Splitting vs. Merging: Mining Object Regions with Discrepancy and Intersection Loss for Weakly Supervised Semantic Segmentation

Tianyi Zhang[1,2], Guosheng Lin[1], Weide Liu[1], Jianfei Cai[3,1], and Alex Kot[1]

[1] Nanyang Technological University, Singapore
{zh0023yi,gslin,weide001,asjfcai,eackot}@ntu.edu.sg
[2] Institute for Infocomm Research, A*star, Singapore
Zhang_Tianyi@i2r.a-star.edu.sg
[3] Monash University, Australia
jianfei.cai@monash.edu

Abstract. In this paper we focus on the task of weakly-supervised semantic segmentation supervised with image-level labels. Since the pixel-level annotation is not available in the training process, we rely on region mining models to estimate the pseudo-masks from the image-level labels. Thus, in order to improve the final segmentation results, we aim to train a region-mining model which could accurately and completely highlight the target object regions for generating high-quality pseudo-masks. However, the region mining models are likely to only highlight the most discriminative regions instead of the entire objects. In this paper, we aim to tackle this problem from a novel perspective of optimization process. We propose a Splitting vs. Merging optimization strategy, which is mainly composed of the Discrepancy loss and the Intersection loss. The proposed Discrepancy loss aims at mining out regions of different spatial patterns instead of only the most discriminative region, which leads to the splitting effect. The Intersection loss aims at mining the common regions of the different maps, which leads to the merging effect. Our Splitting vs. Merging strategy helps to expand the output heatmap of the region mining model to the object scale. Finally, by training the segmentation model with the masks generated by our Splitting vs Merging strategy, we achieve the state-of-the-art weakly-supervised segmentation results on the Pascal VOC 2012 benchmark.

Keywords: Weakly-Supervised Learning, Deep Convolutional Neural Network (DCNN), Semantic Segmentation

1 Introduction

The performance of semantic segmentation has been remarkable improved by recent deep learning developments [16, 14]. The segmentation models trained with pixel-level ground-truth could achieve remarkable segmentation accuracy. However, one of the obstacles to limit the developments of semantic segmentation

is that the pixel-wise segmentation ground-truth is quite time-consuming and expensive to annotate. One way to reduce the need of pixel-wise annotations is to utilize weaker level of supervisions in the training stage. The weak supervisions include but are not limited to bounding boxes, points, scribbles and image-level labels. Among all the supervision formats, image-level label is the easiest format to annotate and has been widely studied in the weakly supervised learning. However, semantic segmentation supervised with image-level labels is a difficult task, since there is no localization and scale information of the ground-truth objects provided by the training images.

Thus, region-mining techniques are utilized to estimate object localization and scales from image-level labels. The region-mining model, or the object localization model, is usually an image-classification model which could induce class-specific localization maps. The highlighted regions of the localization maps usually correspond to the image labels, which is an approximation of the target object localization and scales. However, region mining models usually only select the most discriminative parts, which deviates from our goal to estimate the complete integral object regions. The main underlying reason is that the region mining models are optimized solely with the classification loss. Thus, targeting only the most discriminative parts is enough for the classification purpose.

In order to alleviate such limitations of region mining models, previous works usually follow the erasing vs. mining pipeline, which is to mine out the most discriminative region, erase it in the feature space, then re-train the region mining model to detect the next discriminative region. The final localization map is the union of all the output maps in different erasing steps. Such erasing operation manipulates the feature space in the forward pass, which may be complicated to implement since it requires multiple steps of model training and post-processing operations.

Different from the previous erasing operations in the forward pass, we tackle this problem from the perspective of the backward pass, or the optimization process. Intrinsically speaking, our goal is to search localization maps of different spatial patterns which all satisfy the classification purpose and the union of all the maps can highlight the entire object regions. Thus, we propose a Discrepancy loss which helps to mine out different localization maps. However, optimizing with the Discrepancy loss alone can lead to the trivial solution of splitting the original discriminative region. In order to avoid such phenomena, we further add an Intersection loss which tends to merge the splitted regions in order to regularize the splitting effect. By such splitting vs. merging process, we effectively expand the highlighted regions to the integral object range in a principled pipeline.

In summary, our main contributions are listed as follows:

- We propose to expand the highlighted regions generated by the region mining model from a novel perspective of the backward pass.
- We propose a Discrepancy loss which aims to mine out localization maps of different spatial patterns. It leads to a splitting effect of localization maps.

– We propose an Intersection loss which aims to regularize the splitting phenomena caused by the Discrepancy loss. It leads to a merging effect of localization maps.
– Training the segmentation network with the pseudo-masks generated by our splitting vs. merging process, we achieve state-of-the-art results on weakly supervised semantic segmentation on Pascal VOC 2012 segmentation benchmark.

