Image Co-localization on Unseen Object

Guofeng Xu

U5491523

Supervised by Dr. Weihao Li

June 2020

A thesis submitted in part fulfilment of the degree of

Master of Machine Learning and Computer Vision

The College of Engineering and Computer Science

Australian National University

Abstract

The problem of image co-localization is to identify and localize object from the same category that are included in each set of images. In this work, we propose a simple yet effective approach to localize common objects across a set of distinct images. We solve this problem by modularizing different members in class activation map (CAM) family, which can be applied on deep descriptor transforming (DDT) and common component activation map (CCAM). Different module combinations are explored in this thesis to tackle the problem. The method proposed in this thesis enables a pretrained network to localize common objects which are 'unseen' and proposal free. With an average Corloc evaluation being 0.603, our best approach – Weighted-DDT achieves an outstanding result on image co-localization task.

Acknowledgements

I would like to express my gratitude for all the guidance and assistance from my supervisor Dr. Weihao Li. Your insights and advices on this topic are valuable and help me save large amounts of time.

I would like to thank Mr. Shu Liu as for his technical support and computer vision research experience on this project. We worked together on this project and made a great progress.

Also, I would like to thank Mr. Yuxuan Long for his mathematical viewpoint on CAM and Mr. Haolei Ye for his GPU server resources.

Finally, the endless support and encourage from my family are greatly appreciated during this journey.

Table of Contents

Abstract 2

Acknowledgements 3

List of Figures 6

List of Tables 8

Notations 9

1. Introduction 10

1.1. Contributions 11

1.2. Outline 11

2. Background 12

2.1. Weakly-supervised Object Localization 12

2.2. Object Co-localization 13

3. Method 14

3.1. Class Activation Map (CAM) Family 14

3.1.1. Original CAM 14

3.1.2. A Generalization of CAM 14

3.1.3. Grad-CAM and Grad-CAM++ 17

3.2. Common Component Activation Map with Gradient Based CAM 18

3.3. Improving Deep Descriptor Transforming 19

3.4. The Architecture of Our Method 21

4. Experiment 22

4.1. Dataset 22

4.1.1. How to Get the Valid Dataset 22

4.1.2. Statistical Characteristics of the Dataset 25

4.2. Generating Bounding Boxes 25

4.3. Evaluation Metric 26
5. Results and Discussion 26
5.1. Visualization 27
5.2. Ablation Study 28
5.2.1. VGG19 vs. AlexNet 28
5.2.2. Grad CAM vs. Grad CAM++ 29
5.2.3. DDT vs. CCAM 30
5.2.4. Weighted-DDT vs. DDT 33
5.3. Error Analysis 34
5.3.1. Weighted DDT model 35
5.3.2. Confusion Matrix Analysis 41
6. Conclusion 43
Bibliography 44

List of Figures

Figure 1. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation mas. The CAM highlights the class-specific discriminative regions.[5] 12

Figure 2. Overall architecture of the proposed method. 21

Figure 3. Imagenet subtree 23

Figure 4. URL samples of chipmunk category without image index 23

Figure 5. URL samples of chipmunk category with image index 24

Figure 6. Sample image downloaded from Imagenet with a rename by its index and bounding box labels. 24

Figure 7. Successful visual examples of object co-localization on the six ImageNet Subsets. Each class contains four different scenarios. In these images, the red rectangle is the ground truth bounding box, and the green rectangle is the prediction by Weighted-DDT. (Best viewed in colour and zoomed in). 27

Figure 8. Left: original image; Middle: predicted bounding box (green) of VGG19-based model; Right: predicted bounding box (green) of AlexNet-based model. It is noted AlexNet-based method have high response on background. 28

Figure 9. Comparison of area rates between VGG-based and AlexNet-based method. 29

Figure 10. Left: original image; Middle: predicted bounding box (green) of Grad-CAM++ based model; Right: predicted bounding box (green) of Grad-CAM based model. Grad-CAM results in abnormal heatmaps that are even opposite to left one, as some unnecessary feature maps influence the result. Such problem is handled by Grad-CAM++. 30

Figure 11. Left: Original image; Middle:predicted bounding box (green) by CCAM; Right: predicted bounding box (green) by DDT. Both of them have a Corloc larger than 0.5 in this case. 31

Figure 12. Left: Original image; Middle:predicted bounding box (green) by CCAM; Right: predicted bounding box (green) by DDT.In this case. CCAM outperforms DDT. 32

Figure 13. Left: Original image; Middle:predicted bounding box (green) by CCAM; Right: predicted bounding box (green) by DDT.In this case. DDT outperforms CCAM. 33

Figure 14: Left: heatmap generated from adopted features (474 for rhino and 479 for wheelchair); Right: heatmap generated from abandoned features (38 for rhino and 33 for wheelchair). 34

Figure 15. Visualizations on error type 1 with IOU on each image. 35

Figure 16. Visualizations on error type 2 with IOU on each image. 36

Figure 17. Visualizations on error type 3 with IOU on each image. 36

Figure 18. Visualizations on error type 4 with IOU on each image. 37

FIGURE 19. VISUALIZATIONS ON ERROR TYPE 5 WITH IOU ON EACH IMAGE. 38
FIGURE 20. VISUALIZATIONS ON ERROR TYPE 6 WITH IOU ON EACH IMAGE. 38
FIGURE 21. AN ILLUSTRATION OF DIFFERENT SHAPES OF RAKE IMAGES..... 39
FIGURE 22. VISUALIZATIONS ON ERROR TYPE 7 WITH IOU ON EACH IMAGE. 39
FIGURE 23. VISUALIZATIONS ON ERROR TYPE 8 WITH IOU ON EACH IMAGE. 40
FIGURE 24. CONFUSION MATRIX REPRESENTATIONS ON AN IMAGE..... 41
FIGURE 25. VISUALIZATION OF CONFUSION MATRIX OVER 6 CATEGORIES 41

List of Tables

TABLE 1. SOURCE OF FEATURE MAP & WEIGHT OF CAM FAMILY MEMBERS 18
TABLE 2. FUNCTIONALITY OF CAM FAMILY MEMBERS 18
TABLE 3. STATISTICS OF THE DATASET 25
TABLE 4. CORLOC METRIC ON VARIOUS METHODS IMPLEMENTED IN THE EXPERIMENT ON IMAGE SETS DISJOINT WITH IMAGENET. 26
TABLE 5. COMPARISON OF THE CORLOC METRIC ON TWO DIFFERENT MODELS IN TERMS OF VGG19 VS. ALEXNET. 28
TABLE 6. COMPARISON OF THE CORLOC METRIC ON TWO DIFFERENT MODELS IN TERMS OF GRAD CAM VS. GRAD CAM++ 29
TABLE 7. COMPARISON OF THE CORLOC METRIC ON TWO DIFFERENT MODELS IN TERMS OF GRAD DDT VS. CCAM 30
TABLE 8. COMPARISON OF THE CORLOC METRIC ON TWO DIFFERENT MODELS IN TERMS OF WEIGHTED DDT VS. DDT 33
TABLE 9. COMPARISON OF THE CORLOC METRIC ON TWO DIFFERENT MODELS IN TERMS OF GRAD-CCAM++ VS. WEIGHTED-DDT 34

Notations

In this thesis, the following notations are used: $\boldsymbol{A}$ is a matrix; a is a scalar; $\boldsymbol{a}$ is a vector; $\boldsymbol{A^{(i)}}$ is the i^{th} column of $\boldsymbol{A}$; $(*)^T$ is the transpose; $\|\boldsymbol{A}\|_F$ is the Frobenius form of $\boldsymbol{A}$; $\|\boldsymbol{a}\|_p$ is the p-norm of $\boldsymbol{a}$; $\boldsymbol{A}_{MN}$ is the $M \times N$ dimensional matrix; $\overline{\boldsymbol{A}}$ is a the mean of $\boldsymbol{A}$.

1. Introduction

Over the past few years, deep learning has become the state-of-the-art method for a variety of vision recognition problems, e.g. image classification [1, 2], object detection [3], and object localization [4-6]. Based on the unprecedented success of neural networks, the objective of this thesis is to use pretrained CNN networks to deal with the problem of image co-localization (also refers to object co-localization).

Object localization is an interesting and fundamental problem in computer vision. It aims to recognize the main (or most visible) object in the image and locate it with an axis-aligned bounding box [2]. Recent work from [7, 8] although achieve state-of-the-art performance on object localization, they are trained with fully labeled bounding box annotations. Depending on the scale of the dataset, such annotated bounding box can be very expensive. Therefore, the increasing cost of manual labeling leads to the development of weakly supervised learning models [9-12] on object localization. These weakly supervised object localization (WSOL) methods have attracted more attention since they only require image-level category labels. Recently, one simple but effective method is to use Class Activation Maps (CAM) [5] to generate class-specific localization maps. Followed by [13], global average pooling (GAP), which was initially designed as a structural regularizer to prevent overfitting and reduce parameters, it can also retain remarkable localization ability. Therefore, despite being trained for image classification, this specific CAM structure network can localize the discriminative image regions. However, this CAM structure cannot work on networks without GAP layer [14]. Inspired from this work, Gradient Class Activation Map (Grad-CAM) [14] and Grad-CAM++ [15] were developed, and these CAM related work are summarized as CAM family in this thesis.

Common object localization also named as object co-localization aims to localize common objects of the same class over a set of (two or more) distinct images [4]. The most recent work from [16] and [17] proposed common component activation map (CCAM) and deep descriptor transforming (DDT), where both methods utilize the convolutional feature maps as evidence to predict the common components in the image set. In this work, we propose a method in generating visual explanations of common object by ‘weight-and-feature’ combination. More specifically, we employ different modules in CAM family (where

weights come from) on CCAM and DDT (where features get processed) to build an effective method on image co-localization.

1.1. Contributions

Our main contribution is threefold. Firstly, we extend the traditional CAM and propose the idea of CAM family that clearly bridges the relationship among CAM, Grad-CAM and Grad-CAM++. All members in CAM family are modularized and are able to work in any CNN-based network. Secondly, we employ different modules from CAM family on CCAM and DDT to build a 'weight-and-feature' solution for handling image co-localization problem. Lastly, we conduct thorough ablation studies with observations from both statistical and spatial perspectives. Such experiments demonstrate the superiority of Weighted-DDT on image co-localization task.

1.2. Outline

The remaining part of the thesis is organized as follows: Section 2 reviews and discusses the background of image co-localization. The literature of recent object co-localization works is also presented. Section 3 presents the main methods and techniques that are used in our model. In Section 4, a detailed description of the experimental process is clearly demonstrated. Section 5 provides the image co-localization results under various scenarios using the methods we discussed in Section 4. Thorough comparisons between different influential factors are also presented via visual explanations. An error analysis is evaluated and conducted as well. Lastly, the conclusion and future work is shown in Section6.

2. Background

Our work draws on recent work in weakly-supervised object localization, and object co-localization.

