

LOCALIZING OBJECT POSITION BY USING ONLY
IMAGE-LEVEL LABELS

ZHENFEI ZHANG

A THESIS
IN
THE DEPARTMENT
OF
COMPUTER SCIENCE AND SOFTWARE ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF APPLIED SCIENCE IN SOFTWARE
ENGINEERING

CONCORDIA UNIVERSITY
MONTRÉAL, QUÉBEC, CANADA

MARCH, 2021

© ZHENFEI ZHANG, 2021

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Zhenfei Zhang**

Entitled: **Localizing object Position by Using only Image-level Labels**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science in Software Engineering

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair

Dr. Sudhir Mudur

_____ Examiner

Dr. Adam Krzyzak

_____ Supervisor

Dr. Tien Dai Bui

Approved _____

Dr. Hovhannes Harutyunyan

Chair of Department or Graduate Program Director

_____ 2021 _____

Dr. Amir Asif, Dean

Faculty of Engineering and Computer Science

Abstract

Localizing object Position by Using only Image-level Labels

Zhenfei Zhang

Weakly Supervised Object Localization (WSOL) task attracts more and more attention in recent years, which aims to locate the object by using incomplete labels. Considering the cost of annotation, especially ground-truth bounding box label and training speed of detection task, it is very necessary to improve the performance of WSOL that only requires image-level labels. Most current methods tend to utilize Class Activation Map (CAM) that can only highlight the most discriminative parts rather than the entire target. The common method to address this kind of limitation is to hide the most obvious regions and let the model learn other parts of the target. The main work of this thesis is to eliminate the limitations of current WSOL work and improve the performance of localization. In chapter 3, we design an attention-based selection strategy to dynamically hide the feature maps. In chapter 4, a new hiding method is proposed to further improve the localization performance. In chapter 5, we propose three method to eliminate the issues on CAM level. Our methods are evaluated on CUB-200-2011 and ILSVRC 2016 datasets. Experiments demonstrate that the proposed methods work very well and significantly improve the localization performance.

Acknowledgments

I would like to thank my supervisor Dr. Tien D.Bui firstly who is patient and extremely supportive to my master researching at Concordia University.

I would also like to thank my parents for their support, encouragement and love. Without their help, I can not finish my master study in Canada.

I should also express my sincere thanks to the instructors of courses I took during my master study. Those courses establish a strong theoretical basis for my subsequent research.

Finally, I would like to thank my girlfriend Wendy Song, for her company and encouragement. In this special time, her company is even more precious.

Related Publications

The following publications are related to this thesis:

- **Zhenfei Zhang** and Tien D.Bui. Attention-based Selection Strategy for Weakly Supervised Object Localization. Accepted to International Conference on Pattern Recognition. ICPR 2020.
- **Zhenfei Zhang** and Tien D.Bui. Attention-based Dual Hiding Method for Weakly Supervised Object Localization. Submitted to International Conference on Image Processing. ICIP 2021.
- **Zhenfei Zhang**. Revisiting Class Activation Map: Two Stage Method for Weakly Supervised Object Localization. Preparing to NeurIPS 2021.

Contents

List of Figures	ix
List of Tables	xi
1 Introduction	1
1.1 Motivation	1
1.2 Weakly Supervised Object Localization	2
1.3 Thesis Outline	3
1.4 Contribution of Authors	5
2 Literature Review	6
2.1 Weakly Supervised Object localization	6
2.1.1 Class Activation Map	7
2.1.2 Current Methods	8
2.2 Convolutional Neural Network	9
2.2.1 Convolution operation	9
2.2.2 CNN VS ANN	11
2.2.3 The Typical Layer of CNN	11
2.2.4 Loss Function	15
2.2.5 Regularization	16
2.3 Attention Mechanism	17

2.4	Pre-trained Model	19
2.5	Datasets	20
2.6	Evaluation Metrics	21
2.7	Overview of Proposed method	21
3	Attention-based Selection Strategy	24
3.1	Introduction	25
3.2	Proposed Method	27
3.3	Experiment Results	34
3.3.1	Implementation Details	34
3.3.2	Ablation Study	35
3.3.3	Comparison with State-of-the-art Methods	36
3.4	Conclusion	38
4	Attention-based Dual Hiding Method	39
4.1	Introduction	39
4.2	Proposed method	41
4.3	Experiment Results	43
4.3.1	Implementation Details	43
4.3.2	Comparison with State-of-the-art Methods	43
4.3.3	Ablation Study	44
4.3.4	Results	46
4.4	Conclusion	47
5	Revisiting Class Activation Map	48
5.1	Introduction	48
5.2	Revisiting Class Activation Map	50
5.3	The issues of CAM	51
5.4	Proposed methods	53

5.4.1	Weighted Global Average Pooling	54
5.4.2	Recombining the weights of FC layer	54
5.4.3	Two stage localization	56
5.5	Experiment Results	58
5.5.1	Implementation Details	58
5.5.2	Comparison with State-of-the-art Methods	59
5.6	Ablation Study	59
5.7	Conclusion	60
6	Summary and Future Work	62
6.1	Conclusion	62
6.2	Future work	63

List of Figures

1	The diagram of Class Activation Map. Image is from [8].	7
2	The diagram of 2-D convolution operation. Red squares show one convolution operation in this case.	10
3	The diagram of a typical layer of CNN.	12
4	The diagram of three typical activation functions.	14
5	The diagram of Max-Pooling.	14
6	The diagram of Global Average Pooling.	15
7	Networks before and after using Dropout. Image is from [47]	17
8	The idea of DropBlock. Image is from [50].	18
9	The residual block. Image is from [7].	19
10	Intersection over Union	20
11	Overall architecture of our training model.	22
12	VGGnet configuration. Image is from [19].	23
13	Localization results from CAM [8] and one of the state-of-the-art methods.	25
14	The diagram of our method. Note that this is the one situation of our selection strategy, for easy understanding, other two will be shown in Figure 15 and Figure 16.	28
15	The diagram of our method when the feature map has much non-targeted information.	28

16	The diagram of our method when the input image is relatively small.	29
17	The self-attention map and different thresholding masks.	31
18	Comparison with CAM [8] on CUB-200-2011 (left) and ILSVRC 2016 (right) datasets. Green bounding box is prediction and the red bound- ing box is ground truth.	36
19	The diagram of our hiding method.	41
20	The diagram of area hiding method.	42
21	Comparison with CAM [8] on CUB-200-2011 (left) and ILSVRC 2016 (right) datasets. Green bounding box is prediction and the red bound- ing box is ground truth.	45
22	The figure shows the issue of CAM that can only highlight the most discriminative region of the target.	49
23	The three issues that make the CAM ill-posed. The figure is extracted from [8], we only labeled three numbers to show the problems of CAM.	51
24	The examples of positive feature maps and negative feature maps. . .	52
25	The examples show the issue of thresholding bounding box.	53
26	The diagram of proposed WGAP.	55
27	The diagram of computation of CAM. The first row is the case that the negative channels are inhibited. The second row shows the results without inhibiting the negative feature maps.	56
28	The diagram of proposed two stage localization method.	57
29	The overall procedure of proposed method.	58
30	Comparison with CAM on CUB-200-2011 dataset.	60
31	The case of selecting the optimal threshold to extract bounding box.	61

List of Tables

1	The different Results for the choice of positions to plug our module . . .	35
2	The results according to different methods	36
3	Quantitative evaluation performance on CUB-200-2011 dataset . . .	37
4	Quantitative evaluation performance on ILSVRC 2016 dataset	37
5	Quantitative evaluation performance on CUB-200-2011 dataset. . . .	43
6	Quantitative evaluation performance on ILSVRC 2016 dataset. . . .	44
7	The Different Results For The Choice of positions to plug our module.	46
8	The Different Results by using different drop masks.	46
9	Ground-truth accuracy of each method. Since few methods provided the value of this evaluation metric, we only show available methods in this table.	46
10	Quantitative evaluation performance on CUB-200-2011 dataset. . . .	59

Chapter 1

Introduction

In this chapter, the brief introduction of our work will be given. Firstly, I will introduce the Weakly Supervised Object Localization (WSOL) task and the motivation of our work. Secondly, the outline of this thesis will be mentioned. Finally, I will give the contribution of this thesis.

1.1 Motivation

In recent years, fully supervised object detection [20] [21] [22] [23] and image segmentation [24] [25] [26] [27] have already achieved satisfactory results. However, it requires both classification and ground-truth bounding box labels that are much expensive and time-consuming. Especially for the bounding box annotations, which can show the location of the objects in training data. For each training object location, the label is supposed to have 4 values that are the coordinates of the upper left and lower right corners, the width and height of the box. Therefore, ground-truth bounding box labels are much more complex than the classification labels and they are very expensive to label by human hands. In order to save the costs and meet big data and model speed requirements of further research, it is necessary and meaningful to do weakly supervised [28] or unsupervised learning [29] in detection tasks.

1.2 Weakly Supervised Object Localization

Weakly Supervised Object Localization (WSOL) task aims to recognize the object position by using only image-level labels. Compared with Fully Supervised Learning, WSOL uses incomplete labels to train the model, which is less time-consuming and less redundant. Therefore, WSOL has attracted more and more attention.

By using only image-level labels, the model is trained to do the classification. Class Activation Map (CAM) [8] has been widely used in WSOL techniques. CAM [8] is generated from convolutional neural networks (CNNs) by using Global Average Pooling [5], which covers the most discriminative regions of the target object. Basically, the Class Activation Map [8] is a heatmap that expresses the value of each pixel from classification results. For the targeted pixels, the weights are much larger than others.

In computer vision area, there are four similar tasks. Those are Object Detection [20] [21] [22] [23] [1], Object Localization [2] [30], Image Segmentation [24] [25] [26] [27] and Image Classification [31] [32] [33]. There is something different with these four tasks.

Image Classification For the input data of classification tasks, normally there is only one main target on the image. The aim of the classifier is to predict the main object of the input image. The training labels are the classification labels that can tell the model what the main object of input is .

Object Localization The input data is the same as the classification task. The goal of localization is to tell us where the main body of this picture is. Instead of only image-level labels, ground-truth bounding box annotations are required for this task. Generally it uses a rectangle to express the location of the main target.

Object Detection This task is more complex than the first two, which lets the model not only predict the category but also show the position of the object. Moreover, the model needs to do the classification and localization for multiple objects

in one image. This mission is challenging because the multiple objects may belong to different categories.

Image Segmentation Image Segmentation is similar to Object Detection to some extent. For the output of an object detection task, it is supposed to be a rectangle that can show the targeted location. But for the output of image segmentation, it should be a mask that can express the boundary of the target.

1.3 Thesis Outline

In this section, the outline of this thesis is given.

Chapter 2 mentions the related work that is utilized in this thesis. It includes Convolutional Neural Network(CNN), Attention Mechanism and Class Activation map. Meanwhile, the datasets and evaluation matrix are described in this chapter as well.

Chapter 3 designs an attention-based selection strategy to detect the location of objects effectively and flexibly. Current techniques have already illustrated that hiding methods can effectively address the limitation of Class Activation Map [8]. However, almost all the hiding methods only try to remove the most discriminative part blindly that will make the model learn more unhelpful information such as background. Our approach can eliminate the problems of current hiding methods, which can remove much unhelpful information that will mislead the localization. Moreover, a selection strategy is proposed to dynamically remove the part from different kinds of images. Compared with current state-of-the-art techniques, the proposed method works very well on localization accuracy without too much overhead on CUB-200-2011 and ILSVRC 2016 datasets. In our perception, our method is the first work to consider the different situations of training images. Therefore, our work provides new insights to do the Weakly Supervised Object Localization task. The related paper:

- **Zhenfei Zhang** and Tien D.Bui. Attention-based Selection Strategy for Weakly Supervised Object Localization. Accepted to International Conference on Pattern Recognition. ICPR 2020.

Chapter 4 provides a dual hiding method for Weakly Supervised Object Localization (WSOL) task. Our starting point is based on the strong relationship of each pixel of convolutional feature map. We design a hiding method that can remove the most discriminative part instead of pixels for both channel and spatial space. According to the experiments, we can see that the proposed method can effectively improve the localization performance on CUB-200-2011 and ILSVRC 2016 datasets compared with current WSOL techniques. The related paper:

- **Zhenfei Zhang** and Tien D.Bui. Attention-based Dual Hiding Method for Weakly Supervised Object Localization. Submitted to International Conference on Image Processing. ICIP 2021.

Chapter 5 is based on the three issues that make CAM ill-posed. Based on the issues, we proposed three corresponding methods to eliminate them. Our proposed two stage localization method does not introduce any hyperparameter that needs to be set by experience. In our perception, our method is the first method that can evaluate the bounding box without Ground-truth label and assign the optimal threshold to different images. The improvement is very significant. The related paper:

- **Zhenfei Zhang**. Revisiting Class Activation Map: Two Stage Method for Weakly Supervised Object Localization. Preparing to NeurIPS 2021.

Chapter 6 summaries the thesis with the main contribution of our work. Moreover, it provides the limitations and potential ideas of future work.

1.4 Contribution of Authors

This section shows the contribution of each authors.

Attention-based Selection Strategy for Weakly Supervised Object Localization [18]

Zhenfei Zhang: Model design, implementing, model training, doing experiments, results analysis, paper writing, editing and proofing.

Tien D.Bui: Research supervisor, advising, editing and proofing.

Attention-based Dual Hiding Method for Weakly Supervised Object Localization [62]

Zhenfei Zhang: Model design, implementing, model training, doing experiments, results analysis, paper writing, editing and proofing.

Tien D.Bui: Research supervisor, advising, editing and proofing.

Revisiting Class Activation Map: Two Stage Method for Weakly Supervised Object Localization [66]

Zhenfei Zhang: Model design, implementing, model training, doing experiments, results analysis, paper writing, editing and proofing.

Chapter 2

Literature Review

In this section, I will give the background that is utilized in this thesis. First of all, a brief description of some effective methods of weakly supervised object localization is given, including the Class Activation map [8]. After that, we provide an introduction of Convolutional Neural Network, Attention Mechanism and pre-trained model used. Finally, the datasets and evaluation matrices of this work are provided.

2.1 Weakly Supervised Object localization

Weakly Supervised Object Localization (WOSL) task aims to recognize the object by using only image-level labels. Class Activation Map (CAM) [8] has been widely utilized in Weakly Supervised Object Localization task, which uses Global Average Pooling [5] and Softmax [8] activation function to generate localization maps. For this section, we will introduce Class Activation Map [8] first, which is the core technique to do the Weakly Supervised Object Localization.

2.1.1 Class Activation Map

Convolutional Neural Network (CNN) has been widely utilized in image processing [34] and achieved significant improvement. However, CNN is like a 'black box' that lacks of interpretability. In order to know what happened in each layer of CNN, researchers not only looked for the explanations based on the theory, but also some visualization methods to understand the CNN directly. Class Activation Map [8] is a kind of visualization method, which can show the regions to instruct CNN to do the correct recognition.

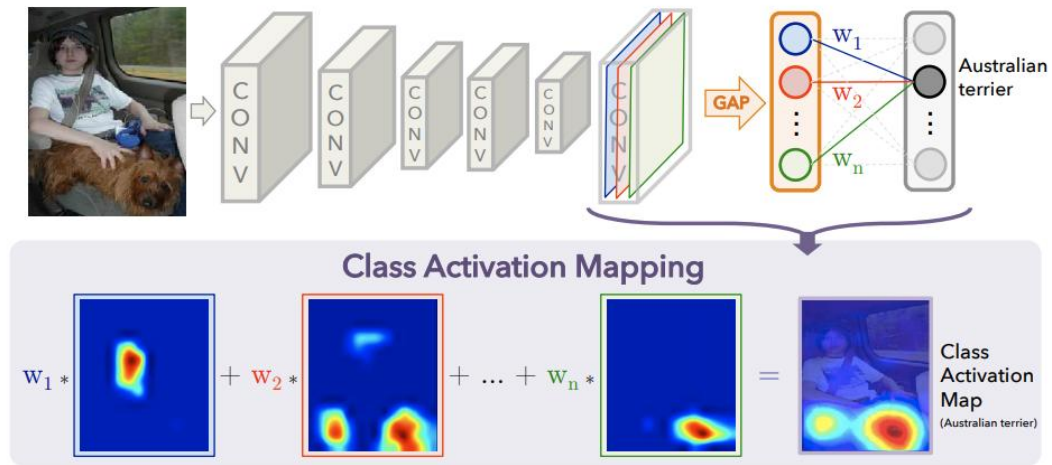


Figure 1: The diagram of Class Activation Map. Image is from [8].

