Where to Look?: Mining Complementary Image Regions for Weakly Supervised Object Localization

Sadbhavana Babar, Sukhendu Das
Visualization and Perception Lab,
Dept. of Computer Science and Engineering, IIT Madras, India
sadbhav@cse.iitm.ac.in, sdas@iitm.ac.in

Abstract

Humans possess an innate capability of recognizing objects and their corresponding parts and confine their attention to that location in a visual scene where the object is spatially present. Recently, efforts to train machines to mimic this ability of humans in the form of weakly supervised object localization, using training labels only at the image-level, have garnered a lot of attention. Nonetheless, one of the well-known problems that most of the existing methods suffer from is localizing only the most discriminative part of an object. Such methods provide very little or no focus on other pertinent parts of the object. In this paper, we propose a novel way of meticulously localizing objects using training with labels as for the entire image by mining information from complementary regions in an image. Primarily, we adapt to regional dropout at complementary spatial locations to create two intermediate images. With the help of a novel Channel-wise Assisted Attention Module (CAAM) coupled with a Spatial Self-Attention Module (SSAM), we parallelly train our model to leverage the information from complementary image regions for excellent localization. Finally, we fuse the attention maps generated by the two classifiers using our Attention-based Fusion Loss. Several experimental studies manifest the superior performance of our proposed approach. Our parallel classifiers aid in mining all relevant parts of the object (e.g., for a dog - its face, forelegs, hindlegs) along with its most-discriminative part (head). In this way, our model learns to focus attention on “where to look” for the specified object in the given input image as well as localize objects in a weakly-supervised manner. During inference, we do not hide any patches of the input image. The test image as a whole is provided to our trained CNN model.

1. Introduction

Given a visual scene, humans have an inherent ability to recognize and localize objects of interest with minimal effort. With the advent of deep convolutional neural networks [16, 17], there has been a remarkable improvement in image recognition [15, 26, 29] and object detection [10, 20, 21, 23, 24, 27, 30, 35]. However, these methods rely on full supervision during training. Recently, there has been an increasing focus on Weakly Supervised Learning (WSL) techniques that require minimal supervision or coarse annotation during training, which reduces the effort of using costly pixel-level annotations. One of the fundamental computer vision tasks like semantic segmentation that require fine pixel-level annotations, can now be trained using only bounding box annotations or image-based labels using the
Weakly Supervised Object Localization (WSOL) aims to classify as well as localize objects without using expensive bounding box annotations during training. Recently, a lot of approaches [7, 22, 28, 39, 42, 48, 49, 51] have been proposed to tackle this challenging problem. Zhou et al. [51] put forward the idea of appending a Global Average Pooling (GAP) [19] layer at the end of convolutional neural networks (CNNs) followed by a fully-connected layer to generate a class activation map (CAM). CAM highlights the discriminative image region used to recognize that object category. However, a crucial limitation of this approach is that it only localizes the most discriminative class-specific region instead of the entire object. For e.g., given an image of a dog, it only tries to generate implicit attention on its face, without paying any heed to its remaining body parts. Hence, it often leads to sub-optimal localization performance.

To overcome this problem, a few recent methods [8, 28, 42, 46] have come up with making changes to input image rather than modifying the algorithm. In the paper, Hide-and-Seek (HaS) [28], Singh and Lee attempt to randomly hide patches of an input image during training so that their model tries to seek other visible relevant parts of the object. Even though this approach focuses on non-discriminative object parts, it loses information during training when the patches are hidden, leading to a limited localization performance. This gives rise to an interesting question: Is there any way to optimize the localization performance by maximally utilizing the information lost in regional dropout?

