ML-LocNet: Improving Object Localization with Multi-view Learning Network

Xiaopeng Zhang[1][0000−0001−6337−5748], Yang Yang[2], and Jiashi Feng[1]

[1] National University of Singapore
{elezxi,elefjia}@nus.edu.sg
[2] University of Electronic Science and Technology of China
dlyyang@gmail.com

Abstract. This paper addresses Weakly Supervised Object Localization (WSOL) with only image-level supervision. We propose a **M**ulti-view **L**earning **Loc**alization **Net**work (ML-LocNet) by incorporating multi-view learning into a two-phase WSOL model. The multi-view learning would benefit localization due to the *complementary* relationships among the learned features from different views and the *consensus* property among the mined instances from each view. In the first phase, the representation is augmented by integrating features learned from multiple views, and in the second phase, the model performs multi-view co-training to enhance localization performance of one view with the help of instances mined from other views, which thus effectively avoids early fitting. ML-LocNet can be easily combined with existing WSOL models to further improve the localization accuracy. Its effectiveness has been proved experimentally. Notably, it achieves 68.6% CorLoc and 49.7% mAP on PASCAL VOC 2007, surpassing the state-of-the-arts by a large margin.

Keywords: Weakly Supervised Learning, Object Localization, Multi-view Learning, Object Instance Mining

1 Introduction

In this paper, we tackle object localization under a weakly supervised paradigm, where *only image-level labels* indicating the presence of an object are available for training. Current Weakly Supervised Object Localization (WSOL) methods [7], [15], [17], [26] usually model the missing object locations as latent variables, and follow a two-phase learning procedure to infer the object locations. The first phase generates object candidates for latent variable initialization, and the second phase refines the candidates to optimize the localization model. Among these optimization strategies, a typical solution is to alternate between model re-training and object re-localization, which shares a similar spirit with Multiple Instance Learning (MIL) [5], [17], [26]. Nevertheless, such optimization is non-convex and easy to get stuck in local minima if the latent variables are not properly initialized. As a result, they suffer limited performance, which is far below that of their fully supervised counterparts [13], [20], [23].

In this paper, we propose a **M**ulti-view **L**earning **Loc**alization **Net**work (ML-LocNet) by incorporating multi-view learning into a two-phase WSOL model. Our approach is motivated by the favorable properties in multi-view learning. In particular, there exist *complementary* relationships among the learned features from different views. Combining these features is beneficial to accurately describe the target regions. Besides, through co-training localization models from multiple views, different object instances would be mined, and such differences could be exploited to mitigate the over-fitting issue. In order to approach *consensus* of multi-view co-training, the detectors from multiple views would become increasingly similar as the co-training process proceeds, until the performance cannot be further improved.

Our proposed ML-LocNet is able to consistently improve the localization accuracy during each phase by exploiting the two properties of multi-view learning. To enable multi-view representation, we intentionally branch the network into several streams with different transforms, such that each stream can be treated as one view over the input. For the first phase, ML-LocNet augments the feature representation by concatenating features learned from multiple views. Benefiting from the complementary property, each view may contain knowledge that other views do not have, and combining them is beneficial to accurately describe the target regions. For the second phase, we propose a multi-view co-training algorithm which simultaneously trains multiple localization models from multiple views. The models are optimized by mining object instances on some views, and training the corresponding detectors on other views. In this way, ML-LocNet avoids overfitting the initial instances and is able to learn more general models. The two-phase learning procedure produces a powerful localization framework. On benchmark dataset PASCAL VOC 2007, we achieve 68.6% localization and 49.7% detection accuracy under weakly supervised paradigm, surpassing the-state-of-the-arts by a large margin.

It is interesting to compare our multi-view learning strategy with multi-fold MIL [5], a variant of MIL. Multi-fold MIL is reminiscent of K-fold cross validation, which proceeds by dividing the training images into K disjoint folds, and re-localize the objects in each fold images using a detector trained from images in the other folds. In this way, Multi-fold MIL avoids convergence to poor local optima. However, such cross validation method reduces the amount of effective training data, and significantly increases the complexity since we need to train K models during each iteration. As a result, it is impractical to apply multi-fold MIL to train deep networks. Instead of dividing the training data, we construct different views of the same image, which avoids overfitting the initial samples without sacrificing the amount of training data, and makes the training on deep networks tractable by sharing features over the lower layers.

Our method is also loosely related with model ensemble, a simple way to improve the performance of deep networks. Model ensemble aims at training different models on the same data and then averaging their predictions. However, making predictions using model ensemble is cumbersome and computationally expensive. Instead of training different models, we improve the performance by training models on different views. Our model can be treated as a kind of model ensemble which shares representation at the lower layers and formulates a multi-task learning by training multiple models simultaneously. In this manner, we are able to compress the knowledge in an ensemble into a single model which does not increase the complexity during training and test stages.

To sum up, we make following contributions. First, we propose an ML-LocNet model, which exploits the *complementary* and *consensus* properties of multiple views, to consistently improve the localization accuracy in WSOL. Second, ML-LocNet is independent of the backbone architecture and can be easily combined with existing WSOL models [1], [15], [17], [26] to further improve the localization accuracy. Using the WSDDN [1] and Fast-RCNN [13] as backbone architectures, we achieve 68.6% localization CorLoc and 49.7% detection mAP on PASCAL VOC 2007 benchmark, which surpasses the state-of-the-arts by a large margin.

2 Related Work

2.1 Weakly Supervised Localization

Most related works formulate WSOL as Multiple Instance Learning (MIL) [9], in which labels are assigned to bags (a group of instances) rather than an individual instance. These methods typically alternate between learning a discriminative representation of the object and selecting the object samples in positive images based on this representation [5], [21], [27]. However, MIL is sensitive to initialization and easy to get stuck in a local minimum. To solve this issue, many works focus on improving the initialization. Song *et al.* [25] proposed a constrained submodular algorithm to identify an initial set of image windows and part configurations that are likely to contain the target object. Wang *et al.* [28] proposed to discover the latent objects/parts via probabilistic latent semantic analysis on the windows of positive samples. Li *et al.* [18] combined the activations of CNN with the association rule mining technique to discover representative mid-level visual patterns.

