Reliability Does Matter: An End-to-End Weakly Supervised Semantic Segmentation Approach

Bingfeng Zhang,[1] **Jimin Xiao,**[1*] **Yunchao Wei,**[2] **Mingjie Sun,**[1] **Kaizhu Huang**[1,3]
[1]Xi'an Jiaotong-liverpool University, Suzhou, China
[2]University of Technology Sydney, Australia
[3]Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou, China
[1]{Bingfeng.Zhang, jimin.xiao, mingjie.sun18, kaizhu huang}@xjtlu.edu.cn, [2]yunchao.wei@uts.edu.au

Abstract

Weakly supervised semantic segmentation is a challenging task as it only takes image-level information as supervision for training but produces pixel-level predictions for testing. To address such a challenging task, most recent state-of-the-art approaches propose to adopt two-step solutions, *i.e.* 1) learn to generate pseudo pixel-level masks, and 2) engage FCNs to train the semantic segmentation networks with the pseudo masks. However, the two-step solutions usually employ many bells and whistles in producing high-quality pseudo masks, making this kind of methods complicated and inelegant. In this work, we harness the image-level labels to produce reliable pixel-level annotations and design a fully end-to-end network to learn to predict segmentation maps. Concretely, we firstly leverage an image classification branch to generate class activation maps for the annotated categories, which are further pruned into confident yet tiny object/background regions. Such reliable regions are then directly served as ground-truth labels for the parallel segmentation branch, where a newly designed dense energy loss function is adopted for optimization. Despite its apparent simplicity, our one-step solution achieves competitive mIoU scores (*val*: 62.6, *test*: 62.9) on Pascal VOC compared with those two-step state-of-the-arts. By extending our one-step method to two-step, we get a new state-of-the-art performance on the Pascal VOC (*val*: 66.3, *test*: 66.5).

Introduction

Recently, weakly supervised semantic segmentation receives great interest and is being extensively studied. Requiring merely low degree (cheaper or simpler) annotations including scribbles (Lin et al. 2016; Vernaza and Chandraker 2017; Tang et al. 2018b), bounding boxes (Dai, He, and Sun 2015; Khoreva et al. 2017), points (Maninis et al. 2018; Bearman et al. 2016) and image-level labels (Ahn and Kwak 2018; Hou et al. 2018; Wei et al. 2018) for training, weakly supervised semantic segmentation offers a much easy way than its fully supervised counterpart adopting pixel-level masks (Chen et al. 2018a; 2017; Long, Shelhamer, and Darrell 2015). Among these weakly supervised labels, the image-level annotation is the simplest one to collect yet also the most challenging case since there is no direct mapping between semantic labels and pixels.

To learn semantic segmentation models using image-level labels as supervision, many existing approaches can be categorized as one-step approaches and two-step approaches. One-step approaches (Papandreou et al. 2015) often establish an end-to-end framework, which augments multi-instance learning with other constrained strategies for optimization. This family of methods is elegant and easy to implement. However, one significant drawback of these approaches is that the segmentation accuracy is far behind their fully supervised counterparts. To achieve better segmentation performance, many researchers alternatively propose to leverage two-step approaches (Wei et al. 2017; Huang et al. 2018). This family of approaches usually aim to take bottom-up (Hou et al. 2017) or top-down (Zhang et al. 2018a; Zhou et al. 2016) strategies to firstly generate high-quality pseudo pixel-level masks with image-level labels as supervision. These pseudo masks then act as ground-truth and are fed into the off-the-shelf fully convolutional networks such as FCN (Long, Shelhamer, and Darrell 2015) and Deeplab (Chen et al. 2014; 2018a) to train the semantic segmentation models. Current state-of-the-arts are mainly two-step approaches, with segmentation performance approaching that of their fully supervised counterparts. However, to produce high-quality pseudo masks, these approaches often employ many bells and whistles, such as introducing additional object/background cues from object proposals (Pinheiro and Collobert 2015) or saliency maps (Jiang et al. 2013) in an off-line manner. Therefore, the two-step approaches are usually very complicated and hard to be re-implemented, limiting their application to research areas such as object localization and video object tracking.

In this paper, we present a simple yet effective one-step approach, which can be easily trained in an end-to-end manner. It achieves competitive segmentation performance compared with two-step approaches. Our approach named *Reliable Region Mining* (*RRM*) includes two branches: one to produce pseudo pixel-level masks using image-level annotations, and the other to produce the semantic segmentation results. In contrast to the previous two-step state-of-the-arts (Ahn and Kwak 2018; Lee et al. 2019) that prefer to mine dense and

[*]Corresponding author

integral object regions, our *RRM* only chooses those confident object/background regions that are usually tiny but with high response scores on the class activation maps. We find these regions can be further pruned into more reliable ones by augmenting an additional CRF operation, which are then employed as supervision for the parallel semantic segmentation branch. With limited pixels as supervision, we designed a regularized loss named dense energy loss, which cooperates with the pixel-wise cross-entropy loss to optimize the training process.

Despite its apparent simplicity, our one-step *RRM* achieves 62.6 and 62.9 of mIoU scores on the Pascal VOC *val* and *test* sets, respectively. These results achieve state-of-the-art performance and it is even competitive compared with those two-step state-of-the-arts, which usually adopt complex bells and whistles to produce pseudo masks. We believe that our proposed *RRM* offers a new insight to the one-step solution for weakly supervised semantic segmentation. Besides, in order to show the effectiveness of our method, we also extend our method to a two-step framework and get a new state-of-the-art performance with 66.3 and 66.5 on the Pascal VOC *val* and *test* sets. Code will be made publicly available.

