Self-supervised Difference Detection for Refinement CRF and Seed Interpolation

Wataru Shimoda Keiji Yanai
Department of Informatics, The University of Electro-Communications, Tokyo

Abstract

To minimize annotation costs associated with training of semantic segmentation models, weakly-supervised segmentation approaches have been extensively studied. In this paper, we propose a novel method: Self-Supervised Difference Detection (SSDD) module which evaluates confidence of each of the pixels of segmentation masks and integrate highly confident pixels of two candidate masks.

1. Introduction

In this paper, we focus on avoiding performance degradation by CRF. In the proposed method, we estimate confidence maps on segmentation masks before and after applying CRF, and integrate them into a final mask by selecting more confident pixels comparing both confident maps. To estimate confidence maps of segmentation masks, we propose a Self-Supervised Difference Detection (SSDD) module. SSDD contains a Difference Detection network (DD-net), which estimates differences of the regions Since the difference of segmentation masks can be obtained automatically without any human supervision, we can train a DD-net without any additional information. Thus, training of DD-net can be considered as self-supervised learning.

It is hard to estimate complicate parts even if an input region mask contains the regions to be estimated, while it is easy to estimate simple parts. Then we regard an output of a DD-net as confidence maps of difference between the masks before or after a refinement process. By using the confidence maps on mask difference, we pick up the pixels having higher confidence from both the masks, and generate re-refined segmentation masks.

In this paper, we demonstrate that the propose SSDD module can be used in both the seed generation stage and the training stage of fully-supervised segmentation as a "re-refinement module". In the seed generation stage, we refine the CRF results of PSA [1] by a SSDD module. In the training stage, we introduce two SSDD modules inside training loop of a fully-supervised segmentation network. In the experiments, we demonstrate the effectiveness of SSDD modules in both stages. Especially, SSDD modules boosted the performance of weakly-supervised semantic segmentation on the PASCAL VOC 2012 dataset greatly and, achieved new state-of-the-art.

2. Related Works

Region refinement for WSS results using CRF CRF [11] can refine the ambiguous outlines using low-level features such as a color of pixels. Chen et al. [13] and Pathak et al. [14] adopted CRF as a post-processing method for region refinement and demonstrated the effectiveness of the CRF for WSS. Kolesnikov et al. [10] proposed to use CRF during training of a semantic segmentation model. Ahn et al. [1] proposed a method to learn pixel-level similarity from CRF results, and apply random walk based region refinement, which achieved the state-of-the-art on the Pascal VOC 2012 dataset. We focus on preventing a segmentation mask from being degraded by applying CRF.

Generating pixel-level labels during training of a fully-supervised semantic segmentation (FSS) model Constrained Convolutional Neural Network (CCNN) [14] and EM-adopt [13] generated pixel-level labels during training using class labels and outputs of the segmentation model. Wei et al. [25] proposed an online Prohibitive Segmentation Learning (PSL). that utilizes classification score and the output of the segmentation model for generating mask during training. Huang et al. [7] proposed Deep Seeded Region Growing (DSRG), which is a method to expand the seed region during training. The authors prepared pixel-level seed labels before training that have unlabeled regions for unconsidered pixels. In this work, we proposed new constraint for the generating pixel-level labels during training of a FSS model.

3. Method for "Re-refinement" of Refinement

In this paper, we focus on avoiding performance degradation by refinement methods of segmentation masks such as CRF. To do that, we propose a Self-Supervised Difference Detection (SSDD) module. Fig.1 shows the basic idea of the processing in SSDD.