Another line of works tries to perform weakly supervised localization in an end-to-end manner. Oquab *et al.* [22] proposed a weakly supervised object localization method by explicitly searching over candidate object locations at different scales during training. Zhou *et al.* [29] leveraged a simple global average pooling to aggregate class-specific activations, which can effectively localize the discriminative parts. Zhu *et al.* [30] incorporated a soft proposal module into a CNN to guide the network to focus on the discriminative visual evidence. However, these works are all based on aggregating pixel-level confidences for image-level classification, which tend to focus on the discriminative details and fail to distinguish multiple objects from the same category. Differently, Bilen [1] *et al.* proposed a WSDDN model, which aggregates region-level scores for image-level loss and conveniently enables detection based on the region scores. Based on WSDDN model, some works exploits context information [16] and MIL refinement [26] to further improve the localization. Nevertheless, these methods also cannot get out of being trapped in discriminative parts.

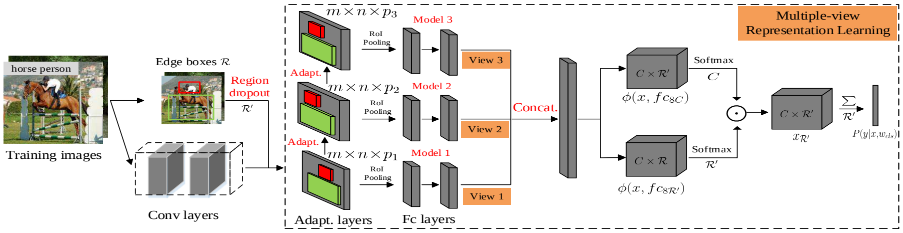

Fig. 1. Illustration of the ML-LocNet architecture during the first phase learning. The network is branched into three views with different convolutional parameters, and these views are concatenated for final feature representation

2.2 Multi-view Learning

Multi-view learning is a classical semi-supervised learning algorithm, which represents examples using different feature sets or different views. By taking advantage of the consensus and complementary principles of multiple view representations, learning models from multi-views will lead to an improvement in learning. Blum *et al.* [2] first proposed the multi-view learning strategy and applied it to the web document classification. Feng *et al.* [12] tackled the image annotating problem by combining multi-view co-training and active learning. Considering the complementary characteristic of different kinds of visual features, such as color and texture features, Chen *et al.* [4] introduced a co-training algorithm to conduct relevance feedback in content-based image retrieval.

3 ML-LocNet: Architecture and Training

In this section we introduce the proposed ML-LocNet with multi-view learning strategy. The model learning consists of two phases. In the first phase, we augment region representation with multi-view feature concatenation, and mine object instances with image-level supervision. In the second phase, we develop a multi-view co-training algorithm to refine the localization model with the mined instances from the first phase. In both phases, we adapt three views for implementation. The two phase procedures will be elaborated as follows.

3.1 Phase One: Multi-view Representation Learning

In the first phase, the localization network aims at mining high quality object instances with only image-level labels. As shown in Fig. 1, the model is based on a feed-forward convolutional network that aggregates region scores for computing classification loss. The early network layers are based on a pretrained network for classification, truncated before any fully connected layers, which we call the base network. Given an image, the base network takes the entire image as input and applies a sequence of convolutional and pooling layers, giving feature maps at the last convolutional block (known as *conv5*). We then add multi-view representation learning components to the base network to improve localization with the following key features:

Multi-view Features. To enable the multi-view feature representation learning, we divide the base network into three branches, with each branch representing one view. We intentionally design different convolutional parameters for different views to ensure the diversity. In particular, we add a convolutional block to each view, which we refer to as the view adaptation block, and follow the fully connected layers with different parameters. Formally, for a feature layer of size $m \times n$ with p_I channels, the view adaptation block is a $3 \times 3 \times p_I \times p_O$ small kernel which produces an output feature map with size $m \times n \times p_O$. The channel p_O is configured to be compatible with the first fully connected layer of that view. Then each view is followed by an ROI pooling layer [13] which projects region proposals $\mathcal{R}$ on the image to the feature maps and produces fixed length vectors $\phi_i(x, \mathcal{R})$, $i = 1, 2, 3$. The features among different views are weightedly combined to form the final representation $\phi(x, \mathcal{R}) = [\alpha_1 \phi_1(x, \mathcal{R})\ \alpha_2 \phi_2(x, \mathcal{R})\ \alpha_3 \phi_3(x, \mathcal{R})]$, where α_i $(i = 1, 2, 3)$ is the weight factor that balances each view, and is automatically learned by the network.

Two-stream Network. In the first phase learning, the only supervision is the image-level labels. We need to combine the region-level features with image-level classification. To this end, we employ the two-stream architecture of WSDDN [1], which explicitly computes image-level classification loss via aggregating region proposal scores. Formally, given an image x with region proposal $\mathcal{R}$ and image-level label $y \in \{1, -1\}^C$, where $y_c = 1$ ($y_c = -1$) indicates the presence (absence) of an object class c. The concatenated output $\phi(x, \mathcal{R})$ is branched into two data streams fc_{8C} and fc_{8R} to obtain the category specific scores. Denote the output of fc_{8C} and fc_{8R} layer as $\phi(x, fc_{8C})$ and $\phi(x, fc_{8R})$, respectively, which is of size $C \times |\mathcal{R}|$. Here, C represents the number of categories and $|\mathcal{R}|$ denotes the number of regions. The score of region r corresponding to class c is the dot product of $\phi(x, fc_{8C})$ and $\phi(x, fc_{8R})$, normalized at different dimensions:

$$x_{cr} = \frac{e^{\phi^{cr}(x, fc_{8C})}}{\sum_{i=1}^{C} e^{\phi^{ir}(x, fc_{8C})}} .* \frac{e^{\phi^{cr}(x, fc_{8R})}}{\sum_{j=1}^{|\mathcal{R}|} e^{\phi^{cj}(x, fc_{8R})}}. \tag{1}$$

