future internet

MDPI

Article

Object Detection Network Based on Feature Fusion and Attention Mechanism

Ying Zhang [1], Yimin Chen [1,2,*], Chen Huang [1] and Mingke Gao [3]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China; youmemeda@shu.edu.cn (Y.Z.); channinghuang@shu.edu.cn (C.H.)
[2] Shanghai Institute for Advanced Communication and Data Science, Shanghai 200444, China
[3] The 32nd Research Institute, China Electronics Technology Group Corporation, No. 63 Chengliugong Road, Jiading District, Shanghai 200444, China; gaomingke@shu.edu.cn
* Correspondence: ymchen@mail.shu.edu.cn

Received: 9 November 2018; Accepted: 25 December 2018; Published: 2 January 2019

Abstract: In recent years, almost all of the current top-performing object detection networks use CNN (convolutional neural networks) features. State-of-the-art object detection networks depend on CNN features. In this work, we add feature fusion in the object detection network to obtain a better CNN feature, which incorporates well deep, but semantic, and shallow, but high-resolution, CNN features, thus improving the performance of a small object. Also, the attention mechanism was applied to our object detection network, AF R-CNN (attention mechanism and convolution feature fusion based object detection), to enhance the impact of significant features and weaken background interference. Our AF R-CNN is a single end to end network. We choose the pre-trained network, VGG-16, to extract CNN features. Our detection network is trained on the dataset, PASCAL VOC 2007 and 2012. Empirical evaluation of the PASCAL VOC 2007 dataset demonstrates the effectiveness and improvement of our approach. Our AF R-CNN achieves an object detection accuracy of 75.9% on PASCAL VOC 2007, six points higher than Faster R-CNN.

Keywords: CNN; object detection network; attention mechanism; feature fusion

1. Introduction

Recently, object detection accuracy has been improved by using deep CNN (Convolutional Neural Network) nets [1]. There are two types of object detection methods: Region-based object detection and regression-based object detection. Representative algorithms of regression-based object detection are R-CNN (region with CNN features) [2], SPP-net [3], Fast R-CNN [4], and Faster R-CNN [5]. Faster R-CNN is an end-to-end object detection network because it combines region proposal and detection into a unified network. The object detection networks have three essential components called the feature extraction net, region proposal network, and classification and regression.

Object detection always uses deep CNN nets (VGG-16(very deep convolutional networks)) [1], ResNet [6]), to extract CNN features due to the discriminative representations of CNN features. Deep CNN features contain high-level semantic information, which can better detect objects. However, deep CNN features lose a lot of detailed texture information because of high abstraction. Object detection is a more challenging task when objects are small. Object detection is difficult to achieve the desired effect due to illumination variations, occlusions, and a complex background. As mentioned above, features are crucial for object detection. Object detection often uses the last CNN feature maps for region proposal net. In [7], the author combined high layer information with low layer information for semantic segmentation. HyperNet [8], which combines the different CNN features, achieved the

desired effect accuracy of object detection. The proper fusion of CNN features is more suitable for proposal generation and detection.

Recently, many studies have hoped to add attention mechanisms to deep networks. The attention model in deep learning actually simulates the attention model of the human brain. In 2015, Kelvin Xu et al. [9] introduced the attention in image caption. The visual attention mechanism can quickly locate the region of interest in the complex scene, finding the target of interest, and selectively ignoring the region of no interest. The visual attention mechanism is divided into two ways: Bottom-up model, which is also called hard attention, and top-down model, which is called soft attention. Anderson P et al. [10] applied hard attention to image caption. However, more research and applications are still more inclined to use soft attention because it can be directly derived for gradient back propagation. The paper [11] proposed the soft attention model and applied it to machine translation. The traditional attention mechanism is soft attention [11,12], which is obtained by deterministic score calculation to obtain the coded hidden state after the attainment. Soft attention is parameterized and it can be guided and embedded in the model for direct training. The gradient can be passed back through the attention mechanism module to other parts of the model. Object detection is difficult due to the complex background. Pixels are treated the same in the CNN feature maps and the region proposal net. The object is easily disturbed by the background, leading to inaccurate detection. Object detection networks need to enhance the impact of areas of interest and weaken interference from an unrelated background. So, we will apply the attention to the target detection depth network.

In this paper, we use the faster R-CNN object detection network as a framework. Deep network VGG-16 is still used in the feature extraction part. We propose a new object detection network that fixes the disadvantages of feature fusion and the attention mechanism in the faster R-CNN in case of background interference and small target problems. The improved object detection network (we called this AF R-CNN) is also an end-to-end network. Our main contributions are two-fold:

1. Feature fusion: We fuse the 3, 4, 5 layers of CNN feature maps by maximum pooling conv-3 and deconvoluting conv-5, which was extracted from VGG-16. The new CNN feature map combines fine, shallow layer information with coarse, deep layer information.
2. Attention mechanism: We propose a new network branch to generate a weight mask, which enhances the interest features and weakens the irrelevant feature in the CNN feature map. The attention network branch assigns attention weights to the CNN feature, making region proposal net more efficient and meaningful.