2 Related Works

2.1 Weakly Supervised Semantic Segmentation

In this part we give a brief review about the weakly supervised semantic segmentation with image-level labels and their key contributions to improve the segmentation results. Recent works always rely on the localization maps to generate localization seeds/pseudo-masks as the substitute of the non-existence of pixel-level groundtruth. The first category [13, 25, 6, 8] investigates training the region mining models to generate initial localization maps which could highlight the object range. The methods in this category are closely related to the object localization tasks [24, 27]. The second category investigates post-processing the localization maps to generate refined pseudo-masks close to the target object regions. The post-processing techniques usually rely on the low-level similarity cues to compensate the seed incompleteness caused by the high-level feature discrimination. Affinitynet [1] proposes to train a network to predict the inter-patch similarity and apply random walk post-processing technique on the localization seeds. The third category investigates a training pipeline for segmentation model which is more stable to the inaccurate/incomplete localization seeds. SEC [10] proposes a pipeline which incorporates expansion loss, CRF constrains loss to the original segmentation loss. Similarly [7] dynamically grows the initial incomplete discriminative seeds into the larger object regions in the training process. The fourth category investigates utilizing additional easily obtained sources such as web sources into weakly supervised semantic segmentation tasks. Web images, which are easily collected by indexing the category names, always possess dominant foreground and clear background regions. Thus the pseudo-masks of web images could be easily estimated by segmentation techniques such as co-segmentation [19] or saliency detection [22]. Consequently the web images could be utilized to compensate the inaccurate localization seeds. Our proposed methods belong to the first category, which focuses on training the region-mining models to highlight the entire object regions.

2.2 Region Mining

In this section we briefly review the region mining techniques which our methods are closely related to. We refer region mining methods as the approaches to estimate the object regions by training image-classification network using the

image-level labels, such as CAM [27] and Grad-CAM [18]. One of the common drawbacks of the region-mining technique is that the result localization map is usually confined to the most discriminative parts instead of the integral object region. Many works focus on enlarging the localization maps from the most discriminative parts to the integral object regions. Adversarial Erasing [21] is the early work that expands the highlighted region by erasing the most discrimnative image region detected by original region-mining model and then re-train the region-mining model with the erased input images. SeeNet [6] utilizes Conditionally Reversed Linear Units to reverse the signs of the feature maps according to the confident foreground/background region. [21, 6] follow sequential training pipeline, which means they alternate between training region-mining models and suppressing the feature space through multiple iterations. Such multi-step training process is time-consuming and complicated to implement. In order to follow a more simple and delicate pipeline, ACoL [26] and GAIN [13] switch the sequential pipeline into an end-2-end manner, which enclose such erasing/suppressing operations in the training steps. Decoupled-net [25] proposes to extend the regions by increasing the dropout rates of dropout layers, which also encloses feature suppression in the training process. The common inherent idea among [21, 6, 26] is to suppress the feature space to reduce its classification differentiation to force the classification model to highlight larger region to achieve classification purpose. Another solution is to aggregate different localization maps for identifying more accurate object regions. [23] aggregates the localization maps generated by different dilation rates in convolution layers, since larger dilation regions will enlarge the respective field of the classification model to help highlight more object parts. [8] aggregates the localization maps generated by different epochs, since the classification network will continuously focus on different object parts at different epochs during training process.

2.3 Co-training

Co-training is a technique which has been initially proposed for multi-view semi-supervised learning. It has been applied in unsupervised domain adaptation tasks [17]. In general, it aims to generate two classifiers with different parameter weights to perform classification from diverse views. Here, we are inspired to generate two diverse localization maps, both of which could be used for the same goal of classification. We assume that different parts of object regions could achieve the classification task. Thus, forcing the region-mining models to mine regions of different patterns could highlight the regions more than the most discriminative parts.

3 Approach

In this section, first we briefly revisit CAM [27], which is one of the most widely used region mining approaches based on classification model. Next we introduce our region mining approach with the proposed Discrepancy loss and Intersection

loss, which is a Splitting vs. Merging process to expand the highlighted region of the localization maps. Finally we normalize the resultant localization maps and generate pseudo-annotations to train the final segmentation model.

3.1 Revisiting CAM

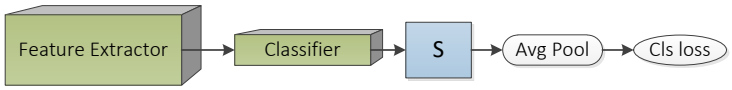

Fig. 1. The brief revisiting of CAM [27] approach. It is a classification model with average pooling step to aggregate the patch-level score map into classification score. The resultant localization map S is obtained by training the model with classification loss.

We briefly review CAM [27], which is one of the most widely used region mining techniques and serves as the basis model for our pipeline. It utilizes global average pooling to aggregate the pixel-level prediction to the image-level score. For simplicity, we introduce an equivalent variance of CAM utilized in ACol [26], which is a more delicate and simple formulation than the original definition [27]. The structure of CAM is illustrated in Fig. 1. The model is sequentially composed of fully convolutional feature extractor and patch-level classifier, which outputs feature $X \in \mathbb{R}^{W \times H \times D}$ and patch-level score map $S \in \mathbb{R}^{W \times H \times C}$, respectively. H and W denote the height and width in the spatial dimension. D denotes the feature dimension and C denotes the number of the classes to classify. Spatial Average pooling is applied on S to aggregate the patch-level score into the image-level prediction score $s \in \mathbb{R}^C$. The whole network is finetuned by calculating and back-propagating the image-classification loss $L_{cls}(s)$. For the multi-label case we utilize MultiLabel Soft Margin Loss, which is a multi-label one-versus-all loss based on max-entropy. Score map S is the resultant class-specific localization map which highlights the corresponding region for each individual class. [26] has proved theoretically that such variance is equivalent to the original CAM [27] and can directly generate the localization map during the forward pass, instead of a separate post-processing step for the map generation.