Related Work

Semantic segmentation is an important task in computer vision (Wei et al. 2018; Xiao et al. 2019; Xie et al. 2018), which requires to predict pixel-level classification. Long *et al.* (Long, Shelhamer, and Darrell 2015) proposed the first fully convolutional network for semantic segmentation. Chen *et al.* (Chen et al. 2014) proposed a new deep neural network structure named ”Deeplab” to conduct pixel-wise prediction using atrous convolution, and a series of new network structures was developed after that (Chen et al. 2018a; 2017; 2018b). However, fully supervised semantic segmentation requires dense pixel-level annotations, which cost expensive human expense. Weakly supervised semantic segmentation has been drawing much attention as less human intervention is needed. There are different categories of weakly supervised semantic segmentation based on the types of supervision: scribble (Tang et al. 2018a; Lin et al. 2016), bounding box (Song et al. 2019; Hu et al. 2018; Rajchl et al. 2017), point (Maninis et al. 2018; Bearman et al. 2016) and image-level class label (Zhang et al. 2018b; Vernaza and Chandraker 2017; Zhang et al. 2018c). In this paper, we focus on image-level supervised semantic segmentation.

Image-level weakly supervised semantic segmentation only provides image-level annotation. Most recent approaches are based on class activation map (CAM) (Zhou et al. 2016), which is to generate initial object seeds or regions from image-level annotation. Such initial object seeds or regions are converted to generate pseudo labels to train a semantic segmentation model. Wei *et al.* (Wei et al. 2017) proposed to erase iteratively the discriminative areas computed by a classification network so that more seed regions can be mined which are then combined with a saliency map to generate the pseudo pixel-level label. Wei *et al.* (Wei et al. 2018) also proved that dilated convolution can increase the receptive filed and improve the weakly segmentation network performance. Besides, Wang *et al.* (Wang et al. 2018) trained a region network and a pixel network to make prediction from image level to region level and from region level to pixel level gradually. Also, this method takes saliency map as extra supervision. Ahn and Suha (Ahn and Kwak 2018) designed an affinity network to compute the relationship between different image pixels and exploited this network to get the pseudo object labels for segmentation model training. Huang *et al.* (Huang et al. 2018) deployed a traditional algorithm named seed growing to iteratively expand the seed regions.

However, all the above methods produced high-quality pseudo masks using a wide varieties of techniques, meaning that we need at least one or two extra networks before training FCNs for semantic segmentation prediction. In this work, we try to design one single network for the whole task to simplify the process. We believe this work offers a new perspective for the image-level weakly supervised semantic segmentation task.

Proposed Method

Overview

Our proposed *RRM* can be divided into two parallel branches including a classification branch and a semantic segmentation branch. Both branches share the same backbone network, and during training both of them update the whole network at the same time. The overall framework of our method is illustrated in Figure 1. The algorithm flow is illustrated in Algorithm 1.

- The classification branch is used to generate reliable pixel-level annotations. Original CAMs will be processed to generate reliable yet tiny regions. The final remained reliable regions are regarded as labeled regions, while other regions are viewed as unlabeled. These labels are used as supervision information for the semantic segmentation branch for training.
- The semantic segmentation branch is used to predict pixel-level labels. This branch deploys a new joint loss function combining the cross entropy loss with a newly designed dense energy loss. The cross entropy loss mainly considers labeled pixels, while the dense energy loss takes into account all pixels by making full use of RGB color and pixel positions.

The overall loss function of our *RRM* is: $\mathcal{L} = \mathcal{L}_{class} + \mathcal{L}_{joint\_seg}$, where $\mathcal{L}_{class}$ represents a conventional classification softmax loss, while $\mathcal{L}_{joint\_seg}$ is a newly introduced joint loss for the segmentation branch.

Classification Branch: Generating Labels for Reliable Regions

High-quality pixel-level annotation has a direct impact on our final semantic segmentation performance as it is the only ground-truth in the training processing. Original CAMs can highlight the most discriminative regions of an object, but they still contain some non-object areas, which are the mislabeled pixels. Therefore, after getting the original CAM regions, post-processing such as dense CRF (Krähenbühl and