2.1. Weakly-supervised Object Localization

Image co-localization shares great similarities with weakly-supervised object localization (WSOL) [9-12, 18, 19] as both try to localize the same type of objects within an image set. However, while WSOL requires manually labeled negative images, image co-localization does not. Bergamo et al [20] proposed a self-taught object localization method. It masks out image regions where the maximal activation can be caused to localize objects. Cinbis [21] and Pinheiro [22] combined CNN features and multiple instance learning to localize object accurately. However, these approaches were not trained end-to-end and therefore very difficult to generalize on real-world dataset. To tackle this problem, Zhou [5] provided another view of localizing objects from CNN visualization perspective and did not require bounding box annotations. To explore visual explanations from deep convolutional neural network, Zhou [2] employed global average pooling on the convolutional feature maps to generate class activation map (CAM), which identifies class-specific discriminative regions.

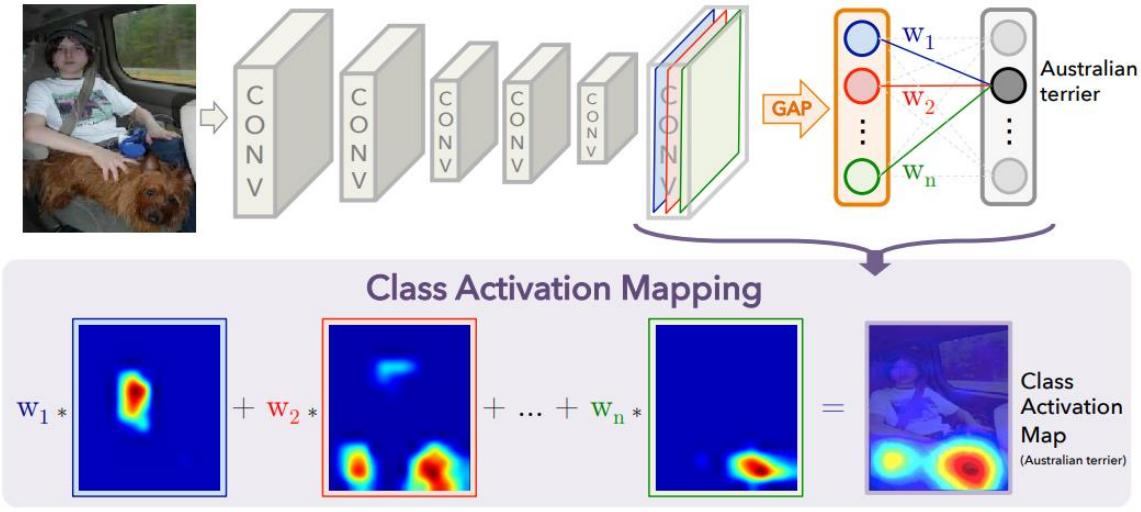


Figure 1. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation mas. The CAM highlights the class-specific discriminative regions.[5]

To better explain the mechanics, as illustrated in *Figure 1*, GAP outputs the spatial average of the feature map of each channel from the last convolutional layer. Then, a weighted sum of these spatial average values can be computed to generate the final output (classification score). Similarly, a weighted sum of the feature maps of the last convolutional layer can also be computed to show the class-specific image regions, which is called class activation map from [5]. By doing this, the class activation map can identify the class-specific discriminative regions of the image by projecting back the weights of the output layer onto the convolutional feature maps.

In [5], CAM modifies AlexNet and VGGnet by removing fully-connected layers before the final output and replace them with GAP layer and a softmax layer. It requires feature maps to precede softmax layers directly, and hence it is only applicable to a particular type of CNN architectures [2] to perform GAP over convolutional maps. Thus, such architecture (i.e. conv feature maps → global average pooling → softmax layer) is strictly required for CAM. Therefore, CAM cannot be applied directly on the original AlexNet or VGGnet as they have no GAP structure.

To overcome this drawback, CAM was further modified and improved to Grad-CAM in [14], allowing a more diverse CNN models and tasks. Based on Grad-CAM, Aditya *et al* [15] proposed a generalized method called Grad-CAM++ that can create a better high-resolution class-discriminative visualization for object localization.

2.2. Object Co-localization

There emerged several object co-localization methods in recent years. Tang *et al* [4] formulated and relaxed co-localization as a convex optimization problem. After that, a common object detector method for object co-localization problem was proposed by Li and Liu [23] . The detector can be learnt by marking its detection confidence scores based on a strongly supervised detector. Then, a conditional random field (CRF) model was adopted to refine the co-localization result. Le [24] extracted category-consistent CNN features followed by an activation co-propagation process to localize the common objects. Motivated by class activation mapping, Li [16] extended CAM on a series of images and is so-called common component activation map (CCAM). By using CCAM, it allows for generating activation map for unseen object and can decompose its neural activations of

the common novel object into semantically interpretable components with several pretrained object categories. This CCAM approach is not restricted in localizing pre-defined object categories. More importantly, it achieves the highest performance based on pre-trained AlexNet on six subsets of ImageNet which are not included in the ILSVRC.

3. Method

To localize unseen objects, we firstly review the development of different CAM related works and propose a basic CAM module for being compared with Grad-CAM and Grad-CAM++ on arbitrary CNN-based network structures. We then design the variants of common component activation map (CCAM) with different members in CAM family. Secondly, we review DDT and combine it with CCAM through weighted PCA.

3.1. Class Activation Map (CAM) Family

We name all CAM related work as 'CAM family' in this thesis, and present different members in CAM family in this section.

3.1.1. Original CAM

Class activation map (CAM) [5], as the first member in CAM family, scalarizes the feature maps by global average pooling (GAP), followed by a weight provided for each feature map. All pixels on one certain feature map share the same weight. These feature-wise weights indicate the importance rate of the corresponding feature map across all maps.

Hence, for the CNN-based models that do not include GAP, original CAM cannot work on these models and therefore promote the emerge of Grad-CAM. To fill this gap, we reformulate CAM and extend it to a general form that does not rely on GAP.

3.1.2. A Generalization of CAM

For a general CNN-based architecture without GAP (e.g. AlexNet, VGGnet), the final classification score Y^c for class c can be expressed as a linear combination of its last convolutional feature map $\boldsymbol{A}_{ij}^k$ and corresponding pixel-wise weights φ_{ij}^k:

$$Y^c = \sum_k \sum_i \sum_j [\varphi_k^c]_{ij} \boldsymbol{A}_{ij}^k \tag{1}$$

where (i, j) and k denotes the spatial location and number of feature maps respectively.

Consider the flattening of feature map as a vectorization operation, the formula above can be rewritten as the most general form:

$$Y^c = \sum_{k=1}^{K} vec\left([\varphi_k^c]_{ij}\right)^T vec\left(\boldsymbol{A}_{ij}^k\right) \tag{2}$$

where $vec\left([\varphi_k^c]_{ij}\right) = \begin{bmatrix} [\varphi_1^c]_{ij} \\ \dots \\ [\varphi_k^c]_{ij} \end{bmatrix} \in R^{KxZ}$, $vec(\boldsymbol{A}) = \begin{bmatrix} vec(A^1) \\ \dots \\ vec(A^k) \end{bmatrix}$ and Z represents the number of pixels in the feature map (or $Z = \sum_i \sum_j \mathbf{1}$). Here Y^c still reserves as a scalar. We then compare (2) with the expression in [5] which shows the formation of classification score Y^c as

$$Y^c = \sum_{k=1}^{K} w_k^c \frac{1}{Z} \sum_i \sum_j \boldsymbol{A}_{ij}^k \,, \tag{3}$$

where w_k^c denotes the class- feature weights.

It also can be written in the vectorized form:

$$\begin{aligned} Y^c &= \sum_{k=1}^{K} vec\left(\begin{bmatrix} w_1^c \, \mathbf{1}^T \\ \dots \\ w_k^c \, \mathbf{1}^T \end{bmatrix}\right)^T vec\left(\boldsymbol{A}_{ij}^k\right) \\ &= \sum_{k=1}^{K} vec\left(\begin{bmatrix} w_1^c \\ \dots \\ w_k^c \end{bmatrix} \mathbf{1}^T\right)^T vec\left(\boldsymbol{A}_{ij}^k\right), \end{aligned} \tag{4}$$

where $\mathbf{1}^T \in R^{Z \times 1}$.

Denote $w^c = [w_1^c, w_2^c, \dots, w_k^c] \in \boldsymbol{R}^k$, by comparing (2) and (4), we can derive the optimal weighs $\boldsymbol{w}_{optim}^c$ in (1) by constructing a convex optimization problem:

$$\boldsymbol{w}_{optim}^{c} = argmin_{\boldsymbol{w}} \left\| vec\left(\begin{bmatrix} w_1^c \\ \dots \\ w_k^c \end{bmatrix} \mathbf{1}^T \right) - vec\left([\varphi_k^c]_{ij} \right)^T \right\|_F^2$$

$$= \frac{1}{k} vec\left(\sum_k [\varphi_1^c]_{ij} \quad \dots \quad \sum_k [\varphi_k^c]_{ij} \right) \tag{5}$$

To make it clear, for the specific n^{th} feature map on class *c*, the optimal weight is

$$w_n^c = \frac{1}{k} \sum_{n=1}^{k} [\varphi_n^c]_{ij} \tag{6}$$

This result from (6) demonstrates that the optimal weights of CAM for network without GAP are equivalent to the average values of last convolutional layer weights. It further indicates that by substituting the GAP weights with the average of pixel-wise weights in each feature map, which enables CAM for an arbitrary network.

At the same time, the proposed method in [5] limits the application range of CAM in a network with single fully connected layer, while its functionality can be reserved through multiple linear mapping operations. Followed by our pervious notation, $\boldsymbol{A}_{ij}^k$ denotes the output feature maps of the last convolution layer, Y_c denotes the classification score, we can have the following expression for a network with n fully connection (e.g. $conv \rightarrow fc_n \rightarrow fc_{n-1} \rightarrow \cdots \rightarrow fc_1 \rightarrow Y_c$).

$$Y^c = \sum_k \sum_i \sum_j \left(\prod_n w_n^c \right) vec\left(\boldsymbol{A}_{ij}^k \right), \tag{7}$$

where w_n^c represents the weight of n^{th} fully connected layer on class *c*. Recall (4) and (5), its class activation map can be expressed based on the formula in [14]. Here, we assume $\boldsymbol{\varphi}_k^c = vec(\sum_k [\varphi_1^c]_{ij} \quad \dots \quad \sum_k [\varphi_k^c]_{ij})$ to be the weight of the fully connected layer after convolution, class activation map $M_c(i,j)$ is given by:

$$M_c(i,j) = \sum_k w_k^c \, \boldsymbol{A}_{ij}^k$$

$$= \sum_k \left(\prod_{n-1} w_{n-1}^c \right) \frac{1}{k} vec \left(\sum_k [\varphi_1^c]_{ij} \quad \cdots \quad \sum_k [\varphi_k^c]_{ij} \right) \boldsymbol{A}_{ij}^k$$

$$= \frac{1}{k} \sum_k \left(\prod_{n-1} w_{n-1}^c \right) \left[\sum_k [\varphi_1^c]_{ij} \quad \cdots \quad \sum_1 [\varphi_k^c]_{ij} \right] \boldsymbol{A}_{ij}^k$$

$$= \frac{1}{k} \sum_k \left(\prod_{n-1} w_{n-1}^c \right) \boldsymbol{\varphi}_k^c \boldsymbol{A}_{ij}^k \tag{8}$$

In this case, such CAM can be applied on any convolutional network which employs fully connected layer (could be multiple *fc* layers) after the convolutional feature maps.