Compared with general CNN, Figure 1 replaces the fully connected layer with a Global Average Pooling [5] Layer. Therefore, we obtain a weight for each channel. Finally, the method performs a weighted summation to output the final result. The output is a heatmap that expresses the most relevant part of the final classification result based on the image-level label. From the visualization of CAM [8], we notice that the classification model only uses the most discriminative area to do the classification. Thus making the detector inaccurate, which can only recognize the most important regions. How to locate the entire target in this task is the main challenge of WSOL.

Apart from CAM [8], some improved methods such as Grad-CAM [35], Grad-CAM++ [36] and smooth Grad-CAM++ [37] that can output the weight of each channel without changing the architecture of the pre-trained network. These methods are similar since they all obtain the weights from gradients of backpropagation. However, the gradients may have noise and saturation problems that may affect the final output. In 2020, Score-CAM [38] was proposed to obtain the weights without using any gradients. We will not go into details of these methods since almost all the WSOL techniques utilize CAM [8] as a visualization method.

2.1.2 Current Methods

As we mentioned in the last subsection, the main issue of CAM is that it can only highlight the most discriminative region. If the target is the whole body of a human, using CAM may only locate the head. That will affect the localization performance. In order to eliminate this issue, various methods have been proposed to capture the entire pattern. Hide-and-see (HaS) [14] is a kind of augmentation method that aims to hide patches of input images randomly. By removing the grid patches of data, the model can focus on other targeted regions instead of only focusing on the most discriminative part. Adversarial Complementary Learning (ACoL) [12] proposes a new architecture to remove the most activated regions adversarially. In 2018, Self-Produced Guidance (SPG) [11] was introduced as an architectural solution that generates three masks (foreground, unsure, background) using three different layers in Inception-V3 [17]. Attention-based Dropout Layer (ADL) [9] has proposed a new attention module to remove the most important region with none additional overheads. The core of these methods is to hide the most discriminative part and make the model learn other parts of the target.

Apart from hiding the most discriminative regions of the images, there are some other methods such as CutMix [13], NL-CCAM [39] and DANet [10]. CutMix [13] is a

kind of data augmentation that the input images are cut and mixed with other images. DANet [10] proposed two new loss functions that are hierarchical divergent activation and discrepant divergent activation to learn entire patterns. For NL-CCAM [39], it proposes a polynomial function to combine several class activation maps and utilize non-local blocks to improve the relationship of targeted pixels.

2.2 Convolutional Neural Network

Deep learning [40] has been widely utilized in computer vision and natural language processing. In 1989, LeCun introduced a convolutional network that is inspired by brain cells. CNN is a kind of network to specially deal with grid structure data such as image data, which performs very well in many domains. So in this section, we will provide the details of CNN.

2.2.1 Convolution operation

Different from Artificial Neural Network (ANN), at least one layer of Convolutional Neural Network uses convolution operation. In mathematics, convolution is a linear operation on two real variable functions. There are two parameters of convolution that are input and kernel function. Two-dimensional convolution is used for image processing. The equation is:

$$V(i, j) = (I * K)(i, j) = \sum_h \sum_w I(h, w) K(i - h, j - w) \quad (1)$$

Where input I is an image and K is a convolutional kernel. w and h are the width and height of the input image. Different from traditional feature extraction methods [42] [43], CNN utilizes sliding convolution kernels to extract the feature. With different kernels, the features are different as well. Convolution operation satisfies the commutative law of equation. However, it is not an important property

in deep learning. So in some deep learning environments, they replace convolution with a cross-correlation that is almost the same as convolution operation. The only difference is that convolution operation flips the kernel. The cross-correlation is:

$$V(i, j) = (I * K)(i, j) = \sum_h \sum_w I(h + i, w + j) K(h, w) \quad (2)$$

In 2-D convolution operation, for each position of the kernel, the values in corresponding position are multiplied. Then the products followed by the summation to obtain the final value. For each kernel position, the output of convolution is a value in the corresponding position. The diagram is shown in Figure 2.

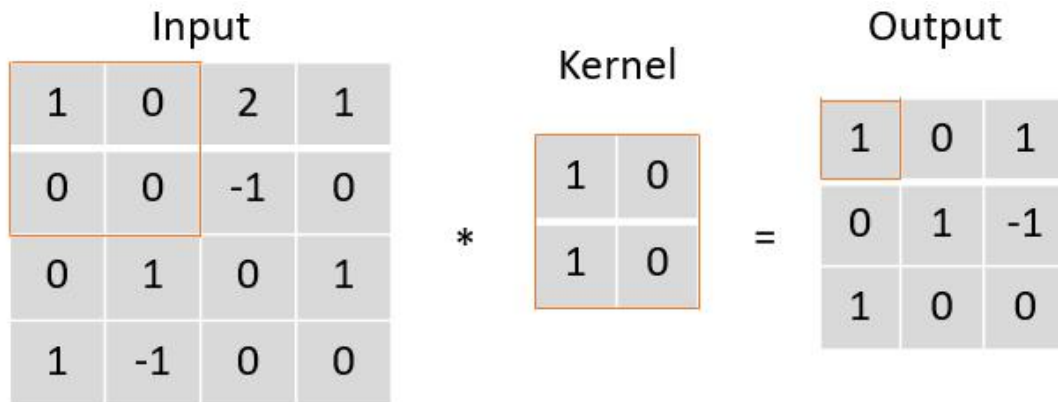


Figure 2: The diagram of 2-D convolution operation. Red squares show one convolution operation in this case.

As we can see in Figure 2. 2-D convolution operation is implemented by sliding convolution kernels on the input. After doing the operation in Figure 2, the size of output is changed. To meet the needs of different tasks, padding is introduced to the 2-D convolution operation, which adds the values around the input matrix. Based on different padding methods, convolution can be divided into three categories that are valid convolution without any padding, same convolution with half or same padding and full convolution with full padding. Valid convolution is shown in Figure 2. For the same padding, it adds some values around the input matrix to make the output

size equal to the input. As for the full convolution, the output size is largest. The function to calculate output size is:

$$o = \left\lceil \frac{i + 2p - k}{s} \right\rceil + 1 \quad (3)$$

Where o and i are output and input dimension. p is padding and k is filter size. s is the strides that indicates the kernel sliding distance.

2.2.2 CNN VS ANN

Compared with the Artificial Neural Network (ANN), there are two important thoughts in CNN, which are sparse interactions and parameter sharing.

In the Artificial Neural Network (ANN), each input neuron is connected with all the neurons in the next layer. However, there is only a local connection in CNN. Instead of dealing the whole image, CNN utilizes convolutional kernels that are much smaller than the image size. By sliding the kernels, the model can scan the whole image. Sparse interactions can reduce the calculation amount and complexity. Parameter sharing means that it utilizes the same filter to generate the feature map. Although parameter sharing can not reduce the operation time, it significantly lowers the number of parameters.

Sparse interactions and parameter sharing are the motivation of researchers to propose CNN. These two important thoughts provide possibilities for deeper network structures.

2.2.3 The Typical Layer of CNN

Normally, a typical layer of CNN consists of three parts, those are convolutional layer, activation function layer and pooling layer. The diagram of a typical layer is shown in Figure 3.

Convolutional layer

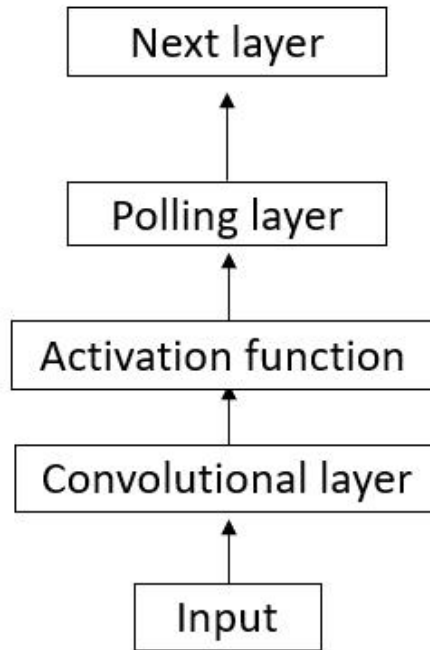


Figure 3: The diagram of a typical layer of CNN.

Convolutional layer is used to extract features from inputs. Different from traditional feature extraction methods [44] in Machine Learning, convolutional layers can obtain the features automatically instead of using hand-crafted features. Thus saving a lot of time and costs. There is a receptive field in each channel, which can locally connect the input. The larger the receptive field, the more global and higher semantic features will be detected. On the contrary, small filters tend to recognize local and detailed features.

Activation function

The goal of neural networks is to address the problems such as classification or regression by predicting the results from input values. Based on the different problems, different functions are needed. There are two types of functions, linear and nonlinear activation functions. For example, for the Machine Learning method such as k-means [44] , it utilizes a linear activation function that is a line to do the classification.

However, if we use a linear activation function in every hiding layer, the hiding layers do not work in this case. Thus, the network is still a single-layer perceptron because of only performing linear transformation. In the multi-layer neural network, only non-linear activation function is effective.

Common nonlinear activation functions are Sigmoid, tanh, ReLU [45].

For the Sigmoid, the function is:

$$y = \textit{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (4)$$

Sigmoid is normally used in two classification problems [46]. From the function image we can see that there are almost no gradients when the input values are somewhat small or large. So in this case, the weights can not update normally. Moreover, Sigmoid is not symmetrical with 0, which may make the model hard to converge. In order to address the second issue, tanh activation can be used. The function is:

$$y = \textit{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (5)$$

Tanh can effectively solve the second issue of Sigmoid activation function. However, the gradient vanishing problem still exists. The most common activation function in computer vision tasks is ReLU [45]. The function is:

$$y = \textit{ReLU}(x) = \max(0, x) \quad (6)$$

ReLU activation function can effectively address the issues of Sigmoid and tanh. ReLU has better performance and faster convergence speed with no complex index calculation. In this work, the pre-trained CNN utilizes ReLU as the activation function. The plots of these functions are shown in Figure 4.

For the output layer, when the prediction has multiple categories, Softmax is usually used. By using this function, we can obtain the normalized probability of each class. The function is:

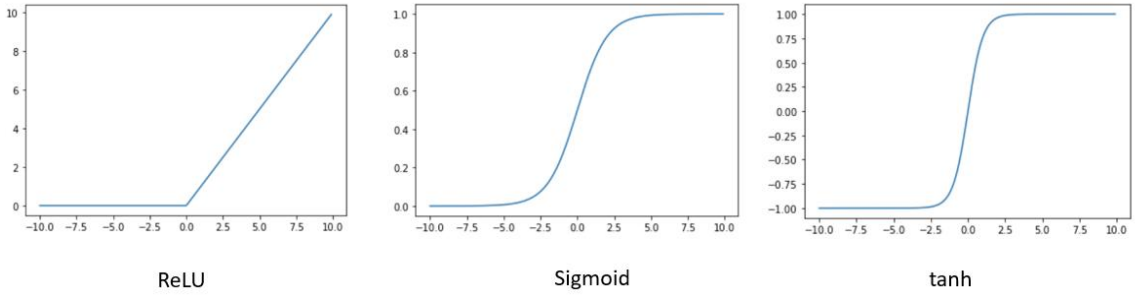


Figure 4: The diagram of three typical activation functions.

$$s_i = e^{a_i} / \sum_{k=1}^K a_k \quad (7)$$

In our second work, spatial-softmax [6] is used to find the expected pixels of feature maps and calculate the classification weights of each channel.

Pooling layer

Pooling is a kind of method to reduce the training parameters and maximize the spatial information. Normally, we do the down-sampling to decrease the dimension of feature maps. Pooling function is used to replace the value of point with adjacent value overall characteristics, which can make the output invariant. Different from

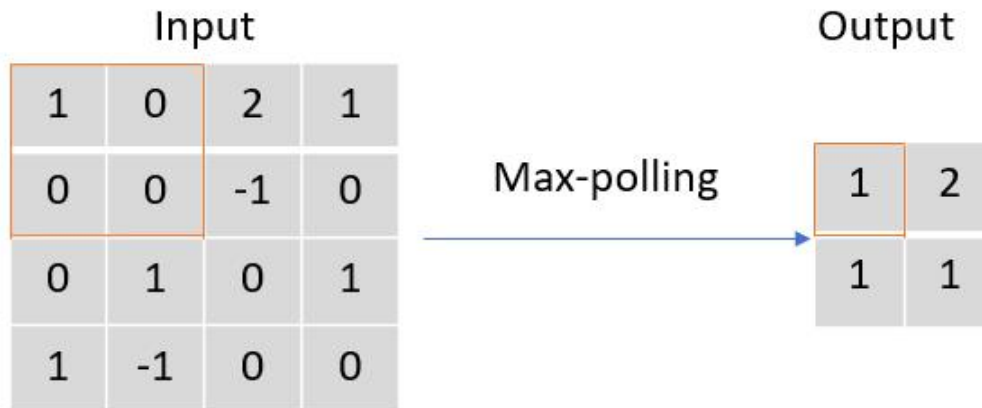


Figure 5: The diagram of Max-Pooling.

convolutional layers, the filter of pooling is parameter-free. Max-pooling and Average-pooling have been widely used in some deep CNN. The diagram of Max-pooling is shown in Figure 5. The filter and strides size are both 2×2 . For the red box area, the maximum value replaces the red box region by using Max-pooling function.

In our WSOL task, we not only use Max-pooling but also Global Average Pooling [5] that uses global average value to replace the whole feature map. Figure 6 expresses the GAP [5]. Nowadays, Global Average Pooling [5] (GAP) has been a layer in many deep learning packages. The most common application of GAP [5] is to replace the fully connected layer in CNN. After doing that, the CNN becomes a Fully Convolutional Neural Network with good generalization ability. GAP [5] is actually a special case of Average-pooling.

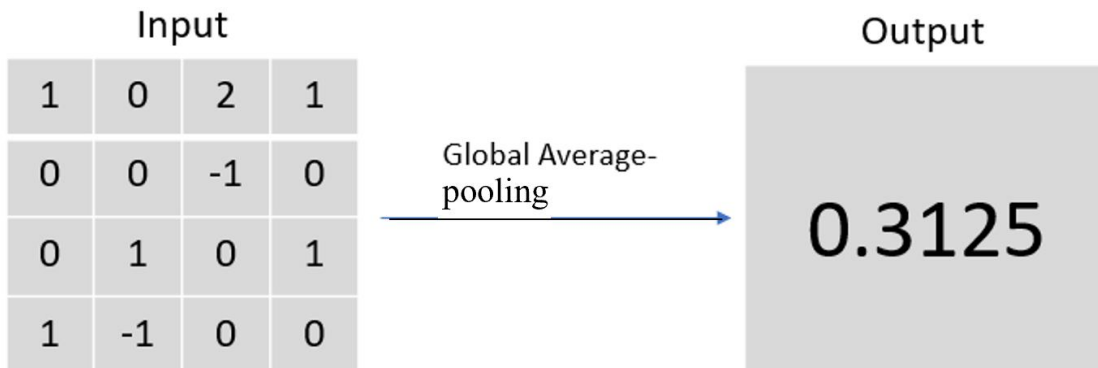


Figure 6: The diagram of Global Average Pooling.

2.2.4 Loss Function

The loss function is to do the evaluation of the neural network, which expresses the distance between prediction and ground-truth labels. The goal of networks is to update parameters and minimize the value of loss function. The common loss functions are Cross-Entropy and Mean Squared Error. The functions are:

$$CE = -\frac{1}{m} \sum_{i=1}^m \left(Y_i \log \hat{Y}_i + (1 - Y_i) \log (1 - \hat{Y}_i) \right) \quad (8)$$

$$MSE = \frac{1}{m} \sum_{i=1}^m \left(Y_i - \hat{Y}_i \right)^2 \quad (9)$$

Where Y_i and \hat{Y}_i are ground-truth labels and predictions. In our WSOL task, we use MSE as a loss function. Because only image-level labels are used, we only use classification loss to update the weights of the network.