3.2 Splitting vs. Merging

One of the limitations of region-mining approaches, including but are not limited to CAM [27], is that it is likely to only highlight the most discriminative parts instead of the integral object region. In this section, we propose to alleviate this problem by our Splitting vs. Merging pipeline.

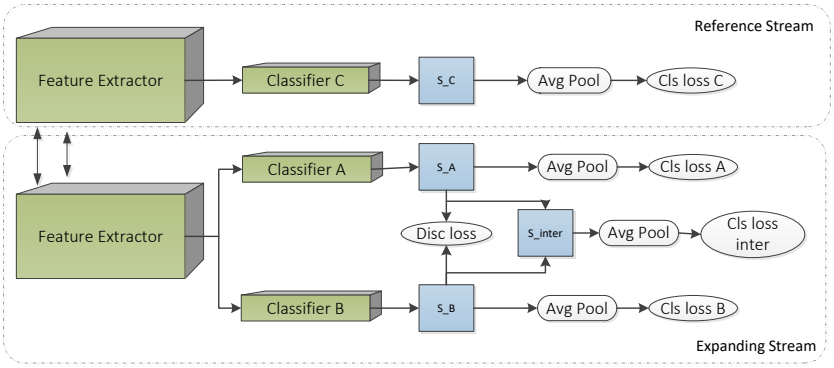

Fig. 2. The structure of our region-mining method. Our structure mainly consists of Reference Stream and Expanding Stream. Expanding Stream aims to expand the highlighted region of the localization maps. It follows an Splitting vs. Merging process, which is achieved by our Discrepancy loss (Disc loss) and Intersection loss (Cls loss inter). Discrepancy loss leads to a splitting effect on the localization maps while the Intersection loss leads to a merging effect. The combination of both losses helps to expand the highlighted regions in the union map output by Expansion Stream.

The structure of our model is illustrated in Fig. 2. Our structure is mainly composed of two streams: Reference Stream and Expanding Stream. The Reference Stream is the original CAM structure which makes sure that the most discriminative region is always mined out. The Expanding Stream aims to expand the localization maps to a larger object scale by mining out localization maps of different spatial patterns. Our main contributions lie in the Expanding Stream, which contains the Splitting vs. Merging strategy formed by our proposed Discrepancy Loss and Intersection Loss.

Our Expanding Stream has similar structure with CAM but with two patch-level classifiers. For clear notation, we denote the classifiers as Classifier A and Classifier B, whose output maps are denoted as S_A and S_B, respectively. Same as CAM, S_A (*resp.* S_B) is aggregated by average pooling to generate classification prediction score s_A (*resp.* s_B) and the corresponding classification loss is denoted as $\mathcal{L}_{cls}(s_A)$ (*resp.* $\mathcal{L}_{cls}(s_B)$).

Besides the classification loss, in order to enforce S_A and S_B to have difference spatial patterns, we propose a Discrepancy loss to regularize S_A and S_B. The Discrepancy loss is depicted as *Disc loss* in Fig. 2. The Discrepancy loss is defined as:

$$\mathcal{L}_{disc} = -\frac{1}{HWC}\sum_{i,j,c}\|z^A_{ijc} - z^B_{ijc}\| \tag{1}$$

where

$$z^A_{ijc} = \frac{e^{s^A_{ijc}}}{\sum_{i,j} e^{s^A_{ijc}}}. \tag{2}$$

s^A_{ijc} is the gird element value of the map S_A, where i, j index the spatial position and c is the class channel index. z^B is calculated following the same spatial normalization from S_B.

Discrepancy loss $\mathcal{L}_{disc}$ could effectively generate two maps with distinct spatial patterns. However, the optimization process is likely to fall into a trivial solution of splitting the original discriminative region, if only using the classification loss and the Discrepancy loss $\mathcal{L}_{disc}$. To avoid such trivial solution, we add an Intersection loss to regularize the optimization. The Intersection loss is denoted as *Cls loss inter* in Fig. 2. Our Intersection loss is defined as follows: we calculate S_{inter} as the element-wise minimum value between S_A and S_B. Average pooling operation is applied on S_{inter} to get the image classification score s_{inter}. The Intersection loss is calculated as the classification loss on s_{inter}, which is denoted as $\mathcal{L}_{cls}(s_{inter})$. By adding Intersection loss $\mathcal{L}_{cls}(s_{inter})$, we force the input maps of the Discrepancy loss (*i.e.* , S_A and S_B) to have large overlapping high-lighted area, which results in a merging effect of the localization maps. Optimizing with both the Discrepancy and the Intersection losses, we follow a Splitting vs. Merging pipeline which forces the Expansion Stream to mine out larger highlighted regions.

Same as CAM, the Reference Stream is optimized by only the classification loss denoted as $\mathcal{L}_{cls}(s_C)$.

