Towards Noiseless Object Contours for Weakly Supervised Semantic Segmentation

Jing Li[1,2,4] Junsong Fan[1,2,3,4] Zhaoxiang Zhang[1,2,3,4*]
[1]Institute of Automation, Chinese Academy of Sciences (CASIA)
[2]University of Chinese Academy of Sciences (UCAS)
[3]Centre for Artificial Intelligence and Robotics, HKISI_CAS
[4]National Laboratory of Pattern Recognition (NLPR)
{lijing2018, fanjunsong2016, zhaoxiang.zhang}@ia.ac.cn

Abstract

Image-level label based weakly supervised semantic segmentation has attracted much attention since image labels are very easy to obtain. Existing methods usually generate pseudo labels from class activation map (CAM) and then train a segmentation model. CAM usually highlights partial objects and produce incomplete pseudo labels. Some methods explore object contour by training a contour model with CAM seed label supervision and then propagate CAM score from discriminative regions to non-discriminative regions with contour guidance. The propagation process suffers from the noisy intra-object contours, and inadequate propagation results produce incomplete pseudo labels. This is because the coarse CAM seed label lacks sufficient precise semantic information to suppress contour noise. In this paper, we train a SANCE model which utilizes an auxiliary segmentation module to supplement high-level semantic information for contour training by backbone feature sharing and online label supervision. The auxiliary segmentation module also provides more accurate localization map than CAM for pseudo label generation. We evaluate our approach on Pascal VOC 2012 and MS COCO 2014 benchmarks and achieve state-of-the-art performance, demonstrating the effectiveness of our method. The source code can be found at https://github.com/BraveGroup/SANCE

1. Introduction

Semantic segmentation has made great progress in recent years thanks to the rapid development of deep neural networks [7, 8, 31, 34, 50]. However, per-pixel annotation for training semantic segmentation network requires huge economic investment and is a time-consuming task. To reduce the heavy burden of precise pixel-level annotation, weakly supervised semantic segmentation (WSSS) is adopted, which uses weak annotations to supervise semantic segmentation network, including but not limited to bounding boxes [9, 22, 35, 40], scribbles [30, 43], point-level [3] and image-level labels [2, 23, 36, 37, 38, 49, 53]. Among all the annotations, image-level label is the most widely used as it is much easier to obtain than other formats, it is available in almost all the datasets. In this paper, we focus on studying image-level label based WSSS problem.

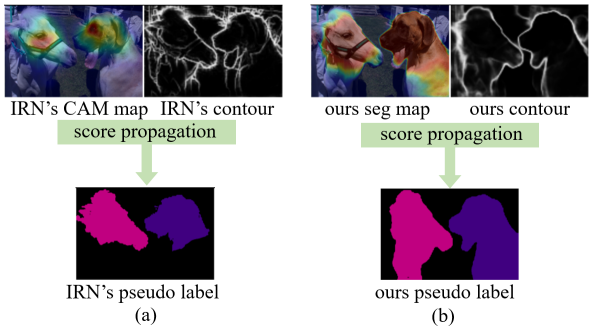

Figure 1. Pseudo label generation of IRNet and ours. (a) IRNet's contour map contains many intra-object edges and misses some true contours, CAM score can not propagate from discriminative region to non-discriminative region. (b) Our SANCE predicts noiseless contour and more complete segmentation map, thus generates better pseudo labels.

Image class label doesn't provide localization, scale or shape information of ground truth objects. Existing approaches usually adopt CAM [55] to estimate localization maps for target objects, the localization maps are then processed to generate pseudo labels for standard segmentation model training. To be specific, these works first train a

* Corresponding author

classification model with image class labels and then obtain CAM by apply the last linear classification layer to each feature vector of the feature map before the global average pooling layer. However, CAM usually highlights only a portion of an object and gives sparse and incomplete object estimates, because the classifier only needs to "see" the most discriminative regions to optimize the classification loss function.

To alleviate the incomplete estimation problem of CAM, some approaches try to expand object estimates during the classification model training process through accumulating activation maps [21], iterative erasing strategy [18], region growing algorithms [19], auxiliary classification task [4], splitting vs. merging strategy [54], etc. Localization maps of these methods cover more complete object regions, but may not give sharp estimates of object shape because the expanding process is controlled implicitly by empirical constraints. Some approaches [1, 6] control the expanding process explicitly with object contour information and thus give sharp object estimates. These methods first train a contour detection network with CAM seeds, then refine CAM by propagating foreground scores from highlighted regions to the missing object parts according to inter-pixel semantic affinity [2] generated from object contour prediction, we call this process as score propagation, finally a standard segmentation network is trained with pseudo labels generated from propagation results. The refined CAM gives sharp object estimates but may still produce inaccurate pseudo labels. As shown in (a) of Fig. 1, the noisy edges at neck region hinder the object score from head region to body region and the pseudo label only covers the head area. This is because the contour supervision signal, CAM seed label, lacks enough high-level semantic information. The CAM is a coarse localization map and we can only get sparse seed labels containing many unsure pixel labels, meanwhile, CAM usually highlights background areas around foreground objects and thus leads to false positive object areas in CAM seed labels.