3.1.3. Grad-CAM and Grad-CAM++

Grad-CAM [14], as another member in CAM family, also takes the feature map of the last convolutional layer to acquire class activation map by calculating the weighted sum over these feature maps. The difference from Grad-CAM to CAM is that Grad-CAM utilizes the partial derivative of the prediction with respect to the feature map as the weight ($\frac{\partial Y^c}{\partial A}$). It shows the advantage on the model with multiple fully connected layer, as it avoids calculating the cumulative products over the weights (e.g. (formula 7)), while it requires more time cost due to the processing of backpropagation. On the other hand. Grad-CAM considers pixel-wise value of the feature map apart from the forward weights. For instance, if a pixel has a value of zero and its weight is not zero, Grad-CAM will assign a zero weight to this pixel as its partial derivative is zero. On the contrary, CAM will incorrectly give it a non-zero value since the pixel-wise value of the feature is not considered. Grad-CAM++ [15] provides further improvement based on Grad-CAM by scaling the weight ($\frac{\partial Y^c}{\partial A}$) with the number of effective pixels that have non-zero values in the corresponding feature map. In other words, they introduce weighting coefficients for the pixel-wise gradients for class *c* and convolutional feature map $\boldsymbol{A}^k$. More mathematical demonstration will be presented in Section 3.2.

In summary, the three members in CAM family apply the similar idea by taking a linear combination of feature maps and feature-wise weights. They share the same feature maps, while the corresponding weights are different. The details are listed in *Table 1*.

Table 1. Source of feature map & weight of CAM family members

	CAM	Grad CAM	Grad CAM++
Feature map	Output of last Conv	Output of last Conv	Output of last Conv
Weight	Weight of FC	Gradient of feature map	Scaled gradient of feature map

Such different weights enable different functionalities for the members in CAM family, which are summarized in *Table 2*.

Table 2. Functionality of CAM family members

	CAM	Grad CAM	Grad CAM++
Network structure unrestricted	N*	Y	Y
Involve pixel-wise value in feature map	N	Y	Y
Involve effective pixel in feature map	N	N	Y

* unrestricted CAM can be realized by (7)

3.2. Common Component Activation Map with Gradient Based CAM

Followed from [16], common component activation map (CCAM) is a visualization method in which the class-specific activation maps are regarded as components to discover the common components in the image set. To generate the activation map for unseen object, we treat the classification score Y from CAM as a component vector for input image. Thus, for a given group of N images $\mathrm{I} = [I_1, \dots, I_N]$ containing an unseen object, we can obtain the common component of the group by averaging output vectors $[Y_1, \dots, Y_N]$ over a group of images on unseen objects . Such average output vector represents a statistical

characteristic across the group and scales the magnitude of activation map generated by the CAM. It therefore works on localizing unseen objects with such statistical information.

$$G = \frac{1}{N}\sum_{i} Y_i \tag{9}$$

Given the vector G, its length equals the number of target class. We treat $P(G)$ as a set of indices of the c largest entries that are also regarded as the c most likely target classes. For each image $I_i \in$ I, we can generate a weighted sum of feature maps to get the CCAM based on the top c components in $P(G)$ [16].

$$M_c(i,j) = \sum_{c \in P(G)} G_p \sum_{k} W_k^c A_{ij}^k\,, \tag{10}$$

where A_{ij}^k represents the feature map of the last convolutional layer for I, while W_k^p denotes the corresponding weight for feature map that aims to calculate Y_c.

To allow CCAM work on other members in CAM family, we replace the weight for CAM in (9) with Grad-CAM and grad-CAM++. Their expressions are shown as below:

$$M_c(i,j) = \sum_{c \in P(G)} G_c \left(\sum_{i}\sum_{j} \frac{\partial Y^c}{\partial A_{ij}^k}\right) A_{ij}^k \tag{11}$$

$$M_c(i,j) = \sum_{c \in P(G)} G_c \sum_{i}\sum_{j} \alpha_{ij}^{kc}.relu\left(\frac{\partial Y^c}{\partial A_{ij}^k}\right) A_{ij}^k\,,$$

$$\alpha_{ij}^{kc} = \begin{cases} \dfrac{1}{\sum_{l,m} \dfrac{\partial Y^c}{\partial A_{lm}^k}}, & \dfrac{\partial Y^c}{\partial A_{ij}^k} = 1 \\ 0, & otherwise \end{cases}, \tag{12}$$

where (11) and (12) are for Grad-CAM based CCAM and Grad-CAM++ based CCAM, which we term them as Grad-CCAM and Grad-CCAM++ respectively.

3.3. Improving Deep Descriptor Transforming

Deep descriptor transforming (DDT) [17] is different from CCAM in two aspects. Firstly, DDT only takes feature maps from the last convolutional layer without considering their

weights for predication. These (optimal) weights are quite important in both Grad-CAM and Grad-CAM++ according to (11) and (12). Secondly, DDT manipulates the feature map based on principal component analysis (PCA) where PCA is utilized as projection direction to transform the feature maps. Since such operation extracts statistical information over the feature maps, it is expected that DDT can present the heatmap in a different vector space from that based on pure feature maps. To combine the advantage of both CCAM and DDT against a dataset consists of unseen object c, we take weight w_k^c for k^{th} feature map which is generated in CAM family for processing a weighted PCA [25] based on DDT method.

The calculation of the weighted mean feature map is given as,

$$\overline{\boldsymbol{A}^c} = \frac{1}{L}\sum_k \sum_{i,j} w_k^c \boldsymbol{A}_{i,j}^k \,, \tag{13}$$

where $L = k \times i \times j$. We then obtain the covariance matrix with the weighted feature map.

$$Cov(\boldsymbol{A}) = \frac{1}{L}\sum_k \sum_{i,j} \left(w_k^c \boldsymbol{A}_{i,j}^k - \overline{\boldsymbol{A}}\right)\left(w_k^c \boldsymbol{A}_{i,j}^k - \overline{\boldsymbol{A}}\right)^T \tag{14}$$

After getting the eigenvectors $\xi_1, \ldots \xi_k$ corresponding to the sorted eigenvalues $\lambda_1 \geq \ \ldots \geq \lambda_k \geq 0$ of $Cov(\boldsymbol{A})$, similar with DDT, we only take the largest eigenvector for calculating the first principal component $M_{(i,j)}$:

$$M_{(i,j)} = \boldsymbol{\xi}_1^T \left(w_k^c \boldsymbol{A}_{i,j}^k - \overline{\boldsymbol{A}}\right) \tag{15}$$

Based on their spatial locations, all $M_{(i,j)}$ from the image are formed into a 2-D matrix with a dimension of $i \times j$. We can call this matrix as indicator matrix as suggested in [17].

$$M = \begin{bmatrix} M_{(1,1)} & \cdots & M_{(1,j)} \\ \vdots & \ddots & \vdots \\ M_{(i,1)} & \cdots & M_{(i,j)} \end{bmatrix} \in \mathcal{R}^{i \times j} \tag{16}$$

The positive (negative) value in M indicates the positive (negative) correlations of feature maps. Thus, we can treat each of these values as the indicator for capturing the corresponding pixels from the original input image. As for image co-localization, the corresponding positive values of M can reflects where the common object in the image is.

The larger the positive values are, the higher positive correlation will be. A nearest interpolation process of M is also applied here as in [17].

Overall, DDT manipulates feature maps based on principle component analysis (PCA) to calculate the indicator matrix of an image for common object prediction. By adding weights onto the feature map, we can select the most relevant feature maps and attenuate the irrelevant ones. Therefore, a more precise indicator matrix can be generated by applying weighted PCA in DDT, which we term the method as Weighted-DDT.

3.4. The Architecture of Our Method

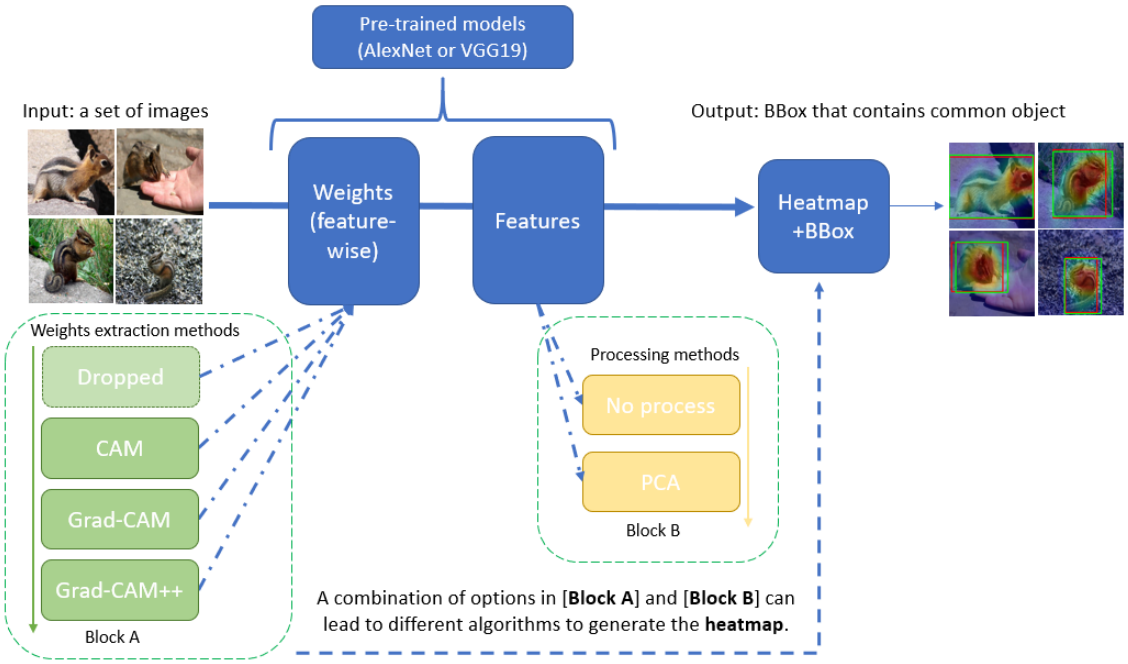


Figure 2. Overall architecture of the proposed method.