Thus we obtain the region score x_{cr} representing the probability of region r belonging to category c. Based on x_{cr}, the probability output y w.r.t. category c is defined as the sum of region-level scores $\phi^c(x, w_{cls}) = \sum_{j=1}^{|\mathcal{R}|} x_{cj}$, where w_{cls} denotes the non-linear mapping from input x to classification output. This network is trained by back-propagating a binary log loss, denoted as

$$L_{cls}(x, y) = \sum_{i=1}^{C} \log(y_i(\phi^i(x, w_{cls}) - 1/2) + 1/2), \tag{2}$$

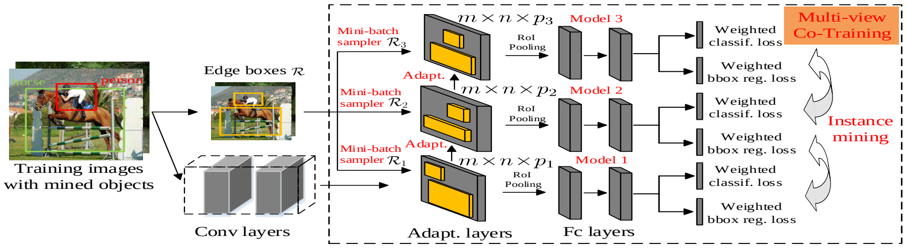

Fig. 2. Illustration of the ML-LocNet architecture during the second phase learning. Given the mined object instances, we refine the localization network via a multi-view co-training strategy. The network performs a multi-task learning procedure and is optimized via iteratively mining new object instances among views

Region Dropout. The two-stream network employs a softmax operator to normalize scores of different regions (r.f. Eqn. (1)), and is able to pick out the one that contains the most salient region. However, since the network is trained with classification loss, the high-score regions tend to focus on object parts instead of the whole object. As a result, the network would quickly converge to local minima due to overfitting a few region proposals. To solve this issue, we introduce a simple region dropout strategy to avoid overfitting. During forward propagation, we perform random dropout on region proposals $\mathcal{R}$, and only pass part of regions $\mathcal{R}'$ to the ROI pooling layers. The advantages of using region dropout are two-folds. First, the network is able to pick up different combinations of regions, which can be treated as some sort of data augmentation and effectively avoid network overfitting. Second, fewer regions at the fully connected layers can efficiently reduce computation and accelerate the training process.

3.2 Phase Two: Multi-view Co-training

In the first phase localization, the network is trained by image-level loss, which inevitably focuses on object parts or groups of multiple objects from the same category. To solve this issue, we introduce a multi-view instance refinement procedure, which trains the network with instance-level loss, and refines the network via a multi-view co-training strategy. The principle is that among the mined object instances, the majority of them are reliable. We hope to transfer the successful localization to those failure ones via a multi-view co-training strategy. The second phase learning is based on the fast-RCNN [13] framework, but with the following adjustments to improve the localization performance.

Initial Object Instances. The first phase learning returns a series of region scores representing their probabilities containing target objects. Since the top scored region easily focuses on object parts, we do not over-optimistically consider the top scored region to be accurate enough. Instead, we consider them to be accurate enough as soft voters. To be specific, given a training image containing class c, we compute its object heat map H^c, which collectively returns the

Algorithm 1 ML-LocNet for Weakly Supervised Object Localization

```
Input: Training set $\mathcal{D}=\{x_i\}_{i=1}^N$ with image labels $Y=\{y_i\}_{i=1}^N$, iteration times $K$;
  First Phase Learning: Given an image $x$ with region proposals $\mathcal{R}$:
    1) Perform region dropout $\mathcal{R}\rightarrow\mathcal{R}'$;
    2) Obtain multi-view features $\phi(x,\mathcal{R}')=[\alpha_1\phi_1(x,\mathcal{R}')\ \alpha_2\phi_2(x,\mathcal{R}')\ \alpha_3\phi_3(x,\mathcal{R}')]$;
    3) Train initial localization network with multi-view features $\phi(x,\mathcal{R}')$;
  Second Phase Learning: Given an image $x$ with region proposals $\mathcal{R}$ and mined
  object instances $x^o$, initialize $x^{o_i}=x_o$ for each view $V_i$:
  for iter $k=1$ to $K$ do
    1). Perform random mini-batch sampling $\mathcal{R}\rightarrow\mathcal{R}_i$ for view $V_i$;
    2). Training localization model $M_k$ with current object instances $x^{o_i}$ for view $V_i$;
    3). Relocalize object instances for view $V_i$ with trained model on views $\{V \setminus V_i\}$;
  end for
Output: Localization model $M_K$.
```

confidence that pixel p lies in an object, *i.e.,* $H^c(p) = \sum_r x_{cr} D_r(p)/Z$, where $D_r(p)=1$ when the r-th region contains pixel p, and Z is a normalization constant such that $\max H^c(p) = 1$. We binarize the heat map H^c with threshold T (set as 0.5), and choose the tightest bounding box that encloses the largest connect component as the mined object instance.

Multi-view Co-training. Due to lack of object annotations, the mined object instances inevitably include false positive samples. Current approaches [17], [26] simply treat these noisy pseudo annotations as ground truths, which are suboptimal and easy to overfit the bad initial seeds. This issue is especially critical for a deep network due to its high fitting capacity. To overcome this issue, we propose a multi-view co-training strategy which aims at mining object instances on some views, while training the localization model on other view. In this way, we are able to effectively avoid overfitting the initial seeds.

