electronics

MDPI

Article

A Two-Branch Network for Weakly Supervised Object Localization

Chang Sun [1], Yibo Ai [1], Sheng Wang [2] and Weidong Zhang [1,*]

[1] National Center for Materials Service Safety, University of Science and Technology Beijing, Beijing 100083, China; b20160401@xs.ustb.edu.cn (C.S.); ybai@ustb.edu.cn (Y.A.)
[2] AI Lab, UCAR Inc., 118 East Zhongguancun Road, Haidian Dist, Beijing 100098, China; sheng.wang03@ucarinc.com
* Correspondence: zwd@ustb.edu.cn; Tel.: +86-10-6233-2239

Received: 20 April 2020; Accepted: 1 June 2020; Published: 8 June 2020

Abstract: Weakly supervised object localization (WSOL) has attracted intense interest in computer vision for instance level annotations. As a hot research topic, a number of existing works concentrated on utilizing convolutional neural network (CNN)-based methods, which are powerful in extracting and representing features. The main challenge in CNN-based WSOL methods is to obtain features covering the entire target objects, not only the most discriminative object parts. To overcome this challenge and to improve the detection performance of feature extracting related WSOL methods, a CNN-based two-branch model was presented in this paper to locate objects using supervised learning. Our method contained two branches, including a detection branch and a self-attention branch. During the training process, the two branches interacted with each other by regarding the segmentation mask from the other branch as the pseudo ground truth labels of itself. Our model was able to focus on capturing the information of all the object parts due to the self-attention mechanism. Additionally, we embedded multi-scale detection into our two-branch method to output two-scale features. We evaluated our two-branch network on the CUB-200-2011 and VOC2007 datasets. The pointing localization, intersection over union (IoU) localization, and correct localization precision (CorLoc) results demonstrated competitive performance with other state-of-the-art methods in WSOL.

Keywords: deep learning; neural network; weakly supervision; object localization; self-attention mechanism; multi-scale detection

1. Introduction

Object detection is a fundamental task in computer vision. It has been widely applied in the field of autonomous driving system and intelligent security system. The training process of object detection methods requires a lot of instance level annotations, which is time consuming and labor intensive. However, weakly supervised object localization (WSOL) performs the detection process with only labels for classification (image level labels), and no related bounding box labels (instance level labels) are provided. During training, the model is taught to classify objects based on the given image level labels. With the features learned in the classification process, the model is also asked to give the bounding box prediction results of target objects.

Multiple instance learning (MIL) [1] is always employed [2–4] in WSOL in order to obtain the locations of target objects. MIL regards an image as a bag of different instances, which are the proposed regions in the image. For a specific class category, if one or more related instances appear in the image, this image is taken as a positive sample. Otherwise, it is taken as a negative sample. A drawback

of MIL is that it is usually stuck into sub-optimal solutions owing to the focus of finding the most valuable object parts [5,6], making it difficult to locate objects accurately.

Currently, convolutional neural network (CNN)-based methods have been widely applied in object classification due to their power in extracting and representing features. Research has proven that CNN architecture trained for classification also contains object spatial information [7]. A CNN model can provide enough information to locate objects with only image level labels. Thus, CNN-based methods have played a key role in WSOL [8–12].

In WSOL, forcing the CNN model to extract features containing information of the entire target object, not only the most discriminative object part, is the key to achieving successful performance. Previous works addressed this challenge by finding second important information [13], extracting foreground regions [14], and randomly erasing or using the most discriminative object regions [15]. To overcome this challenge and to extract strong and powerful features, we focused on using attention mechanisms, which have been proved to effectively enhance the ability of network expression [16]. They have been used in a wide range of computer vision tasks (object detection [17], semantic segmentation [18], and image caption [19]). In the field of WSOL, attention mechanisms were also performed, including channel attention [15] capable of highlighting important channel features and spatial attention [20] capable of locating valuable regions in feature maps. Inspired by this, it is necessary to further exploit the effectiveness of attention mechanisms in WSOL. In order to obtain features with the ability of covering the entire objects, we used a self-attention mechanism [21], which aimed at building connections among image regions to capture information of the whole target object. To better improve the detection performance of CNN-based methods, we further applied multi-scale detection, which was preferred by object detection methods due to its ability to generate multiple scale features. The single-shot multibox detector [22] and feature pyramid networks [23] utilized multi-scale output features by down-sampling and up-sampling to detect target objects with different sizes. Multi-scale CNN [24] used multiple layers to match with target pedestrians of different sizes. They all proved that the detection performance could be enhanced by using multi-scale features in the CNN structure. However, we used the multi-scale output features to measure the different types of localization metrics, instead of dealing with the different sizes of the target objects.

In summary, we studied the potential of self-attention mechanisms and multi-scale detection by building a two-branch model that comprised a detection branch and a self-attention branch. During the training process, the two branches were intersected. A segmentation mask coming from the detection branch was treated as a pseudo ground true label of the self-attention branch, and vice versa. We embedded multi-scale detection into our proposed method by outputting two-scale features for each branch, including features with resolutions 1/8 and 1/16 the size of the input image. Our main contributions are listed below:

1. We presented a two-branch network for WSOL and a self-attention mechanism was embedded to improve the ability of feature expression by connecting object parts.
2. We applied multi-scale detection to output two-scale features in order to improve the detection performance in localization.
3. We evaluated the two-branch network on the Caltech-UCSD Birds-200-2011 (CUB-200-2011) [25] and VOC2007 [26] datasets. The evaluation results demonstrated competitive performance relative to the state-of-the-art methods.

Specifically, the rest of the paper is organized as follows. We introduce related work briefly in Section 2. Thereafter, we describe our proposed two-branch method in Section 3. Section 4 mainly shows the performance of our method on two datasets (CUB-200-2011 and VOC2007). The last section demonstrates the conclusions about our work.