On the detection challenges of PASCAL VOC 2007, we achieved state-of-the-art mAP of 75.9% and 79.7%, outperforming the seminal Faster R-CNN by 6 and 6.5 points, correspondingly.

2. Related Works

2.1. Object Detection

We review object detection methods in this section, especially deep learning based methods. Object detection is designed to localize and identify every object using a bounding box. In the detection framework, object proposals [13] reduce the computational complexity compared with sliding window methods [14]. The traditional methods use manual features, such as edges and shapes. Deep learning based methods use CNN as features. With the great success of the deep learning [1,6], two major object detection methods based on CNN have been proposed: Proposal based methods [5,15] and proposal free methods [16,17]. The Faster R-CNN and its variants [18,19] have been the dominating methods for years. Mask R-CNN [20] uses RolAlign (a simple, quantization-free layer) instead of Rol (region of interest) pooling in the Faster R-CNN. Wang et al. [19] proposed a confrontation network (ASTN, ASDN) for occlusion and deformation. For the small target problem of target detection, Li et al. [21] proposed PGAN (perceptual generative adversarial network) in the object detection framework. Faster R-CNN and its variants are proposal based methods. Different from these methods, YOLO (You Only Look Once) [17] and its variants predict bounding boxes and class probabilities directly from full

images. YOLO 9000 [16], which achieves higher accuracy and speed, proposes a joint training strategy. YOLO and its variants are proposal free methods.

2.2. Visual Attention Mechanism

Recently, neuroscience and cognitive learning theory have developed rapidly. Evidence from the human perception process shows the importance of the attention mechanism. Koch [22] proposed a neurobiology framework that laid the foundation for the visual attention model. The visual attention mechanism is divided into two processing methods: Bottom-up model and top-down model. The bottom-up visual attention model is driven by data from the low-level features of the image. However, the top-down visual attention model is more complicated than the bottom-up visual attention model. The top-down visual attention model is driven by tasks, so it requires tasks to provide relevant prior knowledge.

Various visual attention models have been proposed. Itti et al. [23] proposed the Iitti attention model, which is based on feature integration. Bruce et al. [24] put forward the ATM attention model by training with image sub-blocks. Ali Borji et al. [25] combined bottom-up and top-down models. Recently, tentative efforts have been made towards applying attention to the deep neural network. The attention model has good results in the fields of image caption, machine translation, speech recognition, etc. Bahdanau et al. [11] proposed a single-layer attention model to solve source language alignment problems of different lengths in machine translation. Wang [26] applied a single attention model to news recommendation and screening filed. Muti-attention mechanisms (hierarchical attention, dual attention) can accomplish tasks more accurately. Rijke [27] proposed the hierarchical attention model to complete the abstract extraction of the article. Seo et al. [28] used the dual attention model for the recommendation system.

In object detection, there are several methods based on attention mechanisms that have been proposed. NonlocalNet [29] used self-attention to learn the correlation between features on the feature map and obtain new features. Hu [12] proposed an object relation module based on self-attention. The visual attention mechanism can be roughly divided into two types: Learning weight distribution and task focus. We added the attention module in the classification network to the learn weight. The attention module is a mask branch to enhance the interest feature and weak background interference.

3. Methods

Our object detection system, AF R-CNN, was composed of four modules: A deep convolutional network, VGG-16, that extracts features, feature fusion, and the attention modules, and a Faster R-CNN detector that uses the region proposal network (RPN). The whole object detection system is a unified network. Our AF R-CNN is illustrated in Figure 1. First, AF R-CNN produces feature maps through the VGG-16 net. We fused feature maps, which came from different layers, and then compressed them into a uniform feature map. Next, we obtained the weight from the attention module, and the RPN produced proposals. Finally, these proposals were classified and regressed.

3.1. Deep Convolutional Network:VGG-16 Net

Karen Simonyan proposed a series of VGG-16 networks in the paper [1]. The VGG-16 network contains 13 convolution layers and three full connect layers, as shown in Figure 2. In the convolution layers, there are 13 convolution layers, 13 relu (Rectified Liner Units) layers, and four pooling layers. Figure 2 shows the architecture of the VGG-16 network.