The essence of the multi-view co-training module is to design multiple views and ensure the diversity of multi-view outputs. To this end, firstly we intentionally construct different views with different convolutional parameters, which is similar with the first phase learning. The difference is that instead of concatenating multi-view features to enhance representation, we independently model the outputs of each view with instance-level loss, and formulate the network training as a multi-task learning procedure. Secondly, we introduce random mini-batch sampling for each view, such that different views can see different positive and negative samples during the same forward propagation. The above two designs ensure the diversity among views not only in network structure but also in training samples, and we hope that different views are endowed with different localization capacities. For instance mining, the mean localized outputs of any two views are used for the mined object instance of the rest view during the next training. Based on the consensus principle of multiple-view co-training, the localization models from multiple views will become increasingly similar as the co-training process proceeds, until the performance cannot be further improved.

Weighted Loss. Due to the varying complexity of images, the mined object instances cannot be all reliable. It is suboptimal to treat all these instances equally important. Therefore, we penalize the network outputs with weighted loss, considering the reliability of the mined instances. Formally, let x_c^o be the relocalized object with label $y_c^o = 1$, and $\phi^c(x_c^o, w_{loc}^k)$ be the localization score returned by network M^k, where w_{loc}^k is the network parameter of M^k. The weighted classification loss w.r.t. x_c^o in the next retraining step is defined as

$$L_{cls}(x_c^o, y_c^o, M_{k+1}) = -\phi^c(x_c^o, w_{loc}^k) \log \phi^c(x_c^o, w_{loc}^{k+1}). \tag{3}$$

We employ the weighted loss on both classification and bounding box regression branches, as shown in Fig. 2. The whole algorithm of two-phase learning is summarized in Algorithm 1.

4 Experiments

We evaluate ML-LocNet for weakly supervised localization and detection, providing extensive design evaluation and making comparison with state-of-the-arts.

4.1 Experimental Setup

Datasets and Evaluation Metrics. We evaluate our approach on three widely used detection benchmarks: 1) PASCAL VOC 2007 [11], containing totally 9,963 images of 20 object classes, of which 5,011 images are included in *trainval* and the rest 4,952 in *test*; 2) PASCAL VOC 2012 [10] that is an extended version of PASCAL VOC 2007 and contains 11,540 images for *trainval* and 10,991 images for *test*; 3) Microsoft COCO 2014 [19], a large scale dataset that contains over 135k images spanning 80 categories, of which around 80k images are used for *train* and around 40k for *val*. For PASCAL VOC datasets, We choose the *trainval* split for training, and the *test* split for test, while for MS COCO, we choose the *train* split for training, and the *val* split for test. For performance evaluation, two kinds of measurements are used: 1) CorLoc [6] evaluated on the training set; 2) the VOC protocol which measures detection performance with average precision (AP) on the test set.

Models. We choose three models to evaluate our approach: 1) VGG-CNN-F [3], denoted as model **S**, meaning "small"; 2) VGG-CNN-M [3], denoted as model **M**, for "medium"; 3) VGG-VD [24] (the 16-layer model), denoted as model **L**, meaning "large". The base network is initialized from each pretrained model, truncated before any fully connected layers, while the multi-view module is initialized from the fully connected layers of three models. Our model contains three different views, which nearly triples the network parameters. For efficiency, we reduce the parameters on the fully connected *fc6* and *fc7* layers via a truncated SVD decomposition [13]. Specifically, each *fc* layer with parameters $W \in d \times 4096$ (d is the dimension of input features) is decomposed into two sub-layers *fc-1* and *fc-2*, with weights $W_1 \in d \times 1024$ and $W_2 \in 1024 \times 4096$. We copy the parameters of the reduced fully connected layers to each view for network initialization. This leads to roughly the same amount of parameters for model **M** (1.6× for model **S**, and 0.75× for model **L**).

Implementation Details. We choose edge boxes [31] to generate $|\mathcal{R}| \approx 2500$ region proposals per image on average. For VOC dataset, we choose *five-scales* with $s = \{384, 512, 640, 768, 896\}$ for training and testing, while for COCO dataset, we only use a *single scale* with $s = 640$. We denote the length of its shortest side as the scale s of an image, and cap the longest side at 1500 pixels to avoid exceeding GPU memory. In the first phase learning, we randomly drop half regions of an image during each forward propagation. The training epoch is 20, with a learning rate of 10^{-5} for the first 10 epoches and reduced to 10^{-6} for the last 10 epoches. In the second phase learning, following [13], we regard all proposals that have IoU ≥ 0.5 with the mined objects as positive, and the proposals that have IoU $\in [0.1, 0.5)$ are treated as hard negative samples. The mini-batch sampling is constructed from $N = 2$ images with a mini-batch size of $\mathcal{R}_i = 128$. The training epoch is 16, with a learning rate of 10^{-4} for the first 8 epoches and reduced by a factor of 10 for the last 8 epoches. During multi-view co-training, the iteration times K is set as $K = 3$. The mean outputs of the multi-views are used for performance evaluation.

4.2 Design Evaluation

We conduct experiments to understand how ML-LocNet works, as well as to evaluate the necessity of multi-view designs. Without loss of generality, all experiments in this section are based on VOC 2007 with model **S**.

Model Analysis. We first conduct experiments with different configurations to reveal how each component affects the performance. The localization results are shown in Table 1. From the table we make the following observations:
Multi-view learning is crucial. Multi-view learning is able to consistently improve the localization accuracy during both phase learning. For the first phase, multi-view feature concatenation brings up to 4.6% (52.5% → 57.1%) improvement, which demonstrates that features from multiple views are complementary and are beneficial to accurately describing image regions; For the second phase, Multi-view co-training algorithm brings another 3.7% (57.9% → 61.6%) improvement. We find that simply training fast-RCNN alike models in the second phase brings negligible gain (57.9% → 58.2%), since the network is easy to focus on the initial seeds. By introducing multi-view co-training mechanism, instances are mined from some views but used for training for other view, thus the network can effectively avoid overfitting.
Region dropout makes the model training faster and better. Region dropout improves accuracy by 0.8% (57.1% → 57.9%), since it selects different regions of

Table 1. Effects of various design choices during two phase learning

	Phase One			Phase Two			
Multi-views?		✓	✓		✓	✓	✓
Region Dropout ?			✓				
Random Mini-batch Sampling?						✓	✓
Weighted Loss?							✓
PASCAL VOC 2007 CorLoc (%)	52.5	57.1	**57.9**	58.2	61.6	62.8	**63.7**

an image for classification loss, and can be treated as data augmentation. Another advantage is that the training is faster due to a reduced number of region proposals feeded to the fully connected layers.