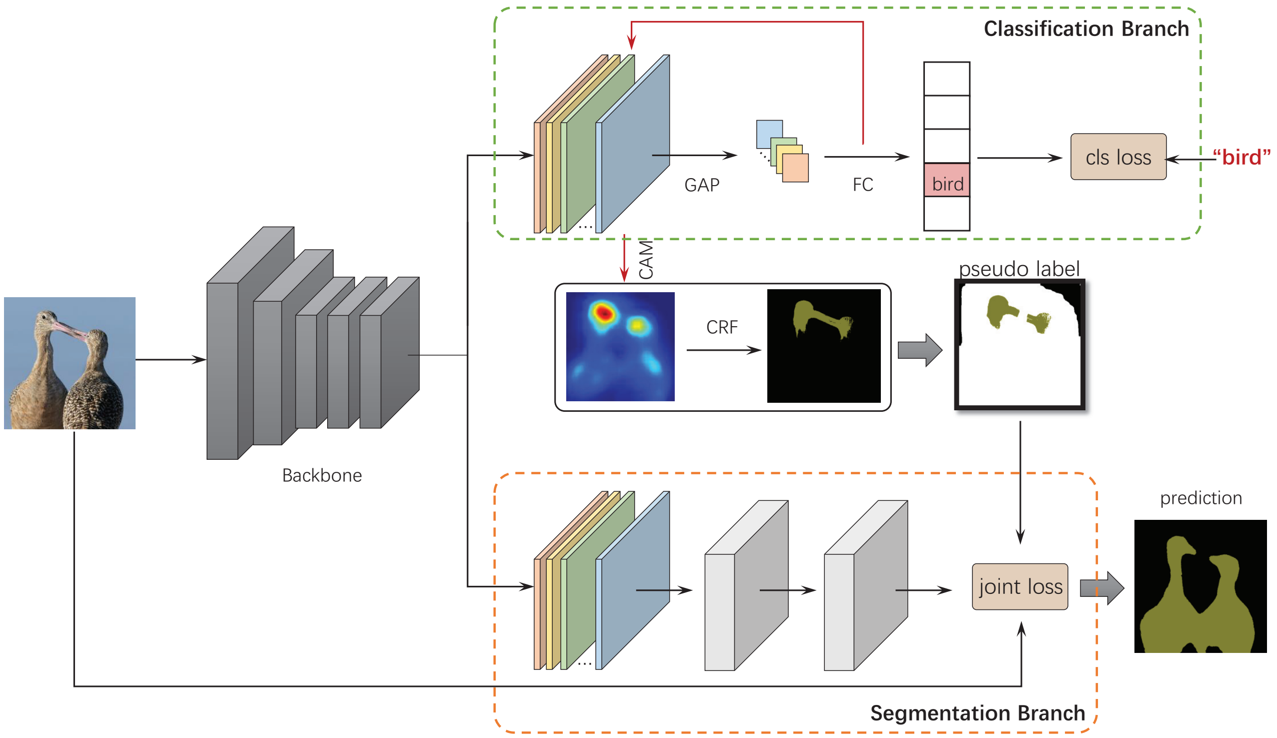

Figure 1: The framework of our proposed *RRM* network. First of all, original regions are calculated through the classification branch, then the pseudo pixel-level masks are generated. Finally, the pseudo labels are applied as supervision to train the semantic segmentation branch. The whole RRM is jointly optimized end-to-end via a standard back-propagation algorithm during training.

Koltun 2013) is needed. We followed this basic idea and do further process for generating the reliable labels.

We compute the initial CAMs of the training dataset following (Zhou et al. 2016). In our network, global average pooling (GAP) is applied on the last convolution layer. The output of GAP is classified with a fully-connected layer. Finally, the fully-connected layer weights are used on the last convolution layer to obtain the heatmap for each class. Besides, inspired by the fact that dilated convolution can increase the respective field (Wei et al. 2018), we add dilated convolution into the last three layers. Details of our network settings are reported in our experiment Section.

Mathematically, given an image *I*, the CAM of class c is:

$$M_{ocam}^c = RS(\sum_{ch=0}^{D} \omega_{ch}^c \cdot F_{ch}), (c \in C_{fg}), \quad (1)$$

where $C_{fg} = \{c_1, c_2, ..., c_N\}$ includes all foreground classes, M_{ocam}^c is the CAM of class c for image I, ω^c denotes the weights of the fully-connected layer for class c, and F is the feature maps from the last convolution layer of the backbone. $RS(\cdot)$ is an operation to resize the input to the shape of I.

Using multi-scale of original images is beneficial for generating a stable CAM. Given I and it is scaled by a factor s_i, $s_i \in \{s_0, s_1, ..., s_n\}$, the multi-scale CAM for I is detonated as:

$$M_{cam}^c = \sum_{i=0}^{n} (M_{ocam}^c(s_i)/(n+1)), \quad (2)$$

where $M_{ocam}^c(s_i)$ is the CAM of class c for the scaled image I with a factor s_i. Figure 2 shows that compared to original CAM (scale=1), the multi-scale CAM provides more accurate object localization.

The CAM scores are normalized, so that we can get the classification probabilities for each pixel in *I*,

$$P_{fg}^c = M_{cam}^c / max(M_{cam}^c), (c \in C_{fg}), \quad (3)$$

where $max(M_{cam}^c)$ is the maximum value in the CAM of class c_j.

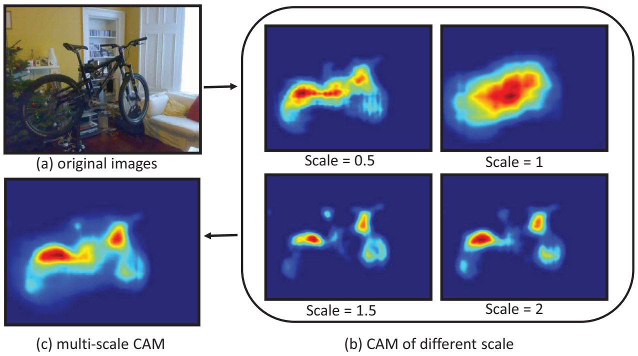

Figure 2: An example of computing multi-scale CAM.

The background score is calculated using a similar way as in (Ahn and Kwak 2018):

$$P_{bg}(i) = (1 - \max_{c \in C_{fg}} (p^c_{fg}(i))^{\gamma}, \gamma > 1. \tag{4}$$

where i is the pixel position index, γ is the decay rate which helps to suppress background labels. The overall probability map, namely P_{fg_bg}, is obtained by concatenating foreground and background probabilities P_{fg} and P_{bg}.

After that, we use the dense CRF (Krähenbühl and Koltun 2013) as post-processing to remove some mislabeled pixels, and the CRF pixel label map is:

$$I_{crf} = CRF(I, [P_{fg}, P_{bg}]). \tag{5}$$

The selected reliable CAM label is:

$$I_{cam}(i) = \begin{cases} \operatorname*{argmax}_{c\in C}(P^c_{fg\_bg}(i)), & \text{if } \max_{c\in C}(P^c_{fg\_bg}(i)) > \alpha \\ 255, & else \end{cases}, \tag{6}$$

where $C = \{c_0, c_1, ..., c_N\}$ includes all classes and the background (c_0). 255 means the class label is not decided yet.