2.2.5 Regularization

The core challenge of Machine learning is to design a method that can perform well on both training data and new input test data. Regularization is a kind of way to improve generalization ability. Some common methods such as Dropout [47], Early Stopping, Data Augmentation and Batch Normalization. Early Stopping is used to stop training when continued training can only obtain little or no improvement. The idea of Data Augmentation is to increase the size of training set by doing transformation. CutMix [13] is a type of Data Augmentation in WSOL task. Batch Normalization can adjust the weights of each layer, thus each layer is trained from a similar starting point. We utilize the idea of Dropout [47] in our work, so we will describe in much detail.

Dropout [47] is an effective method to eliminate over-fitting by dropout the nodes of a network in some probability randomly. For the dropped neurons, they will not work during training. The diagram is shown in Figure 7.

Dropout provides the probability for a deeper network. However, it is less effective on convolutional feature maps. There are two reasons for this limitation: 1) Convolutional layers have much less parameters than fully connected layers, so there is less probability to overfit, 2) Dropout can not abandon all the information because there is a strong relationship between the pixels of a convolutional feature map. In

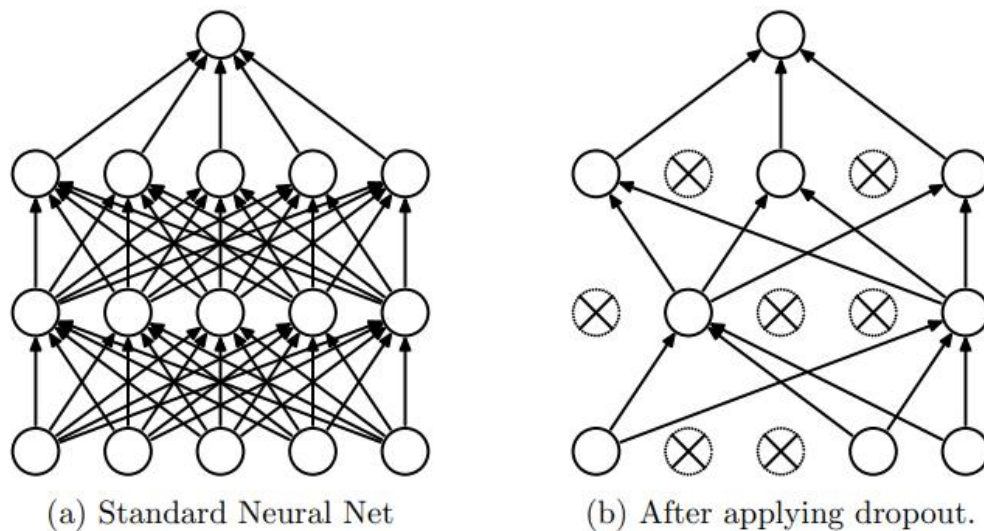


Figure 7: Networks before and after using Dropout. Image is from [47]

order to perform Dropout effectively, SpatialDropout [48] and MaxDrop [49] have been proposed. For our second work in Chapter 4, DropBlock [50] is performed to cut the relationship of pixels in a convolutional feature map. Figure 8 expresses the idea of DropBlock [50].

We can see in Figure 8, dropblock [50] not only drop the points but also surrounding points. The number of dropped surrounding points depends on the value of `block_size`.

2.3 Attention Mechanism

Attention mechanism is based on the habit of human observations. Humans always select important regions to identify the object instead of observing blindly. So for attention mechanisms, the most informative parts of input data are given more weight than others. By using attention, the training model can be much more focused and reduce the interference of irrelevant information so as to improve the performance. Attention is firstly used on natural language processing [51] to calculate the relationship

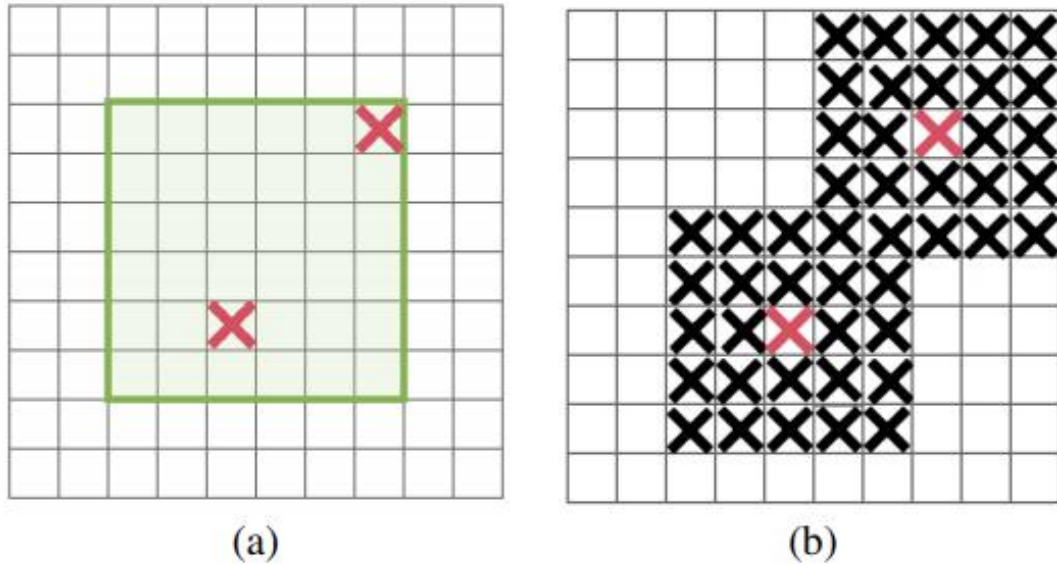


Figure 8: The idea of DropBlock. Image is from [50].

between the source sentence and target description. In the field of image processing and detection[52][53], it also shows excellent benefit. In this decade, attention-based image and text conversion has attracted much attention, such as image captioning[54] and text to image generation[55].

Self-attention has been widely used in both NLP[56] and computer vision[57]. Compared with traditional attention that only calculates the attention weights between source and target, self-attention applies to both source and target respectively and adds the attention weights of source to the attention of target. For example, various work has already used self-attention to improve the performance of classification networks. Squeeze-and-excitation block(SE Block)[58] is proposed to use only 1D channel self-attention map to enhance the classifier. In 2018, Convolutional Block Attention Module[59] utilizes both 1D and 2D self-attention maps to improve the classification model.

2.4 Pre-trained Model

In this work, we use VGG16 [19] and ResNet50 [7] as the backbone networks. VGGnet [19] demonstrates the depth of the network can impact the final performance of the network to some extent. It replaces 7×7 convolutional kernels with 3 3×3 kernels to increase the depth thus improving the performance. The configuration of VGGnet [19] is shown in Figure 12. For our method of WSOL, it is performed at the end of the block of VGGnet [19]. That will be discussed in the following chapters.

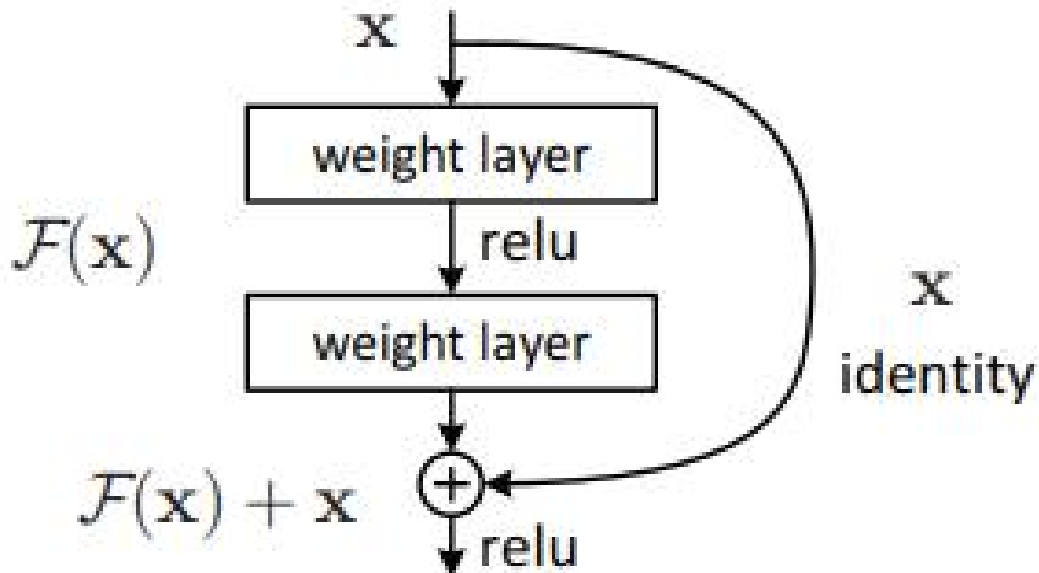


Figure 9: The residual block. Image is from [7].

Deep neural networks are very necessary since as the number of network layers increases, the extracted feature from different levels is richer with much more semantic information. However, only increasing the number of layers will cause gradient dispersion or gradient explosion. Although Batch Normalization can eliminate this limitation to some extent, another problem arises. As the number of network layers increases, the training accuracy decreases. This issue illustrates that deep neural networks are somewhat tough to optimize. In order to solve this issue, ResNet [7] is proposed. The main breakthrough of ResNet [7] is the shortcut connection [7] that

is shown in Figure 9. In the Figure, x is the input. Instead of learning the original function, ResNet [7] optimizes the function that is $F(x) + x$. Learning $F(x)$ to be 0 is much easier than learning $F(x)$ to be identity. In this case, we assume that input and output have the same dimensions.

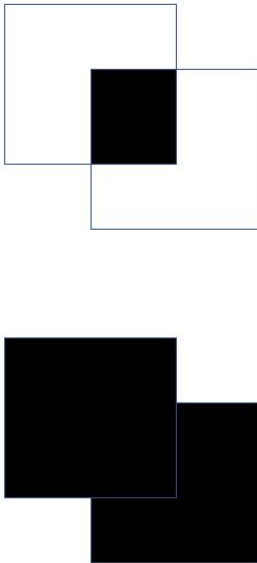
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Figure 10: Intersection over Union

2.5 Datasets

Our method is evaluated on CUB-200-2011 [15] and ILSVRC 2016 [16] datasets. ILSVRC 2016 has about 1.3 million data for training and 50,000 validation images, which consists of 1,000 categories. We use all the training data and image-level labels for training and validation images for testing. For CUB-200-2011, there are 5,994 training data and 5,794 validation images, which consists of 200 different bird classes. We train our model using training images and evaluate it by using validation data. Although CUB-200-2011 has much less data than ILSVRC 2016, it is a challenging dataset since there are fewer comparisons between each category.

2.6 Evaluation Metrics

The same as most current WSOL work, we use Top-1 Localization accuracy and Top-1 Classification accuracy to evaluate the performance of our hiding method. For localization accuracy, the prediction is correct when both classification and location results are equal to the ground-truth. The same as most work, we use intersection over union (IoU) to identify whether the predicted bounding boxes are correct. IoU has been widely used in detection tasks that are shown in Figure 10 .When the value of IoU is greater than 50%, the model will count localization as correct. For the classification accuracy, the result is correct only when classification prediction is equal to the label.

2.7 Overview of Proposed method

In the previous subsections in this chapter, we described the related work used in our proposed method. Our technique is to design a module that can be inserted into the classification model to eliminate the limitations of current Weakly Supervised Object Localization. The diagram is shown in Figure 11. The typical layer is similar to Figure 3. The inputs of our method are feature maps from the pooling layer. All the positions after the typical layer are potential places to perform the proposed module. As we can see in the Figure 12, there are 5 potential positions (after each Max-pooling layer) to apply our new module. How many times to perform a proposed method can lead to best performance, we will discuss in the next two chapters. For the output layer, we replace the fully connected layer with global average pooling [5] layer and softmax activation function to obtain a weight vector. Heatmap of each channel is generated by multiplying the weight vector with the output feature map thus each feature map has a weight based on the training label. Finally, linear addition is performed to generate the Class Activation Map [8] that can highlight the

most discriminative region of the object.

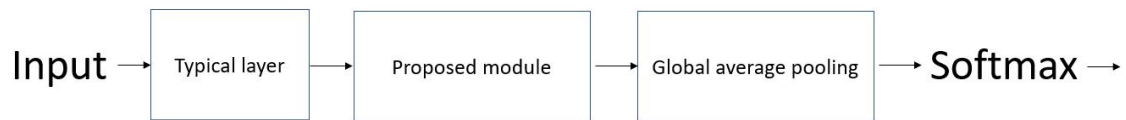


Figure 11: Overall architecture of our training model.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 12: VGGnet configuration. Image is from [19].

Chapter 3

Attention-based Selection Strategy

Weakly Supervised Object Localization (WSOL) task aims to recognize the object position by using only image-level labels. Some previous techniques remove the most discriminative parts for all input images or random images to capture the entire object location. However, these methods can not perform the correct operation on different images such as hiding the data or feature maps that should not be hidden. In this case, both classification and localization accuracy will be affected. Meanwhile, just erasing the most important regions tends to make the model learn the less discriminative parts from outside of the objects. To address these limitations, we propose an Attention-based Selection Strategy (ASS) method to choose images that do need to be erased. Moreover, we use different threshold self-attention maps to reduce the impact of unhelpful information in one of the branches of our selection strategy. Based on our experiments, the proposed method is simple but effective to improve the performance of WSOL. In particular, ASS achieves new state-of-the-art accuracy on CUB-200-2011 dataset and works very well on ILSVRC 2016 dataset.

3.1 Introduction

Class Activation Map (CAM) [8] has been widely used in WSOL techniques. CAM is generated from convolutional neural networks (CNNs) by using Global Average Pooling, which covers the most discriminative regions of the target object. However, object localization tasks using CAM [8] can only discover the most important parts (such as the head of a human) thus reducing the accuracy of localization. The reason is that classification models only utilize the most discriminative parts to predict the target categories.

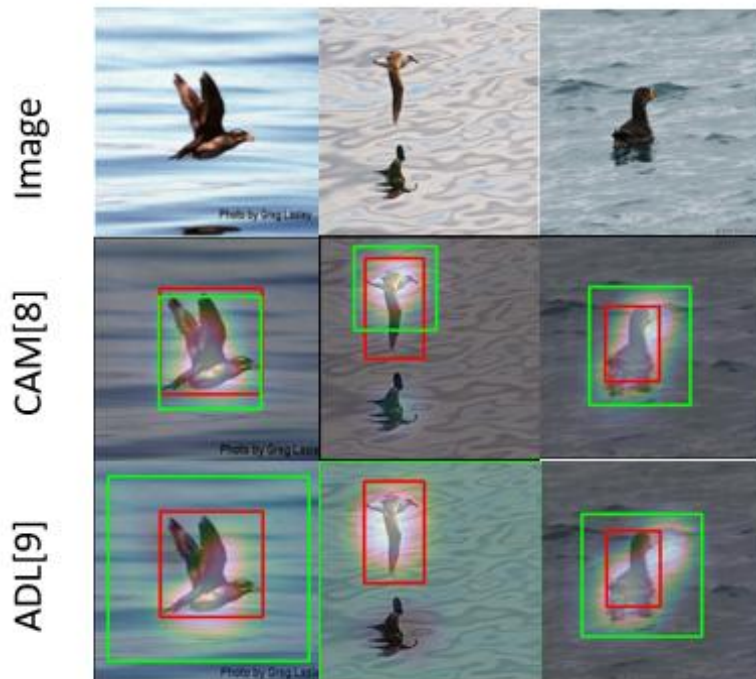


Figure 13: Localization results from CAM [8] and one of the state-of-the-art methods.

Figure 13 shows the examples compared between CAM and one of the current hiding method. For relatively small objects, CAM has already obtained better results than ADL [9] that hides the most discriminative regions to obtain the entire object. Moreover, only removing the most activated parts tends to make the model learn the less discriminative regions from outside of the object. The green bounding box is the

prediction and the red bounding box is ground truth.

To address this problem, various approaches [9] [10] [11] [12] [13] [14] have been proposed and already achieved considerable improvements. Some techniques [9] [12] [14] hide the most discriminative regions of training data or feature maps in order to impel models to learn less discriminative but necessary parts. From their experiments, we can conclude that hiding the most important parts during the training phase is effective to capture the entire object. However, previous works perform the hiding for all training or random data and do not consider the uniqueness of each input image. Hence, both localization and classification results are affected. Meanwhile, most methods highly depend on the high-level feature map, thus ignoring the general (such as boundary) information. As a result, the detection could be redundant or incomplete. Something has to be aware of is that some previous works [12] [11] introduce additional classifiers to obtain the most discriminative regions. Although these methods have achieved satisfactory results, they are too heavy for the WSOL task on both computations and memories.

