Ph.D. DISSERTATION

A Study on Weakly Supervised Semantic Segmentation Using Image Masking and Clustering

이미지 마스킹과 클러스터링을 이용한 약지도 영상 분할

BY

SANGTAE KIM

AUGUST 2022

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

A Study on Weakly Supervised Semantic Segmentation Using Image Masking and Clustering

이미지 마스킹과 클러스터링을 이용한 약지도 영상 분할

BY

SANGTAE KIM

AUGUST 2022

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

A Study on Weakly Supervised Semantic Segmentation Using Image Masking and Clustering

이미지 마스킹과 클러스터링을 이용한 약지도 영상 분할

지도교수 심병효

이 논문을 공학박사 학위논문으로 제출함

2022년 8월

서울대학교 대학원

전기 정보 공학부

김상태

김상태의 공학박사 학위 논문을 인준함

2022년 8월

위원장: 김성철 (인)

부위원장: 심병효 (인)

위원: 이경한 (인)

위원: 문태섭 (인)

위원: 박대영 (인)

Abstract

Image semantic segmentation, a task to classify each pixel among the interested classes, is an important problem with a wide range of applications such as autonomous driving, medical diagnosis, industrial automation, and aerial imaging. In recent years, deep convolutional neural networks have shown outstanding performances in image semantic segmentation. A main bottleneck of these approaches is that it requires large amount of fully-annotated data for training such networks. Since the acquisition of fully-annotated dataset is laborious and expensive, weakly supervised semantic segmentation (WSSS) has been suggested as an promising approach for future research direction. There are various types of weak labels for semantic segmentation, for instance, image-level labels, points, scribbles, and bounding boxes. Among these weak labels, image-level labels are popularly used in WSSS for its simplicity. In essence, image-level label denotes the existence of objects in an image. In this dissertation, we consider the problem of weakly supervised semantic segmentation using image-level label.

In the first part of dissertation, we introduce a new training strategy for weakly supervised semantic segmentation. In the proposed approach, we apply image masking technique inspired by human visual system that focuses on interesting vision field and ignores irrelevant parts. By guiding the attention of classification network using the outputs of the segmentation network, the classification network evaluates the qualities of segmentation output and encourages the segmentation network to generate more accurate output. To boost the segmentation performance, we also introduce simple yet effective technique to train the classification and refine the saliency map. Our experiment results demonstrate that our approach is effective in solving weakly supervised semantic segmentation.

In the second part of dissertation, we introduce a superpixel discovery method that generates semantic-aware superpixels. Our superpixels have new properties that the apart pixels can be grouped into a superpixel if they have similar semantic features. Also, the number of superpixels depends on the complexity of images, not the predefined number. Our superpixel expresses semantically similar group of pixels with a very small number of superpixels. We train the segmentation network using superpixel-guided seeded region growing technique which improves the qualities of initial seed. Our extensive experiments show that our approach achieves competitive segmentation performance with the state-of-the-arts in weakly supervised semantic segmentation.

keywords: Image Classification, Image Semantic Segmentation,
Weakly Supervised Semantic Segmentation, Deep Neural Network,
Visual Attention, Superpixel
student number: 2015-20897

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Backgrounds

Image semantic segmentation, a task to classify each pixel among the interested classes, is an important problem with a wide range of applications such as autonomous driving [1], medical diagnosis [2], industrial automation [3], and aerial imaging [4]. In recent years, deep convolutional neural networks have shown outstanding performances in image semantic segmentation [5, 6]. A main bottleneck of these approaches is that it requires large amount of fully-annotated data for training such networks. Since the acquisition of fully-annotated dataset is laborious and expensive, weakly supervised semantic segmentation (WSSS) has been suggested as an promising approach to mitigate the burden [7, 8]. There are various types of weak labels for semantic segmentation, for instance, image-level labels [9], points [10], scribbles [11], and bounding boxes [12]. Among these weak labels, image-level labels are popularly used in WSSS for its simplicity [13, 14, 7]. In essence, image-level label denotes the existence of objects in an image. In this dissertation, we consider the problem of weakly supervised semantic segmentation using image-level label. Before going into details, we briefly review the basics of image classification and semantic segmentation.

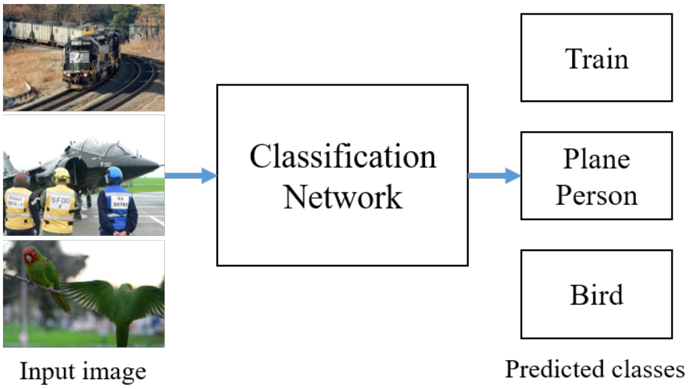


Figure 1.1: The classification network for multi-label classification

1.1.1 Basics of Image Classification

Image classification is a task to predict the classes of input images. In recent years, convolutional neural networks have enjoyed a great success in large-scale image recognition challenge [15, 16]. Image classification task can be roughly categorized into two types: single-label classification [17], multi-label classification [18]. In this dissertation, we use the convolutional neural networks to solve the multi-label classification in which multiple objects of multiple classes can exist in an input image. Fig. 1.1 shows our desired classification network for multi-label classification.

A typical approach to solving this problem is to train a classification network using the multi-label classification loss function. Let $z_c \in \mathbb{R}$ and $t_c \in \{0, 1\}$ be the output of classification network and the label for class c. One of the popularly used multi-label classification losses is the binary cross-entropy ℓ_{bce} which can be expressed as

$$\ell_{bce} = \frac{1}{C} \sum_c (-t_c \log(\sigma(z_c) - (1 - t_c) \log(1 - \sigma(z_c))) \tag{1.1}$$

where C is the number of classes and $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function. After training the classification network, we can predict the classes of input image by

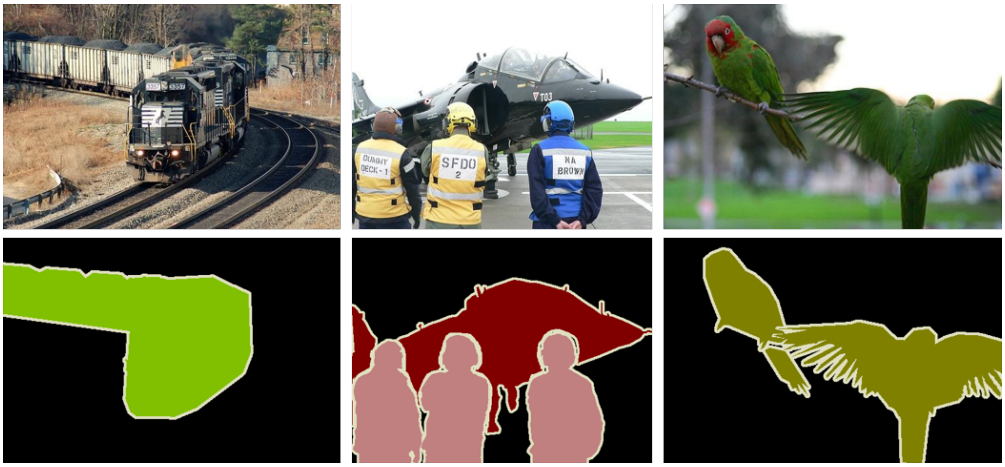


Figure 1.2: Input images and ground truths for semantic segmentation

making decision as following:

$$\begin{cases} \text{class } c \text{ exists} & \text{if } \sigma(z_c) > \tau \\ \text{class } c \text{ does not exist} & \text{if } \sigma(z_c) < \tau \end{cases} \tag{1.2}$$

where τ is the pre-defined threshold.

1.1.2 Basics of Image Semantic Segmentation

Image semantic segmentation is a task to assign classes to all pixels in input image, which can be seen as a pixel-level single-label classification (see Fig. 1.2 for examples of images and labels). Since the amount of required outputs is much larger than the outputs in classification, fully-convolutional networks are widely adopted to make dense prediction [19, 5].

The semantic segmentation network can be constructed by making a small change in the classification network. We first remove the global pooling layer of classification network, which summarizes the feature map into the scalar, and then, we replace the fully-connected layers with convolutional layers. By doing so, we can obtain the output for class c in the form of a map, that is, $Z_c \in \mathbb{R}^{h \times w}$ where h and w are the height and

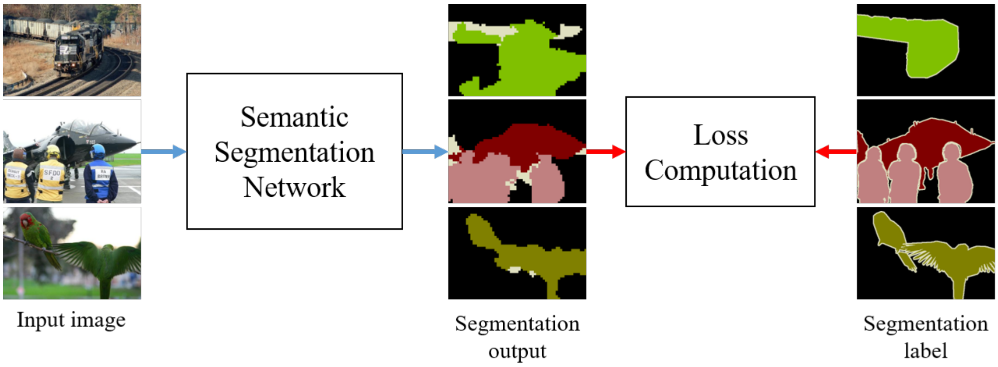


Figure 1.3: Training of the semantic segmentation network

width of output, respectively.

In general, the semantic segmentation network is trained using a pixel-level loss function (see Fig. 1.3). Let $Z_{u,c}$ and $T_{u,c}$ the output of segmentation network and label at position u for class c. A popular choice for segmentation loss is the softmax cross-entropy ℓ_{sce} which can be expressed as

$$\ell_{sce} = \frac{1}{hw} \sum_{u} \sum_{c} -T_{u,c} \log(\text{softmax}(Z_{u,c})) \tag{1.3}$$

where $\text{softmax}(x_c) = e^{x_c} / (\sum\limits_{c} e^{x_c})$ is the softmax function.

After the training process, we infer the segmentation output from input images to assign the best class to each pixel. For input image I, let $\widehat{Z} \in \mathbb{R}^{h \times w}$ be the segmented output. Then, the final output P is determined by finding the class that have the maximum output value at each pixel, that is,

$$P_{u,c} = 1 \text{ if } c = \arg\max_{c} Z_{u,c} \tag{1.4}$$

where the class 0 denotes the background class.

The commonly used performance metric of semantic segmentation is the intersection over union (IOU). The IOU of class c is computed as

$$IOU(c) = \frac{\sum\limits_{u} \mathbf{1}_{intersection(u,c)}}{\sum\limits_{u} \mathbf{1}_{union(u,c)}} \tag{1.5}$$

where

$$\mathbf{1}_{intersection(u,c)} = \begin{cases} 1 & \text{if } P_{u,c} = 1 \text{ and } T_{u,c} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{1.6}$$

and

$$\mathbf{1}_{union(u,c)} = \begin{cases} 1 & \text{if } P_{u,c} = 1 \text{ or } T_{u,c} = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{1.7}$$

After computing IOUs for all classes including the background class, we compute mean IOU for the final performance metric as

$$mIOU = \frac{1}{1+C} \sum_{c} IOU(c) \tag{1.8}$$

1.1.3 Basics of Weakly Supervised Semantic Segmentation

Weakly supervised semantic segmentation aims to train the segmentation network using image-level label (see Fig. 1.4 for desired system). A main concern in this approach is the fact that image-level label does not provide pixel-level information such as object shapes and their classes, which are required to train the segmentation network. In early work, the segmentation network is directly trained using image-level label with constraints [20]. Recently, the class activation mapping technique [21] is widely exploited to identify object regions from the classification network. We briefly review the popular procedure for solving the problem of WSSS (see Fig. 1.5).

Suppose we have a trained the classification network which predicts the output accurately. Then, although the classification network outputs a scalar probability that objects of each class exist in image, this network might have knowledge about which regions of image are important to make a prediction for each class. To extract these regions, we use a technique called class activation mapping [21]. Specifically, for the classification output z, let z^{-1} be the second last layer of classification network. The relation between z and z_{-1} can be expressed as

$$z = f(GAP(z^{-1})) \tag{1.9}$$

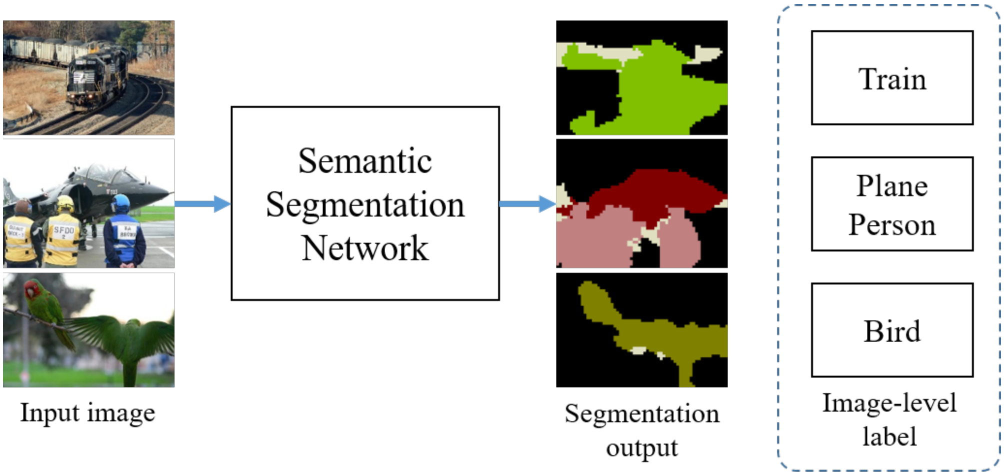


Figure 1.4: A brief illustration for WSSS system

where $f(\cdot)$ is the last fully-connected layer and $GAP(\cdot)$ is the global average pooling layer. The class activation map (CAM) is obtained as

$$M = f'(z^{-1}) \tag{1.10}$$

where $f'(\cdot)$ is the convolutional layer modified from $f(\cdot)$. If the weight $\mathcal{W}$ of $f(\cdot)$ is of size $C_0 \times C$ where C_0 is the input dimension of $f(\cdot)$, $f'(\cdot)$ can be constructed by using the convolutional weight $\mathcal{W}'$ of size $C_0 \times 1 \times 1 \times C$ (i.e., 1×1 convolution) which has the same parameters as $\mathcal{W}$.

After obtaining CAM, we generate the pseudo-label which will be used in training of the segmentation network. Specifically, the CAM M is normalized for each class as

$$M'_c = \frac{M_c}{\max_c M_c}. \tag{1.11}$$

Then, the pseudo-label is generated using the confident pixels in M'_c, that is,

$$T_{u,c} = \begin{cases} 1 \quad \text{if } M'_{u,c} > \tau_2 & \text{and} \qquad c = \arg\max_c M'_{u,c} \\ 0 & \text{otherwise} \end{cases} \tag{1.12}$$

where τ_2 is the pre-defined threshold for confident foreground pixel. The background

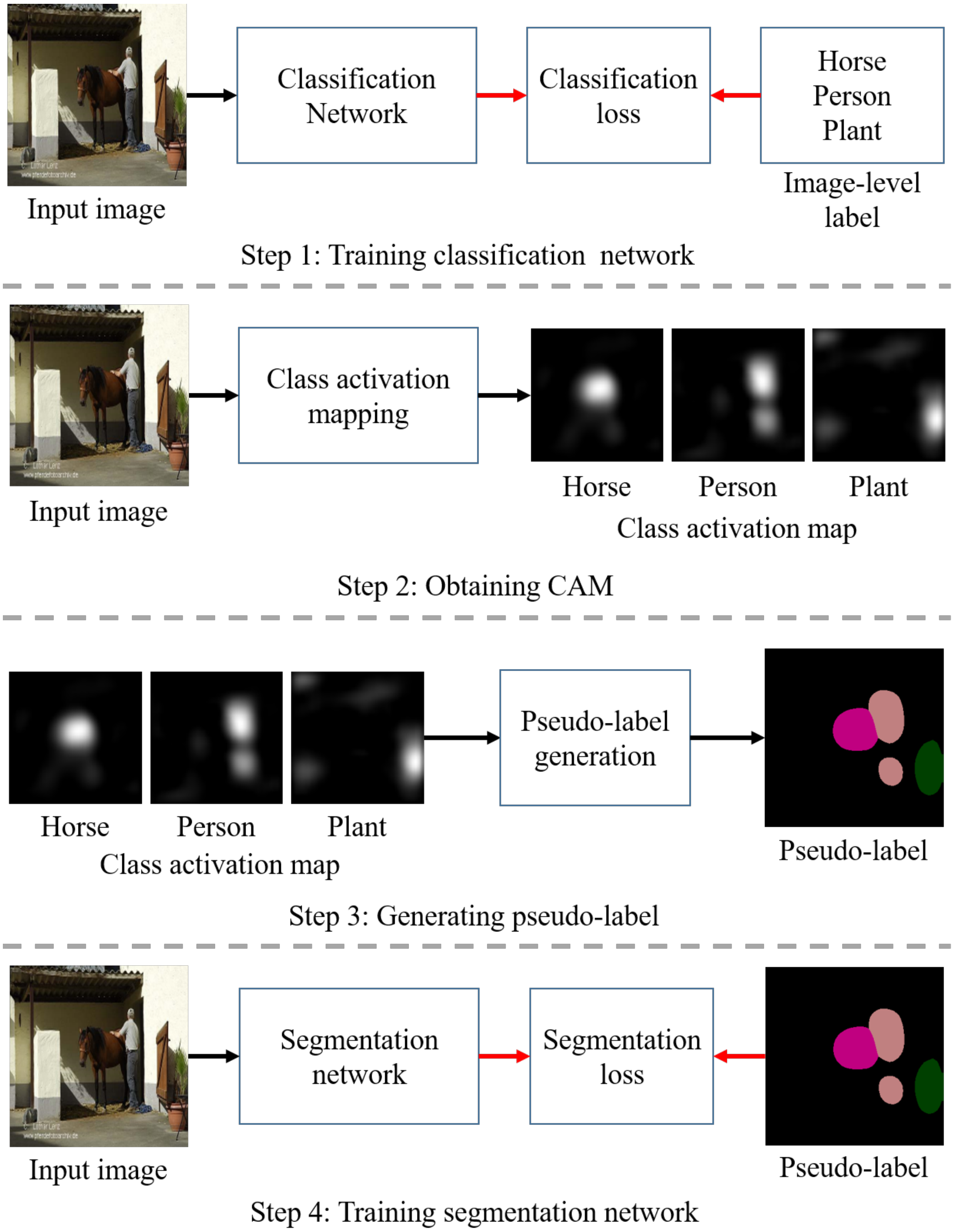


Figure 1.5: A general procedure for WSSS

regions are found as

$$T_{u,0} = \begin{cases} 1 & \text{if } \max_{c} M'_{u,c} < \tau_3 \\ 0 & \text{otherwise} \end{cases} \tag{1.13}$$

where $T_{u,0}$ is the pseudo-label for background class at position u and τ_3 is the predefined threshold for confident background pixel. In the obtained pseudo-label, there could be some pixels not labeled as any classes. These pixel are ignored in training of the segmentation network. Using the psuedo-label, we train the segmentation network as described in 1.1.2.

1.2 Contribution and Organization

In this dissertation, we introduce novel approaches to the problem of weakly supervised semantic segmentation.