Difference Detection Network We denote one pair of segmentation masks as $(m^{\text{before}}, m^{\text{after}})$. Basically, we consider m^{before} is a raw segmentation mask and m^{after} is a processed mask. We denote their difference regions by a binary mask $M^{\text{before,after}}$. The difference regions defined by:

$$M_u^{\text{before,after}} = \begin{cases} 1 & \text{if } (m_u^{\text{before}} = m_u^{\text{after}}) \\ 0 & \text{if } (m_u^{\text{before}} \neq m_u^{\text{after}}) \end{cases}, \quad (1)$$

where $u \in \{1, 2, .., n\}$ indicates a location of pixels, and

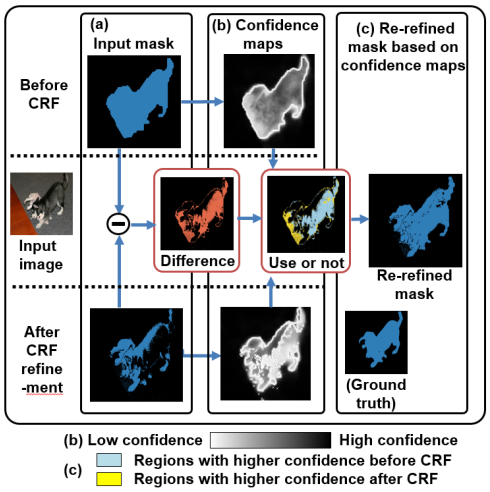

Figure 1. Basic idea of "re-refinement" by Self-Supervised Difference Detection. (a) Segmentation masks before/after applying CRF. (b) Confidence maps obtained by DD-Net. (c) Re-refined mask which is integration of both masks based on confidence maps.

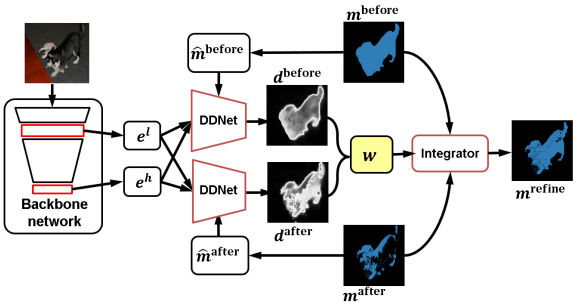

Figure 2. Self-Supervised Difference Detection (SSDD) module.

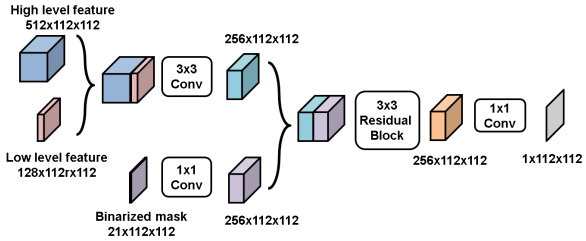

Figure 3. Difference Detection Network (DD-Net).

n is the number of pixels. We use high level features $e^h(x;\theta_e)$ and low level features $e^l(x;\theta_e)$ x is an input image and e is an embedding function parameterized by θ_e. As shown in Fig.2, the confidence map of the input mask, d, is generated by a Difference Detection network, $\text{DDnet}(e^h(x;\theta_e), e^l(x;\theta_e), \hat{m};\theta_d), d \in \mathbb{R}^{H\times W}$, where $\hat{m}$ is a binarized mask with the same number of channels to the target class number. The architecture of DD-Net is shown in Fig.3.

Self-Supervised Difference Detection Module In this section, we describe the detail of a SSDD module shown in Fig.2. We suppose that an input mask which is difficult to predict difference regions is relatively reliable. This insight is based on an assumption that the overall processed masks, m^{after}, have enough correct labels against error labels in different regions in the training phase.

We define confidence score w based on the difference detection results between (d^{before} and d^{after}). We calculate the confidence score w for each pixel u as follows:

$$w_u^{\text{before,after}} = (d_u^{\text{after}} + b_{m_u^{\text{after}}}^{\text{before,after}}) - (d_u^{\text{before}} + b_{m_u^{\text{before}}}^{\text{before,after}}) + \hat{b}, \quad (2)$$

where $\hat{b}$ is a bias for a pair of masks, $b_c^{\text{before,after}}$ is a bias for a class (c, $c \in \mathcal{C}$), and $\mathcal{C}$ is a set of image-level label. We add the class bias $b_c^{\text{before,after}}$ for missing classes between m^{before} and m^{after}. In practice, we select missing classes after applying CRF based on change ratio of pixels. We denote a number of pixels belonging to class c on m^{before} as N_c^{before} and a number of pixels belonging to class c on m^{after} as N_c^{after}. The change ratio of pixels belonging to class c is calculated by:

$$\delta_c^{\text{before,after}} = N_c^{\text{after}} / N_c^{\text{before}}. \quad (3)$$

