Rethinking the Route Towards Weakly Supervised Object Localization

Chen-Lin Zhang   Yun-Hao Cao   Jianxin Wu*

National Key Laboratory for Novel Software Technology

Nanjing University, Nanjing, China

{zhangcl,caoyh}@lamda.nju.edu.cn, wujx2001@nju.edu.cn

Abstract

Weakly supervised object localization (WSOL) aims to localize objects with only image-level labels. Previous methods often try to utilize feature maps and classification weights to localize objects using image level annotations indirectly. In this paper, we demonstrate that weakly supervised object localization should be divided into two parts: the class-agnostic object localization and the object classification. For class-agnostic object localization, we should use class-agnostic methods to generate noisy pseudo annotations and then perform bounding box regression on them without class labels. We propose the pseudo supervised object localization (PSOL) method as a new way to solve WSOL. Our PSOL models have good transferability across different datasets without fine-tuning. With the generated pseudo bounding boxes, we achieve 58.00% localization accuracy on ImageNet and 74.97% localization accuracy on CUB-200, which have a large edge over previous models.

1. Introduction

Deep convolutional neural networks have achieved enormous success in various computer vision tasks, such as classification, localization and detection. However, current deep learning models need a large number of accurate annotations, including image-level labels, location-level labels (bounding boxes and key points) and pixel-level labels (per pixel class labels for semantic segmentation). Many large-scale datasets are proposed to solve this problem [15, 10, 3]. However, models pre-trained on these large-scale datasets cannot be directly applied to a different task due to the differences between source and target domains.

To relax these restrictions, weakly supervised methods are proposed. Weakly supervised methods try to perform detection, localization and segmentation tasks with only image-level labels, which are relatively easy and cheap to obtain. Among these tasks, weakly supervised object localization (WSOL) is the most practical task since it only needs to locate the object *with a given class label*. Most of these WSOL methods try to enhance the localization ability of classification models to conduct WSOL tasks [19, 28, 29, 2, 27] using the class activation map (CAM) [30].

However, in this paper, through ablation studies and experiments, we demonstrate that the localization part of WSOL should be *class-agnostic*, which is not related to classification labels. Based on these observations, we advocate a paradigm shift which divides WSOL into two independent sub-tasks: the class-agnostic object localization and the object classification. The overall pipeline of our method is in Fig. 1. We name this novel pipeline as *Pseudo Supervised Object Localization (PSOL)*. We first generate pseudo groundtruth bounding boxes based on the class-agnostic method deep descriptor transformation (DDT) [26]. By performing bounding box regression on these generated bounding boxes, our method removes restrictions on most WSOL models, including the restrictions of allowing only one fully connected layer as classification weights [30] and the dilemma between classification and localization [19, 2].

We achieve state-of-the-art performances on ImageNet-1k [15] and CUB-200 [25] combining the results of these two independent sub-tasks, obtaining a large edge over previous WSOL models (especially on CUB-200). With classification results of the recent EfficientNet [22] model, we achieve 58.00% Top-1 localization accuracy on ImageNet-1k, which significantly outperforms previous methods.

We summarize our contributions as follows.

- We show that weakly supervised object localization should be divided into two independent sub-tasks: the class-agnostic object localization and the object classification. We propose PSOL to solve the drawbacks and problems in previous WSOL methods.

- Though generated bounding boxes are noisy, we argue that we should directly optimize on them *without using class labels*. With the proposed PSOL, we achieve
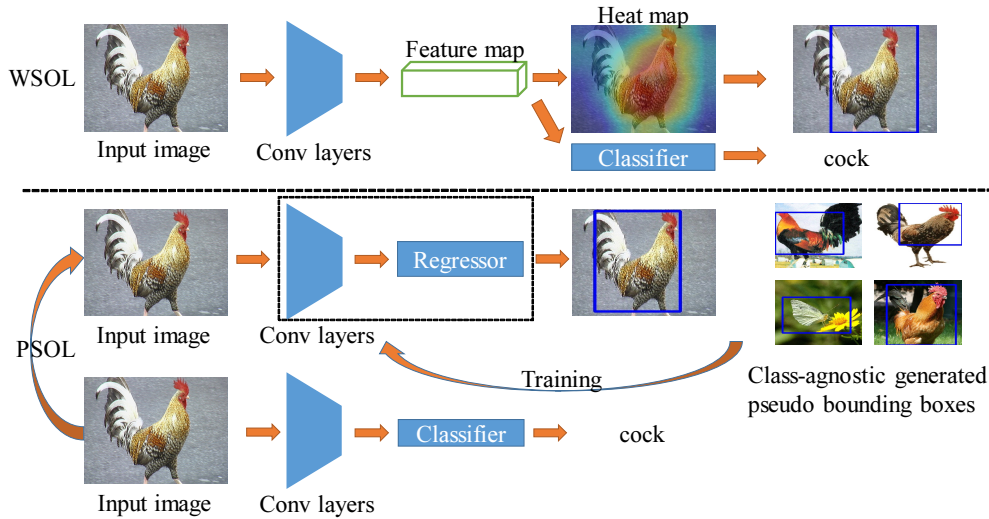
Figure 1: Overall pipeline of previous WSOL methods (top) and our proposed PSOL (bottom). Previous WSOL methods need the final feature map to generate bounding boxes *implicitly*. However, PSOL first generates inaccurate bounding boxes using class-agnostic methods, then perform bounding box regression to predict the bounding box *explicitly*.