In this chapter, we propose an attention-based selection strategy to detect the location of objects effectively and flexibly. Our approach can eliminate some of the problems mentioned above. We design a selection strategy that is inspired by active learning. Active learning is used to proactively make annotation requests and submit some filtered data to experts for annotation when labeled data are scarce and data without class-label are quite rich but manual labeling is very expensive. Our approach is similar to active learning to some extent. Moreover, both high-level and relatively low-level feature maps have been used to relieve the impact of regions in the foreground but do not belong to the target object.

The contributions of this chapter include:

- 1) We reveal the limitations of the existing techniques. In order to solve these problems, we propose an attention-based selection strategy (ASS) that can enhance

both classification and localization performance.

2) Our method is more efficient with less overheads compared to previous techniques.

3) Our work achieves new state-of-the-art accuracy on CUB-200-2011 [15] dataset and works very well on ILSVRC 2016 [16] dataset.

3.2 Proposed Method

In this section, we will show the details of the proposed selection strategy. Our selection strategy is based on self-attention maps from convolutional feature maps. From Figure ??, we can obtain three kinds of information, which are the most discriminative parts, the estimated area of target objects and the unhelpful regions (such as background).

The overall processing is shown in Figure 14. Self-attention map is generated by doing channel-wise pooling of the feature map. By thresholding a self-attention map, we can obtain a drop mask that hides the most discriminative regions of an input image. Drop mask and raw feature map do the spatial-wise multiplication and the output is delivered to the next layer. This case is the third part of our method, in which we only need to hide the most important regions of the image. Note that the outputs of other two situations do the spatial-wise multiplication with the original feature map in the same way.

Our selection strategy has three main parts:

1) For the feature map with much non-targeted and indistinguishable information, our method passes the combined drop mask to the next layer. The diagram is in Figure 15. We can see that the drop mask can not cover the targeted region of the object. So for this kind of feature map, non-targeted information is more discriminative than object. Therefore, our module drops these non-targeted parts and passes the dropped

feature map to the next stage.

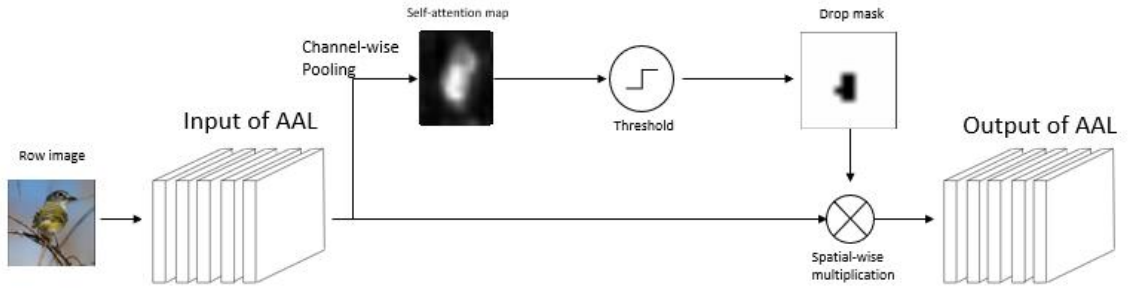


Figure 14: The diagram of our method. Note that this is the one situation of our selection strategy, for easy understanding, other two will be shown in Figure 15 and Figure 16.

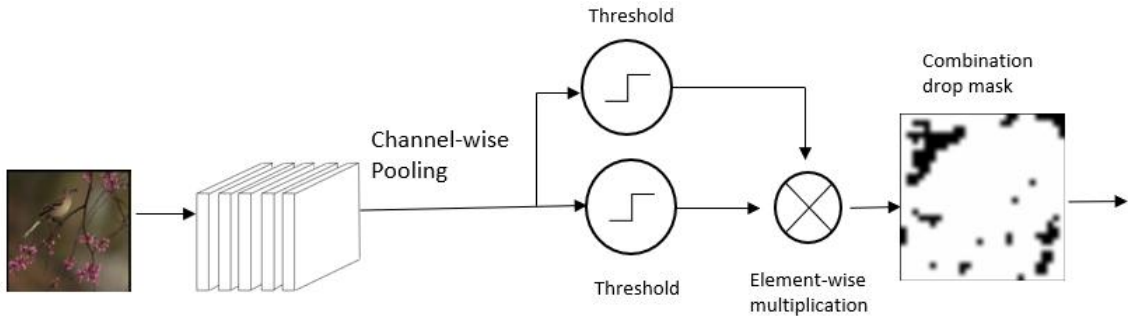


Figure 15: The diagram of our method when the feature map has much non-targeted information.

2) From Figure ??, we can see that after doing the first selection, the self-attention map can focus on the target without much unhelpful information. Moreover, we notice that when the threshold is 0.3, the drop mask can estimate the whole target area to some extent. So we utilize this kind of drop mask to express the entire region of the target. For the feature map that has a relatively small target object or the most activated region has a relatively large proportion of the object region, our model passes the raw feature map directly. The diagram is in Figure 16.

3) For the remaining part of the feature maps, our model hides the most discriminative regions. The details are shown in Figure 14.

The main parts of our method are implemented in sequential order. Images can

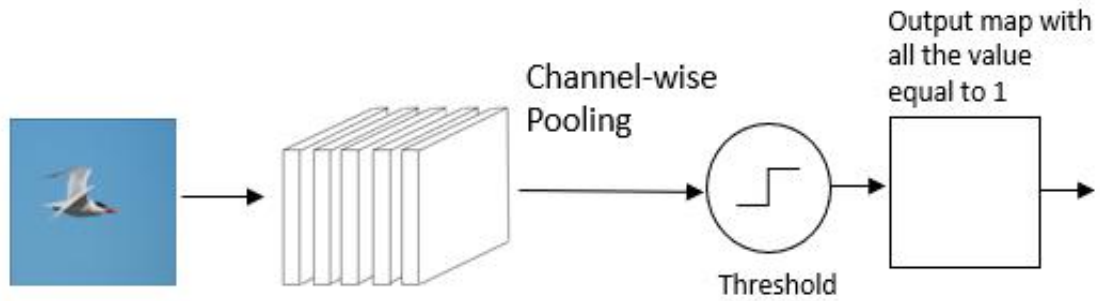


Figure 16: The diagram of our method when the input image is relatively small.

be divided into foreground and background. The target object is usually a part of the foreground region. However, there is non-objective information in foreground such as tree branches that may mislead the localization accuracy of birds. Therefore, we use a combined drop mask to remove these unhelpful pixels. Most combination drop masks are utilized in relatively low-level feature maps, which have much general information. After hiding these unhelpful parts in relatively low-level layers, the self-attention will focus on the target regions much better.

After the first part of the selection strategy, our model can deliver the feature maps with less unhelpful regions. So for the image with a relatively small target, our model passes the raw feature maps from the last layer. There are two reasons for doing this selection: 1) Unnecessary hiding may reduce classification accuracy, 2) Unhelpful removing may promote a model to focus on useless information. Moreover, when the most activated area occupies a relatively large proportion of the target object, it means that the classifier has been able to identify the target without any hiding. Therefore, our model delivers the raw feature map as well.

The remaining part of the feature maps are relatively large objects with less unhelpful parts. So for these feature maps, our method removes the most discriminative regions, similar to the most previous works.

Same as ADL [9], we use Channelwise Average Pooling to obtain a self-attention

map.

$$M_{att} = \left[\sum_{i=0}^c F_i \right] / c \quad (10)$$

Where $F \subseteq R^{H \times W \times C}$ is a convolutional feature map. H and W are the height and width, C is the number of channels. Because the value of each point represents the weight of classification, Channelwise Average Pooling can recognize the most discriminative parts.

Our selection strategy is based on a self-attention map from Eq.(10). From Figure ??, by using different thresholds on the self-attention map, we can obtain different drop masks.

$$M_{drop} = \begin{cases} l_i = 0, & \text{if } l_i \geq \alpha \cdot l_{max} \\ l_i = 1, & \text{else} \end{cases} \quad (11)$$

Where $M_{drop} \subseteq R^{H \times W}$ is the drop mask that can hide relevant regions by using threshold α . l_i is the value of i^{th} pixel. So the essence of the drop mask is to hide each pixel greater than the threshold. We set thresholds equal to 0.8 and 0.1 to obtain the most activated parts and background respectively. Note that when we use 0.1 as the threshold to drop some background regions, the condition in Eq.(11) should be $l_i < \alpha \cdot l_{max}$. For convenience, we use M_{drop1} and M_{drop2} to express them respectively. Figure 17 shows the self-attention map and different thresholding masks. The numbers are the drop thresholds. We can see that by using different thresholds, we can obtain various information from self-attention maps. Moreover, this self-attention map is generated after using a combined drop mask in a relatively low-level layer, and we can see that both the self-attention map and the most discriminative region when the threshold is 0.8 have less non-targeted information. The second row and the third row are the drop mask when threshold equals 0 and 1 respectively. From the fourth to the last row, the threshold is from 0.9 to 0.1, each decrease by 0.1.

In the first part of our strategy, we want to find images with obvious foreground but non-objective regions. Based on the drop mask, we can obtain the most activated

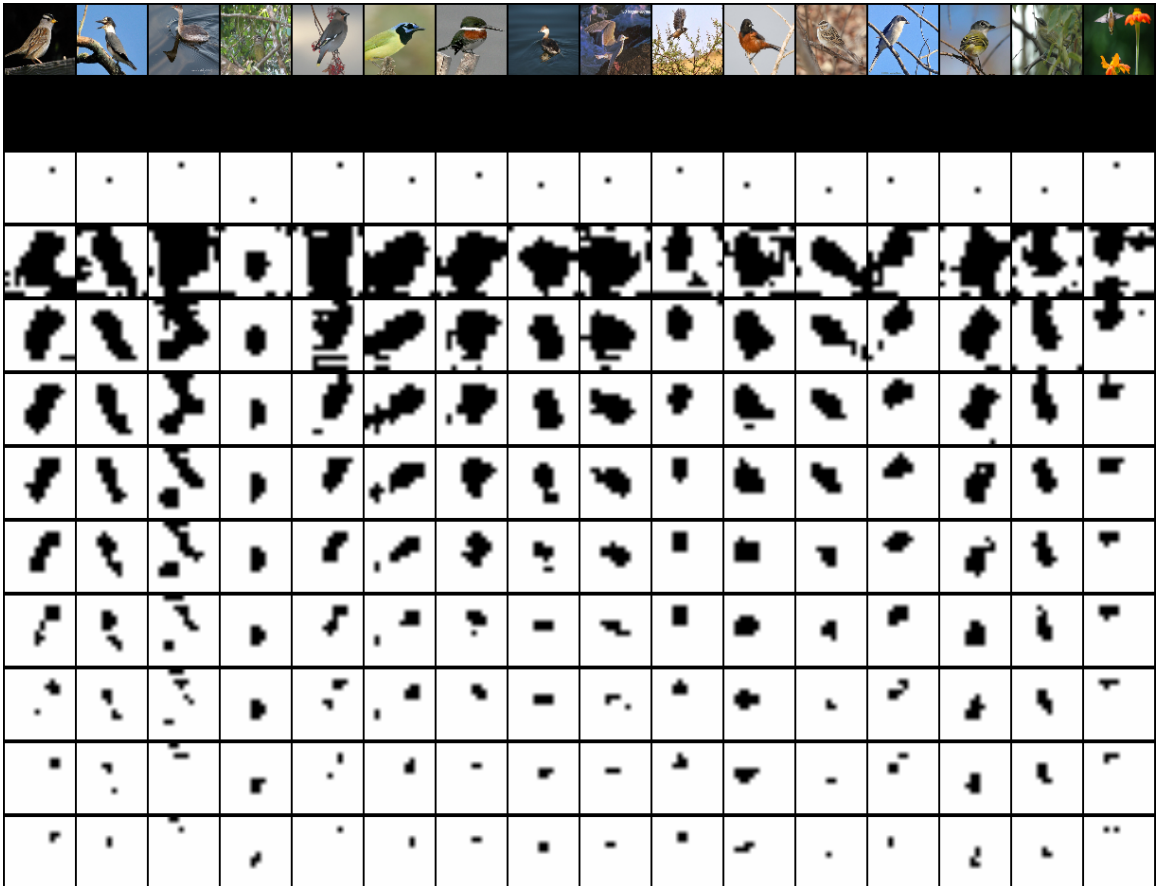


Figure 17: The self-attention map and different thresholding masks.

pixels of the feature map. If some non-targeted regions are much more obvious than targets, it will mislead the localization results. For the good drop mask in Figure ??, the dropout region is a continuous area that can hide the most discriminative part of the object. But for the drop mask in Figure 15, the drop mask can only remove the discrete pixels of the image. So in case, there is much non-targeted information in the background region, which will affect the detector. Therefore, we use drop masks to hide these unhelpful but discriminative pixels and some background regions. We select this kind of feature map mostly from relatively low-level layers because they have much more general information. The selected M_{drop1} of the feature maps needs to satisfy two conditions:

1) M_{drop1} should have a certain number of activated pixels. If there are very few points in the drop mask, the detector does not adequately understand the image in this stage.

2) Important pixels have large dispersion and the most activated regions are discrete. As we mentioned above, a good drop mask is supposed to have a continuous region that can remove the most discriminative part of the object. So when the pixels are very discrete, the attention map can not focus on the target very well. In this case, we use threshold to remove background and foreground but non-targeted information, which can make the detector focus on the target better.

Once M_{drop1} meets all the conditions above, self-attention will be tough to show the weights of targeted objects. One of the examples is shown in Figure 15. In this case, we use M_{drop1} to hide regions that have much non-targeted information. Moreover, we also use M_{drop2} to hide as much background as possible. Condition (1) is simple to implement using the coordinate of each pixel. In mathematics, the variance has been utilized to measure the dispersion of one-dimensional data. Therefore, we use the summation of two single-dimensional data to express the dispersion of the most

activated pixels.

$$D_i = \mu_1(Var(X) + Var(Y)) + \mu_2P \quad (12)$$

Where D_i is dispersion of i^{th} M_{drop1} , X and Y are two arrays of axes, which can obtain from the position of each activated pixel. P is the number of discrete points of a drop mask. μ_1 and μ_2 are two parameters that express the influence level of two values.

After the first selection, self-attention can obtain the drop masks with less non-targeted information, which means drop masks are able to hide the most discriminative regions of targets successfully. So in this case, our model simply delivers raw feature maps for the feature maps with relatively small target areas and removes the most activated regions for the remaining part of the feature maps. One of the examples is shown in Figure 16. In this case, our model outputs a map with all the values equal to 1 by using threshold. So after doing spatial-wise multiplication with the raw feature map, the model will deliver the original feature map to the next layer. We set 0.3 as the threshold to obtain estimated area of target objects from Figure ??.

In general, our method divides images into three categories. The first class is the image with much non-targeted and indistinguishable information, we use equation Eq.(12) to measure the degree of the dispersion of the drop mask to judge whether the drop mask can cover the targeted part or not. For the bad drop mask that can not focus on the target, our method hides the non-targeted regions to make the model learn the target region of this kind of images. The second type is the image with a small target. We use a thresholding self-attention map that is shown in Figure 17. When the threshold is 0.3, the drop mask can estimate the whole region of the object. So we utilize the ratio of the estimated target area to the total area of the image to identify small objects. For the image with a small object, our module just passes the original feature map to the next stage. Finally, for the rest of the images, we only hide the most discriminative part of them. Algorithm 1 shows the processing of our

proposed selection strategy.

Algorithm 1 Selection strategy

Input: $x \subseteq R^{H \times W \times C}$: feature map; b : the total pixel number of one feature map; v : the most discriminative pixel dispersion; d : total pixels of target

Output: y : dropped feature maps

- 1: **for** $i = 1 \rightarrow batch_size$ **do**
- 2: **if** $v \geq$ dispersion threshold **then**
- 3: selected_map[i] = a drop mask that can remove unhelpful information
- 4: **else if** $d/b <$ small area threshold **or** $a/d >$ drop mask proportion threshold **then**
- 5: selected_map[i] = a feature map that all the values are 1
- 6: **else**
- 7: selected_map[i] = a drop mask to remove the most discriminative parts
- 8: **end if**
- 9: **end for**
- 10: **return** $y = x \cdot$ selected_map

3.3 Experiment Results

In this section, we will discuss the experiment results.