The class bias b_c is represented using the change ratio of pixels $\delta_c^{\text{before,after}}$ as follows: if $\delta_c^{\text{before,after}} > th$ then $b_c^{\text{before,after}}=b_{\text{CL}}$ else $b_c^{\text{before,after}}=0$, where b_{CL} is a bias for missing classes and we simply set the threshold th to 0.5.

Finally, we obtain the refined mask m^{refine} using the confident score w by:

$$m_u^{\text{refine}} = \begin{cases} m_u^{\text{before}} & \text{if } (w_u^{\text{before,after}} > 0) \\ m_u^{\text{after}} & \text{if } (w_u^{\text{before,after}} < 0) \end{cases} \quad (4)$$

For simplicity, we denote this region refinement processing by $m^{\text{refine}}=\text{SSDD}(e(x;\theta_e), m^{\text{before}}, m^{\text{after}};\theta_d)$.

4. Introducing SSDD modules into the processing flow of WSS

Seed mask generation stage with static region refinement Pixel-level Semantic Affinity (PSA) [1] is a method to propagate label responses to nearby areas which belong to the same semantic entity. In this section, we refine the outputs of CRF in PSA by the proposed Self-Supervised Difference Detection (SSDD) module.

We denote an input image as x and the probability maps obtained by PSA as $p^{\text{psa}}=PSA(x;\theta_{\mathcal{P}})$ and its CRF results as p^{crf0}. We convert the probability maps (p^{psa},p^{crf0}) to segmentation masks by ($m^{\text{psa}}=\arg\max_{k\in\mathcal{C}^y\cup\mathcal{C}^{bg}} p_k^{\text{psa}}, m^{\text{crf0}}=\arg\max_{k\in\mathcal{C}^y\cup\mathcal{C}^{bg}} p_k^{\text{crf0}}$) , where bg indicates a background class. The difference detection result for target $M^{\text{psa,crf0}}$ is computed by $d^{\text{psa}}=f(e^h(x;\theta_{e0}), e^l(x;\theta_{e0}), \hat{m}^{\text{psa}};\theta_{d0})$, $d^{\text{crf0}}=f(e^h(x;\theta_{e0}), e^l(x;\theta_{e0}), \hat{m}^{\text{crf0}};\theta_{d0})$. From the difference detection results, we obtain a target refined mask m^{ssdd0} by the proposed SSDD module as follows: $m^{\text{ssdd0}}=\text{SSDD}(e(x;\theta_{e0}), m^{\text{psa}}, m^{\text{crf0}};\theta_{d0})$.

Suppose that $\sigma()$ is sigmoid function and S is a set of locations. The loss function for a Difference Detection net-

wort (DD-Net) is given by:

$$\mathcal{L}_{\text{change}}(d^{\text{psa}}, d^{\text{crf0}}, M^{\text{psa,crf0}}) = \frac{1}{|S|}\sum_{u\in S}(J(M^{\text{psa,crf0}}, d^{\text{psa}}, u) + J(M^{\text{psa,crf0}}, d^{\text{crf0}}, u)), \tag{5}$$

$J(M^{\text{psa,crf0}}, d^{\text{psa}}, u) = M_u^{\text{psa,crf0}} \log \sigma(d_u^{\text{psa}}) + (1 - M_u^{\text{psa,crf0}}) \log(1 - \sigma(d_u^{\text{psa}}))$,
$J(M^{\text{psa,crf0}}, d^{\text{crf0}}, u) = M_u^{\text{psa,crf0}} \log \sigma(d_u^{\text{crf0}}) + (1 - M_u^{\text{psa,crf0}}) \log(1 - \sigma(d_u^{\text{crf0}}))$

