WEAKLY SUPERVISED OBJECT LOCALIZATION WITH DEEP CONVOLUTIONAL NEURAL NETWORK BASED ON SPATIAL PYRAMID SALIENCY MAP

Zhiqiang Wan, Haibo He

University of Rhode Island
Department of Electrical, Computer and Biomedical Engineering
Kingston, Rhode Island, USA

ABSTRACT

Supervised object localization requires detailed image annotation, such as bounding box, to indicate the location of the object. However, labeling image with bounding box is labor-intensive. Besides, the labeling process may involve ambiguous decisions. Weakly supervised object localization only needs category annotation which is available in large amounts. Recently, many weakly supervised object localization methods, based on global pooling, have been proposed. However, these methods only localize part of the object. This paper proposes a deep convolutional neural network with spatial pyramid saliency map to localize the full extent of the object. The experimental result on Cub-200 dataset shows that our method outperforms the traditional ones.

Index Terms— Weakly supervised object localization, spatial pyramid saliency map, deep learning.

1. INTRODUCTION

Object localization is to distinguish an object from the background in an image. We can easily localize an object despite cluttered background, various illumination, and occluded object. However, in computer vision community, object localization is still a challenging task because the object can appear at different locations with different scales.

Supervised object localization [1, 2, 3] requires detailed image annotation, such as bounding box, to indicate the position of the object in an image. In general, it is labor-intensive to label a large number of images with bounding box. Besides, this process may involve many subtle and even ambiguous decisions[4], especially when the object is occluded. Weakly supervised object localization [5, 6] only requires category annotation which is available in large amounts in comparison with the bounding box annotation [4].

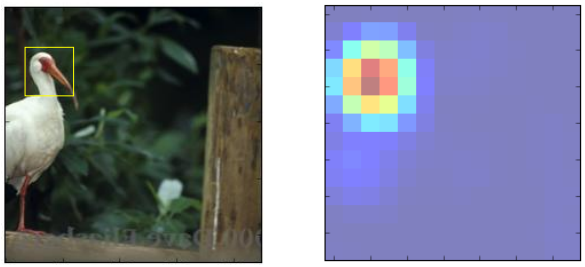
Recently, many researchers have explored weakly supervised object localization using CNN which can localize the object in its convolutional layer [7]. However, this ability is lost when the fully-connected layers are used [8]. In order to take advantage of CNN to localize the object, Oquab *et al.* [4] replace the fully-connected layers with a global max pooling layer. But this method only localizes a point on the boundary of the object. In order to improve the localization performance, Zhou *et al.*[9] apply global average pooling layer to replace the fully-connected layers. However, this method only localizes part of the object. In above methods, the global pooling layer pools the whole feature map together and ignores useful spatial information [10].

In this paper, we propose to replace the fully-connected layers with a spatial pyramid pooling layer to localize the full extent of the object. The spatial pyramid pooling layer partitions the output of CNN into local spatial regions and pools these local spatial regions to keep the spatial information. Spatial pyramid pooling, which has been widely used in computer vision community [11, 12, 13], is robust to object deformation[12] and can improve object localization performance.

Since the experimental result demonstrates that the feature map in CNN can be activated by some specific input patterns, and the activated region corresponds to the position of the object, we propose a spatial pyramid saliency map to localize the object. In order to get the saliency map, first of all, the input image is classified by our model. Then, we identify the importance of the image regions for the classification process. Finally, the importance is represented by the saliency map. Since the object is the most important part in the input image for the classification process, the saliency map can highlight the object. Based on this saliency map, we generate a bounding box to indicate the location of the object. The experimental result on Cub-200 dataset demonstrates that our method outperforms the traditional ones.

The main contributions of this paper are in twofold. First, we propose to replace the fully-connected layers with a spatial pyramid pooling layer in a CNN. Second, we propose a spatial pyramid saliency map to localize the object in the input image.

This work was partially supported by National Science Foundation (NSF) under grant CCF-1439011 and CMMI-1526835.