58.00% Top-1 localization accuracy on ImageNet-1k and 74.97% Top-1 localization accuracy on CUB-200, which is far beyond previous state-of-the-art.

- Our PSOL method has good localization transferability across different datasets *without any fine-tuning*, which is significantly better than previous WSOL models.

2. Related Works

Convolutional neural networks (CNN), since the success of AlexNet [8], have been widely applied in many areas of computer vision, including object localization and object detection tasks. We will briefly review detection and localization with full supervision and weak supervision in this section.

2.1. Fully Supervised Methods

After the success of AlexNet [8], researchers tried to adopt CNN to conduct object localization and detection. The pioneering work OverFeat [17] tried to use sliding window and multi-scale techniques to conduct classification, localization and detection within a single network. VGG-Net [18] adds a per-class regression and model ensemble to enhance the prediction result of localization.

Object detection is another task that can generate bounding boxes and labels simultaneously. R-CNN [5] and Fast-RCNN [4] use the selective search [24] to generate candidate regions and then use CNN to classify them. Faster-RCNN [14] proposes a two-stage network: the region

proposal network (RPN) for generating regions of interest (ROI), then the R-CNN module to classify them and localize the object in the region. These popular two-stage detectors are widely used in detection tasks. YOLO [13] and SSD [11] are one stage detectors with carefully designed network structures and anchors. Recently, some anchor-free detectors are proposed to mitigate the anchor problem in common detectors like CornerNet [9] and CenterNet [31].

However, all these methods need massive, detailed and accurate annotations. Annotations in real-world tasks are expensive and sometimes even hard to get. So we need some other methods to perform object localization tasks without requiring many exact labels.

2.2. Weakly Supervised Methods

Weakly supervised object localization (WSOL) learns to localize the object with only image-level labels. It is more attractive since image-level labels are much easier and cheaper to obtain than object level labels. Weakly supervised detection (WSOD) tries to give the object location and category simultaneously when training images only have image-level labels.

WSOL has the assumption that there is only one object of the specific category in the whole image. Based on this assumption, many methods are proposed to push the limit of WSOL. [30] first generates class activation maps with the global average pooling layer and the final fully connected layer (weights of the classifier). Grad-CAM [16] uses gradients rather than output features to generate more accurate class response maps. Besides these methods which focus

on improving class response maps, some other methods try to make the classification model more suitable for localization tasks. HaS [19] tries to randomly erase some regions in the input image to force the network to be meticulous for WSOL. ACoL [28] uses two parallel classifiers with dynamic erasing and adversarial learning to discover complementary object regions more effectively. SPG [29] generates Self-Produced Guidance masks to localize the whole object. ADL [2] proposes the importance map and the drop mask, with a random selection mechanism to achieve a balance between classification and localization.

WSOD does not have the *one object in one class* restriction. However, WSOD often needs methods to generate region proposals like selective search [24] and edge boxes [32], which will cost much computation resources and time. Furthermore, current WSOD detectors use high resolution inputs to output the bounding boxes, leading to heavy computational burdens. Thus, most WSOD methods are difficult to be applied to large-scale datasets.

3. Methodology

In this section, we will mainly discuss the drawbacks of the current WSOL pipeline and propose our *pseudo supervised object localization (PSOL)*.

3.1. A paradigm shift from WSOL to PSOL

Current WSOL nethods can generate the bounding box with a given class label. However, the community have identified serious drawbacks of this pipeline.

- The learning objective is indirect, which will hurt the model's performance on localization tasks. HaS [19] and ADL [2] show that localization is not compatible with classification when only having a single CNN model. Localization tries to localize the whole object while classification tries to classify the object. The classification model often tries to localize only the most discriminative part of the object in an image.

- Offline CAM [30] has the thresholding parameter and needs to store the three-dimensional feature map for further computation. The thresholding value is tricky and hard to determine.

Those drawbacks make WSOL hard to apply to real-world applications.

