Semantic-Aware Superpixel for Weakly Supervised Semantic Segmentation

Sangtae Kim[1], **Daeyoung Park**[2], **andByonghyo Shim**[1]

[1]Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea
[2] Department of Information and Communication, Inha University, Incheon, Korea
stkim@islab.snu.ac.kr, dpark@inha.ac.kr, bshim@islab.snu.ac.kr

Abstract

Weakly-supervised semantic segmentation aims to train a semantic segmentation network using weak labels. Among weak labels, image-level label has been the most popular choice due to its simplicity. However, since image-level labels lack accurate object region information, additional modules such as saliency detector have been exploited in weakly supervised semantic segmentation, which requires pixel-level label for training. In this paper, we explore a self-supervised vision transformer to mitigate the heavy efforts on generation of pixel-level annotations. By exploiting the features obtained from self-supervised vision transformer, our superpixel discovery method finds out the semantic-aware superpixels based on the feature similarity in an unsupervised manner. Once we obtain the superpixels, we train the semantic segmentation network using superpixel-guided seeded region growing method. Despite its simplicity, our approach achieves the competitive result with the state-of-the-arts on PASCAL VOC 2012 and MS-COCO 2014 semantic segmentation datasets for weakly supervised semantic segmentation. Our code is available at https://github.com/st17kim/semantic-aware-superpixel.

1 Introduction

Image semantic segmentation, a task to assign a semantic label to every pixel, has received much attention due to its wide range of applications such as autonomous driving, medical diagnosis, and aerial imaging (Cordts et al. 2016; Ronneberger, Fischer, and Brox 2015). Recently, deep learning (DL)-based semantic segmentation has received special attention due to its excellent segmentation performance (Long, Shelhamer, and Darrell 2015). One well-known shortcoming of this approach is that it requires large-scale training dataset with dense annotation for the network training. Since the generation of fully-annotated dataset is laborious, weakly supervised learning has received much attention as a surrogate of the fully supervised learning (Kolesnikov and Lampert 2016; Huang et al. 2018). Among the various types of weak supervisions, image-level labels, indicating the existing classes of an image, are popularly used due to the simplicity (Huang et al. 2018). We henceforth refer to the semantic segmentation using the image-level labels as weakly-supervised semantic segmentation (WSSS).

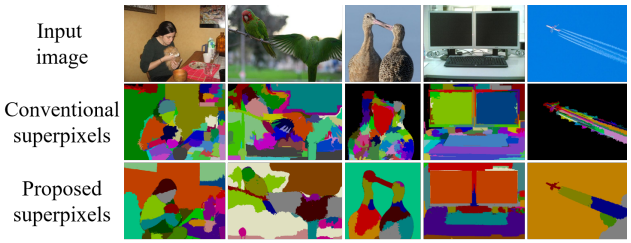

Figure 1: Superpixels obtained by SLIC (Felzenszwalb and Huttenlocher 2004) and our method. The colors are only used to indicate the different superpixels.

One major difficulty of WSSS is to discover object locations and shapes from image-level labels. A typical WSSS approach is to locate the object regions using the class activation mapping (Zhou et al. 2016) and use these region in the training of semantic segmentation network (Kolesnikov and Lampert 2016; Li et al. 2018). However, since the pseudo-labels generated from this approach are sparse and inaccurate, there exists a considerable performance gap between fully-supervised and weakly-supervised semantic segmentation. To obtain the object region information, many recent WSSS approaches exploit the extra supervisions (Yao and Gong 2020; Li et al. 2021). One well-known approach is to employ the saliency detectors to obtain the saliency map indicating the class-agnostic object regions. While the saliency maps are commonly used in many WSSS approaches, the saliency detectors still requires huge effort for the detailed pixel-level annotation.

Recently, DINO (Caron et al. 2021), a self-supervised vision transformer trained by distillation mechanism without labels, has achieved the comparable performance to the state-of-the-art convolutional neural network models. In particular, the feature obtained by DINO appears to contain explicit information about the semantic segmentation of objects in an image. It has been shown that this DINO-based features can be exploited in many computer vision tasks such as unsupervised object detection (Siméoni et al. 2021) or unsupervised saliency detection (Wang et al. 2022).

An aim of this paper is to relieve the thirst for pixel-level

information for WSSS. To this end, we propose a semantic-aware superpixel discovery method using the features obtained from DINO. The main ingredient of our approach is pair-wise relations of pixel indicating semantic similarities of the pair of pixels. Based on the similarities, we identify a seed pixel of an image and find the pixels having similar features to the seed pixel. Then, a group of pixels, referred to *superpixel*, sharing semantic similarities are identified iteratively in an unsupervised manner.

From our observations, our superpixels have two properties: 1) the superpixel contains long-range information even if the consisting pixels are unconnected, meaning that the semantically similar but apart pixels can be grouped together, 2) the number of superpixels depends on the complexity of an input image (e.g., the number or kind of objects), meaning that the number of superpixels is not pre-defined. In Fig. 1, we show some examples for our superpixels.

After obtaining the semantic-aware superpixels, we train the semantic segmentation network using the superpixel-guided seeded region growing method. Although the quality of initial seed (in terms of mIOU) might be poor, by expanding the seeded regions to the neighboring superpixels, the quality can be improved substantially. Unlike the conventional seeded regions growing method that gradually expand the seeded region to adjacent pixels (Huang et al. 2018), our approach expands the seeded region to superpixels if the superpixels are likely to be one of classes. Benefited from the object shape expressed in the superpixel, our method can generate of high-quality seed depicting the detailed object boundaries.