To summarize, the architecture of 'weight-and-feature' method is shown in *Figure 2*. Given a group of images which contain a common object, our objective is to localize the common object over these images. By choosing a pretrained backbone network (AlexNet or VGG19) and feeding these images into the network, the middle process can be broken into two main blocks.

Block A is how the feature-wise weights are extracted. Based on different modules in CAM family, we can obtain the weights according to CAM, Grad-CAM, or Grad-CAM++. Block B is how the feature maps of the last convolutional layer are processed. In DDT [17], it

directly applies PCA on the feature maps and use its first principle component to calculate indicator matrix for prediction. In this way, the feature-wise weights are not even required. Different from DDT, CCAM based approach does not make any processing techniques on features. Hence, if we combine the feature-wise weights from Block A and PCA in Block B, we can get Weighted-DDT that outperforms the original DDT or CCAM. The visual explanations are presented in Section 5.2.4.

4. Experiment

Our experiment is based on official pre-trained deep network models provided in Pytorch [26]. We evaluate our method with two baseline networks, i.e. AlexNet and VGG19. For acquiring fair comparison, we treat VGG19-based Grad-CCAM++ as the baseline and construct its variants with all modules controlled. All images are reshaped to (224, 224) for better visualization and normalized via the mean and standard deviation of ImageNet [27] (means = [0.485, 0.456, 0.406] stds = [0.229, 0.224, 0.225]).

4.1. Dataset

There is no training process in our method, and the pre-trained network are all trained on the 1000 different classes on ILSCRV2012 [2]. We follow [16] to evaluate the performance of our model on 6 subsets from ImageNet that are not included in training set.

4.1.1. How to Get the Valid Dataset

At the moment, ImageNet is under maintenance and as stated in their website, only ILSVRC synsets are included in the search results. Therefore, it makes a lot more difficult for us to get the dataset and corresponding bounding boxes data than previous researchers. Below is the entire process of how we get the valid chipmunk (1 out of 6 unseen subsets used in [16, 17, 23, 24]) dataset as an example.

If we search the specific class such as '*chipmunk*' in ImageNet, i.e. http://imagenet.stanford.edu/search?q=chipmunk, we will get no result. Therefore, we need another way to access the dataset.

One way to do this is to explore the synsets via subtree as shown in *Figure 3*.

Figure 3. Imagenet subtree

If we follow the order below, we can find the expected 'chipmunk' class.

(Animal, animate being, beast, brute, creature, fauna → chordate → vertebrate, craniate → Mammal, mammalian → placental, placental mammal, eutherian, eutherian mammal → Rodent, gnawer → squirrel → chipmunk)

We can only download the URLs of images in this synset and there are 1255 in 'chipmunk' class. However, not every image has a corresponding bounding box. The total number of bounding boxes are only 307. If we compare this number with the total amount of images (1255), it is much less. Secondly, another problem is that the bounding box and image URLs have no relationships such that we cannot match one image to its related bounding box data.

Figure 4 will be the result if we click 'down URLs' button in

http://imagenet.stanford.edu/synset?wnid=n02360282.

```
http://users.xplornet.com/~konecny/wildlife/animals/chips.jpg
http://static.flickr.com/1347/977029738_b78d61dd92.jpg
http://farm2.static.flickr.com/1166/1454741119_b3f2d8a0ba.jpg
http://farm1.static.flickr.com/153/384253101_b70825f1c8.jpg
http://farm1.static.flickr.com/58/214687632_5af929301e.jpg
http://farm1.static.flickr.com/12/14373093_c89997de63.jpg
http://farm1.static.flickr.com/103/278607469_3eb61203fc.jpg
http://laelaps.files.wordpress.com/2007/06/chipmunk.jpg
http://farm1.static.flickr.com/86/219944254_83cdc69e42.jpg
```

Figure 4. URL samples of chipmunk category without image index

However, these URLs and their related images cannot be connected with corresponding bounding box labels as they have no index.

Fortunately, we found a reliable source where we can get all the 1255 chipmunk images with the following link:

http://www.image-net.org/api/text/imagenet.synset.geturls.getmapping?wnid=n02360282.

```
n02360282_13072 http://farm1.static.flickr.com/224/485869153_3c7a25aaaa.jpg
n02360282_13055 http://farm1.static.flickr.com/180/401308314_20642a8b21.jpg
n02360282_3731 http://farm3.static.flickr.com/2139/2063305731_1c75c2bf77.jpg
n02360282_3741 http://farm1.static.flickr.com/116/254289614_6477bb13bf.jpg
n02360282_3709 http://farm2.static.flickr.com/1377/1466297622_574879dfa6.jpg
n02360282_3727 http://www.stewardshipgarden.org/Images/chipmunk.jpg
```

Figure 5. URL samples of chipmunk category with image index

Figure 5 shows sample URLs with image index from the link. It is noted that the n02360282 is the synset ID followed by the image ID (e.g. 13072 in the first lane in *Figure 5*). Although the images' index ID are all random, they can be matched with the bounding box labels which share the same ID. However, some of the URLs are not valid and cannot link to expected chipmunk images. Hence, to get all the valid images, we firstly find the corresponding URLs based on all the bounding box label indexes (xml type). If the webpage is valid and shows a relevant image, then we download it and save its name with the image ID and bounding box labels ([xmin, ymin, xmax, ymax]). This will make the dataloader a lot easier to build and access. One sample image in our test dataset is shown in *Figure 6.*

n02360282_13620 46 60 344 331.jpg

Figure 6. Sample image downloaded from Imagenet with a rename by its index and bounding box labels.

The full ImageNet object class ID can be accessed from:

http://image-net.org/archive/words.txt, where we can find the exact category IDs for our expected 6 subsets.

4.1.2. Statistical Characteristics of the Dataset

As discussed before, due to the failure of image links, some images are no longer available online. Thus, our testing is only based on the existing valid images over these six subsets. *Table 3* shows full existing images with bounding box labels from the ImageNet.

Table 3. Statistics of the dataset

	Chipmunk	rhino	Stoat	Raccoon	Rake	wheelchair
No. of labels	307	213	237	1427	540	459
Valid images	222	141	122	780	314	283
Dataset in [17, 23]	158	88	104	103	145	173
Effective rate	0.723	0.661	0.515	0.547	0.581	0.617

Hence, the dataset is much larger than the dataset used in [17, 23], which means our testing dataset contains more types of images. Also, for *rhino, stoat, raccoon, and wheelchair* category, there is one incorrectly labelled image based on the visualization on all the failed images. These four failed images are all correctly localized from human observations, thus will be added to successful co-localization cases in Corloc metric.

4.2. Generating Bounding Boxes

To generate a bounding box result from image co-localization, we use a similar method in [5, 16] to segment the heatmap. In particular, we segment the regions where only above threshold pixel values are reserved. We set the threshold as 30% of the maximum value of CCAM. However, we only take $K = 5$ top components for computing the CCAM in our experiments as it saves a great computational power and preserves the optimal performance. Lastly, the largest connected component in the segmentation map will be selected to generate the final prediction bounding box.

In our experiment, another bounding box generation method is also attempted since there are some complicated heatmap which may have multiple heat centers. Based on these

scenarios, we tested another bounding box generation method which selects the rectangle that has the maximum average heat. However, despite having some advantages over the largest connected component method on some cases, the overall performance is lower than the former approach.

4.3. Evaluation Metric

Following previous image co-localization works [4, 16, 17, 23, 24], we adopt the correct localization (CorLoc) metric to evaluate our propose method. CorLoc is defined as the percentage of images that are correctly localized based on PASCAL-criterion [28]:

$$\frac{area\left(B_{prediction} \cap B_{ground\ truth}\right)}{area\left(B_{prediction} \cup B_{ground\ truth}\right)} > 0.5, \tag{17}$$

where $B_{prediction}$ and $B_{ground\ truth}$ are prediction and ground truth bounding boxes respectively.

5. Results and Discussion

We evaluate our approach by comparing previous works that are implemented in the test dataset described in 4.1.2. Quantitative evaluation results are shown in *Table 4*.

Table 4. Corloc metric on various methods implemented in the experiment on image sets disjoint with ImageNet.

	Chipmunk	Rhino	Stoat	Raccoon	Rake	Wheelchair	Mean
CCAM (AlexNet) [6]	0.189	0.556	0.271	0.345	0.328	0.222	0.320
Grad-CCAM++ (AlexNet)	0.311	0.683	0.329	0.428	0.357	0.232	0.392
Grad-CCAM (VGG)	0.302	0.563	0.271	0.356	0.331	0.236	0.345
Grad-CCAM++ (VGG)	0.536	0.869	0.575	0.654	0.443	0.430	0.586
DDT (VGG) [11]	0.542	0.830	0.419	0.622	0.544	0.548	0.585
Weighted-DDT (VGG)*	0.543	0.837	0.443	0.678	0.573	0.541	0.603

* It gets weights from Grad-CAM++

5.1. Visualization

We present several sample images containing both original image (scaled) and heatmap together to visualize our object co-localization results. The best result is achieved by using VGG-19 in Pytorch as the pre-trained deep model, followed by a Weighted-DDT approach.

Figure 7 shows four successful predictions for each class on different backgrounds and scenarios. The graphical results demonstrated that the bounding boxes generated by our proposed framework match with ground truth in a good manner. The common object can be localized accurately with various sizes and locations. This approach is object proposal free and super-pixel free. It is also robust to various background noises.