To make the contour model explore object contours with less noisy intra-object edges, besides CAM seed label supervision, we also adopt an auxiliary segmentation module to provide sufficient high-level semantic information for contour model training. Firstly, the segmentation branch shares semantic knowledge to the contour branch though the shared backbone feature. Secondly, we refine the segmentation map to generate online label to offer sufficient high-level semantic supervision to the contour branch. To make the segmentation branch predict accurate results, we adopt the CAM seed label and online label as training signal. On the whole, our model contains a contour branch and a segmentation branch, we call it segmentation-assisted noiseless contour exploration model (SANCE), these two branches share the same backbone and are supervised with CAM seed label and online label. The online label is generated by refining segmentation map through score propagation under contour constraints. For the contour map with intra-object edges, segmentation map covers more object parts than CAM map, thus an object's neighboring regions devided by noisy edges get high foreground scores in refined segmentation maps, which gives complete object estimates in online label and suppress the noisy intra-object edges. On the other hand, the online label gives more accurate object shapes thanks to the contour information, so it forces the segmentation branch to predict complete and precise object localization maps. After training, our SANCE model predicts noiseless object contour and high quality segmentation map, as show in (b) of Fig. 1, we generate more complete pseudo labels based on them. The main contributions of the paper are summarized as follows:

- We identify the intra-object edge problem in recent contour assisted CAM refining methods for WSSS problem. The intra-object edge may hinder the object score propagation and results in incomplete pseudo labels.
- We introduce SANCE framework to explore noiseless object contours by leveraging high-level semantic information of auxiliary segmentation branch.
- On Pascal VOC 2012 benchmark, we train DeepLab-v2 with generated pseudo labels and achieve the new state-of-the-art performance with 72.0% and 72.9% mIoU on val and test sets, respectively. On MS COCO 2014, we also achieve the new state-of-the-art performance with 44.7% mIoU on val set.

2. Related Work

2.1. Weakly-supervised Semantic Segmentation.

Weakly-supervised method for semantic segmentation attracts a large interest recently due to the simplicity and availability of its required labels compared with fully supervision segmentation learning. Various types of annotations are applied as supervision to address the data deficiency problem, including image-level label [2, 23, 36, 37, 38], bounding box [9, 22, 35] , scribble [30, 43], and so on. In particular, image-level labels as the simplest supervision are popularly used since they demand minimum costs and can be obtained from most visual datasets. In this paper, we study the image-level label based WSSS problem.

2.2. Image-level Supervised Learning.

Image-level class label supplies object class information rather than object localization cues, so most of recent image-level label based WSSS approaches first generate pseudo labels (a.k.a. localization seeds) through localization maps (i.e. CAM and grad-CAM [39], etc.), then

train a standard semantic segmentation model using pseudo labels aforementioned. However, CAM usually highlights the most discriminative parts of objects and gives low or zero activations at other areas, so the pseudo label is often incomplete.

To alleviate this problem, some works try to generate more complete CAM by utilizing new training strategy, new model architecture, or new CAM generation methods. SC-CAM [4] introduces a challenging self-supervised task to enforce the classification network to pay attention to more parts of an object by exploiting the sub-category information. Splitting Vs. Merging [54] adopts discrepancy loss to mine out regions of different spatial patterns except the most discriminative ones, which expanding the output heatmap, and adopts intersection loss to prevent excessive expanding. OAA [21] maintains a cumulative attention map for each target category in each training image so that the object regions become more and more integral as the training goes on. FickleNet [27] generates a variety of localization maps from each image by selecting units of hidden layers randomly and accumulates all the localization maps to get more integral estimates. MDC [46] equips a generic classification network with convolutional blocks of different dilated rates to transfer the discriminative information to non-discriminative object regions and expand the activated regions. However, these methods don't guide the expanding process with low-level boundary information and usually get blurry results at object contour areas.

To get accurate results at object contour areas, some other methods utilize semantic affinity matrix generated from contour map to propagate activation scores of discriminative regions to non-discriminative regions. IRNet [1] trains a network to detect object contours implicitly supervised by inter-pixel affinities generated from CAM seeds. BES [6] generates object contour labels from CAM seeds and trains a contour detection model explicitly supervised by contour labels. PSA [2] trains a inter-pixel affinity model with affinities generated from CAM seeds, the model implicitly exploits object contour by predicting inter-pixel affinities. Approaches [1, 2, 6] all adopt CAM seeds as contour training signal. Supervised by sparse object labels derived from highlighted CAM regions, the contour model may not get enough high-level semantic information for training and finally predicts many intra-object edges, which hinder object scores of CAM propagated to non-discriminative regions.

To make contour model predict contours with less noisy intra-object edges, we train a contour model with an auxiliary segmentation branch. During training, our model generates online labels by refining segmentation branch prediction and utilizes these labels to train contour branch and auxiliary segmentation branch. The online label gives more complete and accurate estimates than CAM seeds, and thus can expand the segmentation map and suppress noisy edges of contour map. Finally, we get accurate pseudo labels based on noiseless contour and precise segmentation map.

3. The Proposed Approach

As shown in Fig. 2, the SANCE training process contains two stages. The first stage adopts CAM to estimate initial coarse seeds from image class labels. In stage2, SANCE learns to predict noiseless object contours supervised by coarse CAM seeds. SANCE contains a contour branch and an auxiliary segmentation branch sharing the same backbone, it learns to explore noiseless object contours with the assistance of its auxiliary segmentation branch. After training, SANCE predicts accurate contour maps and segmentation maps, we adopt these two maps to generate reliable pseudo labels for the standard segmentation model training. In the following sections, we will elaborate on the details of SANCE.

3.1. The CAM Seed

Following previous works [1, 6], we apply CAM [55] to generate initial coarse seeds from image class labels. CAM is obtained by reshaping a trained classification model to produce dense 2D activation maps for each class, activation maps are normalized by filtering the negative values and dividing by the maximum values in each channel. Let $S \in [0,1]^{C\times hw}$ denotes the normalized CAM map, where C is the total number of classes in the dataset, h and w are the spatial size of the input image, then an initial mask Y^{init} is obtained by:

$$Y_i^{init} = \begin{cases} \arg\max\limits_c S_{c,i} & \text{if } \max\limits_c S_{c,i} > 0.3 \\ 0 & \text{if } \max\limits_c S_{c,i} < 0.05 \\ 255 & \text{otherwise} \end{cases}, \quad (1)$$

where 0 stands for the background, and 255 stands for unsure pixels.