We propose to solve the above problem by introducing to strategically mine information from complementary image regions. Regional dropout methods [8, 50] have significantly demonstrated the ability to generalize well on image classification and object localization. We also venture to leverage this generalization ability and create two complementary images, each possessing regional dropout at complementary spatial locations in the respective images. To create these input images, we adapt to randomly hide patches in the input image similar to Hide-and-Seek [28], as illustrated in figure 1. We perform joint training of these complementary image regions as two input channels, using two parallel classifiers. Further, we try to fuse the information captured in both these input channels by incorporating a novel Channel-wise Assisted Attention Module (CAAM) along with a Spatial Self-Attention Module (SSAM). Both these modules take input features extracted from pre-trained CNNs. CAAM takes inspiration from [11, 40, 47], and tries to model interactions in the channel dimension between features extracted from two complementary images. SSAM is inspired by [11, 36, 47] to capture feature dependencies in the spatial dimension. We finally aggregate the interdependencies modeled by these two modules: CAAM and SSAM, for better localization ability. We also propose an Attention-based Fusion Loss, inspired by [43], to fuse the two attention maps obtained using the complementary images. Almost all the previous works rely only on the classification objective to learn the implicit attention maps, which serve as a testimony of visual explanations learned by the model to localize objects. However, we feel that relying only on the classification objective for localizing objects limits the overall localization performance. The use of our proposed Attention-based Fusion Loss, along with the usual cross-entropy loss to train our localization model, to the best of our knowledge, is the first of its kind.

Our key contributions are summarized as follows: 1) We propose a novel way of training a network for weakly supervised object localization that mines information from complementary regions in an image, individually as well as when fused. 2) We propose a novel Channel-wise Assisted Attention Module (CAAM). Along with a Spatial Self-Attention Module (SSAM), CAAM jointly aids in localizing integral object regions. 3) We also propose an Attention-based Fusion Loss criteria to fuse the attention maps generated by the two parallel classifiers. Our proposed loss function diligently captures all relevant parts of the concerned object of interest, thereby suppressing background regions. 4) Our method achieves state-of-the-art object localization performances on two benchmark datasets: CUB-200-2011 [34] and ILSVRC 2016 [25]. We achieve a Top-1 localization of 64.70% on CUB-200-2011 and 52.36% on ILSVRC 2016 datasets.

2. Related Work

Correspondence with human visual perception: The two-stream theory of the human visual system proposed by Goodale et al. [13] highlights the two distinct visual pathways in the human visual system viz., the ventral pathway or the “what pathway” and the dorsal pathway or the “where pathway” that jointly aid in recognizing and localizing objects respectively. We take motivation from the human vision system to jointly model the “what” and “where” pathways and efficiently perform object localization in a weakly supervised setting.

Weakly Supervised Learning: Learning strong predictive models using imprecise labels is becoming a trend since it involves cheaper annotations and reduced human effort for manual labeling of data. Also, the huge availability of weakly labeled data in the form of videos and images over the internet makes it possible to explore various real-world problems in deep learning in a weakly supervised paradigm. Although supervised object detection methods [21, 23, 24, 27] have made tremendous progress, the fact that they require costly bounding box annotations has led to the exploration of weakly supervised object detection methods [9, 31, 32, 45] using only image-level labels.

Regional Dropout: Randomly masking certain regions
in an input image have found to be effective in capturing richer object context and better generalization performance. Bazzani et al. [4] proposed to mask out certain regions in an image that lead to a drop in the image recognition performance, finally feeding the regions to an agglomerative clustering algorithm which indicate higher objectness of such merged regions in the input image. In Hide-and-Seek [28], the crux was to randomly hide patches in an input image forcing the network to focus on other relevant object parts. Cutout [8] is yet another successful generalizable approach that drops a certain amount of input region from the input image. However, these methods lose information while training the network using regional dropout. We make use of information lost in regional dropout while training the network, by generating two images to mask complementary spatial locations.

**Attention in Deep Neural Networks:** Attention mechanism was first proposed in the pioneering work [33] by Vaswani et al. in machine translation to model long-range dependencies that the recurrent neural networks failed to handle. Since then, attention has been used in a wide variety of applications including image captioning [37], visual grounding [12], visual question answering [2], sound source localization [3]. Self-attention has also made its way in visual question answering [41], in which it was used for question embedding, and in [40], it was used along with a guided-attention module to model interactions for different feature modalities. In one of the most recent works [47], self-attention was used in image generation. In all these works, self-attention has proved to generate better feature representations. In our proposed framework, we adapt to attention mechanism coupled with regional dropout for more meaningful representations of our complementary images.