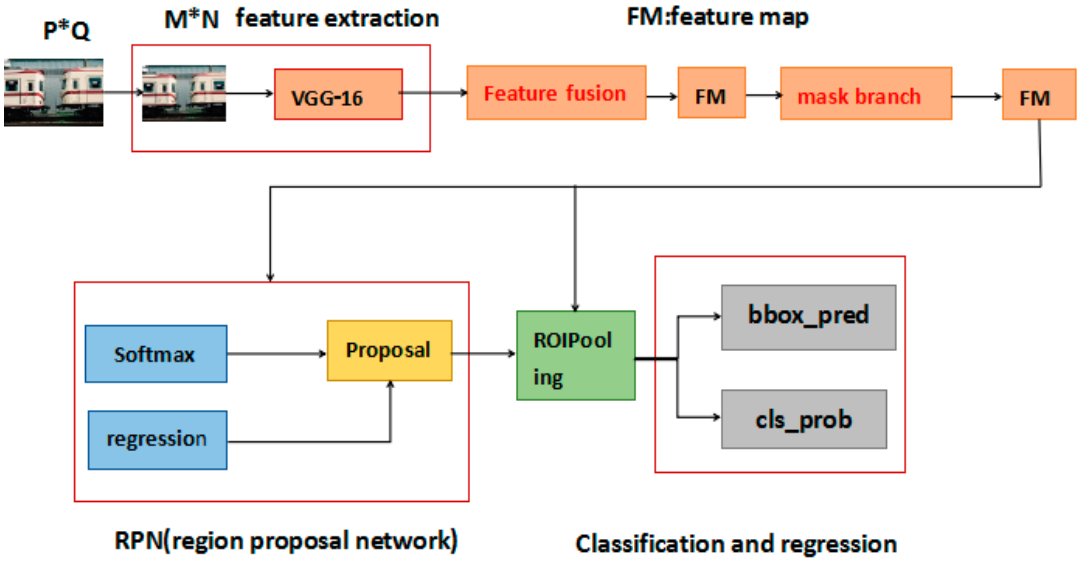
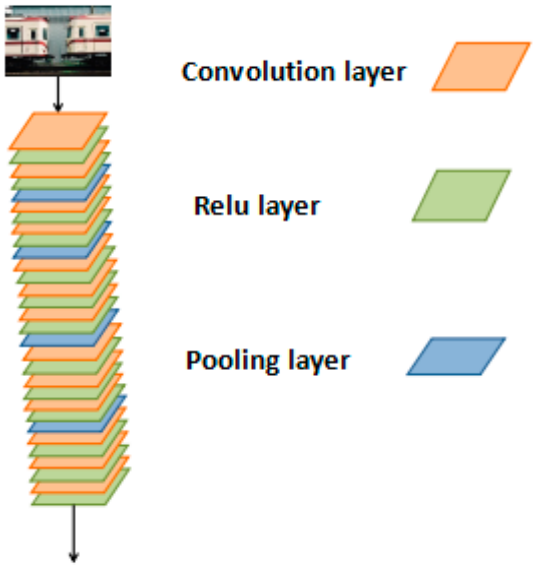


Figure 1. The architecture of AF R-CNN. The red part is the module we added above the original structure. The AF R-CNN is composed of four modules: A deep convolutional network, VGG-16, that extracts features, feature fusion, and the attention modules, and a Faster R-CNN detector that uses the region proposal network (RPN).



16 weight layers
Input(224*224 RGB image)
conv3-64
conv3-64
maxpool
conv3-128
conv3-128
conv3-128
maxpool
conv3-256
conv3-256
conv3-256
maxpool
conv3-512
conv3-512
conv3-512
maxpool
conv3-512
conv3-512
conv3-512
maxpool
FC-4096
FC-4096
FC-1000

(**a**) Structure diagram (**b**) Detailed data information

Figure 2. The architecture of VGG-16. (**a**) shows the structure diagram of the VGG-16 net; (**b**) shows the detailed data information of the VGG-16 net. In the convolution layers, there are 13 convolution layers, 13 relu (Rectified Liner Units) layers, and four pooling layers.

In all convolution layers, the convolution kernel is 3 × 3 and stride is 1. In the Faster R-CNN, the size of the image becomes (m + 2)(n + 2) because all convolution features are padded (pad = 1). The advantage of this is that the output image's size is the same as the input. The convolution formula is given by:

$$output_{size} = \frac{(input_{size}) - kernel_{size} + 2 \times pad}{stride} + 1 \quad (1)$$

where $input_{size}$ represents the size of the image or the size of the feature map in the middle. The $output_{size}$ represents the size of the feature map after the convolution kernel. The $kernel_{size}$ donates the size of the convolution kernel.

Each pooling layer will make the size of the output one-fourth of the input. After four pooling layers, the size of the convolution feature is (m/16)(n/16).

In Figure 3, the detailed padding process is shown. In the picture on the left, the blue area represents the size of the input image ((M × N)). The red area represents the size of the convolution kernel (3 × 3). After padding(pad=1), the image size becomes (M + 2) × (N + 2). Additionally, we can see in the picture on the right, the size of the convolution features are the same as the input picture after the convolution kernel. This is the meaning of the padding.