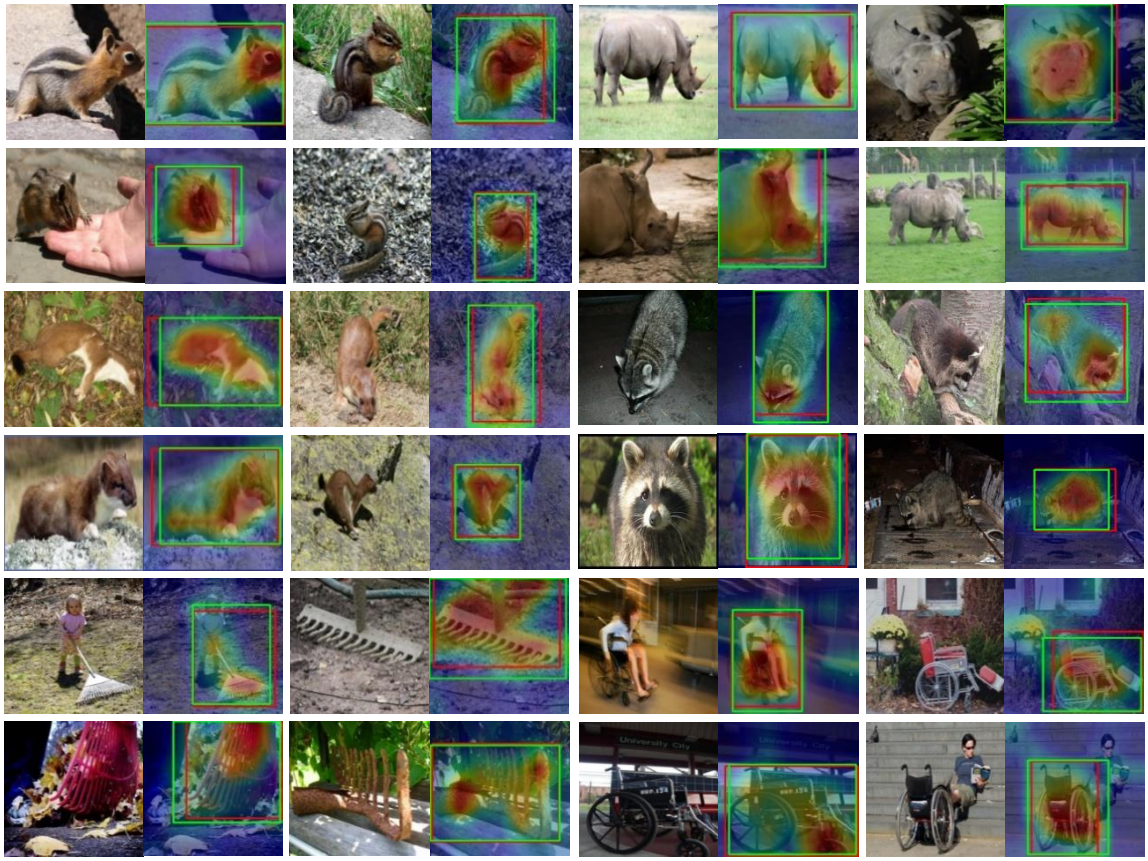


Figure 7. Successful visual examples of object co-localization on the six ImageNet Subsets. Each class contains four different scenarios. In these images, the red rectangle is the ground truth bounding box, and the green rectangle is the prediction by Weighted-DDT. (Best viewed in colour and zoomed in).

5.2. Ablation Study

We will explain our results based on three groups of comparisons, while each consists of an animal and an artifact image for demonstration.

5.2.1. VGG19 vs. AlexNet

Table 5. Comparison of the Corloc metric on two different models in terms of VGG19 vs. AlexNet.

Models\Corloc	Chipmunk	Rhino	Stoat	Raccoon	Rake	Wheelchair	Mean
Grad-CCAM++ (AlexNet)	0.311	0.683	0.329	0.428	0.357	0.232	0.392
Grad-CCAM++ (VGG)	0.536	0.869	0.575	0.654	0.443	0.430	0.586

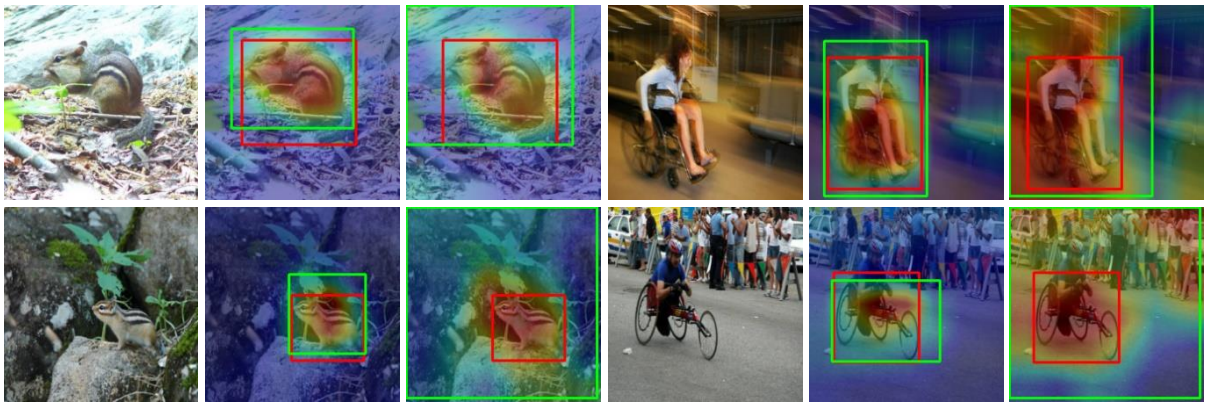


Figure 8. Left: original image; Middle: predicted bounding box (green) of VGG19-based model; Right: predicted bounding box (green) of AlexNet-based model. It is noted AlexNet-based method have high response on background.

Table 5 is the Corloc metric based on two different models that are compared in this section. The prediction results on sample images by these two models are shown in *Figure 8*.

Comparing VGG19 and AlexNet as the backbone for Grad-CAM++ based CCAM, VGG19 outperforms AlexNet by a large margin. This is mainly because compared with VGG19, AlexNet is less robust against background noise as illustrated in *Figure 8*. For demonstrating this finding, we further calculate the area rate α between VGG19 and AlexNet based method, which is given by

$$\alpha = \frac{area\left(B_{prediction}\right)}{area\left(B_{ground\ truth}\right)},$$

where $B_{prediction}$ and $B_{ground\ truth}$ are prediction and ground truth bounding boxes respectively.

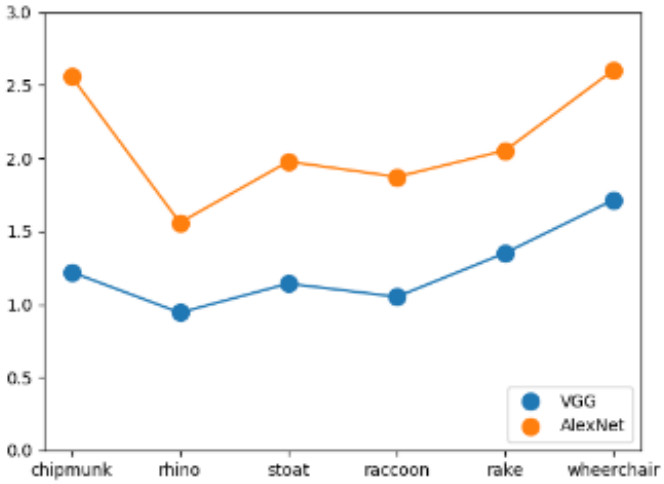


Figure 9. Comparison of area rates between VGG-based and AlexNet-based method.

From the illustration in *Figure 9*, we can find that $\alpha(VGG19)$ across all images within the class is around one through all classes, which means its predicted bounding box have a similar size with the corresponding ground truth box. As for $\alpha(AlexNet)$, it is clearly much larger than $\alpha(VGG19)$ across all six classes. This indicates that AlexNet tends to provide large bounding box predictions that covers more background than VGG19. In general, compared with AlexNet, VGG19 as a backbone has the advantage of not only high classification accuracy but also the robustness to noisy background.

5.2.2. Grad CAM vs. Grad CAM++

Table 6. Comparison of the Corloc metric on two different models in terms of Grad CAM vs. Grad CAM++.

Models\Corloc	Chipmunk	Rhino	Stoat	Raccoon	Rake	Wheelchair	Mean
Grad-CCAM (VGG)	0.302	0.563	0.271	0.356	0.331	0.236	0.345
Grad-CCAM++ (VGG)	0.536	0.869	0.575	0.654	0.443	0.430	0.586

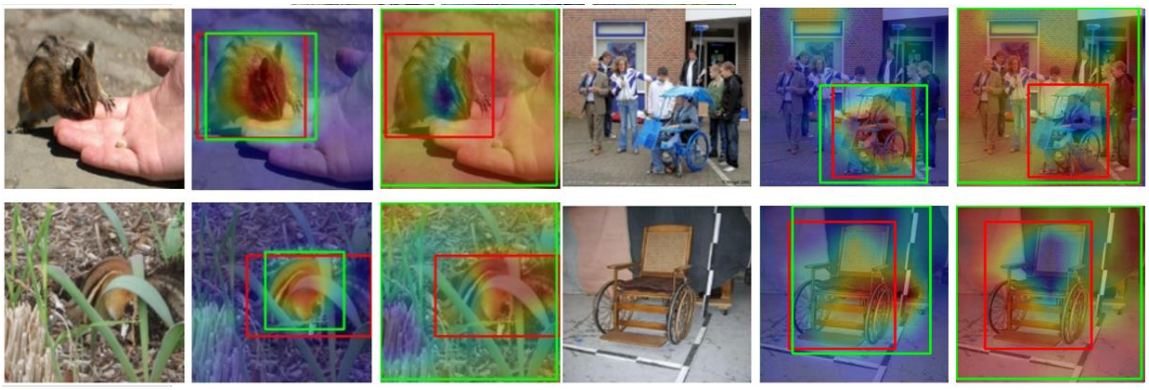


Figure 10. Left: original image; Middle: predicted bounding box (green) of Grad-CAM++ based model; Right: predicted bounding box (green) of Grad-CAM based model. Grad-CAM results in abnormal heatmaps that are even opposite to left one, as some unnecessary feature maps influence the result. Such problem is handled by Grad-CAM++.

Table 6 is the Corloc metric based on two different models that are compared in this section. The prediction results on sample images by these two models are shown in *Figure 10*.

Grad-CAM and Grad CAM ++ are compared based on VGG-based CCAM, and *Figure 10* clearly shows the significant advantages of Grad-CAM++ over Grad-CAM. As proposed in [15], Grad-CAM++ is able to capture better salient features due to an adoption of weighted combination of gradients. It also contributes to co-localization task as the influence of irrelevant feature maps are greatly suppressed.

5.2.3. DDT vs. CCAM

Table 7. Comparison of the Corloc metric on two different models in terms of Grad DDT vs. CCAM

Models\Corloc	Chipmunk	Rhino	Stoat	Raccoon	Rake	Wheelchair	Mean
DDT (VGG) [11]	0.542	0.830	0.419	0.622	0.544	0.548	0.585
Grad-CCAM++ (VGG)	0.536	0.869	0.575	0.654	0.443	0.430	0.586

Table 7 is the Corloc metric based on two different models that are compared in this section. Based on VGG19, the performance of DDT is almost the same with Grad-CCAM++ on mean CorLoc, but are very different on *stoat*, *rake*, and *wheelchair* classes. This is mainly because the PCA used in DDT statistically calculates the dominating spatial feature among all feature maps. It will make DDT focus on the common features that are frequently appeared in all images with the same class but ignore the less important features on the target object. This characteristic of DDT over CCAM can sometimes lose other features of

the common object and result in inaccurate bounding box. However, it sometimes can provide good robustness on noisy images. Here we compare them across three scenarios.

1) Both DDT and CCAM are correct.

In *Figure 11*, we can clear observe that, DDT only focus on the head and part of the body of a chipmunk. As a comparison, CCAM provides attention on the head, main body and tail of the chipmunk, despite locates highest activation on head as well. Therefore, although both methods can give over 0.5 Corloc prediction, CCAM shows some advantages over DDT in this case.