**Other methods for weakly supervised object localization:** The work in [49] generates self-produced guidance masks, which in turn are used in the form of pixel-level supervision for localizing objects. Zhang et al. [48] proposed adversarial erasing in feature space that mine information from two adversarial parallel classifiers for superior localization performance. Choe and Shim, in their work [7], proposed to use self-attention mechanism to generate a drop mask and an importance map from the input feature map and randomly select either of them along with the input feature map for localizing objects. Yang et al. [39] uses a linear combination of activation maps from the highest probability score of a class to the lowest probability score, thereby assisting in suppressing the background regions and focusing more on the foreground object of interest. The most recent work of EIL [22] by Mai et al. attempts to jointly perform adversarial erasing and mining discriminative regions to localize objects efficiently.

3. Proposed Approach

Looking at complementary image regions helps the network in paying attention to the concise details of the object. Our detailed network architecture is illustrated in figure 2.

3.1. Notations

Given an input image \( I \), with its image-level label, \( y_i \), the goal of weakly supervised object localization is to learn a model that is capable of classifying the input image \( I \) into one of \( C \) object categories in the dataset and localizing the
object in that image using a bounding box $B$. From $I$, we create two input images, $X$ and $\tilde{X}$, with regional dropout at complementary spatial locations in an image, as shown in figure 1. We adapt to hide patches in the input image as in [28]. We form our input $X$ by randomly hiding patches of an image. However, we regain the information lost in input $\tilde{X}$ by forming another input $X$, which we call the $X$ complement. Input $\tilde{X}$ reveals the information in the hidden patches of input $X$, whereas hides the information in visible patches of input $X$. We extract features from inputs $X$ and $\tilde{X}$ using a shared CNN having parameters $\theta$. We denote these features as $F_x$ and $F_{\tilde{x}}$, where $F_x, F_{\tilde{x}} \in \mathbb{R}^{C \times H \times W}$.

3.2. Mining Information from Complementary Image Regions

The information captured from CNN features ($F_x$ and $F_{\tilde{x}}$) of both inputs $X$ and $\tilde{X}$ are used individually as well as combined (fused) using Spatial Self Attention Module (SSAM) and Channel-wise Assisted Attention Module (CAAM) modules respectively (shown in figure 2). Finally, we aggregate the features captured both by SSAM and CAAM for better feature representations of $F_x$ and $F_{\tilde{x}}$, denoted by $F_x^t$ and $F_{\tilde{x}}^t$ respectively. $F_x^t$ and $F_{\tilde{x}}^t$ are then passed to a global average pooling layer [19], followed by their respective classifiers. By jointly training the two branches concerning the inputs $X$ and $\tilde{X}$, viz., classifier $X$ and classifier $\tilde{X}$, our model precisely gets an idea regarding “where to look” in the input image while classifying it correctly.

3.3. Channel-wise Assisted Attention Module

To compute CAMs [51], Zhou et al. proposed to multiply the weights of the last fully connected layer of the classifier to the feature maps of the preceding convolution layer. Towards the last layers of the CNN, the feature maps tend to capture the class-specific responses. Hence, CAM highlights the most discriminative region of the object belonging to that category. In the work [11], Fu et al. put forth the idea of a Channel Attention Module to capture long-range inter-dependencies between channels of feature maps in a fully supervised setting for the task of semantic segmentation. Adapting the idea from [11], we attempt to leverage the class-specific inter-dependencies between channels of input features from both the branches, $F_x$ and $F_{\tilde{x}}$. So, our CAAM module takes as input the CNN features, $F_x$ and $F_{\tilde{x}}$ and outputs features with more meaningful representation, denoted by $F_x^c$ and $F_{\tilde{x}}^c$ respectively. $F_x^c, F_{\tilde{x}}^c \in \mathbb{R}^{C \times H \times W}$. A similar approach has been studied in [52] recently using a cross-correlated attention network in the spatial dimension. However, our CAAM module tries to capture inter-dependencies in the channel dimension of two feature maps.

