

# Augmented Feedback in Semantic Segmentation Under Image Level Supervision

Xiaojuan Qi<sup>1</sup>(✉), Zhengzhe Liu<sup>1</sup>, Jianping Shi<sup>2</sup>,  
Hengshuang Zhao<sup>1</sup>, and Jiaya Jia<sup>1</sup>

<sup>1</sup> The Chinese University of Hong Kong, Shatin, Hong Kong  
{xjq,zzliu,hszhao,leojia}@cse.cuhk.edu.hk  
<sup>2</sup> Sense Time Group Limited, Beijing, China  
shijianping@sensetime.com

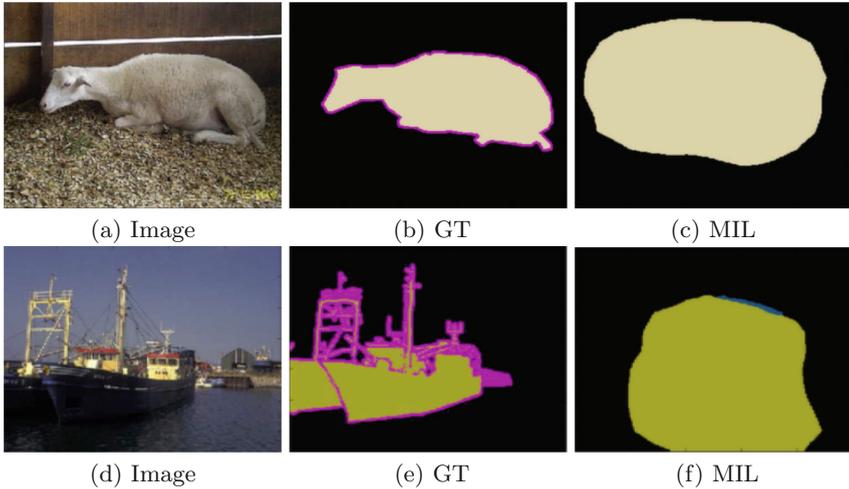
**Abstract.** Training neural networks for semantic segmentation is data hungry. Meanwhile annotating a large number of pixel-level segmentation masks needs enormous human effort. In this paper, we propose a framework with only image-level supervision. It unifies semantic segmentation and object localization with important proposal aggregation and selection modules. They greatly reduce the notorious error accumulation problem that commonly arises in weakly supervised learning. Our proposed training algorithm progressively improves segmentation performance with augmented feedback in iterations. Our method achieves decent results on the PASCAL VOC 2012 segmentation data, outperforming previous image-level supervised methods by a large margin.

**Keywords:** Weakly supervised learning · Semantic segmentation · Image-level supervision · Proposal aggregation

## 1 Introduction

Great improvement was made for semantic segmentation [1–6] based on deep *convolutional neural networks* (DCNNs). The success largely depends on the amount of training data with accurate pixel-level supervision [7–9]. It is well known in this community that collecting accurate annotation in a large quantity is very labor intensive. In our experience, to label a good-quality segmentation map from one image with resolution  $400 \times 600$ , 5–8 min are needed even for an experienced user. It seriously hinders producing a very large set of training data with full labels.

Compared to labor-costly annotation for each pixel, image-level annotation only gives each image several object labels. It is probably the simplest supervision for segmentation, since each image only needs seconds of manual work regardless of its resolution. Compared to the traditional way for segment labeling, image-level supervision can easily scale training data up for hundreds or thousands of times with the same amount of total manual work. This motivates us to conduct research on this topic.



**Fig. 1.** Illustration of MIL prediction. (a) and (d) are the original images. (b) and (e) are the corresponding segmentation ground-truth. (c) and (f) are the corresponding MIL prediction [11]. They are coarse where localization accuracy is not high.

Previous CNN-based image-label supervised segmentation approaches [10–14] can be coarsely cast into two categories. The first line utilizes *multiple instance learning* (MIL) to directly predict pixel labels [10, 11, 13, 14]. Under this setting, each image is viewed as a bag of superpixels/pixels. It is positive when at least one superpixel/pixel is positive. The bag-level image prediction is aggregated by the latent variables, i.e., superpixel/pixel prediction. Since there is no direct pixel-wise supervision from low-level clues during training, this strategy is vulnerable to location variation of objects. One example is shown in Fig. 1, where (c) and (f) are prediction results. Also, MIL heavily relies on good initialization [15–17].

Another stream [12] is based on *Expectation-Maximization* (EM). It iterates between generating temporary segmentation masks and learning with interim supervision. These methods benefit from pixel-level supervision; but errors easily accumulate in iterations.

In this paper, we propose a learning framework that enjoys the benefit from interim pixel-wise supervision. Meanwhile, it suppresses error accumulation in iterations. Instead of obtaining pixel labels solely from previous-round segmentation prediction [12], we introduce an object localization branch to assist supervision generation. This localization branch functions as an object detector, which classifies region proposals to adjust output from the segmentation branch. After localizing the objects, proposals are combined to form a segmentation mask, which also improves segmentation in the other branch.