Given the initial mask Y^{init}, we follow the previous approach IRNet [1] to further refine it by CRF post-processing [24]. Then, the processed masks are taken as initial CAM seeds Y^{CAM} to provide coarse supervision to our SANCE model.

3.2. The SANCE Model

Taken the initial CAM seeds as supervision, the SANCE model is responsible for producing noiseless object contours, then generates accurate pseudo-masks to train the final semantic segmentation model. As discussed in Sec. 1, the main difficulty of previous approaches to obtain proper object contours is lacking high-level semantic information. To alleviate the noisy edge problem, our SANCE model trains an auxiliary segmentation branch to help its contour

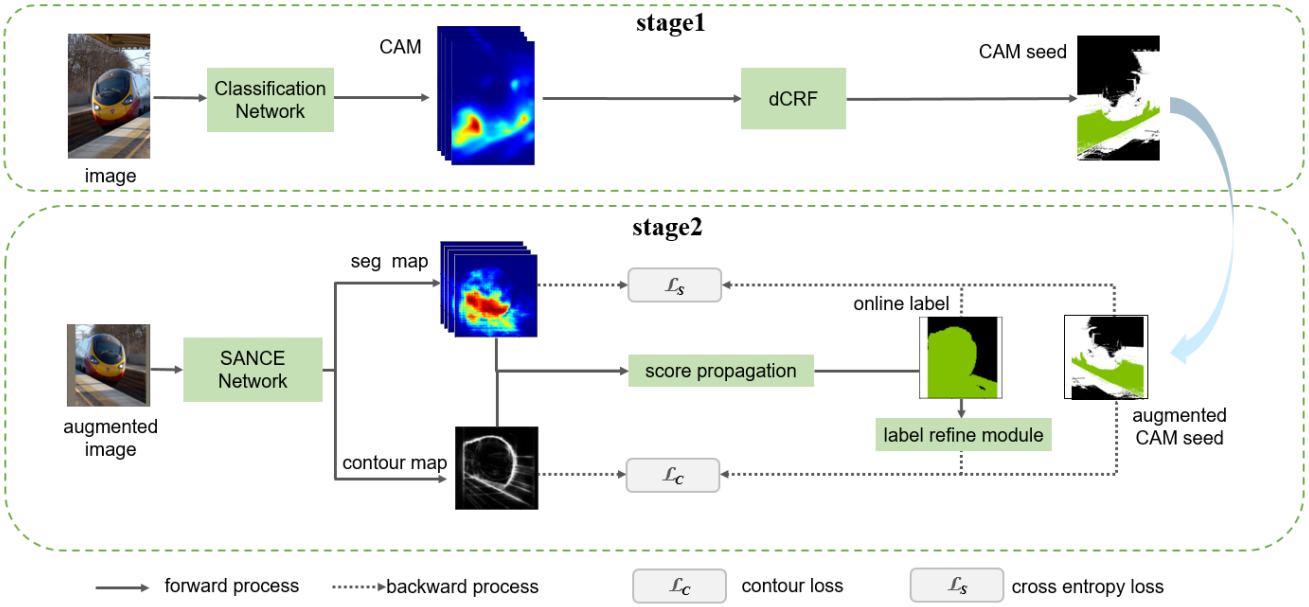

Figure 2. SANCE training process. Given training images, we first obtain their CAM seeds offline from a trained classification network in stage1. Then, we train SANCE with CAM seeds in stage2 by applying data augmentation to training images and CAM seeds. Online label generated from seg map and contour map provides additional training signal to suppress contour noise and expand object areas of seg map. The label refine module refines online label utilizing seg map or saliency map, to make the figure clear, the utilization of these maps is not shown.

branch training, it generates online labels from refined segmentation maps to supervise contour branch and auxiliary segmentation branch, online labels provide high-level semantic information for contour branch and boost auxiliary segmentation branch's performance.

Contour Prediction Branch. The contour prediction branch produces a binary contour map illustrating boundaries across different classes. Formally, let I denote the input image, whose spatial size is $h \times w$, the backbone firstly extracts multi-stage features F from I. Then, the contour branch $\mathcal{C}$ predicts binary maps $B = \mathcal{C}(F)$, whose spatial size is $\frac{h}{4} \times \frac{w}{4}$, we denote $\frac{h}{4} \times \frac{w}{4}$ as $\hat{h} \times \hat{w}$ for convenience. Note that the contour map B is normalized into $[0, 1]$ by the sigmoid function.

To optimize the contour map B with a semantic segmentation seed Y (CAM seeds or online label) , we firstly compute the pixel-pair affinities from B by:

$$a_{ij} = \begin{cases} \left(1 - \max_{k \in \mathcal{P}_{ij}} B_k\right)^n & \text{if } d(i,j) < \delta \\ 0 & \text{otherwise} \end{cases}, \quad (2)$$

where $\mathcal{P}_{ij}$ is the set of pixels along the path from i to j, power n is a hyper-parameter controls the sensitivity of the contours, $d(i,j)$ evaluates the Euclidean distance between pixel i and j, δ is a threshold. By means of this, the affinity a_{ij} owns high scores in coherent local regions without crossing contours.

Then, we derive affinity targets from seed Y:

$$\hat{a}_{ij} = \begin{cases} 0 & \text{if } Y_i \neq Y_j;\ Y_i, Y_j \neq 255;\ d(i,j) < \hat{\delta} \\ 1 & \text{if } Y_i = Y_j;\ Y_i, Y_j \neq 255;\ d(i,j) < \hat{\delta} \\ 255 & \text{otherwise} \end{cases}, \quad (3)$$

where Y_i and Y_j are the labels from Y. Note that the target affinity is only defined in non-ignore regions of Y.

