Reducing Information Bottleneck for Weakly Supervised Semantic Segmentation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Weakly supervised semantic segmentation produces pixel-level localization from class labels; however, a classifier trained on such labels is likely to focus on a small discriminative region of the target object. We interpret this phenomenon using the information bottleneck principle: the final layer of a deep neural network, activated by the sigmoid or softmax activation functions, causes an information bottleneck, and as a result, only a subset of the task-relevant information is passed on to the output. We first support this argument through a simulated toy experiment and then propose a method to reduce the information bottleneck by removing the last activation function. In addition, we introduce a new pooling method that further encourages the transmission of information from non-discriminative regions to the classification. Our experimental evaluations demonstrate that this simple modification significantly improves the quality of localization maps on both the PASCAL VOC 2012 and MS COCO 2014 datasets, exhibiting a new state-of-the-art performance for weakly supervised semantic segmentation.

1 Introduction

Semantic segmentation is the task of recognizing objects in an image using pixel-level allocation of a semantic label. The development of deep neural networks (DNNs) has led to significant advances in semantic segmentation [8, 23]. Training a DNN for semantic segmentation requires a dataset containing a large number of images annotated with pixel-level labels. However, preparing such a dataset requires considerable effort; for example, producing a pixel-level annotation for a single image in the Cityscapes dataset [11] takes more than 90 minutes. This high dependence on pixel-level labels can be alleviated by weakly supervised learning [2, 30, 32, 58].

The objective of weakly supervised semantic segmentation is to train a segmentation network with weak annotations, which provide less information about the location of a target object than pixel-level labels, but are cheaper to obtain. Weak supervision takes the form of scribbles [48], bounding boxes [28, 32, 46], or image-level class labels [2, 3, 30]. In this study, we focus on image-level class labels, because they are the cheapest and most popular option of weak supervision. Most methods that use class labels [2, 6, 30, 31, 51] generate pseudo ground truths for training a segmentation network using localization (attribution) maps obtained from a trained classifier, such as a CAM [62] or a Grad-CAM [44]. However, these maps identify only small regions of a target object that are discriminative for the classification [2, 4, 30] and do not identify the entire region occupied by the object, making the attribution maps unsuitable for training a semantic segmentation network. We interpret this phenomenon using the *information bottleneck* principle [42, 45, 49, 50].

The information bottleneck theory analyzes the information flow through sequential DNN layers: information regarding the input is compressed as much as possible as it passes through the layers of a DNN, while preserving as much of the task-relevant information as possible. This is advantageous

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

for obtaining optimal representations for classification [1, 14] but is disadvantageous when applying the attribution maps from the resulting classifier to weakly supervised semantic segmentation. The information bottleneck prevents the non-discriminative information of the target object from being considered in the classification logit, and thus, the attribution maps focus on only the small discriminative regions of the target object.

We argue that the information bottleneck becomes prominent in the final layer of the DNN due to the use of the double-sided saturating activation function therein (*e.g.,* sigmoid, softmax). We propose a method to reduce this information bottleneck in the final layer of the DNN by retraining the DNN without the last activation function. Additionally, we introduce a new pooling method that allows more information embedded in non-discriminative features, rather than discriminative features, to be processed in the last layer of a DNN. As a result, the attribution maps of the classifier obtained by our method contain more information on the target object.

The main contributions of this study are summarized as follows. First, we highlight that the information bottleneck occurs mostly in the final layer of the DNN, which causes the attribution maps obtained from a trained classifier to restrict their focus to small discriminative regions of the target object. Second, we propose a method to reduce this information bottleneck by simply modifying the existing training scheme. Third, our method significantly improves the quality of the localization maps obtained from a trained classifier, exhibiting a new state-of-the-art performance on the PASCAL VOC 2012 and MS COCO 2014 datasets for weakly supervised semantic segmentation.

2 Preliminaries

2.1 Information Bottleneck

Given two random variables X and Y, the mutual information $\mathcal{I}(X;Y)$ quantifies the mutual dependence between the two variables. Data processing inequality (DPI) [12] infers that any three variables X, Y, and Z that form a Markov Chain $X \to Y \to Z$ satisfy $\mathcal{I}(X;Y) \geq \mathcal{I}(X;Z)$. Each layer in a DNN processes the input only from the previous layer, which means that the DNN layers form a Markov chain. Therefore, the information flow through these layers can be represented using DPI. More specifically, when an $L-$layered DNN generates an output $\hat{Y}$ from a given input X through intermediate features T_l $(1 \leq l \leq L)$, it forms a Markov Chain $X \to T_1 \to \cdots \to T_L \to \hat{Y}$, and the corresponding DPI chain can be expressed as follows:

$$\mathcal{I}(X;T_1) \geq \mathcal{I}(X;T_2) \geq \cdots \geq \mathcal{I}(X;T_{L-1}) \geq \mathcal{I}(X;T_L) \geq \mathcal{I}(X;\hat{Y}). \tag{1}$$

This implies that the information regarding the input X is compressed as it passes through the layers of the DNN.

Training a classification network can be interpreted as extracting maximally compressed features of the input that preserve as much information as possible for classification; such features are commonly referred to as minimum sufficient features (*i.e.,* discriminative information). The minimum sufficient features (optimal representations T^*) can be obtained by the *information bottleneck* trade-off between the mutual information of X and T (compression), and that of T and Y (classification) [14, 50]. In other words, $T^* = \text{argmin}_T \ \mathcal{I}(X;T) - \beta\mathcal{I}(T;Y)$, where $\beta \geq 0$ is a Lagrange multiplier.