(a) Input image (b) Activations of feature map

Fig. 1. (Best seen in color.) Relationship between input image and activations of feature map. The activated region in (b) corresponds to the yellow rectangle in the input image.

2. PROPOSED METHOD

2.1. Motivation

The feature map of CNN can be activated by some specific input patterns, and the activated region of the feature map corresponds to the position of the object in the input image. Fig. 1 illustrates the relationship between the input image and activations of a feature map which is selected from the last convolutional layer of our model. When the input image in Fig. 1a is fed into our model, we can get the activations of the feature map in Fig. 1b. The high activation value is represented by red while the low activation value is represented by blue. It demonstrates that this feature map is activated by bird's head. Moreover, the activated region of the feature map corresponds to the position of the bird's head which is marked by a yellow rectangle in the input image. Therefore, we can localize the object based on the activated region of the feature map.

2.2. Architecture of our model

The architecture of our model, which consists of four parts, is shown in Fig. 2. The first part is a pretrained VGG-Net [14]. The second part is a convolutional layer. The third part is a spatial pyramid pooling layer. The final part is a softmax layer.

In order to localize the object in the input image, first of all, the input image is classified by our model. Since the input image has different size, we crop and resize it into 224×224. The mean RGB value which is computed on the training set is subtracted from each pixel of the resized image. Then, the processed image is fed into a pretrained VGG-Net. On top of the VGG-Net, we add a convolutional layer which has 3×3 receptive field and outputs 1024 feature maps. These feature maps are fed into the spatial pyramid pooling layer which consists of global pooling and quarter pooling. The pooling technique used in our paper is average pooling. The global pooling method pools the whole region of a feature map, denoted as $r1$, into one unit. The quarter pooling method pools each quarter of a feature map, denoted as $r2, r3, r4$, and $r5$, into one unit. Each feature map is pooled by global pooling and quarter pooling separately. Therefore, each feature map is pooled into 5 units by spatial pyramid pooling. Since the previous layer has 1024 feature maps, the output of the spatial pyramid pooling layer is concatenated to form a 5×1024 dimensional vector. Then, this vector is fed into the softmax layer. The output of the softmax layer is the prediction of probabilities for the multi-categories.

2.3. Spatial pyramid saliency map

After the classification process, a spatial pyramid saliency map is proposed to localize the object. The spatial pyramid saliency map is derived based on the activations of the feature maps of the last convolutional layer. The activation of the unit at spatial location (x, y) in the kth feature map is denoted as $f_k(x, y)$. The kth feature map is pooled by global average pooling and quarter average pooling respectively. The global pooling method pools the kth feature map into one unit which is shown in Eq.(1) where $r1$ denotes the whole region of the kth feature map, and N_1 denotes the number of units in region $r1$.

$$z_{k1} = \frac{1}{N_1} \sum_{x,y \in r1} f_k(x, y) \tag{1}$$

The quarter pooling method pools the kth feature map into four units z_{k2}, z_{k3}, z_{k4}, and z_{k5}. For example, z_{k2} is shown in Eq.(2) where $r2$ denotes the quarter of the kth feature map, and N_2 denotes the number of units in region $r2$.

$$z_{k2} = \frac{1}{N_2} \sum_{x,y \in r2} f_k(x, y) \tag{2}$$

Then, z_{k1}, z_{k2}, z_{k3}, z_{k4}, and z_{k5} are fed into the softmax layer. The input of the softmax layer for category c is denoted as S_c in Eq.(3) where w^c_{k1}, w^c_{k2}, w^c_{k3}, w^c_{k4} and w^c_{k5} are the weights connecting category c with corresponding units.

$$\begin{aligned} S_c = &\sum_k w^c_{k1} z_{k1} + \sum_k w^c_{k2} z_{k2} + \sum_k w^c_{k3} z_{k3} \\ &+ \sum_k w^c_{k4} z_{k4} + \sum_k w^c_{k5} z_{k5} \end{aligned} \tag{3}$$