3.3.1 Implementation Details

We use VGG-GAP [8] as the backbone network. Our module is plugged into some certain positions of the backbone. We obtain the heatmaps and bounding boxes using the same method as CAM [8]. For the attention thresholds, we set 0.8 to detect the most discriminative parts, 0.3 and 0.2 to estimate target and background area respectively. For other parameters, we set 20% as the small target area threshold and 30% as the drop mask proportion threshold. Therefore, when the estimated target area is less than 20% of the entire feature map or the most discriminative regions occupy more than 30% of the target region, our module will deliver the raw feature map to the next layer. Meanwhile, we use 40 as the variance threshold to select the feature map with much non-targeted and indistinguishable information. We plug our module into three positions that are pool3 layer, pool4 layer and conv5-3 layer. The output size of the pool3 layer is 28×28 . 14×14 is the output size of both pool4 layer

and conv5-3 layer. Note that the parameters above are the optimal setting based on our experiment, different values may obtain different results. The backbone network is pre-trained on ILSVRC and fine-tuned with the learning rate 0.001 and batch size 32. We trained our model using the GeForce RTX 2060 GPU.

3.3.2 Ablation Study

In this subsection, we use the results of the CUB-200-2011 [15] dataset to do some ablation study.

Table 1 shows the different results according to the choice of positions to plug our module. From these results, CAM [8] has the lowest classification error but the highest localization error. We believe that CAM [8] only uses raw feature maps that will make the model only focus on the most discriminative regions. We also find that our method can further improve the localization accuracy when it is plugged into both high-level layers and relatively low-level layers. Note that for the feature map from the same image, our model may perform the different operations on different plugged layers. Such as for the small object with much non-targeted information, our model hides as much as unhelpful regions on the Pool3 layer. So for the Pool4 and Conv 5-3 layers, the raw feature map will be delivered to the next layer.

Table 1: The different Results for the choice of positions to plug our module

Plugged position	Top-1 cls.err	Top-1 loc.err
N/A(CAM[8])	23.86	66.05
conv 5-3	23.97	51.65
+ pool4	24.82	50.00
+ pool3	25.51	45.45
+ pool2	26.79	47.51
+ pool1	26.92	47.24

Table 2 illustrates the necessity to introduce our selection strategy for WSOL. CAM [8] means the WSOL without any drop masks and CAM + M_{drop1} means that

the drop masks are utilized for every feature map to hide the most discriminative regions. The model with none drop masks obtains the highest classification performance but the lowest localization accuracy. When we use the drop masks for all feature maps, the classification result is very low. We believe that it is because the classifier has less clue to do the classification.

Table 2: The results according to different methods

Method	Top-1 cls.err	Top-1 loc.err
CAM [8]	23.86	66.05
CAM + M_{drop1}	42.63	55.89
ASS(Ours)	25.51	45.45

3.3.3 Comparison with State-of-the-art Methods

In this subsection, we compare the proposed ASS with the state-of-the-art techniques on CUB-200-2011 [15] and ILSVRC 2016 [16] datasets. All the methods utilize VGGnet-GAP [17] as the backbone network. The results are reported in Table 7 and Table 8 respectively.

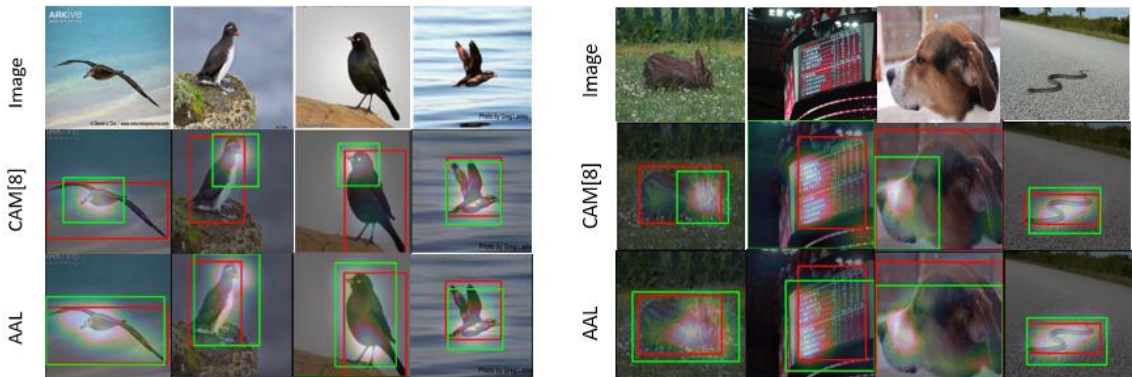


Figure 18: Comparison with CAM [8] on CUB-200-2011 (left) and ILSVRC 2016 (right) datasets. Green bounding box is prediction and the red bounding box is ground truth.

On CUB-200-2011 [15] dataset, although CAM[8] has the highest classification accuracy, ASS outperforms VGGnet-CAM [8] by 20.6% on the localization accuracy. Furthermore, our method achieves 45.45% of the Top-1 localization error, which is better than the current state-of-the-art performance. On the ILSVRC 2016[16] dataset, we can see that ASS improves both classification and localization results compared with CAM [8]. Although ADL [9] obtains the best classification performance, ASS obtains the similar result. Compared with ACoL [12] that has the best localization accuracy apart from ours, our method improves the localization result but not very obviously.

Table 3: Quantitative evaluation performance on CUB-200-2011 dataset

Method	Top-1 cls.err	Top-1 loc.err
VGGnet-CAM [8]	23.86	66.05
VGGnet-ACoL [12]	28.10	54.08
VGGnet-SPG [11]	24.50	50.00
VGGnet-ADL [9]	34.73	47.64
VGGnet-DANet [10]	24.60	47.48
VGGnet-Cutmix [13]	-	47.47
VGGnet-ASS(Ours)	25.51	45.45

Table 4: Quantitative evaluation performance on ILSVRC 2016 dataset

Method	Top-1 cls.err	Top-1 loc.err
VGGnet-CAM [8]	33.40	57.20
VGGnet-ACoL [12]	32.50	54.17
VGGnet-ADL [9]	30.52	55.08
VGGnet-Cutmix [13]	-	56.55
VGGnet-ASS(Ours)	30.59	53.76

For the parameters overheads, ASS is similar to ADL [9] and CAM [8], which does not use any additional classifiers. For the computation overheads, compared with augmentation methods such as Cutmix [13], which cuts one region of training

data and mixes with other images. Ours has less calculation costs.

From Figure 21 , we can see that our approach obtains better results than CAM [8]. Moreover, in the fourth column of both figures, our method maintains the performance of CAM [8] when relatively small objects are the inputs.

3.4 Conclusion

In this chapter, we propose a simple but effective method for Weakly Supervised Object Localization task. We reveal the limitations of current methods and design a selection strategy to eliminate them. Our method only hides the feature maps that are necessary to hide. For the small objects that have already achieved good results on CAM, our module delivers the raw feature map to the next layer. Moreover, our approach can remove much unhelpful information that will mislead the localization. Compared with current state-of-the-art techniques, our method works very well on localization accuracy without too much overhead on CUB-200-2011 and ILSVRC 2016 datasets. In our perception, our method is the first work to consider the different situations of training images. Therefore, our work provides new insights to do the Weakly Supervised Object Localization task.

Chapter 4

Attention-based Dual Hiding

Method

As we mentioned previously, most current methods tend to utilize Class Activation Map (CAM) that can only highlight the most discriminative parts rather than the entire target. The common method to address this kind of limitation is to hide the most discriminative regions during training. However, considering that the pixels of the feature maps from the convolutional layer have a strong relationship. Previous removing techniques can not hide the most important information completely, thus the limitation of Class Activation map may not be solved very well. In this chapter, we propose an Attention-based Dual Hiding method, which can eliminate the limitations of both CAM and current hiding techniques. Experiments demonstrate that the proposed method works very well on CUB-200-2011 and ILSVRC 2016 datasets.

4.1 Introduction

Some current research illustrates that hiding methods can improve the performance of Weakly Supervised Object Localization tasks. However, considering the characteristics of the feature map from the convolutional network, there are strong relationships

between each pixel of the convolutional feature map. Therefore, the current hiding method can not remove the discriminative information very well. Thus, it may affect the final results of the localization model. In this paper, we propose a Dual Hiding method to eliminate the limitation of current hiding techniques. The Dual method is inspired by [61]. Instead of only removing discriminative pixels, we utilize spatial-softmax[6] and proposed an area hiding method to remove expected regions of both channel and spatial level. Spatial-softmax is first used in [6] to generate the coordinates of expected pixels. The overall method is shown in Figure 19. Proposed technique can remove the most expected region of both channel and spatial space. The inputs are feature maps from convolutional networks. For channel space, we first use spatial-softmax to generate the probability distribution map of each channel. After that, Global Average Pooling is performed to obtain the probability of each channel. We select the top-1 channel that has much more classification information to generate the most expected region. For the spatial space, by doing channel-wise pooling, we can obtain a self-attention map that can show the important region of the image. By combining the thresholding self-attention map and expected region, the spatial drop mask is generated. The method to obtain the expected region of spatial space is the same as channel level. We combine the channel drop mask and spatial drop mask to obtain the final drop mask. For the last step, drop mask and original feature map do element-wise multiplication. The output of multiplication are the feature maps without important information. The diagram of area hiding is shown in Figure 20.

The contributions of this work include:

- 1) Based on the limitations of current hiding methods. We propose a dual hiding method that can better remove the most discriminative information and improve the localization performance.
- 2) Compared with data augmentation and the technique that utilizes additional

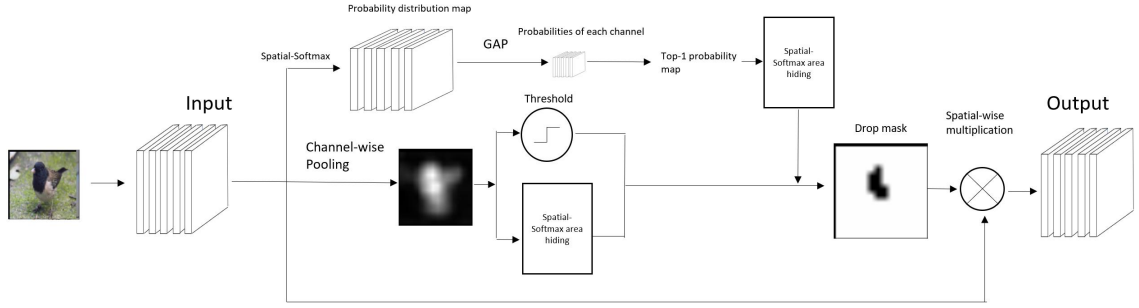


Figure 19: The diagram of our hiding method.

architecture, our method needs less overheads.

3) Our method works very well on CUB-200-2011[15] and ILSVRC 2016[16] datasets.

4.2 Proposed method

Figure 14 shows the proposed attention module. The inputs of the module are the feature maps from the convolutional layer. For the hiding method of channel space, we utilize spatial softmax[6] to generate probability distribution.

$$s_{cij} = e^{a_{cij}} / \sum_{i'j'} a_{i'j'} \quad (13)$$

Where c is the channel of the feature map and (i, j) is the pixel coordinate. By using Eq.(13), we can obtain a probability distribution that represents the pixel weights based on the classification labels. Then we perform Global Average Pooling on the distribution map to produce the weights of each channel.

$$P_G = \left[\sum_{i=1}^w \sum_{j=1}^h P_{1ij}, \sum_{i=1}^w \sum_{j=1}^h P_{2ij}, \dots, \sum_{i=1}^w \sum_{j=1}^h P_{cij} \right] / (w \times h) \quad (14)$$

Where $P \subseteq R^{H \times W \times C}$ is a probability distribution map from spatial softmax layer[6]. H and W are the height and width of feature map, C is channel number. The output of Eq.(14) shows the weights of each channel. Because the classifier is trained to do the classification, so for the feature map that has a larger weight than others,

there is much more classification information on it. Based on the weights of each feature map, we select top-1 probability distribution maps. Spatial soft-argmax[6] is performed on selected top-1 channel respectively to generate top-1 expected pixels. The method to convert the probability to 2D coordinates is shown in [6]. As we mentioned above, the pixels of the feature map from the convolutional layer have a strong relationship. That means when we obtain the expected pixel, only removing this pixel can not remove all the information of this pixel. Therefore, we perform an area hiding method to the expected pixel. By using area hiding, we not only remove the expected pixel but also hide surrounding pixels to try to completely eliminate the information of the expected pixel. The area hiding method is shown in Figure 20.

The self-attention map and the drop mask is generated in the same way as the last section.

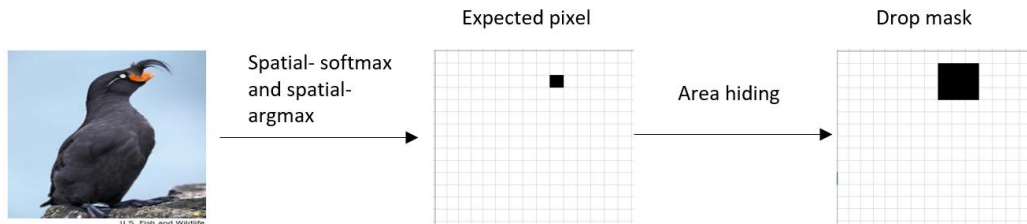


Figure 20: The diagram of area hiding method.

In general, the final drop mask consists of three parts that are the expected region of the top-1 channel, the expected parts of the self-attention map and the thresholding drop mask.

Finally, the final drop mask performs the element-wise multiplication with input feature maps. The output will be the feature map without the most discriminative region. For the selection of drop mask, we utilize the selection strategy in[18] that can dynamically decide the drop mask.

4.3 Experiment Results

In this section, the experiment results are provided.

4.3.1 Implementation Details

We utilize VGGnet[19] and ResNet[7] as backbone networks. The same as CAM[8], we remove the fully connected layer with a global average pooling layer[5]. Our proposed method is inserted into some positions of backbone networks. For example, we plug our method after pool3, pool4 and conv5-3 layer of VGGnet[19]. The predicted bounding boxes are generated in the same way as CAM[8]. For the drop mask of self-attention, we set 0.8 and 0.9 as the threshold for those two backbone networks. The network is pre-trained on ILSVRC dataset [16]and fine-tuned with learning rate 0.001 and batch size 32. We utilize GeForce RTX 2060 GPU to train our model.

4.3.2 Comparison with State-of-the-art Methods

Table 5: Quantitative evaluation performance on CUB-200-2011 dataset.

Method	Top-1 cls.err	Top-1 loc.err
VGGnet-CAM[8]	23.86	66.05
VGGnet-ACoL[12]	28.10	54.08
VGGnet-SPG[11]	24.50	50.00
VGGnet-ADL[9]	34.73	47.64
VGGnet-DANet[10]	24.60	47.48
VGGnet-Cutmix[13]	-	47.47
VGGnet-AMH(Ours)	26.46	43.16
ResNet50-CAM[8]	21.51	58.83
ResNet50-ADL[9]	24.68	48.09
ResNet50-Cutmix[13]	-	45.2
ResNet50-AMH(Ours)	25.17	44.47

In this section, we compare the performance of our method with current techniques on CUB-200-2011 [15] and ILSVRC 2016 [16] datasets. The backbone networks are

Table 6: Quantitative evaluation performance on ILSVRC 2016 dataset.

Method	Top-1 cls.err	Top-1 loc.err
VGGnet-CAM[8]	33.40	57.20
VGGnet-ACoL[12]	32.50	54.17
VGGnet-ADL[9]	30.52	55.08
VGGnet-Cutmix[13]	-	56.55
VGGnet-AHD(Ours)	31.47	53.15
ResNet50-CAM[8]	23.94	54.65
ResNet50-AMH(Ours)	22.07	51.84

VGGnet [19] and ResNet50 [7]. Table 7 and Table 8 show the results of each dataset respectively.