In Chapter 2, we propose a novel WSSS technique that can train the semantic segmentation network without relying on the pseudo-label. Inspired by the visual attention mechanism of the human visual system, we apply image masking technique to limit the visible regions in image. By masking the input image using the masks predicted by the segmentation network and delivering the masked image to the classification network, we can evaluate the qualities of the segmented output and penalize the segmentation network. To train the segmentation network, we adopt a novel combination of two complementary losses: attention loss and saliency loss. The attention loss encourages the segmentation network to predict correct class in object regions and the saliency loss encourages the network to recognize which pixels belong to either background or foreground regions. To boost the segmentation performance, we also propose a training strategy for classification network and saliency map refining technique. We train the classification network using the self-supervision provided by dilated convolutional blocks so that the classification network can detect the objects and their parts better. We refine the noisy saliency maps based on the CAM so that we can find missing object regions or erase the false activations.

In Chapter 3, we propose a superpixel discovery method and the segmentation network training technique using the superpixel. Recent WSSS approaches have been relying on saliency map for additional pixel-level information that cannot be obtained from image-level label. However, such saliency detection methods requires pixel-level annotation for training process. To relieve the dependency on saliency maps, we propose a superpixel discovery method that finds semantically similar pixels based on the feature obtained from the self-supervised vision transformer, in particular, DINO [22]. The proposed superpixel has two following properties: 1) the superpixel contains long-range information even if the consisting pixels are not connected and 2) the number of superpixel depends on the complexity of an input image. We introduce superpixel-guided seeded region growing for training of the segmentation network. During the training process of the segmentation network, the initial seed is refined based on the segmented output and superpixel. Although the labeled regions in the initial seed are very sparse, we can obtain dense and high-quality labels as the segmentation network is trained.

In Chapter 4, we summarizes the contributions of the dissertation and discuss the future research directions based on studies of this dissertation.

Chapter 2

Weakly Supervised Semantic Segmentation Using Image Masking

Weakly-supervised semantic segmentation (WSSS) aims to train a semantic segmentation network using weak labels. Recent approaches generate the pseudo-label from the image-level label and then exploit it as a pixel-level supervision in the segmentation network training. A potential drawback of conventional WSSS approaches is that the pseudo-label cannot accurately express the object regions and their classes, causing a degradation of the segmentation performance. In this chapter, we propose a new WSSS technique that trains the segmentation network without relying on the pseudo-label. Key idea of the proposed approach is to train the segmentation network such that the object erased by the segmentation map is not detected by the classification network. From extensive experiments on the PASCAL VOC 2012 benchmark dataset, we demonstrate that our approach is effective in solving the problem of WSSS.

2.1 Introduction

Image semantic segmentation, a task to classify each pixel among the interested classes, is an important problem with a wide range of applications such as autonomous driving, medical diagnosis, industrial automation, and aerial imaging [1, 2]. Recently, deep

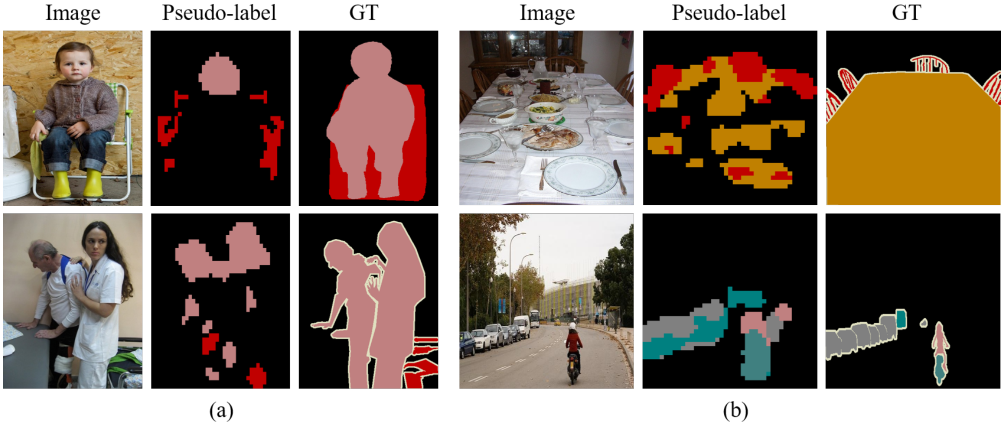


Figure 2.1: Problems of pseudo-labels obtained from CAM.

neural networks (DNN)-based semantic segmentation has received special attention due to its excellent segmentation performance [19, 6]. A potential drawback of the DNN-based approach is that a large number of fully-annotated data are needed to train the networks. Since the generation of fully-annotated dataset is laborious, alternative approaches such as unlabeled or weakly-labeled learning have been suggested in recent years [7, 23]. There are various forms of weak labels such as image-level labels [9], points [10], scribbles [11], and bounding boxes [12]. Among these, image-level label is popularly used for its simplicity [13, 14, 8]. In essence, image-level label indicates whether the foreground objects appear in an image or not (e.g., bird is in an image and cat is not). We henceforth refer to the DNN-based semantic segmentation using the image-level labels as weakly-supervised semantic segmentation (WSSS).

A central challenge of WSSS is that the image-level labels do not provide information on object regions required to train the semantic segmentation networks. A simple way to localize object regions is to use class activation mapping [21]. Basically, this approach figures out what regions in the image are relevant to the semantic classes. The localization map obtained from this technique, called *class activation map* (CAM), indicates the discriminative object regions. In recent WSSS approaches, CAM is used to generate a pseudo-label for the training of semantic segmentation network [7, 24]. While the pseudo-label can well express the object region of interest, it might cause some potential problems hindering the accurate image segmentation. First, the object extent in the non-discriminative region is not accurately expressed (see Fig. 2.1-(a): The labeled regions are focused on the most discriminative object regions (e.g., face of person)). This is because the classification network focuses only on the existence of the objects so that the network tends to ignore the non-discriminative regions which are also parts of the objects. Second, the class assigned in each pixel of the pseudo-label might not be correct when an image contains multiple objects with distinct classes (see Fig. 2.1-(b): The labeled regions are misclassified) since the CAMs are spread to unwanted regions outside the foreground objects. For these reasons, an approach that trains the semantic segmentation network using the pseudo-label might not achieve the satisfactory performance in many practical scenarios.

An aim of this chapter is to propose a novel WSSS technique that can train the semantic segmentation network without relying on the pseudo-label. Basically, our approach is inspired by the visual attention mechanism of the human visual system (HVS) [25]. When HVS perceives the visual information, HVS focuses on the desired object without being interfered by other objects. In order to mimic the human behavior and thereby reduce the interference from irrelevant regions, we mask an input image using an attention map that guides which pixels to attend or ignore. Specifically, a segmentation network generates the segmentation map describing the discovered object regions. Then, the attention map is generated by collecting the discovered regions in the segmentation maps of interesting classes. We exploit the attention map in erasing the discovered regions and therefore focus on the remaining regions in the masked image.

In order to check whether the objects are erased properly, we employ a classification network trained for multi-class multi-label classification. In summary, in the training process, the segmentation network tries to generate the segmentation map covering the object regions. Then, for the image masked by the segmentation map, the classification network tries to find out the interesting objects. For example, when an image contains bird and car, a segmentation network is guided to generate an accurate map of bird or car. If the generated segmentation map contains the bird, then the bird is erased in the masked image, helping the detection of a car in the classification network.

To train the segmentation network in the absence of the pseudo-label, we adopt a novel combination of two complementary loss functions: *attention loss* and *saliency loss*. The attention loss is used to penalize the segmentation network if the segmentation map does not completely cover the objects of a target class. The saliency loss is used to encourage the segmentation network to learn the accurate object extent which cannot be identified by the classification network. By learning the object classes using the attention loss and object extent using the saliency loss, the segmentation network can segment the image without obtaining the class-specific knowledge from pixel-level supervision.

As a means to enhance the segmentation performance, we propose a training strategy for the classification network and a refining technique for the saliency map. First, for the training of the classification network, we exploit the dilated convolutional blocks (see Fig. 2.3). The dilated convolutional blocks are used to find out the object regions outside the most discriminative regions. The regions discovered by dilated convolutional blocks are then used as an additional supervision for the classification network in finding out complete object regions. Second, we refine the saliency map using the CAM obtained by the classification network (see Fig. 2.4). Note that the value in each pixel of the CAM indicates the probability of an object being contained in that pixel. Using these values, we can find out the missing objects and also remove the unwanted objects in the saliency map.

The main contributions of this chapter are as follows:

- We propose a novel segmentation technique for weakly-supervised semantic segmentation. In our work, instead of learning the class-specific knowledge from the pseudo-label, the segmentation network learns the class-specific knowledge directly from the classification network by exploiting the image masking technique.

- We propose a training strategy for the semantic segmentation network. In the proposed approach, the segmentation network is trained using the combination of the attention loss and the saliency loss to accomplish the semantic segmentation task (see Section 2.3.3).
- From numerical experiments on *val* and *test* of the PASCAL VOC 2012 semantic segmentation benchmark [18], we show that our approach achieves mean-intersection-over union 65.5% and 65.4% using VGG16-based network and 67.9% and 68.2% using ResNet101-based network, respectively, which are competitive with the state-of-the-arts.

2.2 Related Work

2.2.1 Weakly-Supervised Semantic Segmentation

Image-level label has been used in many WSSS approaches due to its simplicity. Early works include multiple-instance learning [13], constrained optimization [20], and expectation-maximization techniques [26]. Recently, the class activation mapping technique that finds out the most discriminative object regions has been used to generate a pixel-level pseudo-label from the image-level label [21]. The generated pseudo-label depicting the reliable object regions is used as a supervision for the semantic segmentation network. The segmentation performance of this approach depends strongly on the accuracy of the generated pseudo-labels. Hence, it is of importance to find out accurate object regions for the proper training of the semantic segmentation network.

In order to obtain a reliable pseudo-label, various segmentation techniques have been proposed. In [7], fully-connected conditional random field (CRF) is applied to the predicted segmentation maps to refine the object boundaries. In [8], seeded region growing technique is used to assign classes to unlabeled pixels. Recently, approaches generating the reliable pseudo-labels without relying on the segmentation algorithms have been proposed. In [27], for example, a large number of localization maps are generated and then aggregated into a single localization map. In [28], the localization maps are accumulated through the training process to collect the discriminative regions of different parts in the objects. In [29], multiple dilated convolutional blocks are used to enlarge the receptive fields and transfer the discriminative information to the non-discriminative regions. In [30], adversarial manipulation technique is used to expand the discriminative object regions.

In a nutshell, the proposed approach is a bit similar to the CAM-based approach in the sense that we find out the object regions from the CAM. However, the key distinctive point of the proposed approach is that the segmentation network learns the classes of pixels by directly utilizing the classification network in the training of the segmentation network. As a result, we can train the segmentation network without relying on the pseudo-label.

2.2.2 Visual Attention

Visual attention, an approach to select the search regions and analyze their effects, has been applied to various computer vision tasks such as image classification [31], object detection [32], and image caption generation [33]. In the semantic segmentation, visual attention is often implemented using the image masking, a technique to erase part of an image. In many approaches, discovered object regions are erased to help the discovery of new object regions [34, 35, 36, 37]. For example, in [34, 35], discovered object regions are repetitively erased to find out new object regions. In [38], an approach to find out the object regions using an adversarial network has been proposed. In [36], two-phase learning strategy has been proposed to get a complete region of the foreground objects from the attention maps of two networks. The drawback of these approaches is that it is difficult to figure out whether the masked image still contains part of foreground objects or not. As a consequence, one might simultaneously find out unwanted background objects and the main foreground objects (e.g., water with boat, rail with train). In [37], discriminative object regions are erased to guide the network to find out new object regions. In [39], discriminative object regions are suppressed to spread the attention of the network to adjacent non-discriminative object regions.

In [40, 24], visual attention mechanism is applied to the adversarial learning. In these approaches, an attention map obtained from the main network is used to mask an input image and then the masked image is delivered to the adversarial network. By training the network using the adversarial loss function, the main network is encouraged to generate an attention map which makes the adversarial network output consistent with the image-level label. In [40], an adversarial network is used to discriminate whether the input map is ground truth or generated from the segmentation network. In [24], an input image is masked by the self-attention map. The masked image is passed to the adversarial network to check if the attention map covers regions contributing to the classification output.

2.2.3 Saliency Detection

The main goal of the salient object detection is to identify the visually distinctive objects (or regions) in an image and then segment them out from the background. Since the image-level label does not contain any information on the background regions in WSSS systems, one cannot directly find out the confident background regions using the classification network. To overcome this limitation, the saliency map has been widely used in many WSSS approaches [7, 24, 8, 27, 28]. Key idea of these schemes is to identify the background regions using the pixels with low salient probabilities.

In [41, 42, 43, 44, 45], the saliency map is directly used in the training process of the segmentation networks. For example, in [41], the segmentation network is trained using the saliency maps of simple images to generate the pseudo-labels for complex images. In [42], saliency maps are used to supplement non-discriminative object regions.

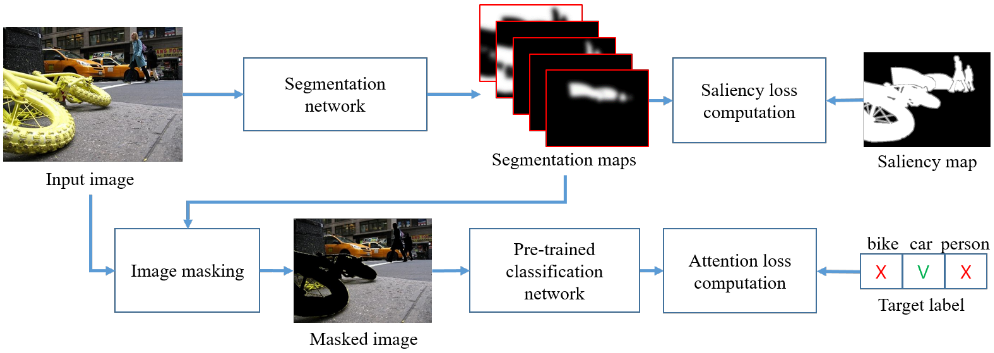


Figure 2.2: The overview of the proposed method to train weakly-supervised semantic segmentation network.

In [43], saliency maps are exploited to guide the seeded region growing method. In [44], saliency-guided self-attention module is used to capture rich contextual information for discovering the integral extent of objects and retrieving high-quality pseudo-label. In [45], an approach that trains the network using pixel-level feedback from combination of saliency maps and image-level labels has been proposed.

2.3 Proposed Weakly-Supervised Semantic Segmentation Network

In this section, we discuss the proposed WSSS framework. We first discuss the classification network training using dilated convolutional blocks and then discuss the refinement of the saliency maps using CAMs obtained from the classification network. We also explain how to train the semantic segmentation network using the image masking technique. The overall network architecture is illustrated in Fig. 2.2

2.3.1 Training of Classification Network

A classification network is a key ingredient in our approach. Basically, the classification network is trained using the multi-class multi-label classification loss. One well-known problem in the conventional classification network is that the network cannot detect the non-discriminative object regions. To address this issue, we use an extra supervision on the non-discriminative object regions in the training of the classification network. To find out the non-discriminative object regions, we use a dilated convolution which enlarges the receptive field without changing the computational cost [5]. With the increased receptive field, the information in the discriminative object regions can be transferred to distant regions, helping the detection of the non-discriminative object regions.

In the training of the classification network, we append dilated convolutional blocks to the classification network (see Fig. 2.3). The dilated convolutional blocks are similar to the standard convolutional block except that their first convolutional layers have unique dilation rates d. Let M^0 be the CAM obtained from the standard convolutional block and $M^1, \cdots, M^D$ be the CAMs obtained from D dilated convolutional blocks. Then, the object regions found by multiple dilated convolutional blocks are added to M^0 using the max-fusion to supplement the non-discriminative object region. A dense CAM, denoted as M, covering the discriminative and non-discriminative object regions is obtained as $M = \max(M^0, \frac{1}{D}\sum_{i=1}^{D}(M^i))$.

The classification network is trained using the multi-class multi-label classification loss and the CAM loss. First, the multi-class multi-label classification loss ℓ_{sig} is

$$\ell_{sig} = \frac{1}{C}\sum_{i\in\mathcal{D}}\sum_{c\in\mathcal{C}}\Big(-t_c\log(\sigma(\widehat{z}_{ic})) - (1-t_c)\log(1-\sigma(\widehat{z}_{ic}))\Big), \qquad (2.1)$$

where C is the number of foreground classes, $\mathcal{D}$ is the set of indices of convolutional blocks, $\mathcal{C}$ is the set of indices of foreground classes, t_c is the image-level label for class c, $\widehat{z}_{ic} = \text{GAP}(M_c^i)$ is the predicted class score for class c (GAP is the global average pooling operation), and $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function. Second, the CAM

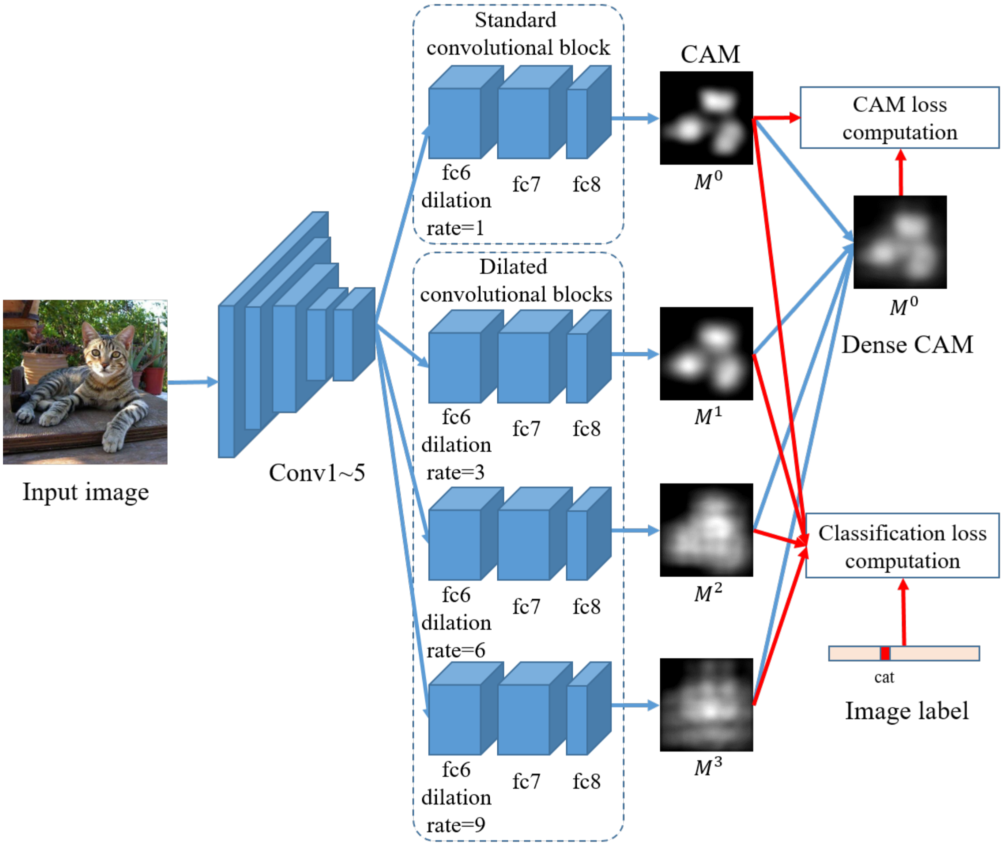


Figure 2.3: The architecture for the training of the classification network using dilated convolutional blocks.

loss ℓ_{cam}, used to match the CAM M^0 to the dense CAM M, is the mean square error (MSE) between M and M^0:

$$\ell_{cam} = \frac{1}{C|\mathcal{S}|} \sum_{u \in \mathcal{S}} \sum_{c \in \mathcal{C}^p} (\phi(M_{u,c}) - \phi(M^0_{u,c}))^2 \tag{2.2}$$

where $\mathcal{S}$ is the set of all positions, $\mathcal{C}^p$ is the set of indices of the present classes, and $\phi(x) = \max(0, x)$ is the ReLU activation function. Also, $M_{u,c}$ is the class score of class c at position u of class activation map M.