2. Related Works

2.1. The CNN-Based Model for WSOL

WSOL is a significant task in computer vision and many related works concentrate on CNN methods. A number of researchers aimed to better exploit information in CNN features by finding all parts of the target object instead of only the most discriminative parts. Zhang et al. [13] proposed an adversarial complementary learning model containing two parallel-classifier branches to capture the most discriminative information and the second remarkable information. A similar idea was embedded in a two-phase learning model [27]. Kim et al. [27] employed a conventional fully convolutional network and an inference conditional feedback to obtain the most and the second salient parts in images. There are also researchers who focused on applying proposal-based methods to WSOL. Kantorov et al. [28] studied the context of region-of-interest (ROI) proposals with two types of context-aware models, including an additive model capable of supporting the network to find the object region and a contrastive model capable of making object regions more outstanding. Bilen et al. [29] presented a CNN architecture, named weakly supervised deep detection network (WSDDN). With the help of the object proposal method and a pooling layer designed to generate valuable spatial information, WSDDN can be powerful in locating objects. Additionally, other ideas with the intent of exploring the spatial information of CNN features were exploited. Durand et al. [30] presented a multi-map WSOL transfer layer and a new spatial aggregation process. Zhang et al. [14] introduced self-produced guidance (SPG) masks to provide foreground regions full of spatial correlation information. Zhu et al. [31] designed an object proposal component (soft proposal), which used a dissimilarity measure of feature difference and spatial distance between regions to generate proposal maps.

2.2. Attention Mechanism in CNN-Based WSOL Methods

In object detection, an attention mechanism aims at capturing the most discriminative features to improve the detection accuracy of CNN-based methods [17,32,33]. Previous works [20,34] in WSOL proved that with the armed attention mechanism, CNN models were able to provide rich localization related information. Teh et al. [35] proposed an attention network providing attention scores for candidate regions by linear mapping and softmax activation. Based on the model proposed in [35], Teh et al. [36] leveraged a regularized attention network to regularize the attention distribution of attention scores. Li et al. [37] performed a guided attention inference network with the purpose of generating more complete attention maps by applying the global average pooling layer to gradients of scores. Hu et al. [38] employed a weakly supervised bilinear attention network aiming at utilizing attention regularization and attention dropout to obtain attention maps. Bilinear attention pooling was used to deal with feature maps and attention maps to generate part features. Zhou et al. [20] combined spatial attention with channel attention to build a dual-attention focused module using global average pooling and channel average pooling to compute channel and spatial attention maps, respectively.

Previous studies [15,20] have exploited the effectiveness of channel attention and spatial attention in discovering the most discriminative parts of target objects. However, as mentioned earlier, simply focusing on the most valuable object parts information might not be enough to locate objects accurately in WSOL. How to extract CNN features containing useful information of the whole target object is still worth studying. In order to tackle this, we constructed a two-branch network embedded with a self-attention mechanism [21], which has been proved to be effective in building connections among regions to find global image structure in Generative Adversarial Networks, but has not been employed in WSOL.

3. The Proposed Method

3.1. Framework Overview

Figure 1 shows the overview of our two-branch network, containing a detection branch and a self-attention branch. CAM is a class activation map. BCE is the binary cross entropy. Output_1 and output_2 are features from the detection branch. Self-attention output_1 and self-attention output_2 are features from the self-attention branch. The training process of the two-branch network is displayed in Algorithm 1 and Figure 2. For an input image with size $H \times W \times 3$, through the backbone of our two-branch network, the output feature map ($F_{det}^{1/8}(x_i)$) was $1/8$ the size of the input image. By applying the detection branch, feature maps ($F_{det}^{1/16}(x_i)$) with the size of $1/16$ the input image were obtained. Then, with the help of the self-attention mechanism, which will be described in Section 3.3 , $F_{det}^{1/8}(x_i)$ and $F_{det}^{1/16}(x_i)$ were used in order to generate self-attention features for our self-attention branch. The generated self-attention features were the same size with their corresponding features from the detection branch, and are represented as $F_{att}^{1/8}(x_i)$ and $F_{att}^{1/16}(x_i)$.

Based on the output features, segmentation masks ($M_{det}(x_i)$) and $M_{att}(x_i)$) can be generated. The two-scale output features $F_{det}^{1/8}(x_i)$ and $F_{det}^{1/16}(x_i)$ were up-sampled by bilinear interpolation to the size of the input image. After that, element-wise maximum operation was applied to the up-sampled features to obtain fusion features ($F_{det}^{fuse}(x_i)$), which were used to calculate the class activation maps (CAMs) [7] $F_{det}^{CAMs}(x_i)$. The segmentation mask $M_{det}(x_i)$ was obtained by normalizing $F_{det}^{CAMs}(x_i)$ to the range of 0 to 1. The same operation was applied to $F_{att}^{1/8}(x_i)$ and $F_{att}^{1/16}(x_i)$ to obtain $M_{att}(x_i)$, which worked as the pseudo label for $F_{det}^{fuse}(x_i)$ and to compute the binary cross entropy (BCE) loss, and vice versa. Four identical classifiers, based on softmax cross entropy, were applied according to the given image level labels.

Compared with the detection branch, the output features in the self-attention branch contained additional information due to the self-attention mechanism. The intersection between the detection and self-attention branches made the whole model more effective in obtaining strong and valuable information of all the object parts for better locating targets in WSOL.

Algorithm 1 Training process for our two-branch network

Input: Training image $X = \{x_i, y_i\}_{i=1}^{N}$, threshold t

1: **while** the training process is far from convergence state **do**
2: Obtain features from detection branch and self-attention branch, $F_{det}^{1/8}(x_i)$, $F_{det}^{1/16}(x_i)$, $F_{att}^{1/8}(x_i)$, and $F_{att}^{1/16}(x_i)$
3: Calculate CAMs for the two branches according to their two-scale features,

$$F_{det}^{CAMs}(x_i) = CAMs(max(Bilinear(F_{det}^{1/8}(x_i)), Bilinear(F_{det}^{1/16}(x_i)))),$$
$$F_{att}^{CAMs}(x_i) = CAMs(max(Bilinear(F_{att}^{1/8}(x_i)), Bilinear(F_{att}^{1/16}(x_i))))$$

4: Generate segmentation masks for the two branches according to their CAMs,

$$M_{det}(x_i) = norm(F_{det}^{CAMs}(x_i)) > t,$$
$$M_{att}(x_i) = norm(F_{att}^{CAMs}(x_i)) > t$$

5: Calculate softmax loss and BCE loss
6: Update the two-branch network
7: **end while**