To compute $F_x^c$, we take the input features $F_x, F_{\tilde{x}}$ and reshape them to $\mathbb{R}^{C \times N}$, where $N = H \times W$ corresponds to the number of pixels in the feature map. We then generate channel attention matrix $Q_x$ as follows:

$$Q_x = Softmax(F_x \otimes F_x^T)$$  \hspace{1cm} (1)

where, $Q_x \in \mathbb{R}^{C \times C}$ and $\otimes$ denotes matrix multiplication. $Q_x$ consists of the learnable attention weights denoted by $\lambda_{ij} \in Q_x$, $i, j \in \{1...C\}$, which capture inter-dependencies between the channels of $F_x$ and $F_{\tilde{x}}$. Further, we multiply transpose of $Q_x$ with $F_{\tilde{x}}$ to get $Y_x$, as:

$$Y_x = Q_x^T \otimes F_{\tilde{x}}$$  \hspace{1cm} (2)

We then reshape $Y_x$ as $\mathbb{R}^{C \times H \times W}$ and generate $F_x^c$ as:

$$F_x^c = F_x + \delta_x Y_x$$  \hspace{1cm} (3)

Here, $\delta_x$ is used to scale the features of $Y_x$. It is initially set to 0 and iteratively trained similar to that in [11, 47]. The detailed expression for $F_x^c$ is as follows:

$$F_x^{c_{ij}} = F_{x_{ij}} + \delta_x \sum_{k=1}^{C} \lambda_{kj} F_{x_{kij}}$$  \hspace{1cm} (4)

$F_x^c$ refers to channel-wise attended features of input $F_x$ assisted by input $F_{\tilde{x}}$. We can see in equation (4) that the final features $F_x^c$ are a weighted sum of features of all locations of input feature $F_{\tilde{x}}$ and the original features $F_x$. Similarly, to compute $F_{\tilde{x}}^c$ we follow the same set of steps as followed for $F_x^c$. However, we save computations as well.

![Figure 3. Channel-wise Assisted Attention Module (CAAM).](image-url)

The input to this module is the CNN features, $F_x$ and $F_{\tilde{x}}$ of inputs $X$ and $\tilde{X}$ respectively, and it outputs the Channel-wise attended features of input $F_x$ assisted by input $F_{\tilde{x}}$. We interchange $F_x$ and $F_{\tilde{x}}$ in figure 3 to obtain the Channel-wise attended features of input $F_{\tilde{x}}$ assisted by input $F_x$. The part indicated in dotted blue rounded rectangle indicates shared computation while computing both $F_x^c$ and $F_{\tilde{x}}^c$. 

Figure 3. Channel-wise Assisted Attention Module (CAAM).
as parameters in generating the channel attention matrix \(Q_x\) (shown in figure 3), as it is the transpose of \(Q_x\).

\[ Q_x = \text{Softmax}(F_x \otimes F_x^T) \quad (5) \]

From equations (1) and (5), it is evident that \(Q_x\) is actually transpose of \(Q_x\). But for the purpose of simplicity, we denote it as \(Q_x\) itself. Also, we denote its learnable attention weights as \(\lambda_x^{ij} \in Q_x\) and \(i, j \in \{1...C\}\). Similarly, for \(F_x^C\):

\[ F_x^{C(i)} = F_x^{ij} + \delta_x \sum_{k=1}^C \lambda_x^{ki} F_x^{kj} \quad (6) \]

Similar to equation (4), \(\delta_x\) is also used as a scaling factor for \(Y_x\). It is initially set to 0 and learns weight as training progresses. \(F_x^C\) refers to channel-wise attended features of input \(F_x\) assisted by input \(F_x\). As shown in equation (6), the features of \(F_x^C\) are a weighted sum of features of all locations of input feature \(F_x\) and the original features \(F_x\).