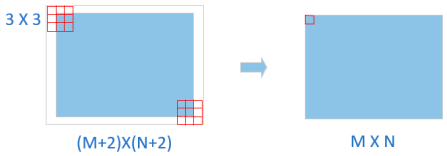


Figure 3. The process of padding. The size of the input image is (M × N). After padding(pad=1), the image size becomes (M + 2) × (N + 2).

3.2. Faster R-CNN Detector

Faster R-CNN combines the feature extraction net, region proposal net, and classification and regression into a network. Recently, many methods have been proposed based on Faster R-CNN. Figure 4 illustrates the Faster R-CNN architecture.

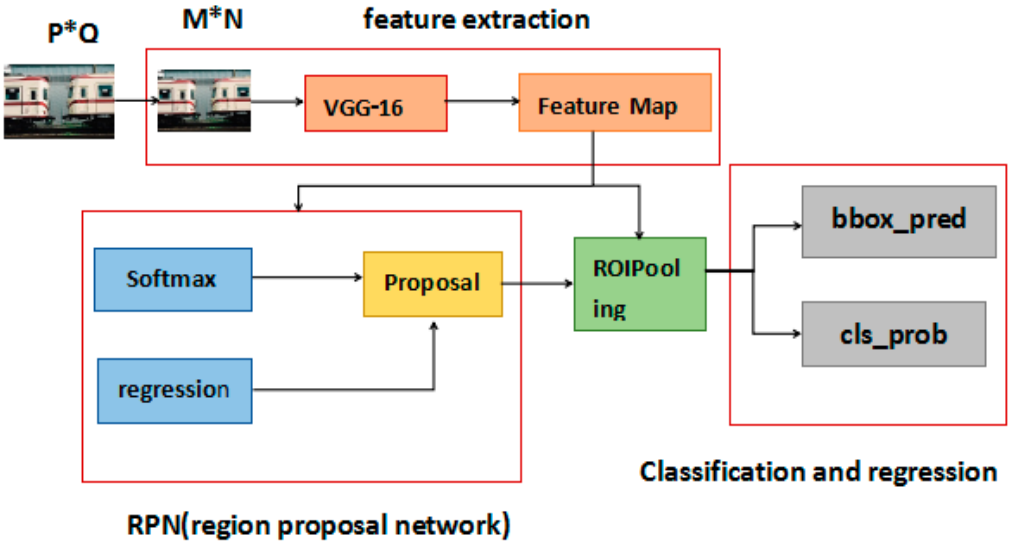


Figure 4. The architecture of Faster R-CNN. Faster R-CNN combines feature extraction net, region proposal net, and classification and regression into a network.

From Figure 4 we can see that Faster R-CNN architecture contains four main modules:

Feature extraction network. Faster R-CNN uses the VGG-16 net to generate convolution feature maps. We know that Faster R-CNN supports any size of input images because the images are normalized before entering the network. We assume that the normalized image size is m*n. After the convolution layers in the VGG-16 net, the size of the image changes to (m/16) ×(n/16). The convolution feature map is (m/16) × (n/16) × 512, which is shared for the region proposal network (RPN) and fully connected layers.

Region Proposal Network (RPN). R-CNN and Faster R-CNN use selective search (SS) as external modules independent of the object detection architecture. Object proposal methods include methods

based on grouping super-pixels [30] and methods based on sliding windows [31]. Faster R-CNN uses the region proposal network (RPN) to generate detection proposals. The anchor mechanism is the most important part of RPN. An anchor is centered at the sliding window in question. Multiple scale windows are required due to different object sizes and length to width ratios. The anchor gives a reference window size that different sizes of windows are obtained in three scales and three aspect ratios. In the region proposal network (RPN), the feature map convolutes with a 3×3 sliding window first. Then, the output divides into two ways by the 1×1 convolution kernel. The softmax function is used to classify anchors (foreground or background). The offset of anchors' bounding box regression can also be obtained to get the precise proposal. The two network branches are combined to obtain the proposal regions.

ROI (Region Of Interest) Pooling. The RoI pooling layer obtains a fixed size of the proposal feature map using the proposal regions and feature map in the RPN and VGG-16.

Classification and regression. The proposal feature map calculates the probability vector, which represents the category of proposals by the full connect layer and softmax function. Additionally, it obtains the position offset to get a more accurate proposal by using bounding box regression.

The steps of Faster R-CNN are as follows:

1. Input test image;
2. Extract about a 2000 proposal region from the image using the selective search algorithm;
3. Input the image into the VGG-16 net for feature extraction;
4. Mapp the proposal region to the feature map;
5. Generate a fixed size feature map in the RoI pooling layer; and
6. Perform classification and bounding box regression.

3.3. Feature Fusion

Faster R-CNN extracts features through a convolutional neural network to generate feature maps for RPN and object classification and border regression. The quality of the feature map greatly affects the performance of the entire object detection algorithm. The Faster R-CNN uses the VGG-16 network as the basic network and produces a final feature map after the fifth convolution layer. Although the feature map obtains high-level semantic information through deep convolution layers, the high-level feature map loses a lot of detailed texture information due to high abstraction, resulting in inaccurate positioning of objects and difficulties in the detection of small target objects.