In this way, the affinity targets are defined to be positive in local regions with the same class label. The loss to train B is then defined as:

$$\mathcal{L}_C(B, Y) = -\sum_{\hat{a}_{ij}=1, Y_i=0} \frac{\log(a_{ij})}{2N_{bg}^+} - \sum_{\hat{a}_{ij}=1, Y_i>0} \frac{\log(a_{ij})}{2N_{fg}^+} - \sum_{\hat{a}_{ij}=0} \frac{\log(1-a_{ij})}{N^-}, \quad (4)$$

where N_{bg}^+, N_{fg}^+, N^- are normalization factors corresponding to the number of positive affinities of background region, positive affinities of foreground region, and negative affinities, respectively.

Auxiliary Segmentation Branch. The auxiliary segmentation branch enhances the contour branch with semantic information in two ways. Firstly, it transfers semantic knowl-

edge to the contour branch through shared backbones. Secondly, we supervise both branches utilizing online labels generated by refining segmentation map with contour information. The online label transfers semantic information of segmentation map to the contour branch and boosts the segmentation branch at the same time. Denote the segmentation branch with symbol $\mathcal{S}$, it produces segmentation maps $M = \mathcal{S}(F)$ from the shared feature F, M with size $C \times \hat{h} \times \hat{w}$is normalized by the softmax function. Parameters of segmentation branch and the shared backbones are optimized by:

$$\mathcal{L}_S(M, Y) = -\frac{1}{N_s} \sum_{\{i|Y_i \neq 255\}} \log M_{Y_i,i}, \tag{5}$$

where, N_s is the normalization factor that equals the number of non-ignore pixels, $M_{Y_i,i}$ is the i-th pixel in Y_i-th channel of M. Here Y can be CAM seed label or online label.

Online Label Generation. CAM seeds, which guide the contours and the auxiliary segmentation branch, provide insufficient high-level semantic information for model training. Although the training of contours is relatively robust to incomplete seeds due to the sparsely sampled local pairwise constraints, better seeds with rich high-level semantic information can effectively provide the contour branch with more reliable pixel pairs. Therefore, we manage to generate online label by revising segmentation map with contour map. The online label is used to supervise the contour map and segmentation map using the loss function Eq. (4) and Eq. (5), respectively.

Given the contour map B and the segmentation map M, we first refine M through the score propagation strategy [1] to obtain boundary-aligned segmentation maps M^b. Specifically, the pixel-to-pixel affinity matrix is obtained by evaluating Eq. (2), $A = [a_{ij}] \in [0, 1]^{\hat{h}\hat{w} \times \hat{h}\hat{w}}$. Then, A is normalized by column and denoted by $\hat{A}$, which describes the propagation probability among different pixels. Given the reshaped $M \in [0, 1]^{C \times \hat{h}\hat{w}}$ and $B \in [0, 1]^{1 \times \hat{h}\hat{w}}$, the propagated scores M^b are obtained by:

$$M^b = [M \odot (1 - B)]\, \hat{A}^t, \tag{6}$$

where, t is the number of iteration, $\odot$ denotes element-wise multiplication.

To reduce the influence of noise, we set M^b's channels of classes not existing in the image I to 0, and set M^b's background channel as a constant value τ_b. Finally, we generate online label Y^b based on M^b by Eq. (7):

$$Y^b[i] = \arg\max_c M^b_c[i] \tag{7}$$

where M^b_c is the c-th channel of M^b, i denotes the i-th pixel.

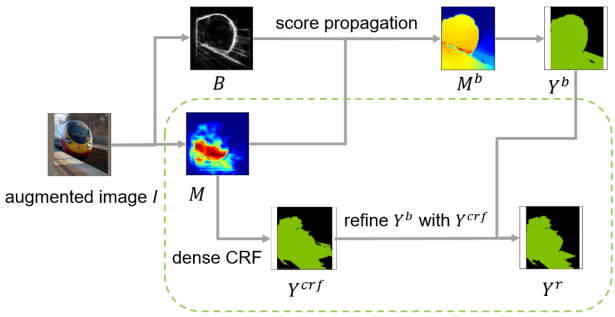

Figure 3. Label refine module. We apply dense CRF to M and get Y^{crf}, we utilize Y^{crf} to revise the false positive foreground areas of online label Y^b.

Label Refine Module. As show in Fig. 3, during the training process, the segmentation map may highlight a small background area, then score propagation will just spread these wrong scores to more background areas and finally results in many false positive foreground pixel labels of Y^b, Y^b with these errors will suppress true object contours and make the excessive propagation problem more serious in later training iterations. To alleviate this problem, we apply dense CRF to segmentation map M and get Y^{crf} with less error foreground labels, then utilize Y^{crf} to refine Y^b and get Y^r as follows:

$$Y^r_i = \begin{cases} Y^{crf}_i & \text{if } Y^{crf}_i = 0 \\ Y^b_i & \text{otherwise} \end{cases} \tag{8}$$

We can also adopt saliency map Y^{sal} generated by saliency models to refine Y^b by replacing Y^{crf} with Y^{sal} in Eq. (8), 0 in Y^{sal} means background, utilizing Y^{sal} usually gets better results.

SANCE Model Training. Both the CAM seed Y^{CAM} and the online label Y^r or Y^b are utilized to supervise our SANCE model by Eqs. (4) and (5). As show in Fig. 2, we adopt data augmentation for input image and CAM seeds, a patch of raw image is resized and placed at a rectangle area of augmented image, the online label generation and supervision is only implemented at corresponding rectangle area of contour map and segmentation map.