3.4. Spatial Self-Attention Module

Apart from the inter-dependencies between the features of channels \(F_x\) and \(F_x\) modeled by CAAM, it is significant to consider their individual contribution as well for efficient feature representation. We also hypothesize that to get the object’s correct spatial location, it is important to have an overall view of the visual scene and give the corresponding weightage to the entire scene as per the objectness. In the work [47], self-attention was used in GANs [14]. Taking motivation from [47], we propose to use spatial self-attention for localizing objects. So the input to our SSAM module is the features \(F_x\) and \(F_x\) and its output is spatially attended features \(F_x^C\) and \(F_x^S\) respectively. Both \(F_x^C\) and \(F_x^S\) are of dimension \(\mathbb{R}^{C \times H \times W}\). As illustrated in figure 4, given features \(F_x\), we compute matrices \(M, L\) and \(P\) using 1x1 convolution where, \(\{M, L\} \in \mathbb{R}^{C \times H \times W}\), where \(C = C/8\), and \(P \in \mathbb{R}^{C \times H \times W}\). Mathematically, we compute \(F_x^S\) as:

\[ K_x = \text{Softmax}(M^T \otimes L) \]
\[ R_x = P \otimes R_x^T \]
\[ F_x^S = F_x + \alpha_x R_x \quad (7) \]

where, \(\alpha_x\) is a weight factor for \(R_x\). The parameter \(\alpha_x\) and the weight matrices \(M, L, P, K_x\) and \(R_x\) are learnt during training. Similarly, \(F_x^S\) can be formulated as:

\[ F_x^S = F_x + \alpha_x R_x \quad (8) \]

3.5. Aggregation

For features \(F_x\) and \(F_x\) coming from each of the input branches, we have two enhanced feature representations, \(\{F_x^C, F_x^S\}\) and \(\{F_x^C, F_x^S\}\): the channel assisted features and the spatially attended features respectively. To take advantage of complementary information in both these features, we fuse them using an element-wise sum. Finally, a convolution layer is used to bind them together as follows:

\[ F_x^t = \text{conv}(F_x^C + F_x^S); \quad F_x^t = \text{conv}(F_x^C + F_x^S) \quad (9) \]

Here, \(F_x^C\) and \(F_x^S\) denote the feature maps from final convolution layer in our proposed framework. We use outputs of these final convolution layers to generate localization maps.

3.6. Attention-based Fusion Loss

We train our model in an end-to-end way to obtain two localization maps, in a manner similar to CAM [51]. We use cross-entropy loss for training both the classifier branches. However, both the classifiers discover complementary object parts during training. Thus, it is necessary to fuse the pair of localization maps. This is done by our Attention-based Fusion Loss, such that our model learns to focus on the entire object during training and generalizes well during testing (as we do not use two branches during testing).

Calculating localization maps: The features from our last convolution layer, \(F_x^t\) and \(F_x^t\) having parameters \(\theta^c\) and \(\theta^s\) consist of \(C\) feature maps each having spatial dimension \(H \times W\). These features are fed to the global average pooling (GAP) [19] layer. Let the value of \(k^{th}\) feature map at spatial location \((m, n)\) of \(F_x^t\) and \(F_x^t\) be denoted as \(F_x^{t_k}(m, n)\) and \(F_x^{t_k}(m, n)\) respectively. After performing GAP on the \(k^{th}\) feature maps, we get the activation units \(G_x^{t_k}\) and \(G_x^{t_k}\) respectively. We pass the outputs from GAP layer to the respective
Algorithm 1: Our training algorithm

**Input:** \(N\) training images along with their image-level labels, \(\{(I_i, y_i)\}_{i=1}^{N}\), hyperparameter \(\beta\).

1. **while** convergence condition not met **do**
2.  Create two images \(X\) and \(\tilde{X}\) with spatial dropout at complementary image locations, for an input image;
3.  Compute CNN features as: \(F_x \leftarrow f(X, \theta)\) and \(F_{\tilde{x}} \leftarrow f(\tilde{X}, \theta)\);
4.  Use CAAM to compute: \(F^c_x \leftarrow f^{CAAM}(F_x)\), \(F^c_{\tilde{x}} \leftarrow f^{CAAM}(F_{\tilde{x}})\);
5.  Use SSAM to compute: \(F^s_x \leftarrow f^{SSAM}(F_x)\), \(F^s_{\tilde{x}} \leftarrow f^{SSAM}(F_{\tilde{x}})\);
6.  Aggregate CAAM and SSAM outputs:
5.1. \(F^c_x \leftarrow \text{conv}(F^c_x + F^c_{\tilde{x}})\), \(F^s_x \leftarrow \text{conv}(F^s_x + F^s_{\tilde{x}})\);
7.  Compute predicted labels as: \(p_x = g_x(X, \theta^x, F^c_x)\), \(p_{\tilde{x}} = g_{\tilde{x}}(\tilde{X}, \theta^{\tilde{x}}, F^s_{\tilde{x}})\);
8.  Compute cross entropy loss for classifiers \(X\) and \(\tilde{X}\):
8.1. \(L_{CE_x} = -\sum y_i \log p_x \), \(L_{CE_{\tilde{x}}} = -\sum y_i \log p_{\tilde{x}}\);
9.  Compute Attention-based Fusion Loss as in eq. (14);
10. Obtain total loss as:
10.1. \(L_{\text{total}} = L_{CE_x} + L_{CE_{\tilde{x}}} + \beta \ast L_{\text{at-fuse}}\);
11. Backpropagate loss and update parameters \(\hat{\theta}, \theta^x, \theta^{\tilde{x}}\);
12. **end**