Encouraged by the class-agnostic process that generats regions of interest (ROI) in selective search [24] and Faster-RCNN [14], we divide WSOL into two sub-tasks: the class-agnostic object localization and the object classification. Based on these two sub-tasks, we propose our PSOL method. PSOL directly optimizes the localization model on explicitly generated pseudo ground-truth bounding boxes. Hence, it removes the restrictions and drawbacks illustrated

---

Algorithm 1 Pseudo Supervised Object Localization

   Input: Training images I_{tr} with class label L_{tr}
   Output: Predicted bounding boxes b_{te} and class labels L_{te} on testing images I_{te}
1:   Generate pseudo bounding boxes $\tilde{b}_{tr}$ on I_{tr}
2:   Train a localization CNN F_{loc} on I_{tr} with $\tilde{b}_{tr}$
3:   Train a classification CNN F_{cls} on I_{tr} with L_{tr}
4:   Use F_{loc} to predict b_{te} on I_{te}
5:   Use F_{cls} to predict L_{te} on I_{te}
6: **Return:** b_{te}, L_{te}

---

in previous WSOL methods, and it is a paradigm shift for WSOL.

3.2. The PSOL Method

The general framework of our PSOL is in Algorithm 1. We will introduce our PSOL step by step. We will discuss the details of generating pseudo groundtruth bounding boxes in Sec 3.2.1, then the localization method used in our model in Sec 3.2.2. For the classification method, we use pre-trained models in the computer vision community directly.

3.2.1 Bounding Box Generation

The critical difference between WSOL and our PSOL is the generation of pseudo bounding boxes for training images. Detection is a natural choice for this task since detection models can provide bounding boxes and classes directly. However, the largest dataset in detection only has 80 classes [10], and it cannot provide a general object localizer for datasets with many classes such as ImageNet-1k. Furthermore, current detectors like Faster-RCNN [14] need substantial computation resources and large input image sizes (like shorter side=600 when testing). These issues prevent detection models from being applied to generate bounding boxes on large-scale datasets.

Without detection models, we can try some localization methods to output bounding boxes for training images directly. Some weakly and co-supervised methods can generate noisy bounding boxes, and we will give a brief introduction to them.

WSOL methods. Existing WSOL methods often follow this pipeline to generate the bounding box for an image. First the image I is feed into the network F, then the final feature map (often the output of the last convolutional layer) G is generated: $G \in \mathbb{R}^{h \times w \times d} = F(I)$, where h, w, d are the height, width and depth of the final feature map. Then, after global average pooling and the final fully connected layer, the label L_{pred} is produced. According to the predicted label L_{pred} or the ground truth label L_{gt}, we can get the class specific weights in the final fully connected layer

$W \in \mathbb{R}^d$. Then each spatial location of G is channel-wise weighed and summed to get the final heat map H for the specific class: $H_{i,j} = \sum_{k=1}^{d} G_{i,j,k} W_k$. Finally, H is upsampled to the original input size, and thresholding is applied to generate the final bounding box.

DDT recap. Some co-supervised methods can also have good performances on localization tasks. DDT has good performance and little computational resource requirement among these co-supervised methods. So we use DDT [26] as an example. Here is a brief recap of DDT. Given a set of images S with n images, where each image $I \in S$ has the same label, or contains the same object in the image. With a pre-trained model F, the final feature map is also generated: $G \in \mathbb{R}^{h \times w \times d} = \mathbb{R}^{hw \times d} = F(I)$. Then these feature maps are gathered together into a large feature set: $G_{all} \in \mathbb{R}^{n \times hw \times d} = \mathbb{R}^{nhw \times d}$. Principal component analysis (PCA) [12] is applied along the depth dimension. After the PCA process, we can get the eigenvector P with the largest eigenvalue. Then, each spatial location of G is channel-wise weighed and summed to get the final heat map H: $H_{i,j} = \sum_{k=1}^{d} G_{i,j,k} P_k$. Then H is upsampled to the original input size. Zero thresholding and max connected component analysis is applied to generate the final bounding box.

We will generate pseudo bounding boxes using both WSOL methods and the DDT method, and evaluate their suitability.

3.2.2 Localization Methods

After generating bounding boxes, we have (pseudo) bounding box annotations for each training image. Then it is natural to perform object localization with these generate boxes. As shown before, detection models are too heavy to handle this task. Thus, it is natural to perform bounding box regression. Previous fully supervised works [18, 17] suggest two methods of bounding box regression: single-class regression (SCR) and per-class regression (PCR). PCR is strongly related to the class label. Since we advocate that localization is a class-agnostic rather than a class-aware task, we choose SCR for all our experiments.

We follow previous work to perform bounding box regression [18]. Suppose the bounding box is in the x, y, w, h format, where x, y are the top-left coordinates of the bounding box and w, h are the width and height of the bounding box, respectively. We first transfer x, y, w, h into x^*, y^*, w^*, h^* where $x^* = \frac{x}{w_i}, y^* = \frac{y}{h_i}, w^* = \frac{w}{w_i}, h^* = \frac{h}{h_i}$, and w_i and h_i are the width and height of the input image, respectively. We use a sub-network with two fully connected layers and corresponding ReLU layers for regression. Finally, the outputs are sigmoid activated. We use the mean squared error loss (ℓ_2 loss) for the regression task.