The contributions of this paper are as follows:

- We propose a threshold-based superpixel algorithm using the self-supervised vision transformer in an unsupervised manner. Our method produces superpixels containing semantically similar pixels which are friendly to semantic segmentation tasks.
- We train the semantic segmentation network using the initial seed while refining the seed using the superpixel-guided seeded region growing method. The refined seed becomes dense during the training process and significantly boosts the segmentation performance.
- Our approach outperforms the state-of-the-art methods on PASCAL VOC 2012 and MS-COCO 2014 semantic segmentation datasets.

2 Related Work

Weakly Supervised Semantic Segmentation Basically, the goal of weakly supervised semantic segmentation is to train a semantic segmentation network using coarse labels such as points, scribbles, or image-level labels. Due to simplicity, WSSS using image-level labels is widely studied (Kolesnikov and Lampert 2016). A typical approach is to train a classification network and obtain an initial seed using the class activation mapping technique. Since the initial seed obtained by this approach is sparse and inaccurate, there have been many efforts to improve the qualities of seed. For examples, in (Wang et al. 2020), self-supervision based on the equivariant attention mechanism is exploited to discover object regions. In (Lee, Kim, and Yoon 2021), adv-CAM method is proposed to find non-discriminative object regions in an anti-adversarial manner. In (Lee et al. 2021), an approach that encourages the network to perceive non-discriminative object regions by reducing information bottleneck is proposed.

Superpixel A superpixel is a set of homogeneous pixels based on features such as bright, color, or texture. To perform the superpixel segmentation, a graph-based method (Felzenszwalb and Huttenlocher 2004) or a clustering-based method (Li and Chen 2015) has been popularly used. The superpixels obtained from these methods are used in many WSSS approaches to recover smooth object boundaries (Zhang et al. 2013; Kwak, Hong, and Han 2017; Fan et al. 2020; Zhang et al. 2021). However, since the superpixels used in these approaches are quite over-segmented, it is difficult to obtain long-range information from these superpixels and discover the meaningful information for WSSS.

Seeded Region Growing The seeded region growing method (Adams and Bischof 1994) is an unsupervised segmentation technique that examines neighboring pixels of initial seed points and determines whether the neighboring pixels should be added to the region depending on a region similarity criterion. For the successful image segmentation, it is important to locate the initial seed to proper pixels and use a criterion that can characterize the image regions. In (Huang et al. 2018), an approach to exploit the initial seed generated by a classification network in training of semantic segmentation network and computes pixel similarity using high-level semantic features has been proposed.

Transformer The transformer and self-attention models have revolutionized machine translation and NLP fields (Vaswani et al. 2017; Devlin et al. 2018). Recently, its adoption to computer vision, the vision transformer (ViT) (Dosovitskiy et al. 2020), has shown great performance gain over the convolutional neural network (CNN) models. Unfortunately, to achieve such performance, the datasets containing enormous number of training images are required (e.g., JFT-300M dataset). As a means to alleviate this burden, self-supervision-based training technique is proposed (Touvron et al. 2021). In particular, in (Caron et al. 2021), it is demonstrated that self-supervised ViTs can automatically segment the background pixels of an image, even though they are not trained using pixel-level supervision.

3 The Proposed Approach

In this section, we discuss the proposed WSSS framework. We first introduce how to discover semantic-aware superpixels from the self-supervised vision transformer-based features. We then discuss how to generate the initial seed for training of the semantic segmentation network. We also explain how to train the semantic segmentation network using the superpixel-guided seeded region growing method.

3.1 Superpixel Generation

For the successful semantic segmentation, the superpixels should meet two following conditions: 1) Each superpixel is a sufficiently large set of homogeneous pixels so that all

pixels have the same semantic class. 2) The number of superpixels depends on the number of the sets of semantically similar pixels, not a pre-defined number. To obtain superpixels satisfying these conditions, we identify a pixel used as a seed of a superpixel and then find out the pixels sharing similar semantic features to the seed pixel. In our approach, we use the vision transformer-based feature to group the pixels into superpixels.

Before we proceed, we briefly review the vision transformer and its components. Vision transformers take a sequence of patches of fixed size $P \times P$ as input. For a color image I of spatial size $H \times W$, we have $N = HW/P^2$ patches. Each patch is first embedded in a d-dimensional latent space via a trained convolutional projection layer and delivered to the series of transformer blocks. The main part of vision transformer consists of multiple blocks including multi-head self-attention layers and multi-layer perceptrons. In the front part of each block, there are three parallel linear layers taking an input $X \in \mathbb{R}^{(N+1)\times d}$ to produce a query Q, a key K, and a value V, all in $\mathbb{R}^{(N+1)\times d}$. The resulting output for each head is given by $Y = \text{softmax}(QK^T/d^{1/2})V$, where softmax is applied row-wise. In our work, we concatenate the keys from all heads in the self-attention layer of the last transformer block to obtain final features that are the main ingredient in discovering the superpixels (Siméoni et al. 2021).

Let $f_p \in \mathbb{R}^{d\times 1}$ be the feature vector corresponding to pixel p of input image I and $\mathcal{P} = \{1, 2, \cdots, N\}$ be the set of indices of candidate pixels. We compute the pair-wise feature similarity matrix A and the binary adjacency matrix B indicating whether the similarity between two pixels is positive as

$$A_{pq} = \frac{f_p^T f_q}{\|f_p\|_2 \|f_q\|_2}, B_{pq} = \begin{cases} 1 & \text{if } A_{pq} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $\|\cdot\|_2$ is ℓ_2 norm.