Shwartz-Ziv *et al.* [45] observe a *compression phase* in the process of finding the optimal representation T^*: when observing $\mathcal{I}(X,T_l)$ for a fixed l, $\mathcal{I}(X,T_l)$ steadily increases during the first few epochs, but decreases in the later epochs. Saxe *et al.* [42] argue that the compression phase is mainly observed in DNNs equipped with double-sided saturating non-linearities (*e.g.,* tanh and sigmoid), and is not observed in those equipped with single-sided saturating non-linearities (*e.g.,* ReLU). This implies that DNNs with single-sided saturating non-linearities experience less information bottleneck than those with double-sided saturating non-linearities. This can also be understood in terms of gradient saturation in the double-sided saturating non-linearities: the gradient of those non-linearities with respect to an input above a certain value saturates close to zero [7]. Therefore, features above a certain value will have near-zero gradients during the back-propagation process and be restricted from additionally contributing to the classification.

2.2 Class Activation Mapping

A class activation map (CAM) [62] identifies regions of an image focused by a classifier. The CAM is based on a convolutional neural network with global average pooling (GAP) before its final

classification layer. This is realized by considering the class-specific contribution of each channel of the last feature map to the classification score. Given a classifier parameterized by $\theta = \{\theta_f, w\}$ where $f(\cdot;\theta_f)$ is the feature extractor prior to GAP, and w is the weight of the final classification layer, a CAM of the class c is obtained from an image x as follows:

$$\texttt{CAM}(x;\theta) = \frac{\mathbf{w}_c^\intercal f(x;\theta_f)}{\max \mathbf{w}_c^\intercal f(x;\theta_f)}\,, \tag{2}$$

where max$(\cdot)$ is the maximum value over the spatial locations for normalization.

2.3 Related Work

Weakly Supervised Semantic Segmentation: Weakly supervised semantic segmentation methods with image-level class labels first construct an initial seed by obtaining a high-quality localization map from a trained classifier. Erasure methods [21, 34, 52] prevent a classifier from only focusing on the discriminative parts of objects by feeding the image from which the discriminative regions have been erased to the classifier. Several contexts of a target object can be considered by combining multiple attribution maps obtained from differently dilated convolutions [30, 53] or from the different layers of a DNN [33]. Diverse images of a target class can be utilized by considering cross-image semantic similarities and differences [16, 47]. Zhang *et al.* [58] analyze the causalities among images, contexts, and class labels and propose CONTA to remove the confounding bias in the classification. Because the localization maps obtained by a classification network cannot accurately represent the boundary of a target object, the initial seed obtained by the above methods is refined using subsequent boundary refinement techniques such as PSA [3] and IRN [2].

Information Bottleneck: Tishby *et al.* [50] and Shwartz *et al.* [45] use the information bottleneck theory to analyze the inner workings of a DNN. The concept of the information bottleneck has been employed in many research fields. Dubois *et al.* [14] and Achille *et al.* [1] utilize the information bottleneck to obtain optimal representations from DNNs. DICE [41] is proposed for a model ensemble with the information bottleneck principle: it aims to reduce not only the unnecessary mutual information between features and inputs but also the redundant information shared between features produced by separately trained DNNs. Jeon *et al.* [25] study the disentangled representation learning of a generative model [18] using the information bottleneck principle. Yin *et al.* [56] design a regularization objective based on information theory to deal with the memorization problem in meta-learning. The information bottleneck principle can also be adopted to generate the visual saliency map of a classifier. Zhmoginov *et al.* [61] find important regions for the classifier with the information bottleneck trade-off, and Schulz *et al.* [43] restrict the information flow by adding noise to intermediate feature maps and quantify the amount of information contained in the image region.

3 Proposed Method

Weakly supervised semantic segmentation methods using class labels produce a pixel-level localization map from a classifier using CAM [62] or Grad-CAM [44]; however, such a map identifies only small discriminative regions of the target object. We analyze this phenomenon with the information bottleneck theory in Section 3.1 and propose RIB, a method to address this problem, in Section 3.2. We then explain how we train a segmentation network with localization maps improved through RIB in Section 3.3.

3.1 Motivation

As mentioned in Section 2.1, the DNN layers with double-sided saturating non-linearities have a larger information bottleneck than those with single-sided saturating non-linearities. The intermediate layers of popular DNN architectures (*e.g.,* ResNet [20] and DenseNet [22]) are coupled with the ReLU activation function, which is a single-sided saturating non-linearity. However, the final layer of these networks is activated by a double-sided saturating non-linearity such as sigmoid or softmax, and the class probability p is computed with the final feature map T_L and the final classification layer w, *i.e.,* $p = \texttt{sigmoid}(w^\intercal \text{GAP}(T_L))$. Therefore, the final layer parameterized by w has a significant bottleneck, and the amount of information transmitted from the last feature T_L to the actual classification prediction will be limited.

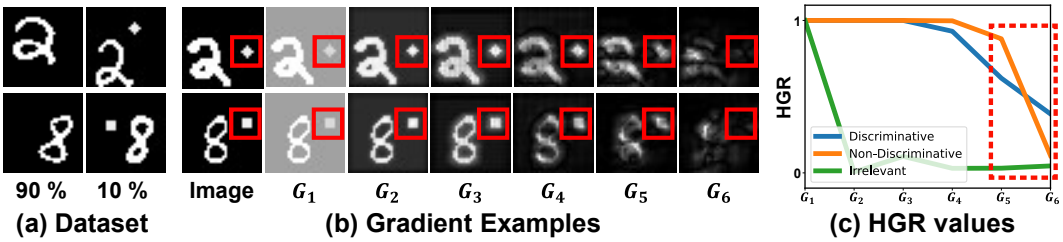

Figure 1: (a) Examples of toy images. (b) Examples of gradient maps G_k. (c) Plot of HGR values of $\mathcal{R}_{\text{D}}$, $\mathcal{R}_{\text{ND}}$, and $\mathcal{R}_{\text{BG}}$ for each layer, averaged over 100 images.