The final pixel label input to the semantic segmentation branch is:

$$I_{final}(i) = \begin{cases} I_{cam}(i), & \text{if } I_{cam}(i) = I_{crf}(i) \\ 255, & else \end{cases} \tag{7}$$

In (6), $\max_{c\in C}(P^c_{fg\_bg}(i)) > \alpha$ selects the highly confident regions. In (7), $I_{crf}(i) = I_{cam}(i)$ considers the CRF constrains. Taking this strategy, highly reliable regions as well as their labels can be obtained. The regions which are detonated as 255 in (7) are regarded as unreliable regions.

Figure 3 shows an example of our approach. It is observed that the original CAM labels (Figure 3 (c)) contain most foreground labels but introduce a number of background pixels as foreground. The CRF label (Figure 3 (d)) can get accurate boundary but at the same time, many foreground pixels are regarded as background. In other words, the CAM label can provide reliable background pixels and CRF label can provide reliable foreground pixels. Combing the CAM label and CRF label map using our method, some wrong pixel-level labels are removed while the reliable regions are still remained, which is especially obvious at the object boundaries (see the difference between Figure 3 (e) and (f)).

Semantic Segmentation Branch: Making Predictions

After getting the reliable pixel-level annotations, they are used as labels for our semantic segmentation branch. Different from the other methods which train their semantic segmentation network with the integral pseudo labels independently, our segmentation branch shares the same backbone network with the classification branch, needing only reliable yet tiny pixel-level labels. Our loss function consists of a cross entropy loss and a energy loss. Cross entropy loss focuses on utilizing the labeled data while the energy loss considers both labeled and unlabeled data. The joint loss is:

$$\mathcal{L}_{joint\_seg} = \mathcal{L}_{ce} + \mathcal{L}_{energy}. \tag{8}$$

In (8), $\mathcal{L}_{ce}$ and $\mathcal{L}_{energy}$ represent the cross entropy loss and the dense energy loss, respectively. The cross entropy loss is:

$$\mathcal{L}_{ce} = -\sum_{c\in C, i\in\Phi} B_c(i) log(P^c_{net}(i)), \tag{9}$$

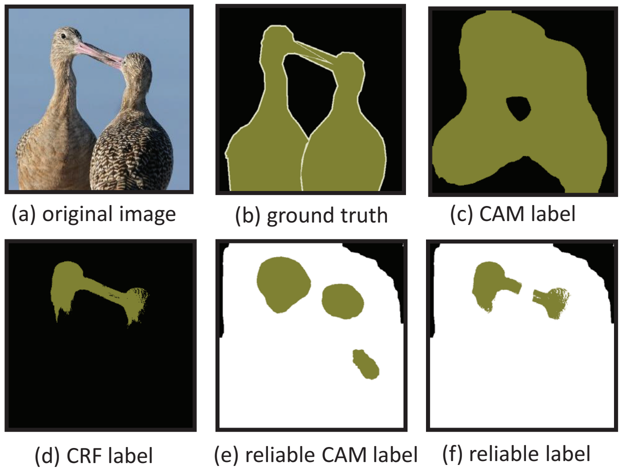
(a) original image
(b) ground truth
(c) CAM label
(d) CRF label
(e) reliable CAM label
(f) reliable label

Figure 3: An example of generating reliable pixel labels. (c) is computed only considering the corresponding class label of P_{fg_bg}. (d) is the result of (5), (e) and (f) are generating through (6) and (7), respectively. The white pixels in (e) and (f) are the unreliable regions.

where $B_c(i)$ is a binary indicator, which equals to 1 if the label of pixel i is c and otherwise 0; Φ denotes the labeled regions, $\Phi = \{i | I_{final}(i) \neq 255\}$; $P^c_{net}(i)$ is the output probability of the trained network

So far, all labeled pixels has been used for training with cross entropy loss, but there are a large number of unlabeled pixels. In order to make predictions for those unlabeled regions, we design a new shallow loss named dense energy loss considering both RGB colors and spatial positions.

We firstly define the energy formulation between pixel i and j based on (Joy et al. 2019):

$$E(i,j) = \sum_{\substack{c_a, c_b \in C \\ c_a \neq c_b}} G(i,j) P^{c_a}_{net}(i) P^{c_b}_{net}(j). \tag{10}$$

In (10), both c_a and c_b are the class labels, $P^{c_a}_{net}(i)$ and $P^{c_b}_{net}(j)$ are the softmax output of our segmentation branch at pixel i and j, respectively. $G(i,j)$ is a Gaussian kernel bandwidth filter:

$$G(i,j) = \frac{1}{W} exp(-\frac{\|D_i - D_j\|^2}{2\sigma_d^2} - \frac{\|I_i - I_j\|^2}{2\sigma_r^2}), \tag{11}$$

where $\frac{1}{W}$ is the normalized weights, D is the pixel spatial position while I is the RGB color. σ_d and σ_r are hyper parameters which control the scale of Gaussian kernels. (10) can be simplified using Potts model (Tang et al. 2018b):

$$\begin{aligned} E(i,j) &= \sum_{\substack{c_a, c_b \in C \\ c_a \neq c_b}} G(i,j) P^{c_a}_{net}(i) P^{c_b}_{net}(j) \\ &= G(i,j) \sum_{c\in C} P^c_{net}(i)(1 - P^c_{net}(j)). \end{aligned} \tag{12}$$