The sum of the p-th row of B is defined as the degree of pixel p, d_p, which indicates the number of pixels having semantically similar features to p. Based on d_p, we can notice how large the group of pixels having similar semantic features to p is. If the features of objects of different class are clearly distinguishable, we may conclude that semantically similar pixels have the same class. Accordingly, we can guess whether p belongs to a large object (e.g., sky, car, or building) or a small object (e.g., bottle, eyes, or wheel). One way to identify a group of pixels representing an object can be to select a pixel p^*, *a seed pixel*, and find the pixels having similar semantic features to p^*.

One can wonder how to select a good seed pixel to find a group of pixels, *a superpixel*. Here, we use a simple rule using the degree of pixels. One can consider selecting p with either the highest or the lowest degree to find a large or small object, respectively. From our extensive experiments, we observe that it is better to identify small objects since there could be a pixel having an overwhelming degree, resulting in a grouping of the most pixels (see supplementary material for comparison). Hence, our strategy to partition an image into multiple superpixels is to find out a superpixel corresponding to the smallest object and repeat this process after excluding the pixels of the discovered superpixel from the candidates.

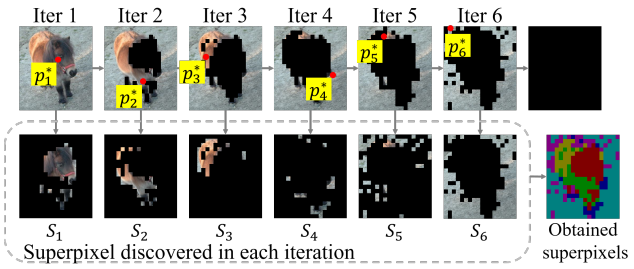

Figure 2: A procedure of the proposed superpixel discovery method.

To sum up, in each iterative step i, the seed pixel p_i^* of a superpixel $\mathcal{S}_i$ is selected by finding out the pixel with the lowest degree as $p_i^* = \arg\min_p \sum_q B_{pq}$. Then, the pixels to be included to superpixel $\mathcal{S}_i$ are determined by following criterion: $\mathcal{S}_i = \{q | A_{p_i^* q} > \tau\}$ where τ is the pre-defined threshold for feature similarity. We exclude the pixels of $\mathcal{S}_i$ from $\mathcal{P}$ and repeat this procedure until $\mathcal{P}$ becomes the empty set. In Fig. 2, we illustrate the procedure of the proposed superpixel method.

3.2 Initial Seed Generation

To generate the initial seed used for training of the semantic segmentation network, we first train a classification network. We follow the common practice to train the classification network using the multi-label classification loss:

$$\ell_{cls} = \frac{1}{C}\sum_{c=1}^{C}\Big(- y_c \log(\sigma(\widehat{x}_c)) - (1-y_c)\log(1-\sigma(\widehat{x}_c))\Big) \quad (2)$$

where C is the number of foreground classes, y_c is the image-level label for class c, $\widehat{x}_c$ is the predicted class score for class c, and $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function. Then, we obtain the class activation map(CAM) M of class c as

$$M_{p,c} = \begin{cases} \frac{w_c^T x'_p}{\max_q w_c^T x'_q} & \text{if } c \text{ is present class} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where x'_p is the output of the second last layer for pixel p and w_c is the weight of the last layer for class c. Using the CAM, we can generate an initial seed L by assigning the class for the confident foreground pixels as

$$L_p = \begin{cases} \arg\max_c M_{p,c} & \text{if } M_{p,c} > \alpha \\ \text{unlabeled} & \text{otherwise} \end{cases} \quad (4)$$

where α is the threshold.

On the other hand, background regions are not directly identified from CAM since the classification network does not learn the background class explicitly. A common approach to identify background regions is to set the low-activated foreground regions in the CAM to the background

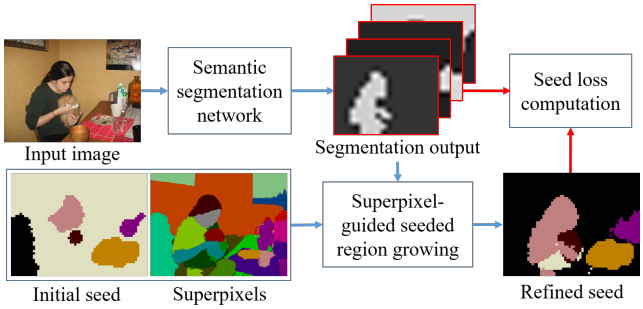

Figure 3: The architecture for training of the semantic segmentation network.

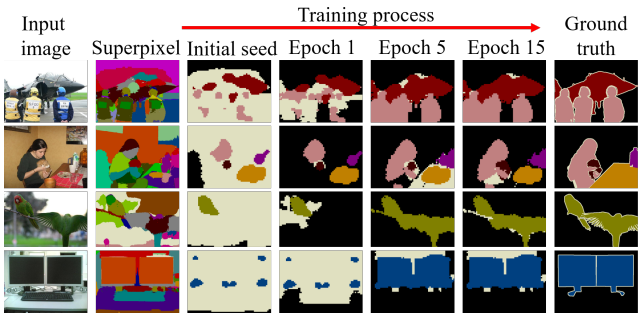

Figure 4: Examples for initial seed refined by superpixel-guided seeded region growing during the training process.

region. However, the discovered regions using this approach may contain the foreground regions which are not expressed in the CAM. To better identify the background regions, we find the superpixel which is the least likely to be foreground regions. Here, we assume that there are background regions in every input image.