These arguments are analogous to the observations in existing methods. The information plane provided by Saxe *et al.* [42] shows that the compression of information is more noticeable in the final layer than in the other layers. Bae *et al.* [4] observe that although the final feature map of the classifier contains rich information on the target object, the final classification layer filters out most of it; thus, the CAM cannot identify the entire area of the target object. This observation empirically supports the occurrence of the information bottleneck in the final layer of a DNN.

To take a closer look at this phenomenon, we design a toy experiment. We collect images containing the digits '2' or '8' from the MNIST dataset [29]. For only a small subset (10%) of these images, we add a circle (●) and a square (■) to the images containing the digits '2' and '8', respectively, at a random location (see Figure 1(a)). When classifying images into the digits '2' or '8', pixels corresponding to the digit are discriminative regions ($\mathcal{R}_{\text{D}}$), those corresponding to the added circle or square are non-discriminative but class-relevant regions ($\mathcal{R}_{\text{ND}}$), and those corresponding to the background are class-irrelevant regions ($\mathcal{R}_{\text{BG}}$).

We train a neural network with five convolutional layers followed by a final fully connected layer. We obtain the gradient map G_l of each feature T_l with respect to an input image x: $G_l = \nabla_x \sum_{u,v} T_l(u,v)$, where u and v are the spatial and channel indices of the feature T_l, and for the final classification layer ($l = 6$), $G_6 = \nabla_x y^c$. Because this gradient map indicates the extent to which each pixel of the image affects each feature, it can be used to examine how much information is passed from the input image to the feature maps of successive convolution layers.

We present examples of G_l in Figure 1(b). As an input image passes through the convolution layers, the overall amount of gradient with respect to the input decreases, indicating the occurrence of the information bottleneck. Specifically, the gradient of $\mathcal{R}_{\text{BG}}$ decreases early on ($G_1 \rightarrow G_2$), which implies that the task-irrelevant information is rapidly compressed. From G_1 to G_5, the gradient in $\mathcal{R}_{\text{D}}$ or $\mathcal{R}_{\text{ND}}$ gradually decreases. However, the decrease in the amount of gradient is prominent in the final layer ($G_5 \rightarrow G_6$), and in particular, the gradients in $\mathcal{R}_{\text{ND}}$ (red boxes) almost disappear. This supports our argument that there is significant information bottleneck in the final layer of a DNN, while also highlighting that the non-discriminative information in $\mathcal{R}_{\text{ND}}$ is particularly compressed.

We analyze this quantitatively. We define the high gradient ratio (HGR) of region $\mathcal{R}$ as the ratio of pixels that have a gradient above 0.3 to the total pixels in region $\mathcal{R}$. HGR quantifies the amount of transmitted information from region $\mathcal{R}$ of an input image to each feature. The trend in the HGR values of each region for each layer is shown in Figure 1(c). The observed trend is analogous to the above empirical observation, once again supporting that significant information bottleneck for $\mathcal{R}_{\text{ND}}$ occurs in the final layer (the red box).

We argue that the information bottleneck causes the localization map obtained from a trained classifier to focus on small regions of the target object. According to Eq. 2, the CAM only includes information that is processed by the final classification weight w_c. However, because only a subset of the information in the feature is passed through the final layer w_c due to the information bottleneck, leaving out most of the non-discriminative information, CAM cannot identify the non-discriminative regions of the target object. It is undesirable to use such CAMs to train a semantic segmentation network, for which the entire region of the target object should be identified. Therefore, we aim to bridge the gap between classification and localization by reducing the information bottleneck.

3.2 Reducing Information Bottleneck

In Section 3.1, we observed that the information contained in an input image is compressed particularly in the final layer of the DNN, due to the use of the double-sided saturating activation function therein.

Therefore, we propose a method to reduce the information bottleneck of the final layer by simply removing the sigmoid or softmax activation function used in the final layer of the DNN. We focus on a multi-class multi-label classifier, which is the default setting for weakly supervised semantic segmentation. Suppose we are given an input image x and the corresponding one-hot class label $t = [t_1, \cdots, t_{\mathcal{C}}]$, where $t_c \in \{0, 1\}$ $(1 \leq c \leq \mathcal{C})$ is an indicator of a class c, and $\mathcal{C}$ is the set of all classes. While existing methods use the sigmoid binary cross-entropy (BCE) loss ($\mathcal{L}_{\text{BCE}}$) to train a multi-label classifier, our method replaces it with another loss function $\mathcal{L}_{\text{RIB}}$ that does not rely on the final sigmoid activation function:

$$\mathcal{L}_{\text{BCE}} = -\sum_{c=1}^{\mathcal{C}} t_c \log \texttt{sigmoid}(y^c) + (1 - t_c) \log(1 - \texttt{sigmoid}(y^c)), \ \mathcal{L}_{\text{RIB}} = -\sum_{c=1}^{\mathcal{C}} t_c \min(m, y^c),$$

where m is a margin, and y^c is the classification logit of image x.