classifiers. Let the weights for a given class \(c\) coming from the \(k^{th}\) activation unit be denoted as \(W^k_x\). The softmax outputs of the classifiers for a particular class \(c\) are denoted by \(H_{x_c}\) and \(H_{\tilde{x}_c}\). Mathematically, we denote this process as:

\[
G^k_x = \sum_{m,n} F^k_x (m, n); \quad G^k_{\tilde{x}} = \sum_{m,n} F^k_{\tilde{x}} (m, n) \quad (10)
\]
\[
H_{x_c} = \sum_k W^k_x G^k_x; \quad H_{\tilde{x}_c} = \sum_k W^k_{\tilde{x}} G^k_{\tilde{x}} \quad (11)
\]

From equations (10) and (11),

\[
H_{x_c} = \sum_{m,n} \sum_k W^k_x F^k_x (m, n); \quad (12)
\]

Similar to equation (12), we express \(H_{\tilde{x}_c}\) in terms of \(W^k_{\tilde{x}}\), \(F^k_{\tilde{x}}\). For a particular class \(c\), we denote the localization maps for both the input features \(F^c_x\) and \(F^c_{\tilde{x}}\), as follows:

\[
A_{x_c} (m, n) = \sum_k W^k_x F^k_x (m, n);
\]
\[
A_{\tilde{x}_c} (m, n) = \sum_k W^k_{\tilde{x}} F^k_{\tilde{x}} (m, n) \quad (13)
\]

We finally combine these localization maps \(A_{x_c}\) and \(A_{\tilde{x}_c}\) using our proposed Attention-based Fusion Loss function (as illustrated in figure 5).

**Fusing the localization maps:** Unlike in [48], which relies on non-differentiable \(\text{max}\) function for fusing localization maps from two classifiers, we propose to combine the localization maps using an Attention-based Fusion loss inspired from [43]. We first convert the obtained localization maps into their respective vectorized forms, i.e., \(V_{x_c} = vec(A_{x_c})\) and \(V_{\tilde{x}_c} = vec(A_{\tilde{x}_c})\) and perform \(l_2\)-normalization of \(V_{x_c}\) and \(V_{\tilde{x}_c}\). Our proposed Attention-based Fusion Loss is formulated as follows:

\[
L_{\text{at-fuse}} = \left( \frac{V_{x_c}}{||V_{x_c}||_2} - \frac{V_{\tilde{x}_c}}{||V_{\tilde{x}_c}||_2} \right)^2 \quad (14)
\]

We simply train our network with the proposed Attention-based Fusion Loss coupled with the categorical cross-entropy loss for efficient and integral object localization. The total loss function for training our model is:

\[
L_{\text{total}} = L_{CE}(y, p_x) + L_{CE}(y, p_{\tilde{x}}) + \beta \ast L_{\text{at-fuse}} \quad (15)
\]

where, \(L_{CE}\) denotes the categorical cross-entropy loss function, \(\beta\) is a hyperparameter used to scale our Attention-based Fusion Loss. Empirically, we choose \(\beta = 50\) in our experiments. \(y\) denotes the true labels, \(p_x\) and \(p_{\tilde{x}}\) denote the predictions made by our complementary classifiers.