Specifically, we compute a class-agnostic foreground activation map F by taking the maximum pixels for present foreground classes as $F_p = \max_{c \in \mathcal{C}} M_{p,c}$ where $\mathcal{C}$ is the set of present classes in I. Then, the foreground score $z(\mathcal{S}_i)$ is computed as the average of F over $\mathcal{S}_i$, that is,

$$z(\mathcal{S}_i) = \frac{1}{|\mathcal{S}_i|} \sum_{p \in \mathcal{S}_i} F_p \tag{5}$$

where $|\mathcal{S}_i|$ is the number of pixels contained in $\mathcal{S}_i$. We select $\mathcal{S}_i$ with the lowest $z(\mathcal{S}_i)$ as background pixels:

$$L_p = 0 \text{ for } p \in \mathcal{S}_i \text{ s.t. } \mathcal{S}_i = \arg\min_{\mathcal{S}'_i} z(\mathcal{S}'_i) \tag{6}$$

where 0 indicates the background class. Although there exist very few images not containing background regions, we can construct reliable seed for background class for the most images.

3.3 Segmentation Network Training

Basically, the semantic segmentation network learns the object regions from sparse initial seed constructed above. During the training process, the superpixel-guided seeded region growing is performed to assign the classes to the promising superpixels. We briefly illustrate the architecture for training the segmentation network in Fig. 3.

Specifically, let H be the softmax output of segmentation network. We apply a simple probability threshold for each superpixel. To preserve the confident pixels in the initial seed, we slightly modify the superpixel such that it excludes the pixels labeled in the initial seed. In other words, we modify the superpixel $\mathcal{S}'_i$ as $\tilde{\mathcal{S}}_i = \mathcal{S}_i \setminus \{p | L_p \text{ is labeled}\}$. Using the segmentation probability H, the average of probability of class c over $\tilde{\mathcal{S}}_i$ is computed as

$$s(\tilde{\mathcal{S}}_i)_c = \frac{1}{|\tilde{\mathcal{S}}_i|} \sum_{p \in \tilde{\mathcal{S}}_i} H_{p,c}. \tag{7}$$

Then, the class c is assigned to L_p if the two following criteria are satisfied:

$$s(\tilde{\mathcal{S}}_i)_c = \max_{c'} s(\tilde{\mathcal{S}}_i)_{c'} \text{ and } s(\tilde{\mathcal{S}}_i)_c > \beta. \tag{8}$$

That is, the class c is assigned to the superpixel $\tilde{\mathcal{S}}_i$ if $\tilde{\mathcal{S}}_i$ is the most likely to be class c and the average of probability if greater than threshold β. Although the initial seed is sparse, the labeled regions are expanding to neighboring superpixels by region growing as the segmentation network is trained. In Fig. 4, we show some examples illustrating the refined seed obtained by superpixel-guided seeded regions growing during the training process of the segmentation network.

We train the semantic segmentation network using the balanced seed loss (Huang et al. 2018) that balances the losses between background and foreground classes:

$$\ell_{seed} = -\sum_{p \in \mathcal{L}^b} \frac{1}{|\mathcal{L}^b|} \log H_{p,0} - \sum_{p \in \mathcal{L}^f, c \in \mathcal{C}} \frac{1}{|\mathcal{L}^f|} \log H_{p,c} \tag{9}$$

where $H_{p,0}$ is the probability of background class at position p, $\mathcal{L}^b = \{p | L_p = 0\}$ is the set of background pixels, and $\mathcal{L}^f = \{p | 1 \leq L_p \leq C\}$ is the set of foreground pixels. In the loss computation, the unlabeled pixels are ignored.

4 Experiments

4.1 Datasets and Experiment Settings

We evaluate the proposed approach on the PASCAL VOC 2012 (Everingham et al. 2015) and MS-COCO 2014 (Lin et al. 2014) segmentation benchmark datasets. PASCAL VOC has 20 foreground classes and one background class and consists of 1,464 training images, 1,449 validation images, and 1,456 test images. As in many practices (Chen et al. 2017; Wei et al. 2017), additional dataset is augmented to training dataset, resulting 10,582 training images in total (Hariharan et al. 2011). MS-COCO has 80 foreground classes and one background class and consists of 82,783 training images and 40,504 validation images. In our experiments, we only utilize image-level annotations for the training of semantic segmentation network. As a performance measure, we use mean intersection-over-union (mIOU), average of IOUs over all classes. To obtain the result on the PASCAL VOC test set, we submit the predicted results to the official PASCAL VOC evaluation server.