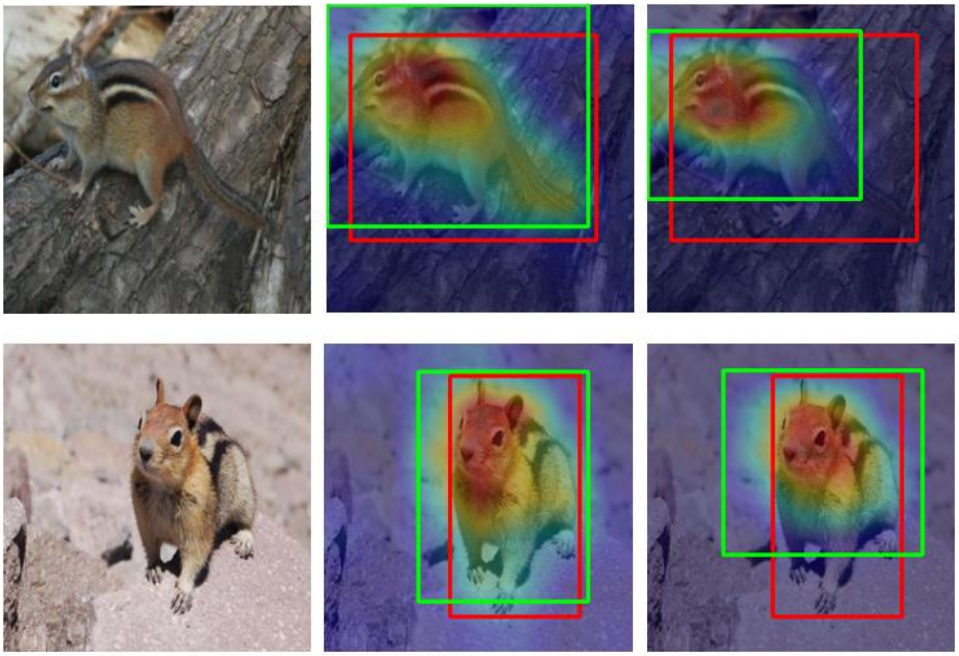


Figure 11. Left: Original image; Middle:predicted bounding box (green) by CCAM; Right: predicted bounding box (green) by DDT. Both of them have a Corloc larger than 0.5 in this case.

2) Grad-CCAM++ (VGG19) correct; DDT incorrect

Similar with 1), as shown in *Figure 12*, CCAM can reflect activations on both the body and head of a stoat, thus lead to a correct prediction with the functionality of weights. However, without using weights, DDT sometimes ignores other less important features of an object, sometimes gives weak response on those features. Depend on how DDT provides response on the less important features, the prediction could be incorrect if it completely neglects those features.

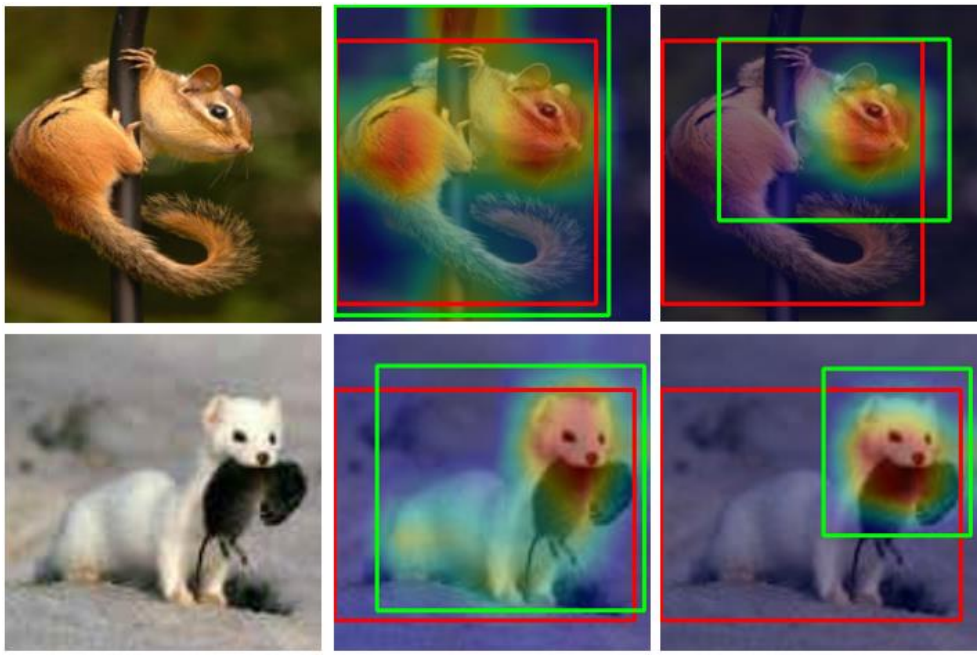


Figure 12. Left: Original image; Middle:predicted bounding box (green) by CCAM; Right: predicted bounding box (green) by DDT.In this case. CCAM outperforms DDT.

3) Grad-CCAM++ (VGG19) incorrect; DDT correct

For noisy class images like rake and wheelchair, DDT shows its great advantage over CCAM as illustrated in *Figure 13*. This is because DDT is less likely to be affected by the noisy factor – people. As a large portion of images in rake category contain both people and rake, CCAM sometimes tends to capture this combination as the 'common object'. Similar phenomenon could also be founded in wheelchair class in which most wheelchairs appear with people sitting on them (207 out of 283 valid images). However, since DDT only focus on features that appear most often, it will only provide a strong response on wheelchair (especially the wheels) rather than other noisy object (such as people). Therefore, the prediction accuracy on rake and wheelchair categories are much higher by using DDT than CCAM as shown in *Table 7*.

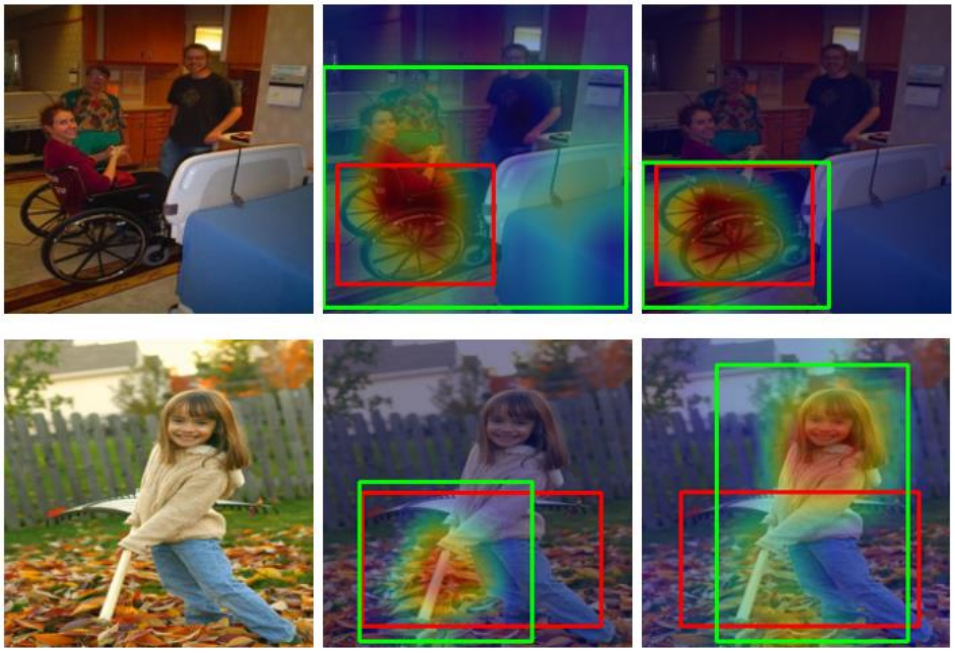


Figure 13. Left: Original image; Middle:predicted bounding box (green) by CCAM; Right: predicted bounding box (green) by DDT.In this case. DDT outperforms CCAM.

5.2.4. Weighted-DDT vs. DDT

Table 8. Comparison of the Corloc metric on two different models in terms of Weighted DDT vs. DDT

Models\Corloc	Chipmunk	Rhino	Stoat	Raccoon	Rake	Wheelchair	Mean
DDT (VGG) [11]	0.542	0.830	0.419	0.622	0.544	0.548	0.585
Weighted-DDT (VGG)	0.543	0.837	0.443	0.678	0.573	0.541	0.603

The Weighted-DDT and DDT have similar Corloc performance as shown in Table *8*. They also have close visualizations on prediction. Based on weight values generated from Grad-CAM++, we discard feature maps which have corresponding zero weight values. The number of abandoned feature maps over the six categories are 55, 38, 30, 42, 53, 33 respectively. The original overall feature map amount is 512, which is in accordance with the output size from the last convolutional layer. The difference can be visualized through the heatmap generated from adopted heatmaps and abandoned heatmaps.

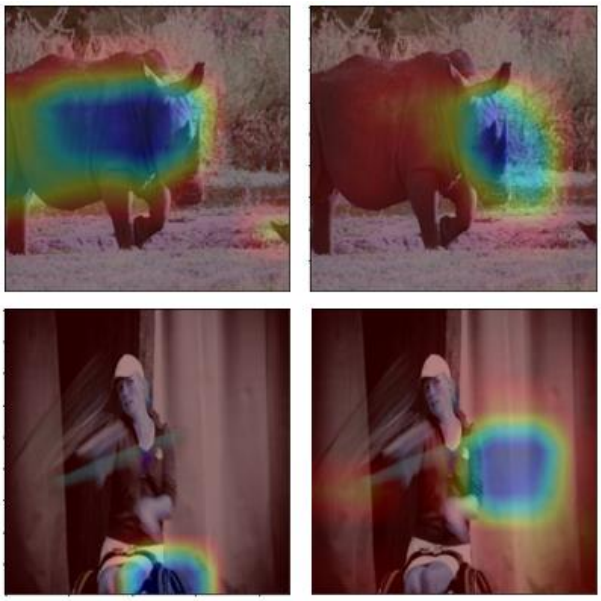


Figure 14: Left: heatmap generated from adopted features (474 for rhino and 479 for wheelchair); Right: heatmap generated from abandoned features (38 for rhino and 33 for wheelchair).

As illustrated in *Figure 14*, the heatmaps from abandoned features have no significant features related to the object. Obviously, they are worse activation maps than those from adopted features. Thus, these abandoned features should have less contributions to localize the expect common object. Therefore, with these small portions of features abandoned, Weighted-DDT shows approximately **2%** of the Corloc increase over DDT.

5.3. Error Analysis

In this Section, we will investigate a detailed error analysis based on Weighted-DDT model which has the Corloc performance listed in *Table 9*.

Table 9. Comparison of the Corloc metric on two different models in terms of Grad-CCAM++ vs. Weighted-DDT

Models\Corloc	Chipmunk	Rhino	Stoat	Raccoon	Rake	Wheelchair	Mean
Weighted-DDT (VGG)	0.543	0.837	0.443	0.678	0.573	0.541	0.603

5.3.1. Weighted DDT model

We summarize eight types of failure cases by using our best model – Weighted-DDT (VGG19/ Grad-CAM++).