Output: M_{det}, M_{att}

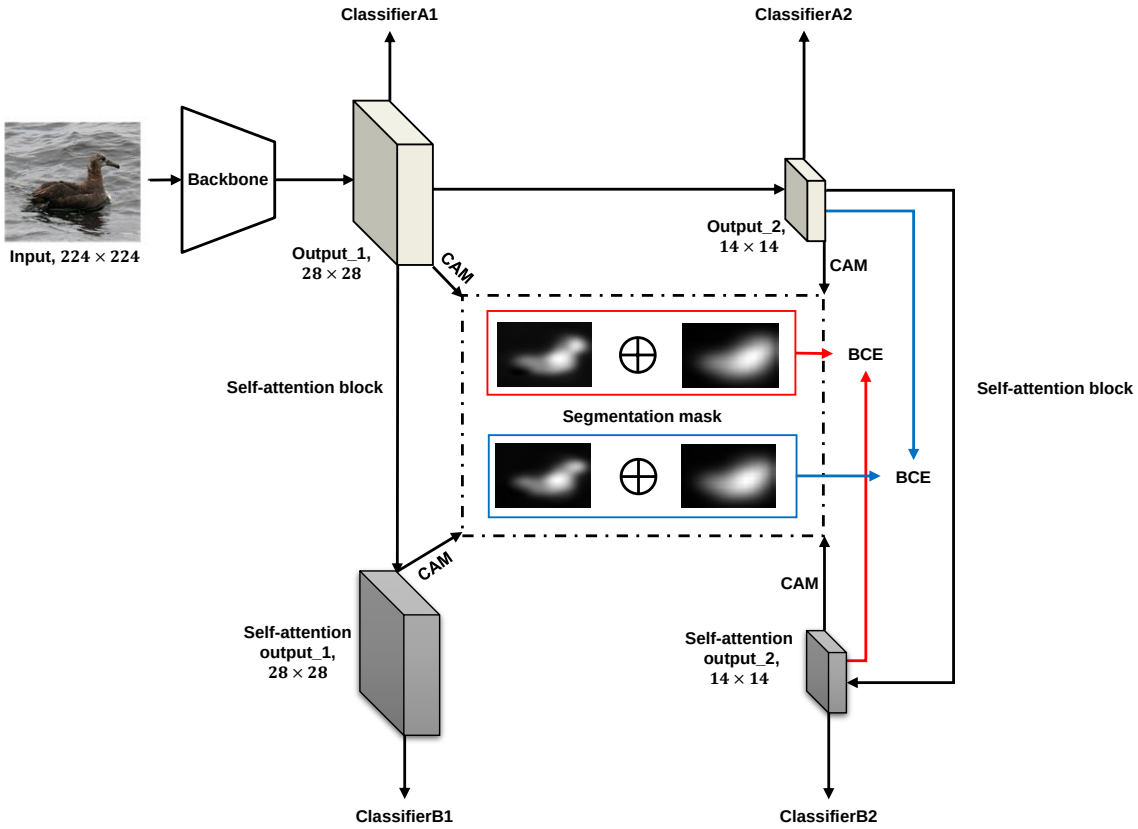


Figure 1. Overview of our proposed two-branch network. Class activation maps (CAMs), binary cross entropy (BCE).

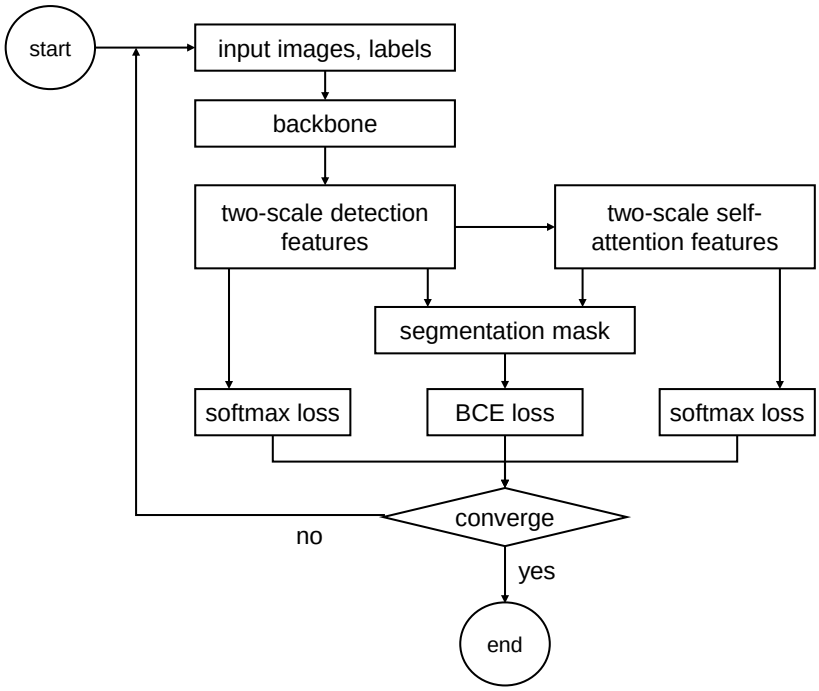


Figure 2. Flow chart of our proposed two-branch network.

3.2. The Detection Branch

We adopted VGG-16 [39] as the backbone of our two-branch network. Following the idea in [13], we dropped layers after the pool5 layer. Additionally, to better maintain the spatial information of the features in VGG-16, we changed the stride of the pool4 and pool5 layers from 2 to 1, making the scale of the pool5 layer 1/8 the size of the input image.

In the detection branch, the two-scale features were output, one was 1/8 the size of the input image (features from backbone), and the other one was 1/16 the size of the input image (features from the max-pooling operation, as shown in Figure 3). Features from the backbone were followed by two convolution layers (kernel size: 3×3; and stride: (1) with one rectified linear unit (ReLu) behind each of the convolution layer. Following a max-pooling operation (kernel size: 3×3; and stride: (2) to the corresponding output features.

For the two-scales output features, the class activation maps (CAMs) [7] were calculated based on the fusion features (generated by up-sampling the two-scales output features to the size of the input image and performing element-wise maximum to the up-sampled features), and segmentation masks were further obtained based on CAMs by thresholding. The mask worked as a pseudo segmentation label in the BCE loss for the self-attention branch.

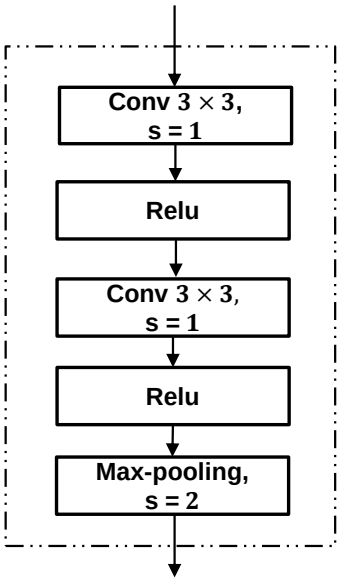


Figure 3. Architecture of the added feature extraction block to generate multi-scale output features.