However, training a classifier with $\mathcal{L}_{\text{RIB}}$ from scratch causes instability in the training because the gradient cannot saturate (please see the Appendix). Therefore, we first train an initial classifier with $\mathcal{L}_{\text{BCE}}$ whose trained weights are denoted by θ_0, and for a given image x, we adapt the weights toward a bottleneck-free model of x. Specifically, we fine-tune the initial model using $\mathcal{L}_{\text{RIB}}$ computed from x and obtain a model parameterized by θ_k $(0 < k \leq K)$, where $\theta_k = \theta_{k-1} - \lambda \nabla_{\theta_{k-1}} \mathcal{L}_{\text{RIB}}$, and K and λ are respectively the total number of iterations and the learning rate for fine-tuning. We name this fine-tuning process RIB. Employing RIB reduces the information bottleneck for x, and we can obtain CAMs that identify more regions of the target object, including non-discriminative regions. We repeat the RIB process for all the training images to obtain the CAMs.

However, the model that is adapted to a given image x can be easily over-fitted to x. Therefore, to further stabilize the RIB process, we construct a batch of size B for RIB by sampling random $B - 1$ samples other than x at each RIB iteration. Note that for each iteration, $B - 1$ samples are randomly selected, while x is fixed.

Effectiveness of RIB: We demonstrate the effectiveness of RIB by applying it to the same classifier as that used for the toy experiments described in Section 3.1. Figure 2 presents (a) examples of G_6 and (b) the HGR values for $\mathcal{R}_{\text{D}}$, $\mathcal{R}_{\text{ND}}$, and $\mathcal{R}_{\text{BG}}$ of G_6, which showed the most significant information bottleneck, at each RIB iteration. The HGR values are averaged over 100 images. The HGR values of $\mathcal{R}_{\text{BG}}$ remain fairly constant during the RIB process, while the HGR values of $\mathcal{R}_{\text{D}}$ and $\mathcal{R}_{\text{ND}}$ increase significantly. This indicates that the RIB process can indeed reduce the information bottleneck, thereby ensuring that more information corresponding to both $\mathcal{R}_{\text{D}}$ and $\mathcal{R}_{\text{ND}}$ is processed by the final classification layer.

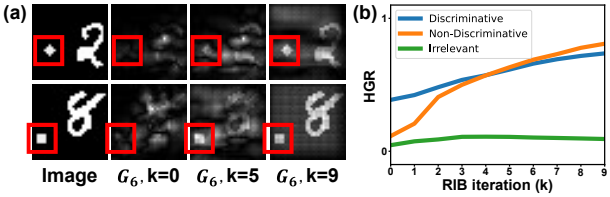

Figure 2: Analysis of G_6 for $\mathcal{R}_{\text{D}}$, $\mathcal{R}_{\text{ND}}$, and $\mathcal{R}_{\text{BG}}$ at each RIB iteration.

Limiting the transmission of information from discriminative regions: Zhang *et al.* [60] showed the relationship between a classification logit y and a CAM, *i.e.,* $y = \text{GAP}(\texttt{CAM})$. This implies that increasing y^c with RIB also increases the pixel values in the CAM. For a CAM to identify a wider area of the target object, it is important to increase the pixel scores of the non-discriminative regions, rather than the discriminative regions. Therefore, we introduce a new pooling method to the RIB process, so that the features that were previously delivering a small amount of information to the classification logit contribute more to the classification.

We propose a global non-discriminative region pooling (GNDRP). Contrary to GAP which aggregates all the values of the spatial location in the feature map T_l, our GNDRP selectively aggregates the values of spatial locations whose CAM scores are below a threshold τ, as follows:

$$\text{GAP}(T_l) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} T_l(u), \quad \text{GNDRP}(T_l) = \frac{1}{|\mathcal{U}_\tau|} \sum_{u \in \mathcal{U}_\tau} T_l(u), \quad \mathcal{U}_\tau = \{u \in \mathcal{U} \,|\, \texttt{CAM}(u) \leq \tau\},$$

where $\mathcal{U}$ is a set of all spatial location indices in T_l.

Obtaining a final localization map: We obtain the final localization map $\mathcal{M}$ by aggregating all the CAMs obtained from the classifier at each RIB iteration k: $\mathcal{M} = \sum_{0 \leq k \leq K} \texttt{CAM}(x; \theta_k)$.

Method	Backbone	PASCAL VOC			MS COCO	
		Seed	CRF	Mask	Seed	Mask
PSA $_{\text{CVPR '18}}$ [3]	WideResNet-38	48.0	-	61.0	-	-
Mixup-CAM $_{\text{BMVC '20}}$ [5]	WideResNet-38	50.1	-	61.9	-	-
Chang *et al.* $_{\text{CVPR '20}}$ [6]	WideResNet-38	50.9	55.3	63.4	-	-
SEAM $_{\text{CVPR '20}}$ [51]	WideResNet-38	55.4	56.8	63.6	25.1†	31.5†
IRN $_{\text{CVPR '19}}$ [2]	ResNet-50	48.8	54.3	66.3	33.5‡	42.9‡
MBMNet $_{\text{ACMMM '20}}$ [37]	ResNet-50	50.2	-	66.8	-	-
BES $_{\text{ECCV '20}}$ [9]	ResNet-50	50.4	-	67.2	-	-
CONTA $_{\text{NeurIPS '20}}$ [58]	ResNet-50	48.8	-	67.9	28.7†	35.2†
RIB (Ours)	ResNet-50	**56.5**	**62.9**	**70.6**	**36.5**‡	**45.6**‡

Table 1: Comparison of the initial seed (Seed), the seed with CRF (CRF), and the pseudo ground truth mask (Mask) on PASCAL VOC and MS COCO training images, in tems of mIoU (%). † denotes the results reported by Zhang *et al.* [58], and ‡ denotes the results obtained by us.