Random mini-batch sampling is helpful. Introducing random mini-batch sampling helps improve the localization accuracy, with a gain of 1.2% (61.6% → 62.8%). This is achieved by increasing the diversity among views, such that different views can exchange localization knowledge to boost the model.

Weighted loss helps. Introducing weighted loss brings about another 0.9% performance gain. This demonstrates that considering the confidence of the mined object instances is able to avoid focusing on the less reliable instances and help boost the performance.

Why Using Different Multi-view Learning Strategies in Two Phases? In our two phase learning, the multi-view strategy is implemented in two different ways. One question is that why bother to train models with different strategies, and is it possible to simply choose the same one, *i.e.,* training both networks with multiple losses as in Fig. 2? We tried this setting, and trained the network in Fig. 1 with losses from multiple views, but the localization performance is limited, with only 54.6%, 3.3% lower than the concatenation method. The possible reason is that the losses from multiple views are hard to optimize with only image-level supervision, since the object instances are mined implicitly from the intermediate layers. Instead, enhancing feature representation is an effective method and is relatively easier to be optimized.

Does Performance Improvement Come from Multi-view Learning? The essence of ML-LocNet is to design multiple views and introduce several modules to ensure the view diversity. It is in doubt that does the performance improvement really come from the diversity of multiple views, or is it the increased parameters (the model is 1.6× larger with multi-view design) that help? To validate this issue, we initialize different views with the same parameters (fully connected layers of model **S**), and without conducting model compression. As a result, the network is nearly 2× larger than ML-LocNet. Then we replace ML-LocNet with the above designed network during each phase learning, respectively. For the first phase, the localization performance drops from 57.9% to 56.2%, while for the second phase, it drops rapidly, from 63.7% to 58.9%. The

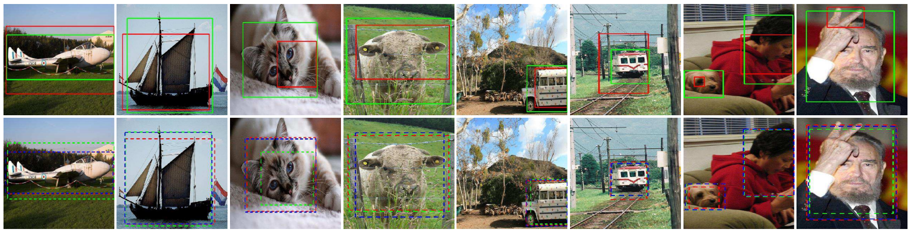

Fig. 3. Localization visualization on VOC 2007 *trainval* split. The top row shows localizations before (red) and after (green) multi-view co-training. The bottom row shows the localizations of multiple views during training

results demonstrate that the multi-view mechanism is especially important for the second phase learning. As an illustration, Fig. 3 shows the localization results before and after the second phase learning (top row), together with the intermediate localization of multiple views (bottom row). It can be seen that multi-view co-training is effective to refine the localization by exchanging knowledge from the diversified views.

4.3 Results and Comparisons

PASCAL VOC 2007. This is a widely used dataset for weakly supervised localization and detection, and we conduct detailed comparisons on this dataset.
CorLoc evaluation. Table 2 shows the localization results on PASCAL VOC 2007 *trainval* split in terms of CorLoc [6]. Our method achieves an accuracy of 63.7% with model **S**, which is 8.1% better than previous best result [8] (55.6%) using the same model, and even outperforms the-state-of-the-art [26] (60.6%) that using deeper model. Replacing with model **L**, we achieve a CorLoc of 67.0%, 6.4% better than the best-performing result. Finally, using the mean outputs of three models, we train another model based on model **L**, which we refer to as ML-LocNet-**L+**. We obtain an accuracy of 68.6%, which is 4.3% higher than [26] (64.3%) using the same training strategy. Note that ML-LocNet-**L+** is based on the outputs of a single model, which is different from model ensemble that combines the outputs of multiple models during test stages. Another advantage is that our best results are based on a reduced fully connected layers, and the network scale is only 0.75× as the original model **L** used in previous works.
AP evaluation. Table 3 shows the detection performance on VOC 2007 *test* split. Just using model **S**, our method achieves an accuracy of 44.4%, 5.1% higher than the best-performing method [8] (39.3%) using model **S**. When switching to model **L**, the detection accuracy increases to 48.4%, which is about 4.1% better than the best-performing result [8] (44.3%). Using the mean localization outputs for model initialization, we obtain a detection accuracy as high as 49.7%, and is 2.7% better than [26] (47.0%) using the same training mechanism. This is a promising result considering the challenge of this task.

Table 2. Localization precision (%) comparisons on PASCAL VOC 2007 *trainval* split