The overall loss ℓ_{cls} for training the classification network is

$$\ell_{cls} = \ell_{sig} + \lambda_{cls}\ell_{cam} \tag{2.3}$$

where λ_{cls} is the weighting factor for balancing two losses.

2.3.2 Saliency Map Refinement

In the segmentation network training, the saliency map is used to learn which pixels belong to either background or foreground regions. While the saliency detector (SD) can find out the detailed shape of the objects, it might also find out unwanted background objects or miss interesting foreground objects since SD is trained without the semantic classes. To overcome this potential drawback, we correct the pixels in the saliency map based on the CAM score. The score in each pixel indicates the probability of an object being contained in that pixel. Since the object detection in the classification network is fairly accurate, we can readily find out the missing foreground regions from the high-scored pixels in the CAM. Note that this does not necessarily mean that the low-scored pixels belong to the background regions since these pixels might belong to the non-discriminative object regions. From our extensive experiments, we observe that correcting these pixels to the background pixels causes a degradation of the segmentation performances. In our work, we set pixels with low scores to unlabeled pixels.

In Fig. 2.4, we illustrate the overall procedure of refining the saliency map. We first obtain the CAM of an input image from the classification network. To improve

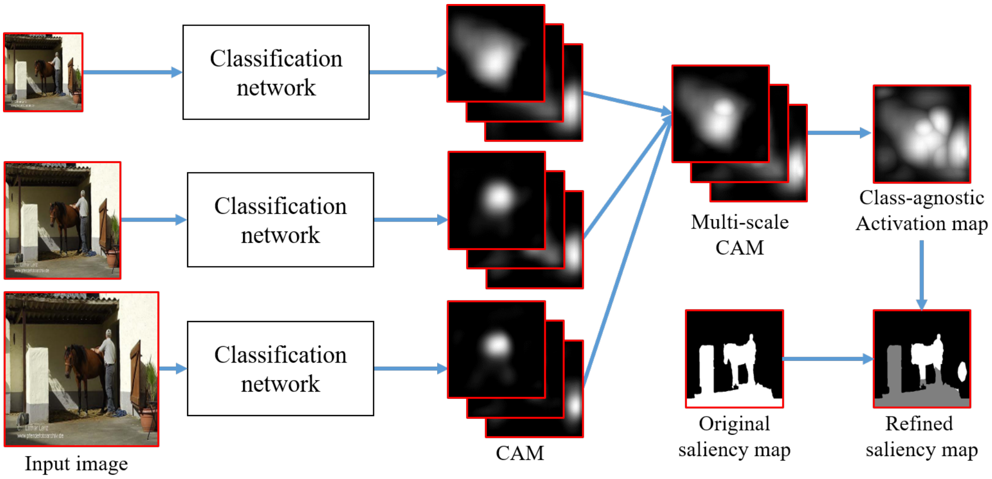


Figure 2.4: Overall procedure to refine the saliency map using the CAM.

the reliability of the CAM, we merge the CAMs of multiple scaled input images. Let $M^0(s_i)$ be the CAM of an input image scaled by a factor s_i $(s_i \in \{s_0, \cdots, s_n\})$, then the reliable CAM M^* is obtained as

$$M^*_{u,c} = \max_i scale(\phi(M^0_{u,c}(s_i))) \tag{2.4}$$

where $scale$ is the scaling operator that changes the size of map to the size of the input image. To obtain a map expressing the foreground object regions, we merge the CAMs of present classes, generating a class-agnostic activation map B whose pixels indicate the probabilities of an object being contained in that pixel:

$$B_u = \max_{c \in \mathcal{C}^p} \frac{M^*_{u,c}}{\max\limits_u M^*_{u,c}}. \tag{2.5}$$

If B_u is larger than the pre-defined threshold τ_1 and the pixel u belongs to the background regions in the saliency map O, we consider this pixel as a foreground pixel. On the other hand, if B_u is smaller than the pre-defined threshold τ_2 and the pixel u belongs to the foreground regions, we consider this pixel as an unlabeled pixel. That is, the refined saliency map R is obtained as

$$R_u = \begin{cases} 1, & \text{if } B_u > \tau_1 \text{ and } O_u = 0 \\ \text{unlabeled}, & \text{if } B_u < \tau_2 \text{ and } O_u = 1 \\ O_u, & \text{otherwise.} \end{cases} \tag{2.6}$$

2.3.3 Training of Segmentation Network

For the training of the segmentation network, we use the saliency loss that encourages the segmentation network to learn the object regions from the saliency map. While the segmentation map has $C + 1$ classes, the saliency map has only two classes. To connect the segmentation map to the saliency map, we design the background map H^b and foreground map H^f from the segmentation map.

Let H_c be the segmentation map of class c ($c = 0$ denotes the background class). Then, the background map is $H^b = H_0$ and the foreground map is $H^f = \sum_{c \in \mathcal{C}^p} H_c$. Let $\mathcal{S}^b$ and $\mathcal{S}^f$ be the set of positions of background and foreground pixels in saliency map, respectively. Then, the saliency loss is defined as a weighted cross-entropy with only two classes (background and foreground):

$$\ell_{sal} = -\sum_{u \in \mathcal{S}^b} \frac{1}{|\mathcal{S}^b|} \log H^b_u - \sum_{u \in \mathcal{S}^f} \frac{1}{|\mathcal{S}^f|} \log H^f_u, \tag{2.7}$$

where u is the position of the pixels. The weights for background and foreground pixels are set to $\frac{1}{|\mathcal{S}^b|}$ and $\frac{1}{|\mathcal{S}^f|}$, respectively, to balance the losses for background and foreground pixels. The first and second terms in (2.7) correspond to the loss for background and foreground classes, respectively. Note that the losses on unlabeled pixels in the saliency map are not computed during the training process. To improve reliability of the network in various scales, we feed the multiple scaled input images to the network and compute the losses individually. Thus, the resulting saliency loss is the sum of cross-entropy losses for $|S|$ scaled outputs (S is the set of input scales).

One potential weakness using the saliency loss is that the segmentation network might predict the class of pixel incorrectly since the class of each pixel is unspecified

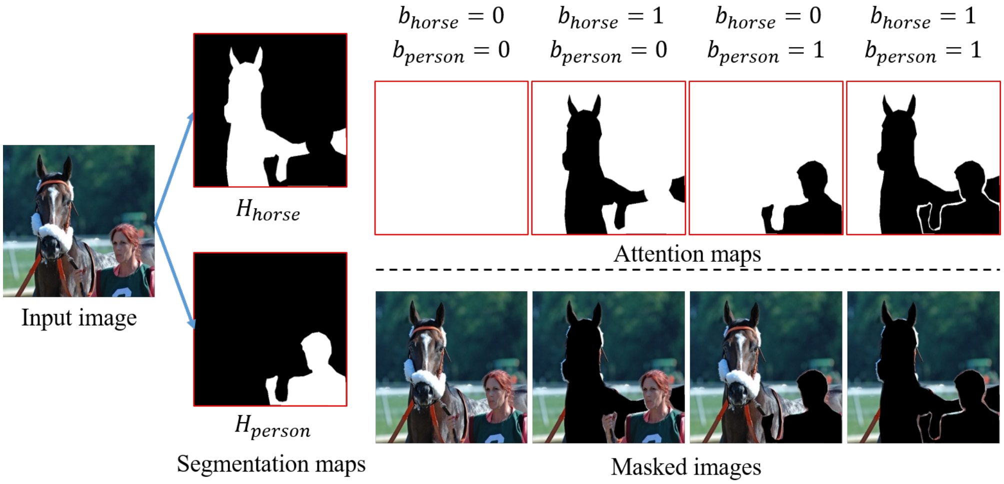


Figure 2.5: Illustration for examples of the attention maps and masked images.

in the saliency map. In order to make sure that the segmentation network predicts the correct class for each pixel, we exploit the image masking technique in the training of the segmentation network. During the training process, an input image is masked using an attention map F that designates which regions are erased. The attention map is obtained from the predicted regions in the segmentation map:

$$F = 1 - \sum_{c \in \mathcal{C}^p} H_c b_c \tag{2.8}$$

where b_c is the binary random number that decides whether the segmentation map H_c is erased in the attention map or not. Using the F, the masked image I' can be expressed as the product of the input image I and the attention map F:

$$I' = F \odot (I - \mu) \tag{2.9}$$

where $\odot$ is the element-wise multiplication and μ is the RGB mean of the training images. For a given class c, when $b_c = 1$, we expect that the objects of class c are erased in I'. Whereas, when $b_c = 0$, we expect that the objects of class c remain in I'. Hence, it is natural to choose $t' = t(1 - b)$ as the modified label corresponding to I'. We illustrate the attention maps and masked images corresponding to b in Fig. 2.5.

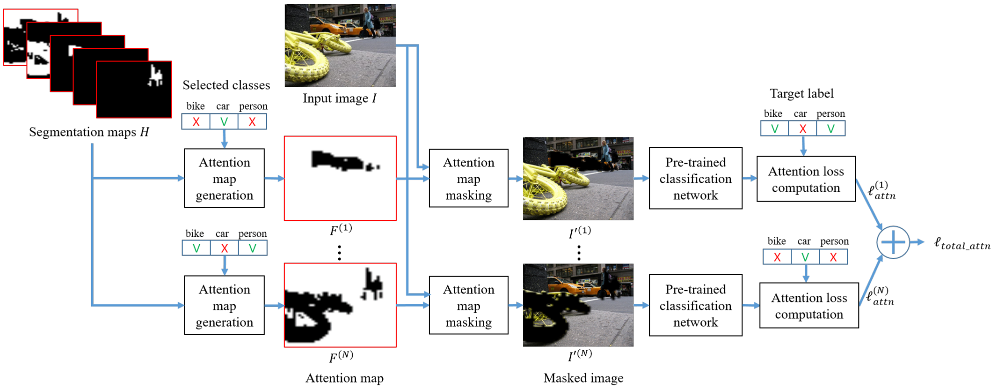


Figure 2.6: Illustration for the attention loss computation

The segmentation network is trained to predict the correct object regions so that the class score corresponding to I' matches t'. An associated loss, the attention loss ℓ_{attn} is defined as the cross-entropy between the class score $\widehat{z'}$ of I' and the target label t':

$$\ell_{attn} = \frac{1}{|\mathcal{C}^p|} \sum_{c \in \mathcal{C}^p} \Big(-t'_c \log(\sigma(\widehat{z'}_c)) - (1 - t'_c) \log(1 - \sigma(\widehat{z'}_c)) \Big). \tag{2.10}$$

In contrast to the classification loss ℓ_{sig}, the attention loss only considers the present classes. By generating multiple masked images using different attention maps, we can investigate the effects of different combinations of the segmentation maps. As illustrated in Fig. 2.6, we can add additional classification paths for other masked images generated using different attention maps. In each path, the same classification network is employed to compute the class score and the attention loss individually. The total attention loss is computed as the average of the attention losses:

$$\ell_{total\_attn} = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} \ell^{(n)}_{attn} \tag{2.11}$$

where N is the number of classification paths.

In summary, an overall loss for training the semantic segmentation network is

$$\ell_{seg} = \ell_{sal} + \lambda_{seg} \ell_{total\_attn} \tag{2.12}$$

where λ_{seg} is the weighting factor for balancing two losses in the segmentation loss. Note that during the training process of the segmentation network, we fix the parameters of the classification network to keep its learned knowledge.

2.4 Experiments

2.4.1 Dataset and Experiment Settings

We evaluate the proposed approach on the PASCAL VOC 2012 segmentation benchmark dataset [18] which has 20 foreground classes and one background class. This dataset has 1,464 training images, 1,449 validation images, and 1,456 test images. As in many practices [6, 34], we use augmented training dataset consisting of 10,582 images [46]. In our experiments, we only utilize image-level annotations for the network training. We employ the saliency detector [47] to obtain saliency map that expresses class-agnostic pixel-wise object scores. As a performance measure, we use mean intersection-over-union (mIOU), average of IOUs over 21 categories. We obtain the result on the test set by submitting the predicted results to the official PASCAL VOC evaluation server.

For classification network, we employ VGG16 [15] pre-trained on ImageNet classification dataset [17]. As illustrated in Fig. 2.3, we replace the last three fully-connected (fc) layers in VGG16 with a standard convolutional block consisting of three convolutional layers. The convolutional blocks consist of two 3×3 convolutional layers (fc6 and fc7 both 1024 outputs) and one 1×1 convolutional layer (fc8). We append three dilated convolutional blocks to the classification network (see Fig. 2.3). The dilation rates in three dilated convolutional blocks are set to $d = \{3, 6, 9, 12, 15, 18, 21, 24\}$. The parameters of the standard and dilated convolutional blocks are initialized from the normal distribution. We apply the GAP layer after fc8 for the training of the classification network.

For segmentation network, we employ DeepLab-ASPP [6] whose backbone architecture is either VGG16 [15] or ResNet101 [16]. We initialize the parameters of VGG16-

and ResNet101-based DeepLab using the convolutionalized VGG16 and ResNet101 pre-trained on MS-COCO [48], respectively. For the last layer, the parameters are initialized from the normal distribution. When training the segmentation network, we use the classification network only with standard convolutional block (i.e., the dilated convolutional blocks are removed). In the training of the ResNet101-based DeepLab, we only update parameters of convolutional layers while fixing the parameters of batch normalization layers. The softmax output of the segmentation network is post-processed by CRF with default parameters [49].

To improve the robustness of the classification network and the segmentation network, we apply data augmentation techniques. We randomly flip and scale (from 0.5 to 1.5) input images. The resulting images are cropped to 321×321 at random location. We also apply color augmentation techniques by randomly changing brightness, contrast, saturation, and hue. We use multi-scale inputs with scales, $S = \{1, 0.75, 0.5\}$ in both training and test phases [50, 6]. We use 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 [51] with initial learning rate 10^{-3} and power 0.9, i.e., learning rate $= 10^{-3} \times (1 - \frac{iter}{maxiter})^{0.9}$. The learning rate for the last layers are multiplied by 10. We set the two thresholds τ_1 and τ_2 in (2.6) used to refine the saliency map to 0.8 and 0.3, respectively, which are found by grid search. The weighting factors λ_{cls} in (2.3) and λ_{seg} in (2.12) are set to 0.1 and 2, respectively. The entries of the binary random vectors b are drawn uniformly. We train the classification network and the segmentation network for 50 and 30 epochs, respectively. Our approach is implemented based on Tensorflow [52]. The classification network and the segmentation network are trained on a single NVIDIA GeForce Titan Xp.

2.4.2 Comparisons with state-of-the-arts

We compare the performance of the proposed method with that of state-of-the-art WSSS methods. In Tables 2.1 and 2.2, we summarize the mIOU obtained by VGG16- and

Table 2.1: Comparison of VGG16-based weakly-supervised semantic segmentation methods' mean IOUs on PASCAL VOC 2012 *val* and *test* set

Method	Use saliency map	Need pseudo-label?	Val	Test
SEC [7]	✓	✓	50.7	51.1
TPL [36]	✓	✓	53.1	53.8
AE-PSL [34]	✓	✓	55.0	55.7
DCSP[35]	✓	✓	58.6	59.2
GAIN[24]	✓	✓	55.3	56.8
MCOF[42]	✓	✓	56.2	57.6
AffinityNet [53]		✓	58.4	60.5
DSRG[8]	✓	✓	59.0	60.4
MDC [29]	✓	✓	60.4	60.8
FickleNet[27]	✓	✓	61.2	61.9
OAA [28]	✓	✓	63.1	62.8
RRM [54]		✓	60.7	61.0
SGAN [44]	✓	✓	64.2	65.0
SAFN [55]	✓	✓	61.9	62.3
ICD [56]	✓	✓	64.0	63.9
DRS [39]	✓	✓	63.5	64.5
GSM [**?**]	✓	✓	63.3	63.6
NSR [57]	✓	✓	65.5	65.3
ESP [45]	✓	✓	67.0	67.3
ECS-Net [37]		✓	62.1	63.4
Ours	✓	✗	**66.5**	**66.9**

Table 2.2: Comparison of ResNet-based weakly-supervised semantic segmentation methods' mean IOUs on PASCAL VOC 2012 *val* and *test* set

Method	Use saliency map	Need pseudo-label?	Val	Test
DCSP [35]	✓	✓	60.8	61.9
MCOF [42]	✓	✓	60.3	61.2
AffinityNet [53]		✓	61.7	63.7
DSRG [8]	✓	✓	61.4	63.2
FickleNet [27]	✓	✓	64.9	65.3
OAA [28]	✓	✓	65.2	66.4
RRM [54]		✓	66.3	66.5
SGAN [44]	✓	✓	67.1	67.2
SAFN [55]	✓	✓	61.9	62.3
ICD [56]	✓	✓	67.8	68.0
DRS [39]	✓	✓	71.2	71.4
GSM [**?**]	✓	✓	68.2	68.5
LIID [58]	✓	✓	66.5	67.5
NSR [57]	✓	✓	70.4	70.2
ESP [45]	✓	✓	71.0	71.8
advCAM [30]		✓	68.1	68.0
GCN [59]		✓	68.7	69.3
AuxSegNet [60]	✓	✓	69.0	68.6
ECS-Net [37]		✓	66.6	67.6
Ours	✓	✗	**69.0**	**69.2**

ResNet-based WSSS approaches. From the results, we observe that our approach performs competitive with the conventional WSSS approaches. Specifically, our approach achieves mIOU of 66.5% and 69.0% for *val* set of PASCAL VOC 2012 segmentation dataset with VGG16- and ResNet101-based DeepLab-ASPP, respectively. Using our approach, we can train the segmentation network such that it learns the class-specific knowledge directly from the classification network. Our results clearly demonstrate that the generation of pseudo-labels is unnecessary for WSSS.

We compare the proposed approach with a few notable WSSS approaches. In GAIN [24], since the main network and the adversarial network are sharing the parameters and also trained simultaneously, the network might be confused when the object regions are poorly discovered. Our approach can avoid this by training the adversarial network in advance and fixing the parameters in the network. MDC uses the classification network having multiple convolutional blocks to generate the pseudo-label [29] . In our approach, the classification network trained to predict the dense CAM is used for the training of the segmentation network directly. Similarly to the proposed approach, MCOF trains the segmentation network using the classification network [42]. In MCOF, the classification network is used to classify the superpixels of an input image. Whereas, in our approach, the pre-trained classification network is used to classify the regions of input image after applying image masking technique.

2.4.3 Ablation studies

In order to prove the effectiveness of each component, we conduct ablation experiments with different settings of the proposed work. In Table 2.3, we summarize the segmentation performance of the proposed approach in different settings. When we say ‘standard’ classification network, it means that the network trained only using multi-class multi-label classification loss function. The ‘dilation’ classification network means the network trained using the classification loss and the CAM loss described in Section 2.3.1.

Table 2.3: Comparison of performances on *val* set with different settings of our approach. GT indicates the ground truth saliency map.

Model	Segmentation network backbone	Saliency map refinement	Classification network	The number of classification paths	val w/o crf	val w/ crf
A1	VGG16	✗	none		62.0	62.2
A2	VGG16	✓	none		61.4	63.8
A3	VGG16	✗	standard	1	62.4	64.0
A4	VGG16	✗	dilation	1	62.9	64.7
A5	VGG16	✓	dilation	1	62.4	66.4
A6	VGG16	✓	dilation	2	62.0	**66.5**
A7	VGG16	✓	dilation	3	62.2	66.4
B1	ResNet101	✗	none		64.2	65.1
B2	ResNet101	✓	none		65.7	66.8
B3	ResNet101	✓	dilation	1	66.2	68.8
B4	ResNet101	✓	dilation	2	66.1	68.7
B5	ResNet101	✓	dilation	3	66.1	69.0
C1	VGG16	GT			66.2	67.5
C2	VGG16	GT	dilation	2	66.6	69.3
C3	ResNet101	GT			70.2	70.9
C4	ResNet101	GT	dilation	2	70.9	73.1

Table 2.4: Comparison of performances on *val* set with different settings of our approach. GT indicates the ground truth saliency map.

