has the same number of pooling layers, as marked in the figure, which ensures that the feature maps used for fusion are consistent in size. The concat layer splices the feature map and maintains constant fusion feature map sizes. Actually, this represents matrix splicing, as is demonstrated clearly in Fig. 1. Batch normalization (BN) and scale operations are added after each convolution layer, which can increase the training rate and the classification effect [8].

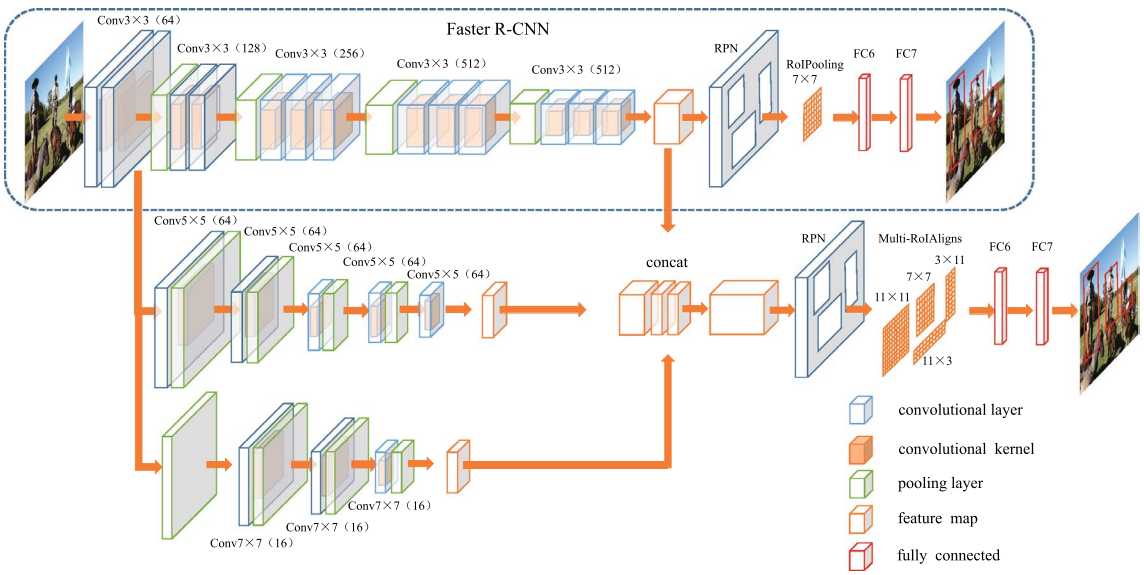

Fig. 1 The improved Faster R-CNN model with multi-scale RoIAligns and cascaded network structure

3.2 Parameter transferring

As shown in Fig. 2, the parameters pre-trained with the Faster R-CNN model are directly transferred to the improved Faster R-CNN model to reduce the training time [15]. Only the parameters pre-trained on Faster R-CNN for the last fc6 layer are not transferred because the use of multi-scale ROIAlign in the improved Faster R-CNN model changes the dimensions of the fc6 layer. Then, additional training is conducted to fine-tune the parameters for the improved Faster R-CNN model. Moreover, the transferred fc7 can be seen as a means of guaranteeing the representation capabilities of the transferred model parameters.

3.3 Multi-scale ROIAligns

The RoIPool operation [17] is a standard operation for extracting a small feature map (e.g., 7×7) from an RoI. First, RoIPool quantizes a floating number RoI to the discrete granularity of the feature map, this quantized RoI is then subdivided into spatial bins which are themselves quantized, Finally, the feature values representative of each bin are aggregated (usually by a max pooling operation). For example, quantization can be performed on a continuous coordinate x by computing *round*(x/16), where 16 is a feature map stride and the *round*(·) function represents rounding. However, these quantizations introduce misalignments between the RoI and the extracted features. While this may not impact classification, which is robust for large objects, it has a largely negative effect on predicting pixel-accurate object boxes (i.e., for small objects). Furthermore, the RoIPool operation breaks pixel-to-pixel translation-equivariance. Therefore, the present work adopts the RoIAlign operation proposed in Mask R-CNN [7]. This eliminates the harsh quantization of RoIPool, and properly aligns the extracted features with the input. As shown in Fig. 3, RoIAlign avoids any quantization of the RoI boundaries or bins (e.g., it applies x/16 rather than *round*(x/16)). Then, bilinear interpolation is employed to compute the exact values of the input features at four regularly sampled locations in each RoI bin, and the result is aggregated using a max pooling operation.

