Doctoral Thesis

Weakly-Supervised Semantic Segmentation Using Deep Learning and Its Application for Food Images

Department of Informatics,
Graduate School of Information and Engineering
The University of Electro-Communications, Tokyo

Supervisor: Prof. Keiji Yanai

Author: Wataru Shimoda

March 2020

Abstract

Semantic segmentation is a task to recognize object categories in an image in pixel-level and is one of the major tasks in the research area. In semantic segmentation task, we need to recognize the category of objects for each pixel, and it can be regarded as more advanced task than image classification task. In recent years, the accuracy of semantic segmentation has improved dramatically using deep learning. However, it requires enormous costs to create training datasets for semantic segmentation because we have to assign class labels to each of the pixels over all the training images. To solve this problem, weakly-supervised segmentation methods have recently attracted attention. In general, while semantic segmentation requires the supervision for each of pixels in training images, image classification needs only the labels of objects shown in training images. Weakly-supervised semantic segmentation task require only image-level object labels in the same way as image classification tasks. If semantic segmentation under the weakly-supervised setting performs almost as high-performance as segmentation under the fully-supervised setting, the cost for annotation can be greatly reduced.

In this thesis, we have investigated methods to improve the accuracy of weakly supervised segmentation. In particular, we improved the accuracy of weakly supervised segmentation by two approaches: a visualization-based approach and a pseudo pixel-level labels-based approach. In Chapter 3 and Chapter 4, we proposed the visualization-based approach. In Chapter 5 and Chapter 6, we proposed the pseudo pixel-level labels-based approach.

To be concrete, in Chapter 3, we proposed a method to combine forward-based visualization and backward-based visualization, which robustly captures pixel-level target objects. In Chapter 4, we proposed a novel visualization method, which is based on only backward-based class-specific saliency maps. In this method, we used residual of the class-specific saliency maps that were obtained from the different class signals to alleviate problems caused in images including multi-class objects. In this thesis, we also have explored the pseudo pixel-level label-based methods. To train a mapping function for semantic segmentation, some weakly-supervised segmentation methods generate pseudo pixel-level labels in the weakly-supervised setting

and use them for training of segmentation models. We consider this approach as training from noisy data and focus on reduction of the noise from the noisy data. In Chapter 5, we estimated "easiness" of the segmentation for each image and retrieved "good seeds", and we used the "good seeds" for effective data augmentation. In Chapter 6, we estimated noise of the pseudo pixel-level labels in pixel-level and interpolate the noise to better labels using self-supervised difference detection based confidence maps. Furthermore, as applications of weakly supervised segmentation, we performed weakly-supervised food segmentation. In Chapter 8, we proposed a weakly supervised food segmentation method based on region proposals and backward-based saliency maps. In Chapter 9, we proposed a novel method to generate proposals that include many food objects. This method is an application of the method proposed in Chapter 4. In Chapter 10, we proposed a novel method to estimate food plate regions in weakly supervised settings. In this work, we found that we were able to estimate food plate regions without any pixel-wise annotation on food plates.