Our segmentation and localization modules form augmented feedback in the unified training procedure. Prediction of segmentation can help select confident

object proposals to supervise training of localization. The result of localization also supplements segmentation to pull results out of local optima. Although in the beginning, masks are very coarse and localization information is not accurate, they are quickly improved with our iterative training procedure. Our main contribution lies in the following folds.

- We propose a new framework for semantic segmentation under image-level supervision, which infers localization and segmentation information.
- We develop an aggregation procedure to generate segmentation masks on top of interim object proposals.
- An effective training method was adopted to make our segmentation and localization benefit from each other with augmented feedback.
- Our method outperforms previous work under similar image-level supervision by a large margin on PASCAL VOC 2012 data.

## 2 Related Work

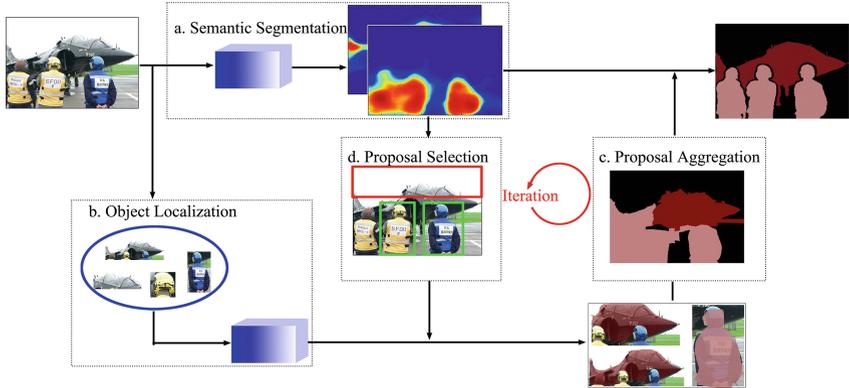
### 2.1 Strongly Supervised Semantic Segmentation

DCNNs have greatly boosted the performance of semantic segmentation [1, 4, 5, 18–20] in the strong supervision setting. The methods fall into two broad categories. One utilizes DCNNs to classify object proposals [4, 5, 18, 20]. The other class adopts fully convolutional networks [1] to make dense prediction. CRF models are applied as post-processing [2] or inside the network to refine prediction [3, 19].

**Weakly Supervised Semantic Segmentation.** Semantic segmentation under weak supervision is practical due to the heavy burden of annotating pixel-wise ground truth. Various forms were proposed [6, 12, 21]. In [6, 12], bounding boxes are used in annotation. Papandreou *et al.* [12] estimated segmentation with the CRF [22] model. Dai *et al.* [6] transferred segmentation mask estimation to a proposal selection problem, which achieves good performance. Russakovsky *et al.* [21] utilized instance points as supervision. To further facilitate annotation, Lin *et al.* [23] used scribbles, which are more natural for human to draw especially for irregular object shapes. Bounding boxes, points and scribbles are different ways to simplify supervision for users to quickly manipulate images.

Without requiring to draw anything in images, image-level labels were used with *multiple instance learning* (MIL) [10, 11, 13, 14, 24]. Each image is viewed as a bag of pixels (or superpixels). Prediction is taken as latent variables while the image result is accomplished by aggregation. The MIL methods generate coarse prediction because the algorithms generally do not use low-level cues.

Papandreou *et al.* [12] adopted an *Expectation-Maximization* (EM) approach for image-level supervision. It iterates between segment mask generation and neural network training. Wei *et al.* [25] used the self-paced learning strategy, initially trained with saliency maps of simple images. It progressively includes more difficult examples. The results update according to output of previous iterations.



**Fig. 2.** Overview of our framework with four parts. Part (a) is the fully convolutional segmentation network. Part (b) is the object localization network. Part (c) is the proposal aggregation module. It aggregates the proposal localization result for segmentation training. Part (d) is the proposal selection module. It selects positive and negative proposals for training of the object localization branch. Our network is updated iteratively. In part (d), the green and red bounding boxes mark positive and negative samples. (Color figure online)

**Weakly Supervised Localization.** Weakly supervised localization uses image-level labels to train detection or localization, which is also related to our task. Part of prior work uses multiple instance learning (MIL) [16, 17, 26–28]. If an image is positively labeled, at least one region is positive. Contrarily, the image is negative if all regions are negative. The learning process alternates between selecting regions corresponding to the object and estimating the object model. The algorithm relies on the learned model for object region selection. This kind of dependency makes algorithms sensitive to initialization quality [27]. Our localization branch differs from these approaches fundamentally. Our region selection procedure is guided by the segmentation branch, which can effectively correct errors of localization.