4. Experiments

4.1. Experimental Setup

**Datasets:** We perform our experiments on two benchmark datasets used for object localization, CUB-200-2011 [34] and ILSVRC 2016 [25]. CUB-200-2011 has a total
of 11,788 images spanning across 200 bird categories, of which 5,994 images are used for training and 5,794 for testing. ILSVRC 2016 has approximately 1.2 million images in the training set across 1000 different categories, and 50,000 images in the validation set. We compare our results across different methods on the ILSVRC 2016 validation set.

Evaluation Metrics: We evaluate our method using the following metrics: 1) Top-1 localization (Top-1 Loc) accuracy [25] calculates the fraction of images that are correctly classified and the predicted bounding box has 50% IoU with the ground truth bounding box. 2) Top-1 classification (Top-1 Clas) accuracy determines the fraction of images that are correctly classified. 3) GT-known localization (GT-Loc) accuracy [28] only considers the fraction of images for which the predicted bounding box has 50% IoU with the ground truth bounding box, independent of the Top-1 classification accuracy. 4) Apart from the above three standard metrics, we also evaluate our method on the recently proposed MaxBoxAccv2 [6] metric (as shown in table 4).

4.2. Implementation Details

We experiment with VGG16 [26] and ResNet50 [15] as the backbone CNN architectures for our proposed approach. As in [51], we remove the layers after conv5-3 in the VGG16 network. We insert our CAAM and SSAM modules after conv5-3 layer of the original VGG16 network. The aggregated outputs from both CAAM and SSAM modules are then fed to a global average pooling (GAP) layer [19], followed by a fully-connected layer for classification. We follow similar steps for ResNet50 backbone as well. Both VGG16 and ResNet50 architectures are initialized with weights pre-trained on ImageNet [25] dataset. We extract our localization maps followed by bounding boxes, in a similar way to [51]. During testing, we do not hide patches in the input image, similar to [28].

activate CAAM and SSAM modules during testing, similar to vanilla CAM [51] model for fair comparison with other existing state-of-the-art methods (shown in tables 1, 2 & 3).

4.3. Ablation Studies

Hyperparameters: For our complementary input images, we use a hide probability of 0.5 to hide and render complementary image locations, in the corresponding input images. Similar to [28], we also experiment with different patch sizes, \{16, 32, 44, 56\} for regional dropout during training. As illustrated in Algorithm 1, we use a hyperparameter \(\beta\) to scale the proposed Attention-based Fusion Loss during training. We set \(\beta\) to 50 in all our experiments.

Importance of each module in the architecture: Our proposed architecture has three components, CAAM, SSAM, and an Attention-based Fusion Loss to fuse localization maps from two distinct branches during training. We study the effect of each of these modules on localization ac-
Table 2. Localization Results on ILSVRC 2016 dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1 Loc</th>
<th>GT-Loc</th>
</tr>
</thead>
<tbody>
<tr>
<td>InceptionV3-CAM [51]</td>
<td>46.29</td>
<td>–</td>
</tr>
<tr>
<td>GoogLeNet-HaS-32 [28]</td>
<td>45.21</td>
<td>60.29</td>
</tr>
<tr>
<td>InceptionV3-SPG [49]</td>
<td>48.60</td>
<td>64.69</td>
</tr>
<tr>
<td>InceptionV3-DANet [38]</td>
<td>47.53</td>
<td>–</td>
</tr>
<tr>
<td>InceptionV3-MEIL [22]</td>
<td>49.48</td>
<td>–</td>
</tr>
<tr>
<td>VGG-CAM [51]</td>
<td>42.80</td>
<td>57.72</td>
</tr>
<tr>
<td>VGG-ACoL [48]</td>
<td>45.80</td>
<td>62.96</td>
</tr>
<tr>
<td>VGG-ADL [7]</td>
<td>44.92</td>
<td>–</td>
</tr>
<tr>
<td>VGG-CCAM [39]</td>
<td>48.22</td>
<td>63.58</td>
</tr>
<tr>
<td>VGG-EIL [22]</td>
<td>46.27</td>
<td>–</td>
</tr>
<tr>
<td>Ours-ResNet50</td>
<td>51.64</td>
<td>66.32</td>
</tr>
<tr>
<td>VGG-ADL [7]</td>
<td>44.92</td>
<td>–</td>
</tr>
<tr>
<td>VGG-CCAM [39]</td>
<td>48.22</td>
<td>63.58</td>
</tr>
<tr>
<td>VGG-EIL [22]</td>
<td>46.27</td>
<td>–</td>
</tr>
<tr>
<td>Ours-ResNet50</td>
<td>51.64</td>
<td>66.32</td>
</tr>
<tr>
<td>VGG-CAM [51]</td>
<td>42.80</td>
<td>57.72</td>
</tr>
<tr>
<td>VGG-ACoL [48]</td>
<td>45.80</td>
<td>62.96</td>
</tr>
<tr>
<td>VGG-ADL [7]</td>
<td>44.92</td>
<td>–</td>
</tr>
<tr>
<td>VGG-CCAM [39]</td>
<td>48.22</td>
<td>63.58</td>
</tr>
<tr>
<td>VGG-EIL [22]</td>
<td>46.27</td>
<td>–</td>
</tr>
<tr>
<td>Ours-ResNet50</td>
<td>51.64</td>
<td>66.32</td>
</tr>
</tbody>
</table>