Substitute Eq.(1) and Eq.(2) into Eq.(3), we can get

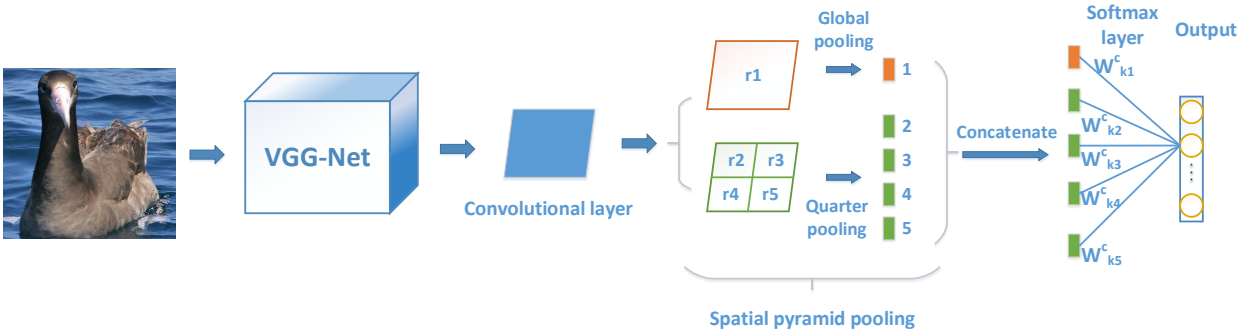


Fig. 2. Architecture of our model. Our model consists of four parts. The first part is a pretrained VGG-Net. The second part is a convolutional layer. The third part is a spatial pyramid pooling layer. The final part is a softmax layer.

$$
\begin{aligned}
S_c = & \frac{1}{N_2} \sum_{x,y \in r2} \sum_k \left(\frac{w_{k1}^c}{4} + w_{k2}^c \right) f_k(x,y) \\
& + \frac{1}{N_2} \sum_{x,y \in r3} \sum_k \left(\frac{w_{k1}^c}{4} + w_{k3}^c \right) f_k(x,y) \\
& + \frac{1}{N_2} \sum_{x,y \in r4} \sum_k \left(\frac{w_{k1}^c}{4} + w_{k4}^c \right) f_k(x,y) \\
& + \frac{1}{N_2} \sum_{x,y \in r5} \sum_k \left(\frac{w_{k1}^c}{4} + w_{k5}^c \right) f_k(x,y)
\end{aligned} \tag{4}
$$

The probability of classifying the input image into category c is denoted as P_c in Eq.(5). Therefore, large S_c corresponds to high probability.

$$
P_c = \frac{\exp(S_c)}{\sum_c \exp(S_c)} \tag{5}
$$

The importance of unit (x, y) in region $r2, r3, r4$, and $r5$ for classifying the input image into category c is denoted as $I_{c2}(x,y)$, $I_{c3}(x,y)$, $I_{c4}(x,y)$, and $I_{c5}(x,y)$ which can be derived from Eq.(4). For example, $I_{c2}(x,y)$ in Eq.(6) is derived from the first term of Eq.(4).

$$
I_{c2}(x,y) = \sum_k \left(\frac{w_{k1}^c}{4} + w_{k2}^c \right) f_k(x,y) \tag{6}
$$

The object is the most important part for classifying the image. Therefore, $I_{c2}(x,y)$, $I_{c3}(x,y)$, $I_{c4}(x,y)$, and $I_{c5}(x,y)$ will have high values in the position of the object. We propose a spatial pyramid saliency map to represent the values of $I_{c2}(x,y)$, $I_{c3}(x,y)$, $I_{c4}(x,y)$, and $I_{c5}(x,y)$. Therefore, the saliency map can highlight the object.