### 3 Our Architecture with Augmented Feedback

Our architecture for weakly-supervised semantic segmentation is illustrated in Fig. 2. It has four main components. Briefly, semantic segmentation and object localization are linked by the aggregation and proposal selection components. The two branches provide augmented feedback to each other to correct errors progressively during training. It is distinct from previous EM/MIL frameworks [12] that only take feedback from the network itself in past iterations.

More specifically, our segmentation branch predicts pixel-wise labels. The per-category scores are clustered into foreground and background as shown in (a). This piece of information is then combined with previous-round object localization prediction to select corresponding positive and negative object proposals for current iteration supervision, as shown in (d). Major errors can be

quickly spotted and removed with this type of augmented feedback. Intriguingly, object localization finds objects by classifying proposals. Through aggregation shown in (c), it can tell to an extent whether current segmentation is reasonable or not. This strategy makes it possible to improve segmentation quality by estimating good reference maps. In what follows, we explain these modules and verify their necessity and effectiveness.

### 3.1 Semantic Segmentation

We adopt a fully convolutional network for the segmentation part [1, 2] corresponding to component (a) in our architecture shown in Fig. 2. This part takes an image as input, and generates a per-class score map through down- and up-sampling, denoted as  $\{s_c | c = 0, 1, 2, \dots, C\}$ . Where  $C$  is the number of categories and class 0 corresponds to background. Each  $s_c$  is with the same size of the image. Since this part is almost standard and does not form our main contribution, we refer readers to previous papers [1, 2] for more details.

### 3.2 Object Localization

We resort to bottom-up object proposal generation [29–31] to obtain a set of mask candidates initially. Object localization are obtained with VGG16 [32]. They turn object localization to the task of proposal classification by predicting semantic probability of a proposal belonging to a class. We adopt the sigmoid cross-entropy loss to predict each class. This process is also common now. Each proposal is with a score  $\{p^c | c = 1, 2, \dots, C\}$  where  $C$  is the number of categories.

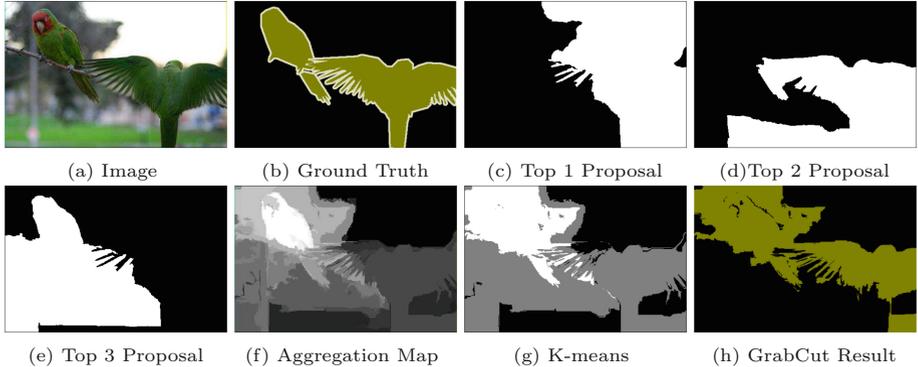
## 4 Important Proposal Aggregation and Sample Selection

As aforesaid, our major contribution includes proposal aggregation and selection modules. They provide effective augmented feedback to improve semantic segmentation with only image-level labels.

### 4.1 Proposal Aggregation

Prediction of localization is inevitably erroneous in the beginning of training. As shown in Fig. 3(c)–(e), birds are mostly mislabeled even with high-confidence localization estimates. This problem is prevalent in weakly supervised localization. Note the results shown in (c)–(e) could significantly degrade learning performance, making parameters stuck at mistaken points.

Our aggregation process is operated independently for each class  $c$ . First, we keep useful proposals with proposal score  $p^c$  larger than 0.5. For every pixel  $i$  in the image, its aggregation score  $a^i$  is calculated as sum of all proposal scores  $p_j^c$ , i.e.,  $a^i = \sum_j p_j^c$  where proposals  $j$  cover this pixel. If pixel  $i$  is not within any proposal,  $a^i = 0$ .



**Fig. 3.** Low-level vision cues to update localization scores. (a) Input image. (c)–(e) Top-3 scored proposals for “bird” and their masks generated by MCG [30]. (f) Aggregation map. (g) Clusters by K-means. Since medium-score pixels are ambiguous, we apply graph cuts to determine a partitioning map, which considers image structures. The result is shown in (h).

