Employing Multi-estimations for Weakly-Supervised Semantic Segmentation

Junsong Fan[1,2], Zhaoxiang Zhang[1,2,3](✉), and Tieniu Tan[1,2,3]

[1] Center for Research on Intelligent Perception and Computing (CRIPAC),
National Laboratory of Pattern Recognition (NLPR), Institute of Automation,
Chinese Academy of Sciences (CASIA), Beijing, China
{fanjunsong2016,zhaoxiang.zhang}@ia.ac.cn, tnt@nlpr.ia.ac.cn

[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
(UCAS), Beijing, China

[3] Center for Excellence in Brain Science and Intelligence Technology,
CAS, Shanghai, China

Abstract. Image-level label based weakly-supervised semantic segmentation (WSSS) aims to adopt image-level labels to train semantic segmentation models, saving vast human labors for costly pixel-level annotations. A typical pipeline for this problem is first to adopt class activation maps (CAM) with image-level labels to generate pseudo-masks (a.k.a. seeds) and then use them for training segmentation models. The main difficulty is that seeds are usually sparse and incomplete. Related works typically try to alleviate this problem by adopting many bells and whistles to enhance the seeds. Instead of struggling to refine a single seed, we propose a novel approach to alleviate the inaccurate seed problem by leveraging the segmentation model's robustness to learn from multiple seeds. We managed to generate many different seeds for each image, which are different estimates of the underlying ground truth. The segmentation model simultaneously exploits these seeds to learn and automatically decides the confidence of each seed. Extensive experiments on Pascal VOC 2012 demonstrate the advantage of this multi-seeds strategy over previous state-of-the-art.

Keywords: Weakly-supervised learning · Semantic segmentation

1 Introduction

Semantic segmentation has achieved rapid progress with deep learning models [3–5,21]. However, these approaches heavily rely on large-scale pixel-level annotations for training, which is very costly to obtain. To reduce the requirement of precise pixel-level annotations for training, researchers proposed weakly-supervised semantic segmentation (WSSS). WSSS adopts only coarse annotations to train the semantic segmentation models, such as scribbles [20,29], bounding boxes [6,28], and image-level class labels [1,2,13,14,17,23,31]. Among them,

© Springer Nature Switzerland AG 2020

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12362, pp. 332–348, 2020.
https://doi.org/10.1007/978-3-030-58520-4_20

the image-level label based WSSS only requires image class labels for training, which are much easier to obtain than other forms of weak annotations. Thus, image label based WSSS got much attention from recent works. In this paper, we focus on the image-level label based WSSS problem.

A common practice to recover targets' spatial information from image-level labels is to adopt the class activation maps (CAM) [39] to generate heat maps for the target objects. These heat maps are utilized to generate pseudo-masks (a.k.a. seeds) to train the desired segmentation models. The CAM is obtained by first training a classification model with the image labels and then applying the last linear classification layer to the feature map columns, which is right before the global average pooling layer. Because the CAM is trained for classification, the highlighted regions are usually only the most discriminative ones. Therefore, only sparse and incomplete seeds can be obtained, and the subsequently trained segmentation models can only predict partial objects.

To alleviate the incomplete seed problem of CAM, researchers adopt multiple dilated convolutions [33], iterative erasing strategy [31], random drop connections [17], region growing algorithms [13], online accumulating activation maps [14], and many other strategies [26] to generate more complete seeds. Though these approaches have achieved significant progress, they usually rely on carefully designed rules and experience-based hyper-parameters to balance the seed's precision and recall, which is hard to generalize.

Instead of struggling for generating a single "perfect" seed for each image by manually designed rules, we propose a novel principled way to employ multiple different seeds simultaneously to train the segmentation models. The different seeds for each image can be seen as the estimates of the common underlying ground truth. We leverage the robustness of the segmentation models to mine useful information from these different seeds automatically. The reasons this strategy works are threefold.

Firstly, different seeds help to reduce the influence of wrong labels. It is generally reasonable to assume that the probability of a pixel obtaining a correct pseudo-label is larger than obtaining a wrong label. Pixels assigned with the same label by all the different seeds are more likely to be right. The contributions of these pixels are not affected, because all the different seeds provide the same label. Meanwhile, pixels with different seed labels provide gradients in different directions; thus, the different gradients can be canceled out to some extent, reducing the risk of optimizing in the wrong direction. *Secondly*, complementary parts may exist in different seeds, making the pseudo-labels more complete as a whole. For example, the mask of a person's body may be absent in one seed, but present in another different seed. *Thirdly*, the segmentation model is robust to noise to some extent. Take the pilot experiments in Table 1 as an example, with 30% of the foreground pixels replaced by noise in the training set, the segmentation model can still achieve about 90% of the performance compared with training with the ground truth. This result may because segmentation models can leverage the knowledge from the whole dataset, thus reducing the