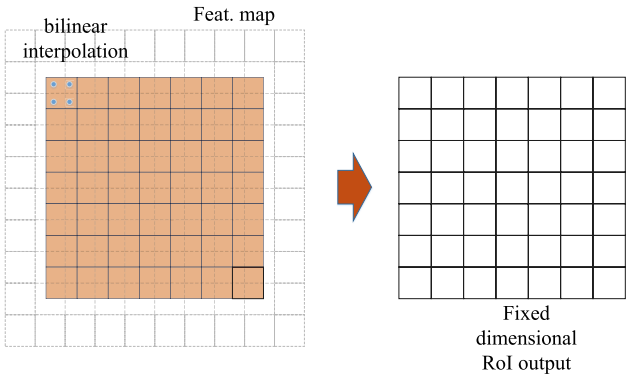

Fig. 3 Process of the RoIAlign operation, where the dashed grid represents a feature map, the solid lines an RoI (with 7×7 bins), and the dots the 4 sampling points in each bin. Here, RoIAlign computes the value of each sampling point by bilinear interpolation based on the nearby grid points on the feature map. No quantization performed on any coordinates involved in the RoI, its bins, or the sampling points

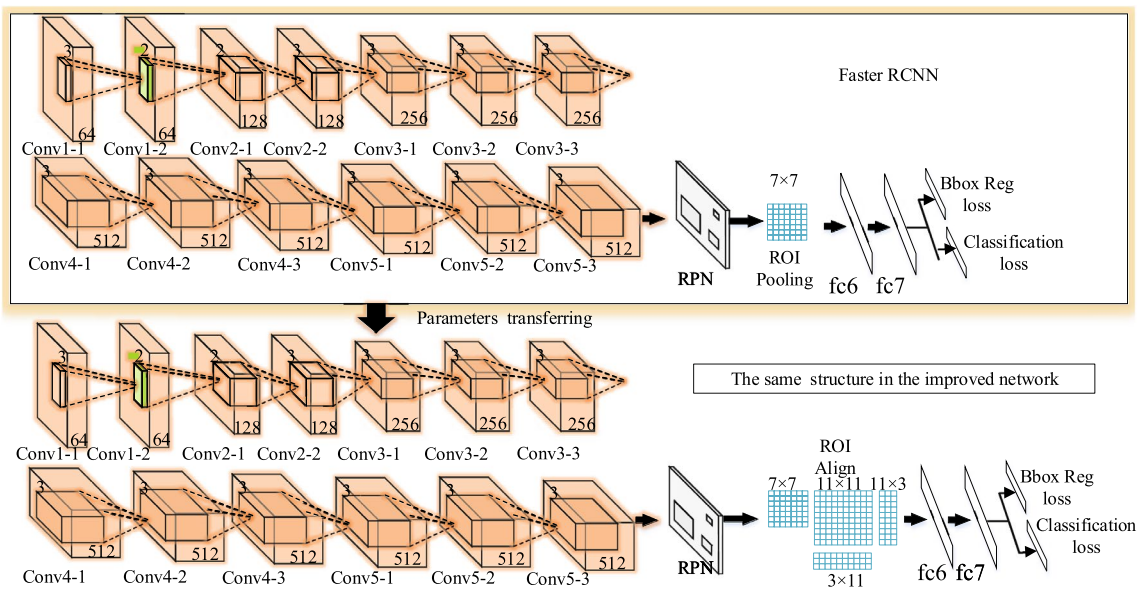

Fig. 2 Faster R-CNN parameter transference to the improved Faster R-CNN model

The Faster R-CNN framework tends to lose a considerable amount of object information during feature map generation, which seriously detracts from its small object detection performance [15]. For example, an object originally composed of 32×32 pixels has only 2×2 pixels remaining in the last layer of the feature map. This problem is generally addressed by enlarging the feature map and utilizing a smaller anchor scale in the RPN. The Faster R-CNN framework applies the RoIPool operation to the feature map with a pool size of 7×7 for each RoI proposed by the RPN. However, capturing object features at different size scales is quite difficult when employing a single pool size. The present work addresses this issue by applying two pooled sizes of 11×3 and 3×11, as has been previously proposed for enhancing small object detection in the R2CNN model [9]. The 3×11 pool size is designed to capture more horizontal features, and therefore aids in the detection of objects with widths that are much greater than their heights. In contrast, the 11×3 pool size is designed to capture more vertical features, and is therefore helpful for detecting objects with heights that are much greater than their widths. Furthermore, a pool size of 11×11 is also added to enhance the robustness of the proposed model for detecting objects at small size scales. In addition, the adoption of a smaller anchor scale has been demonstrated to enhance small object detection [9]. Therefore, we added smaller anchor scales to the original scales of (8, 16, 32) so that the proposed model utilized anchor scales of (4, 8, 16, 32), which would generate 12 anchors in the RPN. The proposed multi-scale RoIAlign operation can therefore pool features extracted at variable size scales, and thereby improves the accuracy of object detection.

3.4 Feature descending fusion

Since we use multi-scale poolingIn addition, which the multi-scale RoIAlign operation leads to the larger dimensions of subsequent provides a fully-connected layer with larger dimensions, and increases the computational time consumed of associated with object detection. The improved Faster R-CNN model also uses the convolution layer as well as the pooling layer to reduce the parameter redundancy of the fully connected layer, as has been previously proposed [21]. It is well known that the use of different dimensions in the multi-scale RoIAlign operation makes direct feature splicing impossible. However, this can be addressed through the use of a flatten layer to transform the pooled feature map (i.e., a multidimensional matrix) into a number of one-dimensional vectors, such as was applied in the R2CNN model. However, the present work seeks to avoid parameter redundancy prior to using the flatten layer by reducing the number of model parameters via the application of a convolutional layer with a kernel size of 1×1 and a step size of 1. We accordingly reduce the dimensionality of each of the four pooled feature maps, while the dimension of the 7×7 feature map is reduced to 512, that of the 11×11 feature map is reduced to 128, and the dimensions of the of 3×11 and 11×3 feature maps are reduced to 256. Then, we use flatten layers to transform the pooled feature map into four one-dimensional vectors, and the concat layer is employed to pass the vectors to the fully connected layer.