Method	Val	Test
IRN (Ahn, Cho, and Kwak 2019)	63.5	64.8
RRM (Zhang et al. 2020a)	66.3	66.5
ICD (Fan et al. 2020)	64.1	64.3
SAEM (Wang et al. 2020)	64.5	65.7
BES (Chen et al. 2020)	65.7	66.6
CONTA (Zhang et al. 2020b)	66.1	66.7
ECSNet (Sun et al. 2021)	66.6	67.6
CDA (Su et al. 2021)	66.1	66.8
CPN (Zhang et al. 2021)	67.8	68.5
CGnet (Kweon et al. 2021)	68.4	68.2
advCAM (Lee, Kim, and Yoon 2021)	68.1	68.0
RIB (Lee et al. 2021)	68.3	68.6
SIPE (Chen et al. 2022)	68.8	69.7
CLIMS (Xie et al. 2022)	69.3	68.7
AMN (Lee, Kim, and Shim 2022)	69.5	69.6
Ours	**69.5**	**70.1**

Table 1: Comparison of ResNet-based weakly-supervised semantic segmentation methods' mean IOUs on PASCAL VOC 2012 *val* and *test* set with only image-level label supervision.

For vision transformer, we employ off-the-shelf ViT-Base/8 (Dosovitskiy et al. 2020) trained using DINO (Caron et al. 2021). Without fine tuning the ViT, we use the key K of the last (12th) transformer block as the features for generating superpixels as used in (Siméoni et al. 2021). For the classification network and segmentation network, we employ ResNet50 and ResNet101 (He et al. 2016) as the backbone network, respectively. Both networks are pre-trained on ImageNet classification dataset (Russakovsky et al. 2015). For the segmentation network architecture, we use deeplab-ASPP module (Chen et al. 2017) appended to the ResNet101 backbone network. For the last layer, the parameters are initialized from the normal distribution. In the training of the segmentation network, we only update parameters of convolutional layers while fixing the parameters of batch normalization layers. The obtained superpixels and the softmax output of the segmentation network is post-processed by CRF (Krähenbühl and Koltun 2011).

To improve the robustness of the segmentation network, we apply the data augmentation techniques. We randomly flip and scale ($\{0.5, 1, 1.5, 2\}$) input images. The resulting images are cropped to 448×448 at random location. We also apply color augmentation techniques by randomly changing brightness, contrast, saturation, and hue. For the segmentation network, we use multi-scale inputs with scales, $S = \{1, 0.75, 0.5\}$ in both training and test phases (Chen et al. 2016, 2017). We set τ to 0.3 in superpixel discovery method. We set $\alpha = 0.6$ to identify the foreground pixels and $\beta = 0.7$ for the criterion in seeded region growing. We use the stochastic gradient descent optimizer with the momentum 0.9. We set the weight decay to 0.0005 and the batch size to 20. We employ polynomial learning rate policy (Liu, Rabinovich, and Berg 2015) with initial learning rate 10^{-3} and power 0.9, i.e., $L = 10^{-3} \times (1 - iter/maxiter)^{0.9}$. In early training iterations, we gradually increase the learning rate from 10^{-6} to 10^{-3} through the first three epochs. The learning rate for the last layers is multiplied by 10. We train the segmentation network for 15 epochs. Our approach is implemented with Tensorflow (Abadi et al. 2016). The classification network and the segmentation network are trained on a single NVIDIA GeForce Titan Xp.

Method	Val
SEC (Kolesnikov and Lampert 2016)	22.4
DSRG (Huang et al. 2018)	26.0
GSM (Li et al. 2021)	28.4
CONTA (Zhang et al. 2020b)	33.4
SGAN (Yao and Gong 2020)	33.6
IRN (Ahn, Cho, and Kwak 2019)	41.4
RIB (Lee et al. 2021)	43.8
SIPE (Chen et al. 2022)	43.6
AMN (Lee, Kim, and Shim 2022)	44.7
Ours	**44.8**

Table 2: Comparison of weakly-supervised semantic segmentation methods' mean IOUs on MS COCO 2014 *val* set.

4.2 Comparisons with State-of-the-Art Methods

We evaluate the performance of the proposed method and the state-of-the-art WSSS methods. In Table 1, we summarize the mIOUs on PASCAL VOC 2012. All method use only image-level labels without additional saliency supervision. In Table 2, we also summarize the mIOUs on MS-COCO 2014. From the results, we observe that our approach outperforms the conventional WSSS approaches. Specifically, our approach achieves mIOU of 69.5% and 70.1% for *val* and *test* set, respectively, on PASCAL VOC 2012 and 44.8% for *val* set on MS-COCO 2014.

In particular, we use the same classification network as used in (Ahn, Cho, and Kwak 2019; Lee et al. 2021). In (Fan et al. 2020), the superpixels are used to recover the object boundaries. In (Zhang et al. 2021), superpixel is exploited in partitioning the input image into complementary patch. Compared to these superpixel-based methods which are only benefited from local information about the object boundaries, our approach takes advantage of local and global information contained in our semantic-aware superpixels.

4.3 Ablation Studies

In the proposed method, we use the feature obtained from the DINO. To investigate the effects of different features, we compare the superpixels generated using various features and discuss the qualities of the obtained superpixels. In this experiment, the features we use are: 1) the RGB values of image itself, the most basic feature of pixels, 2) the CNN features obtained from supervised CNN (ResNet (He et al. 2016)), and self-supervised CNN (MoCov3 (Chen, Xie, and He 2021)), 3) transformer features obtained from supervised transformer (ViT (Dosovitskiy et al. 2020)), and self-supervised transformers (DINO (Caron et al. 2021) and MAE (He et al. 2022) which is known to outperform DINO in down-stream tasks). The backbones of CNNs and transformers are ResNet50 and ViT-Base/16, respectively. The

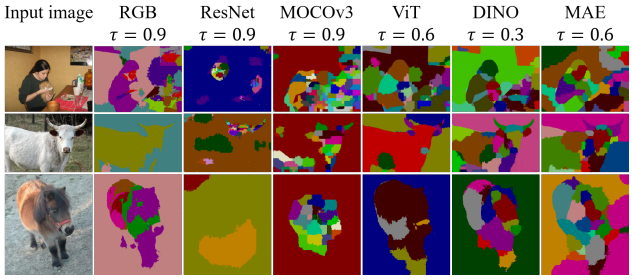

Figure 5: Superpixels generated using different features.