3.3. The Self-Attention Branch

We adopted the self-attention mechanism in a previous study [21] to build our self-attention branch. The input of the self-attention branch was the output features in the detection branch. Two structure-identical self-attention blocks (Figure 4) were followed to process the two-scale input features in order to generate self-attention features. The self-attention block worked as follows: A convolution operation (kernel size: 1×1; and stride: 1) was applied to the input feature maps and the corresponding result was represented as **A** with resolution $C \times H \times W$. The same operation was performed to the same input feature maps another two times, and the corresponding results were represented as **B** and **C** with the same resolution $C \times H \times W$.

The feature maps **A** were reshaped to $C \times (H \times W)$ and transposed to $(H \times W) \times C$. The transposed feature maps were combined with the feature maps **B** by matrix multiplication. The resolution of the corresponding result was $(H \times W) \times (H \times W)$. A Softmax operation was applied to obtain the attention maps with a resolution of $(H \times W) \times (H \times W)$. Then, the attention maps were

combined with the feature maps **C** by matrix multiplication to generate self-attention feature maps of resolution $C \times (H \times W)$.

Finally, the reshaped self-attention feature maps of resolution $C \times H \times W$ were fused with the original input features according to a scale parameter, as shown in Equation (1). The self-attention branch focused on capturing connections among image regions [21] to obtain valuable information with the armed attention maps. With the help of the self-attention branch, the network can be forced to pay attention to full parts of target objects, not only the most discriminative object parts.

$$\mathbf{X} = \mathbf{X}_o + \gamma \mathbf{X}_{att} \tag{1}$$

where γ is the scale parameter. $\mathbf{X}_{att}$ is the self-attention feature maps and $\mathbf{X}_o$ is the original input feature maps.

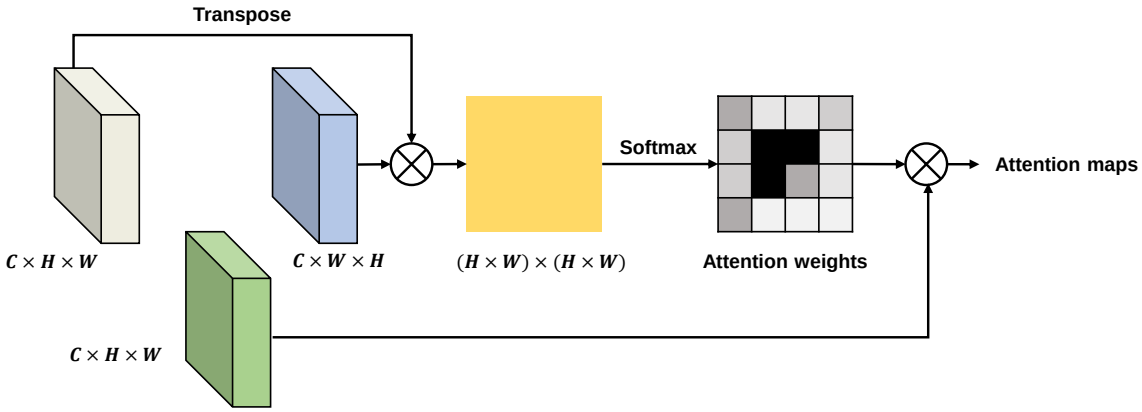


Figure 4. Architecture of the self-attention block [21].

3.4. Objective Function

We trained the two-branch method end-to-end. The loss of our method contained two parts, classification loss based on softmax cross entropy and segmentation loss based on BCE. It is represented by $L_{two-branch}$, as shown in Equation (2):

$$L_{two-branch} = L_{cla} + \lambda L_{BCE} \tag{2}$$

where L_{cla} is the function of the classification loss and adapted from a previous study [13]. L_{BCE} is the function of the segmentation loss and calculates the BCE loss, with all objects regarded as foreground, regardless of target-object category. The symbol λ represents a trade-off parameter between the classification loss and BCE loss and was set to 1 in this study.

4. Experiments and Discussion

4.1. Experiment Setup

The datasets we employed were CUB-200-2011 [25] and VOC2007 [26]. For the CUB-200-2011 dataset, the number of categories was 200. This dataset contained 11,788 images, with 5994 images in the training set and 5794 in the test set. For the VOC2007 dataset, the category number was 20. There were a total of 9963 images, with 5011 images in the training set (the original trainval set) and 4952 in the test set.

To evaluate our two-branch network, we used two types of metrics, containing a classification metric [40] (the widely used top-1 error and top-5 error) and three localization metrics, including an intersection over union (IoU) related localization metric [40], a correct localization precision (CorLoc) metric [41], and a pointing localization metric [31]. The IoU related localization metric estimates the

IoU between a ground truth bounding box and predicted bounding box. IoUs that exceed a defined threshold (set to 0.5) are regarded as correct if and only if the classification results are precise. CorLoc measures the percentage of images with a detected bounding box overlapped with a ground truth box higher than a threshold value (0.5) for each class [28]. The pointing localization metric is described in Equation (3).

$$Pointing\ accuracy = \frac{\sum_{k=1}^{C} \frac{Hits_k}{Hits_k + Misses_k}}{C} \quad (3)$$

where, for each class, a hit represents the predicted pixels of maximum response falls in a ground truth bounding within 15 pixels tolerance [31]. If not, it is called a miss. C is the number of class categories.

We used VGG-16 [39] as the backbone architecture in this work. The Pytorch framework was applied to implement the two-branch method and the model was trained on an NVIDIA GeForce GTX 2080Ti. During training, we used an input image size of 224×224 and weights trained from ImageNet [40] to initialize the backbone with learning rate of 3×10^{-4}. This was decayed at 20 and 40 epochs by decay rate of 1×10^{-1} with a total of 60 training epochs.