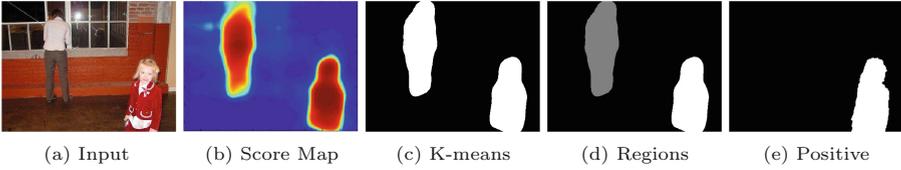
The resulting aggregation score map is denoted as  $S_0$  as illustrated in Fig. 3(f). It is not accurate initially. To combine information from the two branches,  $S_0$  is then fused with previous-round segmentation score map  $s_c$  by element-wise score multiplication. The resulting map is denoted as  $S$ .

We group  $S$  into three clusters with respect to the high, medium, and low scores. They indicate highly-confidence object, ambiguous, and high-confidence background pixels. The image is accordingly partitioned into three clusters by K-means (Fig. 3(g)). Top and bottom-score pixels are regarded as object and background seeds. Finally, we use GrabCut [33] to partition ambiguous pixels in (g) into objects and background considering image edges. This step makes use of low-level vision cues. In the first pass, the cut result, as shown in Fig. 3(h), is still noisy because of highly inaccurate localization. We will explain in Sect. 5.1 and show in Fig. 5 that it becomes much more reasonable with iteratively improved localization and segmentation with augmented feedback in our framework.

With the above procedure, multiple objects map are generated separately regarding each class. Our final segmentation map is constructed by only assigning each pixel the class with the highest score in  $S$ .

## 4.2 Positive and Negative Sample Selection

To train the localization branch, we need to generate positive and negative samples. These samples are collected by a new strategy by combining segmentation and localization results produced in previous round. Since there are already scores  $s_c$  output from the segmentation branch, we collect and group pixels by applying K-means again with cluster number 2. Only the pixels with larger scores are kept. In image space, there can be multiple regions formed for the remaining pixels as shown in Fig. 4(d), corresponding to different objects. We consider



**Fig. 4.** Proposal selection. (b) Segmentation score map. (c) Clusters by scores. (d) Confident regions. (e) Selected positive proposal mask.

---

### Algorithm 1. Positive and Negative Sample Selection

---

Input:

- Segmentation score maps  $\{s_c | c = 1, 2, \dots, C\}$  where  $C$  is the number of classes;
- Proposal scores  $\{p_j^c | j = 1, 2, \dots, J, c = 1, 2, \dots, C\}$  where  $J$  is the number of proposals;
- IoU thresholds  $\gamma_1$  and  $\gamma_2$ , where  $\gamma_1 > \gamma_2$  for selecting samples;
- Proposal score thresholds  $\eta_1$  and  $\eta_2$  where  $\eta_1 > \eta_2$ .

Procedure:

- 1: **for**  $c \in [1, C]$  **do**
  - 2:   **if** Image does not contain object in class  $c$  **then**
  - 3:     Continue.
  - 4:   **end if**
  - 5:   Cluster segmentation score map  $s_c$  into  $R_1$  and  $R_2$  for confident and ambiguous regions respectively.
  - 6:   **for**  $j \in [1, J]$  **do**
  - 7:     Calculate IoU values between the mask of proposal  $j$  and all regions in  $R_1$ .
  - 8:     Label proposal as positive if its highest IoU  $> \gamma_1$  and  $p_j^c > \eta_1$ .
  - 9:     Label proposal as hard negative if the highest IoU  $\leq \gamma_2$  and  $p_j^c > \eta_1$ .
  - 10:    Label proposal as negative if its highest IoU  $\leq \gamma_2$  and  $p_j^c \leq \eta_2$ .
  - 11:   **end for**
  - 12: **end for**
- Output: positive, negative and hard negative samples.
- 

their intersection-over-union (IoU) score with the proposal masks. A large score indicates good chance to be an object.

To reduce the influence of incorrect prediction from current segmentation, we consider previous-round proposal score  $p_j^c$  for each proposal  $j$  to perform positive and negative sample selection. We assign a positive label to proposal  $j$  when the IoU overlap with any above mentioned confident segmentation region is higher than a threshold  $\gamma_1$  and its current proposal score  $p_j^c$  is higher than a threshold  $\eta_1$ . We assign it a negative label contrarily when the IoU overlap is lower than a threshold  $\gamma_2$  with any confident segment region and either of the following conditions is satisfied: (1) its previous-pass proposal score  $p_j^c$  is lower than  $\eta_2$ ; (2) its previous-pass proposal score is higher than  $\eta_1$ . And the latter are regarded as hard negatives. The selection procedure is summarized in Algorithm 1.

To train the localization branch, positive and negative samples are used. Proposals neither positive nor negative are omitted accordingly. Since positive

---

**Algorithm 2.** Our Complete Network System with Augmented Feedback

---

Input:

Input image set; proposal set; maximum iteration number  $T$ ;

Procedure:

- 1: Initialize the proposal score with the classification network [32, 35];
- 2: **for**  $i \in [1, T]$  **do**
- 3:   Aggregate proposals to generate segmentation masks (described in Sect. 4.1);
- 4:   Train semantic segmentation branch;
- 5:   Select positive and negative samples (described in Sect. 4.2);
- 6:   Train object localization branch;
- 7:   Re-localize object proposals using the trained localization branch;
- 8: **end for**

Output: Semantic segmentation model.