Fig. 3 shows how to get the saliency map for category c. The activations of the units in region $r1$ of the kth feature map are multiplied by weight $\frac{w_{k1}^c}{4}$ to get M_{k1}. The activations of the units in the regions $r2, r3, r4$, and $r5$ of the kth feature map are multiplied by weight $w_{k2}^c, w_{k3}^c, w_{k4}^c$, and w_{k5}^c respectively. Then, these four parts are combined together to get M_{k2} which has the same size as M_{k1}. Then, M_{k1} and M_{k2} are added together to form the saliency map M_k for the kth feature map. Finally, the spatial pyramid saliency map M equals to $\sum_k M_k$.

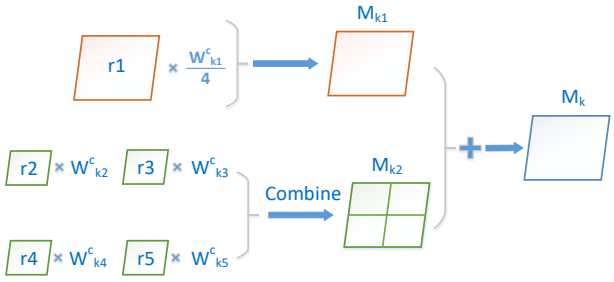


Fig. 3. Spatial pyramid saliency map. The units in the region $r1$ are multiplied by weight $\frac{w_{k1}^c}{4}$ to get M_{k1}. The units in the regions $r2, r3, r4$, and $r5$ are multiplied by their corresponding weights. Then, they are combined together to form M_{k2}. Then, M_{k1} and M_{k2} are added together to get M_k. Finally, the spatial pyramid saliency map M equals to $\sum_k M_k$.

The size of the spatial pyramid saliency map is 14×14. In order to localize the object, we use bilinear interpolation to upsample the saliency map into the size of the input image.

3. EXPERIMENT

3.1. Setup

In order to localize the object, first of all, we train our model to classify the input image. Momentum gradient descent optimization algorithm with mini-batch of 40 images is applied in the training process. The learning rate is initialized to 0.01 and decayed every epoch with a base of 0.99. In order to al-

leviate the overfitting problem, dropout [15] with the dropout rate of 0.5 is used. After the training process, the spatial pyramid saliency map is used to localize the object. Our code is written in Python using *Tensorflow* which is an open source deep learning framework developed by Google Brain Team [16].

3.2. Experimental result on Cub-200 dataset

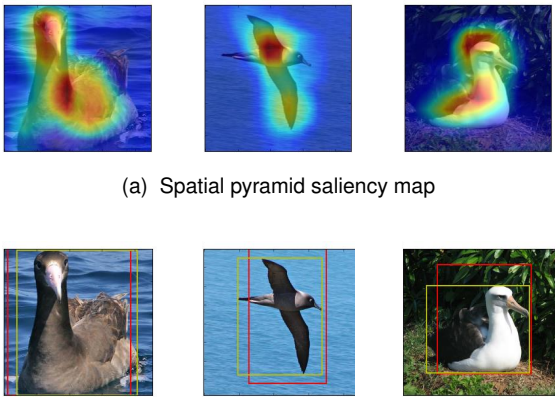
Cub-200 dataset [17] is used to evaluate the object localization performance of our method. The training set contains 5,994 images, and the test set contains 5,794 images. Each image has a ground truth bounding box to indicate the location of the object in the image. Therefore, we can quantitatively evaluate the localization performance of our method.

Some object localization results of our method are shown in Fig. 4. Fig. 4a shows the spatial pyramid saliency maps generated by our method. The object is highlighted and distinguished from the background. In Fig. 4b, the ground truth location of the object is indicated by a yellow rectangle. In order to compare the localization result of our method with the ground truth, we generate a bounding box to indicate the location of the object. The bounding box is generated by the technique proposed in [9] where a rectangle is generated to cover the region of the saliency map whose value is above 20% of the max value of the saliency map. In Fig. 4b, the generated bounding box, indicated by a red rectangle, matches the ground truth well.