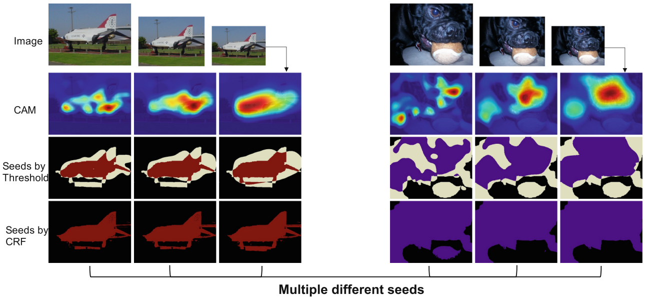

Fig. 1. Examples of the CAM and the generated seeds. The three columns of each group correspond to three different CAM scales. The first row is the image. The second row is the CAM from the VGG16 backbone, and the last two rows show two types of the generated seeds. Our approach simultaneously adopts these different seeds to train the segmentation models.

influence of unsystematic noise. This property helps to mine useful information from multiple seeds.

To further enhance the training process's robustness, we propose a weighted selective training (WST) strategy, which adaptively adjusts the weights among different seeds for each pixel. Compared with previous approaches [14,17,33] that merge multiple CAMs by hand-crafted rules, e.g., average or max fusion, our method can leverage the segmentation model's knowledge to assign weights among different seeds dynamically. We conduct thorough experiments to demonstrate the effectiveness of the proposed approach. On Pascal VOC 2012 dataset, we achieve new state-of-the-art performance with mIoU 67.2% and 66.7% on the validation and the test set, respectively, demonstrating the advantage of our approach. In summary, the main contributions of this paper are as following:

- We propose a new principled approach to alleviate the inaccurate seed problem for WSSS, which simultaneously employs many different seeds to train the segmentation models. A weighted selective training strategy is proposed to mitigate the influence of noise further.
- We conduct thorough experiments to demonstrate the approach's effectiveness and reveal the influence of different kinds of seeds.
- The proposed approach significantly outperforms the single-seed baseline and achieves new state-of-the-art performance on Pascal VOC 2012 dataset with only image-level labels for training.

2 Related Work

2.1 Semantic Segmentation

Recently deep learning based approaches [3–5,21,38] have dominated the semantic segmentation community. These approaches usually adopt fully convolutional layers and take the semantic segmentation task as a per-pixel classification task. Though these approaches have achieved great progress, they need pixel-level annotations for training, which cost vast human labors to obtain.

2.2 Weakly-Supervised Semantic Segmentation

Weakly-supervised semantic segmentation (WSSS) is proposed to alleviate the annotation burden of segmentation tasks. According to the types of annotations, WSSS approaches can be classified as bounding box based [6,28], scribble based [20,29], and image-level label based [13,14,17,31,33] approaches. In this paper, we focus on the image-level label based WSSS.

Most of the present image-level label based WSSS approaches adopt a two-stage training strategy. It firstly estimates the pseudo-masks (a.k.a. seeds) of target objects from image-level labels and then takes these seeds to train a regular semantic segmentation model. Because of the lack of supervision, the seeds are often incomplete. To alleviate this problem, AE-PSL [31] proposes an iterative erasing strategy that iteratively erases already obtained pseudo-masks in the raw image and re-estimate new regions. MDC [33] proposes to adopt multiple layers with different dilation rates to expand the activated regions. DSRG [13] proposes a seed region growing algorithm to expand the initial seeds gradually. FickleNet [17] uses random connections to generate many different activation maps and assemble them together. OAA [14] accumulates the activation maps along the process of training the CAM to obtain more complete estimates. These approaches apply various hand-crafted rules and carefully adjusted hyperparameters to generate a single seed for each image. However, it is generally hard to balance the recall and the precision for the underlying target objects. In contrast, we propose to simultaneously adopt many different seeds to train the semantic segmentation models and leverage the segmentation models' robustness to extract useful information from these seeds automatically.

2.3 Learning from Noisy Labels

Some related works also adopt multiple noisy labels to learn [19,36,37]. These approaches rely on noise distribution assumptions that may not hold in the WSSS problem, adopt complicated rules to pre-fuse the labels, or train additional modules to merge them. In contrast, our approach is more computation efficient and can exploit the pseudo-labels dynamically.

3 Pilot Experiments

Before illustrating the detailed approaches, we first conduct pilot experiments to demonstrate that the segmentation model benefits from multiple sets of labels that contain noise. To this end, we manually add noise to the ground truth labels by randomly set partial foreground blocks as background, as shown in Fig. 2. Then we adopt these noisy labels to train the segmentation model. We

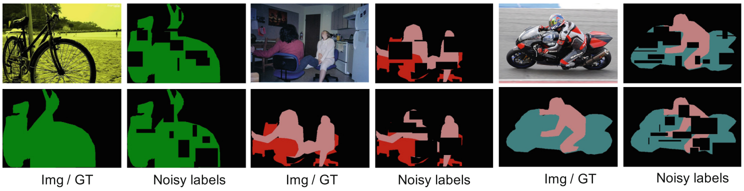

Fig. 2. The hand-crafted noisy labels. Blocks with random sizes are put on the foreground objects. Two different labels with the same noise ratio are shown in the columns.