3.3 Weakly Supervised Semantic Segmentation

Because a CAM [62] is obtained from down-sampled intermediate features produced by a classifier, it should be up-sampled to the size of the original image. Therefore, it tends to localize the target object coarsely and cannot represent its exact boundary. Many weakly supervised semantic segmentation methods [5, 6, 30, 37, 51, 58] produce pseudo ground truths by modifying their initial seeds using established seed refinement methods [2, 3, 9, 24, 26]. Similarly, we obtain pseudo ground truths by applying IRN [2], a state-of-the-art seed refinement method, to the coarse map $\mathcal{M}$.

In addition, because an image-level class label is void of any prior regarding the shape of the target object, salient object mask supervision is popularly used in existing methods [21, 30, 35, 55]. Salient object mask supervision can also be applied to our method to refine the pseudo ground truths: when a foreground pixel in a pseudo label is identified as background on this map, or vice versa, we ignore such pixels in the training of the segmentation network.

4 Experiments

4.1 Experimental Setup

Dataset and evaluation metric: We evaluated our method quantitatively and qualitatively by conducting experiments on the PASCAL VOC 2012 [15] and the MS COCO 2014 [36] datasets. Following the common practice in weakly supervised semantic segmentation [2, 3, 30, 58], we used the PASCAL VOC dataset, which is augmented by Hariharan *et al.* [19], containing 10,582 training images with objects from 20 classes. The MS COCO 2014 dataset contains approximately 82K training images containing objects of 80 classes. We evaluated our method on 1,449 validation images and 1,456 test images from the PASCAL VOC dataset and on 40,504 validation images from the MS COCO dataset, by calculating the mean intersection-over-union (mIoU) values.

Reproducibility. We implemented CAM [62] by following the procedure from Ahn *et al.* [2], which is implemented with the PyTorch framework [40]. We used the ResNet-50 [20] backbone for the classification. We fine-tuned our classifier for $K = 10$ iterations with a learning rate of 8×10^{-6} and a batch size of $B = 20$. We set the margin m to 600. For the GNDRP, we set τ to 0.4. For the final semantic segmentation, we used the PyTorch implementation of DeepLab-v2-ResNet101 offered by [39]. We used an initial model pre-trained on the ImageNet dataset [13]. For the MS COCO dataset, we cropped the training images with the crop size of 481×481 rather than 321×321 used for the PASCAL VOC dataset, considering the size of the images in this dataset.

4.2 Weakly Supervised Semantic Segmentation

4.2.1 Quality of the initial seed and pseudo ground truth

PASCAL VOC 2012 dataset: In Table 1, we report the mIoU values of the initial seed and pseudo ground truth masks generated from our method and from other recent techniques. Fol-

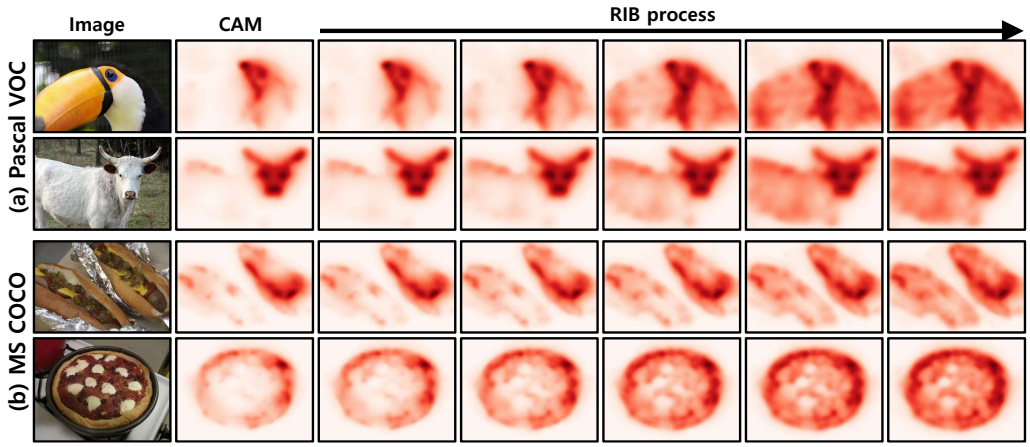

Figure 3: Examples of localization maps obtained during the RIB process for (a) PASCAL VOC training images and (b) MS COCO training images.

lowing SEAM [51], we evaluate a range of thresholds to distinguish between the foreground and the background in the map $\mathcal{M}$ and then determine the best initial seeds. Our initial seeds exhibit 7.7%p improvement from the original CAMs, a baseline for comparison, and simultaneously outperform those from the other methods. Note that our initial seeds are better than those of SEAM, which further refines the initial CAM on a pixel-level by considering the relationship between pixels through an auxiliary self-attention module.

We applied a post-processing method based on conditional random field (CRF) [27] for pixel-level refinement of the initial seeds obtained from the method proposed by Chang *et al.* [6], SEAM [51], IRN [2], and our method. On average, applying CRF improved all the seeds by more than 5%p, with the exception of SEAM. CRF improved SEAM by only 1.4%p, and it is reasonable to believe that this unusually small improvement occurred because the self-attention module had already refined the seed from CAM. When the seed produced by our method is refined with CRF, it is 6.1%p better than that from SEAM and consequently outperforms all the recent competitive methods by a large margin.

Additionally, we compare the pseudo ground truth masks obtained after seed refinement with those obtained using other methods. Most of the compared methods use PSA [3] or IRN [2] to refine their initial seeds. According to Table 1, the masks from our method yield an mIoU of 70.6, thereby outperforming other methods by a large margin.