Intersection over union (IOU) score metric [18] is used to quantitatively compare the localization performance of our method with Zhou's method [9] and Oquab's method [4]. The formulation of IOU score is shown in Eq.(7)

$$IOU = \frac{B_1 \cap B_2}{B_1 \cup B_2} \tag{7}$$

where B_1 and B_2 denote the area of the generated bounding box and ground truth respectively. The numerator of Eq.(7) denotes the intersection area between these two boxes while the denominator denotes the union area. IOU score ranges from 0 to 1, and larger IOU score corresponds to better object localization performance. For example, the IOU scores of Fig. 4b are 0.831, 0.732 and 0.704 respectively. With IOU score, we can set a threshold. When IOU score is larger than the threshold, the target object is successfully localized. We calculate the success rate over the whole test set when the threshold ranges from 0.4 to 0.9 with step 0.1. The corresponding success rate is shown in Fig. 5 where the red solid line shows the performance of our method. The performance of Zhou's method and Oquab's method are demonstrated by blue dashed line and black dot-dashed line respectively. These results are obtained by training and evaluating Zhou's model and Oquab's on Cub-200 dataset. Fig. 5 demonstrates that our method outperforms Zhou's method and Oquab's method.



(a) Spatial pyramid saliency map

(b) Localization result and ground truth

Fig. 4. (Best seen in color.) Object localization results of our method on Cub-200 dataset. The object is highlighted in (a). The ground truth location of the object is indicated by a yellow rectangle while the bounding box generated by our method is indicated by a red rectangle.

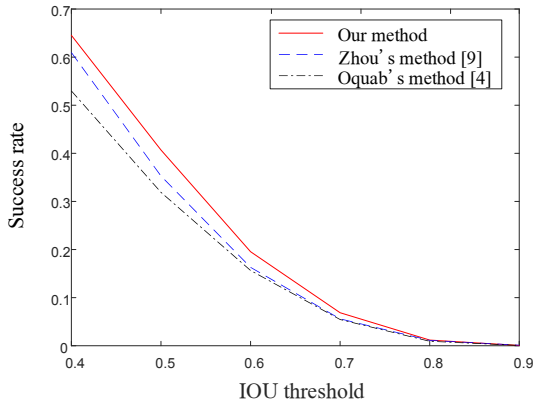


Fig. 5. Success rate under different IOU threshold. The red solid line shows the performance of our method. The performance of Zhou's method [9] and Oquab's method [4] are demonstrated by blue dashed line and black dot-dashed line respectively.

4. CONCLUSION AND FUTURE WORK

In this paper, we demonstrated that the feature map in CNN can be activated by some specific input patterns, and the activated region in the feature map corresponds to the location of the object in the input image. Besides, we proposed a deep convolutional neural network with spatial pyramid saliency map to localize object. The experimental results on Cub-200 dataset demonstrated that our method can localize the full extent of the object and outperformed the traditional methods. For the future work, we will implement our method into object tracking task.

5. REFERENCES

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

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

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

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

[5] L. Bazzani, A. Bergamo, D. Anguelov, and L. Torresani, "Self-taught object localization with deep networks," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2016, pp. 1–9.

[6] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014, pp. 1717–1724.

[7] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *2014 European Conference on Computer Vision (ECCV)*. Springer, Sept 2014, pp. 818–833.

[8] Zhou Bolei, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, "Object detectors emerge in deep scene cnns," in *2015 International Conference on Learning Representations (ICLR)*, May 2015.

[9] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2921–2929.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, Sept 2015.

[11] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," in *2005 IEEE International Conference on Computer Vision (ICCV)*, Oct 2005, pp. 1458–1465.

[12] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2006, pp. 2169–2178.

[13] J. Sivic and A. Zisserman, "Video google: a text retrieval approach to object matching in videos," in *2003 IEEE International Conference on Computer Vision (ICCV)*, Oct 2003, pp. 1470–1477.

[14] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[16] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[17] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[18] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.