Table 1. Results on the Pascal VOC 2012 val set by the VGG16-Largefov model. The training data is from the training set with different ratios of hand-crafted noise.

Noise ratio	mIoU (%)	
	One label	Two labels
0 (ground truth)	72.1	-
0.1	70.6	71.0
0.2	68.1	69.0
0.3	64.8	66.2

compare results obtained by only utilizing a single noise label and utilizing two different noisy labels. The results are shown in Table 1. When adopting two sets of different noisy labels, the segmentation model consistently outperforms the single label counterparts under various noise rates.

The intuitive reason for the improvement is because there exists complementary information between the two sets of labels. We discuss a simplified two-class case as an example for illustration. Assume the noise is evenly distributed among all the pixels in the dataset, and the probability of noise is r. When only a single label is available, the signal-to-noise rate is $(1 - r)/r$. When there are two sets of labels, the probability of a pixel obtains two true or two false labels are $(1 - r)^2$ and r^2, respectively. The remaining $2r(1 - r)$ of the pixels receive two contradictory labels, thus do not contribute to the gradients. In this situation, the signal-to-noise rate becomes $(1 - r)^2/r^2$. Generally, r is less than $(1 - r)$, thus simultaneously adopting two different labels helps to reduce the proportion of gradients from wrong noise labels. In other words, those pixels with confusable labels are depreciated. Similar conclusions can be easily generalized to the situation of multiple classes and more sets of different labels.

In the setting of WSSS, seeds are estimated from image-level labels to approach the unknown pixel-level ground truth, and utilized to train the segmentation models. Although different seeds generally are not independent of each other and the noise is not evenly distributed, our experiments empirically demonstrate that there is still some complementary information available from multiple seeds.

Thus, adopting multi-seeds can help the segmentation model recognize more robust estimates and improve the training, as shown in Sect. 5.

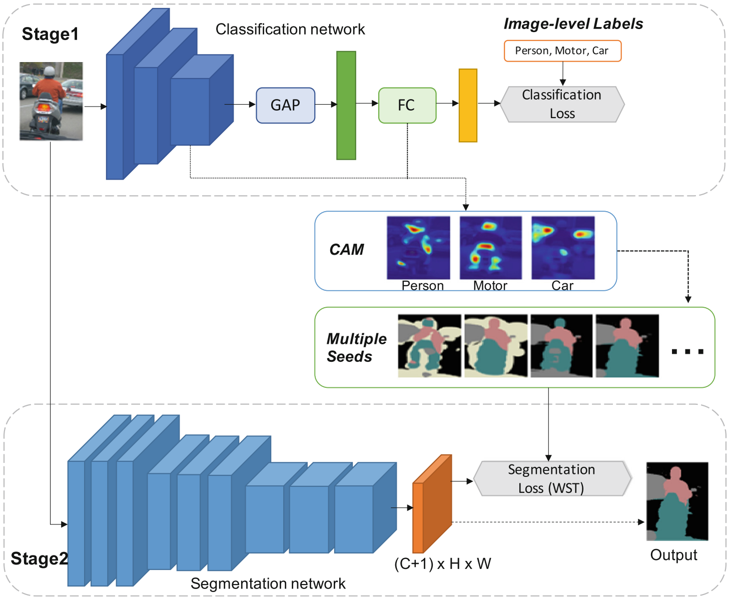

Fig. 3. The framework of our approach. The pipeline for WSSS contains two stages. In the first stage, we train the CAM by the image-level labels and generate multiple seeds via different approaches. In the second stage, we adopt these seeds simultaneously to train the segmentation model. Finally, the segmentation model outputs the semantic segmentation predictions for evaluation.

4 Approach

The whole framework of our approach contains two stages, as shown in Fig. 3. The first stage generates many different seeds from the CAM, and the second stage utilizes all of these seeds to train segmentation models. After training, segmentation results are obtained by inferring the segmentation models.

4.1 The Class Activation Map

The class activation map (CAM) [39] is widely adopted to generate initial estimates (seeds) for WSSS. The first step is to adopt the image-level labels to train a classification network, which contains a global average pooling layer right before the final classification layer. The training loss is simply the multi-class sigmoid loss:

$$L_{cam}(X) = -\sum_{c=1}^{C} y_c \log \sigma(p_c) + (1 - y_c) \log[1 - \sigma(p_c)], \tag{1}$$

Where, X is the input image, p_c is the model's prediction for the c-th class, $\sigma(\cdot)$ is the sigmoid function, C is the total number of foreground classes. y_c is the image-level label for the c-th class, whose value is 1 if the class present in the image else 0.