Table 3. Classification performance on ILSVRC 2016 dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1 Clas (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>InceptionV3-CAM [51]</td>
<td>68.10</td>
</tr>
<tr>
<td>GoogLeNet-HaS-32 [28]</td>
<td>70.70</td>
</tr>
<tr>
<td>VGG-CAM [51]</td>
<td>66.60</td>
</tr>
<tr>
<td>VGG-ACoL [48]</td>
<td>67.50</td>
</tr>
<tr>
<td>VGG-ADL [7]</td>
<td>69.48</td>
</tr>
<tr>
<td>VGG-CCAM [39]</td>
<td>66.60</td>
</tr>
<tr>
<td>VGG-EIL [22]</td>
<td>70.48</td>
</tr>
<tr>
<td>Ours-VGG</td>
<td>71.24</td>
</tr>
</tbody>
</table>

Table 4. Evaluating our method on MaxBoxAccv2. We evaluate our model on the recently proposed MaxBoxAccv2 metric [6] on VGG16 as the backbone. Experiments for ResNet50 are provided in the Supplementary Section A.

Table 5. Effect of each module in the architecture. For the above experiment, we have used VGG16 as the backbone CNN.

Table 6. Localization accuracies with different patch sizes. For the above experiment, we have used ResNet50 as the backbone CNN architecture.

Effect of Patch Size on localization accuracy: We perform experiments with different patch sizes as in [28]. The patch sizes we use during training are either one of \{16, 32, 44, 56\}. We also come up with a Mixed model wherein the patch size is randomly sampled among the patch sizes \{16, 32, 44, 56\}, with uniform probability, for every image in every epoch during training. Unlike [28], we do not show full image during training in our Mixed approach. Our Mixed approach outperforms all the existing state-of-the-art models on localization accuracy (as shown in table 6). We do lose on some classification accuracy, as our model never encounters full image during training. Still, our method achieves comparable Top-1 Clas and best Top-1 Loc performance on both CUB-200-2011 and ILSVRC 2016 datasets. Qualitative results (shown in figure 6) ensure that our model looks at all object parts to make correct predictions. In future, we plan to evaluate our method on the recently proposed OpenImages30K [5, 6] dataset.

5. Discussion and Conclusion

We propose a novel way of mining information from complementary image regions to tackle the problem of Weakly-Supervised Object Localization. We show that our novel Channel-wise Assisted Attention Module (CAAM), when combined with a Spatial Self-Attention Module (SSAM), boosts existing feature representations for localizing integral object regions. We also propose a novel Attention-based Fusion Loss function to fuse the localization maps coming from two different input branches during training. In this way, our method is able to focus on discriminative as well as non-discriminative object parts for precise localization. Even though we study the task of single-object detection in a weakly-supervised manner, it will be interesting to explore the case of detecting multiple objects in a scene, laying the foundation for significantly bridging the gap between supervised and weakly-supervised methods.
References