4.2. Experimental Results and Discussion

4.2.1. Performance on the CUB-200-2011 Dataset

We trained our method on the CUB-200-2011 training set and evaluated our training model on the CUB-200-2011 test set. Table 1 shows the classification results of the top-1 error and top-5 error. We compared our proposed method with the state-of-the-art approaches attention-based dropout layer (ADL) [15], SPG [14], and adversarial complementary learning (ACoL) [13]. A comparison result on the classification indicated that our method outperformed SPG [14] by 3.05% (top-1 error) and 2.02% (top-5 error) and outperformed ACoL [13] by 2.01% (top-1 error) and 1.50% (top-5 error).

Table 1. Classification error (%) on the CUB-200-2011 test set.

Methods	Top-1 Err.	Top-5 Err.
ADL [15]	25.45	-
SPG# [14]	25.18	7.65
ACoL# [13]	24.14	7.13
ours	**22.13**	**5.63**

We trained the self-produced guidance (SPG)# and adversarial complementary learning (ACoL)# models using the open source codes [42,43].

Pointing and IoU localization metrics were both performed to evaluate our method, ADL [15], SPG [14], and ACoL [13] on the CUB-200-2011 test set. The evaluation results are shown in Tables 2 and 3, respectively. For the pointing localization metric, our two-branch method achieved 24.28% in top-1 error, which was 2.75% lower than SPG [14] and 5.23% lower than ACoL [13]. The top-5 error of our method was 8.28%, which was 1.78% lower than SPG [14] and 6.34% lower than ACoL [13]. For the IoU localization metric, the top-1 error of our method was 51.22%, which was 2.14% lower than SPG [14] and 2.86% lower than ACoL [13]. Compared with ADL [15], our result was 4.26% higher. The top-5 error of our method was 41.55%, which was 0.73% lower than SPG [14] and 1.94% lower than ACoL [13].

Figure 5 visualizes the attention maps, pointing, and bounding box detection results on some sample images in the CUB-200-2011 test set. The attention maps of resolution 28×28 and 14×14 were obtained by applying the element-wise maximum to features from the detection and self-attention branches of the same size, respectively. In Section 4.2.3, we will show that the attention maps of resolution 28×28 and 14×14 were sensitive in obtaining better IoU localization performance and pointing localization performance, respectively. In Figure 5c,d, the red boxes are the ground truth bounding boxes. Figure 5c,d shows that our method was able to locate target objects effectively.

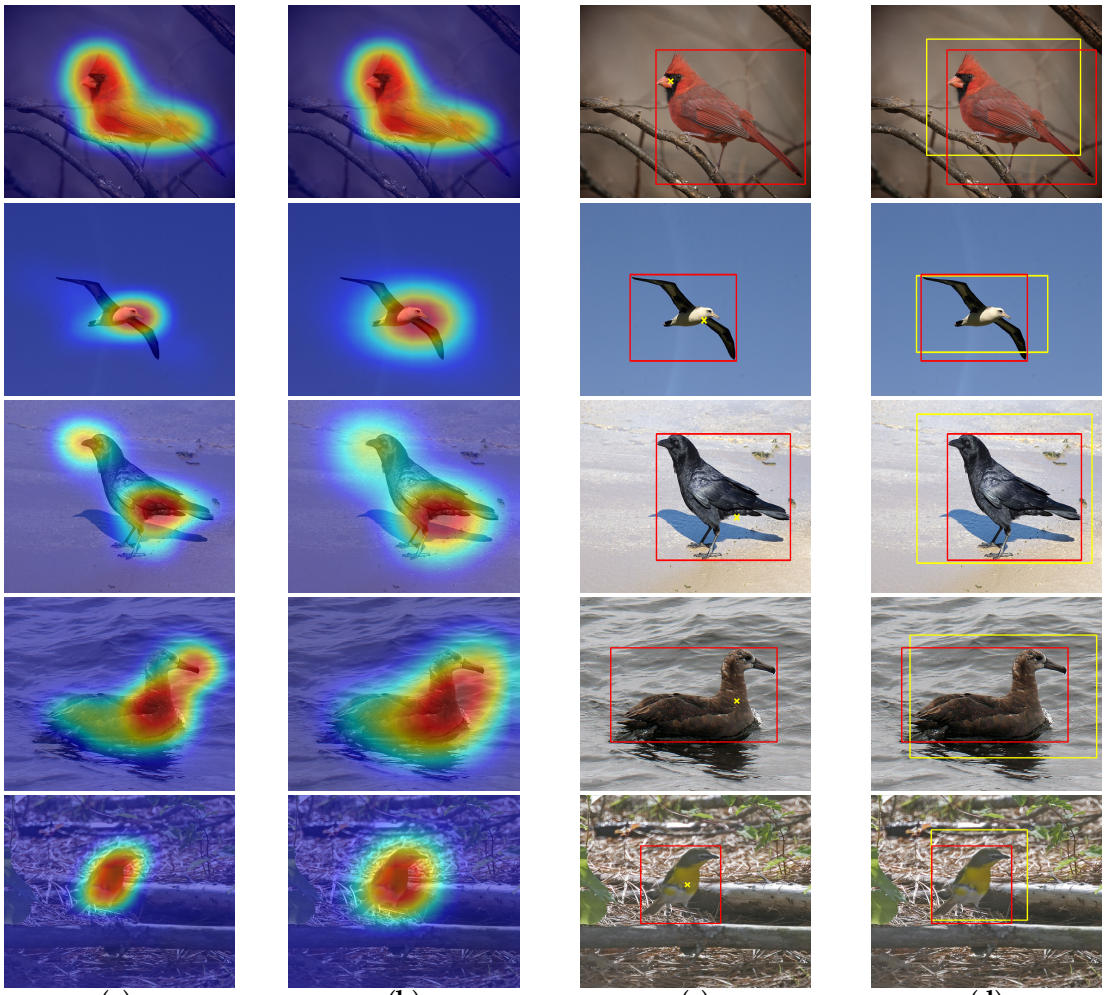


Figure 5. Visualization results in the CUB-200-2011 test set. (**a**) Attention maps (28×28), (**b**) attention maps (14×14), (**c**) pointing detection results, and (**d**) bounding box detection results.

Table 2. Pointing localization error (%) on the CUB-200-2011 test set.