After training, the global average pooling layer is removed, and the final linear classification layer is directly applied to each column of the last feature map to derive the CAM:

$$M_{i,j}^{c} = \mathbf{f}_{i,j}^{T} \mathbf{w}^{c}, \qquad \{c \in C_{fg}\} \tag{2}$$

Where, $\mathbf{w}^c$ is the weight vector for the c-th class in the classification layer, $\mathbf{f}_{i,j}$ is the feature vector in the feature map at spatial location $\{i, j\}$. $M_{i,j}^{c}$ is the corresponding value of CAM of the c-th class at location $\{i, j\}$. C_{fg} is the set of foreground classes present in the image. For those classes that are not present in the image, the corresponding maps are directly set to zero.

Before generating the seeds, the CAM is normalized by filtering out negative values and dividing the spatial maximum:

$$\tilde{M}_{i,j}^{c} = \frac{[M_{i,j}^{c}]_{+}}{\max\limits_{i \in 1 \sim H, j \in 1 \sim W} \{[M_{i,j}^{c}]_{+}\}}, \qquad \{c \in C_{fg}\} \tag{3}$$

Where, operator $[\cdot]_{+}$ sets the negative values to 0. H and W are the height and width of the CAM, respectively. The obtained CAM $\tilde{M} \in \mathbb{R}^{C \times H \times W}$ is then bilinearly interpolated to the original image size and utilized to generate the seeds.

4.2 Multi-type Seeds

A common practice to generate seeds is to use the CAM and a hard threshold to estimate foreground regions and adopt saliency models to estimate background regions. CRF is also widely adopted to refine the estimate. In this paper, we adopt two different approaches to generate two types of seeds.

The first approach simply adopts the threshold method to generate seeds. We take pixels with normalized CAM scores large than 0.1 as the foreground. We adopt the same saliency model [11] utilized by previous approaches [14] to estimate the background. Pixels with saliency scores less than 0.06 are taken as background, which follows the same setting in previous approaches. The remaining unassigned pixels and those pixels with conflict assignments are marked as unknown and will be ignored when training the segmentation models.

The second approach concatenates the background scores from the saliency and the foreground scores from the normalized CAM, then adopts the CRF algorithm [16] to refine the scores. Finally, the seeds are obtained by selecting the class with the largest score for each pixel. With this strategy, every pixel is assigned with a valid pseudo-label, and there are no conflicts and unknowns. Examples of the generated seeds are shown in Fig. 1.

Although the seeds generated by the CRF approach have more details, it may raise additional false positives and false negatives compared with the threshold approach. Therefore, there may exist complementary information in these two types of seeds. As demonstrated in Table 2, even though CRF based seeds perform better than threshold based seeds, there is still a considerable boost by simultaneously adopting both of them for training.

4.3 Multi-scale Seeds

Scale plays an important role in deep convolutional network based approaches. Because the receptive field is fixed, the convolution kernels face quite different input patterns with different scales. The network is forced to handle different scales simultaneously; thus, it generally needs different parameters for different scales. As a result, different patterns may be highlighted by the network in different input scales, and further derives different seeds. To utilize this property, we resize the input image size with different scales and feed them into the network to obtain CAMs under different scales. We adopt these CAMs with different scales to generate multiple seeds.

4.4 Multi-architecture Seeds

Generally, different architectures of the backbone do not produce identically the same predictions. Thus, different architectures can potentially provide different estimates for the underlying ground truth masks. VGG16 [27] and ResNet38 [34] are two widely adopted networks for generating seeds in the WSSS community. We adopt these two different architectures to generate different seeds.

4.5 The Weighted Selective Training

A plain way to adopt different seeds to train the segmentation model is to compute the per-pixel cross-entropy loss with each seed and adopts the average:

$$L_{plain}(X) = \frac{1}{N_k HW} \sum_{k} \sum_{i,j} \sum_{c} y_{i,j,c}^{(k)} \log p_{i,j,c}, \tag{4}$$

Where, X is the input image, $y_{i,j,c}^{(k)}$ is the label for pixel $\{i, j\}$ from the k-th seed, which equals 1 if the label belongs to the c-th class or else 0. $p_{i,j,c}$ is the prediction of the segmentation model at location $\{i, j\}$ for the c-th class, which is normalized by the softmax operator. N_k is the total number of the seeds for the given image. H and W are the height and width of the feature, respectively.