Step 2 and step 3 in Algorithm 1 may be combined, i.e.,

F_{cls} and F_{loc} can be integrated into a single model, which is jointly trained with classification labels and generated bounding boxes. However, we will show empirically that localization and classification models should be separated.

4. Experiments

4.1. Experimental Setups

Datasets. We evaluate our proposed method on two common WSOL datasets: ImageNet-1k [15] and CUB-200 [25]. The ImageNet-1k dataset is a large dataset with 1000 classes, containing 1,281,197 training images and 50,000 validation images. For training images, bounding box annotations are incomplete, and bounding box labels are complete for validation images. In this paper, *we do not use any accurate training bounding box annotations.* In our experiments, we generate pseudo bounding boxes on training images by previous methods. The detailed ablation studies will be in Sec 5.1. We train all models on the generated bounding box annotations and classification labels and test them on the validation dataset.

For the CUB-200 dataset, it contains 200 categories of birds with 5,994 training images and 5,794 testing images. Each image in the dataset has an accurate bounding box annotation. We follow the strategies on ImageNet-1k to train and test models.

Metrics. We use three metrics for evaluating our models: Top-1/Top-5 localization accuracy (*Top-1/Top-5 Loc*) and localization accuracy with known ground truth class (*GT-Known Loc*). They are following previous state-of-the-art methods [30, 2]: *GT-Known Loc* is correct when given the ground truth class to the model, the intersection over union (IoU) between the ground truth bounding box and the predicted box is 50% or more. *Top-1 Loc* is correct when the Top-1 classification result and *GT-Known Loc* are both correct. *Top-5 Loc* is correct when given the Top-5 predictions of groundtruth labels and bounding boxes, there is one prediction which the classification result and localization result are both correct.

Base Models. We prepare several baseline models for evaluating our method on localization tasks: VGG16 [18], InceptionV3 [21], ResNet50 [6] and DenseNet161 [7]. Previous methods try to enlarge the spatial resolution of the feature map [28, 29, 2], we do not use this technology in our PSOL models. Previous WSOL methods need the classification weights to turn a 3D feature map into a 2D spatial heat map. However, in PSOL, we do not need the feature map for localization, our model will directly output the bounding box for object localization. For a fair comparison, we modified VGG16 into two versions: VGG-GAP and VGG16. VGG-GAP replaces all fully connected layers in VGG16 with GAP and a single fully connected layer, and VGG16 keeps the original structures in VGG16. For

other models, we keep the original structure of each model. For regression, we use a two-layer fully connected network with corresponding ReLU layers to replace the last layer in original networks, as illustrated in Sec 3.2.2.

Joint and Separate Optimization In the previous section, we discussed the problem of joint optimization of classification and localization tasks. For ablating this issue, we prepare several models for each base model. For joint optimization models, we add a new bounding box regression branch to the model (`-Joint` models), and then train this model with both generated bounding boxes and class labels simultaneously. For separate optimization models, we replace the classification part with the regression part (`-Sep` models), then train these two models separately, i.e., localization models are trained with only generated bounding boxes while classification models are trained with only class labels. The hyperparameters are kept same for all models.

4.2. Implementation Details

We use the PyTorch framework with TitanX Pascal GPUs support. For all models, we use pre-trained classification weights on ImageNet-1k and fine-tune on target localization and classification tasks.

For experiments on ImageNet-1k, the hyperparameters are set the same for all models: batch size 256, 0.0005 weight decay and 0.9 momentum. We will fine-tune all models with a start learning rate of 0.001. Added components (like the regression sub-network) will have a larger learning rates due to the random initialization. We train 6 epochs on ImageNet and 30 epochs on CUB-200. For localization only tasks, we keep the learning rate fixed among all eppochs. The reason is that DDT generated bounding boxes are noisy, which contain many inaccurate or even totally wrong bounding boxes. The conclusion in [23] shows that for noisy data, we should retain large learning rates. For classification related tasks (including single classification and joint classification and localization tasks), we divide the learning rate by 10 every 2/10 epochs on ImageNet/CUB-200.

For testing models, we use ten crop augmentations on ImageNet to output results of the final classification following [28] and [29] on ImageNet and single crop classification results on CUB200, and use single image inputs for all our localization results. We use the center crop techniques to get the image input, e.g., resize to 256×256 then center crop to 224×224 for most models except InceptionV3 (resize to 320x320 then center crop to 299×299), following the setup in [2, 27]. For state-of-the-art classification models, we also follow the input size in their paper, e.g., 600 for EfficientNet-B7.