This process is illustrated in Fig. 4, where we have added a 1×1 convolution layer after each multi-scale pooling layer. As is well known, the convolution process using a 1×1 kernel typically acts to decrease dimensionality, which here refers to the number of image channels (thickness), while the width and height of the image is not changed.

4 Adversarial network

The functionality of an adversarial network $A(X)$, where X is a set of features, is first analyzed by comparing the loss function obtained for an object detector network $F(X)$ to that obtained for $A(X)$, while adopting the terms $F_c(X)$ and $F_l(X)$ to represent the class and predicted bounding box location outputs, respectively, and C and L to represent the respective groundtruth class and bounding box locations for X. Accordingly, the loss function of $F(X)$ can be given as follows.

$$E_F = E_{soft\max}(F_c(X), C) + E_{bbox}(F_l(X), L) \quad (1)$$

Here, the first term is the SoftMax loss and the second term is the loss based on $F_l(X)$ and L. The purpose of an adversarial network is to learn how to predict those X that $F_l(X)$ would fail to accurately classify. Accordingly, $A(X)$ generates new adversarial examples for a given X, which are then

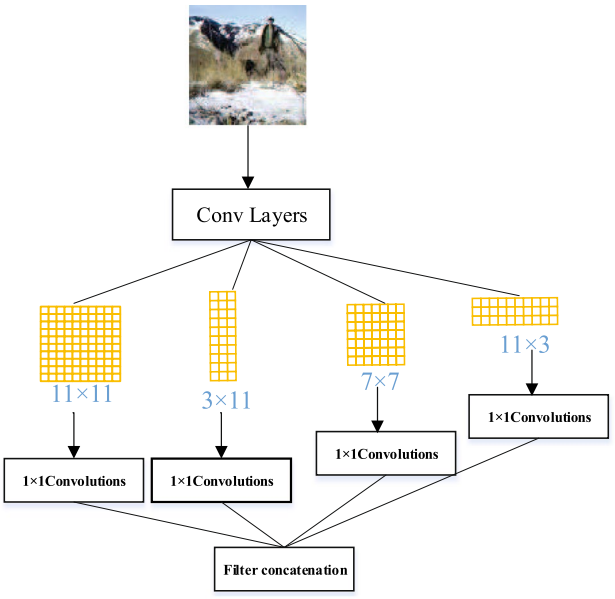

Fig. 4 The feature filtering structure using convolution layers with a kernel size of 1×1

added to the training samples. The adversarial network is trained via the following loss function.

$$E_A = -E_{soft\max}(F_c(A(X)), C) \tag{2}$$

Therefore, obtaining a low value of E_F for examples generated by $A(X)$ that are easily classified by $F(X)$ results in a high value of E_A. In contrast, obtaining a high value of E_F for examples generated by $A(X)$ that are difficult for $F(X)$ to classify results in a low value of E_A. As such, the two networks perform exactly opposite tasks.

4.1 Adversarial spatial dropout network training

We apply stage-wise training to the ASDN, as was conducted in a previous work [24]. Here, the ASDN is first pre-trained on a multi-label image dataset to obtain a preliminary perception of the dataset appropriate for use during the joint training of the ASDN and the improved Faster R-CNN. Subsequently, the ASDN is trained by fixing all of the network layers.

As shown in Fig. 5, the ASDN has the same structure as the improved Faster R-CNN framework in terms of the convolutional layers, RoIPooling layer, and the fully connected layers. The convolutional features for each feature map after the RoIPooling layer are applied as the inputs for the ASDN. Given a feature map of size $d \times d$, the ASDN will generate a mask representative of those parts of the feature map to be occluded by assigning zeros in an effort to increase the value of E_F obtained for the improved Faster R-CNN by introducing occluded features that are more difficult to classify. This is conducted by applying a $d/3 \times d/3$ sliding window that deletes the values in all the channels at its corresponding position, and thereby generates a new feature vector. All of the new feature vectors obtained in this manner are passed to the Softmax loss layer to calculate the loss function, and the feature vector obtaining the highest loss is selected. Then, the window creates a single $d \times d$ mask with 1 for the central window location and 0 for the other pixels. The sliding window process is represented by mapping the window back to the image, as shown in Fig. 6a. In this way, the ASDN generates spatial masks for n feature maps and obtains n training samples that have high losses. The binary cross entropy loss is used to train the ASDN, which is given as follows.

$$E = -\frac{1}{n}\sum_{p}^{n}\sum_{i,j}^{d}\left[\tilde{M}_{ij}^{p} A_{ij}(X^p) + \left(1-\tilde{M}_{ij}^{p}\right)\left(1-A_{ij}(X^p)\right)\right] \tag{3}$$

Here, $A_{ij}(X^p)$ represents the outputs of the ASDN at location (i, j) given an input feature map X^p, and if $M_{ij} = 1$, we drop out the values of all the channels in the corresponding spatial location of the feature map X. The output generated by the ASDN is not a binary mask but rather a continuous heatmap. The ASDN uses importance sampling to select the 1 / 3 of the pixels in a heatmap, which are assigned a value of 1, while the remaining 2 / 3 pixels are set to 0. As illustrated in Fig. 6b, the application of occlusions generating high loss in the ASDN learning process results in a recognition of those parts of objects that are most significant for classification.In this case, we use the masks to occlude these parts to make the classification harder.