Note that $M^{\text{psa,crf0}}$ corresponds to $M^{\text{before,after}}$ in Eq.(1). The proposed method is not effective in the cases that either or both of the segmentation masks has no correct labels. We exclude the bad training samples by simple processing. and define it by the change ratio of pixels δ. We calculate the change ratio $\delta_c^{\text{psa,crf0}}$ in the same manner to Eq.(3) and obtain the decisions $\gamma_c^{\text{psa,crf0}}$ for the bad training samples by: if $\delta_c^{\text{psa,crf0}} > 0.5$ then $\gamma_c^{\text{psa,crf0}}$=1 else $\gamma_c^{\text{psa,crf0}}$=0. The interpolated loss function $\overline{\mathcal{L}}_{\text{change}}$ as follows: if $|\sum_{k\in\mathcal{C}^y} \gamma_k^{\text{psa,crf0}}| > 0$ then $\overline{\mathcal{L}}_{\text{change}}$=0 else $\overline{\mathcal{L}}_{\text{change}}$=$\mathcal{L}_{\text{change}}$. $\mathcal{C}$ is a set of class and $\mathcal{C}^y \in \mathcal{C}$ is a set of category $\mathcal{C}$ for image-level label y.

We also train embedding function θ_{e0} by training a segmentation network with m^{psa} as follows:

$$\mathcal{L}_{\text{seg}} = -\frac{1}{\sum_{k\in\mathcal{C}}|S_k^{\text{psa}}|}\sum_{k\in\mathcal{C}}\sum_{u\in S_k^{\text{psa}}} \log(h_u^k(x;\theta_{s0})), \tag{6}$$

where S_k^{psa} is a set of locations that belong to class k on the mask m^{psa}. Final loss function for static region refinement using difference detection is given by:

$$\mathcal{L}_{\text{static}} = \mathcal{L}_{\text{seg}} + \overline{\mathcal{L}}_{\text{change}} \tag{7}$$

Training stage of a fully-supervised segmentation model with dynamic region refinement In this work, we propose a novel approach to constrain interpolation of the seed labels during training of a segmentation model. The idea of the constraint is to limit interpolation of seed labels to only predictable regions of the difference be-tween newly generated pixel-level labels and seed labels.

In the first step, for an input image x, we obtain outputs of a segmentation model p^{seg}=$g(x;\theta_{s1})$ and its CRF outputs p^{crf1}, where g is a function of the segmentation model. We convert them to segmentation masks (m^{seg}= $\underset{k\in\mathcal{C}^y\cup\mathcal{C}^{bg}}{\arg\max}\ p_k^{\text{seg}}$, m^{crf1}= $\underset{k\in\mathcal{C}^y\cup\mathcal{C}^{bg}}{\arg\max}\ p_k^{\text{crf1}}$). Then, we obtain refined pixel-level labels m^{ssdd1} by applying the proposed refinement method as follows: m^{ssdd1}=SSDD$(e(x;\theta_{e1}), m^{\text{seg}}, m^{\text{crf1}}; \theta_{d1})$. In the second step, we apply the proposed method to the obtained mask m^{ssdd1} and seed labels m^{ssdd0}. The further refined mask m^{ssdd2} is obtained by m^{ssdd2}=SSDD$(e(x;\theta_{e1}), m^{\text{ssdd1}}, m^{\text{ssdd0}}; \theta_{d2})$. We generate the mask m^{ssdd2} in each iteration and train the segmentation model using the generated mask m^{ssdd2}.