The final optimization objective is formulated as

$$\begin{aligned} f^*_T = & arg\,min_{\mathbf{G},\mathbf{w}^a,\mathbf{w}^b,\mathbf{w}^c}[\mathcal{L}_{cls}(s_A) + \mathcal{L}_{cls}(s_B) + \mathcal{L}_{cls}(s_C) + \mathcal{L}_{cls}(s_{inter})] \\ & + \beta * arg\,min_{\mathbf{w}^a,\mathbf{w}^b}\mathcal{L}_{disc} \end{aligned} \tag{3}$$

where $\mathbf{w}^a$, $\mathbf{w}^b$ and $\mathbf{w}^c$ denote the parameters of Classifiers A, B and C, respectively. $\mathbf{G}$ denotes the parameters of the feature extractors, including the feature extractors of Reference stream and Expanding stream. β is the weight parameter of the Discrepancy loss. f^*_T is the resultant optimal model parameters.

3.3 Mask Generation

In this section, we introduce how to normalize the localization maps to (0,1) scale and how to generate the pseudo-annotations for training the segmentation models.

For each localization map S (*i.e.* , S_A, S_B and S_C), we pass the map through a RELU operation and perform min-max normalization for each class channel to obtain the normalized map M (*i.e.* ,M_A, M_B and M_C). The union of different localization maps is calculated as the element-wise maximum map between the normalized maps. We denote the union operation as U. For example, $U(S_A, S_B) \in (0, 1)$ denotes the element-wise maximum result over M_A and M_B.

We utilize denseCRF [11] post-processing approach to estimate the hard annotations from normalized localization maps. The unary term of DenseCRF for each foreground class is the normalized localization maps M (*i.e.* M_A, M_B

and $U(S_A, S_B)$, etc.). Since the normalized map only indicates the probability for each foreground class, we need to estimate the probability of the background class. Similar to [1]. The background probability M_{bg} is calculated as $M_{bg} = (1 - M_{fore})^{\alpha}$, where M_{fore} is the foreground probability and α is the parameter to decide the weight of the background class. We utilize the normalized saliency score $M_{sal} \in (0, 1)$ and the normalized localization map M to calculate M_{fore} as $M_{fore}(i, j) = \max(\max_c M_{ijc}, M^{ij}_{sal})$, which is a channel-wise maximum operation on M followed by an element-wise maximum operation with M_{sal}.

M_{bg} and M are concatenated as the unary term of the denseCRF to generate the hard pseudo-annotations for the training images. Then, we train the segmentation models using the pseudo-annotations. In the testing stage, we directly apply the segmentation model on the validation/testing images to predict the segmentation masks.

4 Experiments

4.1 Datasets and Implementation Details

We perform experiments on the PASCAL VOC 2012 datasets [3] which contains 21 semantic classes in total. Following the common practice, we augment the dataset to 10582 training images with [4] datasets. We report the segmentation results on 1449 validation images (*val*) and 1456 testing images (*test*) using mean intersection-over-union (mIoU) as the segmentation criteria.

Our region-mining model utilizes vgg-16 model as the backbone. We remove the original classifier and the last pooling layer. The feature extractor is initialized with the Imagenet-pretrained weights. The feature extractors of the two streams share weights of the first two blocks. Each of the Classifiers is sequentially composed of a convolutional layer with 512 output channels and a convolutional layer with C output channels, both of the layers are with 1×1 kernel size. A relu layer is added between the two convolutional layers. The training process lasts for 20 epochs. The learning rate is set as 0.01 for the feature extractor and 0.1 for the classifiers. The training images are augmented with random cropping and random flipping and are resized to the size 224×224.

In our experiments we utilize the saliency model in PoolNet [15] using the resnet50 backbone w/o edge model.

For the segmentation model, we utilize the Deeplab-v2 like model in [20] which is based on vgg-16 or resnet50 model. It is similar to Deeplab-v2 structure, but with a global average pooling stream. The input image is of the size 320×320. The initial learning rate is set as 16e-4 and are diminished by rate 0.1 after 8 epoches. We utilize multi-scale merging technique in the reference stage following the common practice. The final segmentation output is post-processed by denseCRF [11] methods.

4.2 Ablation Study

Properties of Mining Region In this section we perform detailed analysis on our generated localization maps to show the effect of our proposed Splitting

Table 1. The Evaluation of the splitting effect of Discrepancy loss. With the increase of the Discrepancy loss weight β, the similarity score between the two maps regularized by Discrepancy loss constantly decreases, which shows that the Discrepancy loss helps to generate two score maps with different spatial patterns.

β	0	10	20	50
similarity score	93.54	78.88	77.64	64.31

vs. Merging pipeline. Unlike the previous work [25] which needs to evaluate the hard-annotations transferred from the soft localization maps, we aim to directly evaluate the properties of the soft localization maps in a more elegant way which neglects the influence of other post-processing parameters such as the hard threshold. Thus we propose three evaluation criteria for localization map evaluation: $Soft_{overlap}$, $Soft_{pre}$ and $Soft_{rec}$.