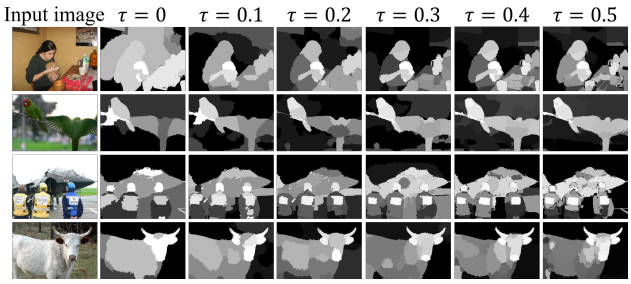

Figure 6: Superpixels according to different thresholds τ.

τ	0	0.1	0.2	0.3	0.4	0.5
Val	64.8	65.0	64.6	65.1	63.8	62.9
Val+crf	69.3	**69.5**	69.1	**69.5**	68.4	67.5

Table 3: Comparison of mean IOUs on PASCAL VOC *val* and *test* set using different superpixels.

features of CNNs and transformers are the output of the last layer and the key of the last transformer block, respectively.

We discuss the superpixels obtained from different features. We show the examples of superpixels generated from different features in Fig. 5. In these experiments, we first generate the superpixels using the DINO feature with setting $\tau = 0.3$. The τ-values for other features are adjusted so that the numbers of superpixels are similar to that of superpixels generated from the DINO feature.

- RGB feature: We observe that we can find superpixels using RGB values. However, since the RGB values are low-level features, we cannot clearly partition the image.
- CNN features: We observe that we cannot properly generate the superpixels using CNN features of both supervised and self-supervised networks. This is because the CNN feature of each pixel depends heavily on the neighboring pixels, resulting in very high similarities between almost all pairs of pixels. Hence, we cannot partition the image when $\tau < 0.8$. By setting τ to high value (e.g., 0.9), we can obtain the partitioned images with poor qualities.
- Transformer features: We observe that we can obtain the superpixels with reasonable qualities using the features of transformers. One notable point is that we need to set τ to high value when we use the features of ViT and MAE. This is because the ViT is trained to classify images which forces the network to recognize the object itself and MAE is trained to predict the masked regions which forces the network to understand overall context of images. Hence, the ViT and MAE may not pay much attention to the details of images, thereby generating highly similar features on objects. On the other hand, DINO is trained to extract diverse features for each image patch. In fact, the features of DINO represents not only the objects but also their parts in detail so we used them in the generation of superpixels.

In our superpixel discovery method, the seed pixel is the pixel with the lowest degree so that the seed pixel might fall in the smallest objects or their parts. By varying τ, we can decide how many pixels will be grouped with the seed pixel. In Fig. 6, we show some examples for our superpixel for different threshold. The brightness indicates the order of discovered superpixels, that is, the bright one is discovered first and dark one is discovered later. When τ is small, we obtain the superpixels containing whole object of semantic class but may suffer from bad segmentation particularly for small objects. When τ is large, we can obtain the superpixels whose pixels are highly likely to have the same semantic class but may suffer from the oversegmentation.

To investigate the effect of τ in the segmentation performance, we generate various superpixels using different τ and use them to train the segmentation network. We summarize the results in Table 3. We can observe that the segmentation performance degrades when we use oversegmented superpixel.

To examine the effect of α in generating the initial seed, we conduct experiments using different initial seeds. We summarize the results in Table 4. From the results, we see that the best performance is obtained for $\alpha = 0.6$. A notable point is that our initial seeds are generated in a different way from the conventional approaches, in which there are many efforts on obtaining the dense initial seeds. Interestingly, we can achieve higher performance gain from superpixel-guided seeded region growing (denoted as 'RG') when the initial seed is more sparse (i.e., α is high).

We study the effects of β in superpixel-guided seeded regions growing. Similarly to (Huang et al. 2018), we apply different β for background and foreground classes, β_{bg} and β_{fg}, respectively. We summarize the segmentation performance using various combinations of β_{bg} and β_{fg} in Table 5. From the results, we see that we can achieve good segmentation performance when we choose the two parameters

α	RG	0.3	0.4	0.5	0.6	0.7	0.8
Val	N	51.3	52.7	53.3	53.6	52.4	50.8
Val+crf	N	58.0	60.0	60.0	59.7	57.1	53.9
Val	Y	59.7	62.8	65.0	66.3	66.7	65.5
Val+crf	Y	66.0	68.6	69.2	**69.4**	69.0	67.6

Table 4: Comparison of mean IOUs on PASCAL VOC *val* set using different α for generating initial seed.