4.2 Joint training

We jointly optimize the pre-trained ASDN and our improved Faster R-CNN model. In the joint model, the ASDN shares the convolutional layers and RoIPooling layer with the improved Faster R-CNN model, but uses its own separate fully connected layers. Naturally, the parameters of the two networks must be optimized independently in accordance with their diametrically opposed tasks. For training the improved Faster R-CNN model, we first use the pre-trained ASDN to generate masks for creating modified feature maps after the RoIPoolings layer during the forward propagation training stage, and then pass the modified features to the improved Faster R-CNN model for calculating losses and

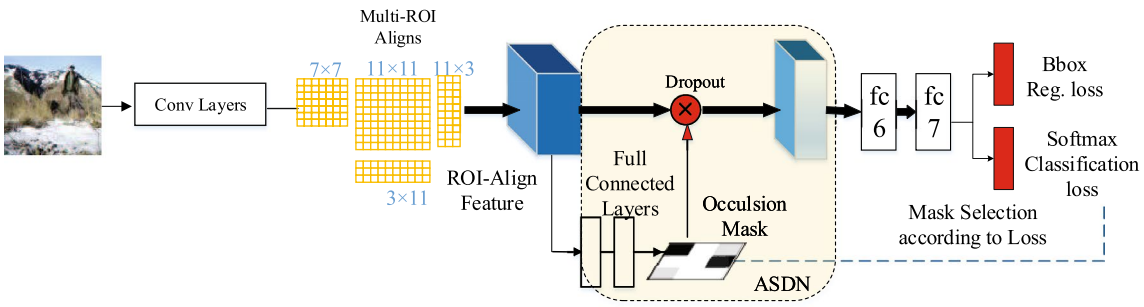

Fig. 5 Architecture of the Adversarial Spatial Dropout Network (ASDN) in combination with the improved Faster R-CNN framework. Occlusion masks are created to generate training examples that are difficult to classify

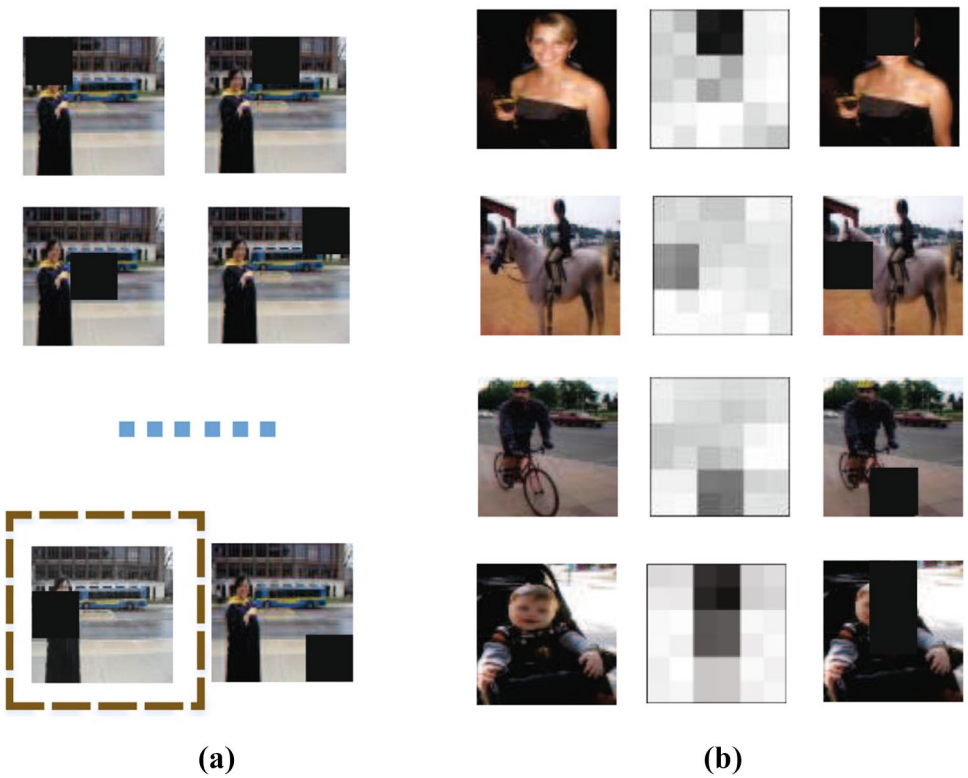

Fig. 6 **a** Examples of occlusions that are sifted to select the hard occlusions, and are used as the groundtruth for training the ASDN. **b** Examples of occlusion masks generated by the ASDN, where the black regions represent occlusions representative of the most significant pixels for classification

model training. Although the features are modified, their labels remain unchanged. This ensures that more diverse examples are introduced when training the improved Faster R-CNN model, and results in greater robustness for classifying objects with occlusions. For training the ASDN, the sampling strategy applied to convert the heatmap into a binary mask makes the classification loss calculation non-differentiable, so that the gradients from the classification loss are not available for back-propagation during training. Same as A Fast R-CNN [24], only those hard example masks are used as ground-truth to train the adversarial network by using the same loss as described in Eq. (3) to compute which binary masks lead to significant drops in Robust Faster R-CNN classification scores.