Previous WSOL methods can provide multiple boxes for a single image with different labels. However, our SCR model can only provide one bounding box output for

Table 1: The *GT-Known Loc* accuracy on the ImageNet-1k validation dataset of various weakly and co-supervised localization (DDT) methods.

Model	ImageNet-1k	CUB-200
VGG16-CAM [30]	59.00	57.96
VGG16-ACoL [28]	62.96	59.30
SPG [29]	64.69	60.50
DDT-ResNet50 [26]	59.92	72.39
DDT-VGG16 [26]	61.41	84.55
DDT-InceptionV3 [26]	51.87	51.80
DDT-DenseNet161 [26]	61.92	78.09

each image. Thus, we combine the output bounding box with Top-1/Top-5 classification outputs of baseline models (`-Sep` models) or with outputs of the classification branch (`-Joint` models) to get the final output to evaluate on test images.

For experiments on CUB-200, we change the batch size from 256 to 64, and keep other hyperparameters the same as ImageNet-1k.

5. Results and Analyses

In this section, we will provide empirical results, and perform detailed analyses on them.

5.1. Ablation Studies on How to Generate Pseudo Bounding Boxes

Previous WSOL methods can generate bounding boxes with given ground truth labels. Some co-localization methods can also provide bounding boxes with a given class label. Since some annotations are missing in ImageNet-1k training images, we test these methods on the validation/test set of ImageNet-1k and CUB-200 to choose a better method to generate pseudo bounding boxes for PSOL. For the DDT method, we first resize the training images to the resolution size of 448×448, then perform DDT on training images. According to the statistics collected on training images, we generate bounding boxes on test images with the correct class label. For other WSOL methods, we follow original instructions in their papers and use pre-trained models to generate bounding boxes on validation/test images with the correct class label.

We list the *GT-Known Loc* of DDT and weakly supervised localization methods in Table 1. As shown in Table 1, DDT achieves comparable results with WSOL methods on ImageNet-1k, but achieves better performance than all WSOL methods on CUB-200. DDT results on CUB-200 indicate that object localization should not be related to classification labels. Furthermore, these WSOL methods need large computational resources, e.g., storing the feature map of each image, then perform off-line CAM operation to

Table 2: Empirical localization accuracy results on CUB-200 and ImageNet-1k. The first column of the paper shows the model name, and the second column shows the backbone network for each model. Parameter number and FLOPs are shown in the third and fourth column. Then *Top-1/Top-5 Loc* accuracy of CUB-200 and ImageNet-1k are shown in the next four columns. The last column illustrates the *GT-Known Loc* accuracy on ImageNet-1k. For separate models like DDT and our `-Sep` models, we combine their localization results with classification results of baseline models. For FLOPs calculation, we only calculate convolutional operations as FLOPs and using networks on ImageNet as counting examples. Results with bold are best among the same backbone networks.

Model	Backbone	Parameters	FLOPs	CUB-200		ImageNet-1k		
				Top-1 Loc	Top-5 Loc	Top-1 Loc	Top-5 Loc	GT-Known Loc
VGG16-CAM [30]	VGG-GAP	14.82M	15.35G	36.13	-	42.80	54.86	59.00
VGG16-ACoL [28]	VGG-GAP	45.08M	43.32G	45.92	56.51	45.83	59.43	62.96
ADL [2]	VGG-GAP	14.82M	15.35G	52.36	-	44.92	-	-
VGG16-Grad-CAM [16]	VGG16	138.36M	15.42G	-	-	43.49	53.59	-
CutMix [27]	VGG-GAP	138.36M	15.35G	52.53	-	43.45	-	-
DDT-VGG16 [26]	VGG16	138.36M	15.42G	62.30	78.15	47.31	58.23	61.41
PSOL-VGG16-Sep	VGG16	274.72M	30.83G	**66.30**	**84.05**	**50.89**	**60.90**	**64.03**
PSOL-VGG16-Joint	VGG16	140.46M	15.42G	60.07	75.35	48.83	59.00	62.1
PSOL-VGG-GAP-Sep	VGG-GAP	29.64M	30.70G	59.29	74.88	48.36	58.75	63.72
PSOL-VGG-GAP-Joint	VGG-GAP	15.08M	15.35G	58.39	72.64	47.37	58.41	62.25
SPG [29]	InceptionV3	38.45M	66.59G	46.64	57.72	48.60	60.00	64.69
ADL [2]	InceptionV3	38.45M	66.59G	53.04		-	48.71	-	-
PSOL-InceptionV3-Sep	InceptionV3	53.32M	11.42G	**65.51**	**83.44**	**54.82**	**63.25**	**65.21**
PSOL-InceptionV3-Joint	InceptionV3	29.21M	5.71G	60.32	78.98	52.76	61.10	62.83
ResNet50-CAM [30]	ResNet50	25.56M	4.10G	29.58	37.25	38.99	49.47	51.86
ADL [2]	ResNet50-SE	28.09M	6.10G	62.29	-	48.53	-	-
CutMix [27]	ResNet50	26.61M	4.10G	54.81	-	47.25	-	-
PSOL-ResNet50-Sep	ResNet50	50.12M	8.18G	**70.68**	**86.64**	**53.98**	**63.08**	**65.44**
PSOL-ResNet50-Joint	ResNet50	26.61M	4.10G	68.17	83.69	52.82	62.00	64.30
DenseNet161-CAM	DenseNet161	29.81M	7.80G	29.81	39.85	39.61	50.40	52.54
PSOL-DenseNet161-Sep	DenseNet161	56.29M	15.46G	**74.97**	**89.12**	**55.31**	**64.18**	**66.28**
PSOL-DenseNet161-Joint	DenseNet161	29.81M	7.80G	74.24	87.03	54.48	63.41	65.39