MS COCO 2014 dataset: Table 1 presents the mIoU values of the initial seed and pseudo ground truth masks obtained by our method and by other recent methods for the MS COCO dataset. We obtained the results of IRN [2] using the official code to set the baseline performance. Our method improved the initial seed and pseudo ground truth masks of our baseline IRN [2], by mIoU margins of 3.0%p and 2.7%p, respectively.

Figure 3 illustrates localization maps gradually refined by the RIB process for the PASCAL VOC and the MS COCO datasets.

4.2.2 Performance of weakly supervised semantic segmentation

PASCAL VOC 2012 dataset: Table 2 presents the mIoU values of the segmentation maps on PASCAL VOC 2012 validation and test images, predicted by our method and other recently introduced weakly supervised semantic segmentation methods, which use bounding box labels or image-level class labels. All the results in Table 2 were obtained using a ResNet-based backbone [20]. Our method achieves mIoU values of 68.3 and 68.6 for the validation and test images, respectively, on the PASCAL VOC 2012 semantic segmentation benchmark, outperforming all the methods that use image-level class labels as weak supervision. In particular, our method outperforms CONTA [58], the best-performing method among our competitors, achieving an mIoU value of 66.1. However, CONTA depends on SEAM [51], which is known to outperform IRN [2]. When CONTA was implemented with IRN for a fairer comparison with our method, its mIoU value decreased to 65.3, which our method surpasses by 3.0%p.

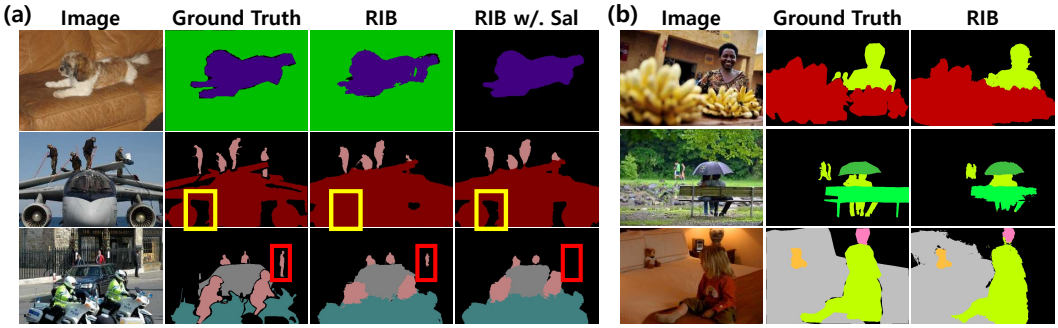

Figure 4: Examples of predicted segmentation masks from IRN [2] and our method for (a) PASCAL VOC 2012 validation images and (b) MS COCO 2014 validation images.

Method	*val*	*test*
Supervision: Bounding box labels		
Song *et al.* $_{\text{CVPR '19}}$ [46]	70.2	-
BBAM $_{\text{CVPR '21}}$ [32]	73.7	73.7
Supervision: Image class labels		
IRN $_{\text{CVPR '19}}$ [2]	63.5	64.8
SEAM $_{\text{CVPR '20}}$ [51]	64.5	65.7
BES $_{\text{ECCV '20}}$ [9]	65.7	66.6
Chang *et al.* $_{\text{CVPR '20}}$ [6]	66.1	65.9
RRM $_{\text{AAAI '20}}$ [57]	66.3	66.5
CONTA $_{\text{NeurIPS '20}}$ [58]	66.1	66.7
RIB (Ours)	**68.3**	**68.6**

Table 2: Comparison of semantic segmentation performance on PASCAL VOC 2012 validation and test images.

Method	Sup.	*val*	*test*
SeeNet $_{\text{NeurIPS '18}}$ [21]	$\mathcal{S}$	63.1	62.8
FickleNet $_{\text{CVPR '19}}$ [30]	$\mathcal{S}$	64.9	65.3
CIAN $_{\text{AAAI '20}}$ [16]	$\mathcal{S}$	64.3	65.3
Zhang *et al.* $_{\text{ECCV '20}}$ [59]	$\mathcal{S}$	66.6	66.7
Fan *et al.* $_{\text{ECCV '20}}$ [17]	$\mathcal{S}$	67.2	66.7
Sun *et al.* $_{\text{ECCV '20}}$ [47]	$\mathcal{S}$	66.2	66.9
LIID $_{\text{TPAMI '20}}$ [38]	$\mathcal{S}_I$	66.5	67.5
Li *et al.* $_{\text{AAAI '21}}$ [35]	$\mathcal{S}$	68.2	68.5
Yao *et al.* $_{\text{CVPR '21}}$ [55]	$\mathcal{S}$	68.3	68.5
RIB (Ours)	$\mathcal{S}$	**70.2**	**70.0**

Table 3: Comparison of semantic segmentation performance on PASCAL VOC 2012 validation and test images using explicit localization cues. $\mathcal{S}$: salient object, $\mathcal{S}_I$: salient instance.

Table 3 compares our method with other recent methods using additional salient object supervision. We utilized salient object supervision used by Li *et al.* [35] and Yao *et al.* [55]. Our method achieves mIoU values of 70.2 and 70.0 for the validation and test images, respectively, outperforming all the recently introduced methods under the same level of supervision.

Figure 4(a) shows examples of predicted segmentation maps by our method with and without saliency supervision. The boundary information provided by saliency supervision allows our method to produce a more precise boundary (yellow boxes). However, the non-salient objects in an image are often ignored when using saliency supervision, while RIB successfully identifies them (*e.g.*, a 'sofa' in the first column and 'person' in red boxes in Figure 4(a)). This empirical finding inspires a potential future work that can simultaneously identify a precise boundary and non-salient objects.