For the classification performance, CAM performs very well on birds dataset with VGGnet [19]. But for the results on ILSVRC 2016 [16] dataset, classification accuracy of CAM is not better than current method. For the localization results, our method works very well on CUB-200-2011 [15] dataset using both VGGnet [19] and ResNet50 [7]. The results of bird dataset have significant improvement compared with current techniques. For ILSVRC 2016 [16] dataset, although proposed method works better, the improvement is not very obvious. After checking the bounding box labels of ILSVRC 2016 [16] dataset, we noticed that for a group of same objects in one image, the ground-truth bounding box of it covers only one of them. For example, there are 6 apples in the image, our method can detect all the apples and generate the predicted bounding box. However, the label only shows one of them. That will absolutely impact the final accuracy.

4.3.3 Ablation Study

In this section, we use the result on CUB-200-2011[15] dataset with VGGnet[19] to provide some ablation study.

Table 7 shows the different results when our module is inserted into different

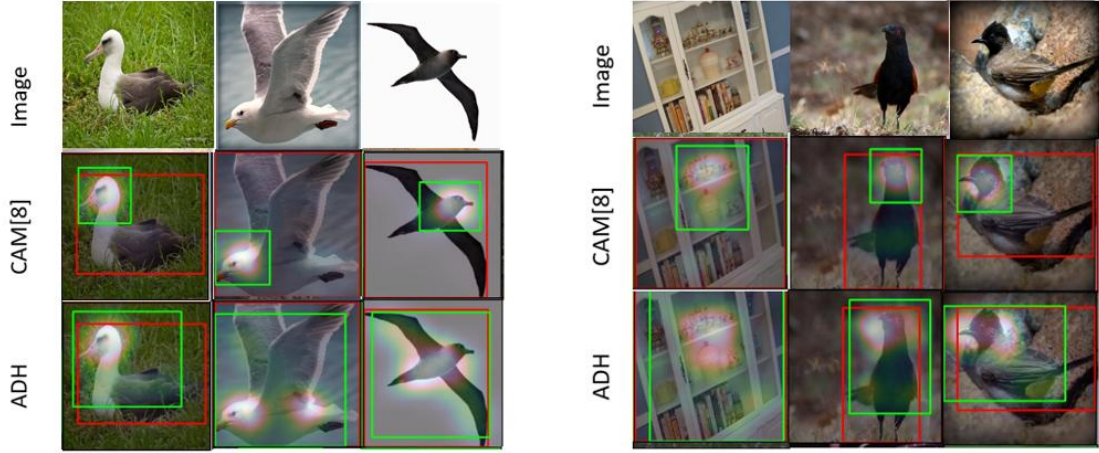


Figure 21: Comparison with CAM [8] on CUB-200-2011 (left) and ILSVRC 2016 (right) datasets. Green bounding box is prediction and the red bounding box is ground truth.

positions of the backbone network. The result of '+' pool4 is generated by using our method in conv 5-3 and pool4 layer. From the table, we can see that the detector has the highest classification accuracy without any hiding. As the number of layers increases, the performance of the classifier decreases. We already mentioned in the first section that the classifier utilizes the most discriminative part of the object to do the classification. Therefore, when we perform a hiding method, the model has less clue to do the classification. For the localization accuracy, we can see that when we add the method into three different layers, the result is the best.

Table 8 shows the results by using different drop masks. CAM[8] obtains the best classification performance without any drop masks. For the localization result, the performance is increasing when we add the expected channel and spatial region. As the drop mask becomes larger, the classification result decreases because there is less classification information. We also do the experiments that select top-n ($n > 1$) expected points of both channel and spatial space, the results are worse than only top-1 selection.

For the overheads, our hiding method does not perform any augmentation or introduces any additional classifier.

Table 7: The Different Results For The Choice of positions to plug our module.

Plugged position	Top-1 cls.err	Top-1 loc.err
N/A(CAM[8])	23.86	66.05
conv 5-3	23.95	52.19
+ pool4	24.12	46.31
+ pool3	26.46	43.16
+ pool2	27.08	45.07
+ pool1	29.30	47.81

Table 8: The Different Results by using different drop masks.

Drop mask	Top-1 cls.err	Top-1 loc.err
N/A(CAM[8])	23.86	66.05
drop mask of self-attention	24.13	45.04
+ Top-1 spatial region	24.25	44.47
+ Top-1 channel region	26.46	43.16
+ Top-2 spatial region	26.86	44.94
+ Top-2 channel region	27.15	46.80

Table 9: Ground-truth accuracy of each method. Since few methods provided the value of this evaluation metric, we only show available methods in this table.

Method	Ground-truth loc.err
VGGnet-CAM[8]	36.77
VGGnet-ADL[9]	30.64
VGGnet-AMH(Ours)	25.23

4.3.4 Results

In this section, we compare the performance of our method with current techniques on CUB-200-2011[15] and ILSVRC 2016[16] datasets. The backbone networks are

VGGnet[19] and ResNet[7]. Table 7 and Table 8 show the results of each dataset respectively.

For the classification performance, CAM performs very well on birds dataset with VGGnet[19]. But for the results on ILSVRC 2016[16] dataset, classification accuracy of CAM is not better than the current method. For the localization results, our method works very well on CUB-200-2011[15] dataset using both VGGnet[19] and ResNet50[7]. The results of the bird dataset have significant improvements compared with current techniques. For ILSVRC 2016[16] dataset, although the proposed method works better, the improvement is not very obvious. After checking the bounding box labels of ILSVRC 2016[16] dataset, we noticed that for a group of the same objects in one image, the ground-truth bounding box of it covers only one of them. For example, there are 6 apples in the image, our method can detect all the apples and generate the predicted bounding box. However, the label only shows one of them. That will absolutely impact the final accuracy. In [60], in order to evaluate the WSOL performance much more fair, [60] suggested future work providing Ground-truth localization accuracy. Our method works better from table 9.

4.4 Conclusion

In this chapter, we proposed a dual hiding method for Weakly Supervised Object Localization (WSOL) task. Our starting point is based on the strong relationship of each pixel of convolutional feature map. According to the experiments, we can see that the proposed method can effectively improve the localization performance on CUB-200-2011 and ILSVRC 2016 datasets compared with current WSOL techniques. The performance on ILSVRC 2016 dataset is not very satisfactory. We mentioned the reason in the last subsection. Therefore, our future work is supposed to improve the performance of ILSVRC 2016 dataset.

Chapter 5

Revisiting Class Activation Map

We reviewed the limitations of the Class Activation Map that can only highlight the most discriminative region of the object. In order to eliminate the issue of CAM, the hiding methods are proposed. In this chapter, I will revisit the processing of the Class Activation Map and review the reasons that make the CAM such ill-posed. Based on those reasons, we proposed corresponding solutions to improve the localization accuracy on CAM level. Proposed methods do not introduce any hyperparameters and additional network. Proposed two stage localization method can evaluate each potential threshold and select the optimal threshold for different images. The localization performance is improved significantly compared with current WSOL methods.

5.1 Introduction

Class Activation Map [8] has an issue that can only cover the most discriminative part of the target. Current work has shown that hiding methods can promote the CAM [8] to cover the entire pattern of the object. However, previous hiding methods will introduce redundant thresholds or overheads, which will make the model harder to train. Meanwhile, the pre-trained backbone greatly changed the architecture of the network, which sacrifices the classification accuracy. The localization performance is

based on the classification accuracy to some extent, so how to localize better without much classification accuracy reducing is a big challenge. The more details of the ill-posed hiding method are in [60].



Figure 22: The figure shows the issue of CAM that can only highlight the most discriminative region of the target.

In this chapter, we rethink the processing of CAM [8] and find three causes of ill-posed CAM. According to those three issues, we proposed three corresponding methods to eliminate them. Our methods do not require any additional hyperparameters and network. We proposed a two stage localization method to evaluate each potential threshold and select the optimal threshold for different images. The results illustrate that the proposed method can significantly improve the localization performance without much classification accuracy reduction.

The contribution of this work include:

- 1) Based on the three issues of CAM, we propose corresponding methods to eliminate them.

- 2) In our perception, the proposed two stage localization method is the first work that can evaluate each potential threshold and select the optimal threshold for different images in WSOL tasks.

- 3) Our method significantly improves the localization performance without much classification sacrifices.

5.2 Revisiting Class Activation Map

The biggest difference between the pre-trained backbone and the CAM model is the Global Average Pooling between the last convolutional layer and fully connected layer.

$$P_G = \left[\sum_{i=1}^w \sum_{j=1}^h P_{1ij}, \sum_{i=1}^w \sum_{j=1}^h P_{2ij}, \dots, \sum_{i=1}^w \sum_{j=1}^h P_{cij} \right] / (w \times h) \quad (15)$$

Where $P \subseteq R^{H \times W \times C}$ is the feature map from the last convolutional layer. H and W are the height and width of feature map, C is channel number. The output of Eq.(15) shows the weights of each channel.

Next step, we pass the weight of each channel to the fully connected layer to compute the final Class Activation Map [8]. The equation of calculating the CAM [8] is:

$$M_c = \sum_{k=1}^k w_{k,c} \cdot P_c \quad (16)$$

Where $M \subseteq R^{H \times W}$ is the Class Activation Map and $P \subseteq R^{H \times W \times C}$ is the feature map from the last convolutional layer. c is the predicted classification result. We can see in the Eq.(16) that the Class Activation map is generated from the weighted linear summation of the feature maps. After obtaining M_c , the heatmap is normalized using Min-Max normalization and resized to the input image size. We named the final Class Activation map M_c^f . Then we use a threshold to extract the bounding box of the target, which is:

$$\eta_l = \delta_l \cdot \max M_c^f \quad (17)$$

The δ_l in Eq.(17) is a number ranging from $[0,1]$.

5.3 The issues of CAM

In this section, we are not going to discuss the obvious limitation of CAM that only highlights the most discriminative region. We are planning to discuss the issues of why CAM only covers the most discriminative part.

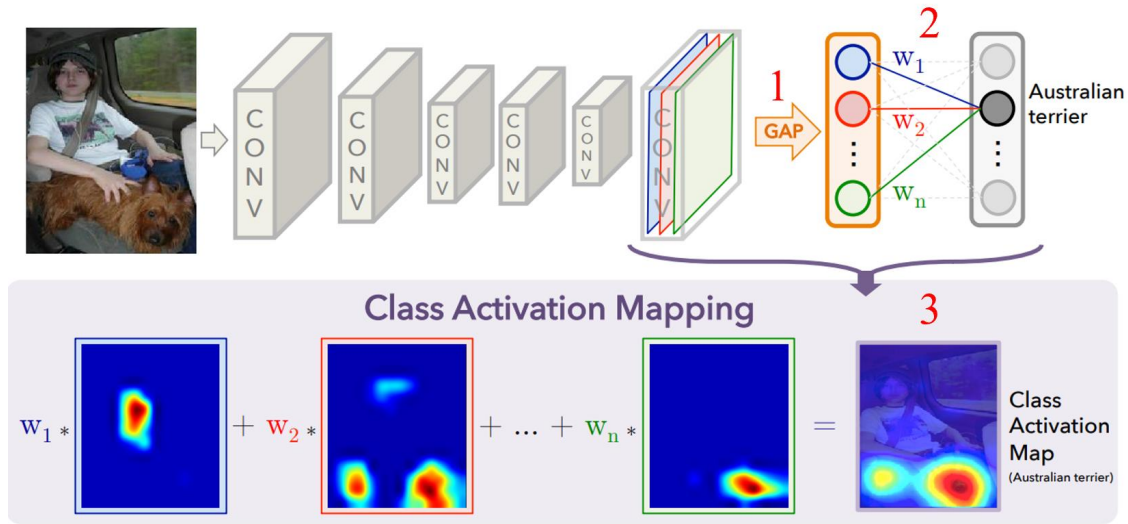


Figure 23: The three issues that make the CAM ill-posed. The figure is extracted from [8], we only labeled three numbers to show the problems of CAM.

From Figure 23 and the introduction of the last section, we notice that there are three issues to make the CAM only focus on the most discriminative region.

1) **Global Average pooling** [5] GAP plays an important role in CAM. However, GAP assigns equal weights to every pixel whether the pixels are activated or not. In this way, the background pixels will be encouraged and the targeted region will be penalized, which is unfair for the stored spatial information.

2) **The weights of the feature maps** The fully connected layer assigns both positive and negative weights to the feature maps based on the classification results. The positive feature maps are beneficial to the final classification and the negative feature maps will be inhibited. During the training, the network tries to keep the

spatial information of the input the image. Even if some pooling methods are performed between each block, the feature map from the last convolutional layer still retains the much amount of spatial information. In other words, the less important information can not disappear during training. So the only explanation is that the less important parts are inhibited in the negative weights. The theoretical basis is that the classification task only requires the most discriminative region to do the classification. Therefore, the FC layer will assign the positive weights to the feature maps that only have the most discriminative information. The figure 24 shows some examples. We can see that the negative feature maps also retain the target region, especially the less important part.

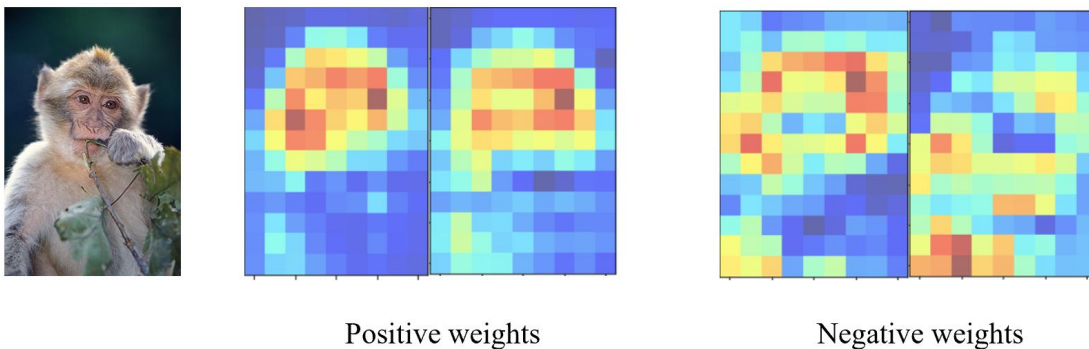


Figure 24: The examples of positive feature maps and negative feature maps.

3) **The threshold to extract bounding box** As we discussed in the last section, a binary mask is generated by thresholding the final Class Activation map to extract the bounding box. That is a hyperparameter that can not be trained, so the only way to assign that is based on the experiment. We usually set 0.2 as the threshold. However, the threshold of the heatmap highly depends on the peak value. When the peak value is very large, the bounding box may only focus on a small part of the activated region, even if the activated regions have already covered the entire object. On the contrary, the bounding box may be divergent to the whole image. The examples are shown in figure 25. As we can see in the figure, even if the Class

Activation Map has activated some less important region of the target. Due to the improper threshold selection, the bounding box can not cover the whole activated region, which affects the localization performance. Based on the three issues of CAM, we proposed three corresponding methods to eliminate the problems so as to improve the localization accuracy. The details will be discussed in the next section.

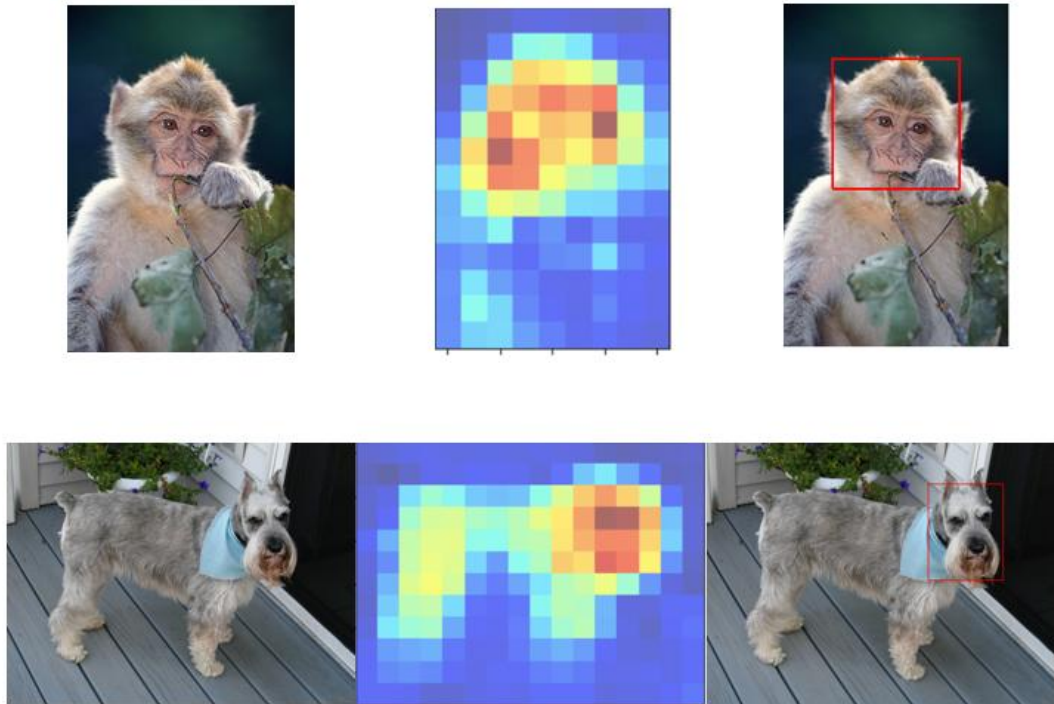


Figure 25: The examples show the issue of thresholding bounding box.