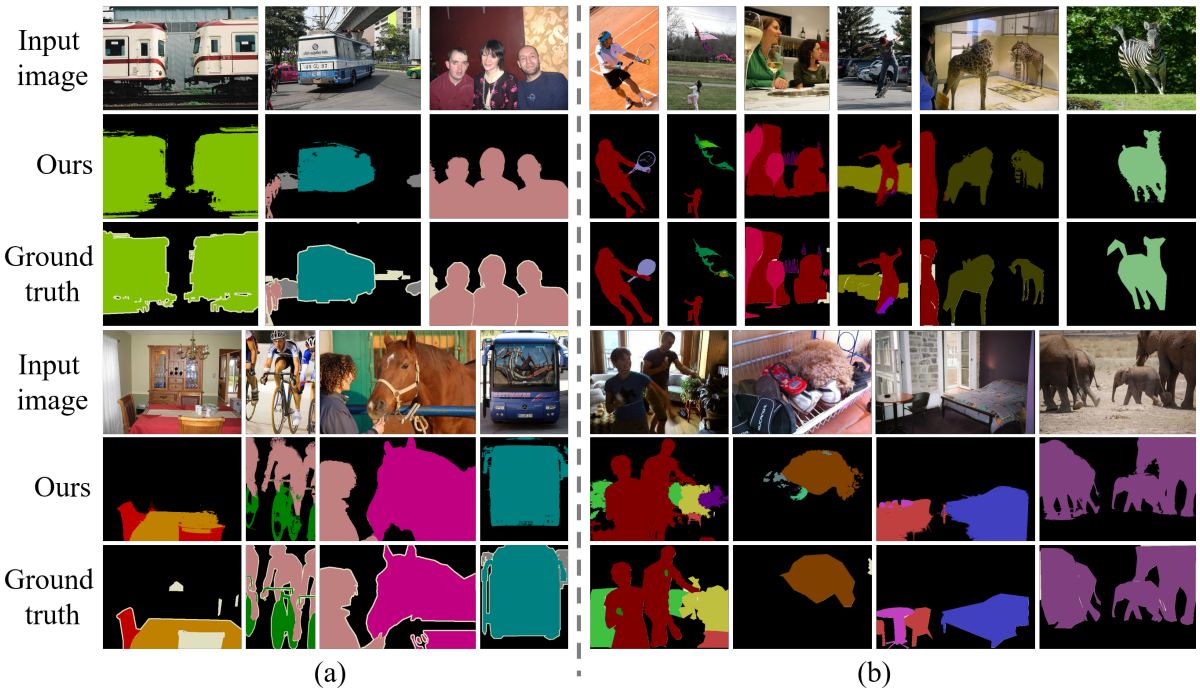

Figure 7: Examples of our segmentation output for (a) PASCAL VOC 2012 and (b) MS-COCO 2014 *val* set.

		β_{bg}			
		0.5	0.6	0.7	0.8
β_{fg}	0.5	60.3/64.3	59.3/62.7	56.5/59.7	51.5/53.9
	0.6	62.1/66.5	62.3/66.5	61.0/65.2	57.5/61.4
	0.7	58.0/61.8	61.3/65.7	**62.3/66.8**	61.3/66.2
	0.8	51.9/53.6	55.5/58.5	60.3/64.8	61.2/66.6

Table 5: Comparison of mean IOUs for different β_{bg} and β_{fg} on PASCAL VOC 2012 *val* set.

similarly. The best result is obtained using $\beta_{bg} = 0.7$ and $\beta_{fg} = 0.7$. If β is too low, the classes can be easily assigned to superpixel, resulting in an incorrect segmentation. In contrast, if β is too high, only highly-confident classes can be assigned to superpixel so some superpixels could never be labeled.

4.4 Qualitative Results

In Fig. 7, we provide qualitative results obtained from our segmentation network. Although we do not use external saliency map in the training of the segmentation network, our approach can predict the objects with accurate boundaries.

In Fig. 8, we also provide some failure cases for the refined seed in the training process and wrong prediction for similar images in *val* set. In particular, for the classes known to be difficult such as table or sofa, the seeded regions in the initial seed rarely expand to the other superpixels.

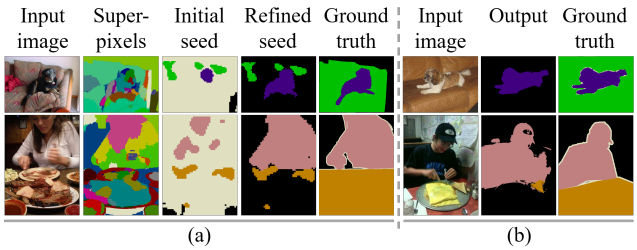

Figure 8: Examples for failure cases of (a) the refined seed in training process and (b) wrong prediction for similar images in *val* images.

5 Conclusion

In this work, we have proposed a simple superpixel discovery method to find out the semantic-aware superpixels in an unsupervised manner. Without relying on external pixel-level labels, we can exploit the pixel-level information on object boundaries contained in our superpixels. We also have shown that our semantic segmentation network training strategy using the superpixel-guided seeded region growing method outperforms the conventional WSSS approaches. Our extensive experiments have demonstrated that our approach is effective in solving WSSS problem. A limitation of the proposed approach is that it only shows the effectiveness of superpixels in WSSS systems. We believe that the proposed superpixel is helpful to solve more challenging computer vision tasks such as an unsupervised segmentation segmentation.

Acknowledgments

This work was supported in part by Samsung Electronics Co., Ltd (IO210201-08353-01), the National Research Foundation of Korea (NRF) Grant through the Ministry of Science and ICT (MSIT), Korea Government, under Grand (2020R1A2C2102198, 2022R1A5A1027646), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2022-00155915, Artificial Intelligence Convergence Innovation Human Resources Development (Inha University)).

References

Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 265–283.