Finally, our dense energy loss can be written as:

$$\mathcal{L}_{energy} = \sum_{i=0}^{N} \sum_{\substack{j=0 \\ j \neq i}}^{N} S(i)E(i,j). \tag{13}$$

In (13), considering the fact that cross entropy loss is designed for supervised learning with label information 100% accurate, but in this task, all pixel labels are not 100% reliable, which means that using cross entropy loss might introduce some errors. Thus, our dense energy loss is applied to mitigate this problem. Based on this idea, we design a soft filter $S(i)$ for pixel i:

$$S(i) = \begin{cases} 1 - \max\limits_{c \in C}(P_{net}^{c}(i)), & i \in \Phi \\ 1, & else \end{cases} \tag{14}$$

Algorithm 1 Algorithm flow of our proposed approach.

```
Input:  Images $I$ with their image-level class labels $C_{fg}$;
Output: The trained end-to-end network, $Net$;
while iteration is true do
    Use the classification Network branch to get the original CAMs;
    Get the multi-scale CAMs with (2) for each class;
    Use (3) and (4) to get foreground probability $P_{fg}$ and background probability $P_{bg}$;
    Get the overall CAM probability map $P_{fg\_bg}$ by combining $P_{bg}$ and $P_{fg}$;
    Calculate reliable CAM label $I_{cam}$ and CRF label $I_{crf}$;
    Get the reliable regions and label $I_{final}$ from $I_{cam}$ and $I_{crf}$ using (6)(7);
    Produce predictions and update the whole network using loss function $\mathcal{L} = \mathcal{L}_{class} + \mathcal{L}_{ce} + \mathcal{L}_{energy}$;
end while
```

Experiments

Dataset and Implementation Details

Dataset. Our *RRM* is trained and validated on PASCAL VOC 2012 (Everingham et al. 2010) as well as its augmented data, including $10,582$ images for training, $1,449$ images for validating and $1,456$ images for testing. Mean intersection over union (mIoU) is considered as the evaluation criterion.

Implementation Details. The backbone network is a ResNet model with 38 convolution layers (Wu, Shen, and Van Den Hengel 2019). We remove all the fully connected layers of the original network and engage dilated convolution for the last three resnet blocks (a resnet block is a set of residual units with the same output size), the dilated rate is 2 for the last third layer, and 4 for the last 2 layers. For the semantic segmentation branch, we add two dilation convolution layers of the same configuration after the backbone (Wu, Shen, and Van Den Hengel 2019), with kernel size 3, dilated rate 12, and padding size 12. Cross entropy loss is computed for background and foreground individually. σ_d and σ_r in our dense energy loss are set as 15 and 100, respectively.

The training learning rate is 0.001 with weight decay being $5e$-4. The training images are resized with a ratio randomly sampled from $(0.7, 1.3)$, and they are randomly flipped. Finally, they are normalized and randomly cropped to size 321*321.

To generate reliable regions, the scale ratio in (2) is set to $\{0.5, 1, 1.5, 2\}$, γ in (5) is set to 4 for P_{fg_bg}. The CRF parameters in (5) follow the setting in (Ahn and Kwak 2018). In (6), an α value is chosen with 40% pixels selected as labeled pixels for each class. During validating and testing, dense CRF is applied as a post-processing method, and the parameters are set as the default values given in (Huang et al. 2018). During training, both two branches update the backbone network. During testing, only the segmentation branch is used to produce the predictions.

Reproducibility: PyTorch (Paszke et al. 2017) was used. All the experiments were performed on NVIDIA RTX 2080 Ti. Code now is available at: https://github.com/zbf1991/RRM.

Analysis of Our Approach

Our *RRM* has two important aspects: using the reliable yet tiny pseudo masks for supervision and a new joint loss function for end-to-end training. Ablation studies are conducted to illustrate their individual and joint effectiveness, with results reported in Table 1 and Table 2.

Ratio	0.1	0.2	0.3	0.4	0.6	0.8	1.0
CE loss	0.486	0.487	0.484	0.485	0.483	0.495	0.557
Joint loss	0.428	0.623	0.626	0.626	0.609	0.594	0.582

Table 1: Performance on PASCAL VOC 2012 *val* set based on different mined region. Ratio means the proportion of reliable regions which is mined by our method to the whole pixels. ”CE loss” means only cross entropy loss was used for our segmentation branch and ”Joint loss” means our dense energy loss was combined with cross entropy loss was used for the segmentation branch.

We firstly validate the influence of different pseudo mask size. We do this by changing α. Table 1 reports the results. A smaller pseudo mask size means that more reliable regions are selected for the segmentation branch, while a larger size means that fewer reliable pixels are labeled. Table 1 demonstrates that 20%-60% labeled pixels lead to the best performance. On one hand, too few labeled pixels cannot get satisfied performance since the segmentation network cannot get enough labels for learning. On the other hand, too many labeled pixels means more incorrect labels are used, which are noise for the training processing.

	CE loss	Joint loss
CAM	0.461	0.557
Ours-RRM	0.485	0.626

Table 2: Analysis of our method. CAM means class activate maps directly as pseudo masks. Ours-*RRM* means that we used our method to produce pseudo masks. Both CAM and ours-*RRM* use top 40% pixels according to Table 1.