---

samples are generally much less than negative ones, to balance the learning process, we adopt a random sampling strategy. For each batch, we randomly sample negatives to make the numbers of positive and negative samples around 1 : 3 [34]. Hard negative samples provided in Algorithm 1 are with the highest priority to be kept for training. Specifically, when sampling negative training data, we first consider the hard negatives. If their total number is not large enough for final ratio 1 : 3, we include other negative ones.

### 4.3 Training Algorithm

We adopt an iterative optimization scheme. The training procedure begins with classifying proposals using the network of [35]. In each pass, we train the segmentation branch with supervision of the segmentation mask produced in previous round. The available prediction scores of both branches are used to select positive and negative samples, which was described in Sect. 4.2. Then we train the object localization branch, which are followed by re-localizing object proposals.

Finally, we aggregate the re-localized object proposals using the method presented in Sect. 4.1, generating semantic segmentation masks. These steps iterate in a few passes until no obvious change on results can be observed. The process is summarized in Algorithm 2.

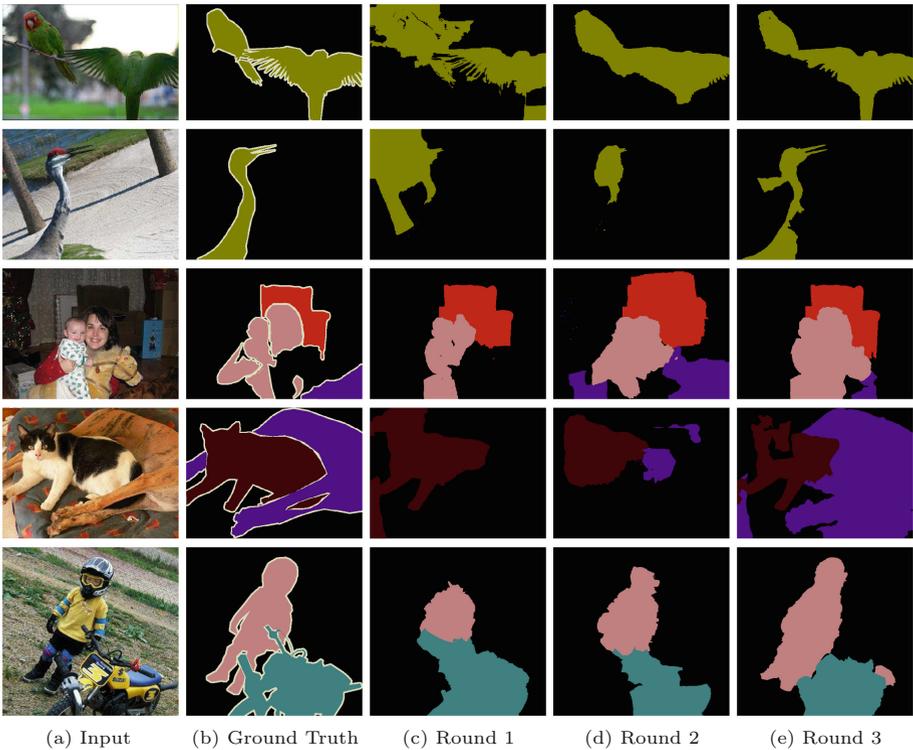
## 5 Experiments

Our method is evaluated on the PASCAL VOC 2012 segmentation benchmark dataset [7]. This dataset has 20 object categories and 1,464 images for training. Following the procedure of [2, 6, 12] to increase image variety, the training data is augmented to 10,582 images. In our experiments, we only use image-level labels. The result is evaluated on the PASCAL VOC validation and test sets, which contain 1,449 and 1,456 images respectively. The performance is evaluated regarding Intersection over union (IoU). We adopt Deeplab-largeFOV [2] as the baseline network. 100 held-out random validation images are used for cross-validation to set hyper-parameters.

## 5.1 Training Strategy

The network is trained as shown in Algorithm 2. Our optimization alternates between semantic segmentation and object localization training. We use the Caffe framework [36]. Parameters of the segmentation branch are initialized with the model provided in [2, 12]. In the training phase, a min-batch size of 8 images is adopted and  $321 \times 321$  image patches are randomly cropped following the procedure of [2, 12] as the network input. The initial learning rate is 0.001. It is divided by 10 after every 2 epochs. Training terminates after 8,000 iterations.

The localization branch is initialized with VGG16 [32, 35]. For its training, 30 proposals randomly sampled from one image with positive-negative ratio 1 : 3 form a mini-batch. They are all resized to resolution  $224 \times 224$ . The initial learning rate is 0.0001. It is divided by 10 for every epoch. Training terminates after 15,000 iterations. Momentum and weight decay are set to 0.9 and 0.0005 for both branches.