To solve the above problems, the AF R-CNN model we proposed in this paper fuses features from different convolutional layers. Given a picture, AF R-CNN uses the pre-trained model, VGG-16, to compute feature maps. Shallow and Deep CNN features are complementary for object detection. Figure 5 shows the detailed process of feature fusion. We fused feature maps from the third, fourth, and fifth convolution layers (con_3, con_4, con_5). These feature maps have different resolutions because of subsampling and pooling operations. To get the feature maps of the same resolution, we adopted different sampling strategies. For the shallow layer, we added the maximum pooling to conduct subsampling. Additionally, we added the deconvolutional operation to carry out upsampling in the deep layer. The fourth convolutional layer remained at the original size. Then, we normalized multiple feature maps using local response normalization (LRN) [32]. For each feature map, an additional convolution layer was required. The feature maps were connected by the connection operation to form a new feature map. The new feature map contained shallow and deep layer information that was effective for object detection. The feature map resolution was more suitable for detection.

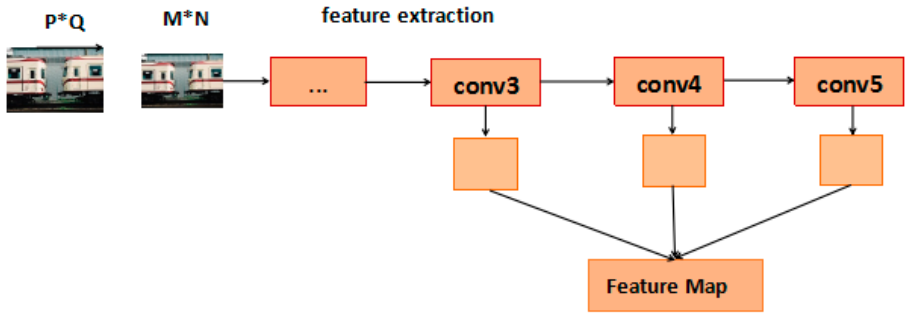


Figure 5. Detailed process of feature fusion. We fused feature maps from the third, fourth, and fifth convolution layers(con_3, con_4, con_5).

3.4. Visual Attention Mechanism

AF R-CNN uses the two parallel branches of the trunk and the attention branch to generate attention-aware features and then select regions on the feature map after adding an attention mask. Our attention modules were constructed by a mask branch. Given the feature map, $T(x)$, as the input, the mask branch learns the mask, $M(x)$ by using the bottom-up top-down structure [7,33–35]. The mask, $M(x)$, is used as control gates for $T(x)$. The attention module $H(x)$ is defined as:

$$H_{i,c}(x) = M_{i,c}(x) \times T_{i,c}(x) \tag{2}$$

where i ranges over all spatial positions and c represents channels.

Figure 6 display the architecture of the mask branch. The mask branch contains two steps: Feed-forward sweep and top-down feedback. The two steps behave as a bottom-up top-down fully convolutional structure. The feed-forward sweep operation extracts global information quickly. Additionally, the top-down feedback operation feeds back the global information to the feature map. Firstly, max pooling was performed in the fusion feature map to improve the receptive field. The global information, which was expanded by the top-down structure, guides features when reaching the lowest resolution. The extended structure up samples global prior information by bilinear interpolation. To make the size of the output feature map the same as the input feature map, the output dimension of the bilinear interpolation was the same as the max pooling. Then, the attention module used the sigmoid layer to normalize the output results, with $M(x)$. $M(x)$ ranging from 0 to 1. This was the attention weight of the feature map at each location. This weight matrix, $M(x)$, and the feature map matrix, $T(x)$, were multiplied to obtain the desired attention-weighted feature map.

The structure of this encoder-decoder (bottom-up top-down) is often used in image segmentation, such as FCN [7]. We applied a bottom-up top-down structure to the object detection net. The structure of bottom-up top-down first extracted high-level features and increased the receptive field of the model through a series of convolution and pooling. Pixels activated in the high-level feature map can reflect the area where attention is located. The size of the feature map was enlarged by the up sample to the same size as the original input, so that the area of the attention was mapped to each pixel of the input. We called this the attention map, ($M(x)$). Each pixel value in the attention map ($M(x)$) output by the soft mask branch was equivalent to the weight of each pixel value on the original feature map, which enhanced the meaningful features and suppressed meaningless information. This weight matrix, $M(x)$, and the feature map matrix, $T(x)$, were multiplied to obtain the desired attention-weighted feature map, ($H(x)$).