get the final bounding boxes. Compared to these methods, DDT has little computation requirements and achieves comparable results. For the base model choice of DDT, though DDT-DenseNet161 has higher accuracy than DDT-VGG16 on ImageNet-1k, it runs much slower due to the dense connections and has lower accuracy than DDT-VGG16 on CUB-200. Based on these observations, we choose DDT with VGG16 to generate our bounding boxes on training images in PSOL.

5.2. Comparison with State-of-the-art Methods

In this section, we will compare our PSOL models with state-of-the-art WSOL methods: CAM [30], HaS [19], ACoL [28], SPG [29] and ADL [2] on CUB-200 and ImageNet-1k.

We list experimental results in Table 2. Furthermore, we visualize bounding boxes generated by CAM [30], DDT [26] and our methods in Fig. 2. According to these results, we have the following findings.

- Without any training, DDT already performs well on both CUB-200 and ImageNet. DDT-VGG16 achieves 47.31% *Top-1 Loc* accuracy, which has a 2∼3% edge over WSOL models based on VGG16. Since DDT is a class-agnostic method, it suggests that WSOL should be divided into two independent sub-tasks: class-agnostic object localization and object classification.

- All PSOL models with separate training perform better than PSOL models with joint training. In all five baseline models, `-Sep` models consistently perform better than `-Joint` models by large margins. These results indicate that learning with joint classification and localization is not suitable.

- All our PSOL models enjoy a large edge (mostly > 5%) on CUB-200 compared with state-of-the-art WSOL methods, including the DDT-VGG16 method. CUB-200 is a fine-grained dataset which contains many categories of birds. The within-class varia-

CAM

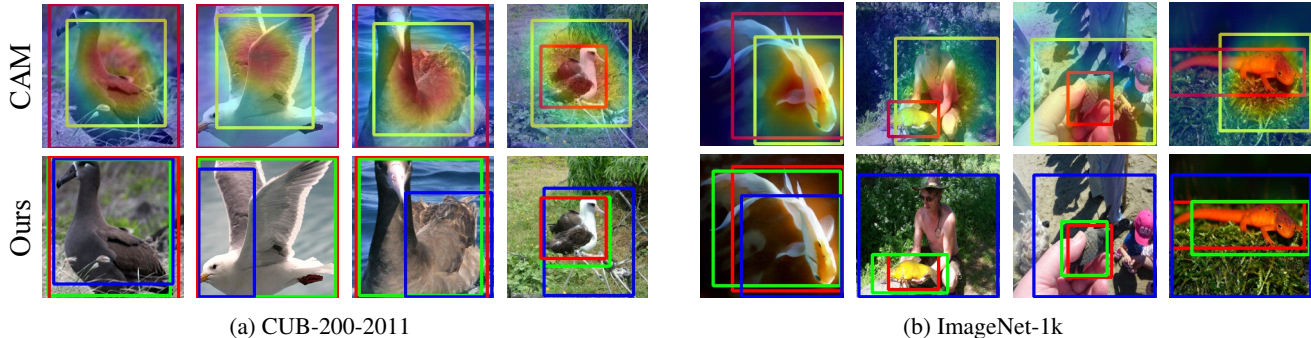Ours

(a) CUB-200-2011

(b) ImageNet-1k

Figure 2: Comparison of our methods with CAM and DDT. Please note that in CAM figures, yellow boxes are CAM predicted boxes and red boxes are groundtruth boxes. In figures of our methods, blue boxes are DDT generated boxes, green boxes are predicted boxes by our regression model and red boxes are groundtruth boxes. We use the DenseNet161-Sep model to output DDT and predict boxes. This figure is best viewed in color and zoomed in.

tion is much larger than the between-class variation in most fine-grained datasets [25]. The exact label may not help the process of localization. Hence, the co-localization method DDT will perform better than previous WSOL methods.