Methods	Top-1 Err.	Top-5 Err.
ACoL# [13]	29.51	14.62
SPG# [14]	27.03	10.06
ours	**24.28**	**8.28**

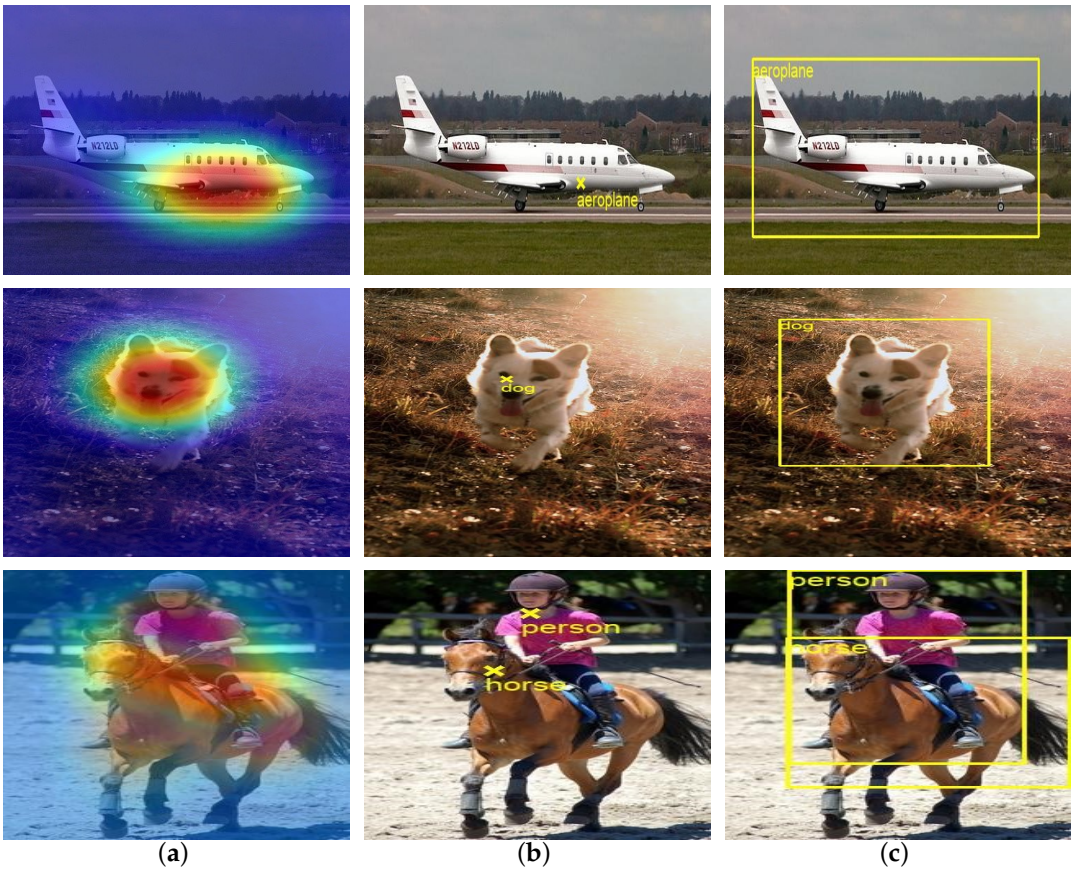
We trained the SPG# and ACoL# models using the open source codes [42,43].

Table 3. Intersection over union (IoU) localization error (%) on the CUB-200-2011 test set.

Methods	Top-1 Err.	Top-5 Err.
CAM [7]	59.00	-
ACoL [13]	54.08	43.49
SPG [14]	53.36	42.28
ADL [15]	**46.96**	-
ours	51.22	**41.55**

4.2.2. Performance on the VOC2007 Dataset

We then evaluated the performance of the two-branch network on the VOC2007 dataset. The comparison results on the VOC2007 test set with other methods were measured by pointing localization accuracy (Table 4) and Corloc (Table 5). The pointing localization accuracy of our method was 82.20%, which was 2.90% higher than the MWP [44] model, but 2.90% lower than the c-MWP [44] model. The Corloc of our method was 30.65%, which was 6.87% higher than the second best method, SPG [14]. Additionally, our result was 7.73% higher than ACoL [13] and 8.66% higher than Wildcat [30]. These results indicated the competitive performance of our method with other methods. Figure 6 visualizes the attention maps, pointing detection, and bounding box detection results on some sample images in the VOC2007 test set. We only showed attention maps for resolution 14×14, which were obtained by applying the element-wise maximum to features from the detection and self-attention branches of the same size. The attention feature maps that contained more than one target object were the fusion results by using those that contained each one of the targets.



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

Figure 6. Visualization results for the VOC2007 test set. (**a**) Attention maps, (**b**) pointing detection results, and (**c**) bounding box detection results.

Table 4. Pointing localization accuracy (%) on the VOC2007 test set.

Methods	Accuracy
ACoL# [13]	61.27
Wildcat# [30]	62.18
SPG# [14]	77.43
MWP [44]	79.30
CAM [7]	80.80
c-MWP [44]	**85.10**
ours	82.20

We trained the SPG#, ACoL#, and Wildcat# models using the open source codes [42,43,45].

Table 5. Corloc results (%) on the VOC2007 test set.

Methods	Accuracy
Wildcat# [30]	21.99
ACoL# [13]	22.92
SPG# [14]	23.78
ours	**30.65**

We trained the SPG#, ACoL#, and Wildcat# models using the open source codes [42,43,45].

4.2.3. Ablation Study

In this work, we output two layers of feature maps for each branch, making the number of output feature layers four. The scales of the output feature maps were 28×28 and 14×14. We compared these two-scale output features by pointing localization and IoU localization metrics, and results are shown in Table 6. We used the fusion features generated by applying element-wise maximum operations to feature maps of the same scale from the detection and self-attention branches to evaluate the performance of our method. Table 6 indicates that feature maps of resolution 14×14 performed better in pointing localization metric and feature maps of resolution 28×28 can obtain better performance in the IoU localization metric. This finding demonstrated that multi-scale output features in our two-branch method were effective.

Table 6. Comparison of feature maps with different scales on the CUB-200-2011 dataset.

Feature Map Scales	**28 × 28**		**14 × 14**	
Pointing localization	top-1 err.	top-5 err.	top-1 err.	top-5 err.
	26.57	11.11	**24.28**	**8.28**
IoU localization	top-1 err.	top-5 err.	top-1 err.	top-5 err.
	51.22	**41.55**	61.00	53.12

We further compared the localization performance of the detection branch with the self-attention branch. As proven earlier, feature maps of scale 14×14 were a better choice to evaluate the pointing localization accuracy. Therefore, we used feature maps with a resolution of 14×14 to calculate the pointing localization error of the two branches. Table 7 demonstrates that fusion feature maps outperformed feature maps from the detection branch or self-attention branch only.