5 Experiment

5.1 Datasets and evaluation metrics

The PASCAL VOC 2007 and 2012 datasets employed in the experiments contain a total of 9963 and 22,531 images, respectively, and are divided into train, val, and test subsets. Our experiments employed 5011 trainval and 4952 test images for VOC 2007 and 11,540 trainval and 10,991 test images for VOC 2012. The average precision (AP) and the mean of the AP (mAP) were employed as the evaluation metrics in compliance with the PASCAL challenge protocols. Test speed and convergence speed are also important metrics for evaluating model performance. The experimental results obtained for the proposed improved Faster R-CNN and Robust Faster R-CNN frameworks were compared with results obtained using several state-of-the-art approaches, including Faster R-CNN, A-Fast-R-CNN, SSD, and RON. All of the experimental results were obtained by running the models on a PC equipped with an i7 processor with a 4.20 GHz clock speed, a GTX 1080Ti single core GPU, and 16 GB memory.

5.2 Convergence and joint model training

We initialized the parameters of the improved Faster R-CNN with the Faster R-CNN parameters trained on the VOC 2007 trainval subset. To accommodate the changed dimensions of the fully connected fc6 layer in the improved model, this layer was initialized from zero-mean Gaussian distributions with standard deviations 0.01, and the learning rate was set to 0.01 and scaled by a factor of 0.1 every 20 epoches based on momentum and weight decay values of 0.9 and 0.0005, respectively, for a total of 60 epoches. Training for the Faster R-CNN model and the improved Faster R-CNN model included a number of iterations set to 60 epochs, where each epoch consisted of 2000 iterations. The mAP values of the training models were calculated at different iterations during the training processes prior to generating the final models, and the results are presented in Fig. 7. The

figure indicates that the mAP scores for the training models began to converge after a little less than 40 epochs, or 70K iterations. Over these number of iterations, the improved Faster R-CNN training model yielded an mAP score of 77.5% and that of the Faster R-CNN training model was 73.2%. These results demonstrate that the improved Faster R-CNN model has a faster convergence rate than the Faster R-CNN model. The ASDN was pre-trained for 12K iterations. Then, the joint model was trained for 120K iterations. We again adopted a varying learning rate, which was initially 0.001 and decreased to 0.0001 after 60K iterations based on the momentum and weight decay values adopted in the previous part.

5.3 Ablation experiments

The ablation experiments were designed to evaluate the influence of different anchor scales and different RoIPool sizes on the object detection performance of the models trained with the VOC 2007 dataset, including the Faster R-CNN, cascaded network, which is an equivalent network structure to that of the improved Faster R-CNN, but which adopts the standard RoIPool operation, and Robust Faster R-CNN. Although RoIPooling can also capture the different scale features of the object, but compared with RoIAligns having lower accuracy. Owing to RoIAligns removes the strict quantization of RoIPooling, correctly aligning the extracted features with RoI. These quantization of RoIPooling introduce misalignment problem between the RoI and the extracted features. Furthermore RoIPooling breaks pixel-to-pixel translation-equivariance. Meanwhile quantization lead to miss some information of features. While this may not effect on accuracy of detecting large objects, for small objects, the problem of quantization will reduce the accuracy of recognition. The results are presented in Table 1. The results clearly indicate that the cascaded network with four pool sizes ($3 \times 11, 11 \times 3, 7 \times 7, 11 \times 11$) performed better than Faster R-CNN with a single pool size (7×7), and the cascaded network with a single pool size (7×7) and three pool sizes ($3 \times 11, 11 \times 3, 7 \times 7$). Firstly, these results demonstrate the benefits of the developed multi-scale RoIAlign operation over the standard RoIPool operation owing to the enhanced capability of the multi-scale operation to extract features at variable scales. Secondly, these results demonstrate the cascaded network we designed has a very positive effect on the accuracy from the experimental results and the cascaded network can capture more information so that it can recognize more objects of different sizes. With the increase of depth of the network, the feature map becomes more and more abstract. Some information will be ignored through convolution and pooling, especially small objects. Low-resolution feature map is unfavorable to the recognition of small objects. Hence, we designed a cascaded network structure to extract features from objects with different scales. Finally, these results demonstrate the benefits of including horizontally and vertically biased pool sizes, and also demonstrate that the addition of the 11×11 pool size enhances the object detection performance of the cascaded network. This latter benefit is mainly because the additional 11×11 pool size can enhance the detection of smaller objects in the VOC dataset. FT means fine-tuning and it has also contributed greatly to the improvement of model performance. Compared with the results obtained for the improved Faster R-CNN model, the Robust Faster R-CNN