method	aer	bik	brd	boa	btl	bus	car	cat	cha	cow	tbl	dog	hrs	mbk	prs	plt	shp	sfa	trn	tv	mAP
WSDDN[1]	65.1	58.8	58.5	33.1	39.8	68.3	60.2	59.6	34.8	64.5	30.5	43.0	56.8	82.4	25.5	41.6	61.5	55.9	65.9	63.7	53.5
ODGA-**S**[8]	83.5	70.9	65.4	42.4	39.0	63.9	80.8	58.6	30.2	69.5	24.8	51.0	66.2	78.4	25.2	48.7	66.6	26.7	63.3	55.9	55.6
DSD[15]	72.7	55.3	53.0	27.8	35.2	68.6	81.9	60.7	11.6	71.6	29.7	54.0	64.3	88.2	22.2	53.7	72.2	52.6	68.9	75.5	56.1
ConLocNet[7]	83.9	72.8	64.5	44.1	40.1	65.7	82.5	58.9	33.7	72.5	25.6	53.7	67.4	77.4	26.8	49.1	68.1	27.9	64.5	55.7	56.7
ODGA[8]	85.5	75.0	66.9	47.5	43.6	67.4	83.6	61.7	36.8	75.1	29.8	55.9	70.4	80.6	29.0	52.9	71.0	31.2	66.9	58.1	59.4
OICR[26]	81.7	80.4	48.7	49.5	32.8	81.7	85.4	40.1	40.6	79.5	35.7	33.7	60.5	88.8	21.8	57.9	76.3	59.9	75.3	81.4	60.6
ML-LocNet-**S**	76.9	78.2	65.8	39.8	45.9	78.0	85.1	57.0	16.9	70.9	68.5	56.5	77.0	90.6	47.4	52.2	65.6	60.7	75.5	65.2	63.7
ML-LocNet-**M**	78.0	78.5	66.2	43.2	51.5	76.5	86.8	65.7	34.9	69.5	59.7	55.1	79.5	88.1	40.3	58.3	71.5	64.0	77.0	61.6	65.3
ML-LocNet-**L**	78.6	82.3	68.2	42.0	53.3	78.5	88.5	70.3	36.4	70.2	60.5	58.0	80.5	88.2	38.8	59.2	75.0	69.0	78.2	64.5	**67.0**
OICR-**L+**[26]	81.7	80.4	48.7	49.5	32.8	81.7	85.4	40.1	40.6	79.5	35.7	33.7	60.5	88.8	21.8	57.9	76.3	59.9	75.3	81.4	64.3
ML-LocNet-**L+**	81.7	82.9	68.7	44.4	53.9	80.3	88.9	70.5	32.6	74.0	62.7	61.7	81.4	91.6	46.0	60.6	75.2	69.2	78.7	65.8	**68.6**

Table 3. Detection precision (%) comparisons on PASCAL VOC 2007 *test* split

method	aer	bik	brd	boa	btl	bus	car	cat	cha	cow	tbl	dog	hrs	mbk	prs	plt	shp	sfa	trn	tv	mAP
WSDDN[1]	39.4	50.1	31.5	16.3	12.6	64.5	42.8	42.6	10.1	35.7	24.9	38.2	34.4	55.6	9.4	14.7	30.2	40.7	54.7	46.9	34.8
ConLocNet[7]	57.1	52.0	31.5	7.6	11.5	55.0	53.1	34.1	1.7	33.1	49.2	42.0	47.3	56.6	15.3	12.8	24.8	48.9	44.4	47.8	36.3
ODGA-**S** [8]	45.7	58.1	37.2	24.8	19	64.8	53.7	35.2	9.7	44.8	22.6	33.7	50.4	57.8	15.9	21.7	40.8	48.2	55.4	45.8	39.3
OICR[26]	58.0	62.4	31.1	19.4	13.0	65.1	62.2	28.4	24.8	44.7	30.6	25.3	37.8	65.5	15.7	24.1	41.7	46.9	64.3	62.6	41.2
DSD[15]	52.2	47.1	35.0	26.7	15.4	61.3	66.0	54.3	3.0	53.6	24.7	43.6	48.4	65.8	6.6	18.8	51.9	43.6	53.6	62.4	41.7
ODGA[8]	50.9	61.2	40.5	31.4	21.1	71.6	58.1	42.9	11.7	46.4	30.7	44.5	48.3	64.9	16.8	24.8	47.1	55.7	61.7	55.8	44.3
ML-LocNet-**S**	57.0	64.0	42.6	22.1	17.9	59.3	64.0	39.5	2.2	47.6	55.0	38.9	66.4	68.1	30.6	23.5	43.2	44.1	55.4	46.3	44.4
ML-LocNet-**M**	57.2	64.6	44.5	26.3	21.1	65.7	67.1	56.4	16.1	51.7	50.5	37.2	64.2	69.4	24.5	25.4	51.9	51.3	56.9	42.2	47.2
ML-LocNet-**L**	59.3	68.9	45.7	29.0	24.5	64.8	68.4	59.3	18.6	49.1	50.2	43.1	65.8	70.2	19.9	24.3	48.1	54.2	62.8	41.8	**48.4**
OICR-**L+**[26]	65.5	67.2	47.2	21.6	22.1	68.0	68.5	35.9	5.7	63.1	49.5	30.3	64.7	66.1	13.0	25.6	50.0	57.1	60.2	59.0	47.0
ML-LocNet-**L+**	60.8	70.6	47.8	30.2	24.8	64.9	68.4	57.9	11.0	51.3	55.5	48.1	68.7	69.5	28.3	25.2	51.3	56.5	60.0	43.1	**49.7**
Fast-RCNN[13]	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8	66.9

Table 4. Localization precision (%) comparisons on PASCAL VOC 2012 *trainval* split

method	aer	bik	brd	boa	btl	bus	car	cat	cha	cow	tbl	dog	hrs	mbk	prs	plt	shp	sfa	trn	tv	mAP
ConLocNet[7]	78.3	70.8	52.5	34.7	36.6	80.0	58.7	38.6	27.7	71.2	32.3	48.7	76.2	77.4	16.0	48.4	69.9	47.5	66.9	62.9	54.8
DSD[15]	82.4	68.1	54.5	38.9	35.9	84.7	73.1	64.8	17.1	78.3	22.5	57.0	70.8	86.6	18.7	49.7	80.7	45.3	70.1	77.3	58.8
OICR[26]	86.2	84.2	68.7	55.4	46.5	82.8	74.9	32.2	46.7	82.8	42.9	41.0	68.1	89.6	9.2	53.9	81.0	52.9	59.5	83.2	62.1
ML-LocNet-**L**	87.0	84.8	69.8	47.1	58.9	88.8	77.0	47.3	41.7	79.9	30.3	62.1	83.2	91.4	33.5	63.6	76.9	60.4	72.6	70.3	**66.3**
OICR-**L+**[26]	89.3	86.3	75.2	57.9	53.5	84.0	79.5	35.2	47.2	87.4	43.4	43.8	77.0	91.0	10.4	60.7	86.8	55.7	62.0	84.7	65.6
ML-LocNet-**L+**	88.1	85.5	71.2	49.4	57.4	90.7	77.6	53.5	42.6	79.6	34.1	69.1	81.7	91.9	35.4	64.6	79.3	64.3	79.3	69.6	**68.2**