Because of the robustness of the segmentation model and the effect of increasing the signal-to-noise rate by multiple seeds, directly adopting L_{plain} is able to boost the performance over single-seed baselines. To further improve the robustness over noise labels, we propose to utilize the segmentation model's online predictions to weight different seeds. The training loss becomes:

$$L_{wst}(X) = \frac{1}{HW} \sum_{k} \sum_{i,j} w_{i,j,k} \sum_{c} y_{i,j,c}^{(k)} \log p_{i,j,c}, \tag{5}$$

Where, $w_{i,j,k} \in [0,1]$ is the weight for the k-th seed label at location $\{i,j\}$, which is computed by comparing the label with the segmentation model's online prediction:

$$w_{i,j,k} = \exp(s \cdot \tilde{w}_{i,j,k}) / \sum_{k} \exp(s \cdot \tilde{w}_{i,j,k}), \tag{6}$$

$$\tilde{w}_{i,j,k} = \mathbb{I}[\arg\max_{c}(p_{i,j,c}) = \arg\max_{c}(y_{i,j,c})], \tag{7}$$

That is, we take the value as 1 if and only if the segmentation model's prediction matches the pseudo-label, then we adopt the softmax operator to normalize all the values across different seeds to ensure that $\sum_k w_{i,j,k} = 1$. s is a scale factor to control the sharpness of the weight. When s equals 0, the loss is identical to the plain training loss L_{plain}. In practice, we simply set s equals 1. Along the training process, the segmentation model converges and predicts more stable results. As a result, outliers that contradict the prediction will be inhibited, further reducing the influence of the noise.

5 Experiments

5.1 Dataset

Following previous works, we adopt the Pascal VOC 2012 dataset [8] to evaluate our approach. It contains 20 foreground classes and a background class for semantic segmentation. The extended training set [10] contains 10582 images, the validation set contains 1449 images, and the test set contains 1456 images. For training our weakly-supervised models, only the image-level labels are used, i.e., the image class labels of the 20 foreground object classes. The performance is evaluated by the standard mean intersection over union (mIoU) with all the 21 classes.

5.2 Implementation Details

We adopt two popular backbones to generate the seeds, i.e., VGG16 [27] and ResNet38 [34], which are widely adopted in the WSSS community. To obtain larger receptive fields for the details of the objects, we follow the DeepLab's setting to set the last two downsampling layers' strides to 1 and adopt dilated convolutions in the following layers. The total downsampling rate of the feature map is 8. Backbones are pre-trained by the ImageNet classification task [7], New layers are initialized by Normal distribution with a standard deviation 0.01. The initial learning rate is 0.001 and is poly decayed with power 0.9 every epoch. The learning rate for newly initialized layers is multiplied by 10. We adopt the SGD optimizer and train 20 epochs with the batch size 16. The input images for training are randomly scaled between 0.5 and 1.5, randomly mirrored horizontally with a probability 0.5, and randomly cropped into size 321. After obtaining the seeds, we adopt the proposed approach and follow the standard hyper-parameters to train the DeepLab-v2 segmentation models.

Table 2. Ablation study of the multi-type seeds. Results are obtained by the VGG16-Largefov segmentation model and evaluated on the VOC 12 val set using mIoU (%). The numbers in the parenthesis represent the number of seeds adopted.

CAM-Backbone	Seed-Type		
	Threshold (T.)	CRF (C.)	Both (T.+C.)
VGG16	61.4 (1)	62.9 (1)	**63.8** (2)
RES38	62.3 (1)	62.6 (1)	**63.5** (2)

Table 3. Ablation study of the multi-scale seeds. Results are obtained by the VGG16-Largefov segmentation model and evaluated on the VOC 12 val set using mIoU (%). The numbers in the parenthesis represent the number of seeds adopted.

Seed-Type	Single-Scale	Multi-Scale	Merge-Scale	
			Max	Avg
Threshold (T.)	61.4 (1)	**62.5** (3)	60.2 (1)	61.1 (1)
CRF (C.)	62.9 (1)	**63.5** (3)	62.5 (1)	62.8 (1)
Both (T.+C.)	63.8 (2)	**64.0** (6)	63.7 (2)	63.9 (2)

5.3 The Influence of Multiple Seeds

Multi-type Seeds. We firstly demonstrate that adopting the multi-type seeds helps to train the segmentation network. We generate seeds by both the *threshold* based approach and the *CRF* based approach, as described in Sect. 4.2. The baseline results are obtained by training on these two types of seeds separately, and our approach takes both of them for training. The results are summarized in Table 2. With only a single type of seeds, the best result is achieved by using the CRF based seeds, showing that CRF provides more details based on the low-level RGB cues. However, additional wrong labels may also be incurred by the CRF. Thus by simultaneously adopting these two types of seeds with our approach, there is a further 0.9% improvement, demonstrating that our method can mine useful complementary information from multi-type seeds.

Table 4. Ablation study of the multi-architecture seeds. Results are obtained by the VGG16-Largefov segmentation model and evaluated on the VOC 12 val set using mIoU (%). The numbers in the parenthesis represent the number of seeds adopted.