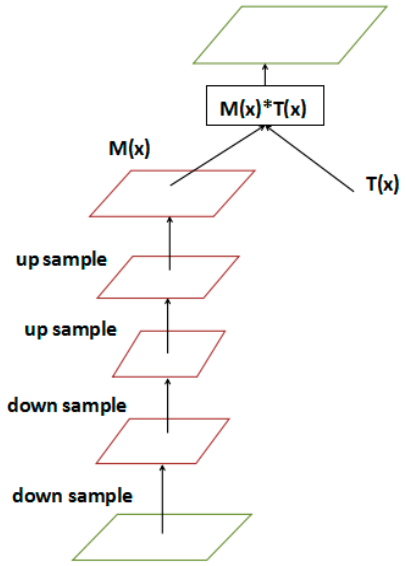


Figure 6. The architecture of the mask branch. The mask branch contains two steps: Feed-forward sweep and top-down feedback. The two steps behave as a bottom-up top-down fully convolutional structure.

The object detection module shares the convolution parameters with the attention module to achieve an integrated end-to-end object detection network. The mask branch aims at improving features rather than solving complex problems directly. It worked as a feature selector, which enhances good features and suppress noises from features.

3.5. AF R-CNN Loss Function

We minimize an objective function following the loss in the Faster R-CNN. The loss function in AF R-CNN is defined as:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{3}$$

Hence, i represents the index of the anchor. The p_i is the predicted probability whether anchor, i, is an object and t_i is a vector of four parameters in the predicted bounding box. The classification loss, L_{cls}, is the loss over two classes (object vs. not object). We used N_{cls} and N_{reg} to normalize two terms, cls(classification) and reg(regression).

The ground-truth label is defined as follows:

$$p_i^* = \begin{cases} 1, \textit{ the anchor is positive} \\ 0, \textit{ the anchor is negative} \end{cases} \tag{4}$$

The classification and regression loss are defined as:

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*) \tag{5}$$

where, R is the robust loss, which is defined as follows:

$$R(x) = \begin{cases} 0.5x^2 if\ |x| < 1 \\ |x| - 0.5 otherwise \end{cases} \tag{6}$$

We set $\lambda = 10$, so that the weight of the classification and regression terms were roughly equal. The t_i is defined as follows:

$$t_x = \frac{(x - x_a)}{w_a}, t_y = \frac{(y - y_a)}{h_a} \tag{7}$$

$$t_w = log(\frac{w}{w_a}), t_h = log(\frac{h}{h_a}) \tag{8}$$

$$t_x^* = \frac{(x^* - x_a)}{w_a}, t_y^* = \frac{(y^* - y_a)}{h_a} \tag{9}$$

$$t_w^* = log\left(\frac{w^*}{w_a}\right), t_h^* = log\left(\frac{h^*}{h_a}\right) \tag{10}$$

where x, y, w, and h are the center coordinates of the box and width and height. $x_a(y_a, w_a, h_a)$ represents the same mean in the anchor box and $x^*(y^*, w^*, h^*)$ represents the ground-truth box.

4. Experiments

We performed a series of comprehensive experiments on the dataset, PASCAL VOC 2007 [36] and 2012. The PASCAL VOC 2007 dataset contains 20 object categories. The pixel size of the image varies, and is usually (horizontal view) 500 × 375 or (longitudinal view) 375 × 500. The mean average precision was used to measure the performance of the object detection network. All experiments were performed on a ThinkStation P510 PC with Intel(R) Xeon(R) E5-2623 2.6GHZ CPU and Quadro M5000 GPU with an 8 GB memory. Our experiments were implemented in Python with TensorFlow.

In our experiments, the AF R-CNN was trained on a training set and the detection results were obtained on the test set (PASCAL VOC 2007 test set). Training data 07 represents VOC 2007 trainval and 07+12 represents the union set of VOC 2007 trainval and VOC 2012 trainval. AF R-CNN and faster R-CNN use the same pre-trained model, VGG-16, which has 13 convolutional layers and three fully-connected layers. All the networks were trained using stochastic gradient descent (SGD) [37]. The initial learning rate was set to 0.001. As AF R-CNN, we re-scaled the images and resized the short side to 600. The initial image needed to be processed. The size of the image was fixed to 224 × 224 by mapping the image. Then, the fixed size image was input into the VGG network for feature extraction. We compared the AF R-CNN and faster R-CNN for object detection on the dataset, PASCAL VOC 2007.

Our experiments contained two parts. In the first part, we analyzed the effectiveness of each component in the AF R-CNN, including feature fusion and the soft mask branch in the attention module. After that, we compared our network with state-of-the-art results in the PASCAL VOC 2007 dataset.

Table 1 shows the results of the different object detection network. The first and second columns represent the results of the Faster R-CNN on the dataset, PASCAL VOC 2007. The third(fourth) column shows the results of the method, which we only added the attention module (feature fusion) in the Faster R-CNN. Additionally, the last two columns show the results of AF R-CNN, which we proposed in the article that contained both feature fusion and the attention module.