End-to-End Method	bkg	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mIOU
EM-Adapt (Papandreou et al. 2015)	67.2	29.2	17.6	28.6	22.2	29.6	47.0	44.0	44.2	14.6	35.1	24.9	41.0	34.8	41.6	32.1	24.8	37.4	24.0	38.1	31.6	33.8
Ours-RRM (one-step)	**87.9**	**75.9**	**31.7**	**78.3**	**54.6**	**62.2**	**80.5**	**73.7**	**71.2**	**30.5**	**67.4**	**40.9**	**71.8**	**66.2**	**70.3**	**72.6**	**49.0**	**70.7**	**38.4**	**62.7**	**58.4**	**62.6**

Table 3: Performance on the PASCAL VOC 2012 *val* set, compared with other end-to-end weakly supervised approaches.

End-to-End Method	bkg	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mIOU
EM-Adapt (Papandreou et al. 2015)	76.3	37.1	21.9	41.6	26.1	38.5	50.8	44.9	48.9	16.7	40.8	29.4	47.1	45.8	54.8	28.2	30.0	44.0	29.2	34.3	46.0	39.6
Ours-RRM (one-step)	**87.8**	**77.5**	**30.8**	**71.7**	**36.0**	**64.2**	**75.3**	**70.4**	**81.7**	**29.3**	**70.4**	**52.0**	**78.6**	**73.8**	**74.4**	**72.1**	**54.2**	**75.2**	**50.6**	**42.0**	**52.5**	**62.9**

Table 4: Performance on the PASCAL VOC 2012 *test* set, compared with other end-to-end weakly supervised approaches.

Method	Baseline	Sup.	Extra Data	End-to-end	val (mIoU)	test (mIoU)
Deeplab (ICLR'15) (Chen et al. 2014)	VGG-16	F	-	-	67.6	70.3
Deeplab-v2 (Chen et al. 2018a)	ResNet-101	F	-	-	76.8	79.7
WSSL (ICCV'15) (Papandreou et al. 2015)	VGG-16	B	-	-	60.6	62.2
BoxSup (ICCV'15) (Dai, He, and Sun 2015)	VGG-16	B	-	-	62.0	64.6
ScribbleSup (CVPR'16) (Lin et al. 2016)	VGG-16	S	-	-	63.1	-
Kernel Cut (ECCV'18) (Tang et al. 2018b)	ResNet-101	S	-	-	75.0	-
CrawlSeg (CVPR'17) (Hong et al. 2017)	VGG-16	L	YouTube Videos	×	58.1	58.7
DSRG (CVPR'18) (Huang et al. 2018)	VGG-16	L	MSRA-B	×	59.0	60.4
DSRG (CVPR'18) (Huang et al. 2018)	ResNet-101	L	MSRA-B	×	61.4	63.2
FickleNet (CVPR'19) (Lee et al. 2019)	ResNet-101	L	MSRA-B	×	64.9	65.3
EM-Adapt (ICCV'15)(Papandreou et al. 2015)	VGG-16	L	-	✓	38.2	39.6
SEC (ECCV'16) (Kolesnikov and Lampert 2016)	VGG-16	L	-	×	50.7	51.7
AugFeed (ECCV'16) (Qi et al. 2016)	VGG-16	L	-	×	54.3	55.5
AdvErasing (CVPR'17) (Wei et al. 2017)	VGG-16	L	-	×	55.0	55.7
AffinityNet (CVPR'18) (Ahn and Kwak 2018)	VGG-16	L	-	×	58.4	60.5
AffinityNet (CVPR'18) (Ahn and Kwak 2018)	ResNet-38	L	-	×	61.7	**63.7**
Ours-RRM-VGG (two-step)	VGG16	L	-	×	**60.7**	**61.0**
Ours-RRM-ResNet (two-step)	ResNet-101	L	-	×	**66.3**	**66.5**
Ours-RRM (one-step)	ResNet-38	L	-	✓	**62.6**	62.9

Table 5: Comparison with the state-of-the-art approaches on PASCAL VOC 2012 *val* and *test* dataset. Sup.-supervision information, GT-ground truth, F-full supervision, L-image-level class label, B-bounding box label, S-scribble label.

Table 2 shows the effectiveness of our introduced two main parts: reliable region mining and the joint loss. The results obtained using original CAM regions and the mined reliable regions with *RRM* are compared. It is observed that the pseudo label generated by *RRM* outperforms CAM labels. If we remove the joint loss from our segmentation branch, it also shows that the reliable pseudo labels generated by *RRM* improves the segmentation performance.

In addition, the comparison between *Ours-RRM* with CE loss and *Ours-RRM* with Joint loss in Table 2 illustrates the effectiveness of the introduced joint loss. Without the joint loss, the mIoU obtained with *RRM* with CE loss gets lower. This is because the mined reliable regions with *RRM* cannot provide sufficient labels for segmentation model training when only considering cross entropy loss. After adopting the joint loss, segmentation performance improves with a big margin from 48.5 to 62.6, which is a 14.1 increase. Similar comparison result is obtained between *CAM* with CE loss and *CAM* with Joint loss.

Comparisons with Previous Approaches

In Table 3 and Table 4, we make detailed comparisons with other end-to-end network for image-level-only supervised semantic segmentation. Although there are various different networks for this task, only EM-Adapt (Papandreou et al. 2015) adopts an end-to-end structure, and it can be seen that *Ours-RRM (one-step)* outperforms it with a big margin. First of all, compared with EM-Adapt (Papandreou et al. 2015), which uses an expectation–maximization (EM) algorithm to update the network parameters, our method adopts a more direct and explicit learning procedure to update the whole network, using our designed joint loss function. Secondly, EM-Adapt (Papandreou et al. 2015) can only give a rough segmentation result as only the image-level information is considered, while *Ours-RRM (one-step)* designs a pilot mechanism to provide reliable pixel-level labels, which leads to more accurate segmentation predictions.