Model		mIOU
A1	baseline	62.2
A2	A1+refined saliency map	63.8
A3	A1+attention loss	64.0
A4	A3+dilated convolution	64.7
A5	A1+ all techniques	**66.5**

Our baselines are the VGG16- and ResNet101-based segmentation networks trained only using the saliency loss associated with the original saliency map obtained by SD [47] (see A1 and B1). From the results, we observe that the segmentation performance can be improved by refining the saliency map. Specifically, the models using the refined saliency map (A2 and B2) achieve about 3% improvement in mIOU over the baseline models. By comparing the performance of A1 and A3, we also observe that the segmentation performance can be improved by exploiting the classification network in the training of the segmentation network. Moreover, we can observe that the segmentation performance can be further improved by exploiting the dilated convolution-based classification network (see A3 and A4). We also observe that the performances can be enhanced by employing multiple classification paths (see A5 to A7 and B3 to B5). We also conduct experiments when high-quality saliency map is available. For these experiments, we use the ground truth saliency map obtained by binarized ground truth. We can observe that the segmentation network trained using the ground truth saliency map attains 67.5% and 70.9% with VGG16 and ResNet101 backbone, respectively. By applying our image masking-based approach to these networks, we can further improve the performance by 1.8% and 2.2% for the segmentation networks with each backbone.

To investigate the efficacy of refining saliency map, we conduct experiments using different saliency maps: 1) original saliency map obtained from saliency detector, 2) refined saliency map in which low-scored foreground pixels are corrected to background pixels, and 3) refined saliency map in which low-scored foreground pixels are considered as unlabeled pixels. From the results in Table 2.5, we observe that the segmentation performance is degraded when the low-scored foreground pixels are corrected to background pixels. We also observe that the segmentation performance is significantly improved when the low-scored foreground pixels are considered as unlabeled pixels.

To observe the effect of combination of input scale used for refining saliency maps, we conduct experiments by varying the number of the input scales. The input scales are chosen among the scales used in data augmentation $\{0.5, 0.75, 1, 1.25, 1.5\}$. From the results, we see that the best segmentation performance is obtained when three input scales are used (see S4 and S7 in Table 2.6).

We have conducted experiment using the MS-COCO dataset (see the results in Table 2.8). In this dataset, our segmentation network performs slightly worse than the conventional networks. The main reason for this is as follows; In our work, instead of generating the pseudo-label, we exploit the classification network in the training of the segmentation network. To train the segmentation network using the attention loss, the classification network should detect the objects and then output high scores for the corresponding classes for both input and masked images. Unfortunately, the VGG16-based classification network we used in the segmentation network training is not quite excellent in finding out small objects (i.e., fork, tie, and toothbrush) or the objects of rare classes (i.e., carrot, toaster, and hair-drier) so that the performance of the segmentation network for such objects is not so excellent. Nonetheless, the segmentation network could find out normal objects (i.e., person, animals, and vehicles) quite well.

We also test the performances for various number of dilated convolutional blocks D. From the results shown in Table 2.7, we observe that the segmentation performance

Table 2.5: Comparison of the segmentation performance using different saliency maps

	Saliency map	mIOU on val
1	original saliency map	60.7
2	refined saliency map without unlabeled pixels	59.9
3	refined saliency map with unlabeled pixels	65.5

Table 2.6: Segmentation performances with respect to the combination of input scales for refining saliency map.

Configuration	Input scales	mIOU on val
S1	$\{0.5, 0.75, 1, 1.25, 1.5\}$	66.0
S2	$\{0.5, 0.75, 1, 1.25\}$	66.0
S3	$\{0.75, 1, 1.25, 1.5\}$	66.1
S4	$\{0.5, 0.75, 1\}$	66.5
S5	$\{0.75, 1, 1.25\}$	65.7
S6	$\{1, 1.25, 1.5\}$	64.8
S7	$\{0.5, 1, 1.5\}$	66.5

slightly improves with the number of dilated convolutional blocks at the expense of the additional computations and training time.

2.4.4 Qualitative Results

In Fig. 2.7, we provide qualitative results obtained from ResNet101-based DeepLab-ASPP. (a): the baseline network trained only using the saliency loss with the original saliency map, (b): the network trained only using the saliency loss with the refined saliency map, and (c): the network trained using the attention loss in addition to the saliency loss with the refined saliency map. The bottom two rows show some failure cases. From the results, we can observe that our saliency map refining strategy is

Table 2.7: Segmentation performances with respect to different number of dilated convolutional blocks.

Model	# of dilated conv. blocks	Dilation rates	mIOU on val
D1	$D = 8$	$\{3, 6, 9, 12, 15, 18, 21, 24\}$	65.8
D2	$D = 4$	$\{6, 12, 18, 24\}$	65.6
D3	$D = 3$	$\{3, 6, 9\}$	65.5
D4	$D = 2$	$\{3, 6\}$	65.2

Table 2.8: Comparison of weakly-supervised semantic segmentation methods' mean IOUs on MS-COCO *val* set.

Method	Segmentation network	mIOU on val
SEC [7]	DeepLab-LargeFOV	22.4
DSRG [8]	DeepLab-ASPP	26.0
GSM [61]	DeepLab-ASPP	28.4
SGAN [44]	DeepLab-ASPP	33.6
EPS [45]	DeepLab-ASPP	35.7
Ours	DeepLab-ASPP	30.2

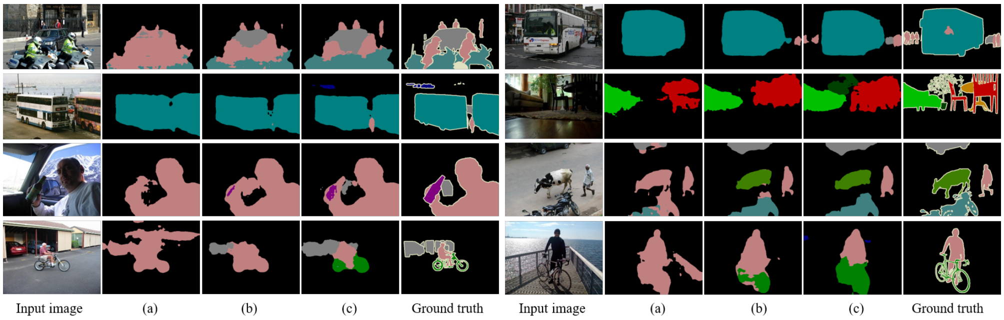


Figure 2.7: Qualitative results obtained from ResNet101-based DeepLab-ASPP.

effective in finding out the objects which might not be detected by SD and removing the falsely activated background objects. Also, we can observe that our image masking-based training strategy enables the segmentation network to learn the object classes precisely even when the objects are very small. Also, we would like to mention some failure cases. One of the most frequent failure scenarios is that there is an object which covers a large portion of the image. For example, sofa or table can be confused as background.

In Fig. 2.8, we provide qualitative results for the proposed approach and conventional approaches ((a): input image, (b): ground truth, (c): DRS [39], (d): GSM [**?**], (e): NSR [57], (f): ours). From the results, we observe that the proposed approach predicts the detailed object region (see the first three columns in Fig. 8) while the conventional approaches make false activation (see the last two columns in Fig. 2.8).

2.5 Summary of Chapter 2

In this chapter, we proposed a new WSSS technique that can train the segmentation network without pixel-level pseudo-labels. To prevent the performance degradation caused by inaccurate pseudo-label in conventional WSSS approaches, we have exploited the image masking technique in the training of the segmentation network. We also

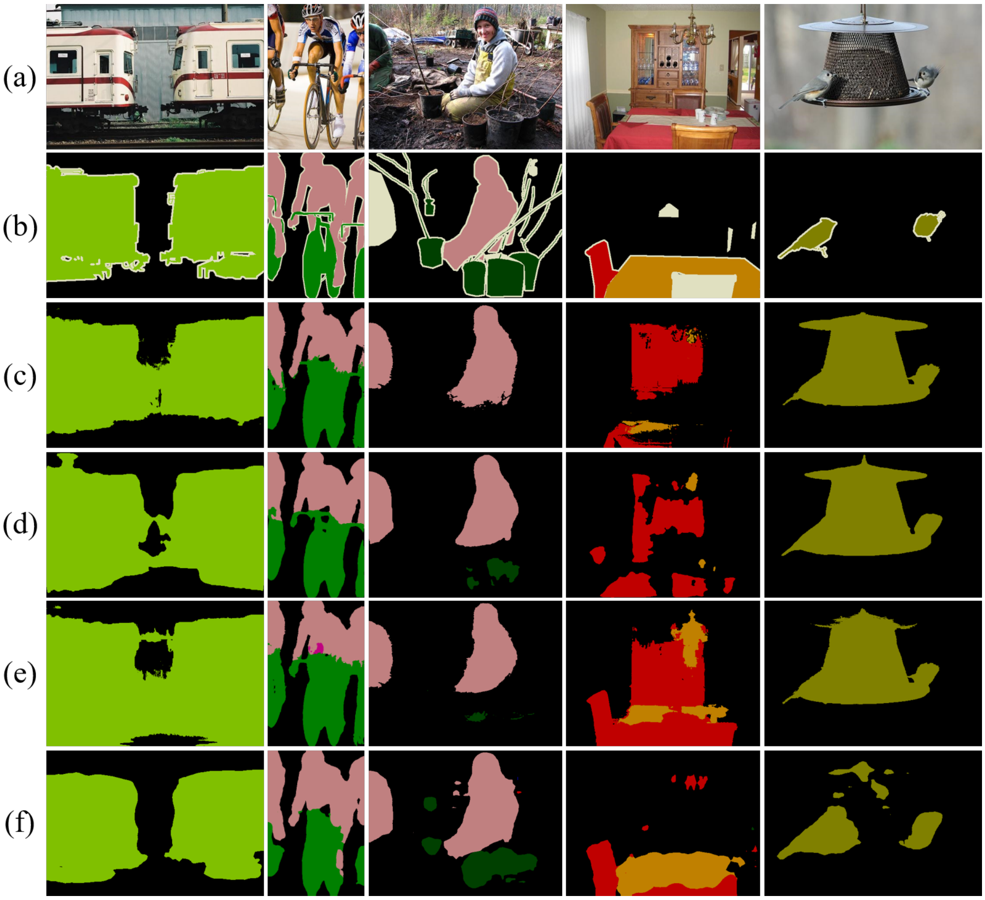


Figure 2.8: Qualitative results obtained from various WSSS approaches.

introduced an approach to refine the saliency map, which significantly improves the segmentation performance. Extensive experiments demonstrate that our approach is effective in solving the problem of WSSS.

Chapter 3

Weakly Supervised Semantic Segmentation Using Image Clustering

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 the information contained in image-level label is deficient in identifying accurate object regions, additional modules such as saliency detector have been exploited in weakly supervised semantic segmentation, which requires pixel-level label for training. In this chapter, 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 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 dataset for weakly supervised semantic segmentation.

3.1 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 and medical diagnosis [1, 2]. Recently, deep neural networks (DNN)-based semantic segmentation has received special attention due to its excellent segmentation performance [19, 6]. A main bottleneck of the DNN-based approach is that it requires large-scale data with dense annotation for training of the networks. Since the generation of fully-annotated dataset is laborious, one of the alternative approaches, weakly-labeled learning have been broadly studied [7, 8]. There are various forms of weak labels such as image-level labels [9], points [10], scribbles [11], and bounding boxes [12]. Among these, image-level label, indicating the existence of the objects, is popularly used due to its simplicity [13, 14, 8]. We henceforth refer to the DNN-based semantic segmentation using the image-level labels as weakly-supervised semantic segmentation (WSSS).

A main challenge of WSSS is to discover object locations and extent from image-level label. In recent WSSS approaches, class activation mapping method [21] is popularly used to locate the object regions for the training of semantic segmentation network [7, 24]. However, since the pseudo-label generated using this approach is sparse, there exist a performance gap between fully-supervised and weakly-supervised semantic segmentation. To bridge the performance gap, many recent WSSS approaches exploit the extra supervisions. One of the popular choice is the saliency map obtained by the saliency detectors. Although many WSSS approaches take the saliency map for granted from saliency detectors, it fundamentally requires massive effort on annotating detailed pixel-level label.

An aim of this chapter is to relieve the thirst for pixel-level information for WSSS. To this end, we approach WSSS problem by exploiting a vision transformer which is trained using only self-supervision. The vision transformer trained by distillation with no labels, DINO [22], have shown the performance comparable with the state-of-the-arts convolutional neural network models. In particular, the feature obtained by DINO

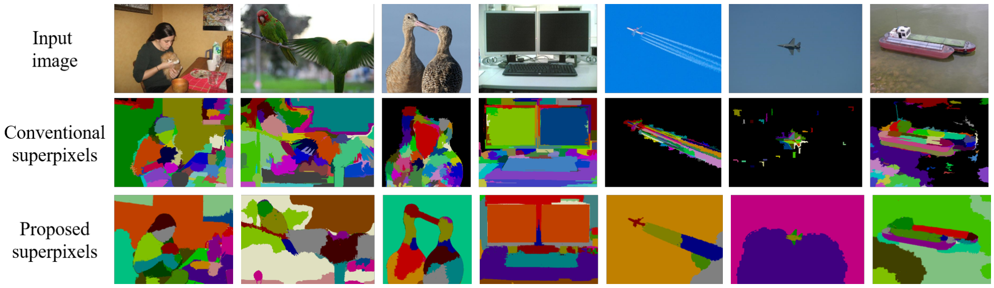


Figure 3.1: Examples for superpixels obtained by conventional and our method.

appear to contain explicit information about the semantic segmentation of objects in an image. Recently, this DINO-based feature has been exploited in the challenging computer vision tasks such as unsupervised object detection [62] or unsupervised saliency detection [63].

In this chapter, we propose a semantic-aware superpixel discovery method to resolve the problem of WSSS. In our approach, we use off-the-shelf ViT trained by DINO to obtain the feature without any fine-tuning. By iteratively identifying a seed pixel of an input image and discovering the pixels having similar feature to the seed pixel, we obtain the groups of pixels sharing semantic similarities in a unsupervised manner. In generating the superpixels, we only consider the pair-wise feature similarities between pixels. The generated superpixels have two following properties: 1) the superpixel contains long-range information even if the consisting pixels are not connected, 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, meaning that the number of superpixels is not pre-defined so that we can avoid oversegmentation. In Fig. 3.1, we show some examples for conventional (SLIC [64]) and our superpixels. Note that the colors are only used to illustrate the different superpixels.

After obtaining semantic-aware superpixels, we train the semantic segmentation network using superpixel-guided seeded region growing method. Using the rough initial seed as a main supervision to the segmentation network, the seeded regions are expanded to the neighboring superpixels. Unlike the conventional seeded regions growing method that gradually expand the seeded region to adjacent pixels [8], our method expands the seeded region to group of pixels if a criterion is satisfied. Moreover, since the superpixel keeps the shape of objects (or their parts), we can obtain high-quality seed that depicts the detailed object boundaries.

The contributions of this chapter are as follows:

- We propose a simple method to group the similar pixels using the self-supervised vision transformer in a unsupervised manner. Our method produces superpixels containing semantically similar pixels which are friendly to semantic segmentation task.
- In our approach, we train the semantic segmentation network using the initial seed labeled on confident pixels while refining the seed using 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-arts methods on PASCAL VOC 2012 and MS-COCO 2014 semantic segmentation dataset with only using image-level labels.

3.2 Related Work

3.2.1 Weakly Supervised Semantic Segmentation

The goal of WSSS is to train semantic segmentation network from coarse labels such as points, scribbles, or image-level label. Due to the simplicity, WSSS using image-level label is widely studied. A typical approach is to train a classification network and obtain initial seed using class activation mapping technique. Since the initial seed obtained by this approach is sparse, there have been many efforts to improve the qualities of seed. For examples, in [65], self-supervision based on equivariant attention mechanism is exploited to discover object regions. In [30], advCAM method is proposed to find out non-discriminative object regions in an anti-adversarial manner. In [66], an approach that encourages the network to perceive non-discriminative object region by reducing information bottleneck is proposed.

3.2.2 Superpixel

Superpixel is a set of homogeneous pixels based on features such as bright, color, or texture. To perform superpixel segmentation, graph-based method [64] or clustering-based methods [67] have been popularly exploited. The superpixels obtained from these methods are used in many WSSS approaches to recover smooth object boundaries [68, 69, 56, 70]. However, since the superpixels used in these approaches are quite over-segmented, having a few hundreds of segmented regions, it is difficult to obtain long-range information from these superpixels and discover the meaningful information for WSSS.

3.2.3 Seeded Region Growing

The seeded region growing [71] is an unsupervised approach to segmentation 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. To successfully accomplish segmentation of an image, it is important to locate the initial seed to proper pixels and use a criterion that can characterize the image regions. In [8], an approach that utilizes initial seed generated by classification network in training of semantic segmentation network and computes pixel similarity using high-level semantic features is proposed.

3.2.4 Transformer

Transformer and self-attention models have revolutionized machine translation and NLP fields. Recently, its adoption to computer vision, the vision transformer (ViT) [72], has shown great performance beyond convolutional neural network (CNN) models. However, in order to achieve such performance, the datasets containing enormous number of training images are required (e.g., JFT-300M dataset). As a way to alleviate this burden, self-supervision-based training technique is proposed [73]. In particular, in [22], 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.

3.3 Superpixel-guided Weakly Supervised Semantic Segmentation

In this section, we discuss the proposed WSSS framework. We first introduce how to discover semantic-aware superpixels from self-supervised vision transformer-based features. Then, we 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 superpixel-guided seeded region growing method.

3.3.1 Superpixel Generation

In our perspective, an appropriate superpixels for semantic segmentation should satisfy two following properties: 1) each superpixel is as large set as possible consisting 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 the pre-defined number. To obtain such superpixels, we first identify a pixel which will be seed of a superpixel and find out the pixels sharing similar semantic features to the seed pixel. In our approach, we vision transformer-based feature to perform superpixel discovery method. Before going into details, 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 self-attention layer of the last transformer block to obtain final features which are the main ingredient in discovering the superpixels.

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 binary adjacency matrix B denoting positive similarities between two pixels as

$$A_{pq} = \frac{f_p^T f_q}{\|f_p\|_2 \|f_q\|_2}, B_{pq} = \begin{cases} 1 & \text{if } A_{pq} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

where $\|\cdot\|_2$ is ℓ_2 norm.

The sum of 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 large object (e.g., sky, car, or building) or small object (e.g., bottle, eyes, or wheel). One of the ways to identify a group of pixels representing an object can be to select a pixel p^*, *a seed pixel*, and find out the pixels having similar semantic features to p^*.