1) Incorrect heatmap (wrong object detection)

Figure 15. Visualizations on error type 1 with IOU on each image.

This error is the most difficult to correct since it is closely related to the pre-trained model. The incorrect heatmap indicates that our pre-trained model – VGG19 is not able to identify the target object we want and therefore results in a completely wrong prediction. As in *Figure 15*(a), the strongest response is shown on a cat rather than the raccoon. It is to some extent understandable as the identified cat looks very similar with raccoon by human observations. As in *Figure 15*(c), we can also find that the activated region contains a rod, which shares similar features with the rake. To solve this issue, we might need more powerful CNN models that has more accurate object classification and detection capabilities.

However, it is worth mentioning that these three images are the only three that have zero (or near zero) intersection over union (IOU) ratio over the entire dataset.

2) Inaccurate heatmap

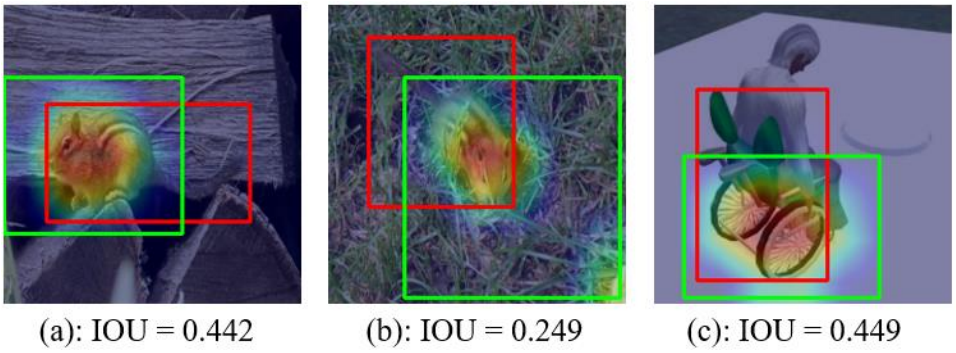


Figure 16. Visualizations on error type 2 with IOU on each image.

Compared with error type 1 where error occurs in object-wise level, type 2 error only occurs in feature-wise level.

The error is also mainly due to the characteristic of PCA. Since PCA is a statistical tool that selects the most dominate feature along all the feature maps, DDT or Weighted-DDT would only focus on the most significant feature that appears most frequently. Depend on how much more attention paid on less important features, the prediction result can be very different. *Figure 16*(a) and (b) show that the tails of chipmunks in the give images are not detected. Similarly, the heatmap in *Figure 16*(c) only focuses on wheels and ignores the backrest of a wheelchair.

3) Correct heatmap, inaccurate bounding box

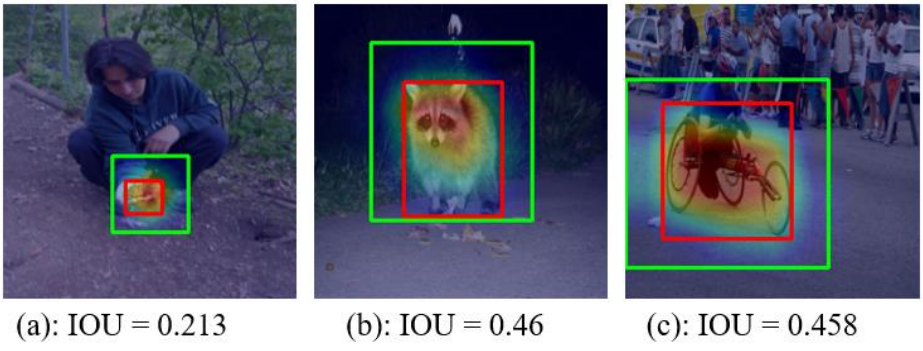


Figure 17. Visualizations on error type 3 with IOU on each image.

This is the most common type of error and most failed images are of this type. We can find that the activations are tightly enclosed in the red boxes for all images in *Figure 17*. However, due to a larger prediction box (green box) outside the ground truth box, the Corloc value is computed less than 0.5 which leads to a failed co-localization case. One

straight idea to solve the problem is to increase the threshold of pixel value during bounding box generation. However, the threshold increase can result in a performance drop on other images. The previous successfully localized images could fail because of this threshold change. Hence, an adaptive or more intelligent bounding box generation algorithm is needed in the future works.

4) Noisy image

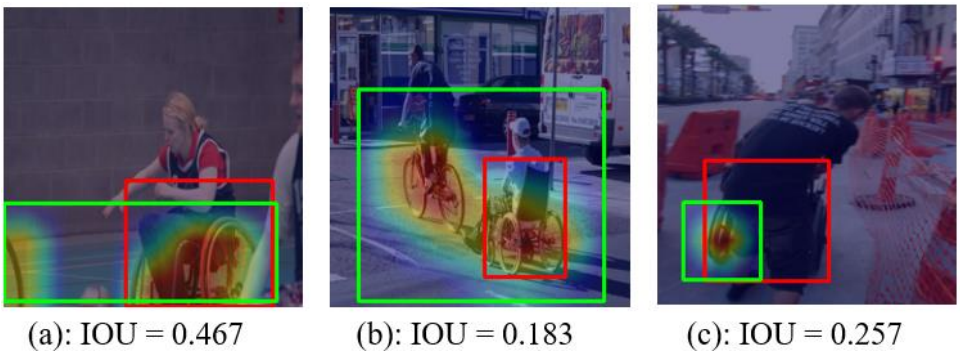


Figure 18. Visualizations on error type 4 with IOU on each image.

Noisy image could be another reason to result in an error. As in *Figure 18* (a), the prediction is seriously affected by a strong response from left bottom corner. It is quite understandable as the response is indeed a part of the wheel from a second wheelchair. Thus, the green box (prediction) combines the two main activations while red box (ground truth) only includes one. Similar case happens in (b) as bicycle share close features (wheels) with wheelchair, resulting in a wrong prediction bounding box. As in (c), we can clearly observe that the people who is behind the wheelchair blocks most of the object. Accordingly, lots of important features such as armrest, backrest and right-side wheel are lost. This occlusion of the object can bring extra noise on the co-localization. To resolve this issue, we might be able to detect salient object first for *Figure 18*(a) and (b). For scenario (c), we probably need more such images and labels in order to apply supervised learning.

5) Uncommon shape of the object

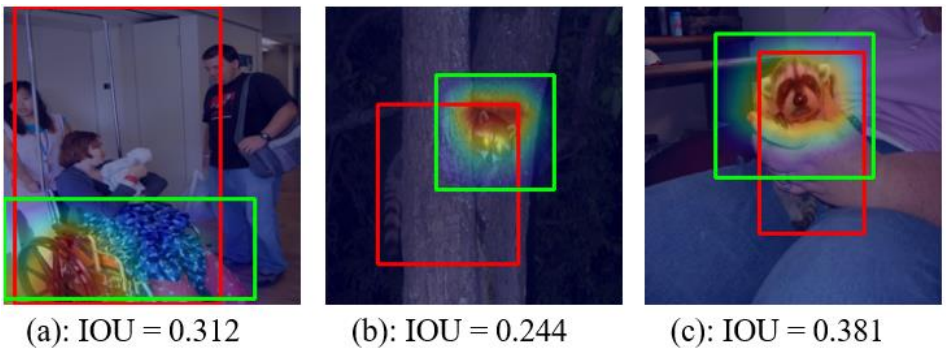


Figure 19. Visualizations on error type 5 with IOU on each image.

Some images are quite different compared with the others within the group. For example, the wheelchair in *Figure 19*(a) has two long rods on the backrest which are also part of the object. This kind of feature is extremely difficult to identify and therefore leads to a wrong prediction even though the main features of the object are activated. Among all 283 images in wheelchair category, this is the only wheelchair with this shape. *Figure 19*(b) and (c) show a similar case where the main body of the object is blocked. Hence, this object looks quite unusual compared with other objects within the group.

6) Potential inaccurate ground truth bounding box labels

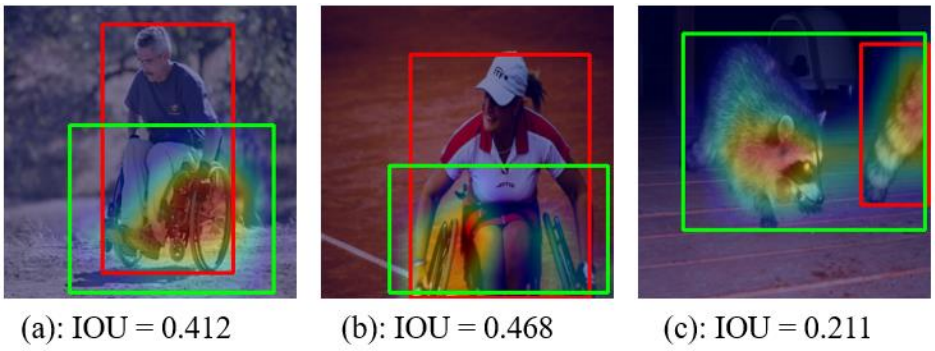


Figure 20. Visualizations on error type 6 with IOU on each image.

There are a few images that have different ground truth bounding box labels with the author's observation. Some samples are presented in *Figure 20*. These labels could be wrong due to human mistakes. However, these incorrectly localized images with this type of error are still recognized as failures in our Corloc result.

7) Too many similar objects within the image

Figure 21. An illustration of different shapes of rake images.

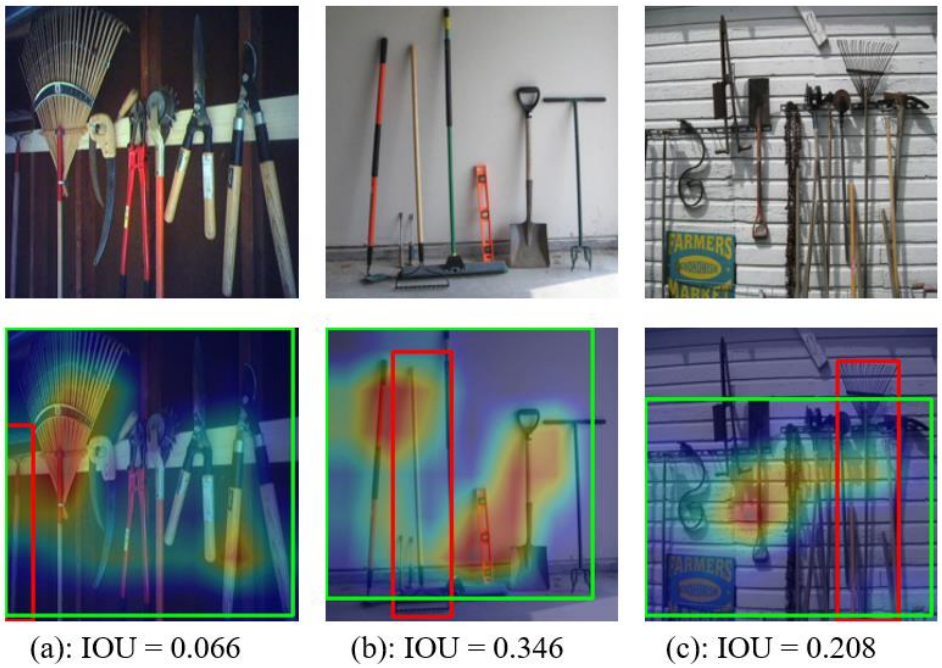


Figure 22. Visualizations on error type 7 with IOU on each image.