3.3. Synthesizing Pseudo Label for Segmentation Models

The ultra goal of WSSS is to obtain a normal segmentation model for deployment. Thus, after training the proposed SANCE model, we adopt it to generate pseudo labels to train standard segmentation models for evaluation.

The pseudo labels are generated by composing the contour map and segmentation map in a similar way as the online label Y^b generation. The difference is that we input

Setting	stride	map	label (*train*)	deeplabv2(*val*)
baseline	16	CAM	66.6	64.7
SANCE-naive	16	CAM	69.9	68.2
SANCE-naive	8	CAM	70.7	68.7
SANCE-naive	8	seg	74.6	69.2
SANCE	8	seg	75.7	69.9
SANCE+TTA	8	seg	**76.9**	**70.9**

Table 1. Pseudo label quality and deeplabv2 performance when adding different components to baseline, evaluated on the PASCAL VOC 2012 *train* set and *train* set, respectively. baseline: the model with only one contour branch, supervised by CAM seeds , SANCE-naive: SANCE without label refine module, TTA: use multi-scale data augmentation when generation pseudo label, stride: stride of model backbone, map: the map revised by contour for pseudo label generation.

multi-scale images into the SANCE model and use the averaged contour map and segmentation map for pseudo label generation.

4. Experiments

4.1. Dataset and Evaluation Metrics

We conduct experiments on two segmentation benchmarks, PASCAL VOC 2012 [11] and MS COCO 2014 [32]. PASCAL VOC 2012 has 20 object classes and a background class for semantic segmentation. Following the common practice, we extend the training set by adopting the training data form SBD [15]. In total, there are 10582 training images, 1449 validation images and 1456 test images. COCO 2014 onsists of 80 object classes and a background class, with 80K and 40K images for training and validation, respectively. For all experiments, we only leverage the image-level class labels for model optimization. We use mean intersection-over-union (mIoU) to evaluate the semantic segmentation performance. We also evaluate trained models' contour quality on SBD benchmark [15], which contains semantic boundary annotations of 11355 images from PASCAL VOC 2011 dataset. Our contours are class-agnostic, so we transform ground-truth semantic contour labels into class-agnostic contour labels and test the raw contour maps with Maximal F-measure (ODS) on SBD *val* set containing 2857 images.

4.2. Implementation Details

Classification Network. Following IRNet [1], we adopt ResNet101, ResNet50 [16] pretrained on ImageNet [10] as backbone on VOC 2012 and COCO 2014, respectively, utilizing SGD with weight decay $1e$-4 and momentum 0.9 as optimizer. The initial learning rate is 0.1 and is decayed at every iteration with polynomial policy [33]. The model is trained with batch size 16 for 5 epochs.

SANCE Network. The backbone is based on ResNet101, ResNet50 [16] pretrained on ImageNet [10] for VOC 2012 and COCO 2014, respectively. Following deeplab's [8] strategy, the stride in the last two stages is reduced from 2 to 1, we adopt dilation 4 in stage5 and 2 in stage4. To predict the contour map, we adopt feature maps from 5 stages of the backbone to get rich semantic information and spatial information. We apply Aspp [8] on the top of the backbone as segmentation branch. We adopt the SGD optimizer with weight decay $5e$-4 and momentum 0.9. The initial learning rate for backbone is $2.5e-4$ and is decayed every 10 iterations with polynomial policy [33], the learning rate for new layers of two branched is amplified by 10. The model is trained with batch size 10 for 16 epochs.

Standard Segmentation Network. we adopt the deeplab v2 framework [8] to evaluate the pseudo label generated by SANCE, the training setting is the same as [8], We employ the multi-scale merging strategy during model evaluation. The final prediction is further refined with dense CRF [24] on VOC 2012 but not on COCO 2014.

Hyper-Parameter. We set $\delta, \hat{\delta} = 10, n = 1$ for contour loss Eq. (4), set $\delta = 3, n = 10, t = 8$ for online label generation, set $\delta = 5, n = 10, t = 8$ for pseudo label generation.

Data Augmentation. For both classification network and SANCE, training images are augmented with random scaling, random flipping and randomly cropped into size 512.

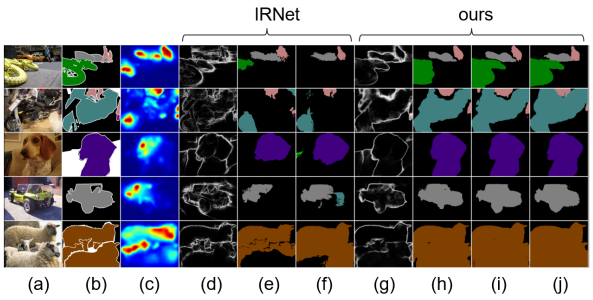

Figure 4. Results of IRNet and ours on PASCAL VOC 2012 *train* set. (a) Input image. (b) Ground truth. (c) CAM. (d) IRNet's contour. (e) IRNet's CAM propagation result. (f) Prediction of deeplabv2 trained with labels in (e). (g) Ours contour. (h) Ours CAM propagation results. (i) Ours segmentation map propagation results. (j) Prediction of deeplabv2 trained with labels in (i).

Setting	backbone stride	MF(%)
baseline	16	57.0
SANCE-naive	16	60.9
SANCE-naive	8	61.5
SANCE	8	**62.4**

Table 2. Contour quality when adding different components to baseline, evaluated on the SBD *val* set.

seg	contour	label (*train*)	deeplabv2 (*val*)
		67.8	66.3
✓		66.5	65.0
	✓	71.3	68.2
✓	✓	**74.9**	**69.6**

Table 3. Pseudo label quality and deeplabv2 performance of different online label supervision setting, evaluated on the PASCAL VOC 2012 *train* set and *val* set, respectively.