We may wonder how to select a good seed pixel to find out a group of pixel, *a superpixel*. Here, we use simple rule based on the degree of pixels. We can consider to select p with either the highest or the lowest degree to find out large or small object,

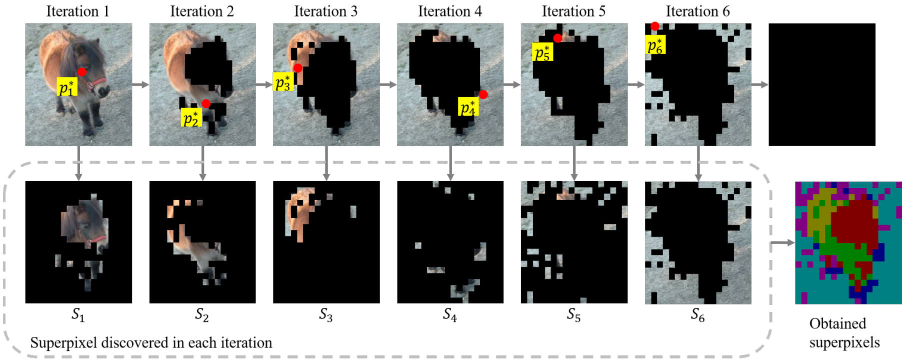


Figure 3.2: A procedure of the proposed superpixel discovery method.

respectively. From our extensive experiments, we observe that it is better to identify small objects than large objects since there could be a pixel having an overwhelming degree, resulting in grouping the most of pixels. Hence, our strategy to partition an image into multiple superpixels is to find out a superpixel corresponding to smallest object and repeat this process after excluding the pixels of the discovered superpixel from the candidates.

To sum up, in each iterative step i, the seed pixel p_i^* of a superpixel $\mathcal{S}_i$ is selected by finding the pixel with 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 empty set. In Fig. 3.2, we illustrate the procedure of the proposed superpixel method.

3.3.2 Initial Seed Generation

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

$$\ell_{cls} = \frac{1}{C}\sum_{c=1}^{C}\left(-y_c \log(\sigma(\widehat{x}_c)) - (1-y_c)\log(1-\sigma(\widehat{x}_c))\right) \tag{3.2}$$

Table 3.1: Pseudocode for the proposed superpixel discovery method

```
**Algorithm**: Superpixel discovery method
**Input**
$f_p \in \mathbb{R}^{D \times 1}$: feature at position $p$ in image $I$
$\tau \in [0, 1]$: pre-defined threshold
**Initialize**
$f_p \leftarrow f_p / \|f_p\|_2$ for all $p$                                normalize feature
$A_{pq} \leftarrow f_p^T f_q$                                              compute similarity matrix
$B_{pq} \leftarrow \begin{cases} 1 \text{ if } A_{pq} > 0 \\ 0 \text{ otherwise} \end{cases}$    compute adjacency matrix
$\mathcal{P} \leftarrow \{1, \cdots, N\}$                                  set of all positions
$i \leftarrow 0$
**While** $\mathcal{P} \neq \emptyset$
    $i \leftarrow i + 1$
    $d_p \leftarrow \sum_{q \in \mathcal{P}} B_{pq}$ for $p \in \mathcal{P}$              compute degree
    $p^* \leftarrow \arg\min_p d_p$                                        find seed pixel
    $\mathcal{S}_i \leftarrow \{q | A_{p^* q} > \tau$ and $q \in \mathcal{P}\}$   find superpixel according to $p^*$
    $\mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{S}_i$                          exclude currently found pixels
**End While**
$n \leftarrow i$                                                           the number of superpixels
**Output**
$\mathcal{S}_i$ for $i \in \{1, \cdots, n\}$: superpixels
```

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 M of class c as

$$M_c = \begin{cases} \frac{w_c^T x'}{\max w_c^T x'} & \text{if } c \text{ is present class} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where x' is the output of the second last layer and w is the weight of the last layer of the classification network. Using the CAM, we assign the class for confident foreground pixels to initial seed L by taking threshold as

$$L_p = \begin{cases} \arg\max_c M_{p,c} & \text{if } M_{p,c} > \alpha \\ \text{unlabeled} & \text{otherwise.} \end{cases} \tag{3.4}$$

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 region. However, the discovered regions using this approach may contain the foreground regions which are not expressed in the CAM. To identify the background regions better, we find out 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 class-agnostic foreground activation map F by taking the maximum pixels for present foreground classes as $F = \max_{c\in\mathcal{C}} M_c$ 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, $z(\mathcal{S}_i) = \frac{1}{|\mathcal{S}_i|}\sum_{p\in\mathcal{S}_i} F_p$ where $|\mathcal{S}_i|$ is the number of pixels contained in $\mathcal{S}_i$. We select the $\mathcal{S}_i$ with the lowest $z(\mathcal{S}_i)$ as background pixels:

$$L_p = 0 \text{ for } p \in \mathcal{S}_i \text{ s.t. } \mathcal{S}_i = \arg\min_{\mathcal{S}'_i} z(\mathcal{S}'_i) \tag{3.5}$$

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.

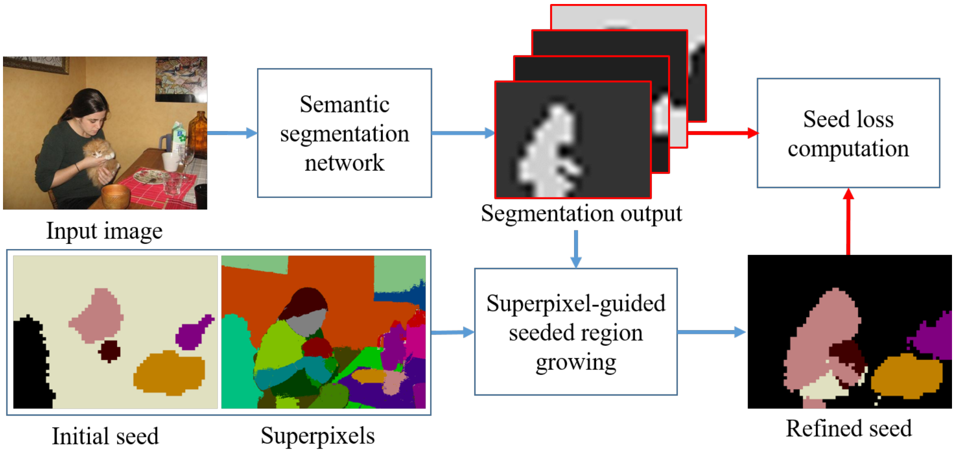


Figure 3.3: The architecture for training of the semantic segmentation network.

3.3.3 Segmentation Network Training

The semantic segmentation network basically 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. 3.3.

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 initial seed, we slightly modify the superpixel by excluding the pixels labeled in the initial seed. That is to say, 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

$$s(\tilde{\mathcal{S}}_i)_c = \frac{1}{|\tilde{\mathcal{S}}_i|} \sum_{p \in \tilde{\mathcal{S}}_i} H_{p,c}. \tag{3.6}$$

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

$$s(\tilde{\mathcal{S}}_i)_c = \max_{c'} s(\tilde{\mathcal{S}}_i)_{c'} \text{ and } s(\tilde{\mathcal{S}}_i)_c > \beta. \tag{3.7}$$

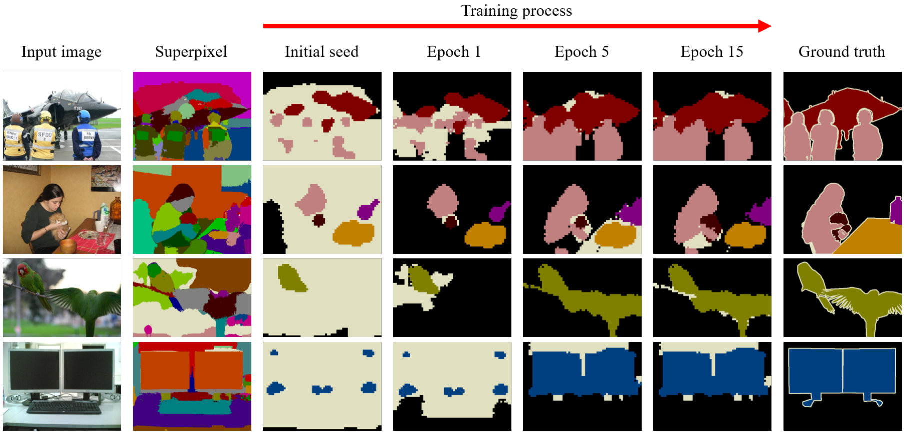


Figure 3.4: Examples for initial seed refined by superpixel-guided seeded region growing during the training process

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. 3.4, 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 [8] that balances the losses between background and foreground classes:

$$\ell_{seed} = -\sum_{p\in\mathcal{L}^b} \frac{1}{|\mathcal{L}^b|} \log H_{p,0} - \sum_{p\in\mathcal{L}^f, c\in\mathcal{C}} \frac{1}{|\mathcal{L}^f|} \log H_{p,c} \tag{3.8}$$

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.

3.4 Experiments

3.4.1 Dataset and Experiment Settings

We evaluate the proposed approach on the PASCAL VOC 2012 segmentation benchmark dataset [18] which has 20 foreground classes and one background class and MS-COCO segmentation dataset [48] which has 80 foreground classes and one background class. PASCAL VOC dataset has 1,464 training images, 1,449 validation images, and 1,456 test images. As in many practices [6, 34], additional dataset is augmented to training dataset, resulting 10,582 training images in total [46]. MS-COCO dataset has 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 21 (for PASCAL VOC) or 81 (for MS-COCO) categories. We obtain the result on the test set by submitting the predicted results to the official PASCAL VOC evaluation server.

For vision transformer, we employ off-the-shelf ViT-Base/8 [72] trained using DINO [22]. Without fine tuning the ViT, we use the key K of the last (12th) transformer block as the features for generating superpixels following [62]. For classification network and segmentation network, we employ ResNet50 and ResNet101 [16] as the backbone network. Both networks are pre-trained on ImageNet classification dataset [17]. For the segmentation network architecture, we use deeplab-ASPP module [6] appended to the ResNet101 backbone network. For the last layer, the parameters are initialized from the normal distribution. In the training of the ResNet101-based DeepLab, 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 [49].

To improve the robustness of the segmentation network, we apply data augmentation techniques. We randomly flip and scale ($\{0.5, 1, 1.5\}$) 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 [50, 6]. We set τ to 0.3 in superpixel discovery method. We set $\alpha = 0.5$ in identifying foreground pixels and $\beta = 0.7$ for the criterion in seeded region growing. We use 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 [51] 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 are multiplied by 10. We train the segmentation network for 15 epochs. Our approach is implemented based on Tensorflow [52]. The classification network and the segmentation network are trained on a single NVIDIA GeForce Titan Xp.

3.4.2 Comparisons with state-of-the-arts

We compare the performance of the proposed method with that of state-of-the-art WSSS methods. In Table 3.2, we summarize the mIOU obtained by WSSS approaches on PASCAL VOC 2012. All method use only image-level labels without additional saliency supervision. 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. In Table 3.3, we summarize the mIOU obtained by WSSS approaches on MS-COCO 2014. From the results, we also observe that our approach outperforms the conventional WSSS approaches. Specifically, our approach achieves mIOU of 44.8% for *val* set.

In particular, we use the same classification network as used in [74], which is also exploited in [30, 66]. In [56], the superpixels are used to recover the object boundaries. In [70], superpixel is exploited in partitioning the input image into complementary patch. Compared to these superpixel-based methods which are benefited from local information about the object boundaries, our approach can take advantage of local and

Table 3.2: Comparison of ResNet-based weakly-supervised semantic segmentation methods' mean IOUs on PASCAL VOC 2012 *val* and *test* set

Method	Publication	Backbone	Val	Test
AffinityNet [53]	CVPR'18	ResNet38	61.7	63.7
IRN [74]	CVPR'19	ResNet50	63.5	64.8
RRM [54]	AAAI'20	ResNet101	66.3	66.5
ICD [56]	CVPR'20	ResNet101	64.1	64.3
SAEM [65]	CVPR'20	ResNet38	64.5	65.7
SC-CAM [75]	CVPR'20	ResNet101	66.1	65.9
BES [76]	ECCV'20	ResNet101	65.7	66.6
CONTA [77]	NeurIPS'20	ResNet38	66.1	66.7
ECSNet [37]	ICCV'21	ResNet38	66.6	67.6
CDA [78]	ICCV'21	ResNet38	66.1	66.8
CPN [70]	ICCV'21	ResNet38	67.8	68.5
CGnet [79]	ICCV'21	ResNet38	68.4	68.2
advCAM [30]	CVPR'21	ResNet101	68.1	68.0
RIB [66]	NeurIPS'21	ResNet101	68.3	68.6
Ours		ResNet50	**67.3**	**66.9**
		ResNet38	**68.3**	**68.4**
		ResNet101	**69.5**	**70.1**

Table 3.3: Comparison of weakly-supervised semantic segmentation methods' mean IOUs on MS-COCO 2014 *val* set

Method	Publication	Backbone	Val
SEC [7]	ECCV'16	VGG16	22.4
DSRG [8]	CVPR'18	VGG16	26.0
ADL [80]	TPAMI'20	VGG16	30.8
GSM [61]	AAAI'21	VGG16	28.4
CONTA [77]	NeurIPS'20	ResNet50	33.4
SGAN [44]	Access'20	VGG16	33.6
IRN [74]	CVPR'19	ResNet101	41.4
RIB [66]	NeurIPS'21	ResNet101	43.8
Ours		ResNet101	**44.8**

global information contained in our semantic-aware superpixels.

We show the semantic segmentation performances in Table 3.4. Our segmentation network architecture is ResNet101-based DeepLab-ASPP [6].

3.4.3 Comparison of Superpixels

Comparison of Superpixels generated using different methods

We compare the qualities of our superpixels with the conventional methods: SLIC [64], SEEDS [81], and LSC [67]. We set the parameters of methods to adjust the number of superpixels similarly. Specifically, for SLIC and LSC, we set the sizes of superpixels to $\{50, 80, 100, 130\}$. For SEEDS, we set the number of superpixels to $\{30, 50, 80, 100\}$. We set the number of iteration to 30 for all methods. For all other parameters, we follow the default settings.

The qualities of superpixels are measured using the undersegmentation error (UE),

Table 3.4: Mean IOUs on PASCAL VOC *val* and *test* set

Class	Val	Test	Class	Val	Test	Class	Val	Test
background	91.2	91.3	car	78.1	81.0	motorbike	76.4	81.8
aeroplane	80.4	82.7	cat	89.3	87.0	person	77.7	80.4
bicycle	40.6	39.2	chair	32.0	32.1	pottedplant	55.3	68.1
bird	78.5	75.6	cow	83.2	80.8	sheep	83.6	86.4
boat	63.4	52.7	diningtable	28.3	33.2	sofa	41.8	46.4
bottle	72.8	70.2	dog	85.2	85.1	train	76.8	73.3
bus	87.5	89.2	horse	82.1	82.9	tvmonitor	54.6	52.0
						mIOU	69.5	70.1

the boundary recall (BR), the boundary precision (BP), and the achievable segmentation accuracy (ASA). Let $\mathcal{S}_k$ be the set of pixels in superpixel k and $\mathcal{G}_i$ be the set of pixels in segmentation ground truth of class i. The UE measures leakages of superpixels across the ground truth:

$$UE(\mathcal{S},\mathcal{G}) = \frac{\sum_i \sum_k \min\{|\mathcal{S}_k \cap \mathcal{G}_i|, |\mathcal{S}_k - \mathcal{G}_i|\}}{\sum_i |\mathcal{G}_i|}. \tag{3.9}$$

The BR measures the percentage of the ground truth boundaries recovered by superpixel boundaries:

$$BR(\mathcal{S},\mathcal{G}) = \frac{\sum_{p\in\delta\mathcal{G}} \mathbf{1}(\min_{q\in\delta\mathcal{S}} \|p-q\| < \epsilon)}{\delta\mathcal{G}} \tag{3.10}$$

where $\delta\mathcal{S}$ and $\delta\mathcal{G}$ are the sets of pixels in all boundaries of $\mathcal{S}$ and $\mathcal{G}$, respectively, and ϵ is the limit distance. We use $\epsilon = 2$. The BP measures the percentage of the superpixel boundaries covering the ground truth boundaries:

$$BP(\mathcal{S},\mathcal{G}) = \frac{\sum_{q\in\delta\mathcal{S}} \mathbf{1}(\min_{p\in\delta\mathcal{G}} \|p-q\| < \epsilon)}{\delta\mathcal{S}}. \tag{3.11}$$

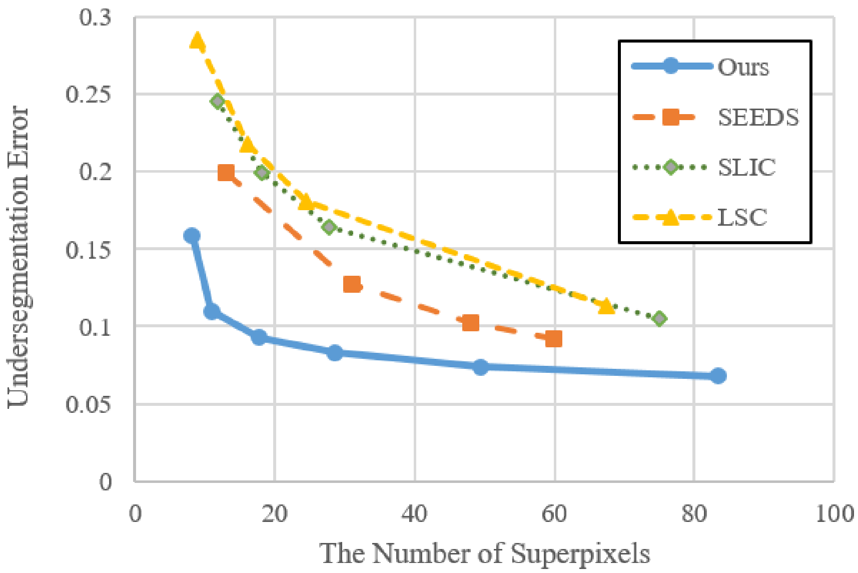


Figure 3.5: Comparison of undersegmentation errors of superpixel methods

The ASA measures the segmentation performance upperbound of the superpixels:

$$ASA(\mathcal{S}, \mathcal{G}) = \frac{\sum_k \max_i |\mathcal{S}_k \cap \mathcal{G}_i|}{\sum_i |\mathcal{G}_i|}. \tag{3.12}$$

We summarize the superpixel measures in Table 3.5 (also see Fig. 3.5 for UE, Fig. 3.6 for PR curve, and Fig. 3.7 for ASA). To compute BP and BR, we use contour finding method provided by *opencv*. We observe that the qualities of our superpixels are better than others when the number of superpixels are small and large. We also show some examples for superpixels when the number of superpixels is small (see Fig. 3.8) and large (see Fig. 3.19).