- CNN has the ability to tolerate some incorrect annotations, and retain high accuracy on validation sets. For all separate localization models, the *GT-Known Loc* is higher than DDT-VGG16. This phenomenon indicates that CNN can tolerate some annotation errors and learn robust patterns from noisy data.

- Some restrictions and rules-of-thumb in WSOL do not carry over to PSOL. In previous WSOL papers, only one final fully connected layer is allowed, and large spatial size of the output feature map is recommended. Many methods try to remove the stride of the last downsample convolution layer, which will result in large FLOPs (such as SPG and VGG16-ACoL). Besides this, three fully connected layers in VGG16 are all removed, which will directly affect the accuracy. However, in our experiments, VGG-Full performs significantly better than VGG-GAP. Since CAM requires GAP and only one FC layer, when this restriction is removed, VGG16 can get better performance. Another restriction is the inference path of the network. WSOL needs the output of the last convolutional layer in the model, and often uses simple forward networks (VGG16, GoogLeNet, and InceptionV3). Complex network structures like DenseNet are not recommended and do not perform well in the WSOL problem [30]. As we show in Table 2, CAM achieves poor performance with DenseNet161. DenseNet will use features of every block, not just the last feature to conduct classification. Thus, the semantic meaning of the last feature may not as clear as the last feature of

sequential networks like ResNet and VGG. However, PSOL-DenseNet models are directly trained on noisy bounding boxes, which can avoid this problem. Moreover, DenseNet161 achieves the best performance.

5.3. Transfer Ability on Localization

In this section, we will discuss the transferability of different localization models.

Previous weakly supervised localization models need the exact label to generate bounding boxes, regardless of the correctness of the label. However, our proposed method does not need the label and directly generate the bounding box. So we are interested in that: Is single object localization task transferable? Does the model trained directly on object localization tasks like trained on image recognition tasks, have good generalization ability?

We perform the following experiment. We take object localization models trained on ImageNet-1k, then predict on CUB-200 test images directly, i.e., without any training or fine-tuning process. We add previous WSOL methods for a fair comparison. Since they need exact labels, we fine-tune all these models. For all models marked with *, they are only fine-tuned with classification parts (the last fully connected layer), i.e., features learned on ImageNet-1k are directly transferred to CUB-200. For models marked without *, they are fine-tuned on CUB-200 with all layers. We take our VGG-GAP-Sep model for fair comparison and DenseNet161-Sep model for better results. Results are in Table 3.

It is surprising that without any supervision, PSOL object localization models can transfer well from ImageNet-1k to CUB-200, which performs significantly better than previous WSOL methods, including models which only fine-tune the classification weight (models marked with *), and models which fine-tune the whole weights. It further indicates that objection localization is not dependent on classifica-

Table 3: Transfer results of our models on CUB-200 and ImageNet-1k. For a fair comparison, we add VGG-GAP with CAM, VGG16-ACoL [28] and SPG [29] for transfer experiments. VGG-GAP is fine-tuned with all layers while VGG-GAP* only fine-tunes the final fully connected layer. Please note that PSOL models trained on ImageNet-1k do not have any training or fine-tuning process on CUB-200.

Model	Trained	Target	GT-Known Loc
VGG-GAP + CAM	CUB-200	CUB-200	57.96
VGG-GAP* + CAM	ImageNet	CUB-200	57.53
VGG16-ACoL + CAM	CUB-200	CUB-200	59.30
VGG16-ACoL* + CAM	ImageNet	CUB-200	58.70
SPG + CAM	CUB-200	CUB-200	60.50
SPG* + CAM	ImageNet	CUB-200	59.70
PSOL-VGG-GAP-Sep	CUB-200	CUB-200	80.45
PSOL-VGG-GAP-Sep	ImageNet	CUB-200	89.11
PSOL-DenseNet161-Sep	CUB-200	CUB-200	92.54
PSOL-DenseNet161-Sep	ImageNet	CUB-200	92.07

Table 4: Top-1 and Top-5 Loc results by combining localization of our models with more state-of-the-art classification models on ImageNet-1k.

Model	Top-1	Top-5
VGG16-ACoL+DPN131	53.94	61.15
VGG16-ACoL+DPN-ensemble	54.86	61.45
SPG + DPN131	55.19	62.76
SPG + DPN-ensemble	56.17	63.22
PSOL-InceptionV3-Sep + DPN131	55.72	63.64
PSOL-DenseNet161-Sep + DPN131	56.59	64.63
PSOL-InceptionV3-Sep + EfficientNet-B7	57.25	64.04
PSOL-DenseNet161-Sep + EfficientNet-B7	58.00	65.02

tion, and it is inessential to perform object localization with the class label. Furthermore, it proves the advantage of our PSOL method.