**Fig. 5.** Training masks in iterations. With our augmented feedback, the estimated segmentation mask gets improved. The first-row example corresponds to that in Fig. 4, which shows how the segmentation estimate is updated in iterations.

**Table 1.** Comparison with our baseline model. Results are evaluated on the PASCAL VOC 2012 segmentation validation dataset. Results from different iterations are listed.

Methods	Mean IoU %
WSSL [12]	35.60
Classification network baseline	38.63
Ours (Round 1)	47.40
Ours (Round 2)	48.12
Ours (Round 3)	<b>50.41</b>

**Table 2.** Results on PASCAL VOC 2012 segmentation validation dataset.

Methods	mAP%@0.5
Classification-net baseline [35]	5.71
RMI [26]	12.02
Single-branch baseline	16.75
Ours	<b>28.25</b>

## 5.2 Evaluation of the Two-Branch Framework

We use the classification network finely tuned on VOC 2012 dataset [35] to predict scores of object proposals and aggregate the result to generate the first-iteration segmentation mask, which is our training baseline. Then our training procedure proceeds as depicted in Algorithm 2. In each iteration, the estimated masks get more accurate as shown in Fig. 5. This process benefits from object localization with our proposal aggregation module. The performance on the VOC 2012 validation set of the iterative training procedure is summarized in Table 1.

More than 11% performance improvement is yielded using our learning procedure. Compared with WSSL [12], which uses the trained semantic segmentation model to perform next-round segmentation mask estimation, our method outperforms it with about 15% higher mean IoU. The intuitive explanation is that positive feedback is much augmented with our proposal aggregation and selection strategies in iterations. The statistics manifest that our method does not easily accumulate errors in iterations.

## 5.3 Evaluation of the Localization Branch

We evaluate localization performance in terms of bounding box mAP (mean average precision) [7] on the PASCAL VOC 2012 segmentation validation set with IoU threshold 0.5. The object proposals are obtained from the MCG [30] method. The initial classification network [35] serves as our baseline. We also compare with weakly-supervised object localization method RMI [26] and our own constructed single-model baseline that uses only the object localization branch to

**Table 3.** Results evaluated on PASCAL VOC 2012 validation set. \* denotes utilizing CRF [22] in the testing phase as post processing.

Proposal generation methods	Mean IoU%
SS	46.98
SS*	51.62
MCG	50.41
MCG*	<b>54.34</b>

perform positive and negative sample selection. To compare with RMI [26], we extract the VGG16 [32] fc7 feature and conduct SVM classification following the procedure of [26]. The performance is summarized in Table 2.

Our method outperforms the baseline classification network by a large margin, which shows the effectiveness of our method in terms of localization with the iterative update. Compared to RMI [26] and our own single-branch baseline, the two-branch architecture iteratively and stably provides more and more accurate samples especially in later passes with the effective selection procedure.

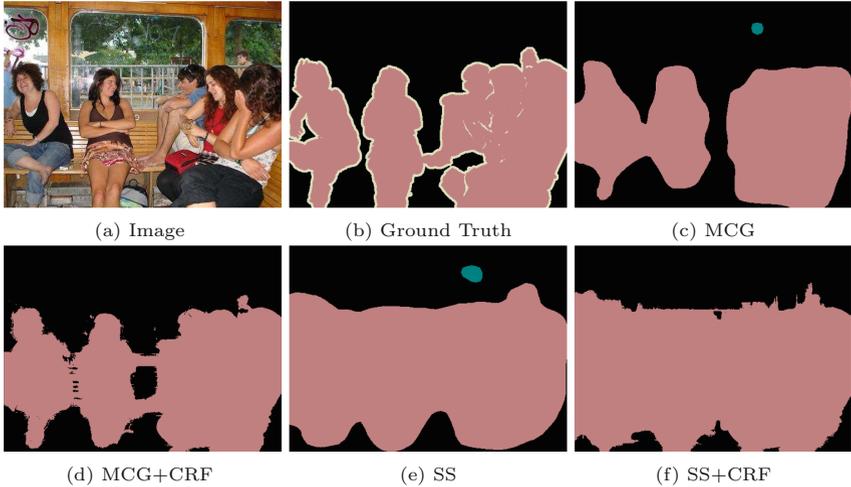
#### 5.4 Using Different Object Proposal Generation Strategies

We compare different object proposal generation strategies, including selective search (SS) [29] and MCG [30] for our framework. Segmentation performance with these strategies is listed in Table 3. MCG [30] achieves higher accuracy when training the semantic segmentation module. This is because the segmentation quality of MCG is higher than selective search while our estimated semantic mask is aggregated from the proposal mask where a high-quality contour would be helpful. We note, during testing, our method does not rely on object proposals. So the higher accuracy of MCG makes our network benefit from better segmentation mask. When CRF [22] is applied in post-processing in the segmentation testing phase, the performance difference reduced to 2%.