Adams, R.; and Bischof, L. 1994. Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence*, 16(6): 641–647.

Ahn, J.; Cho, S.; and Kwak, S. 2019. Weakly supervised learning of instance segmentation with inter-pixel relations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2209–2218.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9650–9660.

Chen, L.; Wu, W.; Fu, C.; Han, X.; and Zhang, Y. 2020. Weakly supervised semantic segmentation with boundary exploration. In *European Conference on Computer Vision*, 347–362. Springer.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4): 834–848.

Chen, L.-C.; Yang, Y.; Wang, J.; Xu, W.; and Yuille, A. L. 2016. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3640–3649.

Chen, Q.; Yang, L.; Lai, J.-H.; and Xie, X. 2022. Self-supervised Image-specific Prototype Exploration for Weakly Supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4288–4298.

Chen, X.; Xie, S.; and He, K. 2021. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9640–9649.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Everingham, M.; Eslami, S. A.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2015. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1): 98–136.

Fan, J.; Zhang, Z.; Song, C.; and Tan, T. 2020. Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4283–4292.

Felzenszwalb, P. F.; and Huttenlocher, D. P. 2004. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2): 167–181.

Hariharan, B.; Arbeláez, P.; Bourdev, L.; Maji, S.; and Malik, J. 2011. Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision*, 991–998. IEEE.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16000–16009.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, Z.; Wang, X.; Wang, J.; Liu, W.; and Wang, J. 2018. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7014–7023.

Kolesnikov, A.; and Lampert, C. H. 2016. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European Conference on Computer Vision*, 695–711. Springer, Cham.

Krähenbühl, P.; and Koltun, V. 2011. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, 109–117.

Kwak, S.; Hong, S.; and Han, B. 2017. Weakly supervised semantic segmentation using superpixel pooling network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Kweon, H.; Yoon, S.-H.; Kim, H.; Park, D.; and Yoon, K.-J. 2021. Unlocking the potential of ordinary classifier: Class-specific adversarial erasing framework for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6994–7003.

Lee, J.; Choi, J.; Mok, J.; and Yoon, S. 2021. Reducing Information Bottleneck for Weakly Supervised Semantic Segmentation. *Advances in Neural Information Processing Systems*, 34.

Lee, J.; Kim, E.; and Yoon, S. 2021. Anti-Adversarially Manipulated Attributions for Weakly and Semi-Supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4071–4080.

Lee, M.; Kim, D.; and Shim, H. 2022. Threshold Matters in WSSS: Manipulating the Activation for the Robust and Accurate Segmentation Model Against Thresholds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4330–4339.

Li, K.; Wu, Z.; Peng, K.-C.; Ernst, J.; and Fu, Y. 2018. Tell me where to look: Guided attention inference network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9215–9223.

Li, X.; Zhou, T.; Li, J.; Zhou, Y.; and Zhang, Z. 2021. Group-wise semantic mining for weakly supervised semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1984–1992.

Li, Z.; and Chen, J. 2015. Superpixel segmentation using linear spectral clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1356–1363.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer, Cham.

Liu, W.; Rabinovich, A.; and Berg, A. C. 2015. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer, Cham.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252.

Siméoni, O.; Puy, G.; Vo, H. V.; Roburin, S.; Gidaris, S.; Bursuc, A.; Pérez, P.; Marlet, R.; and Ponce, J. 2021. Localizing Objects with Self-Supervised Transformers and no Labels. *arXiv preprint arXiv:2109.14279*.

Su, Y.; Sun, R.; Lin, G.; and Wu, Q. 2021. Context decoupling augmentation for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7004–7014.

Sun, K.; Shi, H.; Zhang, Z.; and Huang, Y. 2021. ECS-Net: Improving Weakly Supervised Semantic Segmentation by Using Connections Between Class Activation Maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7283–7292.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 10347–10357. PMLR.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Y.; Shen, X.; Hu, S.; Yuan, Y.; Crowley, J.; and Vaufreydaz, D. 2022. Self-Supervised Transformers for Unsupervised Object Discovery using Normalized Cut. *arXiv preprint arXiv:2202.11539*.

Wang, Y.; Zhang, J.; Kan, M.; Shan, S.; and Chen, X. 2020. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12275–12284.

Wei, Y.; Feng, J.; Liang, X.; Cheng, M.-M.; Zhao, Y.; and Yan, S. 2017. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1568–1576.

Xie, J.; Hou, X.; Ye, K.; and Shen, L. 2022. CLIMS: Cross Language Image Matching for Weakly Supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4483–4492.

Yao, Q.; and Gong, X. 2020. Saliency Guided Self-Attention Network for Weakly and Semi-Supervised Semantic Segmentation. *IEEE Access*, 8: 14413–14423.

Zhang, B.; Xiao, J.; Wei, Y.; Sun, M.; and Huang, K. 2020a. Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12765–12772.

Zhang, D.; Zhang, H.; Tang, J.; Hua, X.-S.; and Sun, Q. 2020b. Causal intervention for weakly-supervised semantic segmentation. *Advances in Neural Information Processing Systems*, 33: 655–666.

Zhang, F.; Gu, C.; Zhang, C.; and Dai, Y. 2021. Complementary patch for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7242–7251.

Zhang, L.; Song, M.; Liu, Z.; Liu, X.; Bu, J.; and Chen, C. 2013. Probabilistic graphlet cut: Exploiting spatial structure cue for weakly supervised image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1908–1915.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929.