Table 5. Detection precision (%) comparisons on PASCAL VOC 2012 *test* split

method	aer	bik	brd	boa	btl	bus	car	cat	cha	cow	tbl	dog	hrs	mbk	prs	plt	shp	sfa	trn	tv	mAP
ConLocNet[7]	64.0	54.9	36.4	8.1	12.6	53.1	40.5	28.4	6.6	35.3	34.4	49.1	42.6	62.4	19.8	15.2	27.0	33.1	33.0	50.0	35.3
DSD[15]	60.8	54.2	34.1	14.9	13.1	54.3	53.4	58.6	3.7	53.1	8.3	43.4	49.8	69.2	4.1	17.5	43.8	25.6	55.0	50.1	38.3
OICR[26]	67.7	61.2	41.5	25.6	22.2	54.6	49.7	25.4	19.9	47.0	18.1	26.0	38.9	67.7	2.0	22.6	41.1	34.3	37.9	55.3	37.9
ML-LocNet-**L**	68.1	63.3	43.7	19.9	26.5	61.1	53.0	36.7	14.8	45.8	11.9	46.1	58.4	73.4	16.8	26.9	42.5	35.3	54.5	45.4	**42.2**
OICR-**L+**[26]	71.4	69.4	55.1	29.8	28.1	55.0	57.9	24.4	17.2	59.1	21.8	26.6	57.8	71.3	1.0	23.1	52.7	37.5	33.5	56.6	42.5
ML-LocNet-**L+**	53.9	60.4	40.4	23.3	18.7	58.7	63.3	52.5	13.3	49.1	46.8	33.5	61.0	65.8	21.3	22.9	46.8	48.1	52.6	40.4	**43.6**

Error analysis. To analysis the detection performance of our model in more details, we use the analysis tool from [14] to diagnose the detector errors. Fig. 4 shows the error analysis on VOC 2007 *test* split with model **L+** (49.7% mAP). The classes are categorized into four categories, *animals*, *vehicles*, *furniture*, and *person*. Our method achieves promising results on categories *animals* and *vehicles*, but it does not work well on detecting *furniture* and *person*. This is mainly because *furniture* are usually in cluttered scenes, thus very hard to pick out for model training, and the error distribution is scattered. While for *person*, the

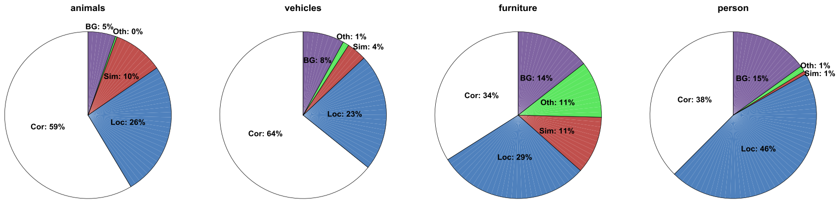

Fig. 4. Detection error analysis [14] of ML-LocNet on VOC 2007 *test* split. The detections are categorized as correct (Cor), false positive due to poor localization (Loc), confusion with similar categories (Sim), with others (Oth), and with background (BG)

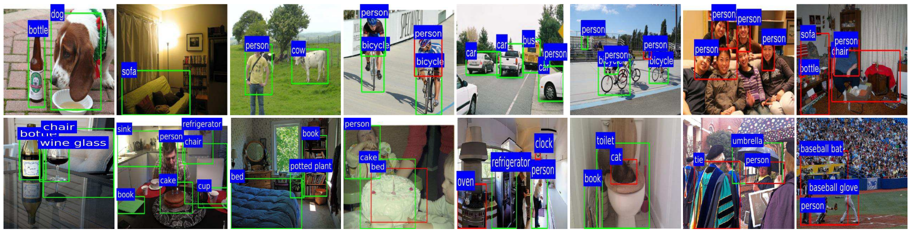

Fig. 5. Example detections on PASCAL VOC 2007 (top row) and MS COCO 2014 (bottom row). The successful detections (IoU ≥ 0.5) are marked with green bounding boxes, and the failed ones are marked with red. We show all detections with scores ≥ 0.7 and use nms to remove duplicate detections

majority of errors come from inaccurate localization (blue regions). As an illustration, we show some detection results in Fig. 5. The correct detections are marked with green bounding boxes, while the failed ones are marked with red. Our detectors are able to successfully detect objects in images with relatively simple background, and is fine for vehicles even in complex images. However, detections are easy to fail in complex scenes for other categories, and are often focus on object parts, or grouping multiple objects from the same class.
Comparing with fully supervised fast-RCNN. It is interesting to compare our weakly supervised detections with the fast-RCNN [13] method, which makes use of ground truth bounding boxes for training. As shown in Table 3, the performance of ML-LocNet is around 17% lower than fast-RCNN. However, for vehicles such as *bicycle* and *motorbike*, the performance approaches the fully supervised one (70.6% vs 78.3% for *bicycle* and 70.2% vs 73.0% for *motorbike*). This implies that it is possible to train corresponding detection models on these classes without requiring object annotations. However, for classes such as *chair* and *person*, the performance gap is still large. It remains a further research direction to correctly localize these objects for detection model training.

PASCAL VOC 2012. We choose the same settings as in VOC 2007 experiments, and evaluate the performance on VOC 2012. Table 4 and Table 5 show the localization and detection results, respectively. We see the same performance

Table 6. Localization and detection precision comparisons on MS COCO 2014

Methods	CorLoc(%)	mAP (%)
WSDDN[1]	26.1	11.5
WCCN [7]	-	12.3
ODGA[8]	-	12.8
ML-LocNet-**S**	**34.7**	**16.2**

trend as we observed on VOC 2007. For localization, ML-LocNet achieves an accuracy of 66.3% with model **L**, 4.2% point better than previous best result [26] (62.1%). The accuracy improves to 68.2% with ML-LocNet-**L+** model. For detection, the result is 42.2% with model **L**, 3.9% more accurate than [15] (38.3%). The result can be further improved to 43.6% with ML-LocNet-**L+** model.