For IoU localization, we also testified that feature maps of scale 28×28 were more suitable. To this end, feature maps with a resolution of 28×28 were utilized to obtain the IoU localization error of the detection branch and self-attention branch. Table 7 shows that fusion feature maps outperformed feature maps from the detection branch or self-attention branch only.

Table 7. Comparison of feature maps from different branches on the CUB-200-2011 dataset.

Feature Maps	**Detection Branch Features**		**Self-Attention Branch Features**		**Fusion Features**	
Pointing localization	top-1 err.	top-5 err.	top-1 err.	top-5 err.	top-1 err.	top-5 err.
	24.43	8.42	24.36	8.42	**24.28**	**8.28**
IoU localization	top-1 err.	top-5 err.	top-1 err.	top-5 err.	top-1 err.	top-5 err.
	51.73	42.26	51.68	42.17	**51.22**	**41.55**

Direct comparison of the detection branch with the self-attention branch revealed that the localization performance (point localization error and IoU localization error) of the two branches were relatively close and that the self-attention branch outperformed the detection branch slightly (the point localization top-1 error was 0.07% lower and the IoU localization top-1 error was 0.05% lower). We propose that the reason is that the two branches were intersected during the training

process by regarding the segmentation mask of the other branch as the pseudo ground truth label of itself, making the two branches improve together.

We compared the performance of different backbones (VGG-16 and GoogLeNet [46]). The two backbones shared the same input image size. Weights from ImageNet were both used to initialize the two backbones. The learning rates of VGG-16 and GoogLeNet were 3×10^{-4} and 1×10^{-3}. They decayed at 20 and 40 epochs for VGG and 30 and 60 epochs for GoogLeNet by a decay rate of 1×10^{-1} with a total of 60 and 90 training epochs, respectively. Table 8 shows the comparison results. The results indicated that the VGG backbone outperformed the backbone GoogLeNet, both in classification and localization. Specifically, the VGG backbone outperformed the GoogLeNet backbone by 3.38% (top-1 error) and 2.79% (top-5 error) in classification evaluation. As for the pointing localization and IoU localization, the VGG backbone outperformed the GoogLeNet backbone by 2.86%, 2.16% (top-1 error, top-5 error) and 8.78%, 9.43% (top-1 error, top-5 error).

Table 8. Comparison of different backbones on the CUB-200-2011 test set.

Backbones	**VGG**		**GoogLeNet**	
Classification	top-1 err.	top-5 err.	top-1 err.	top-5 err.
	22.13	**5.63**	25.51	8.42
Pointing localization	top-1 err.	top-5 err.	top-1 err.	top-5 err.
	24.28	**8.28**	27.14	10.44
IoU localization	top-1 err.	top-5 err.	top-1 err.	top-5 err.
	51.22	**41.55**	60.00	50.98

4.2.4. Result Analysis

We evaluated our proposed two-branch network on two datasets, CUB-200-2011 filled with single object per image and VOC2007 filled with multi-objects per image. For CUB-200-2011 dataset, three metrics (classification, pointing localization, and IoU localization) were applied to evaluate the performance of our method and other methods. Considering the classification and pointing localization results, our method achieved the best performance, and it outperformed the second best methods (ACoL and SPG) by 2.01% and 2.75% on the top-1 error. Considering the IoU localization results, ADL achieved the best performance; the top-1 error of our method was 4.26% higher. However, the top-1 classification error of our method was 3.32% lower than ADL. For VOC dataset, two metrics (pointing localization and Corloc) were applied to evaluate the performance of our method and other methods. Considering the Corloc results, our method achieved the best performance, and it outperformed the second best methods (SPG) by 6.87%. Considering the pointing localization results, our method achieved the second best performance, and it was 2.90% lower than the best performance method (c-MWP). Evaluation results demonstrated that our proposed two-branch model was competitive with state-of-the-art methods in WSOL by introducing multi-scale output features and self-attention mechanism.

Our model contained a detection branch and a self-attention branch. The two branches intersected with each other by regarding the segmentation mask from the other branch as the pseudo ground truth labels of itself. In the inference process, the features used to predict were the fusion results of the features from the detection branch and the self-attention branch of the same size. We proved that the fusion features achieved better performance than features from the detection branch or the self-attention branch only. In addition, the two-scale features were used to predict the target objects. We proved that feature maps of scale 28×28 and 14×14 were suitable for IoU localization and pointing localization, respectively. These findings demonstrated that the multi-scale output features suited our method, and the two-branch structure was also effective.

5. Conclusions

In this work, we developed a two-branch network by embedding multi-scale output features and a self-attention mechanism to capture information of the entire target objects not only the most discriminative object parts. The self-attention mechanism armed our model with the ability to connect different regions of objects. The multi-scale detection technique helped our model to achieve better performance in locating target objects. We compared our two-branch network with other methods focusing on exploiting powerful information of the entire object. Experiments on the CUB-200-2011 and VOC2007 datasets verified that the two-branch method was competitive against other methods not only in detecting single objects but also in detecting multi-objects of an input image. In the future, we plan to improve the performance of our method in detecting multi-objects by building multi-connections, which could aggregate the regions of the same target and split the regions of different targets.

Author Contributions: Conceptualization, C.S. and Y.A.; methodology, C.S.; software, C.S. and S.W.; supervision, W.Z.; validation, C.S.; funding acquisition, Y.A. and W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Funds for Central Universities of China (Nos. FRF-GF-18-009B, FRF-MP-19-014 and FRF-BD-19-001A) and the 111 Project (grant No. B12012).