Comparison of Superpixels generated from different features

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

Table 3.5: Comparison of UE, BP, BR, and ASA on PASCAL VOC *train* set for conventional superpixel methods and ours

Method	# of superpixels	UE ↓	BP↑	BR↑	ASA ↑
SLIC [64]	11.72	0.246	0.057	0.635	0.879
	18.22	0.199	0.057	0.636	0.902
	27.67	0.164	0.055	0.644	0.92
	75.04	0.105	0.051	0.675	0.949
SEEDS [81]	13.10	0.199	0.074	0.366	0.902
	31.03	0.127	0.062	0.463	0.938
	47.98	0.102	0.06	0.511	0.951
	59.90	0.092	0.058	0.532	0.956
LSC [67]	9.00	0.285	0.052	0.689	0.859
	16.06	0.218	0.050	0.718	0.893
	24.47	0.181	0.049	0.733	0.911
	67.42	0.113	0.046	0.763	0.945
Ours, $\tau = 0$	8.14	0.159	0.104	0.515	0.922
Ours, $\tau = 0.1$	10.89	0.110	0.103	0.564	0.947
Ours, $\tau = 0.2$	17.74	0.093	0.099	0.591	0.955
Ours, $\tau = 0.3$	28.57	0.083	0.094	0.615	0.960
Ours, $\tau = 0.4$	49.35	0.074	0.087	0.652	0.965
Ours, $\tau = 0.5$	83.39	0.068	0.079	0.678	0.968

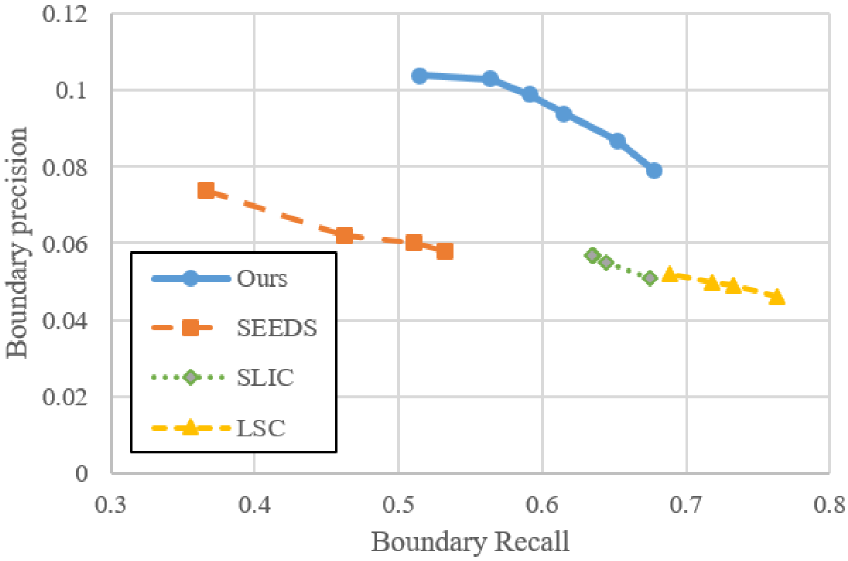


Figure 3.6: Comparison of boundary precision-recall curves of superpixel methods

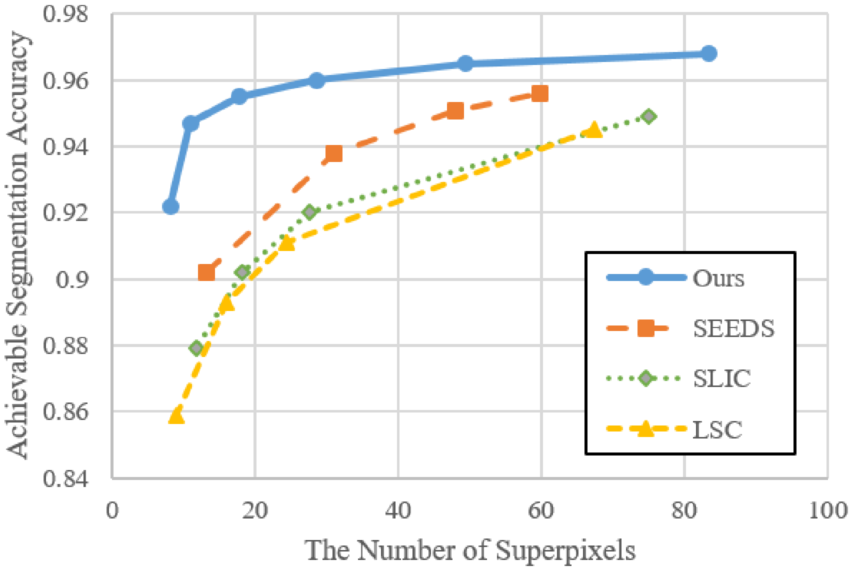


Figure 3.7: Comparison of achievable segmentation accuracies of superpixel methods

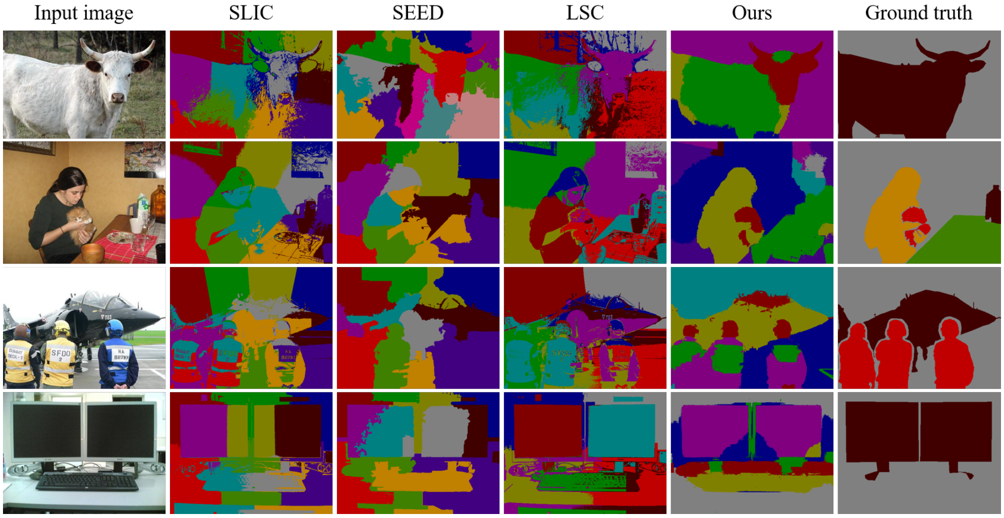


Figure 3.8: Examples of superpixels with the number of superpixels ranging from 10 to 15.

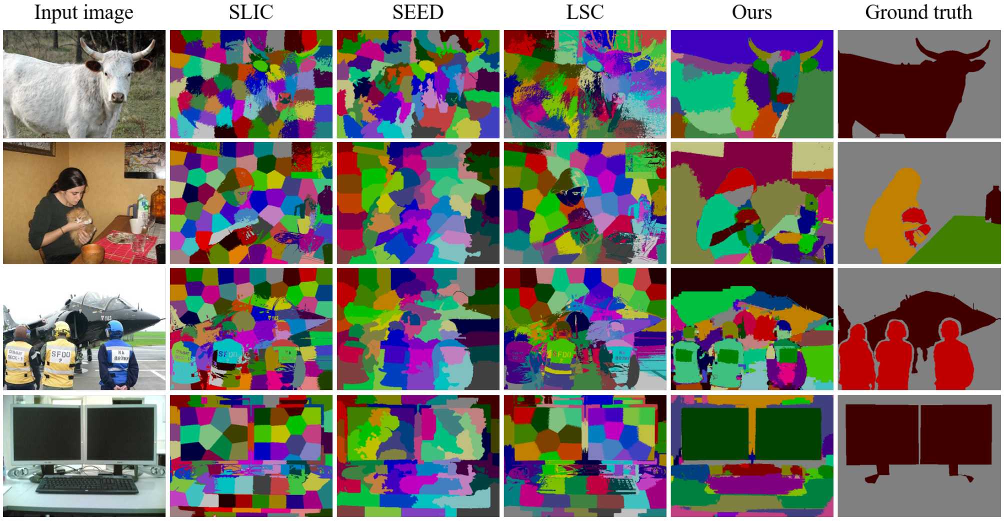


Figure 3.9: Examples of superpixels with the number of superpixels ranging from 50 to 80.

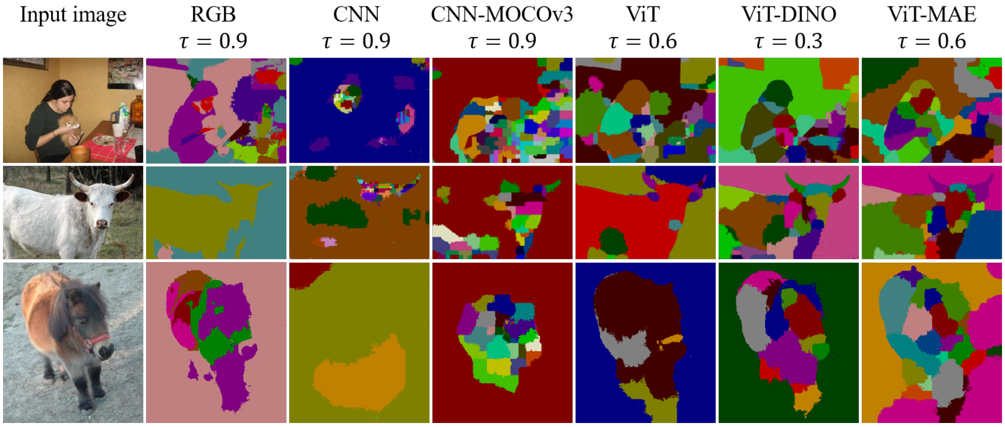


Figure 3.10: Superpixels generated using different features

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 [16]), and self-supervised CNN (MoCov3 [82]), 3) transformer features obtained from supervised transformer (ViT [72]), and self-supervised transformers (DINO [22] and MAE [83] which is known to outperform DINO in down-stream tasks). The backbones of CNNs and transformers are ResNet50 and ViT-Base/16, respectively. The 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 using different features in Fig. 3.10. In this experiment, we first generate the superpixels using the DINO feature with setting $\tau = 0.3$. The τ for other features are adjusted so that the number 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 could not 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 ViTs. 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, 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.

Comparison of Superpixels generated using different bipartition rules

In the proposed method, we select the seed pixel by finding a pixel having the lowest degree in each iteration step. Hence, we can identify superpixels representing the smallest object or its part among the remaining pixels. We compare the superpixels obtained using the proposed methods which identify the pixel with lowest or highest degree in each iteration, which are denoted as low first (LF) or high first (HF), respectively. We show some examples in Fig. 3.11. The brightness of superpixel indicates the order the superpixel is discovered (bright first, dark last). As we can observe, we fail to partition the image using HF with low τ. If we set τ to high value, we can obtain good partitions of images using HF but there are still more undersegmented regions than LF.

To compare the superpixels generated using different bipartitioning rule, we apply normalized cut (Ncut [84]) to obtain superpixels and compare the proposed superpixel

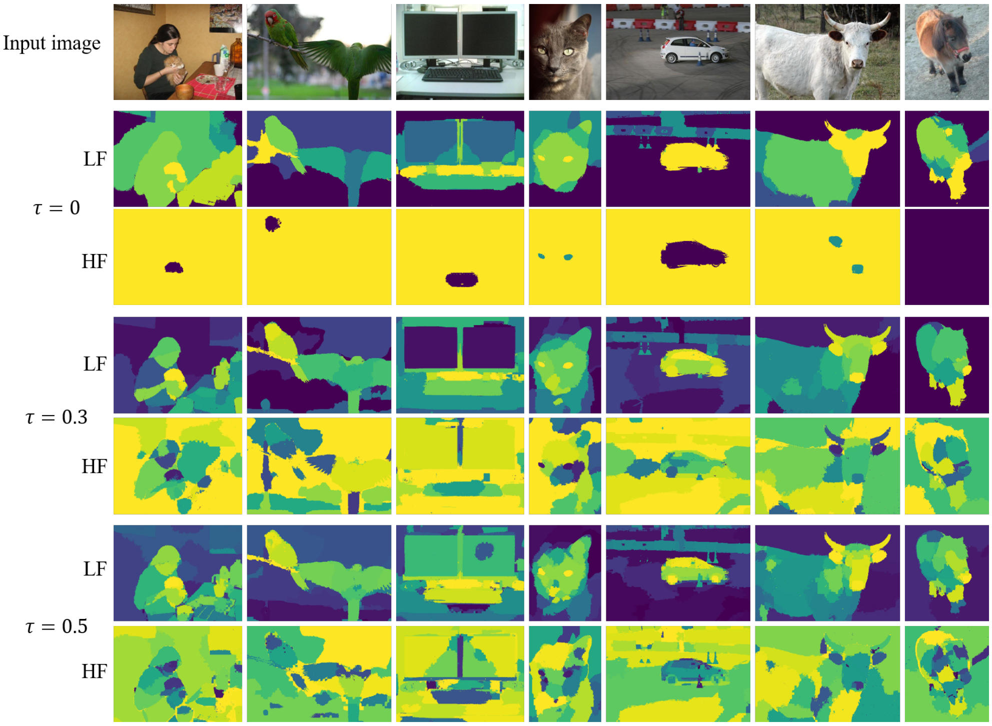


Figure 3.11: Examples of superpixels generated using different rule for seed pixel

with Ncut-based superpixel. We first compute the similarity matrix A as (3.1). Then, the adjacency matrix W is obtained as

$$W_{pq} = \begin{cases} 1 & \text{if } A_{pq} > \tau \\ \epsilon & \text{otherwise} \end{cases} \tag{3.13}$$

where ϵ is the small constant. Also, the diagonal matrix D is obtained as

$$D_{pp} = \sum_{q} W_{pq}. \tag{3.14}$$

By solving the eigenvalue system

$$(D - W)y = \lambda Dy \tag{3.15}$$

and finding the second smallest eigenvector, we can obtain the vector representing bipartition of image. Since the eigenvector is real valued, we set the indices greater than the average of elements of eigenvector to partition A and the rest indices to partition B. As in the proposed approach, we choose the small partition as the currently discovered superpixel and repeat the procedure.

We compare the superpixel measures of the proposed superpixels and Ncut-based superpixel. As illustrated in Fig. 3.12, we observe that the qualities of the proposed superpixel are better than the qualities of the Ncut-based superpixel. We show some qualitative results for both superpixels. As illustrated in Fig. 3.13, Ncut-based approach can group the all pixels of an object (see the first four columns in 3.13) whereas may undersegment images(see the last two columns in 3.13). On the other hand, the proposed approach may oversegment the objects but can segment the image properly by increasing the threshold τ. One notable point is the inference time of superpixel algorithms. In the proposed approach, we need to simply take threshold to similarity matrix to discover superpixel in each iteration. On the other hand, in Ncut-based approach, we need to solve large eigenvalue system of size $> 2,000$ in each iteration. As a result, the proposed approach processes 1,000 images in $8 \sim 100$ minutes whereas the Ncut-based approach requires $24 \sim 60$ hours to process the same amount of images. Note that the processing time depends on τ.

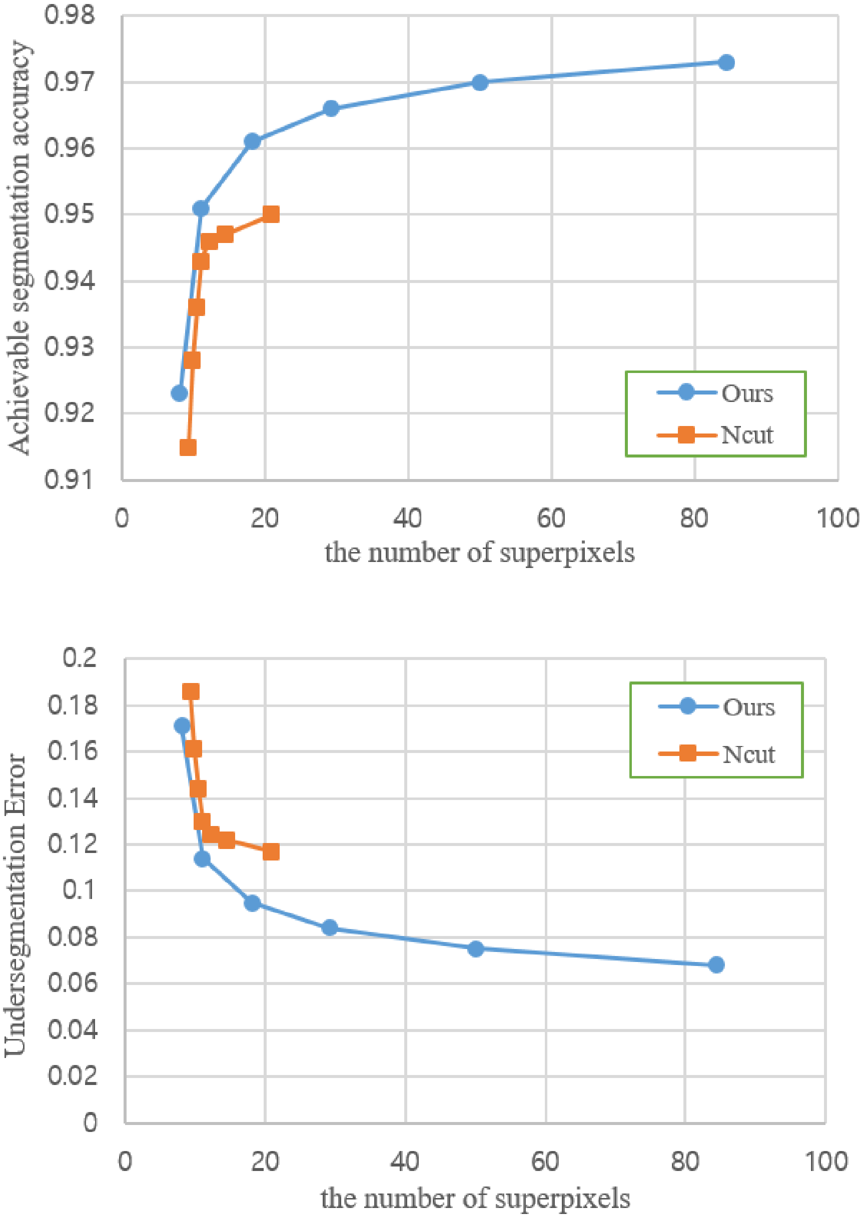


Figure 3.12: Comparison of ASA and UE of superpixels generated from various bipartition methods

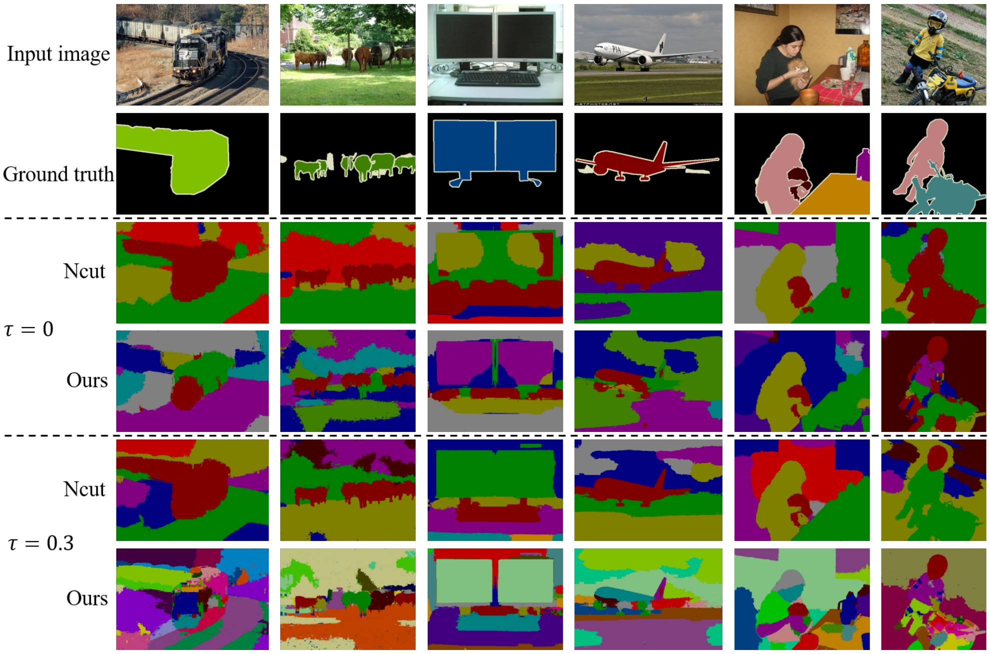


Figure 3.13: Comparison of superpixels generated from various bipartition methods

3.4.4 Effects of Hyperparameters

The Effect of τ

In the proposed superpixel method, we use threshold τ in grouping the similar pixels. Before we examine the effect of different τ, we investigate the characteristics of the adjacency matrix indicating the positiveness of pixel pair. We may wonder if the pixels belonging to the objects of our interest have positive similarities to each other. Also, we may wonder if there is any pair of pixels in the object having negative similarity. To observe this, we compute the number of pixels in objects having positive or negative similarity and visualize in Fig. 3.14. The brightness of 'positiveness' and 'negativeness' indicate the number of pixels in object having positive and negative similarities, respectively. From the results in the upper row of Fig. 3.14, we observe that the most of pixels belonging to the object are positively similar to each other. On

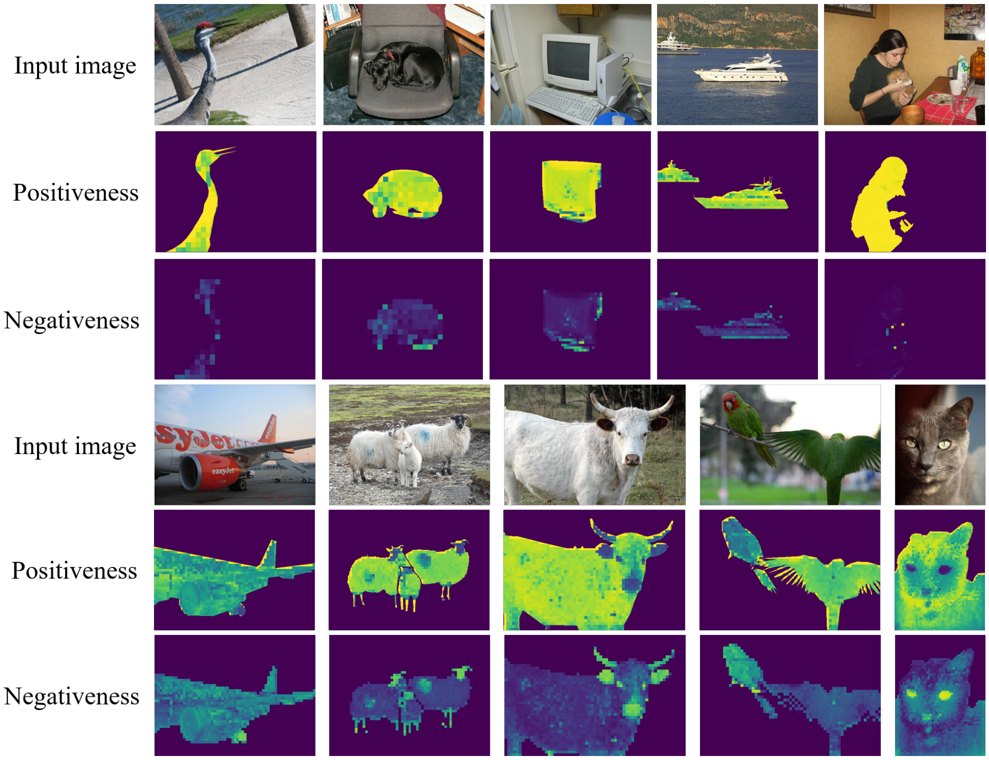


Figure 3.14: Examples of positiveness and negativeness maps indicating the number of positive and negative pixels in an object.

the other hand, from the results in the lower row of Fig. 3.14, we observe that the pixels belonging to some parts of objects or different object of same class may have negative similarities to others. Based on this observation, we notice that we may not find a group of pixels only using the adjacency matrix B.