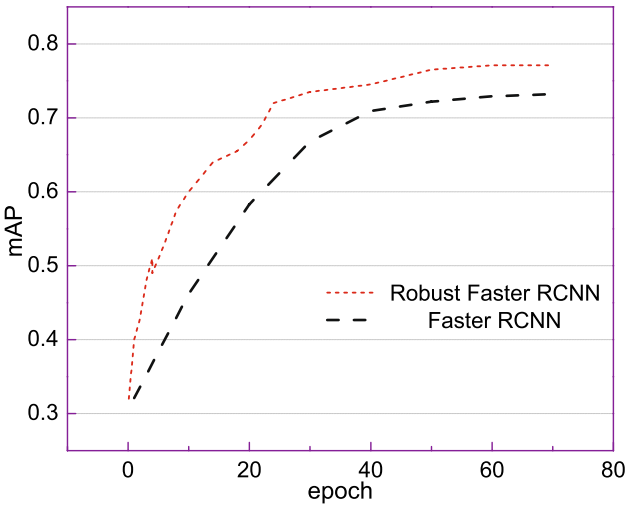

Fig. 7 The mAP scores obtained during model training based on the PASCAL VOC 2007 dataset

Table 1 Ablation experiment results on the VOC 2007 dataset

Approach	Anchor	Pooled sizes	mAP
Faster RCNN	(8,16,32)	7×7	73.2
Faster RCNN	(4,8,16,32)	7×7	73.3
Cascade network	(4,8,16,32)	7×7	73.9
Cascade network+RoIAligns	(4,8,16,32)	$3 \times 11, 11 \times 3, 7 \times 7$	74.5
Cascade network+RoIAligns	(4,8,16,32)	$3 \times 11, 11 \times 3, 7 \times 7, 11 \times 11$	74.8
Cascade network+RoIAligns+FT	(4,8,16,32)	$3 \times 11, 11 \times 3, 7 \times 7, 11 \times 11$	75.2
Cascade network+RoIAligns+FT+ASDN	(4,8,16,32)	$3 \times 11, 11 \times 3, 7 \times 7, 11 \times 11$	77.5

model provided an mAP that was 2.3% greater, reflecting the effectiveness of the ASDN.

5.4 Results

The AP and mAP values obtained for various images in the VOC 2007 dataset and the VOC 2012 dataset by means of the proposed object detection frameworks and the various state-of-the-art approaches are listed in Tables 2 and 3, respectively. The results indicate that the detection performances of the proposed Robust Faster R-CNN models are significantly better than that of Faster R-CNN, and their detection accuracy for small objects in particular, such as bird and plant, is significantly improved. These results confirm the feasibility of the proposed multi-scale RoIAlign operation. Although the mAP value obtained by the state-of-the-art RON approach for the VOC 2007 dataset is slightly greater than that obtained by the proposed robust model, the robust model performs 4.3% better than Faster R-CNN, which demonstrates the effectiveness of our approach. The results in the tables also clearly demonstrate that the inclusion of the ASDN provides significantly greater object detection performance than the improved Faster R-CNN model, which confirms the effectiveness of the ASDN.

Table 2 Object detection results on the PASCAL VOC 2007 dataset

	Faster R-CNN	A-Fast-RCNN	SSD	RON	Robust Faster R-CNN
aero	76.5	75.7	79.8	**86.0**	79.8
bike	79.0	83.6	79.5	82.5	**84.1**
bird	70.9	68.4	74.5	**76.9**	76.8
boat	65.5	58.0	63.4	69.1	**68.0**
blt	52.1	44.7	51.9	**59.2**	57.4
bus	83.1	81.9	84.9	86.2	**87.8**
car	84.7	80.4	85.6	85.5	**88.1**
cat	86.4	86.3	87.2	87.2	**88.5**
chair	52.0	53.7	56.6	**59.9**	59.0
cow	81.9	76.1	80.1	81.4	**84.4**
tabel	65.7	72.5	70.0	**73.3**	72.3
dog	84.8	82.6	85.4	85.9	**86.9**
hrs	84.6	83.9	84.9	86.8	**90.0**
mbk	77.5	77.1	80.9	82.2	**83.2**
per	76.7	73.1	78.2	79.6	**82.6**
plant	38.8	38.1	49.0	**52.4**	43.6
shp	73.6	70.0	**78.4**	78.2	77.2
sofa	73.9	69.7	72.4	76.0	**77.0**
train	83.0	78.8	84.6	**86.2**	85.6
tv	72.6	73.1	75.5	**78.0**	77.6
mAP	73.2	71.4	75.1	**77.6**	77.5

Bold values indicate the best performance under each test item

Table 3 Object detection results on the PASCAL VOC 2012 dataset

	Faster R-CNN	A-Fast-RCNN	SSD	RON	Robust Faster R-CNN
aero	84.9	82.2	84.9	86.5	**87.0**
bike	79.8	75.6	82.6	82.9	**83.5**
bird	74.3	69.2	74.4	76.6	**78.9**
boat	53.9	52.0	55.8	**60.9**	60.1
blt	49.8	47.2	50.0	55.8	**57.6**
bus	77.5	76.3	80.3	81.7	**83.2**
car	75.9	71.2	78.9	80.2	**80.5**
cat	88.5	88.5	88.8	**91.1**	90.2
chair	45.6	46.8	**53.7**	57.3	51.6
cow	77.1	74.0	76.8	81.1	**82.4**
tabel	55.3	58.1	59.4	60.4	**61.6**
dog	86.9	85.6	87.6	87.2	**89.9**
hrs	81.7	80.3	83.7	84.8	**89.8**
mbk	80.9	80.5	82.6	84.9	**82.8**
per	79.6	74.7	81.4	81.7	**86.6**
plant	40.1	41.5	47.2	51.9	**47.4**
shp	72.6	70.4	75.5	**79.1**	74.2
sofa	60.9	62.2	65.6	68.6	**70.0**
train	81.2	77.4	84.3	84.1	**86.6**
tv	61.5	67.0	68.1	**70.3**	69.9
mAP	70.4	69.0	73.1	75.4	**75.7**