5.4 Proposed methods

We reviewed the limitations of CAM [8] in the last section and we proposed three methods to eliminate them.

5.4.1 Weighted Global Average Pooling

Inspired by the motivation in the last section. We proposed a weighted GAP that can assign different weights to the feature map based on the classification. The weighted matrix is:

$$S_{cij} = e^{a_{cij}} / \sum_{i'j'} a_{i'j'} \quad (18)$$

Where c is the channel of the feature map and (i, j) is the pixel coordinate. By using Eq.(18), we can obtain a probability distribution that represents the pixel weights based on the classification labels. By multiplying the feature maps with the weights matrix, the weighted feature maps are generated that the targeted pixels will be encouraged and the background region will be penalized. So the proposed WGAP is:

$$P_G = \left[\sum_{i=1}^w \sum_{j=1}^h P_{1ij} S_{1ij}, \sum_{i=1}^w \sum_{j=1}^h P_{2ij} S_{2ij}, \dots, \sum_{i=1}^w \sum_{j=1}^h P_{cij} S_{cij} \right] / (w \times h) \quad (19)$$

Where $P \subseteq R^{H \times W \times C}$ is the feature map from the last convolutional layer. H and W are the height and width of feature map, C is channel number. For the output of WGAP, that will highly depend on the activated pixels, which is more beneficial to store the spatial information. The diagram is figure 26.

5.4.2 Recombining the weights of FC layer

The weights of fully connected layer is used to encourage the positive feature maps and inhibit the negative ones, which can be express like this:

$$M_c = M_p + M_n = \sum_{p=1}^p w_{p,c} \cdot F_p + \sum_{n=1}^n w_{n,c} \cdot F_n \quad (20)$$

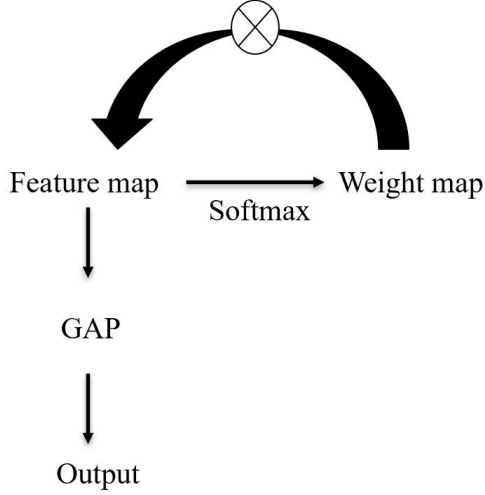


Figure 26: The diagram of proposed WGAP.

Where $M \subseteq R^{H \times W}$ is the Class Activation Map. $M_p \subseteq R^{H \times W}$ and $M_n \subseteq R^{H \times W}$ are the positive and negative CAM respectively. p and n are the number of the positive and negative weights respectively.

However, we have discussed in the last section that the negative channel still retains much target information, especially less important regions. Therefore, the negative feature maps are not supposed to be inhibited. Figure 27 shows the details of how negative weights affect the final CAM.

According to figure 27, we can see that the original CAM inhibits the negative channels that will make the CAM ill-posed. When the negative feature maps are not inhibited, we can see the result of the second row, the less important regions are activated. So our Recombining weights for CAM is:

$$M_c = M_p - M_n = \sum_{p=1}^p w_{p,c} \cdot F_p - \sum_{n=1}^n w_{n,c} \cdot F_n \quad (21)$$

By recombining the weights of the FC layer, the better CAM is generated.

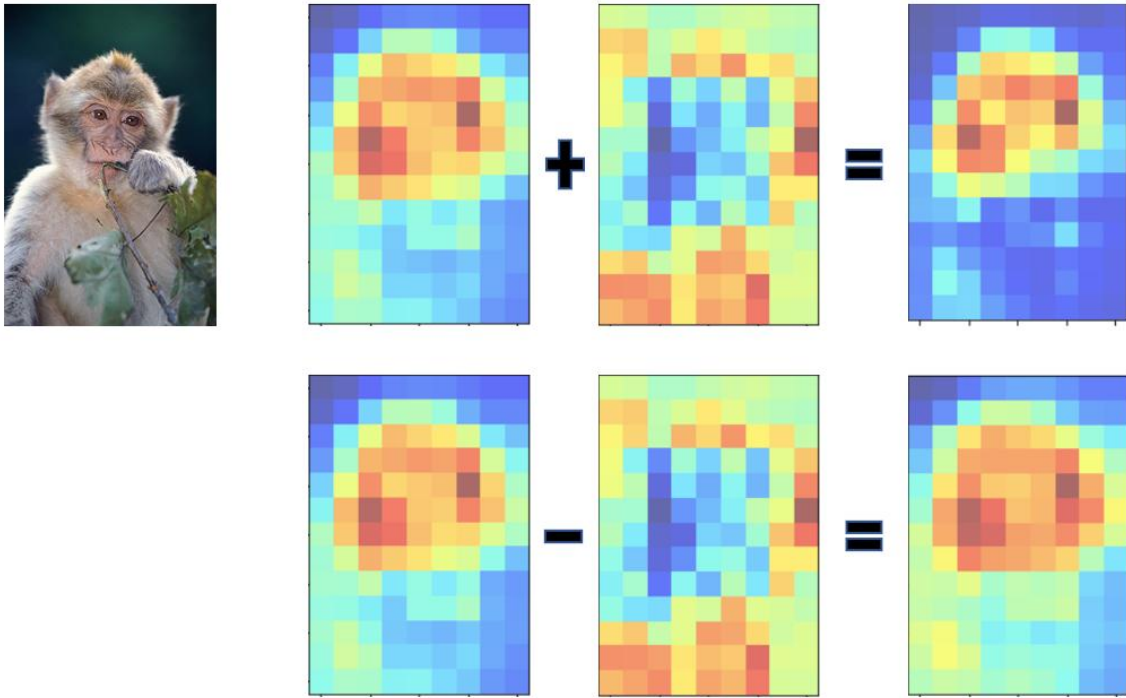


Figure 27: The diagram of computation of CAM. The first row is the case that the negative channels are inhibited. The second row shows the results without inhibiting the negative feature maps.

5.4.3 Two stage localization

As we discussed in the last section, although the CAM can cover the whole region of the target, the bounding box highly depends on the threshold. The threshold is tough to be selected. Normally, we set the threshold based on the experiment or the optimal value of the evaluation, which is unfair to the different images. We can find the bad samples very easily, which is shown in figure 25. So in this subsection, we proposed a two stage localization method that can evaluate multiple thresholds and select the optimal one for each image. The diagram is shown in figure 28.

For the first stage, we generate the Class Activation Map by using proposed WGAP and recombining weights methods. Then we enumerate k thresholds to extract corresponding bounding boxes. Based on the bounding box, some seed vectors are extracted, which determine the bounding box. The number of the seeds is not set

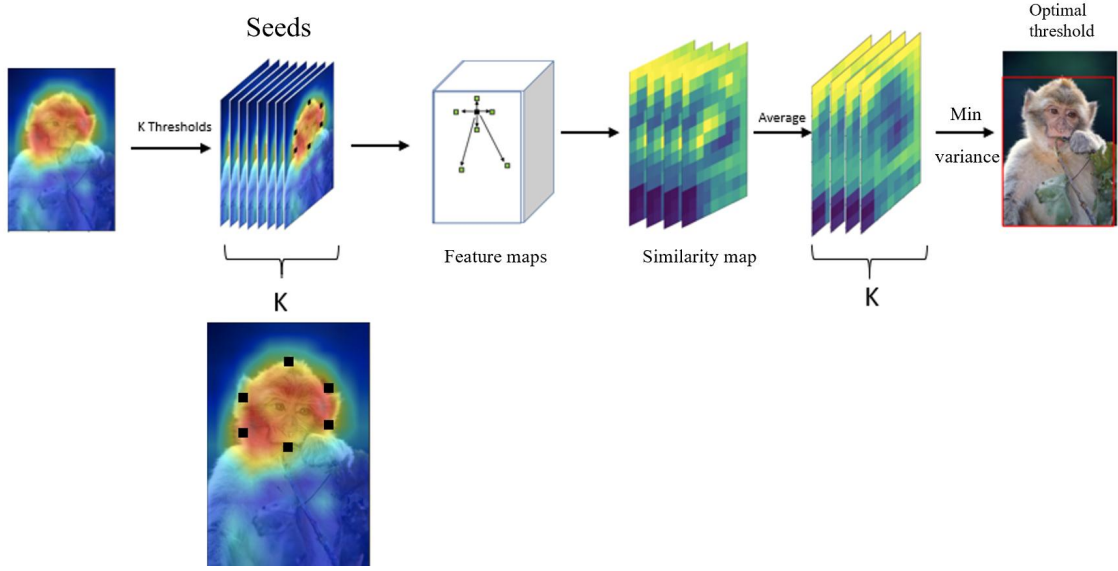


Figure 28: The diagram of proposed two stage localization method.

by the experience, it is decided by how many border points to determine the bounding box. For each seed, we compute the cosine similarity with the rest feature points in the last layer feature maps. The equation to calculate the similarity is shown in Eq.(22).

$$S_{sim} = \frac{A \cdot P_{i,j}}{\|A\| \|P_{i,j}\|} \quad (22)$$

Where $S_{sim} \subseteq R^{H \times W}$ is the similarity map. A is the seed vector. i and j are the position of the rest pixels in the last layer feature map. So for each potential bounding box, we will get n similarity maps, which depends on the seed number. And then we perform spatial average for each position and obtain one similarity map for each bounding box. This similarity map can express the relationship between the bounding box and the rest feature pixels. Considering the property of the object boundary, which is the border of the foreground and background pixels. So the boundary points should be close to both targeted and background pixels. In other words, the best bounding box should have the lowest similarity contrast. We perform variance to measure the contrast of each similarity map of the bounding box. The corresponding

bounding box and the threshold with the minimum contrasted similarity map will be the optimal for this image. The diagram of the proposed two stage localization method is shown in figure 28.

The figure 29 shows the overall processing of the proposed method.

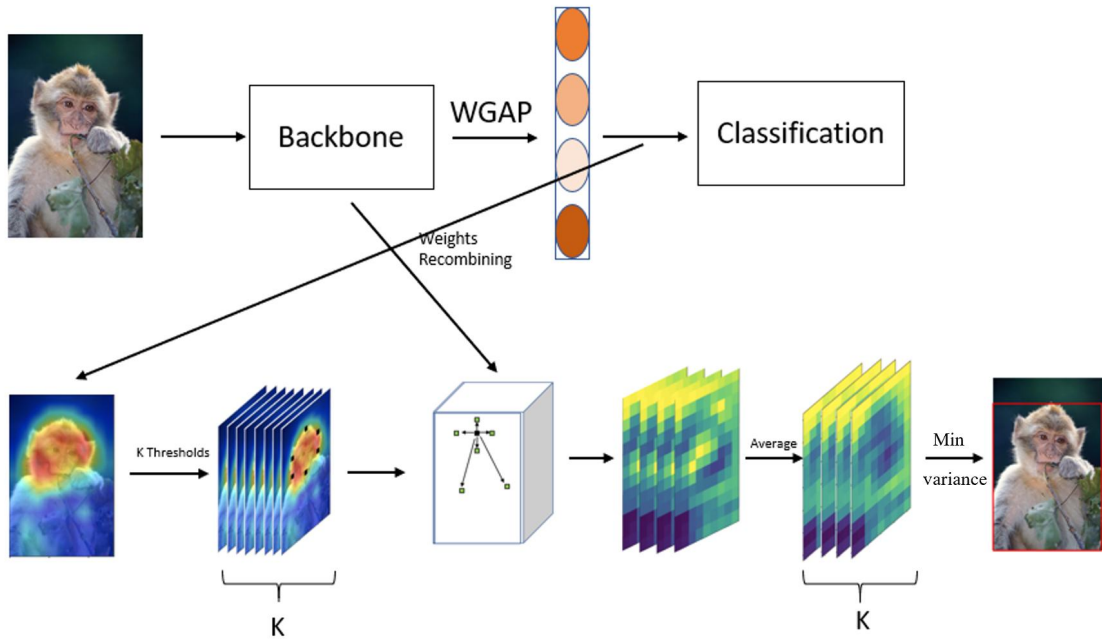


Figure 29: The overall procedure of proposed method.

5.5 Experiment Results

In this section, the experiment results are provided.

5.5.1 Implementation Details

We utilize VGGnet [19] as the backbone network. There is very little hyperparameter that needs to be set. We only select the threshold every 0.05 from the range. The maximum range of the threshold is not fixed either. Because we need at least 4

points to determine a bounding box. When the points of border are less than 4, we will not consider this threshold because there is no bounding box for this threshold. The network is pre-trained on ILSVRC dataset [16] and fine-tuned with learning rate 0.0001 and batch size 32. We use GeForce RTX 2080 GPU to train the model.

5.5.2 Comparison with State-of-the-art Methods

Table 10: Quantitative evaluation performance on CUB-200-2011 dataset.

Method	Top-1 cls.err	Top-1 loc.err
VGGnet-CAM[8]	23.86	66.05
VGGnet-ACoL[12]	28.10	54.08
VGGnet-SPG[11]	24.50	50.00
VGGnet-ADL[9]	34.73	47.64
VGGnet-DANet[10]	24.60	47.48
VGGnet-Cutmix[13]	-	47.47
VGGnet-EMIL[63]	25.23	42.54
VGGnet-MCI[64]	27.41	41.88
VGGnet-ICL[65]	26.6	42.5
VGGnet-RCAM(Ours)	24.06	38.09

We evaluate our method on CUB-200-2011 [15] dataset using VGGnet [19] as backbone. From table 10, we can see that our method significantly improves the localization performance with less classification sacrifice. Figure 30 shows the localization examples. We can see that the Class Activation Map can cover the whole pattern very well and the bounding box covers all the activated regions.

5.6 Ablation Study

We are using the contrast of each similarity map to select the optimal bounding box. The figure 31 expresses a case of that. When the threshold is zero, the bounding box has the same size as the image. 0.1 is the optimal threshold and 0.6 is the maximum

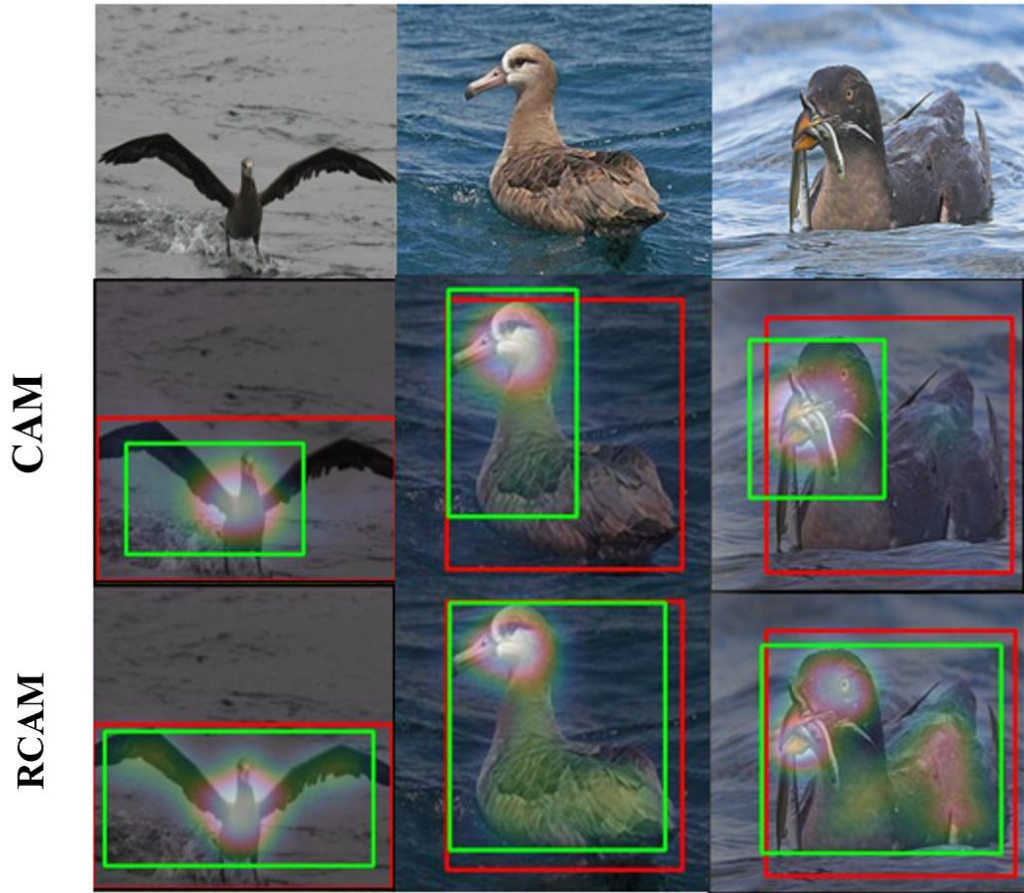


Figure 30: Comparison with CAM on CUB-200-2011 dataset.

threshold for this image. When the threshold is greater than 0.6, there will be less than 4 boundary points, in which the bounding box is nonexistent.