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

References

1. Dietterich, T.G.; Lathrop, R.H.; Lozano-Pérez, T. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.* **1997**, *89*, 31–71.
2. Li, D.; Huang, J.B.; Li, Y.; Wang, S.; Yang, M.H. Weakly supervised object localization with progressive domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3512–3520.
3. Wang, C.; Ren, W.; Huang, K.; Tan, T. Weakly supervised object localization with latent category learning. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 431–445.
4. Gokberk Cinbis, R.; Verbeek, J.; Schmid, C. Multi-fold mil training for weakly supervised object localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 2409–2416.
5. Wan, F.; Liu, C.; Ke, W.; Ji, X.; Jiao, J.; Ye, Q. C-MIL: Continuation multiple instance learning for weakly supervised object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 2199–2208.
6. Wei, Y.; Shen, Z.; Cheng, B.; Shi, H.; Xiong, J.; Feng, J.; Huang, T. Ts2c: Tight box mining with surrounding segmentation context for weakly supervised object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Venue, Munich, Germany, 8–14 September 2018; pp. 434–450.
7. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929.
8. Jie, Z.; Wei, Y.; Jin, X.; Feng, J.; Liu, W. Deep self-taught learning for weakly supervised object localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1377–1385.
9. Zhang, Y.; Bai, Y.; Ding, M.; Li, Y.; Ghanem, B. Weakly-supervised object detection via mining pseudo ground truth bounding-boxes. *Pattern Recognit.* **2018**, *84*, 68–81.
10. Shen, Y.; Ji, R.; Wang, C.; Li, X.; Li, X. Weakly supervised object detection via object-specific pixel gradient. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 5960–5970.
11. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 685–694.

12. Diba, A.; Sharma, V.; Pazandeh, A.; Pirsiavash, H.; Van Gool, L. Weakly supervised cascaded convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 914–922.
13. Zhang, X.; Wei, Y.; Feng, J.; Yang, Y.; Huang, T.S. Adversarial complementary learning for weakly supervised object localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 1325–1334.
14. Zhang, X.; Wei, Y.; Kang, G.; Yang, Y.; Huang, T. Self-produced guidance for weakly-supervised object localization. In Proceedings of the European Conference on Computer Vision (ECCV), Venue, Munich, Germany, 8–14 September 2018;pp. 597–613.
15. Choe, J.; Shim, H. Attention-based dropout layer for weakly supervised object localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 2219–2228.
16. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3156–3164.
17. Zhu, Y.; Zhao, C.; Guo, H.; Wang, J.; Zhao, X.; Lu, H. Attention couplenet: Fully convolutional attention coupling network for object detection. *IEEE Trans. Image Process.* **2018**, *28*, 113–126.
18. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 3146–3154.
19. Liu, M.; Li, L.; Hu, H.; Guan, W.; Tian, J. Image caption generation with dual attention mechanism. *Inf. Process. Manag.* **2020**, *57*, 102178.
20. Zhou, Y.; Chen, Z.; Shen, H.; Liu, Q.; Zhao, R.; Liang, Y. Dual-attention Focused Module for Weakly Supervised Object Localization. *arXiv* **2019**, arXiv:1909.04813.
21. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-attention generative adversarial networks. *arXiv* **2018**, arXiv:1805.08318.
22. 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 (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
23. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
24. Cai, Z.; Fan, Q.; Feris, R.S.; Vasconcelos, N. A unified multi-scale deep convolutional neural network for fast object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 354–370.
25. Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; Perona, P. *Caltech-UCSD Birds 200*; Technical Report CNS-TR-2010-001; California Institute of Technology: Pasadena, CA, USA, 2010.
26. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338.
27. Kim, D.; Cho, D.; Yoo, D.; So Kweon, I. Two-phase learning for weakly supervised object localization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 3534–3543.
28. Kantorov, V.; Oquab, M.; Cho, M.; Laptev, I. Contextlocnet: Context-aware deep network models for weakly supervised localization. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 350–365.
29. Bilen, H.; Vedaldi, A. Weakly supervised deep detection networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26–30 June 2016; pp. 2846–2854.
30. Durand, T.; Mordan, T.; Thome, N.; Cord, M. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 642–651.
31. Zhu, Y.; Zhou, Y.; Ye, Q.; Qiu, Q.; Jiao, J. Soft proposal networks for weakly supervised object localization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1841–1850.

32. Tao, X.; Gong, Y.; Shi, W.; Cheng, D. Object detection with class aware region proposal network and focused attention objective. *Pattern Recognit. Lett.* **2018**, *130*, 353–361.
33. Li, H.; Liu, Y.; Ouyang, W.; Wang, X. Zoom out-and-in network with map attention decision for region proposal and object detection. *Int. J. Comput. Vis.* **2019**, *127*, 225–238.
34. Jiang, W.; Zhao, Z.; Su, F. Weakly supervised detection with decoupled attention-based deep representation. *Multimed. Tools Appl.* **2018**, *77*, 3261–3277.
35. Teh, E.W.; Rochan, M.; Wang, Y. Attention Networks for Weakly Supervised Object Localization. In Proceedings of the British Machine Vision Conference (BMVC), York, UK, 19–22 September 2016;pp. 1–11.
36. Teh, E.W.; Guo, Z.; Wang, Y. Object localization in weakly labeled data using regularized attention networks. In Proceedings of the 2017 IEEE Visual Communications and Image Processing (VCIP), St. Petersburg, FL, USA, 10–13 December 2017; pp. 1–4.
37. Li, K.; Wu, Z.; Peng, K.C.; Ernst, J.; Fu, Y. Tell me where to look: Guided attention inference network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 9215–9223.
38. Hu, T.; Xu, J.; Huang, C.; Qi, H.; Huang, Q.; Lu, Y. Weakly supervised bilinear attention network for fine-grained visual classification. *arXiv* **2018**, arXiv:1808.02152.
39. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
40. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252.
41. Deselaers, T.; Alexe, B.; Ferrari, V. Weakly supervised localization and learning with generic knowledge. *Int. J. Comput. Vis.* **2012**, *100*, 275–293.
42. Zhang, X. SPG. 2018. Available online: https://github.com/xiaomengyc/SPG (accessed on 16 December 2019).
43. Zhang, X. ACoL. 2018. Available online: https://github.com/xiaomengyc/ACoL (accessed on 3 December 2019).
44. Zhang, J.; Bargal, S.A.; Lin, Z.; Brandt, J.; Shen, X.; Sclaroff, S. Top-down neural attention by excitation backprop. *Int. J. Comput. Vis.* **2018**, *126*, 1084–1102.
45. Durand, T. wildcat.pytorch. 2017 Available online: https://github.com/durandtibo/wildcat.pytorch (accessed on 20 December 2019).
46. Christian, S.; Wei, L.; Yangqing, J.; Pierre, S.; Scott, R.; Dragomir, A.; Dumitru, E.; Vincent, V.; Andrew, R.; others. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.