Seed-Type	CAM-Backbone		
	VGG16	RES38	VGG16+RES38
Threshold (T.)	61.4 (1)	62.3 (1)	**63.5** (2)
CRF (C.)	62.9 (1)	62.6 (1)	**63.1** (2)
Both (T.+C.)	63.8 (2)	63.5 (2)	**64.2** (4)

Table 5. Ablation study of the weighted selective training (WST). Results are obtained by the VGG16-Largefov segmentation model and evaluated on the VOC 12 val set using mIoU (%).

Multi-Type	Multi-Scale	Multi-Architecture	w/o WST	w/ WST
✓			**63.9**	63.8
✓	✓		63.8	**64.0**
✓	✓	✓	63.9	**64.6**

Multi-scale Seeds. To verify the effectiveness of exploiting multi-scale seeds, we generate seeds by inferring CAMs with three different scales, i.e., 1, 0.75, and 0.5. Examples of the multi-scale CAMs and corresponding seeds are shown in Fig. 1. As shown in Table 3, employing multiple seeds of different scales consistently provides improvement over single-scale counterparts. It is also noteworthy that adopting both multi-type seeds and multi-scale seeds simultaneously further improves the performance, demonstrating the effectiveness of our approach. To more concretely demonstrate the advantage of utilizing multiple seeds over single seeds, we also generate a single set of seeds by merging the multi-scale CAMs. Specifically, we merge the CAMs from all the three scales by the max- or the average-fusion. We generate seeds from the merged CAM and adopt them for training. The last two columns in Table 3 shows the results. If we only adopt the merged single-type seeds for training, there is no obvious improvement over the baseline. It may because simply merging multi-scale CAMs introduces some ambiguity and additional noise. In contrast, our approach generates seeds from different CAMs independently and utilizes these different seeds for training, which is more robust to leverage the multi-scale information.

Multi-architecture Seeds. Because of the difference in network depth, receptive field, and connection structures, different networks usually produce different activation maps for the same input. To leverage this character, we adopt different backbones to generate the seeds. In previous works, either VGG16 or ResNet38 is adopted to generate the seeds. Thus we choose these two networks to conduct experiments of the multi-architecture seeds. Results in Table 4 shows that taking seeds from these two architectures always improves the performance, demonstrating that seeds from different networks can also provide complementary information to the segmentation models.

5.4 The Weighted Selective Training

We conduct ablation studies to demonstrate the effectiveness of the proposed weighted selective training (WST) strategy, as shown in Table 5. The results show that the influence of the WST approach is more apparent when there are more different kinds of seeds. When only adopting the multi-type seeds, there is no noticeable improvement. It may because many ambiguous pixels are set to empty in the threshold-based seeds, which reduces the number of conflict noise labels between the two types of seeds. It is also noteworthy that even without the WST approach, adopting multiple seeds for training improves over the baseline with a clear margin, demonstrating that the segmentation model can effectively learn from multiple seeds, even with noise. Figure 4 is the visualization of the weights among different seeds. It shows that the assigned per-pixel weights can generally inhibit noisy labels.

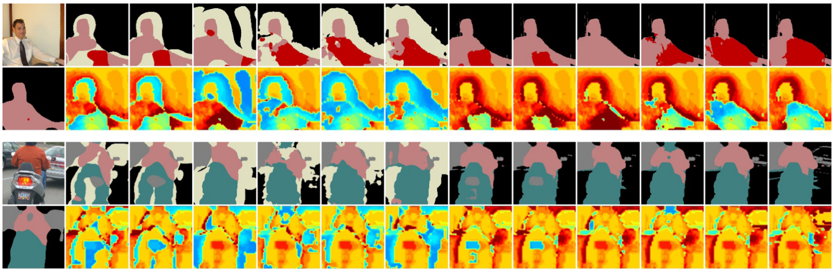

Fig. 4. Visualization of the WST weights among different seeds. The first column shows the input images and online predictions. The rest columns show the seeds and the corresponding weights obtained by the WST approach.

5.5 Comparison with Related Works

We employ all the above approaches to generate many different seeds to train our model to compare with related works. Specifically, two types, three scales, and two architectures are adopted, resulting in 12 different seeds. The VGG16-LargeFov and the ResNet101-LargeFov are two widely used segmentation models for evaluating the WSSS approaches. We report results for both of them, as shown in Table 6 and Table 7, respectively. To the best of our knowledge, previous best results on the VGG16 backbone are achieved by AISI [9] and OAA [14]. Our approach significantly outperforms them by 1.5% and 1.1% mIoU scores on the validation set and the test set, respectively. It is noteworthy that ablation results

Table 6. Comparison with related works. All the results are obtained by the VGG16 backbone and evaluated on the VOC 12 semantic segmentation benchmark.