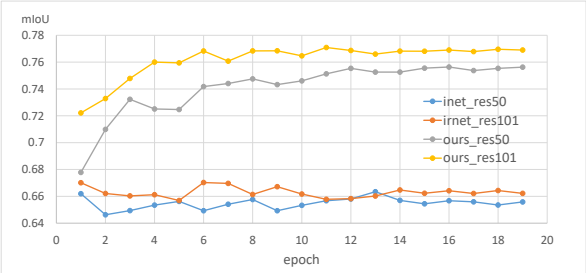

Figure 5. pseudo label quality of IRNet and SANCE at different training epochs, evaluated on the PASCAL VOC 2012 *train* set.

Figure 6. Qualitative results of deeplabv2 on the PASCAL VOC 2012 validation set. (a) Input images. (b) Ground truth. (c) Our results.

refining for M	refining for B	label (*train*)	deeplabv2 (*val*)
		74.9	69.6
✓		76.0	70.6
	✓	**76.9**	**70.9**
✓	✓	75.7	70.4

Table 4. Pseudo label quality and deeplabv2 performance when refine the online label for different branches, evaluated on the PASCAL VOC 2012 *train* set and *val* set, respectively.

4.3. Ablation Study

Element-Wise Module Analysis. We reimplemnet IRNet [1] as our basline, the learning rate and training parameters are the same with SANCE model. The baseline adopts ResNet101 of stride 16 as backbone, it contains a contour branch supervised only with CAM seeds and refines CAM map with single-scale contour result to generate pseudo labels. To compare SANCE and baseline, we evaluate their pseudo labels and deeplabv2 model trained with pseudo label on PASCAL VOC 2012 *train* set and *val* set, respectively, we also evaluate their contour quality on As shown in Tabs. 1 and 2, when refining CAM map, with backbone of stride 16, SANCE-naive improves baseline result by 3.3%, 3.5% and 3.9% for pseudo label, deeplabv2 and contour quality, respectively, and Fig. 4 shows that our method predicts contours with less noisy intra-object edges, demonstrating the effectiveness of online label supervision. Adopting stronger backbone of stride 8 further improves SANCE-naive's result by 0.8%, 0.5% and 0.6% in three aspects mentioned earlier. When refining more accurate segmentation map, pseudo label and deeplabv2 are improved by 3.9% and 0.5%, demonstrating the effectiveness of high quality segmentation map for pseudo label generation. When utilizing label refine module (refine online label with CRF label Y^crf), SANCE further improves both pseudo label and contour quality compared with SANCE-naive. Finally, multi-scale data augmentation (TTA) also improves pseudo label quality and deeplabv2 performance remarkably.

Pseudo Label at Different Training Epochs. In this part we study the pseudo label quality at different training epochs and compare them with the baseline aforementioned. As Fig. 5 shows, baseline's pseudo labels achieve best performance at the first epoch and tends to degrade as training continues. Our model's pseudo labels get better results at the first epoch and improve continually as training goes on. The result shows that the semantic information existing in the model can be extracted online to supervise the model itself, so that online label and model can be improved iteratively.

Influence of Online Label Supervision. We study the influence of online label supervision for SANCE-naive. Specially, we train the network with four different settings by choosing whether to supervise each branch with online label Y^b, here we don't utilize label refine module and take the setting without Y^b supervision as baseline. We evaluate the pseudo label and deeplabv2 model trained with pseudo label on PASCAL VOC 2012 *train* set and *val* set, respectively. Results in Tab. 3 show that adding online label supervision to contour branch can improve pseudo label quality and supervising both branches with Y^b further produces the best result, demonstrating that online label can indeed improve contour branch. Only supervising segmentation map with Y^b produces slightly worse results than baseline, this is because contour branch usually overfits the noisy CAM seeds without online label supervision.

Influence of Label Refine Module. During the SANCE training process, we adopt label refine module to alleviate the excessive propagation problem. We study the influence of label refine module by conducting four experiments of refining online label Y^b for different branches, all experiments utilize CRF label Y^{crf} to refine Y^b. We evaluate the pseudo label and deeplabv2 model trained with pseudo label on PASCAL VOC 2012 *train* set and *val* set, respec-

Method	Backbone	Label	Val	Test
MCOF CVPR'18 [44]	ResNet-101	I.+S.	60.3	61.2
DCSP BMVC'17 [5]	ResNet-101	I.+S.	60.8	61.9
DSRG CVPR'18 [19]	ResNet-101	I.+S.	61.4	63.2
AffinityNet CVPR'18 [2]	Wide ResNet-38	I.	61.7	63.7
SeeNet NIPS'18 [18]	ResNet-101	I.+S.	63.1	62.8
Zeng *et al* ICCV'19 [51]	DenseNet-169	I.+S.	63.3	64.3
OAA ICCV'19 [21]	ResNet-101	I.+S.	63.9	65.6
CIAN AAAI'20 [13]	ResNet-101	I.+S.	64.1	64.7
FickleNet CVPR'19 [27]	ResNet-101	I.+S.	64.9	65.3
CONTA NeurIPS'20 [52]	ResNet38	I.	66.1	66.7
ICD CVPR'20 [12]	ResNet101	I.+S.	67.8	68.0
IRNet CVPR'19 [1]	ResNet50	I.	63.5	64.8
SC-CAM CVPR'20 [4]	ResNet101	I.	66.1	65.9
SEAM CVPR'20 [45]	ResNet38	I	64.5	65.7
BES ECCV'20 [6]	ResNet101	I.	65.7	66.1
Zhang *et al* ECCV'20 [54]	ResNet50	I.+S.	66.6	66.7
Fan *et al* ECCV'20 [14]	ResNet101	I.+S.	67.2	66.7
AuxSegNet ICCV'21 [48]	ResNet38	I.+S.	69.0	68.6
Zhang *et al* ICCV'21 [53]	ResNet38	I.	67.8	68.5
ECS-Net ICCV'21 [42]	ResNet38	I.	66.6	67.6
Kweon *et al*. ICCV'21 [25]	ResNet38	I.	68.4	68.2
CDA ICCV'21 [41]	ResNet38	I.	66.1	66.8
RIB NeurIPS'21 [26]	ResNet101	I.+S.	70.2	70.0
EDAM CVPR'21 [47]	ResNet101	I.+S.	70.9	70.6
Yao *et al* CVPR'21 [49]	ResNet101	I.+S.	68.3	68.5
EPS CVPR'21 [29]	ResNet101	I.+S.	70.9	70.8
AdvCAM CVPR'21 [28]	ResNet101	I.	68.1	68.0
Ours	ResNet101	I.	70.9	72.2
Ours-sal	ResNet101	I.+S.	**72.0**	**72.9**