Visual quality of the four strategies is compared in Fig. 6. Using MCG proposal masks, the learned network predicts semantic segmentation results with reasonable contours. With CRF, the object contour becomes even clearer.

#### 5.5 Comparison Regarding Different Weak Annotation Strategies

We compare with other weakly-supervised methods using different ways of annotation. The results are listed in Table 4. This is to demonstrate the performance gap among strategies using image-level annotation and other strategies with more information for supervision. The comparison with that of [10] is not that fair because the network deployed in this method uses the overfeat [37] architecture. Moreover, This method uses 700k images to train the network, while all others take only 10k images.



**Fig. 6.** Results of using different proposal generation methods. (c) and (e) are results using the MCG and selective search proposal masks respectively. (d) and (f) are the corresponding results incorporating CRF [22] post-processing in the testing phase.

Except for [10], all other networks are built on top of VGG-16 [32]. Our method outperforms all other image-level weak supervision methods [10, 11, 27] with more than 10% mean IoU difference. Our method even performs better than the method with point supervision, which needs extra point annotation. Compared with the box supervised methods [6, 11] and scribble supervised method [23], our method degrades gracefully with regard to the annotation effort needed for getting bounding boxes or scribbles. Nevertheless, image level labels are the easiest to obtain without any image touch-up.

## 5.6 More Comparison Regarding Image-Level Supervision

We provide more comparison with other image-level weakly-supervised semantic segmentation solutions. Performance was evaluated on the PASCAL VOC 2012 *test* set. Results are listed in Table 5. WSSL [12] and MIL-FCN [11] are built on VGG16 [32]. Our method performs better by a large margin of 15%, which shows the effectiveness of utilizing object localization to provide extra information during training.

Our method does not have the issues of the EM based method [12] since we utilize two branches to get extra supervision for next round optimization. Our process can be fairly effective by enjoying possible pixel-level information. Our method behaves on par with the transferable learning method [39], which uses 60k images with pixel-level annotation to learn the transferable knowledge. We show our result on Pascal VOC 2012 validation set in Fig. 7. The figure shows that our network can well capture the shape of objects. By using CRF [22] in the testing phase, more details along boundaries are revealed.

**Table 4.** Comparison of weakly-supervised semantic segmentation methods using different ways for annotation on the PASCAL VOC 2012 validation dataset. \* denotes CRF [22] applied in post-processing.

Methods	Annotation	Mean IoU%
MIL-FCN [27]	Image level	25.1
MIL-sppxl [10]	Image level	38.6
MIL-obj [10]	Image level	37.8
MIL-seg [10]	Image level	42.0
WSSL* [11]	Image level	38.2
Point supervision [21]	Spot	46.1
BoxSup* [6]	Bx	62.0
WSSL* [11]	Box	60.6
Scribblesup* w/o pairwise terms [23]	Scribbles	60.5
Scribblesup* w/ pairwise terms [23]	Scribbles	63.1
Ours+ss	Image level	46.98
Ours+mcg	Image level	50.41
Ours+ss*	Image level	52.62
Ours+mcg*	Image level	54.34

**Table 5.** Comparison with state-of-the-art weakly supervised methods on the PASCAL VOC 2012 test set. \* denotes CRF [22] used in post-processing in the testing phase.

Methods	Annotation	Mean IoU%
MIL-sppxl [10]	Image level 700k	35.8
MIL-obj [10]	Image level 700k	37.0
MIL-seg [10]	Image level 700k	40.6
MIL-FCN [11]	Image level 10k	24.9
WSSL* [12]	Image level 10k	39.6
CCNN [14]	Image level 10k	35.6
CCNN+size [14]	Image level 10k+size	43.3
CCNN+size* [14]	Image level 10k +size	45.1
SN.B [38]	Image level 10k	43.2
STC* [25]	Image level 60k	51.2
TransferNet [39]	Image level 10k +pixel level 60k	51.2
Ours+ss	Image level 10k	47.8
Ours+mcg	Image level 10k	50.6
Ours+ss*	Image level 10k	52.7
Ours+mcg*	Image level 10k	<b>55.5</b>



**Fig. 7.** Our network result is the output from the semantic segmentation branch up-sampled to the original image resolution. Network+CRF means utilizing CRF [22] in post processing.