Method	Publication	mIoU (%)	
		val	test
CCNN [23]	ICCV15	35.3	35.6
EM-Adapt [22]	ICCV15	38.2	39.6
MIL [24]	CVPR15	42.0	40.6
SEC [15]	ECCV16	50.7	51.7
AugFeed [25]	ECCV16	54.3	55.5
STC [32]	PAMI17	49.8	51.2
AE-PSL [31]	CVPR17	55.0	55.7
DCSP [2]	BMVC17	58.6	59.2
AffinityNet [1]	CVPR18	58.4	60.5
GAIN [18]	CVPR18	55.3	56.8
MCOF [30]	CVPR18	56.2	57.6
DSRG [13]	CVPR18	59.0	60.4
MDC [33]	CVPR18	60.4	60.8
AISI [9]	ECCV18	61.9	63.1
SeeNet [12]	NIPS18	61.1	60.7
FickleNet [17]	CVPR19	61.2	61.9
SSNet [35]	ICCV19	57.1	58.6
OAA [14]	ICCV19	63.1	62.8
Ours		**64.6**	**64.2**[a]

[a] http://host.robots.ox.ac.uk:8080/anonymous/QR5OFW.html

Table 7. Comparison with related works. All the results are obtained by the ResNet101 backbone and evaluated on the VOC 12 semantic segmentation benchmark.

Method	Publication	mIoU (%)	
		val	test
DCSP [2]	BMVC17	60.8	61.9
MCOF [30]	CVPR18	60.3	61.2
DSRG [13]	CVPR18	61.4	63.2
AISI [9]	ECCV18	64.5	65.6
SeeNet [12]	NIPS18	63.1	62.8
FickleNet [17]	CVPR19	64.9	65.3
OAA [14]	ICCV19	65.2	66.4
Ours		**67.2**	**66.7**[b]

[b] http://host.robots.ox.ac.uk:8080/anonymous/ZHCI9F.html

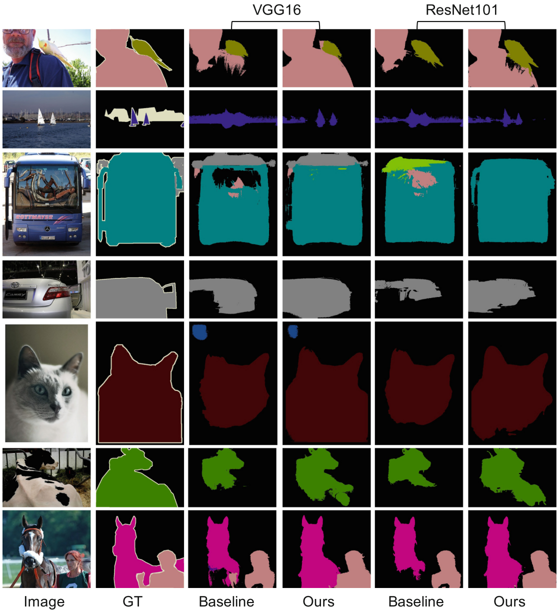

Fig. 5. The prediction results of single-seed baseline and our approach on VOC 2012 val set. The first two columns are images and ground truth (unavailable for training). The third and the fourth columns are obtained by the VGG16 based segmentation model. The last two columns are obtained by the ResNet101 based segmentation model.

in Table 3 reveal that even with only VGG16-CAM based seeds, our approach could achieve mIoU 64.0% on the validation set, which outperforms previous best results by 0.9%, demonstrating the advantage of adopting multiple different seeds. The ResNet101 based segmentation model generally performs better than the VGG16. Our approach also works with this stronger segmentation model, which outperforms previous best results by 2.0% and 0.3% on the validation and the test set, respectively, demonstrating our approach's generalization ability. Figure 5 shows the visualization results of the segmentation models' predictions. Compared with the single seed baseline, our approach generally obtains more complete and robust predictions.

6 Conclusions

Image-level label based weakly-supervised semantic segmentation suffers from incomplete seeds for training. To alleviate this problem, we propose a novel approach to employing multiple different seeds simultaneously to train the segmentation models. We propose a weighted selective training strategy to reduce further the influence of noise in the multiple seeds. Extensive experiments demonstrate that our training framework can effectively mine reliable and complementary information from a group of different seeds. Our approach significantly improves over the baseline and achieves new state-of-the-art performance on the Pascal VOC 2012 semantic segmentation benchmark with only image-level labels for training.

Acknowledgement. This work was supported in part by the National Key R&D Program of China (No. 2018YFB1402605), the National Natural Science Foundation of China (No. 61836014, No. 61761146004, No. 61773375).