Table 5. Comparison of WSSS methods on the PASCAL VOC 2012 *val* and *test* sets. I. means image-level labels, S. means external saliency maps.

tively. Results in Tab. 4 show that refining online label for any branch can improve SANCE's performance, and refining online label for one branch gets better results than refining both branches, we think this is because refining both branches suppress the object area expanding process too much and SANCE will produce pseudo labels with less complete objects. We get the best result when refine online label for contour branch and take this refining choice as SANCE's default setting.

4.4. Comparisons to the State-of-the-Art

To compare our approach with other related works on PASCAL VOC 2012, we train a DeppLab-ASPP [8] model using pseudo labels generated by SANCE+TTA. DeppLab-ASPP and SANCE all adopt Resnet101 [16] as backbone. The results in Tab. 5 show that our approach outperforms all the previous image-level label supervised methods both in val and $test$ set of PASCAL VOC 2012. Some previous works [5, 12, 18, 19, 21, 27, 29, 44, 47, 48, 49] use saliency model [17, 20] to generate precise background seeds for segmentation model training, **Ours** outperforms these methods without the help of external saliency model.

Method	Backbone	Label	Val
EPS CVPR'21 [29]	VGG16	I.+S.	**35.7**
AuxSegNet ICCV'21 [48]	ResNet38	I.+S.	33.9
SEAM CVPR'20 [45]	ResNet38	I.	31.9
Kweon *et al* ICCV'21 [25]	ResNet38	I.	36.4
CDA ICCV'21 [41]	ResNet38	I.	33.2
CONTA NeurIPS'20 [52]	ResNet50	I.	33.4
IRNet CVPR'19 [1]	ResNet101	I.	41.4†
RIB NeurIPS'21 [26]	ResNet101	I.	43.8
Ours	ResNet50	I.	**44.7**

Table 6. Comparison of WSSS methods on the MS COCO *val* set. I. means image-level labels, S. means external saliency maps.

With saliency seeds assistance for SANCE training (refine online label with saliency seeds during SANCE training), **Ours-sal** achieves new state-of-the-art and outperforms other works significantly. We also show some segmentation results in Fig. 6, our segmentation results are very close to the ground truth label.

In Tab. 6, we further show the result on COCO 2014 dataset. Here we generate pseudo labels utilizing SANCE+TTA with Resnet50 backbone, then train a DeppLab-ASPP [8] based on Resnet50 [16] and evaluate its performance on validation set of COCO 2014. Our method achieves new state-of-the-art and outperforms related works remarkably.

5. Conclusion

In this paper, we propose a simple yet effective approach to generate accurate pseudo segmentation labels from sparse and incomplete CAM seeds. We train a SANCE model to predict object contour map and segmentation map, supervised by CAM seeds and online label simultaneously. The online label is generated from both maps through score propagation and helps SANCE to learn object contour with less noisy intra-object edges. We finally generate pseudo labels by refining segmentation prediction with contour prediction of trained SANCE model. Extensive experiments demonstrate that the online label boosts both maps' performance, which contributes more accurate pseudo label. Based on the pseudo label, the trained segmentation models achieve new state-of-the-art performance on the Pascal VOC 2012 and MS COCO 2014 segmentation benchmark.

Acknowledgement. This work was supported in part by the Major Project for New Generation of AI (No.2018AAA0100400), the National Natural Science Foundation of China (No. 61836014, No. U21B2042, No. 62072457, No. 62006231).

† This result is reported by RIB [26], IRNet [1] doesn't offer COCO 2014 results.

References

[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, 2019. 2, 3, 5, 6, 7, 8

[2] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, 2018. 1, 2, 3, 8

[3] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016. 1

[4] Yu-Ting Chang, Q. Wang, Wei-Chih Hung, Robinson Piramuthu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Weakly-supervised semantic segmentation via sub-category exploration. In *CVPR*, 2020. 2, 3, 8

[5] Arslan Chaudhry, Puneet K Dokania, and Philip HS Torr. Discovering class-specific pixels for weakly-supervised semantic segmentation. In *BMVC*, 2017. 8

[6] L. Chen, Weiwei Wu, Chenchen Fu, Xiaojing Han, and Yun-Tao Zhang. Weakly supervised semantic segmentation with boundary exploration. In *ECCV*, 2020. 2, 3, 8

[7] Liang-Chieh Chen, G. Papandreou, I. Kokkinos, Kevin Murphy, and A. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR*, abs/1412.7062, 2015. 1

[8] Liang-Chieh Chen, G. Papandreou, I. Kokkinos, Kevin Murphy, and A. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 40:834–848, 2018. 1, 6, 8