In order to find out good partition of image, we can use threshold τ. For seed pixel p^*, we find the pixels satisfying $A_{pp^*} > \tau$.

lemma 1 For normalized vectors $a \in \mathbb{R}^{D\times 1}$, $b \in \mathbb{R}^{D\times 1}$, and $c \in \mathbb{R}^{D\times 1}$, if $a^T b > \tau$ and $a^T c > \tau$, then $b^T c > 2\tau^2 - 1$.

Proof: From the assumption that $a^T b > \tau$ and $a^T c > \tau$, there exist α, β satisfying $\cos\alpha > \tau$ and $\cos\beta > \tau$. Then, $b^T c$ has the relation with α and β as $b^T c > \cos(\alpha+\beta)$ or $b^T c > \cos(\alpha - \beta)$.

case i) When $b^T c = \cos(\alpha + \beta)$, this is minimized when $\sin\alpha \sin\beta > 0$.

$$\begin{aligned}
b^T c &\geq \cos(\alpha+\beta) && (3.16)\\
&= \cos\alpha\cos\beta - \sin\alpha\sin\beta && (3.17)\\
&> \cos\alpha\cos\beta - \frac{1}{2}(\sin^2\alpha + \sin^2\beta) && (3.18)\\
&= \cos\alpha\cos\beta - \frac{1}{2}(1 - \cos^2\alpha + 1 - \cos^2\beta) && (3.19)\\
&= \frac{1}{2}(2\cos\alpha\cos\beta + \cos^2\alpha + \cos^2\beta - 2) && (3.20)\\
&= \frac{1}{2}((\cos\alpha + \cos\beta)^2 - 2) && (3.21)\\
&> \frac{1}{2}((2\tau)^2 - 2) && (3.22)\\
&= 2\tau^2 - 1 && (3.23)
\end{aligned}$$

(11) is because $x^2 + y^2 > 2xy$ when $xy > 0$.

case ii) When $b^T c = \cos(\alpha - \beta)$, this is minimized when $\sin\alpha \sin\beta < 0$.

$$\begin{aligned} b^T c &\geq \cos(\alpha - \beta) && (3.24) \\ &= \cos\alpha\cos\beta + \sin\alpha\sin\beta && (3.25) \\ &= \cos\alpha\cos\beta - |\sin\alpha\sin\beta| && (3.26) \\ &> \cos\alpha\cos\beta - \frac{1}{2}(\sin^2\alpha + \sin^2\beta) && (3.27) \\ &> 2\tau^2 - 1. && (3.28) \end{aligned}$$

We have (21) as in (11)-(16).

Based on lemma 1, if we set τ to be $2\tau^2 - 1 = 0$ or $\tau = 1/\sqrt{2} \approx 0.71$, we can guarantee that all pixels in a superpixel have positive similarity to others. However, this setting results in oversegmentation of image (see Fig. 3.15).

In our superpixel discovery method, the seed pixel is the pixel with 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. 3.16, 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 3.6. We can observe that the segmentation performance degrades when we use oversegmented superpixel.

The Effect of α

To examine the effect of α in generating the initial seed, we conduct some experiments using different initial seeds. We summarize the results in Table 3.7. From the results,

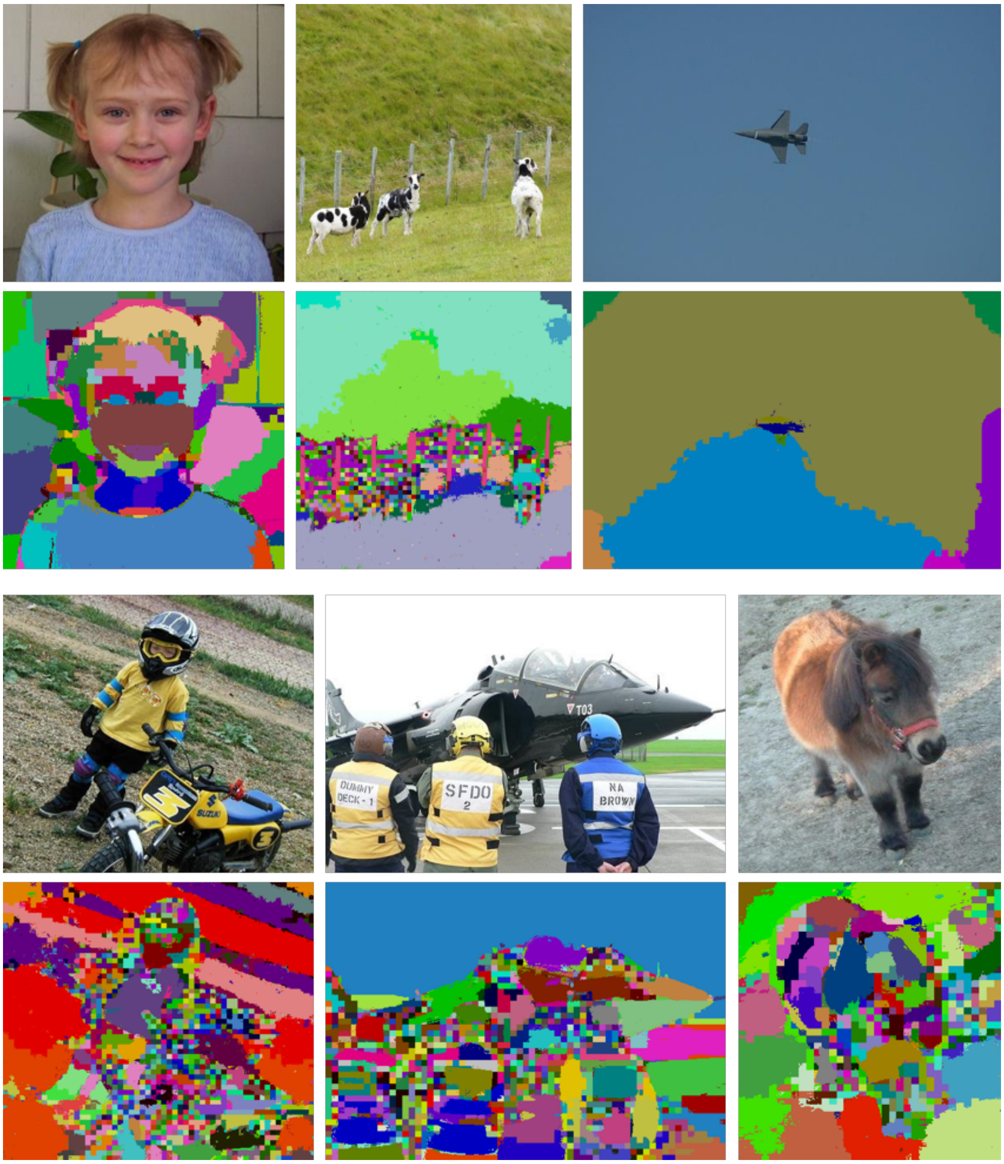


Figure 3.15: Examples of superpixels generated when $\tau = 0.7$

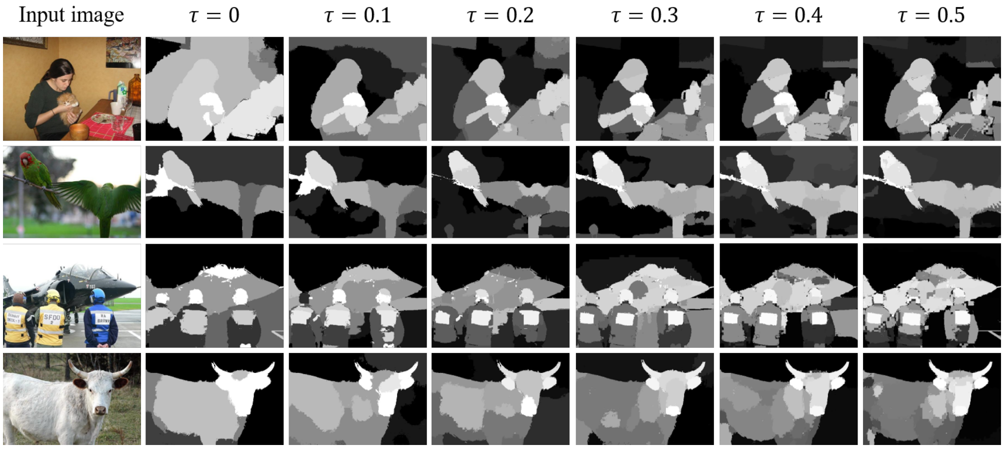


Figure 3.16: Superpixels according to different thresholds τ

Table 3.6: Comparison of mean IOUs on PASCAL VOC *val* and *test* set using different superpixels

τ	Val w/o crf	Val with crf	Test
0	64.8	69.3	-
0.1	65.0	**69.5**	69.3
0.2	64.6	69.1	-
0.3	65.1	**69.5**	70.1
0.4	63.8	68.4	-
0.5	62.9	67.5	-

Table 3.7: Comparison of mean IOUs on PASCAL VOC *val* set using different α for generating initial seed

α	Val	Val + crf
0.3	59.7	66.0
0.4	62.8	68.6
0.5	65.0	69.2
0.6	66.3	69.4
0.7	66.7	69.0
0.8	65.5	67.6

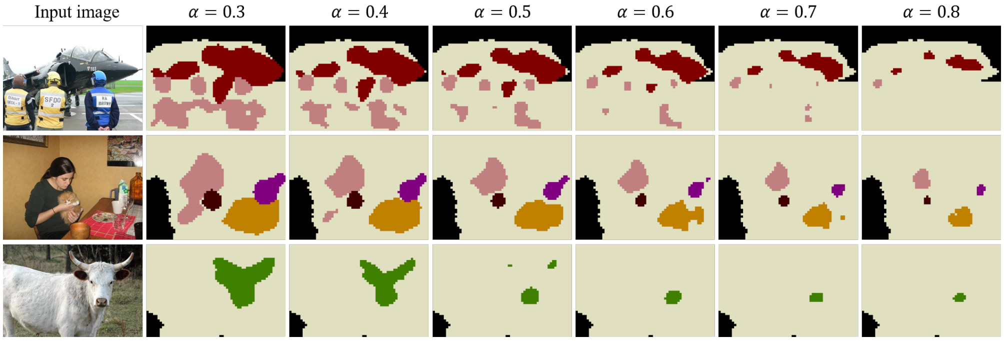


Figure 3.17: Examples for initial seeds generated by varying α

Table 3.8: Comparison of mean IOUs for different β_{bg} and β_{fg} on PASCAL VOC 2012 *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

we see that the best performance is obtained when $\alpha = 0.6$ is used. 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 good segmentation performance when the initial seed is very sparse (i.e., α is high). We illustrate the initial seeds in Fig. 3.17 to compare how the initial seeds are sparse.

The Effect of β

We study the effects of β in superpixel-guided seeded regions growing. As done in [8], 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 3.8. From the results, we see that we can achieve good segmentation performance when we choose the two parameters 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, leading 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.

3.4.5 Qualitative Results

In Fig. 3.18, we provide qualitative results obtained from our segmentation network on PASCAL VOC 2012. 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. 3.19, we also provide qualitative results obtained from our segmentation network on MS-COCO 2014.

In Fig. 3.20, we provide some failure cases for refined seed in the training process (Fig. 3.20 (a)) and wrong prediction for similar images in *val* set (Fig. 3.20 (b)). 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.

3.5 Summary of Chapter 3

In this chapter, we have proposed a simple superpixel discovery method that finds out the semantic-aware superpixels in a 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 superpixel-guided seeded region growing method outperforms the conventional WSSS approaches. Extensive experiments demonstrates that our approach is effective in solving WSSS problem.

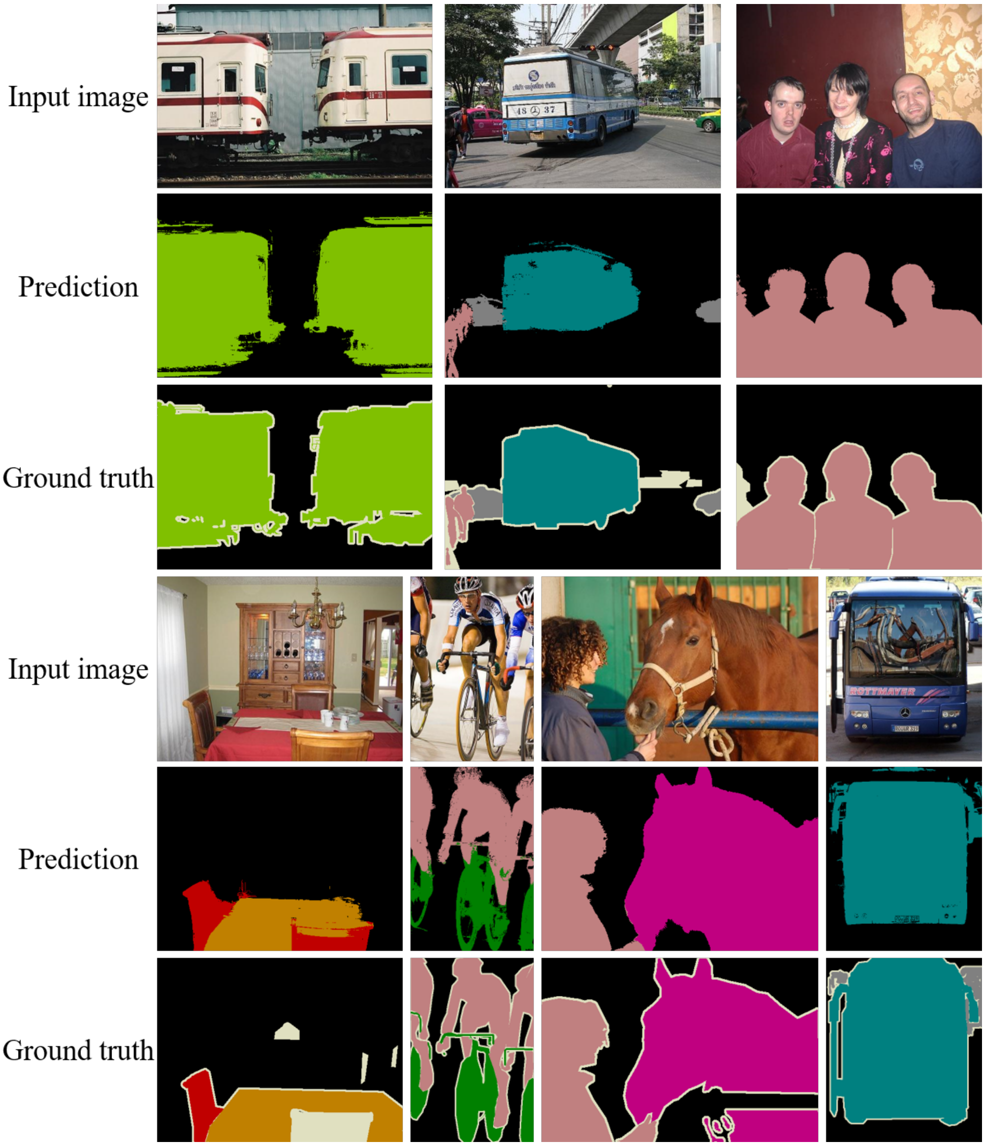


Figure 3.18: Examples of our segmentation outputs for PASCAL VOC 2012 *val* set.