References

1. Ahn, J., Kwak, S.: Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. arXiv preprint arXiv:1803.10464 (2018)
2. Chaudhry, A., Dokania, P.K., Torr, P.H.: Discovering class-specific pixels for weakly-supervised semantic segmentation. arXiv preprint arXiv:1707.05821 (2017)
3. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv preprint arXiv:1412.7062 (2014)
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2018)
5. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
6. Dai, J., He, K., Sun, J.: BoxSup: exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1635–1643 (2015)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 248–255. IEEE (2009)
8. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The Pascal visual object classes (VOC) challenge. Int. J. Comput. Vision **88**(2), 303–338 (2010). https://doi.org/10.1007/s11263-009-0275-4
9. Fan, R., Hou, Q., Cheng, M.M., Yu, G., Martin, R.R., Hu, S.M.: Associating inter-image salient instances for weakly supervised semantic segmentation (2018)
10. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors (2011)
11. Hou, Q., Cheng, M.M., Hu, X., Borji, A., Tu, Z., Torr, P.H.: Deeply supervised salient object detection with short connections. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3203–3212 (2017)
12. Hou, Q., Jiang, P.T., Wei, Y., Cheng, M.M.: Self-erasing network for integral object attention. arXiv preprint arXiv:1810.09821 (2018)
13. Huang, Z., Wang, X., Wang, J., Liu, W., Wang, J.: Weakly-supervised semantic segmentation network with deep seeded region growing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7014–7023 (2018)
14. Jiang, P.T., Hou, Q., Cao, Y., Cheng, M.M., Wei, Y., Xiong, H.K.: Integral object mining via online attention accumulation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2070–2079 (2019)

15. Kolesnikov, A., Lampert, C.H.: Seed, expand and constrain: three principles for weakly-supervised image segmentation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 695–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_42
16. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected CRFs with Gaussian edge potentials. In: Advances in Neural Information Processing Systems, pp. 109–117 (2011)
17. Lee, J., Kim, E., Lee, S., Lee, J., Yoon, S.: FickleNet: weakly and semi-supervised semantic image segmentation using stochastic inference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5267–5276 (2019)
18. Li, K., Wu, Z., Peng, K.C., Ernst, J., Fu, Y.: Tell me where to look: guided attention inference network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9215–9223 (2018)
19. Li, S., et al.: Coupled-view deep classifier learning from multiple noisy annotators. In: AAAI, pp. 4667–4674 (2020)
20. Lin, D., Dai, J., Jia, J., He, K., Sun, J.: ScribbleSup: scribble-supervised convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3159–3167 (2016)
21. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
22. Papandreou, G., Chen, L.C., Murphy, K., Yuille, A.L.: Weakly- and semi-supervised learning of a DCNN for semantic image segmentation. arXiv preprint arXiv:1502.02734 (2015)
23. Pathak, D., Krähenbühl, P., Darrell, T.: Constrained convolutional neural networks for weakly supervised segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1796–1804 (2015)
24. Pinheiro, P.O., Collobert, R.: From image-level to pixel-level labeling with convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1713–1721 (2015)
25. Qi, X., Liu, Z., Shi, J., Zhao, H., Jia, J.: Augmented feedback in semantic segmentation under image level supervision. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 90–105. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_6
26. Shimoda, W., Yanai, K.: Self-supervised difference detection for weakly-supervised semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5208–5217 (2019)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
28. Song, C., Huang, Y., Ouyang, W., Wang, L.: Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3136–3145 (2019)
29. Vernaza, P., Chandraker, M.: Learning random-walk label propagation for weakly-supervised semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 3, p. 3 (2017)
30. Wang, X., You, S., Li, X., Ma, H.: Weakly-supervised semantic segmentation by iteratively mining common object features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1354–1362 (2018)

31. Wei, Y., Feng, J., Liang, X., Cheng, M.M., Zhao, Y., Yan, S.: Object region mining with adversarial erasing: a simple classification to semantic segmentation approach. In: IEEE CVPR, vol. 1, p. 3 (2017)
32. Wei, Y., et al.: STC: a simple to complex framework for weakly-supervised semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(11), 2314–2320 (2017)
33. Wei, Y., Xiao, H., Shi, H., Jie, Z., Feng, J., Huang, T.S.: Revisiting dilated convolution: a simple approach for weakly- and semi-supervised semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7268–7277 (2018)
34. Wu, Z., Shen, C., Van Den Hengel, A.: Wider or deeper: revisiting the ResNet model for visual recognition. Pattern Recogn. **90**, 119–133 (2019)
35. Zeng, Y., Zhuge, Y., Lu, H., Zhang, L.: Joint learning of saliency detection and weakly supervised semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 7223–7233 (2019)
36. Zhang, D., Han, J., Zhang, Y.: Supervision by fusion: towards unsupervised learning of deep salient object detector. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4048–4056 (2017)
37. Zhang, J., Zhang, T., Dai, Y., Harandi, M., Hartley, R.: Deep unsupervised saliency detection: a multiple noisy labeling perspective. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9029–9038 (2018)
38. Zheng, S., et al.: Conditional random fields as recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1529–1537 (2015)
39. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929 (2016)