We train the semantic segmentation model with the generated mask m^{ssdd2} by:

$$\mathcal{L}_{\text{seg-main}} = -\frac{1}{\sum_{k\in\mathcal{C}}|S_k^{\text{ssdd2}}|}\sum_{k\in\mathcal{C}}\sum_{u\in S_k^{\text{ssdd2}}} \log(h_u^k(x;\theta_{s1})). \tag{8}$$

The difference detection network (DD-net) between m^{seg} and m^{crf1} is optimized by a below loss:

$$\mathcal{L}_{\text{dd-crf}}(d^{\text{seg}}, d^{\text{crf1}}, M^{\text{seg,crf1}}) = \frac{1}{|S|}\sum_{u\in S}(J(M^{\text{seg,crf1}}, d^{\text{seg}}, u) + J(M^{\text{seg,crf1}}, d^{\text{crf1}}, u)), \tag{9}$$

$J(M^{\text{seg,crf1}}, d^{\text{seg}}, u) = M_u^{\text{seg,crf1}} \log \sigma(d_u^{\text{seg}}) + (1 - M_u^{\text{seg,crf1}}) \log(1 - \sigma(d_u^{\text{seg}}))$,
$J(M^{\text{seg,crf1}}, d^{\text{crf1}}, u) = M_u^{\text{seg,crf1}} \log \sigma(d_u^{\text{crf1}}) + (1 - M_u^{\text{seg,crf1}}) \log(1 - \sigma(d_u^{\text{crf1}}))$.

In the similar manner to Sec.4, in the second stage, we also exclude bad samples based on the change ratio of pixels. We define the interpolated loss function $\overline{\mathcal{L}}_{\text{dd-crf}}$ for the pair of masks (m^{seg}, m^{crf1}) by applying below processing. If $|\sum \gamma_k^{\text{seg,crf1}}| > 0$ then $\overline{\mathcal{L}}_{\text{dd-crf}} = 0$ else $\overline{\mathcal{L}}_{\text{dd-crf}} = \mathcal{L}_{\text{dd-crf}}$.

We explain about the training of the DD-net for (m^{ssdd1}, m^{ssdd0}). The difference detection between m^{ssdd1} and m^{ssdd0} is an easy task because the seed masks m^{ssdd0} are always constant during training. To avoid the problem, we add a new segmentation branch to a segmentation network, which is also trained with the pixel-level seed labels m^{ssdd0}. We obtain segmentation probability maps from this branch p^{sub} and convert the map to mask by m^{sub}= $\underset{k\in\mathcal{C}^y\cup\mathcal{C}^{bg}}{\arg\max}\ p^{\text{sub}}$. We train the difference detection network, DD-net, to predict the difference of the pair of masks (m^{ssdd0}, m^{sub}) and (m^{sub}, m^{ssdd1}). This replacing of the mask makes it hard to predict the difference between m^{ssdd0} and m^{ssdd1} because the situation of training and inference become different. We denote the loss function for the difference detection in this case as follows:

$$\mathcal{L}_{\text{dd-seed}}(d^{\text{ssdd0}}, d^{\text{ssdd1}}, M^{\text{ssdd0,ssdd1}}) = \frac{1}{|S|}\sum_{u\in S}(J(M^{\text{ssdd0,sub}}, d^{\text{ssdd0}}, u) + J(M^{\text{ssdd1,sub}}, d^{\text{ssdd1}}, u)), \tag{10}$$

$J(M^{\text{ssdd0,sub}}, d^{\text{ssdd0}}, u) = M_u^{\text{ssdd0,sub}} \log \sigma(d_u^{\text{ssdd0}}) + (1 - M_u^{\text{ssdd0,sub}}) \log(1 - \sigma(d_u^{\text{ssdd0}}))$
$J(M^{\text{ssdd1,sub}}, d^{\text{ssdd1}}, u) = M_u^{\text{ssdd1,sub}} \log \sigma(d_u^{\text{ssdd1}}) + (1 - M_u^{\text{ssdd1,sub}}) \log(1 - \sigma(d_u^{\text{ssdd1}}))$.