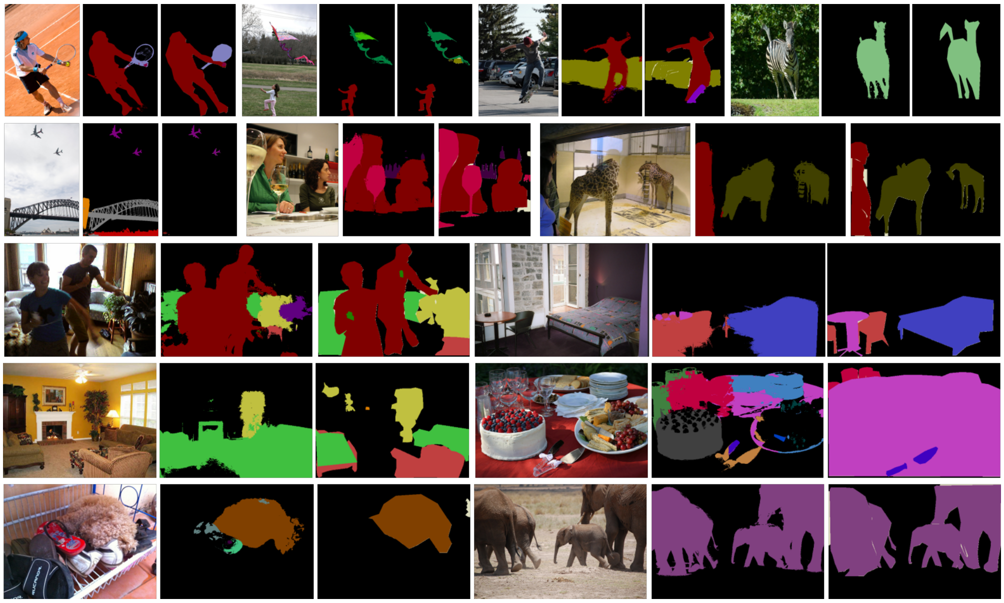
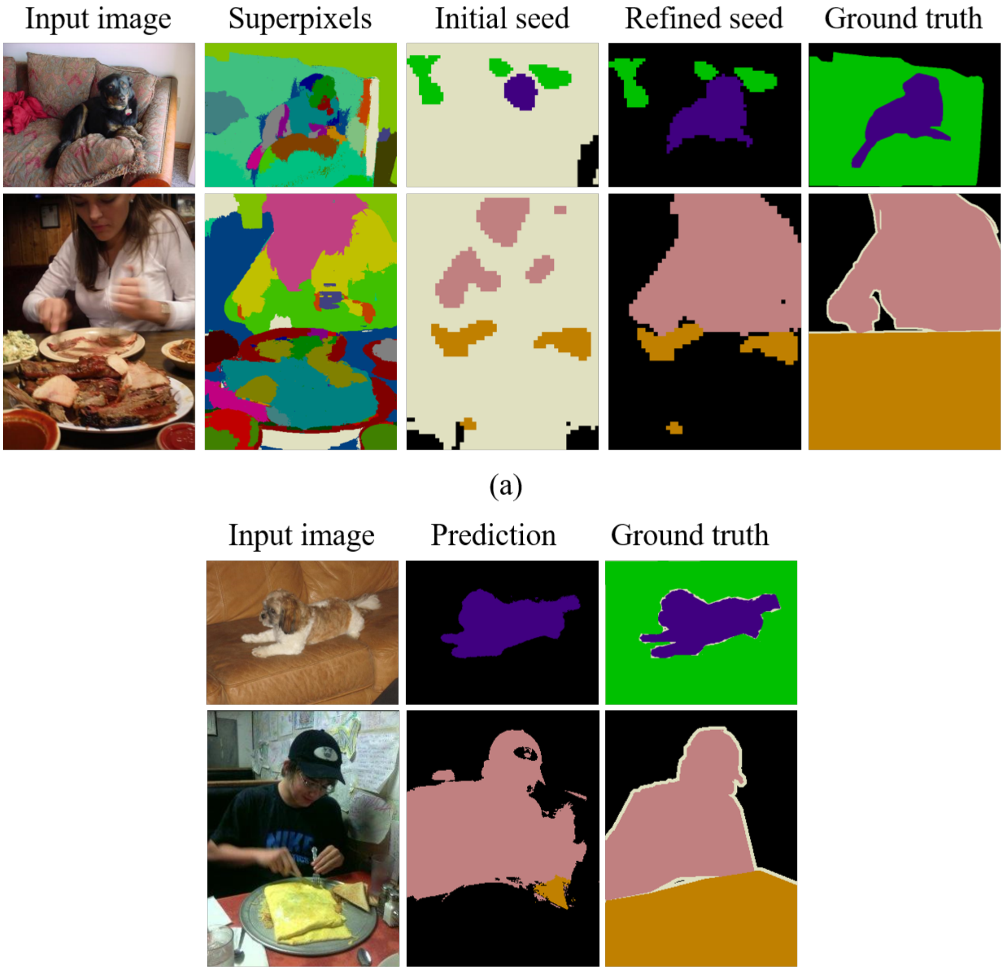


Figure 3.19: Examples of our segmentation outputs for PASCAL VOC 2012 *val* set.



(a)

(b)

Figure 3.20: Examples for failure cases of the refined seeds in training process and wrong predictions for similar images in *val* images.

Chapter 4

Conclusion and Future Research

In this dissertation, we studied the problem of weakly supervised semantic segmentation when the image-level label is given. Although the recently developed deep neural network outperforms the conventional network, I focused on the fundamental techniques which can improve the segmentation performances of any semantic segmentation networks. In specific, I made the following contributions:

In Chapter 2, we proposed a new WSSS technique that can train the segmentation network without pixel-level pseudo-labels. To prevent the performance degradation caused by inaccurate pseudo-label in conventional WSSS approaches, we have exploited the image masking technique in the training of the segmentation network. We also introduced an approach to refine the saliency map, which significantly improves the segmentation performance. Extensive experiments demonstrate that our approach is effective in solving the problem of WSSS. As an extension of this work, a new training strategy for segmentation network aided by more powerful classification network having different recognition mechanism from CNN could be a desirable direction for the future work.

In Chapter 3, we proposed a simple superpixel discovery method that finds out the semantic-aware superpixels in a 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 showed that our semantic segmentation network training strategy using superpixel-guided seeded region growing method outperforms the conventional WSSS approaches. Extensive experiments demonstrates that our approach is effective in solving WSSS problem. Using the superpixels obtained from the self-supervised vision transformers, to perform the unsupervised semantic segmentation could be a promising future direction of research.

Bibliography

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

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

[3] R. Ren, T. Hung, and K. C. Tan, "A generic deep-learning-based approach for automated surface inspection," *IEEE transactions on cybernetics*, vol. 48, no. 3, pp. 929–940, 2017.

[4] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler, "Learning aerial image segmentation from online maps," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 11, pp. 6054–6068, 2017.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[6] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution,

and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

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

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

[9] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1742–1750, 2015.

[10] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, "What's the point: Semantic segmentation with point supervision," in *European conference on computer vision*. Springer, Cham, 2016, pp. 549–565.

[11] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3159–3167.

[12] J. Dai, K. He, and J. Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1635–1643.

[13] P. O. Pinheiro and R. Collobert, "From image-level to pixel-level labeling with convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1713–1721.

[14] S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele, "Exploiting saliency for object segmentation from image level labels," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 5038–5047.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[18] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[20] D. Pathak, P. Krahenbuhl, and T. Darrell, "Constrained convolutional neural networks for weakly supervised segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1796–1804.

[21] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

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

[23] P. Tang, X. Wang, X. Bai, and W. Liu, "Multiple instance detection network with online instance classifier refinement," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2843–2851.

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

[25] M. M. Chun and J. M. Wolfe, "Visual attention," *Blackwell handbook of sensation and perception*, pp. 272–310, 2005.

[26] G. Papandreou, L.-C. Chen, K. Murphy, and A. Yuille, "Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. arxiv, 2015," *arXiv preprint arXiv:1502.02734*.

[27] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon, "Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5267–5276.

[28] P.-T. Jiang, Q. Hou, Y. Cao, M.-M. Cheng, Y. Wei, and H.-K. Xiong, "Integral object mining via online attention accumulation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2070–2079.

[29] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, "Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7268–7277.

[30] J. Lee, E. Kim, and S. Yoon, “Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4071–4080.

[31] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, “Residual attention network for image classification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3156–3164.

[32] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[33] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *International conference on machine learning*. PMLR, 2015, pp. 2048–2057.

[34] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, “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*, 2017, pp. 1568–1576.

[35] A. Chaudhry, P. K. Dokania, and P. H. Torr, “Discovering class-specific pixels for weakly-supervised semantic segmentation,” *arXiv preprint arXiv:1707.05821*, 2017.

[36] D. Kim, D. Cho, D. Yoo, and I. Kweon, “Two-phase learning for weakly supervised object localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3534–3543.

[37] K. Sun, H. Shi, Z. Zhang, and Y. Huang, “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, 2021, pp. 7283–7292.

[38] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T. S. Huang, “Adversarial complementary learning for weakly supervised object localization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1325–1334.

[39] B. Kim, S. Han, and J. Kim, “Discriminative region suppression for weakly-supervised semantic segmentation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1754–1761.

[40] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, “Semantic segmentation using adversarial networks,” *arXiv preprint arXiv:1611.08408*, 2016.

[41] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao, and S. Yan, “Stc: A simple to complex framework for weakly-supervised semantic segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2314–2320, 2016.

[42] X. Wang, S. You, X. Li, and H. Ma, “Weakly-supervised semantic segmentation by iteratively mining common object features,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1354–1362.

[43] F. Sun and W. Li, “Saliency guided deep network for weakly-supervised image segmentation,” *Pattern Recognition Letters*, vol. 120, pp. 62–68, 2019.

[44] Q. Yao and X. Gong, “Saliency guided self-attention network for weakly and semi-supervised semantic segmentation,” *IEEE Access*, vol. 8, pp. 14 413–14 423, 2020.

[45] S. Lee, M. Lee, J. Lee, and H. Shim, “Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation,” in *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 5495–5505.

[46] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 991–998.

[47] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3917–3926.

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

[49] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Advances in neural information processing systems*, 2011, pp. 109–117.

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

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

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

[53] J. Ahn and S. Kwak, "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation," in *Proceedings of*

the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4981–4990.

[54] B. Zhang, J. Xiao, Y. Wei, M. Sun, and K. Huang, “Reliability does matter: An end-to-end weakly supervised semantic segmentation approach,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 765–12 772.

[55] L. Xu, M. Bennamoun, F. Boussaid, and F. Sohel, “Scale-aware feature network for weakly supervised semantic segmentation,” *IEEE Access*, vol. 8, pp. 75 957–75 967, 2020.

[56] J. Fan, Z. Zhang, C. Song, and T. Tan, “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*, 2020, pp. 4283–4292.

[57] Y. Yao, T. Chen, G.-S. Xie, C. Zhang, F. Shen, Q. Wu, Z. Tang, and J. Zhang, “Non-salient region object mining for weakly supervised semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2623–2632.

[58] Y. Liu, Y.-H. Wu, P.-S. Wen, Y.-J. Shi, Y. Qiu, and M.-M. Cheng, “Leveraging instance-, image-and dataset-level information for weakly supervised instance segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[59] S.-Y. Pan, C.-Y. Lu, S.-P. Lee, and W.-H. Peng, “Weakly-supervised image semantic segmentation using graph convolutional networks,” in *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.

[60] L. Xu, W. Ouyang, M. Bennamoun, F. Boussaid, F. Sohel, and D. Xu, “Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation,”

in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6984–6993.

[61] X. Li, T. Zhou, J. Li, Y. Zhou, and Z. Zhang, "Group-wise semantic mining for weakly supervised semantic segmentation," *arXiv preprint arXiv:2012.05007*, 2020.

[62] O. Siméoni, G. Puy, H. V. Vo, S. Roburin, S. Gidaris, A. Bursuc, P. Pérez, R. Marlet, and J. Ponce, "Localizing objects with self-supervised transformers and no labels," *arXiv preprint arXiv:2109.14279*, 2021.

[63] Y. Wang, X. Shen, S. Hu, Y. Yuan, J. Crowley, and D. Vaufreydaz, "Self-supervised transformers for unsupervised object discovery using normalized cut," *arXiv preprint arXiv:2202.11539*, 2022.

[64] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International journal of computer vision*, vol. 59, no. 2, pp. 167–181, 2004.

[65] Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen, "Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 275–12 284.

[66] J. Lee, J. Choi, J. Mok, and S. Yoon, "Reducing information bottleneck for weakly supervised semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[67] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1356–1363.

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

[69] S. Kwak, S. Hong, and B. Han, “Weakly supervised semantic segmentation using superpixel pooling network,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[70] F. Zhang, C. Gu, C. Zhang, and Y. Dai, “Complementary patch for weakly supervised semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7242–7251.

[71] R. Adams and L. Bischof, “Seeded region growing,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 16, no. 6, pp. 641–647, 1994.

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

[73] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.

[74] J. Ahn, S. Cho, and S. Kwak, “Weakly supervised learning of instance segmentation with inter-pixel relations,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2209–2218.

[75] Y.-T. Chang, Q. Wang, W.-C. Hung, R. Piramuthu, Y.-H. Tsai, and M.-H. Yang, “Weakly-supervised semantic segmentation via sub-category exploration,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8991–9000.

[76] L. Chen, W. Wu, C. Fu, X. Han, and Y. Zhang, “Weakly supervised semantic segmentation with boundary exploration,” in *European Conference on Computer Vision*. Springer, 2020, pp. 347–362.

[77] D. Zhang, H. Zhang, J. Tang, X.-S. Hua, and Q. Sun, “Causal intervention for weakly-supervised semantic segmentation,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 655–666, 2020.

[78] Y. Su, R. Sun, G. Lin, and Q. Wu, “Context decoupling augmentation for weakly supervised semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7004–7014.

[79] H. Kweon, S.-H. Yoon, H. Kim, D. Park, and K.-J. Yoon, “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*, 2021, pp. 6994–7003.

[80] J. Choe, S. Lee, and H. Shim, “Attention-based dropout layer for weakly supervised single object localization and semantic segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 12, pp. 4256–4271, 2020.

[81] M. V. d. Bergh, X. Boix, G. Roig, B. d. Capitani, and L. V. Gool, “Seeds: Superpixels extracted via energy-driven sampling,” in *European conference on computer vision*. Springer, 2012, pp. 13–26.

[82] X. Chen, S. Xie, and K. He, “An empirical study of training self-supervised vision transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9640–9649.

[83] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

[84] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

초 록

영상 분할은 영상 속 모든 픽셀을 관심있는 클래스로 분류하는 작업으로, 자율 주행, 의료 진단, 산업 자동화, 위성 영상 등에 널리 활용될 수 있는 중요한 문제이다. 최근에는 딥 컨볼루셔널 뉴럴 네트워크를 사용하여 영상 분할을 해결하는 방법이 그 우수한 성능으로 주목 받고있다. 이 접근 방법의 어려운 점은 네트워크를 학습시키기 위해서 대량의 정교하게 제작된 레이블이 필요하다는 점이다. 이러한 데이터로 구성된 데이터셋을 얻는것기에는 시간과 비용이 많이 소모되기 때문에 미래의 연구 방향으로 약지도 상황에서 영상 분할을 수행하는 것이 유망한 접근 방법으로써 다루어지고 있다. 영상 분할에 사용 할만한 약지도를 위한 레이블의 종류에는 영상 단위의 레이블 또는 점, 낙서, 경계 사각형 등이 있다. 이 중 영상 속에 존재하는 물체의 종류를 나타내는 영상 단위의 레이블이 가장 단순하고 제작이 쉽기 때문에 대부분의 연구에서 이 레이블이 활용되고 있다. 이 논문에서는 영상 단위의 레이블을 사용한 약지도 영상 분할 문제를 다룬다.

논문의 첫번째 부분에서는 양지도 영상분할을 위한 새로운 학습 기법을 소개한다. 제안하는 방법에서는 관심있는 시각 영역에 집중하고 관련 없는 부분을 무시하는 인간의 시각계로부터 영감을 얻은 이미지 마스킹 기법을 활용한다. 분할 네트워크로부터 얻은 출력으로 분류 네트워크가 집중 할 영역을 제한하여 분류 네트워크가 분할 네트워크의 출력의 질을 평가하도록 하며, 분할 네트워크가 더욱 정확하게 출력할 수 있도록 한다. 분할 성능을 향상시키기 위하여 간단하지만 효과적인 분류 네트워크 학습 방법과 특징 지도 개선 방법을 제안한다. 다양한 실험을 통하여 제안하는 방법으로 약지도 영상 분할을 효과적으로 해결할 수 있음을 보인다.

논문의 두번째 부분에서는 의미 인지 슈퍼픽셀을 생성하는 알고리즘을 제안한다. 제안하는 알고리즘으로 얻은 슈퍼픽셀은 멀리 떨어져 있더라도 비슷한 성징을 가질 경우에 하나의 묶음으로 합쳐질 수 있다는 새로운 특징이 있다. 또한, 슈퍼픽셀의 수는 미리 정해놓은 개수로 정해지는 것이 아닌 영상의 복잡도에 의해 정해진다는 특징이 있다. 제안하는 방법으로 얻은 슈퍼픽셀은 의미가 비슷한 픽셀들을 아주 적은 수의 슈퍼픽셀들로 표현해 낼 수 있으며 제안하는 슈퍼픽셀을 사용하여 기존의 슈퍼픽셀로는 달성하기 어려운 높은 정확도의 약지도 영상 분할 성능을 얻을 수 있다. 제안하는 분할 네트워크를 학습시키기 위하여 슈퍼픽셀에 의해 제한되는 시드 영역 확장 방법을 통해 밀도가 낮은 레이블의 질을 향상시키고 이것을 새로운 레이블로 사용한다. 다양한 실험을 통해 제안하는 방법이 약지도 영상 분할에 효과적임을 보인다.

주요어: 영상 분류, 영상 분할, 약지도학습, 딥러닝, 시각적 어텐션, 슈퍼픽셀

학번: 2015-20897

감사의 글

고등학교 시절 새로운 기술이 개발되었다는 신문 기사를 보고 멋있고 재미있겠다는 생각을 했고, 대학교에 입학한 후 그러한 개발을 하기위해서는 박사과정을 마쳐야 한다는 것을 알게 되어 대학원에 진학하여 어느덧 박사과정 졸업을 앞두게 되었습니다. 길었던 이 과정동안 도움을 주신 많은 분들께 짧은 글로나마 감사의 말씀을 전하고자 합니다.

우선 박사 과정동안 열성적으로 지도해주신 심병효 교수님께 감사의 말씀을 전합니다. 2010년 고려대학교에서의 강의로 처음 뵙게 된 때가 엊그제같은데 교수님과 동고동락한 과거의 시간들을 되돌아 생각하면 감회가 새롭습니다. 부족한 저를 위해 끊임없이 지도해주시고 연구에만 전념할 수 있도록 해주신 덕에 새로운 분야의 연구도 진행 해 볼 수 있었습니다. 부족한 결과물을 갈고 닦는데 쓰신 그 노고에 진심으로 감사드리며 평생 잊지 않겠습니다. 학위 논문 심사과정에서 위원장을 맡아주시고 심사해주신 김성철 교수님께 깊이 감사드립니다. 아울러 바쁘신 와중에도 심사위원을 맡아주시고 소중한 조언을 해주신 이경한 교수님, 문태섭 교수님, 인하대학교 박대영 교수님께도 감사드립니다.

대학원 생활 동안 함께한 소중한 연구실 친구들에게도 감사드립니다. 대학원 기간 전반에 걸쳐 가장 오랜 기간 함께 많은 일을 겪으며 친구처럼 친하게 지낸 Nguyen Trung Luong에게 감사드립니다. 먼저 졸업하고 조언을 해준 원준, 준한, 선도에게 감사드립니다. 아울러 301동에서 연구실 생활을 함께 했던 진홍, 용준, 규홍, 선우, 지섭, Khoa, 동훈, 성욱, 지영, 진우, 인국, 구상, 인수에게 감사드립니다. 짧은 시간이나마 뉴미에서 함께 했던 승년, 현규, Jiao, 지훈, 현수, 윤성, 용석, 윤서, 정재, 안호, Yiying, 석현에게도 감사드립니다.

사랑하는 가족들에게도 감사의 말씀을 전합니다. 항상 보살펴주시고 믿어주시

고 지원해주신 조부 고 김윤호, 조모 이수자, 아버지 김창원과 어머니 백경화 감사드리고 사랑합니다. 그리고 같이 자라오며 서로에게 의지가 되어 준 누나 김지연, 동생 김소연, 김소희 감사드립니다. 저를 아들처럼 생각하시고 항상 응원해주신 장인어른 백우인, 장모님 김경미 감사드립니다. 또한 부족한 저희를 위해 지원해주신 형님 송근영, 처형 백은혜와 처조카 송지우 감사드립니다.

마지막으로 저를 믿고 인생의 동반자가 되어준 사랑하는 아내 백은샘과 축복속에서 태어난 아들 김준희에게 특별한 감사의 말씀을 전합니다. 세 식구 모두 어려운 환경속에서 각자의 역할을 수행하느라 힘들었겠지만 항상 우리 가족은 행복했고 저 또한 성공적으로 박사과정을 마무리 할 수 있게 되었습니다. 의미 있고 뜻 깊은 시간을 함께 보내게 된 우리 가족에게 다시 한 번 깊이 감사드립니다.

2022년 7월 24일

김 상 태