To understand the benefit of the attention mechanism, we calculated the mean average precision of the object. The attention module in the AF R-CNN is designed to suppress noise while keeping useful information by applying the dot product between the feature map and soft mask. From Table 1, we can see that the performance of the object detection network with the addition of the attention module was improved. In the experiments, we evaluated the effectiveness of the attention mechanism. The Faster RCNN+attention achieved an mAP (mean average precision) of 70.3%, 0.4 points higher than the Faster R-CNN. To understand the benefit of the attention mechanism, we calculated the mAP (mean average precision) of 20 object categories. The attention mechanism can enhance the feature contrast. The attention-aware features were more effective for significant objects, such as the car, boat, and person. However, for small targets, the detection efficiency was slightly reduced, such as the bottle, bird, and plant. This is because the attention mechanism was more effective for significant

objects. However, for less significant targets, such as smaller targets and shallower targets, the attention mechanism did not achieve the desired results.

Table 1. Results of the Faster R-CNN and AF R-CNN on the dataset, PASCAL VOC 2007.

	Faster R-CNN	**Faster R-CNN**	**Faster R-CNN+Attention**	**Faster R-CNN+Fusion**	**Ours**	**Ours**
Training data	07	07+12	07	07	07	07+12
mAP	69.9	73.2	70.3	75.0	75.9	79.7
areo	70.0	76.5	71.4	73.6	76.2	83.0
bike	80.6	79.0	81.4	82.4	81.2	87.0
bird	70.1	70.9	63.9	75.1	77.9	81.4
boat	57.3	65.5	60.5	62.3	66.2	74.0
bottle	49.9	52.1	47.8	60.2	62.8	68.5
bus	78.2	83.1	79.5	80.2	80.2	87.7
car	80.4	84.7	81.5	83.3	86.3	88.0
cat	82.0	86.4	82.1	83.6	87.5	88.1
chair	52.2	52.0	50.3	59.3	56.9	62.4
cow	75.3	81.9	75.8	77.2	85.1	86.8
table	67.2	65.7	67.6	74.5	71.3	70.8
dog	80.3	84.8	81.6	84.8	87.2	88.6
horse	79.8	84.6	81.8	86.5	86.2	87.3
mbike	75.0	77.5	76.1	78.4	80.3	83.8
person	76.3	76.7	77.8	80.9	79.6	82.8
plant	39.1	38.8	35	53.8	47.3	53.2
sheep	68.3	73.6	68.8	70.4	77.3	81.1
sofa	67.3	73.9	67.8	72.2	75.2	77.6
train	81.1	83.0	83.3	83.2	79.1	84.3
tv	67.6	72.6	69.8	74.6	74.8	79.2

The mAP of the detection network Faster R-CNN+fusion was 75.0%, which was 5.1% higher than the Faster R-CNN on the dataset, PASCAL VOC 2007. As we have shown above, this is because the fusion features were more accurate than the deep convolution features. The results benefitted from more informative features. The reasonable resolution of features made for better object detection, especially when the object size was small. Our detection network outperformed the Faster R-CNN when the object size was small. For the plant, Faster R-CNN+fusion achieved a 53.8% mAP, a 14.7 points improvement, and for the bottle, the Faster R-CNN+fusion achieved a 60.2% mAP, 10.3 points higher than the Faster R-CNN. It showed that the multi-layer feature fusion could effectively improve the detection of small targets.

Our AF R-CNN outperformed the Faster R-CNN regardless of the object size. We compared our AF R-CNN with state-of-the-art methods, including Faster R-CNN [5], A-Fast-RCNN [19], CRAFT [38], and SSD [39] on the dataset, PASCAL VOC 2007. The results are shown in Table 2. Our AF R-CNN outperformed all methods on the PASCAL VOC 2007 dataset. The object detection network that we proposed in this article achieved an mAP of 75.9% on the dataset, PASCAL VOC 2007, 6 points higher than the Faster R-CNN because of feature fusion and the attention module. The above results suggest that our method enjoyed high efficiency and good performance.

Table 2. A comparison between the proposed method with the existing models on the dataset, PASCAL VOC 2007.

Method	**Faster R-CNN**	**A-Fast-RCNN [28]**	**CRAFT [37]**	**SSD [38]**	**Ours**
VOC 2007	69.9	71.4	75.7	71.6	75.9

Figure 7 shows the visualization of the object detection. Left: The input images. Middle: The object detection of the Faster R-CNN. Right: The object detection of AF R-CNN. We found a few more

representative pictures. First, in order to see the effect of our method on the object size, we selected a big plant (the fourth row) and a small bird (the second row). To assess the effectiveness of our approach to multiple goals and target defects, we chose the pictures in the first, third, and fifth cows. Of course, these pictures had crossovers, such as the last picture had many objects and the objects were very small. As can be seen from the results shown in Figure 7, the effect of the overall network was improved. The attention module helped detect significant targets while the feature fusion could effectively detect small targets. The experiments proved that our proposed method was more effective than Faster R-CNN.