The parameter $\theta_{s1'}$ of the new segmentation branch is trained by:

$$\mathcal{L}_{\text{seg-sub}} = \alpha\mathcal{L}^{\text{ssdd0}} + (1-\alpha)\mathcal{L}^{\text{ssdd2}}, \tag{11}$$

$$\mathcal{L}^{\text{ssdd0}} = -\frac{1}{\sum_{k\in\mathcal{C}}|S_k^{\text{ssdd0}}|}\sum_{k\in\mathcal{C}}\sum_{u\in S_k^{\text{ssdd0}}} \log(h_u^k(x;\theta_{s1'}))$$

$$\mathcal{L}^{\text{ssdd2}} = -\frac{1}{\sum_{k\in\mathcal{C}}|S_k^{\text{ssdd2}}|}\sum_{k\in\mathcal{C}}\sum_{u\in S_k^{\text{ssdd2}}} \log(h_u^k(x;\theta_{s1'})).$$

α is a hyper parameter of the mixing ratio of $\mathcal{L}^{\text{ssdd0}}$ and $\mathcal{L}^{\text{ssdd2}}$.

The final loss function of the proposed dynamic region refinement method is calculated as below:

$$\mathcal{L}_{\text{dynamic}} = \overline{\mathcal{L}}_{\text{dd-crf}} + \mathcal{L}_{\text{dd-seed}} + \mathcal{L}_{\text{seg-main}} + \mathcal{L}_{\text{seg-sub}}. \tag{12}$$

Table 1. Results on PASCAL VOC 2012 *val set*.

methods	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	**mIoU**
PSA [1]	88.2	68.2	30.6	81.1	49.6	61.0	77.8	66.1	75.1	29.0	66.0	40.2	80.4	62.0	70.4	73.7	42.5	70.7	42.6	68.1	51.6	61.7
SSDD	89.0	62.5	28.9	83.7	52.9	59.5	77.6	73.7	87.0	34.0	83.7	47.6	84.1	77.0	73.9	69.6	29.8	84.0	43.2	68.0	53.4	64.9
Gain	*+0.8*	*-5.7*	*-1.7*	*+2.6*	*+3.3*	*-1.5*	*-0.2*	*+7.6*	*+11.9*	*+5.0*	*+17.7*	*+7.4*	*+3.7*	*+15.0*	*+3.5*	*-4.1*	*-12.7*	*+13.3*	*+0.6*	*-0.1*	*+1.8*	*+3.2*

Table 2. Comparison with WSS methods w/o additional supervision.

Method	Val	Test
FCN-MIL [15]ICLR2015	25.7	24.9
CCNN [14]ICCV2015	35.3	35.6
EM-Adapt [13]ICCV2015	38.2	39.6
DCSM [22]ECCV2016	44.1	45.1
BFBP [19]ECCV2016	46.6	48.0
SEC [10]ECCV2016	50.7	51.7
CBTS [18]CVPR2017	52.8	53.7
TPL [9]ICCV2017	53.1	53.8
MEFF [4]CVPR2018	-	55.6
PSA [1]CVPR2018	61.7	63.7
PSA (Re-implementation)	59.0	-
SSDD (Static)	61.4	-
SSDD (Dynamic)	**64.9**	**65.5**

Table 3. Comparison with WSS methods w/ additional supervision.

Method	Additional supervision	Val	Test
MIL-seg [16]CVPR2015	Saliency mask + Imagenet images	42.0	40.6
MCNN [23]ICCV2015	Web videos	38.1	39.8
AFF [17]ECCV2016	Saliency mask	54.3	55.5
STC [26]PAMI2017	Saliency mask + Web images	49.8	51.2
Oh et al. [20]CVPR2017	Saliency mask	55.7	56.7
AE-PSL [25]CVPR2017	Saliency mask	55.0	55.7
Hong et al. [5]CVPR2017	Web videos	58.1	58.7
WebS-i2 [8]CVPR2017	Web images	53.4	55.3
DCSP [2]BMVC2017	Saliency mask	60.8	61.9
GAIN [12]CVPR2018	Saliency mask	55.3	56.8
MDC [27]CVPR2018	Saliency mask	60.4	60.8
MCOF [24]CVPR2018	Saliency mask	60.3	61.2
DSRG [7]CVPR2018	Saliency mask	61.4	63.2
Shen et al. [21]CVPR2018	Web images	63.0	63.9
SeeNet [6]NIPS2018	Saliency mask	63.1	62.8
AISI [3]ECCV2018	Instance saliency mask	63.6	64.5
SSDD (Dynamic)	-	**64.9**	**65.5**

5. Experiments

We evaluated the proposed methods using the PASCAL VOC 2012 data. For calculating the mean IoU on val and test sets, We used the official evaluation server.