Given the normalized heatmap $M \in \mathbb{R}^{W\times H\times C} \in [0,1]$ and one-hot segmentation binary ground-truth $G \in \mathbb{R}^{W\times H\times C} \in \{0,1\}$ of one image, the overlap score $Soft_{overlap}$ for class c is defined as

$$Soft_{overlap} = \frac{\sum_{i,j} \min(M_{ijc}, G_{ijc})}{\sum_{i,j} \max(M_{ijc}, G_{ijc})}, \tag{4}$$

the precision score $Soft_{pre}$ is defined as

$$Soft_{pre} = \frac{\sum_{i,j} \min(M_{ijc}, G_{ijc})}{\sum_{i,j} M_{ijc}}, \tag{5}$$

the recall score $Soft_{rec}$ is defined as

$$Soft_{rec} = \frac{\sum_{i,j} \min(M_{ijc}, G_{ijc})}{\sum_{i,j} G_{ijc}}. \tag{6}$$

The $Soft_{pre}$ score indicates whether the highlighted region is located within the groundtruth object region. The $Soft_{rec}$ score indicates whether the range of the target object is covered by the highlighted regions. The $Soft_{overlap}$ score is the overall criteria to evaluate the quality of the localization maps. We utilize the mean score over all foreground classes (excluding the background), which are denoted as $mSoft_{overlap}$, $mSoft_{pre}$ and $mSoft_{rec}$ as our final criteria.

First, we show that our Discrepancy loss helps to generate two localization maps with different spatial patterns. We calculate the similarity score between the input maps of the Discrepancy loss, which are S_A and S_B. To calculate the similarity of the maps we rely on the $mSoft_{overlap}$ defined as Eq. 4 with the input maps replaced by M_A and M_B. The results are shown in Table 1. It shows that with the increase of the Discrepancy loss weight β, the similarity score between the two maps constantly decreases, which shows that the Discrepancy loss helps to generate two localization maps with different spatial patterns. We also provide visualization results of the input maps (*i.e.* S_A and S_B) with different weight of the Discrepancy loss in Fig. 3. It clearly shows that the localization maps regularized by larger weight of Discrepancy loss are more visually different.

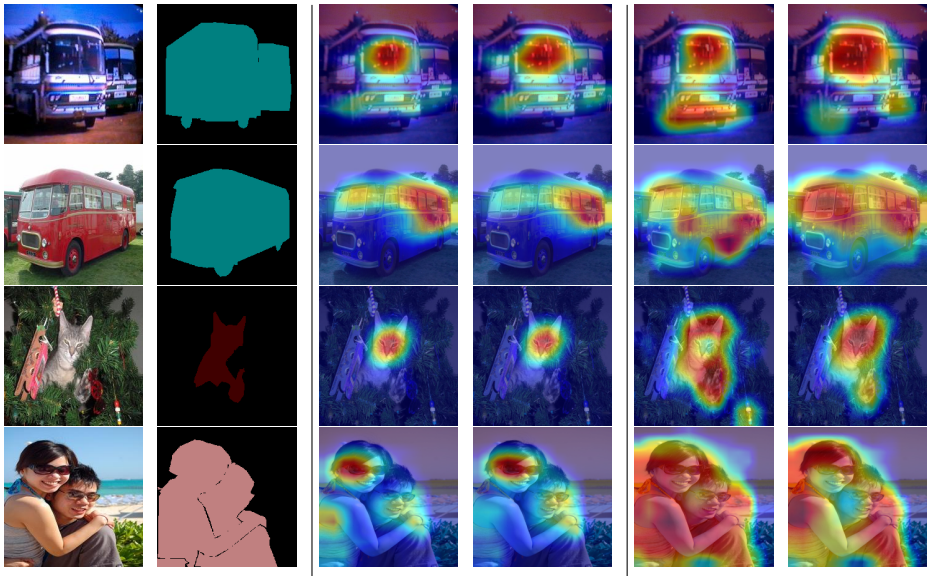

Fig. 3. Visualization results of the input localization maps (S_A and S_B) of the Discrepancy loss with weights $\beta = 10$ and $\beta = 50$. It shows that with the larger weight of Discrepancy loss, the spatial patterns of the input localization maps are more visually different.

Next we investigate the expansion effect of the Discrepancy loss over the union maps. We generate map $U(S_A, S_B)$, which is the union map of the input maps of the Discrepancy loss, and evaluate its variance with the increase of Discrepancy loss weight. The results are listed in the lower block of Table 2. It shows that with the increase of Discrepancy loss weight (β), the recall score $mSoft_{rec}$ constantly increases, which means it is likely to cover more of the target objects. The precision score $mSoft_{pre}$ constantly decreases, which shows that it is more likely to leak out of the boundary of target object regions. It shows that larger Discrepancy loss helps to expand the highlighted regions of the localization maps. For more intuitive understanding of the expansion effect caused by the Discrepancy loss, we provide the visual examples of the union maps $U(S_A, S_B)$ in Fig. 4. It shows that with the increase of the weight of the Discrepancy loss the expansion effect of the union map becomes more obvious.