MS COCO 2014 dataset: Table 4 compares our method with other recent methods on MS COCO 2014 validation images. Our method achieves an improvement of 2.4%p in terms of the mIoU score compared with our baseline IRN [2], and outperforms the other recent competitive methods. In the comparison with CONTA [58], the result of IRN reported in CONTA [58] differs from the one we obtained. Therefore, we compare relative improvements: CONTA achieves a 0.8%p improvement compared with IRN (32.6 $\rightarrow$ 33.4), whereas our method achieves 2.4%p (41.4 $\rightarrow$ 43.8). Figure 4(b) presents examples of predicted segmentation maps by our method for the MS COCO validation images.

Method	Backbone	mIoU
ADL $_{\text{TPAMI '20}}$ [10]	VGG16	30.8
CONTA $_{\text{NeurIPS '20}}$ [58]	ResNet50	33.4
Yao *et al.* $_{\text{Access '20}}$ [54]	VGG16	33.6
IRN $_{\text{CVPR '19}}$ [2]	ResNet101	41.4
RIB (Ours)	ResNet101	43.8

Table 4: Comparison of semantic segmentation on MS COCO validation images.

4.3 Ablative Studies

In this section, we analyze our method through various ablation studies conducted on the PASCAL VOC dataset.

Fine-tuning	Seed
Init.	48.8
BCE w/. Tanh	49.7
BCE w/. Sigmoid	50.5
BCE w/. Softsign	50.9
$\mathcal{L}_{\text{RIB}}$	**56.5**

(a)

m	λ	Seed
300	8×10^{-6}	54.0
600	5×10^{-6}	54.9
600	8×10^{-6}	**56.5**
600	1×10^{-5}	56.0
1000	8×10^{-6}	55.9

(b)

Method	Seed
CAM	48.8
RIB-GAP	54.8
RIB-GNDRP (τ=0.3)	55.8
RIB-GNDRP (τ=0.4)	**56.5**
RIB-GNDRP (τ=0.5)	56.0

(c)

Table 5: Comparison of mIoU scores of the initial seed (a) with different activation functions for the final layer, (b) with different values of m and λ, and (c) with different values of τ.

Influence of the total number of RIB iterations K: We analyze the influence of the iteration number K on the effectiveness the RIB process. Figure 5 shows the mIoU score of the initial seed obtained by our baseline CAM, and that of each iteration of the RIB process with GAP or GNDRP. As the RIB process progresses, the localization map is significantly improved, regardless of the pooling method. However, the increase in the performance of RIB with GAP is limited, and even slightly decreases in later iterations. This is because GAP allows features that have already delivered sufficient information to the classification to become even more involved in the classification. Because our proposed GNDRP limits the increase in the contribution of these discriminative regions to the classification, RIB with GNDRP can effectively allow non-discriminative information to be more involved in the classification, resulting in a better localization map.

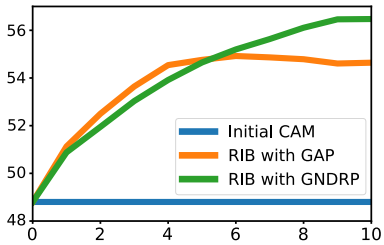

Figure 5: Analysis of RIB with GAP or GNDRP in terms of mIoU (%) of the initial seed.

Fine-tuning with $\mathcal{L}_{\text{RIB}}$: To verify the effectiveness of $\mathcal{L}_{\text{RIB}}$, we fine-tune a model using the BCE loss with various double-sided saturating activation functions. Table 5 (a) shows the mIoU scores of the initial seeds, obtained from a model fine-tuned by the BCE loss with sigmoid, tanh, and softsign activations, and our $\mathcal{L}_{\text{RIB}}$. We adjusted the output of tanh and softsign to have a value between zero and one through the affine transform. Fine-tuning using the BCE loss with double-sided saturating activations improves the initial seed to some extent, which demonstrates the effectiveness of per-sample adaptation; however, their performance improvement is limited due to the remaining information bottleneck. Note that the softsign activation function provides better localization maps than tanh and sigmoid. We believe this is because the gradients from softsign reach zero at a higher value compared with the others (please see the Appendix), and consequently, softsign has less information bottleneck. Our $\mathcal{L}_{\text{RIB}}$ effectively addresses the information bottleneck and achieves the best performance.

Analysis of the sensitivity to hyper-parameters: We analyze the sensitivity of the mIoU of the initial seed to the hyper-parameters involved in the RIB process. Table 5 (b) presents the mIoU values of the initial seed obtained using different combinations of values for the margin m and the learning rate λ. Overall, a slightly lower performance is observed when the strength of the RIB process is weakened by small values of m and λ. For sufficiently large m and λ, the performance of the RIB process is competitive. Table 5 (c) analyzes the influence of the threshold τ involved in the GNDRP. Increasing τ from 0.3 to 0.5 results in less than 1%p change in the mIoU, and thus, we conclude that the RIB process is robust against the changes in τ.

5 Conclusions

In this study, we addressed the major challenge in weakly supervised semantic segmentation with image-level class labels. Through the information bottleneck principle, we first analyzed why the localization map obtained from a classifier identifies only a small region of the target object. Our analysis highlighted that the amount of information delivered from an input image to the output classification is largely determined by the final layer of the DNN. We then developed a method to reduce the information bottleneck through two simple modifications to the existing training scheme: the removal of the final non-linear activation function in the DNN and the introduction of a new pooling method. Our method significantly improved the localization maps obtained from a classifier, exhibiting a new state-of-the-art performance on the PASCAL VOC and MS COCO datasets.

References

[1] Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *TPAMI*, 2018.