Implementation details Our experiments are heavily based on previous work [1]. For the generating results of PSA, we used implementations provided by the authors that are publicly available. We followed the paper [1] and set hyperparameters that achieved the best performance in their paper. As segmentation model we used a ResNet-38 model, which is almost the same to the architecture used in [1] except for upsampling rate. We explore good hyperparameters in the proposed method by grid search.

Comparison Table 1 shows the comparison of dynamic region refinement method with PSA. We denote the dynamic region refinement as "SSDD" in all the tables. We observe that the proposed method outperforms PSA with over 3.2 point margin. In Table 1, we show the gains between the proposed method and PSA for detailed analysis as well.

Table 2 shows the results of the proposed method and recent weakly supervised segmentation methods that use no additional supervisions on PASCAL VOC 2012 validation data and PASCAL VOC 2012 test data. We observe that our method achieves the highest score compared with all the existing methods, which use the same types of supervision [14, 13, 22, 19, 10, 9, 18, 4, 1].

Table 3 shows the comparison of the proposed method with some weakly supervised segmentation methods, which employ relatively cheap additional information. Though completely fair comparisons for them are difficult because of the difference of network model, augmentation technique, the number of iteration epochs and so on, the proposed method demonstrates comparable or better performance without any additional information for training. Fig.4 shows the examples of difference detection results and Fig.5 shows the examples of semantic segmentation results.

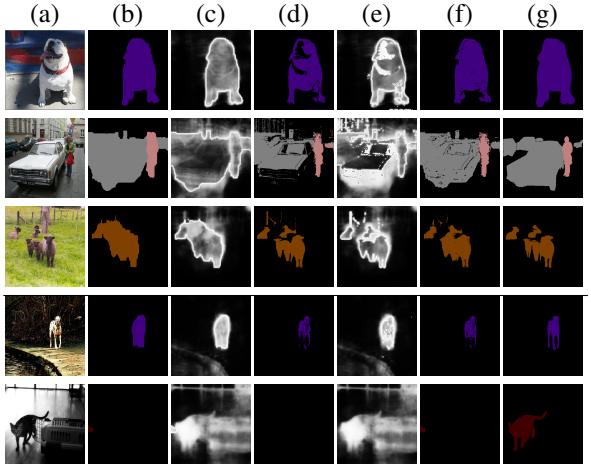

Figure 4. For each row, from the left, (a) input images, (b) Raw PSA segmentation masks, (c) Difference detection maps of (b), (d) CRF masks of (b), (e) Difference detection maps of (d), (f) Refined segmentation masks by the proposed method and (g) Ground truth masks. Two bottom rows show failure cases.

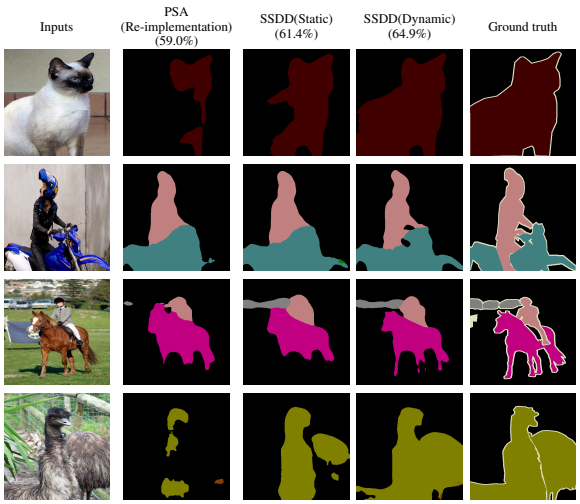

Figure 5. Segmentation examples of results on Pascal VOC 2012.