Third we evaluate the effect of the Intersection loss. The results are listed in Table 2. The upper block lists the results without Intersection loss while the lower block lists the results with intersection loss. We observe that in general the Discrepancy loss helps expand the target region whether with or without the Intersection loss. However, the expansion effect without Intersection loss is not stable or obvious enough, especially under large Discrepancy loss weight. One reasonable explanation is that if the Discrepancy loss weight is too large, the

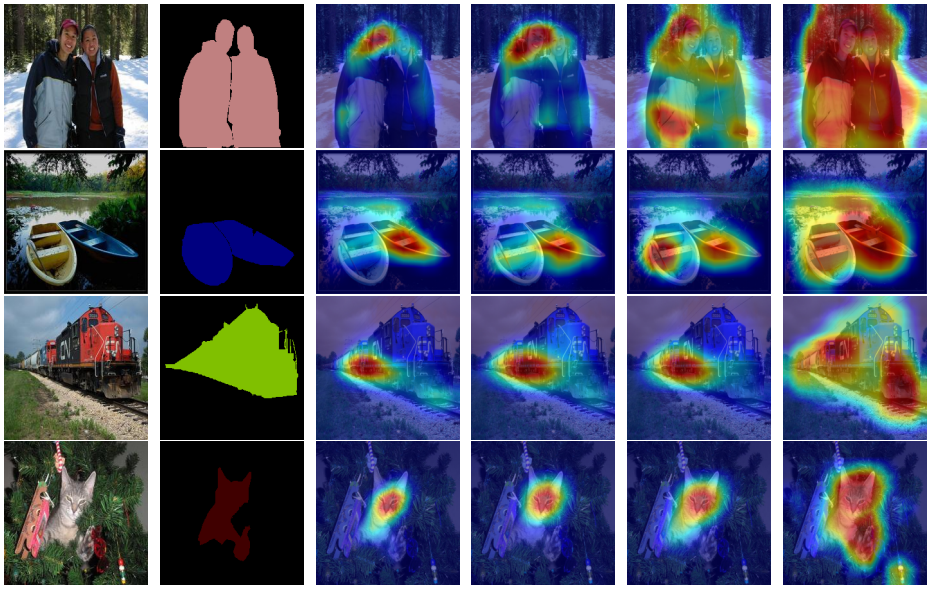

Fig. 4. The visual examples of the union maps $U(S_A,S_B)$. It shows that with the increase of the weight of the Discrepancy loss (β) the expansion effect of the union map becomes more obvious.

Table 2. Evaluation of the union maps of the input maps of the Discrepancy loss on the *val* images. It shows that with the increase of the weight of the Discrepancy loss (β), the recall $mSoft_{rec}$ increases while the precision $mSoft_{pre}$ decreases, which shows the expansion effect of the Discrepancy loss. It also shows that adding the Intersection loss (Inter loss) helps the expansion effect more stable and obvious.

β	Inter loss	0	10	20	50
$mSoft_{pre}$	-	51.65	45.86	45.25	45.5
$mSoft_{rec}$	-	28.5	32.96	26.48	34.00
$mSoft_{overlap}$	-	22.6	24.0	20.0	24.5
$mSoft_{pre}$	✓	52.01	50.03	48.92	36.82
$mSoft_{rec}$	✓	26.31	30.77	32.24	49.15
$mSoft_{overlap}$	✓	21.22	23.7	24.17	27.35

optimization with only the Discrepancy loss may likely to be stuck into a tricky solution of simply splitting the original discriminative region. Thus we utilize Intersection loss to regularize the optimization with Discrepancy loss for more stable effect of the region expansion. We visually compare the localization maps generated with/without the Intersection loss in Fig. 5. It shows that without the Intersection loss, the large Discrepancy loss weight mainly leads to splitting a single discriminative region instead of having obvious expansion effect.

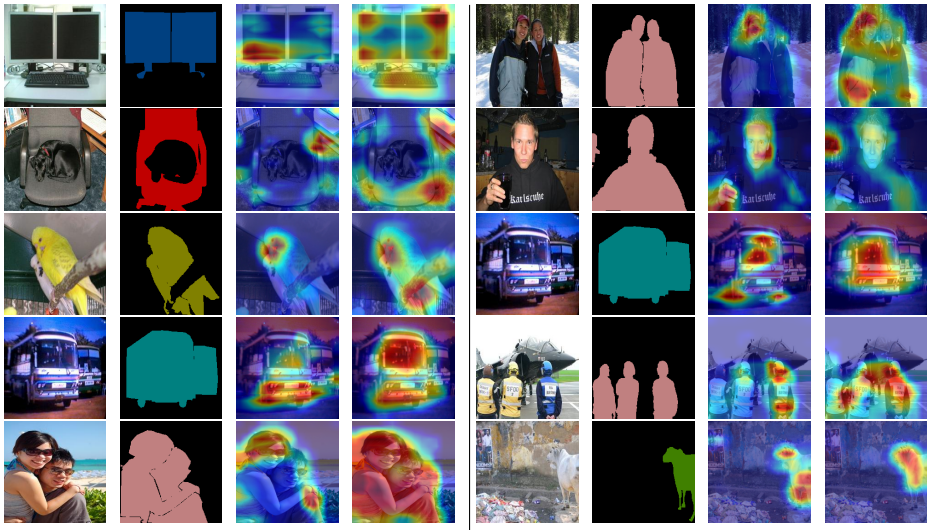

Fig. 5. The visual comparison of the union localization maps generated with/without the Intersection loss. It shows that without the Intersection loss, the large Discrepancy loss mainly leads to split a single discriminative region instead of having obvious expansion effect.