(**a**) Original images (**b**) Faster R-CNN results (**c**) AF R-CNN results

Figure 7. Results of object detection. (**a**) shows the input images of the object detection networks; (**b**) shows the visualization of the detection results of the Faster R-CNN; (**c**) shows the visualization of the detection results of the AF R-CNN.

5. Conclusions

In this paper, we proposed AF R-CNN, a fully trainable deep architecture for object detection. AF R-CNN provides an efficient combination of the attention module and feature fusion for a deep object network. Our methods enhanced the impact of salient features and combined deep, but semantic, and

shallow, but high-resolution, CNN features effectively. Thus, AF R-CNN improved the overall object detection accuracy.

However, our model still needs to be improved in terms of speed and real-time. How to balance the computational complexity and performance remains a big challenge. In the future, we would like to discover a lower computational burden and system complexity. Also, better pre-trained models, like res-net, will be applied to the research with the development of deep networks.

Author Contributions: Methodology, Experimental analysis and Paper Writing, Y.Z.; Writing-review and Data analysis, Y.Z. and Y.C.; Data and Writing Correction, C.H. and M.G.; The work was done under the supervision and guidance of Y.C.

Funding: This work is partially supported by Shanghai Innovation Action Plan Project (No. 16511101200) of Science and Technology Committee of Shanghai Municipality.

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

References

1. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
2. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
3. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014.
4. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
5. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
6. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *arXiv*, 2015; arXiv:1512.03385.
7. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
8. Kong, T.; Yao, A.; Chen, Y.; Sun, F. HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 845–853.
9. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
10. Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; Zhang, L. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. *arXiv*, 2017; arXiv:1707.07998.
11. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv*, 2014; arXiv:1409.0473.
12. Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; Wei, Y. Relation Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
13. Kuo, W.; Hariharan, B.; Malik, J. Deepbox: Learning objectness with convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
14. Pepikj, B.; Stark, M.; Gehler, P.; Schiele, B. Occlusion patterns for object class detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013.
15. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S.; et al. Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3296–3297.

16. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
17. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
18. Chen, Y.; Li, W.; Sakaridis, C.; Dai, D.; Van Gool, L. Domain Adaptive Faster R-CNN for Object Detection in the Wild. *arXiv*, 2018; arXiv:1803.03243.
19. Wang, X.; Shrivastava, A.; Gupta, A. A-Fast-RCNN: Hard Positive Generation via Adversary for Object Detection. *arXiv*, 2017; arXiv:1704.03414.
20. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *arXiv*, 2017; arXiv:1703.06870.
21. Li, J.; Liang, X.; Wei, Y.; Xu, T.; Feng, J.; Yan, S. Perceptual Generative Adversarial Networks for Small Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1951–1959.
22. Koch, C.; Ullman, S. Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry. *Hum. Neurobiol.* **1987**, *4*, 219–227.
23. Itti, L.; Koch, C.; Niebur, E. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 1254–1259. [CrossRef]
24. Bruce, N.D.B.; Tsotsos, J.K. Saliency based on information maximization. In Proceedings of the International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 5–8 December 2005; pp. 155–162.
25. Borji, A. Boosting bottom-up and top-down visual features for saliency estimation. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 438–445.
26. Wang, X.; Yu, L.; Ren, K.; Tao, G.; Zhang, W.; Yu, Y.; Wang, J. Dynamic Attention Deep Model for Article Recommendation by Learning Human Editors' Demonstration. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 2051–2059.
27. Ren, P.; Chen, Z.; Ren, Z.; Wei, F.; Ma, J.; de Rijke, M. Leveraging Contextual Sentence Relations for Extractive Summarization Using a Neural Attention Model. In Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 95–104.
28. Seo, S.; Huang, J.; Yang, H.; Liu, Y. Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction. In Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, 27–31 August 2017; pp. 297–305.
29. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
30. Uijlings, J.R.; van de Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]
31. Alexe, B.; Deselaers, T.; Ferrari, V. Measuring the objectness of image windows. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2189–2202. [CrossRef] [PubMed]
32. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.B.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014.
33. Noh, H.; Hong, S.; Han, B. Learning deconvolution networkfor semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
34. Badrinarayanan, V.; Handa, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv*, 2015; arXiv:1505.07293.
35. Newell, A.; Yang, K.; Deng, J. Stacked hourglass networks for human pose estimation. *arXiv*, 2016; arXiv:1603.06937.
36. Everingham, M.; van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. 2007. Available online: http://www.pascal-network.org/challenges/VOC/voc2007/index.html (accessed on 2 June 2010).
37. Zinkevich, M.; Weimer, M.; Li, L.; Smola, A.J. Parallelized stochastic gradient descent. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 6–9 December 2010.

38. Yang, B.; Yan, J.; Lei, Z.; Li, S.Z. CRAFT Objects from Images. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 6043–6051.
39. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.