MS COCO 2014. To further validate the ML-LocNet model, we evaluate the performance on a much larger dataset MS COCO 2014. Comparing with PASCAL VOC 2012, MS COCO is more challenging: it includes more images (135k vs 22k), more categories (80 vs 20), more complex scenes (7.7 instances per image vs 2.3 instances per image, in average), and more images with smaller objects. As far as we know, few works have reported results on MS COCO under weakly supervised paradigm. Table 6 shows the localization and detection results with model **S**. On this challenging dataset, we obtain 34.7% localization accuracy, improving the baseline WSDDN [1] by 8.6%. For detection, the result is 16.2%, which is 3.4% better than [8] (12.8%). Fig. 5 shows some detection results on this dataset. Although objects are usually within more complex scenes, our model is still able to successfully detect rigid objects such as *book* and *refrigerator*.

5 Conclusions

This paper proposed a multi-view learning strategy to improve the localization accuracy in WSOL. Our method incorporates multi-view learning into a two-phase WSOL model, and is able to consistently improve the localization for both phases. In the first phase, we augment feature representation by concatenating features learned from multiple views, which is effective in describing image regions. In the second phase, we develop a multi-view co-training algorithm to refine the localization models. The models are optimized by iteratively mining new object instances among different views, thus effectually avoiding overfitting the initial seeds. Our method can be easily combined with other techniques to further improve the performance. Experiments conducted on PASCAL VOC and MS COCO benchmarks demonstrate the effectiveness of the proposed approach.

Acknowledgements. The work was supported in part to Jiashi Feng by NUS IDS R-263-000-C67-646, ECRA R-263-000-C87-133 and MOE Tier-II R-263-000-D17-112, in part to Yang Yang by NSFC under Project 61572108.

References

1. Bilen, H., Vedaldi, A.: Weakly supervised deep detection networks. In: CVPR. pp. 2846–2854 (2016)
2. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Computational Learning Theory. pp. 92–100. ACM (1998)
3. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. BMVC (2014)
4. Cheng, J., Wang, K.: Active learning for image retrieval with co-svm. Pattern recognition **40**(1), 330–334 (2007)
5. Cinbis, R.G., Verbeek, J., Schmid, C.: Multi-fold mil training for weakly supervised object localization. In: CVPR. pp. 2409–2416 (2014)
6. Deselaers, T., Alexe, B., Ferrari, V.: Weakly supervised localization and learning with generic knowledge. IJCV **100**(3), 275–293 (2012)
7. Diba, A., Sharma, V., Pazandeh, A., Pirsiavash, H., Van Gool, L.: Weakly supervised cascaded convolutional networks. CVPR pp. 914–922 (2017)
8. Diba, A., Sharma, V., Stiefelhagen, R., Van Gool, L.: Object discovery by generative adversarial & ranking networks. arXiv preprint arXiv:1711.08174 (2017)
9. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artifi. Intell. **89**(1), 31–71 (1997)
10. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. IJCV **111**(1), 98–136 (2015)
11. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV **88**(2), 303–338 (2010)
12. Feng, H., Shi, R., Chua, T.S.: A bootstrapping framework for annotating and retrieving www images. In: ACM Multimedia. pp. 960–967. ACM (2004)
13. Girshick, R.: Fast r-cnn. In: ICCV. pp. 1440–1448 (2015)
14. Hoiem, D., Chodpathumwan, Y., Dai, Q.: Diagnosing error in object detectors. In: European conference on computer vision. pp. 340–353. Springer (2012)
15. Jie, Z., Wei, Y., Jin, X., Feng, J., Liu, W.: Deep self-taught learning for weakly supervised object localization. CVPR (2017)
16. Kantorov, V., Oquab, M., Cho, M., Laptev, I.: Contextlocnet: Context-aware deep network models for weakly supervised localization. In: ECCV. pp. 350–365. Springer (2016)
17. Li, D., Huang, J.B., Li, Y., Wang, S., Yang, M.H.: Weakly supervised object localization with progressive domain adaptation. In: CVPR. pp. 3512–3520 (2016)
18. Li, Y., Liu, L., Shen, C., Van Den Hengel, A.: Mining mid-level visual patterns with deep cnn activations. IJCV **121**(3), 344–364 (2017)
19. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755. Springer (2014)
20. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV. pp. 21–37 (2016)
21. Nguyen, M.H., Torresani, L., de la Torre, F., Rother, C.: Weakly supervised discriminative localization and classification: a joint learning process. In: Proc. Int. Conf. Comput. Vis. pp. 1925–1932 (2009)
22. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: CVPR. pp. 685–694 (2015)

23. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. pp. 779–788 (2016)
24. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
25. Song, H.O., Lee, Y.J., Jegelka, S., Darrell, T.: Weakly-supervised discovery of visual pattern configurations. In: NIPS. pp. 1637–1645 (2014)
26. Tang, P., Wang, X., Bai, X., Liu, W.: Multiple instance detection network with online instance classifier refinement. In: CVPR (July 2017)
27. Vijayanarasimhan, S., Grauman, K.: Keywords to visual categories: Multiple-instance learning forweakly supervised object categorization. In: CVPR. pp. 1–8. IEEE (2008)
28. Wang, C., Ren, W., Huang, K., Tan, T.: Weakly supervised object localization with latent category learning. In: ECCV, pp. 431–445 (2014)
29. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR. pp. 2921–2929. IEEE (2016)
30. Zhu, Y., Zhou, Y., Ye, Q., Qiu, Q., Jiao, J.: Soft proposal networks for weakly supervised object localization. arXiv preprint arXiv:1709.01829 (2017)
31. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: ECCV, pp. 391–405 (2014)