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

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

[4] Wonho Bae, Junhyug Noh, and Gunhee Kim. Rethinking class activation mapping for weakly supervised object localization. In *European Conference on Computer Vision*, pages 618–634. Springer, 2020.

[5] Yu-Ting Chang, Qiaosong Wang, Wei-Chih Hung, Robinson Piramuthu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Mixup-cam: Weakly-supervised semantic segmentation via uncertainty regularization. In *BMVC*, 2020.

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

[7] Binghui Chen, Weihong Deng, and Junping Du. Noisy softmax: Improving the generalization ability of dcnn via postponing the early softmax saturation. In *CVPR*, 2017.

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 2017.

[9] Liyi Chen, Weiwei Wu, Chenchen Fu, Xiao Han, and Yuntao Zhang. Weakly supervised semantic segmentation with boundary exploration. In *ECCV*, 2020.

[10] Junsuk Choe, Seungho Lee, and Hyunjung Shim. Attention-based dropout layer for weakly supervised single object localization and semantic segmentation. *IEEE TPAMI*, 2020.

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[12] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[14] Yann Dubois, Douwe Kiela, David J Schwab, and Ramakrishna Vedantam. Learning optimal representations with the decodable information bottleneck. In *NeurIPS*, 2020.

[15] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[16] Junsong Fan, Zhaoxiang Zhang, and Tieniu Tan. Cian: Cross-image affinity net for weakly supervised semantic segmentation. *AAAI*, 2020.

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

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

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

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

[21] Qibin Hou, PengTao Jiang, Yunchao Wei, and Ming-Ming Cheng. Self-erasing network for integral object attention. In *NeurIPS*, 2018.

[22] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[23] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019.

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

[25] Insu Jeon, Wonkwang Lee, Myeongjang Pyeon, and Gunhee Kim. Ib-gan: Disengangled representation learning with information bottleneck generative adversarial networks. In *AAAI*, 2021.

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

[27] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, 2011.

[28] Viveka Kulharia, Siddhartha Chandra, Amit Agrawal, Philip Torr, and Ambrish Tyagi. Box2seg: Attention weighted loss and discriminative feature learning for weakly supervised segmentation. In *ECCV*, 2020.

[29] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[30] 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.

[31] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Frame-to-frame aggregation of active regions in web videos for weakly supervised semantic segmentation. In *ICCV*, 2019.

[32] Jungbeom Lee, Jihun Yi, Chaehun Shin, and Sungroh Yoon. Bbam: Bounding box attribution map for weakly supervised semantic and instance segmentation. In *CVPR*, 2021.

[33] Sungmin Lee, Jangho Lee, Jungbeom Lee, Chul-Kee Park, and Sungroh Yoon. Robust tumor localization with pyramid grad-cam. *arXiv preprint arXiv:1805.11393*, 2018.

[34] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Guided attention inference network. *IEEE TPAMI*, 2019.

[35] Xueyi Li, Tianfei Zhou, Jianwu Li, Yi Zhou, and Zhaoxiang Zhang. Group-wise semantic mining for weakly supervised semantic segmentation. In *AAAI*, 2021.

[36] 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.

[37] Weide Liu, Chi Zhang, Guosheng Lin, Tzu-Yi HUNG, and Chunyan Miao. Weakly supervised segmentation with maximum bipartite graph matching. In *ACMMM*, 2020.

[38] Yun Liu, Yu-Huan Wu, Pei-Song Wen, Yu-Jun Shi, Yu Qiu, and Ming-Ming Cheng. Leveraging instance-, image-and dataset-level information for weakly supervised instance segmentation. *TPAMI*, 2020.

[39] Kazuto Nakashima. DeepLab with PyTorch. https://github.com/kazuto1011/deeplab-pytorch.

[40] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[41] Alexandre Rame and Matthieu Cord. Dice: Diversity in deep ensembles via conditional redundancy adversarial estimation. In *ICLR*, 2021.

[42] Andrew Michael Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan Daniel Tracey, and David Daniel Cox. On the information bottleneck theory of deep learning. In *International Conference on Learning Representations*, 2018.

[43] Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. Restricting the flow: Information bottlenecks for attribution. *ICLR*, 2020.

[44] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[45] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

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

[47] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *ECCV*, 2020.

[48] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised cnn segmentation. In *CVPR*, 2018.

[49] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[50] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.

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

[52] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*, 2017.

[53] 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.

[54] Qi Yao and Xiaojin Gong. Saliency guided self-attention network for weakly and semi-supervised semantic segmentation. *IEEE Access*, 2020.

[55] Yazhou Yao, Tao Chen, Guosen 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.

[56] Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, and Chelsea Finn. Meta-learning without memorization. In *ICLR*, 2019.

[57] Bingfeng Zhang, Jimin Xiao, Yunchao Wei, Mingjie Sun, and Kaizhu Huang. Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. *AAAI*, 2020.

[58] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xiansheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. 2020.

[59] Tianyi Zhang, Guosheng Lin, Weide Liu, Jianfei Cai, and Alex Kot. Splitting vs. merging: Mining object regions with discrepancy and intersection loss for weakly supervised semantic segmentation. In *ECCV*, 2020.

[60] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, 2018.

[61] Andrey Zhmoginov, Ian Fischer, and Mark Sandler. Information-bottleneck approach to salient region discovery. In *ICML Workshop on Self-Supervised Learning*, 2019.

[62] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Please see Section 4
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Our theoretical assumption is already widely accepted by previous literature.
   (b) Did you include complete proofs of all theoretical results? [Yes] Our theoretical proofs are already widely done by previous literature.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Please see the Appendix
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see Section 4
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Please see the Appendix
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please see the Appendix
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Please see Section 4
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]