5.7 Conclusion

In this work, we review the three issues that make the CAM ill-posed. Based on the issues, we proposed three corresponding methods to eliminate them. Our proposed two stage localization method does not introduce any hyperparameter that needs to be set by experience. In our perception, our method is the first method that can evaluate the bounding box without Ground-truth label and assign the optimal

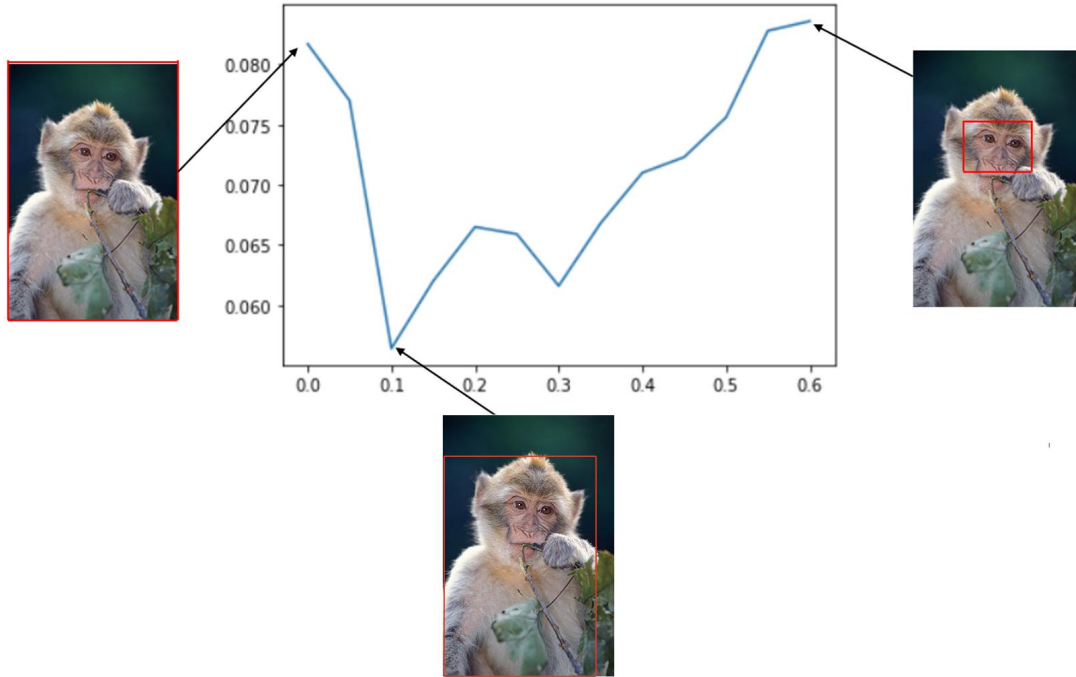


Figure 31: The case of selecting the optimal threshold to extract bounding box.

threshold to different images. The improvement is very significant.

Chapter 6

Summary and Future Work

In this chapter, I will give the overall conclusion of the proposed method and possible future work direction.

6.1 Conclusion

In this thesis, we focus on the Weakly Supervised Object Localization task, which can locate the object using incomplete labels. The most common limitation is that the detector can only highlight the most discriminative part of the target. We believe that it is because classification models only use the most obvious regions to identify the categories of objects.

For the attention-based selection strategy, we design a selection method that can dynamically generate drop masks based on different feature maps. Instead of the current hiding method, our method is smarter and more flexible. In our perception, our method is the first work to consider the different situations of training images. Therefore, our work provides new insights to do the Weakly Supervised Object Localization task.

Based on the first work, it can only hide the most discriminative pixels. Considering the strong relationship of each pixel in a convolutional feature map, we design

a dual hiding method that can remove the regions in both channel and spatial space. According to the experiments, we can see that the proposed method can effectively improve the localization performance.

For the third work, we find some issues that make CAM [8] ill-posed. Based on the issues, corresponding methods are proposed. Our method improves the localization result very significantly without classification sacrifices and hyperparameters to be set by experience. In our perception, the proposed two stage localization method is the first work that can evaluate the potential bounding boxes and assign the optimal thresholds to different images in WSOL tasks.

6.2 Future work

From the results in each section, the current localization accuracy still exist a big gap compared with fully supervised object localization. So in this section, I will provide some potential future work directions.

New loss function

Current WSOL methods highly depend on the classification loss function. However, based on the main work of localization, designing the loss function that is suitable for localization is a possible future direction.

Using stronger transformer

Transformer has attracted more and more attention this year. Therefore, using stronger transformer might be a feasible method.

Bibliography

- [1] T. Lin, P. Dollar, and R. Girshick, “Feature Pyramid Networks for Object Detection,” in *Processing of IEEE Conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.
- [2] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, “Efficient Object Localization Using Convolutional Networks,” in *Processing of IEEE Conference on computer vision and pattern recognition*, 2015, pp.648–656.
- [3] J. Redmon, S. Divvala, R. Girshivk, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *Processing of IEEE Conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [4] J. Caicedo and S. Lazebnik, “Active Object Localization With Deep Reinforcement Learning,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2488–2496.
- [5] M. Lin, Q. Chen and S. Yan, “Network In Network,” *arxiv preprint arXiv:1312.4400*, 2014.
- [6] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-End Training of Deep Visuomotor Policies,” *arxiv preprint arXiv:1504.00702*, 2015.
- [7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *arxiv preprint arXiv:1512.03385*, 2015.

- [8] B. Zhou, A. Khosla, A. Lapedriza, A. Olive, and A. Torralba, "Learning Deep Features for Discriminative Localization," in Processing of IEEE Conference on computer vision and pattern recognition, 2016, pp. 2921–2929
- [9] J. Choe and H. Shim, "Attention-based Dropout Layer for Weakly Supervised Object Localization," in Processing of IEEE Conference on computer vision and pattern recognition, 2019, pp. 2219–2228.
- [10] H. Xue, C. Liu, F. Wan, J. Jiao, X. Ji, and Q. Ye, "DANet: Divergent Activation for Weakly Supervised Object Localization," in Processing of IEEE Conference on computer vision, 2019, pp. 6589–6598.
- [11] X. Zhang, Y. Wei, G. Kang, Y. Yang, and T. S. Huang, "Self-produced Guidance for Weakly-supervised Object Localization," in European Conference on Computer Vision, 2018, pp. 597–613.
- [12] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T. S. Huang, "Adversarial Complementary Learning for Weakly Supervised Object Localization," in Processing of IEEE Conference on computer vision and pattern recognition, 2018, pp. 1325–1334.
- [13] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: Regulation Strategy to Train Strong Classifiers With Localization Feature," in Processing of IEEE Conference on computer vision, 2019, pp. 6023–6032.
- [14] K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a Network to be Meticulous for Weakly-supervised Object and Action Localization," in Processing of IEEE Conference on computer vision, 2017, pp. 3544–3553
- [15] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Computation and Neural Technical Report, 2011.

- [16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. F. Fei, "Imagenet Large Scale Visual Recognition Challenge," in *International Journal of Computer Vision*, 2015.
- [17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, and S. Reed, "Going Deeper with Convolutions," in *Processing of IEEE Conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [18] Z. Zhang and T. D. Bui, "Attention-based Selection Strategy for Weakly Supervised Object Localization," *IEEE International Conference on Pattern Recognition*, 2020.
- [19] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arxiv preprint arXiv:1409.1556*, 2014.
- [20] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation Networks for Object Detection," in *Processing of IEEE Conference on computer vision and pattern recognition*, 2018, pp. 3588-3597.
- [21] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "DetNet: Design Backbone for Object Detection," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 334-350.
- [22] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully Convolutional One-Stage Object Detection," in *Proceedings of the International Conference on Computer Vision*, 2019, pp. 9627-9636.
- [23] X. Zhou, J. Zhuo, and P. Krahenbuhl, "Bottom-Up Object Detection by Grouping Extreme and Center Points," in *Processing of IEEE Conference on computer vision and pattern recognition*, 2019, pp. 850-859.

- [24] C. Liu, L. Chen, F. Schroff, H. Adam, W. Hua, A. L. Yuille, and L. Fei-Fei, "Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation," in Proceedings of the International Conference on Computer Vision, 2019, pp. 82-92.
- [25] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," arxiv preprint arXiv:1706.05587, 2017.
- [26] Z. Zhou, M.M.R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A Nested U-Net Architecture for Medical Image Segmentation," in Proceedings of Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, 2018, pp. 3-11.
- [27] F. Milletari, N. Navab, and S. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," arxiv preprint arXiv:1606.04797v1, 2016.
- [28] Z. Zhou, "A brief introduction to weakly supervised learning," in Proceedings of National Science Review, 2018, pp. 44-53.
- [29] H.B. Barlow, "Unsupervised Learning," in Proceedings of Neural Computation, 2018, pp. 295-311.
- [30] J. C. Caicedo and S. Lazebnik, "Active Object Localization With Deep Reinforcement Learning," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2488-2496.
- [31] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual Attention Network for Image Classification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3156-3164.

- [32] D. Lu and Q. Wang, "A survey of image classification methods and techniques for improving classification performance," in Proceedings of International Journal of Remote Sensing , 2007, pp. 823-870.
- [33] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of Tricks for Image Classification with Convolutional Neural Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition , 2019, pp. 558-567.
- [34] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "CNN-RNN: A Unified Framework for Multi-Label Image Classification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition , 2016, pp. 2285-2294.
- [35] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization," in Proceedings of the IEEE Conference on Computer Vision , 2017, pp. 618-626.
- [36] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks," in Proceedings of the IEEE Winter Conference on Applications of Computer Vision , 2018, pp. 839-847.
- [37] D. Omeiza, S. Speakman, C. Cintas, and K. Weldermariam, "Smooth Grad-CAM++: An Enhanced Inference Level Visualization Technique for Deep Convolutional Neural Network Models," arxiv preprint arXiv: arXiv:1908.01224, 2019.
- [38] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, Piotr Mardziel, and X. Hu, "Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural

- Networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 24-25.
- [39] S. Yang, Y. Kim, Y. Kim, and C. Kim, ”Combinational Class Activation Maps for Weakly Supervised Object Localization ,” in Proceedings of the IEEE Winter Conference on Applications of Computer Vision , 2020, pp. 2941-2949.
- [40] I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. MIT press, 2016.
- [41] Y. LeCun, B. Boser, J. S. Denker, and D. Henderson, ”Combinational Class Activation Maps for Weakly Supervised Object Localization,” in Proceedings of Neural Computation , 1989, pp. 541-551.
- [42] P. C. Ng and S. Henikoff, ”SIFT: predicting amino acid changes that affect protein function,” in Proceedings of Nucleic Acids Research, 2003, pp. 3812–3814.
- [43] X. Wang, T. X. Han, and S. Yan, ”An HOG-LBP human detector with partial occlusion handling,” in Proceedings of International Conference on Computer Vision, 2009, pp. 32–39.
- [44] K. Krishna, and M. N. Murty, ”Genetic K-means algorithm,” in Proceedings of IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 1999, pp. 433-439.
- [45] V. Nair and G. E Hinton, ”Rectified linear units improve restricted boltzmann machines,” in Proceedings of the 27th International Conference on Machine Learning, 2010, pp. 807–814.
- [46] D. Gao and Y. Ji, ”Classification methodologies of multilayer perceptrons with sigmoid activation functions,” in Proceedings of Pattern Recognition, 2005, pp. 1469-1482.

- [47] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov., "Dropout: a simple way to prevent neural networks from overfitting," *Journal of machine learning research*, 2014, pp. 1929-1958.
- [48] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient Object Localization Using Convolutional Networks," in *Processing of IEEE Conference on computer vision and pattern recognition*, 2015, pp. 648–656.
- [49] S. Park and N. Kwak, "Analysis on the Dropout Effect in Convolutional Neural Networks," in *Asian Conference on Computer Vision*, 2016, pp. 189–204.
- [50] G. Ghiasi, T. Lin, and Q. V. Le, "DropBlock: A regularization method for convolutional networks," *Advances in Neural Information Processing Systems*, 2018, pp. 10727–10737.
- [51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Conference on Neural Information Processing System*, 2017.
- [52] J. Fu, H. Zheng, and T. Mei, "Look Closer to See Better: Recurrent Attention Convolutional Neural Network for Fine-Grained Image Recognition," in *Processing of IEEE Conference on computer vision and pattern recognition*, 2017, pp. 4438–4446.
- [53] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, and Z. Zheng, "Multiple Granularity Descriptors for Fine-Grained Categorization," in *Processing of IEEE Conference on computer vision and pattern recognition*, 2015, pp. 2399–2406.
- [54] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," in *International Conference on Machine Learning*, 2015.

- [55] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "AttenGAN: Fine-Grained Text to Image Generation With Attentional Generative Adversarial Networks," in Processing of IEEE Conference on computer vision and pattern recognition, 2018, pp. 1316–1324.
- [56] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding," in AAAI Conference of Artificial Intelligence, 2018, pp. 5446–5455.
- [57] H. Zhang, L. Goodfellow, D. Metaxas, and A. Odena, "Self-Attention Generative Adversarial Networks," arxiv preprint arXiv:1805.08318, 2019.
- [58] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation Networks," in Processing of IEEE Conference on computer vision and pattern recognition, 2018, pp. 7132–7141.
- [59] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in European Conference on Computer Vision, 2018, pp. 3–19.
- [60] J. Choe, S. J. Oh, S. Lee, S. Chun, Z. Akata and H. Shim, "Evaluating Weakly Supervised Object Localization Methods Right," in Processing of IEEE Conference on computer vision and pattern recognition, 2020, pp. 3133-3142.
- [61] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang and H. Lu, "Dual Attention Network for Scene Segmentation," in Processing of IEEE Conference on computer vision and pattern recognition, 2019, pp. 3146-3154.
- [62] Z. Zhang and T. D. Bui, "Attention-based Dual Hiding Method for Weakly Supervised Object Localization," submitted to ICIP 2021.

- [63] J. Mai, M. Yang and W. Luo, "Erasing Integrated Learning : A Simple yet Effective Approach for Weakly Supervised Object Localization," in Processing of IEEE Conference on computer vision and pattern recognition, 2020, pp. 8766-8775.
- [64] S. Babar and S. Das, "Where to Look?: Mining Complementary Image Regions for Weakly Supervised Object Localization," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 1010-1019.
- [65] M. Ki, Y. Uh, W. Lee and H. Byun, "In-sample Contrastive Learning and Consistent Attention for Weakly Supervised Object Localization," in Proceedings of the Asian Conference on Computer Vision, 2020.
- [66] Z. Zhang, "Revisiting Class Activation Map: Two Stage Method for Weakly Supervised Object Localization," Preparing to NeurIPS, 2021.