Bold values indicate the best performance under each test item

Some examples of object detection results obtained by the Robust Faster R-CNN model for the VOC 2007 and 2012 datasets are shown in Fig. 8. These examples demonstrate qualitatively that Robust Faster R-CNN can recognize objects with different sizes and width-to-height aspect ratios, and can predict their locations well, particularly for objects like planes, birds, and people. The results in Fig. 8 also demonstrate the robustness of the proposed approach to occlusions, such as in the car, plant, and people images that include occlusions.

We also qualitatively compare some examples of object detection results obtained by the Robust Faster R-CNN and Faster R-CNN models for the VOC 2007 and 2012 datasets in Fig. 9. In the first case, a bus suffering from occlusion at the top left of the image is ignored by Faster R-CNN, while the proposed method correctly labeled this vague object as a bus. In the second case, a woman on the rightmost side of figure is shown with only half a body and is carrying a small child in her arms. Here, Faster R-CNN detects no object whatsoever at this location in the image, while a person is detected using our proposed method. These examples represent a striking contrast between Faster R-CNN and the proposed method. Finally, the third case presents a chair suffering from occlusion, which is ignored by Faster R-CNN,

Fig. 8 Selected examples of object detection results on the PASCAL VOC 2007 and VOC 2012

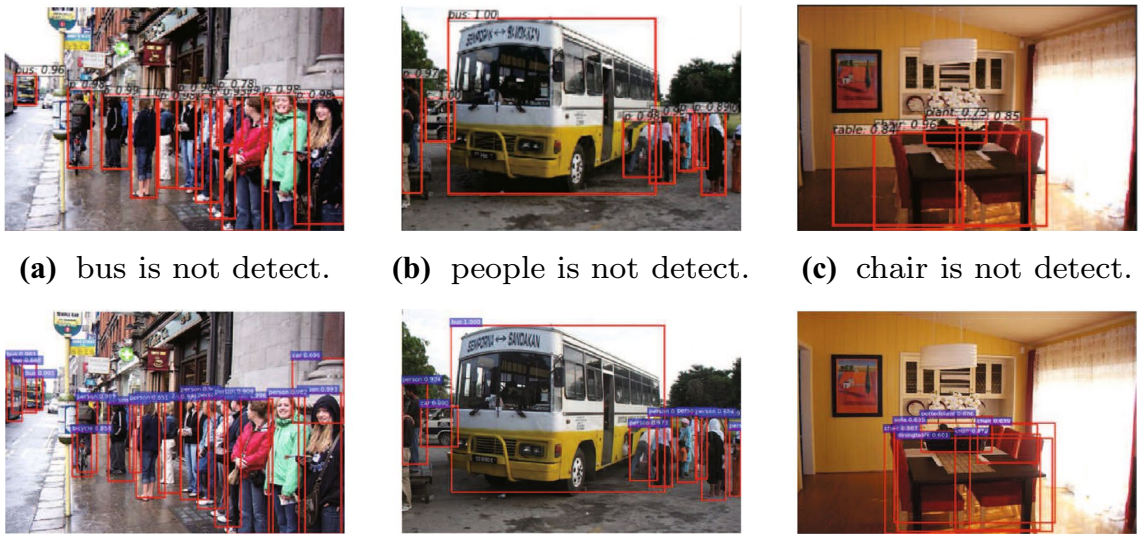

Fig. 9 Qualitative results of faster R-CNN vs. Robust Faster R-CNN on VOC. In every pair of detection results (top vs. bottom), the top is based on faster R-CNN, and the bottom is detection result of Robust Faster R-CNN

while the proposed method correctly labels this object as a chair. These illustrations demonstrate the obvious advantages of the proposed method over Faster R-CNN for identifying small objects and objects with occlusions.

Finally, Table 4 lists the detection and computational performance results obtained by Faster R-CNN and Robust Faster R-CNN, which are two-stage methods, and SSD and RON, which are single-stage methods, for the VOC 2012 dataset images. Here, we collected the object detection time for each image, and averaged all of the detection times (ms/image). The results indicate that the two-stage methods generally provide a greater accuracy but lower computational speed than the one-stage methods. In addition, we

Table 4 Detection and computational performance results of the proposed Robust Faster R-CNN and one-stage SSD and RON methods on the PASCAL VOC 2012 dataset

Method	One-stage		Two-stage	
	SSD	RON	Faster R-CNN	Robust Faster R-CNN
Test time (ms/image)	**46**	67	200	234
mAP	73.1	75.4	70.4	**75.7**

The two-stage methods generally provide greater accuracy but lower computational speed than the one-stage methods

Bold values indicate the best performance under each test item

note that, while the computational speed of Robust Faster R-CNN was less than that of Faster R-CNN, this is expected because the use of the multi-scale RoIAlign operation in Robust Faster R-CNN consumes more computational time than the RoIPool operation in Faster R-CNN. Moreover, the difference between the two is quite small, and Robust Faster R-CNN still meets the requirements of real-time object detection. Consequently, the proposed approach provides dramatically increased detection performance relative to Faster R-CNN with only a slight reduction in computational speed.