Table 3. $mSoft_{overlap}$ of the union of different localization maps on the *val* images. It shows that the localization maps (S_A,S_B and S_C) are complimentary to each other and the union of all the three maps achives the highest $mSoft_{overlap}$ score.

β	0	10	20	50
S_c	22.25	23.65	23.56	24.36
S_A	20.96	22.48	22.80	24.47
S_B	20.93	22.71	23.33	24.96
$U(S_A, S_B)$	21.22	23.70	24.17	27.35
$U(S_A, S_B, S_C)$	24.57	26.58	27.07	28.57

Fourth we show that our pipeline outputs complementary localization maps. The results are listed in Table 3. We report the $mSoft_{overlap}$ score over different localization maps, such as S_A, S_B and $U(S_A, S_B)$. It shows that the union of maps S_A and S_B has higher $mSoft_{overlap}$ score than each single map alone. The union of all the three maps S_A,S_B and S_C shows the highest $mSoft_{overlap}$ score. Thus we utilize the union of the three maps for pseudo-mask generation.

Finally, we generate pseudo-annotations following the practice in Sec. 3.3. We set the background parameter α=2. We utilize the traditional intersection-over-union criteria $mIoU$ on hard masks to evaluate our generated annotations. The results are listed in Table 4, which show that we achieve better quality of

Table 4. $mIoU$ of the generated pseudo-annotations on the *train* images with different localization maps. Compared with other two methods, our maps lead to pseudo-annotations of higher quality.

	Ours, β=0	Ours, β=10	Ours, β=20	Ours, β=50	SeeNet [6]	OAA$^+$ [8]
$mIoU$	59.65	60.92	**61.27**	57.22	54.47	57.96

Table 5. Segmentation results on *val* and *test* images using $vgg16$ segmentation backbone. We list the condition whether additional training data (web data) are added and whether supervised saliency (S-Sal) are utilized.

Method	web data	S-Sal	*val*	*test*
SEC [10]	-	-	50.7	51.7
Two-phase [9]	-	-	53.1	53.8
Decou-Net[25](vgg16)	-	-	55.4	56.4
Affinity [1] Deeplab	-	-	58.4	60.5
STC [22]	✓	✓	49.3	51.2
Crawled-Video [5]	✓	-	58.1	58.7
BoostTrap [20](vgg16)	✓	-	58.8	60.2
DCSP-VGG16 [2]	-	✓	58.6	59.2
AE-PSL [21]	-	✓	55.0	55.7
DSRG (vgg16) [7]	-	✓	59.0	60.4
FickleNet (vgg16)[12]	-	✓	61.2	61.9
MDC [23]	-	✓	60.4	60.8
GAIN [13]	-	✓	55.3	56.8
SeeNet (vgg16) [6]	-	✓	61.1	60.7
OAA$^+$ (vgg16) [8]	-	✓	63.1	62.7
Ours (vgg16)	-	✓	63.7	64.5

the pseudo-masks over the case without Discrepancy loss. We utilize the pseudo-masks of the highest quality (β=20) to train the final segmentation models. We further generate annotations using the localization maps provided by SeeNet [6] and OAA$^+$ [8] using our methods and report the $mIoU$ in Table 4. We observe that our mask quality outperforms that of SeeNet and OAA$^+$.

4.3 Segmentation Results

Finally we train the segmentation networks using the pseudo-masks generated by our localization maps and report our segmentation results in Table 5 and Table 6. For clear and fair comparison, we list the extra information knowledge that may improve the segmentation results, such as whether extra images are added into training images or whether supervised saliency methods are utilized. It shows that we achieve the state-of-the-art weakly supervised semantic segmentation results. We list the segmentation results of each category in Table 7. We also generate pseudo-annotations by training Affinitynet [1] on localization maps, which does not enclose supervised saliency. The $mIoU$ of segmentation results

Table 6. Our Segmentation results on *val* and *test* images using *resnet* segmentation backbone. We list the condition whether additional training data (web data) are added and whether supervised saliency (S-Sal) are utilized.

Method	web data	S-Sal	*val*	*test*
Decou-Net[25](resnet101)	-	-	58.2	60.1
Affinity [1] Resnet-34	-	-	61.7	63.7
Co-segmentation [19]	✓	-	56.4	56.9
BoostTrap [20](resnet50)	✓	-	63.0	63.9
DCSP-ResNet-101 [2]	-	✓	60.8	61.9
DSRG (resnet101) [7]	-	✓	61.4	63.2
FickleNet (resnet101)[12]	-	✓	64.9	65.3
SeeNet (resnet101)[6]	-	✓	63.1	62.8
OAA$^+$ (resnet101)[8]	-	✓	65.2	66.4
Ours (resnet50)	-	✓	**66.6**	**66.7**

Table 7. Our Segmentation results for each class on *val* and *test* images. We utilize both vgg16 and resnet50 as the base model of the segmentation model.