6. Conclusion

In this paper, we proposed a novel method to refine a segmentation mask from a pair of segmentation masks before and after refinement process such as CRF by the proposed SSDD module. We demonstrated the proposed method can be used effectively in two stages: static region refinement in the seed generation stage, and dynamic region refinement in the training stage.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number 17J10261, 15H05915, 17H01745, 19H04929 and 17H06100.

References

[1] J. Ahn and S. Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, 2018. 1, 2, 4

[2] A. Chaudhry, K. P. Dokania, and P. H. Torr. Discovering class-specific pixels for weakly-supervised semantic segmentation. In *BMVC*, 2017. 4

[3] R. Fan, Q. Hou, M. M. Cheng, G. Yu, R. R. Martin, and S. M. Hu. Associating inter-image salient instances for weakly supervised semantic segmentation. In *ECCV*, 2018. 4

[4] W. Ge, S. Yang, and Y. Yu. Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In *CVPR*, 2018. 4

[5] S. Hong, D. Yeo, S. Kwak, H. Lee, and B. Han. Weakly supervised semantic segmentation using web-crawled videos. In *CVPR*, 2017. 4

[6] Q. Hou, P. T. Jiang, Y. Wei, and M. M. Cheng. Self-erasing network for integral object attention. In *Advances in Neural Information Processing Systems*, 2018. 4

[7] Z. Huang, W. Xinggang, J. Wang, W. Liu, and W. Jingdong. Weakly-supervised semantic segmentation network with deep seeded region growing. In *CVPR*, 2018. 1, 4

[8] B. Jin, M. Segovia, and S. Susstrunk. Webly supervised semantic segmentation. In *CVPR*, 2018. 4

[9] D. Kim, D. Cho, D. Yoo, and I. Kweon. Two-phase learning for weakly supervised object localization. In *ICCV*, 2017. 4

[10] A. Kolesnikov and C. H.Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, 2016. 1, 4

[11] P. Krahenbuhl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, 2011. 1

[12] K. Li, Z. Wu, K.-C. Peng, J. Ernest, and Y. Fu. Tell me where to look: Guided attention inference network. In *CVPR*, 2018. 4

[13] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. In *ICCV*, 2015. 1, 4

[14] D. Pathak, P. Krahenbuhl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, 2015. 1, 4

[15] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully convolutional multi-class multiple instance learning. In *ICLR*, 2015. 4

[16] P. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 2015. 4

[17] X. Qi, Z. Liu, J. Shi, H. Zhao, and J. Jia. Augmented feedback in semantic segmentation under image level supervision. In *ECCV*, 2016. 4

[18] A. Roy and S. Todorovic. Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation. In *CVPR*, 2017. 4

[19] F. Saleh, M. Akbarian, M. Salzmann, L. Petersson, S. Gould, and J. M.Alvares. Built-in foreground/background prior for weakly-supervised semantic segmentation. In *ECCV*, 2016. 4

[20] O. Seong, B. Rodrigo, K. Anna, A. Zeynep, F. Mario, and S. Bernt. Exploiting saliency for object segmentation from image level labels. In *CVPR*, 2017. 4

[21] T. Shen, G. Lin, C. Shen, and R. Ian. Bootstrapping the performance of webly supervised semantic segmentation. In *CVPR*, 2018. 4

[22] W. Shimoda and K. Yanai. Distinct class saliency maps for weakly supervised semantic segmentation. In *ECCV*, 2016. 4

[23] P. Tokmakov, K. Alahari, and C. Schmid. Weakly-supervised semantic segmentation using motion cues. In *ECCV*, 2016. 4

[24] X. Wang, S. You, X. Li, and H. Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. In *CVPR*, 2018. 4

[25] Y. Wei, J. Feng, X. Liang, M. Cheng, Y. Zhao, and S. Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*, 2017. 1, 4

[26] Y. Wei, X. Liang, Y. Chen, X. Shen, M. Cheng, Y. Zhao, and S. Yan. STC: A simple to complex framework for weakly-supervised semantic segmentation. In *IEEE Trans. on PAMI*, 2017. 4

[27] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. Huang. Revisiting dilated convolution: A simple approach for weakly- and semisupervised semantic segmentation. In *CVPR*, 2018. 4