6 Conclusion

This paper presented an effective framework denoted as Robust Faster R-CNN for detecting objects with different size scales and occlusions. The use of a cascaded network as well as the multi-scale RoIAlign operation to learn semantic multi-scale feature representations made the proposed model invariant to objects with different sizes and width-to-height aspect ratios, such as people, cars, and planes. An ASDN was combined with the proposed network to generate training samples with occlusions significantly affecting the classification ability of the model, which improved its robustness to occlusions. Experimental results obtained by the proposed approach and various state-of-the-art approaches for images in the PASCAL VOC 2012 and 2007 datasets demonstrated that the Robust Faster R-CNN model generally obtained superior detection accuracy, and the speed of detection was not significantly reduced relative to that of the Faster R-CNN model.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Nos. 61966004, 61663004, 61762078, 61866004), the Guangxi Natural Science Foundation (Nos. 2016GXNSFAA380146, 2017GXNSFAA198365, 2018GXNSFDA281009), the Research Fund of Guangxi Key Lab of Multi-source Information Mining and Security (16-A-03-02, MIMS18-08), the Guangxi Special Project of Science and Technology Base and Talents (AD16380008), the Guangxi "Bagui Scholar" Teams for Innovation and Research Project, Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing.

References

1. Chen Y, Wang Z, Peng Y, Zhang Z, Yu G, Sun J (2018) Cascaded pyramid network for multi-person pose estimation. Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7103–7112
2. Dai J, Li Y, He K, Sun J (2016) R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems, pp 379–387
3. Everingham M, Williams C (2010) The pascal visual object classes challenge 2010 (voc2010). In: International conference on machine learning, pp 117–176
4. Girshick R (2015) Fast r-cnn. In: Advances in neural information processing systems, pp 91–99
5. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of IEEE international conference on computer vision and pattern recognition, pp 580–587
6. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904
7. He K, Gkioxari G, Dollar P, Girshick R (2017) Mask r-cnn. IEEE Trans Pattern Anal Mach Intell 99:1–1
8. Huang G, Liu Z, Laurens VDM, Weinberger KQ (2016) Densely connected convolutional networks. In: Proceedings of IEEE international conference on computer vision and pattern recognition, pp 2261–2269
9. Jiang Y, Zhu X, Wang X, Yang S, Li W, Wang H, Fu P, Luo Z (2017) R2cnn: Rotational region cnn for orientation robust scene text detection. In: Proceedings of IEEE international conference on computer vision and pattern recognition, pp 2261–2269
10. Kong T, Yao A, Chen Y, Sun F (2016) Hypernet: Towards accurate region proposal generation and joint object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 845–853
11. Kong T, Sun F, Yao A, Liu H, Lu M, Chen Y (2017) Ron: Reverse connection with objectness prior networks for object detection. In: Proceedings of IEEE international conference on computer vision and pattern recognition, vol 1
12. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
13. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2117–2125
14. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2015) Ssd: Single shot multibox detector. In: European conference on computer vision, pp 21–37
15. Oquab M, Bottou L, Laptev I, Sivic J (2014) Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of IEEE international conference on computer vision and pattern recognition, pp 1717–1724
16. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Computer vision and pattern recognition
17. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99
18. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vision 115(3):211–252
19. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, Lecun Y (2013) Overfeat: Integrated recognition, localization and detection using convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition
20. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv :14091556
21. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the thirty-first AAAI conference on artificial intelligence
22. Tao Z, Li Z, Zhang C, Lan L (2018) An improved convolutional neural network model with adversarial net for multi-label image classification. In: Pacific Rim international conference on artificial intelligence

23. Uijlings JR, Van De Sande KE, Gevers T, Smeulders AW (2013) Selective search for object recognition. Int J Comput Vision 104(2):154–171
24. Wang X, Shrivastava A, Gupta A (2017) A-fast-rcnn: Hard positive generation via adversary for object detection. In: Proceedings of IEEE international conference on computer vision and pattern recognition, pp 21–26
25. Wei S, Li Z, Zhang C (2018) Combined constraint-based with metric-based in semi-supervised clustering ensemble. Int J Mach Learn Cybernet 9(7):1085–1100
26. Wei Y, Xia W, Lin M, Huang J, Ni B, Dong J, Zhao Y, Yan S (2016) Hcp: A flexible cnn framework for multi-label image classification. IEEE Trans Pattern Anal Mach Intell 38(9):1901–1907
27. Zheng Y, Li Z, Zhang C (2018) A hybrid architecture based on cnn for cross-modal semantic instance annotation. Multimedia Tools and Applications 77(7):8695–8710