[9] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 1, 2

[10] Jia Deng, W. Dong, R. Socher, L. Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[11] M. Everingham, L. Gool, C. K. Williams, J. Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88:303–338, 2009. 6

[12] Junsong Fan, Zhaoxiang Zhang, Chunfeng Song, and Tieniu Tan. Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation. In *CVPR*, 2020. 8

[13] Junsong Fan, Zhaoxiang Zhang, and Tieniu Tan. Cian: Cross-image affinity net for weakly supervised semantic segmentation. In *CVPR*, 2019. 8

[14] Junsong Fan, Zhaoxiang Zhang, and Tieniu Tan. Employing multi-estimations for weakly-supervised semantic segmentation. In *ECCV*, 2020. 8

[15] Bharath Hariharan, Pablo Arbeláez, Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 6

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 8

[17] Q. Hou, Ming-Ming Cheng, X. Hu, A. Borji, Zhuowen Tu, and P. Torr. Deeply supervised salient object detection with short connections. In *CVPR*, 2017. 8

[18] Q. Hou, Peng-Tao Jiang, Yunchao Wei, and Ming-Ming Cheng. Self-erasing network for integral object attention. In *NeurIPS*, 2018. 2, 8

[19] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *CVPR*, 2018. 2, 8

[20] Huaizu Jiang, Zejian Yuan, Ming-Ming Cheng, Yihong Gong, Nanning Zheng, and Jingdong Wang. Salient object detection: A discriminative regional feature integration approach. *IJCV*, 123:251–268, 2013. 8

[21] Peng-Tao Jiang, Qibin Hou, Yang Cao, Ming-Ming Cheng, Yunchao Wei, and Hong-Kai Xiong. Integral object mining via online attention accumulation. In *ICCV*, 2019. 2, 3, 8

[22] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 2017. 1, 2

[23] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, 2016. 1, 2

[24] Philipp Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, 2011. 3, 6

[25] Hyeokjun Kweon, Sung-Hoon Yoon, Hyeonseong Kim, Daehee Park, and Kuk-Jin Yoon. Unlocking the potential of ordinary classifier: Class-specific adversarial erasing framework for weakly supervised semantic segmentation. In *ICCV*, 2021. 8

[26] Jungbeom Lee, Jooyoung Choi, Jisoo Mok, and Sungroh Yoon. Reducing information bottleneck for weakly supervised semantic segmentation. In *NeurIPS*, 2021. 8

[27] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *CVPR*, 2019. 3, 8

[28] Jungbeom Lee, Eunji Kim, and Sungroh Yoon. Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In *CVPR*, 2021. 8

[29] Seungho Lee, Minhyun Lee, Jongwuk Lee, and Hyunjung Shim. Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In *CVPR*, 2021. 8

[30] Di Lin, Jifeng Dai, J. Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016. 1, 2

[31] Guosheng Lin, A. Milan, Chunhua Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017. 1

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6

[33] W. Liu, Andrew Rabinovich, and A. Berg. Parsenet: Looking wider to see better. *ArXiv*, abs/1506.04579, 2015. 6

[34] J. Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[35] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. In *ICCV*, 2015. 1, 2

[36] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly super-

vised segmentation. In *ICCV*, 2015. 1, 2

[37] Pedro O Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 2015. 1, 2

[38] Pedro O Pinheiro and Ronan Collobert. Weakly supervised semantic segmentation with convolutional networks. In *CVPR*, 2015. 1, 2

[39] R. R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, D. Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *IJCV*, 128:336–359, 2019. 2

[40] Chunfeng Song, Y. Huang, Wanli Ouyang, and L. Wang. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In *CVPR*, 2019. 1

[41] Yukun Su, Ruizhou Sun, Guosheng Lin, and Qingyao Wu. Context decoupling augmentation for weakly supervised semantic segmentation. In *ICCV*, 2021. 8

[42] Kunyang Sun, Haoqing Shi, Zhengming Zhang, and Yongming Huang. Ecs-net: Improving weakly supervised semantic segmentation by using connections between class activation maps. In *ICCV*, 2021. 8

[43] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *CVPR*, 2017. 1, 2

[44] Xiang Wang, Shaodi You, Xi Li, and Huimin Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. In *CVPR*, 2018. 8

[45] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *CVPR*, 2020. 8

[46] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S Huang. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *CVPR*, 2018. 3

[47] Tong Wu, Junshi Huang, Guangyu Gao, Xiaoming Wei, Xiaolin Wei, Xuan Luo, and Chi Harold Liu. Embedded discriminative attention mechanism for weakly supervised semantic segmentation. In *CVPR*, 2021. 8

[48] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, Ferdous Sohel, and Dan Xu. Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation. In *ICCV*, 2021. 8

[49] Yazhou Yao, Tao Chen, Guo-Sen Xie, Chuanyi Zhang, Fumin Shen, Qi Wu, Zhenmin Tang, and Jian Zhang. Non-salient region object mining for weakly supervised semantic segmentation. In *CVPR*, 2021. 1, 8

[50] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 1

[51] Yu Zeng, Yunzhi Zhuge, Huchuan Lu, and Lihe Zhang. Joint learning of saliency detection and weakly supervised semantic segmentation. In *CVPR*, 2019. 8

[52] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xiansheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. In *NeurIPS*, 2020. 8

[53] Fei Zhang, Chaochen Gu, Chenyue Zhang, and Yuchao Dai. Complementary patch for weakly supervised semantic segmentation. In *ICCV*, 2021. 1, 8

[54] T. Zhang, Guosheng Lin, W. Liu, Jianfei Cai, and A. Kot. Splitting vs. merging: Mining object regions with discrepancy and intersection loss for weakly supervised semantic segmentation. In *ECCV*, 2020. 2, 3, 8

[55] B. Zhou, A. Khosla, Àgata Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 1, 3