## 6 Conclusion and Future Work

We have proposed a weakly supervised method for semantic segmentation under image-level supervision. Our method includes the semantic segmentation and object localization branches in a unified framework, in which the two branches profit each other. With our designed object proposal aggregation and proposal selection modules, the positive feedback from the two branches can be augmented. Such augmented positive feedback greatly improve the image level weakly supervised semantic segmentation task. As a byproduct, object localization performance gets also improved. In future work, we will exploit the framework for other weakly supervised learning tasks, e.g. simultaneous segmentation and detection [18], and extend our method to the semi-supervised setting. Further, since our annotation is economical, scaling our segmentation method up to

include thousands of object categories is also a target to pursue. Finally, we plan to extend our framework to handle more challenging context segmentation (scene parsing) tasks which include both objects and stuff.

**Acknowledgments.** This work is supported by a grant from the Research Grants Council of the Hong Kong SAR (project No. 2150760) and by the National Science Foundation China, under Grant 61133009. We thank the anonymous reviewers for their suggestive comments and valuable feedback, and Mr. Zhuotun Zhu for the helpful discussion regarding the topic of multiple instance learning.

## References

1. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
2. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv (2014)
3. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.: Conditional random fields as recurrent neural networks. In: ICCV (2015)
4. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: CVPR (2015)
5. Dai, J., He, K., Sun, J.: Convolutional feature masking for joint object and stuff segmentation. In: CVPR (2015)
6. Dai, J., He, K., Sun, J.: Boxesup: exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: ICCV (2015)
7. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis. IJCV* **88**(2), 303–338 (2010)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: CVPR (2009)
9. 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: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 740–755. Springer, Heidelberg (2014)
10. Pinheiro, P.O., Collobert, R.: From image-level to pixel-level labeling with convolutional networks. In: CVPR (2015)
11. Pathak, D., Shelhamer, E., Long, J., Darrell, T.: Fully convolutional multi-class multiple instance learning. arXiv (2014)
12. Papandreou, G., Chen, L.C., Murphy, K., Yuille, A.L.: Weakly-and semi-supervised learning of a DCNN for semantic image segmentation. arXiv (2015)
13. Xu, J., Schwing, A.G., Urtasun, R.: Learning to segment under various forms of weak supervision. In: CVPR (2015)
14. Pathak, D., Krahenbuhl, P., Darrell, T.: Constrained convolutional neural networks for weakly supervised segmentation. In: ICCV (2015)
15. Song, H.O., Girshick, R., Jegelka, S., Mairal, J., Harchaoui, Z., Darrell, T.: On learning to localize objects with minimal supervision. arXiv (2014)
16. Kumar, M.P., Packer, B., Koller, D.: Self-paced learning for latent variable models. In: NIPS (2010)

17. Deselaers, T., Alexe, B., Ferrari, V.: Localizing objects while learning their appearance. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 452–466. Springer, Heidelberg (2010)
18. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part VII. LNCS, vol. 8695, pp. 297–312. Springer, Heidelberg (2014)
19. Liu, Z., Li, X., Luo, P., Loy, C.C., Tang, X.: Semantic image segmentation via deep parsing network. In: ICCV (2015)
20. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: ICCV (2015)
21. Russakovsky, O., Bearman, A.L., Ferrari, V., Li, F.F.: What’s the point: semantic segmentation with point supervision. arXiv (2015)
22. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. arXiv (2012)
23. Lin, D., Dai, J., Jia, J., He, K., Sun, J.: Scribblesup: scribble-supervised convolutional networks for semantic segmentation. In: CVPR (2016)
24. Vezhnevets, A., Buhmann, J.M.: Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In: CVPR (2010)
25. Wei, Y., Liang, X., Chen, Y., Shen, X., Cheng, M.M., Zhao, Y., Yan, S.: STC: a simple to complex framework for weakly-supervised semantic segmentation. arXiv (2015)
26. Wang, X., Zhu, Z., Yao, C., Bai, X.: Relaxed multiple-instance svm with application to object discovery. In: ICCV (2015)
27. Cinbis, R.G., Verbeek, J., Schmid, C.: Weakly supervised object localization with multi-fold multiple instance learning. arXiv (2015)
28. Song, H.O., Lee, Y.J., Jegelka, S., Darrell, T.: Weakly-supervised discovery of visual pattern configurations. In: NIPS (2014)
29. Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. *Int. J. Comput. Vis. (IJCV)* **104**(2), 154–171 (2013)
30. Arbeláez, P., Pont-Tuset, J., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR (2014)
31. Carreira, J., Sminchisescu, C.: Constrained parametric min-cuts for automatic object segmentation. In: CVPR (2010)
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv (2014)
33. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. (TOG)* **23**(3), 309–314 (2004)
34. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR (2014)
35. Hong, S., Noh, H., Han, B.: Decoupled deep neural network for semi-supervised semantic segmentation. In: NIPS (2015)
36. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: *Multimedia* (2014)
37. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv (2013)
38. Wei, Y., Liang, X., Chen, Y., Jie, Z., Xiao, Y., Zhao, Y., Yan, S.: Learning to segment with image-level annotations. *PR* (2016)
39. Hong, S., Oh, J., Han, B., Lee, H.: Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. arXiv (2015)