5.4. Combining with State-of-the-art Classification

Previous methods try to combine localization outputs with state-of-the-art classification outputs to achieve better localization results. SPG [29] and ACoL [28] combine with DPN networks including DPN-98, DPN-131 and DPN-ensemble [1]. For a fair comparison, we also combine other models' (InceptionV3 and DenseNet161) results with DPN-131. Moreover, EfficientNet [22] achieves better results on ImageNet-1k recently. We combine our localization outputs with EfficientNet-B7's classification outputs. Results are in Table 4.

From the table we can see that our model achieves better localization accuracy on ImageNet-1k compared with SPG [29] and ACoL [28] when combining the same classification results from DPN131 [1]. Furthermore, when combining with EfficientNet-B7 [22], we can achieve 58.00%

Table 5: Compare our method with state-of-the-art fully supervised methods on ImageNet-1k validation datasets.

Model	supervision	Top-5 Loc
GoogLeNet-GAP [30]	weak	57.1
GoogLeNet-GAP (heuristics) [30]	weak	62.9
VGG16-Sep	weak	60.9
DenseNet161-Sep	weak	64.2
GoogLeNet [20]	full	73.3
OverFeat [17]	full	70.1
AlexNet [8]	full	65.8
VGG16 [18]	full	70.5
VGGNet-ensemble [18]	full	73.1
ResNet + Faster-RCNN-ensemble [14]	full	90.0

Top-1 localization accuracy.

5.5. Comparison with fully supervised methods

We also compare our PSOL with fully supervised localization methods on ImageNet-1k. Fully supervised methods use training images with accurate bounding box annotations in ImageNet-1k to train their models. Results are in Table 5.

With the bounding box regression sub-network, our DenseNet161-Sep model can roughly match fully supervised AlexNet with Top-5 Loc accuracy. However, our performances are still worse than fully supervised OverFeat, GoogLeNet and VGGNet. It is noticeable that ResNet + Faster-RCNN-ensemble [14] achieves the best Top-5 Loc accuracy. They transfer region proposal networks trained on ILSVRC detection track, which has 200 classes of fully labeled images, to the 1000-class localization tasks directly. The region proposal network shows good generalization ability among different classes without fine-tuning, which indicates that localization is separated with classification.

6. Discussions and Conclusions

In this paper, we proposed the pseudo supervised object localization (PSOL) to solve the drawbacks in previous weakly supervised object localization methods. Various experiments show that our methods obtain a significant edge over previous methods. Furthermore, our PSOL methods have good transfer ability across different datasets without any training or fine-tuning.

For future works, we will try to dive deep into the joint classification and localization problem: We will try to integrate both tasks into a single CNN model with less localization accuracy drop. Another direction is trying to improve the quality of generating bounding boxes with class-agnostic methods. Finally, novel network structures or algorithms on localization problems should be found, which should prevent the high input resolution and computational resources in the current detection framework to apply to large-scale datasets.

References

[1] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. In *NIPS*, pages 4467–4475, 2017. 8

[2] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *CVPR*, pages 2219–2228, 2019. 1, 3, 4, 5, 6

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 1

[4] Ross Girshick. Fast R-CNN. In *ICCV*, pages 1440–1448, 2015. 2

[5] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[7] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017. 4

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 2, 8

[9] Hei Law and Jia Deng. CornerNet: Detecting objects as paired keypoints. In *ECCV*, volume 11218 of *LNCS*, pages 734–750, 2018. 2

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, volume 8693 of *LNCS*, pages 740–755, 2014. 1, 3

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, volume 9905 of *LNCS*, pages 21–37, 2016. 2

[12] Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. 4

[13] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 2

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 2, 3, 8

[15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 1, 4

[16] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017. 2, 6

[17] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, pages 1–15, 2014. 2, 4, 8

[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, pages 1–14, 2015. 2, 4, 8

[19] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, pages 3544–3553, 2017. 1, 3, 6

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 8

[21] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 4

[22] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114, 2019. 1, 8

[23] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, pages 5552–5560, 2018. 5

[24] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 2, 3

[25] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 1, 4, 7

[26] Xiu-Shen Wei, Chen-Lin Zhang, Jianxin Wu, Chunhua Shen, and Zhi-Hua Zhou. Unsupervised object discovery and co-localization by deep descriptor transformation. *Pattern Recognition*, 88:113–126, 2019. 1, 4, 5, 6

[27] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, page in press, 2019. 1, 5, 6

[28] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, pages 1325–1334, 2018. 1, 3, 4, 5, 6, 8

[29] Xiaolin Zhang, Yunchao Wei, Guoliang Kang, Yi Yang, and Thomas Huang. Self-produced guidance for weakly-supervised object localization. In *ECCV*, volume 11216 of *LNCS*, pages 610–625, 2018. 1, 3, 4, 5, 6, 8

[30] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 1, 2, 3, 4, 5, 6, 7, 8

[31] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2

[32] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, volume 8693 of *LNCS*, pages 391–405, 2014. 3