To have a clear view of the error, the original images are also presented in this case. This type of error only happens in rake category. Within the rake image group, the target common object 'rake' has shown various types in *Figure 21*. Therefore, in order to localize the specific common object within images in *Figure 22*, it is quite hard for the network and our model to distinct rake from other similar objects. Since other objects like shovel and broom also contain key features of rake (such as long rod, and triangle structure at the junction), our model is not effective in this scenario. However, there are only a few of such cases and these images are even hard to co-localize for human.

8）Unrecognizable images

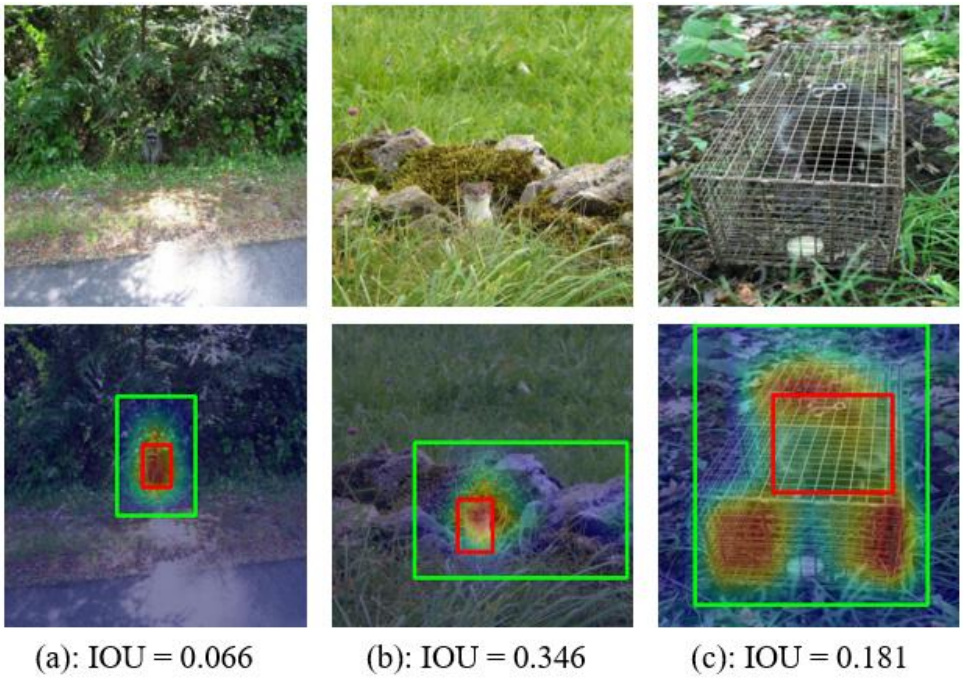
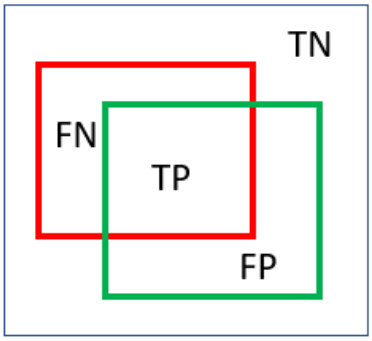


Figure 23. Visualizations on error type 8 with IOU on each image.

There are a small number of images in our dataset that are very confusing for image-localization. Most of these images have complicated background and it is difficult to discriminate the object from the background. To have a clearer observation, the original images are also presented in *Figure 23*. It can be illustrated that the objects we expect to be localized in *Figure 23*(a) (b) and (c) are quite hard to detect. These objects seem to be hidden in the surrounding environments. Even for humans, it is hard to recognize and localize these objects by observations.

5.3.2. Confusion Matrix Analysis



Red box = ground truth bounding box
Green box = prediction bounding box
TP = True Positive
FP = False Positive
FN = False Negative
TN = True Negative

Figure 24. Confusion matrix representations on an image

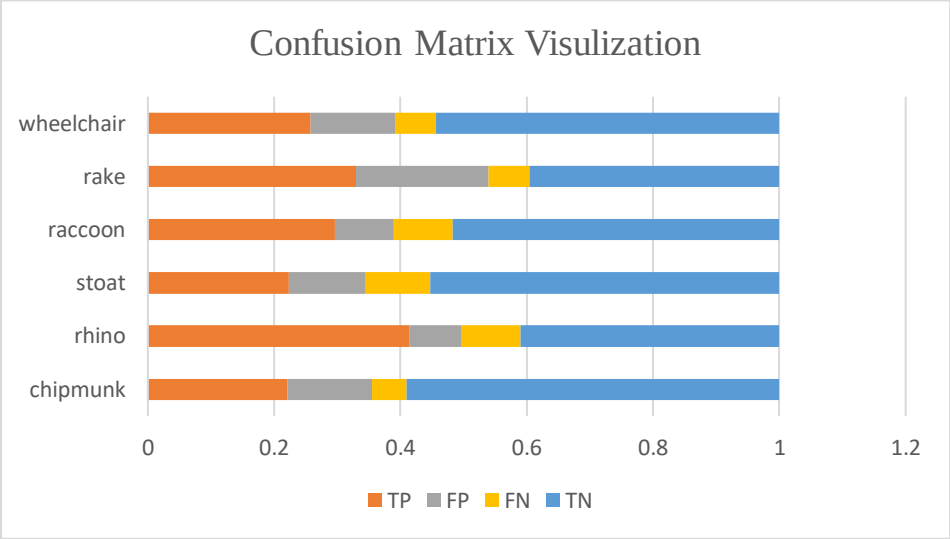


Figure 25. Visualization of confusion matrix over 6 categories

To better understand the localization accuracies, we quantize the confusion matrix on these six subsets as shown in *Figure 25*. According to *Figure 24*, we can add TP and FN to represent the ground truth bounding box size with respect to the whole image. Similarly, FP plus TN stands for the prediction bounding box size. There is one key finding that matches with the intuition.

Rhino has the largest average ground truth bounding box sizes (TP+FN). It means rhino has the highest object over image size ratio across all six categories in the dataset. Hence, it indicates that rhino is most likely to be recognized as a salient object in the image, therefore has the highest mean CorLoc value. This finding highly matches with our intuition: rhino is the easily category to do co-localization across all six classes.

Accordingly, not only from the statistical Corloc results but also from visualizing localization effect perspective, we can draw a conclusion that common objects are more easily and accurately to be localized if they appear as salient object in the image. Therefore, this could inspire our future work where we need to detect salient object first and then proceed image co-localization based on the salient object detection.

6. Conclusion

In this work, we review the development of CAM and propose the concept, CAM family, that takes the weighted features to generate visual activation map on unseen objects. By modularizing the members in CAM family, which can be applied on DDT and CCAM, the localization performance on unseen objects are evaluated. We test our method and its corresponding variants on six unseen classes from ImageNet. With extensive experiments on the dataset, we find the best module combinations on tackling image co-localization, that is Weighted DDT with Grad-CAM++ weights using pretrained VGG19 network. Thorough comparisons between different models are investigated. As a future work, it is worth exploring the feasibility of using better backbone networks and discover the influence of specific network module on visual explanations. More options of manipulating features can be attempted. Also, a better and more robust bounding box generation method should be explored.

Bibliography

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580-587.

[4] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei, "Co-localization in real-world images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1464-1471.

[5] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921-2929.

[6] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 648-656.

[7] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91-99.

[9] C. Wang, W. Ren, K. Huang, and T. Tan, "Weakly supervised object localization with latent category learning," in *European Conference on Computer Vision*, 2014: Springer, pp. 431-445.

[10] R. Gokberk Cinbis, J. Verbeek, and C. Schmid, "Multi-fold mil training for weakly supervised object localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2409-2416.

[11] T. Deselaers, B. Alexe, and V. Ferrari, "Weakly supervised localization and learning with generic knowledge," *International journal of computer vision*, vol. 100, no. 3, pp. 275-293, 2012.

[12] M. Pandey and S. Lazebnik, "Scene recognition and weakly supervised object localization with deformable part-based models," in *2011 International Conference on Computer Vision*, 2011: IEEE, pp. 1307-1314.

[13] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[14] 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 international conference on computer vision*, 2017, pp. 618-626.

[15] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks," *arXiv preprint arXiv:1710.11063*, 2017.

[16] W. Li, O. H. Jafari, and C. Rother, "Localizing Common Objects Using Common Component Activation Map," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 28-31.
[17] X.-S. Wei, C.-L. Zhang, J. Wu, C. Shen, and Z.-H. Zhou, "Unsupervised object discovery and co-localization by deep descriptor transformation," *Pattern Recognition*, vol. 88, pp. 113-126, 2019.
[18] W. Ren, K. Huang, D. Tao, and T. Tan, "Weakly supervised large scale object localization with multiple instance learning and bag splitting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 405-416, 2015.
[19] Z. Shi, T. M. Hospedales, and T. Xiang, "Bayesian joint topic modelling for weakly supervised object localisation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2984-2991.
[20] L. Bazzani, A. Bergamo, D. Anguelov, and L. Torresani, "Self-taught object localization with deep networks," in *2016 IEEE winter conference on applications of computer vision (WACV)*, 2016: IEEE, pp. 1-9.
[21] R. G. Cinbis, J. Verbeek, and C. Schmid, "Weakly supervised object localization with multi-fold multiple instance learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 1, pp. 189-203, 2016.
[22] P. O. Pinheiro and R. Collobert, "From image-level to pixel-level labeling with convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1713-1721.
[23] Y. Li, L. Liu, C. Shen, and A. van den Hengel, "Image co-localization by mimicking a good detector's confidence score distribution," in *European Conference on Computer Vision*, 2016: Springer, pp. 19-34.
[24] H. Le, C.-P. Yu, G. Zelinsky, and D. Samaras, "Co-localization with category-consistent features and geodesic distance propagation," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 1103-1112.
[25] L. Delchambre, "Weighted principal component analysis: a weighted covariance eigendecomposition approach," *Monthly Notices of the Royal Astronomical Society*, vol. 446, no. 4, pp. 3545-3555, 2015.
[26] A. Paszke *et al.*, "Automatic differentiation in pytorch," 2017.
[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009: Ieee, pp. 248-255.
[28] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, no. 1, pp. 98-136, 2015.