In order to show the effectiveness and scalability of our idea, we also extend our method to a two-step framework. The difference is that for our one-step method (*Ours-RRM (one-step)*), we produce the predictions through our segmentation branch directly. Whereas for our two-step method, we firstly used our *Ours-RRM (one-step)* network to produce the pseudo masks for the training dataset. Following that, we train and evaluate Deeplab (Chen et al. 2014) with those generated pixel labels, which is named as *Our-RRM-VGG (two-step)*. Using the same setting, we also evaluate the performance when Deeplab-v2 (Chen et al. 2018a) with ResNet-101 backbone was used, called *Our-RRM-ResNet (two-step)*. The final results can be found in Table 5. It is observed that among existing methods solely using image-level label without extra data, AffinityNet (Ahn and Kwak 2018) is the most performing one. However, both *Ours-RRM-VGG (two-step)* and *Ours-RRM-ResNet (two-step)* perform much better than it when the same backbone was used. One more thing should be noticed is that AffinityNet (Ahn and Kwak 2018) used ResNet-38 (Wu, Shen, and Van Den Hengel 2019) as baseline, which is more powerful than ResNet-101 (Lee et al. 2019), and even in this case *ours-RRM-ResNet (two-step)* still outperforms it with a big margin. Note that AffinityNet (Ahn and Kwak 2018) applies three different DNNs with many bells and whistles, while we get equivalent results with only one end-to-end network (*Ours-RRM (one-step)*).

To the best of our knowledge, the previous state-of-the-art, FickleNet (Lee et al. 2019), achieves the mIoU score of 64.9 and 65.3 on PASCAL VOC *val* and *test* set, but it uses class agnostic saliency map (Liu et al. 2010) as extra supplement information and uses two individual networks separately. *Ours-RRM-ResNet (two-step)* gives a better performance with mIoU scores of 66.3 and 66.5 on PASCAL VOC *val* and *test* set, which represents 1.4 and 1.2 improvement. Note that we do not use extra data or information in our case. Therefore, *Ours-RRM-ResNet (two-step)* is the new state-of-the-art for two-step image-level label weakly supervised semantic segmentation.

In Figure 4, we report some subjective semantic segmentation results of ours methods, which are compared with EM-Adapt (Papandreou et al. 2015), the state-of-the-art end-to-end network. *Ours-RRM (one-step)* obtains much better segmentation results on both large and small objects, with much accurate boundaries. We also show some results of our two-step approaches, and it can be seen that among our three methods, *ours-RRM-ResNet (two-step)* obtains the best performance duo to the powerful network architecture.

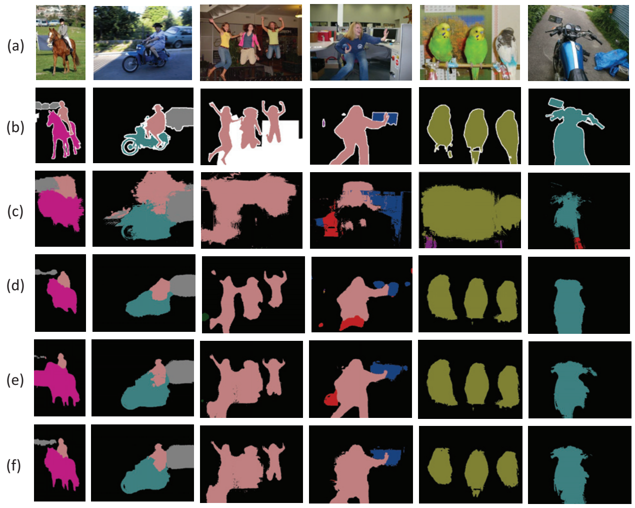

Figure 4: Qualitative segmentation results on PASCAL VOC 2012 *val* set. (a) Original images. (b) Ground-truth. (c) EM-Adapt results. (d) *Ours-RRM (one-step)* results. (e) *Ours-RRM-VGG (two-step)* results. (f) *Ours-RRM-ResNet (two-step)* results.

Conclusion

In this paper, we proposed the *RRM*, an end-to-end network for image-level weakly supervised semantic segmentation. We revisited drawbacks of the state-of-the-arts, which adopt the two-step approach. We proposed a one-step approach through mining reliable yet tiny regions and used them as ground-truth labels directly for segmentation model training. With limited pixels as supervision, we designed a new loss named dense energy loss, which takes shallow features (RGB colors and spatial information) and cooperates with the pixel-wise cross-entropy loss to optimize the training process. Based on our one-step *RRM*, we extend a two-step method. Both our one-step and two-step approaches achieve state-of-the-art performance. More importantly, our *RRM* offers a different perspective from the traditional two-step solutions. We believe that the proposed one-step approach could further boost research in this direction.

Acknowledgment

The work was supported by National Natural Science Foundation of China under 61972323 and 61876155, and Key Program Special Fund in XJTLU under KSF-T-02, KSF-P-02, KSF-A-01, KSF-E-26.

References

Ahn, J., and Kwak, S. 2018. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, 4981–4990.

Bearman, A.; Russakovsky, O.; Ferrari, V.; and Fei-Fei, L. 2016. What's the point: Semantic segmentation with point supervision. In *ECCV*, 549–565. Springer.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2018a. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. 40(4):834–848.

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018b. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 801–818.

Dai, J.; He, K.; and Sun, J. 2015. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *CVPR*, 1635–1643.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *IJCV* 88(2):303–338.

Hong, S.; Yeo, D.; Kwak, S.; Lee, H.; and Han, B. 2017. Weakly supervised semantic segmentation using web-crawled videos. In *CVPR*, 7322–7330.