	bkg	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mIoU
vgg16 *val*	89.9	84.6	36.0	79.7	57.7	65.5	81.2	75.5	82.3	23.2	65.5	31.5	78.0	72.1	72.4	74.8	36.8	75.5	31.6	70.7	52.3	63.7
vgg16 *test*	90.4	78.7	34.5	82.2	50.7	63.8	76.5	74.1	80.1	24.6	69.7	35.4	77.7	77.1	78.3	74.9	46.6	78.5	34.8	70.5	54.4	64.5
resnet50 *val*	90.4	85.6	38.9	78.9	62.0	73.4	83.7	74.3	82.9	25.8	77.8	30.1	81.1	79.3	76.1	73.9	38.6	85.0	32.7	72.8	55.7	66.6
resnet50 *test*	90.7	85.9	37.3	82.5	50.5	64.8	83.1	77.6	82.8	28.4	76.8	34.6	81.2	82.9	80.5	73.6	43.9	85.7	32.0	71.7	55.2	66.7

with resnet50 model on *val*/*test* dataset is 61.7/62.7, which is competitive with other state-of-the-arts without supervised saliency.

5 Conclusion

In this paper, our goal is to propose a region-mining method in order to generate pseudo-masks for weakly supervised semantic segmentation. We aim to train a region mining model which identifies the integral object regions instead of only the most discriminative parts. In order to achieve this goal, we tackle the problem from a novel perspective of the backward optimization pass. We propose a Splitting vs. Merging pipeline, which is mainly composed of a Discrepancy loss and an Intersection loss. With the pseudo annotations generated from our region mining models, we achieve the state-of-the art weakly supervised segmentation results on the PASCAL VOC12 benchmark.

Acknowlegements This research was mainly carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. The ROSE Lab is supported by the National Research Foundation, Singapore, and the Infocomm Media Development Authority, Singapore. This work is also partly supported by the National Research Foundation Singapore under its AI Singapore Programme (Award Number: AISG-RP-2018-003), the MOE Tier-1 research grant:RG126/17 (S) and RG28/18 (S) and the Monash University FIT Start-up Grant.

References

1. Ahn, J., Kwak, S.: Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In: CVPR (2018)
2. Chaudhry, A., Dokania, P.K., Torr, P.H.: Discovering class-specific pixels for weakly-supervised semantic segmentation. In: BMVC (2017)
3. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV **88**(2) (2010)
4. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: ICCV (2011)
5. Hong, S., Yeo, D., Kwak, S., Lee, H., Han, B.: Weakly supervised semantic segmentation using web-crawled videos (2017)
6. Hou, Q., Jiang, P., Wei, Y., Cheng, M.: Self-erasing network for integral object attention. In: NeurIPS (2018)
7. Huang, Z., Wang, X., Wang, J., Liu, W., Wang, J.: Weakly-supervised semantic segmentation network with deep seeded region growing. In: CVPR (2018)
8. Jiang, P., Hou, Q., Cao, Y., Cheng, M., Wei, Y., Xiong, H.: Integral object mining via online attention accumulation. In: ICCV (2019)
9. Kim, D., Yoo, D., Kweon, I.S., et al.: Two-phase learning for weakly supervised object localization. In: ICCV (2017)
10. Kolesnikov, A., Lampert, C.H.: Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In: ECCV (2016)
11. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: NIPS (2011)
12. Lee, J., Kim, E., Lee, S., Lee, J., Yoon, S.: Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In: CVPR (2019)
13. Li, K., Wu, Z., Peng, K., Ernst, J., Fu, Y.: Tell me where to look: Guided attention inference network. In: CVPR (2018)
14. Lin, G., Milan, A., Shen, C., Reid, I.: RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In: CVPR (2016)
15. Liu, J., Hou, Q., Cheng, M., Feng, J., Jiang, J.: A simple pooling-based design for real-time salient object detection. arXiv preprint arXiv:1904.09569 (2019)
16. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
17. Luo, Y., Zheng, L., Guan, T., Yu, J., Yang, Y.: Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2507–2516 (2019)
18. Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., Batra, D.: Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. In: ICCV (2017)
19. Shen, T., Lin, G., Liu, L., Shen, C., Reid, I.: Weakly supervised semantic segmentation based on co-segmentation (2017)
20. Shen, T., Lin, G., Shen, C., Reid, I.: Bootstrapping the performance of webly supervised semantic segmentation. In: CVPR (2018)
21. Wei, Y., Feng, J., Liang, X., Cheng, M., Zhao, Y., Yan, S.: Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In: CVPR (2017)
22. Wei, Y., Liang, X., Chen, Y., Shen, X., Cheng, M., Zhao, Y., Yan, S.: Stc: A simple to complex framework for weakly-supervised semantic segmentation. TPAMI **39**(11), 2314–2320 (2017)

23. 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: CVPR (2018)
24. Zhang, J., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-down neural attention by excitation backprop. In: ECCV (2016)
25. Zhang, T., Lin, G., Cai, J., Shen, T., Shen, C., Kot, A.C.: Decoupled spatial neural attention for weakly supervised semantic segmentation. TMM (2019)
26. Zhang, X., Wei, Y., Feng, J., Yang, Y., Huang, T.S.: Adversarial complementary learning for weakly supervised object localization. In: CVPR (2018)
27. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR (2016)