Hou, Q.; Cheng, M.-M.; Hu, X.; Borji, A.; Tu, Z.; and Torr, P. H. 2017. Deeply supervised salient object detection with short connections. In *CVPR*, 3203–3212.

Hou, Q.; Jiang, P.; Wei, Y.; and Cheng, M.-M. 2018. Self-erasing network for integral object attention. In *NIPS*, 549–559.

Hu, R.; Dollár, P.; He, K.; Darrell, T.; and Girshick, R. 2018. Learning to segment every thing. In *CVPR*, 4233–4241.

Huang, Z.; Wang, X.; Wang, J.; Liu, W.; and Wang, J. 2018. Weakly-supervised semantic segmentation network with deep seeded region growing. In *CVPR*, 7014–7023.

Jiang, H.; Wang, J.; Yuan, Z.; Wu, Y.; Zheng, N.; and Li, S. 2013. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, 2083–2090.

Joy, T.; Desmaison, A.; Ajanthan, T.; Bunel, R.; Salzmann, M.; Kohli, P.; Torr, P. H.; and Kumar, M. P. 2019. Efficient relaxations for dense crfs with sparse higher-order potentials. *SIAM Journal on Imaging Sciences* 12(1):287–318.

Khoreva, A.; Benenson, R.; Hosang, J.; Hein, M.; and Schiele, B. 2017. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 876–885.

Kolesnikov, A., and Lampert, C. H. 2016. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, 695–711. Springer.

Krähenbühl, P., and Koltun, V. 2013. Parameter learning and convergent inference for dense random fields. In *International Conference on Machine Learning*, 513–521.

Lee, J.; Kim, E.; Lee, S.; Lee, J.; and Yoon, S. 2019. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. *arXiv preprint arXiv:1902.10421*.

Lin, D.; Dai, J.; Jia, J.; He, K.; and Sun, J. 2016. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 3159–3167.

Liu, T.; Yuan, Z.; Sun, J.; Wang, J.; Zheng, N.; Tang, X.; and Shum, H.-Y. 2010. Learning to detect a salient object. *IEEE Transactions on PAMI* 33(2):353–367.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440.

Maninis, K.-K.; Caelles, S.; Pont-Tuset, J.; and Van Gool, L. 2018. Deep extreme cut: From extreme points to object segmentation. In *CVPR*, 616–625.

Papandreou, G.; Chen, L.-C.; Murphy, K.; and Yuille, A. 2015. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. *http://arxiv. org/abs/1502* 2734.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch.

Pinheiro, P. O., and Collobert, R. 2015. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 1713–1721.

Qi, X.; Liu, Z.; Shi, J.; Zhao, H.; and Jia, J. 2016. Augmented feedback in semantic segmentation under image level supervision. In *ECCV*, 90–105. Springer.

Rajchl, M.; Lee, M. C.; Oktay, O.; Kamnitsas, K.; Passerat-Palmbach, J.; Bai, W.; Damodaram, M.; Rutherford, M. A.; Hajnal, J. V.; Kainz, B.; et al. 2017. Deepcut: Object segmentation from bounding box annotations using convolutional neural networks. *IEEE Transactions on Medical Imaging* 36(2):674–683.

Song, C.; Huang, Y.; Ouyang, W.; and Wang, L. 2019. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. *arXiv preprint arXiv:1904.11693*.

Tang, M.; Djelouah, A.; Perazzi, F.; Boykov, Y.; and Schroers, C. 2018a. Normalized cut loss for weakly-supervised cnn segmentation. In *CVPR*, 1818–1827.

Tang, M.; Perazzi, F.; Djelouah, A.; Ben Ayed, I.; Schroers, C.; and Boykov, Y. 2018b. On regularized losses for weakly-supervised cnn segmentation. In *ECCV*, 507–522.

Vernaza, P., and Chandraker, M. 2017. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *CVPR*, 7158–7166.

Wang, X.; You, S.; Li, X.; and Ma, H. 2018. Weakly-supervised semantic segmentation by iteratively mining common object features. In *CVPR*, 1354–1362.

Wei, Y.; Feng, J.; Liang, X.; Cheng, M.-M.; Zhao, Y.; and Yan, S. 2017. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*, 1568–1576.

Wei, Y.; Xiao, H.; Shi, H.; Jie, Z.; Feng, J.; and Huang, T. S. 2018. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *CVPR*, 7268–7277.

Wu, Z.; Shen, C.; and Van Den Hengel, A. 2019. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition* 90:119–133.

Xiao, J.; Xie, Y.; Tillo, T.; Huang, K.; Wei, Y.; and Feng, J. 2019. Ian: the individual aggregation network for person search. *Pattern Recognition* 87:332–340.

Xie, Y.; Xiao, J.; Huang, K.; Thiyagalingam, J.; and Zhao, Y. 2018. Correlation filter selection for visual tracking using reinforcement learning. *arXiv preprint arXiv:1811.03196*.

Zhang, J.; Bargal, S. A.; Lin, Z.; Brandt, J.; Shen, X.; and Sclaroff, S. 2018a. Top-down neural attention by excitation backprop. *IJCV* 126(10):1084–1102.

Zhang, X.; Wei, Y.; Feng, J.; Yang, Y.; and Huang, T. S. 2018b. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, 1325–1334.

Zhang, X.; Wei, Y.; Kang, G.; Yang, Y.; and Huang, T. 2018c. Self-produced guidance for weakly-supervised object localization. In *ECCV*, 597–613.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *CVPR*, 2921–2929.