Abstract

領域分割は画像内における物体のクラスをピクセルレベルで認識するタスクであり、画像認識における重要なタスクの一つである。クラス分類タスクが画像内に写っている物体がなんであるかを認識するタスクであるのに対して、領域分割においてはピクセルごとに物体のカテゴリを認識する必要があり、より発展的なタスクであると言える。近年、深層学習を活用した手法により、領域分割の精度は飛躍的に向上した。しかしながら、領域分割における教師情報は出力と同様にピクセル単位でクラスのラベルを割り振る必要があり、これには膨大なコストが要求される。この問題を解決するために、近年弱教師あり領域分割手法が注目を集めている。領域分割がピクセル数分のクラスの教師情報を必要とするのに対して、クラス分類タスクにおける教師情報は画像内に写っている物体のラベルのみである。弱教師あり領域分割タスクは、クラス分類タスクの教師情報を用いて、領域分割モデルを学習させるという問題である。弱教師あり学習による領域分割が可能となれば，領域分割における学習データを収集するための大幅なコストの削減が期待できる。本研究においては, 弱教師あり領域分割の精度向上のための研究を行った。特に、可視化による弱教師あり領域分割の精度向上、仮の教師情報を用いた領域分割モデルの学習、２つのアプローチによる弱教師あり領域分割の精度向上を達成した。可視化による弱教師あり領域分割の精度向上においては、第 3 章において Forward の計算により得られる可視化結果、Backward の計算により得られる可視化結果を組み合わせることで精度を向上させる手法を提案した。また、第 4 章において Backward の計算により得られる可視化結果がマルチクラスの物体が映っている画像において、失敗しやすいという問題について、それぞれのクラスの可視化結果の差分をとることで、この問題を緩和する手法を提案した。可視化結果から領域分割への写像関数の精度向上においては、第 5 章において Pixel-level の仮の教師情報を用いて領域分割モデルを学習するというアプローチについて、これをノイズを含む教師情報からの学習であるととらえて、画像の領域分割の容易度を推定しノイズを含まないよい教師情報を使って Data augmentation をする手法を提案した。また、第 6 章において教師情報に含まれるノイズを画像単位ではなくピクセルレベルで推定し、さらにどのような教師情報に変更するのがよいかを、変化領域の推論結果による領域の信頼度から推定して上書きする手法を提案した。さらに、本研究では弱教師あり領域分割の応用として、食事画像の弱教師あり領域分割を行い、弱教師あり領域分割の有効性と食事画像における困難さと課題の解決方

法についての研究を行っている。第 8 章では、プロポーザルと Backward の計算により得られる可視化結果を用いた食事画像の弱教師あり領域分割手法を提案した。第 9 章では、第 4 章の手法を応用し画像中における食事らしい領域を推定し、食事領域プロポーザルを生成する手法を提案した。第 10 章では、食事クラス識別器と食事/非食事識別器の可視化結果の違いから皿領域を推定し、第 6 章の手法をベースとし皿領域の推論結果を活用した弱教師あり食事領域分割手法を提案した。

Acknowledgements

I am deeply grateful to my supervisor, Professor Keiji Yanai of the University of Electro-Communications, for his support and guidance as I proceeded with this research and compiled my dissertation. I am also deeply grateful to Professors Itsu Shono, Hiroki Takahashi, Naomi Hashimoto, and Yoichi Haneda, who served as reviewers and provided much advice in preparing this dissertation. Finally, I would like to thank my parents, grandparents, and uncles and aunts who have been so warmly watching over and supporting me.

Contents

Chapter 1

Introduction

1.1 Background

Semantic segmentation is a promising image recognition technology that enables detailed analysis of images for various practical applications. However, semantic segmentation methods require a large number of training images annotated with pixel-level labels which are costly to obtain. On the other hand, collecting images with image-level labels is easier than those with pixel-level labels, since many images attached with tags are available on hand-crafted open image datasets such as ImageNet [1] as well as on the Web. We illustrate the difference of annotations between image-level labels and pixel-level labels in Figure 1.1.

In recent years, various weakly-supervised semantic segmentation methods that require only image-level annotation have been proposed to resolve the annotation problem on semantic segmentation. Under the weakly-supervised settings, while we use only image-level labels in the training phase, we perform pixel-level semantic segmentation in the test phase. This setting is very challenging considering the large difference between image-level labels and pixel-level labels that are supervisions for an image-level classification task and a pixel-level semantic segmentation task, respectively. While we call the setting in which image-level labels are using in training as weakly-supervised segmentation, we call the default setting in which pixel-level labels are used for training as fully-supervised segmentation (Figure 1.2). Although weakly-supervised segmentation has got a lot of attention recently, there is still a large performance gap between fully-supervised and weakly-supervised methods regarding segmentation accuracy. In this thesis, we aim at mak-

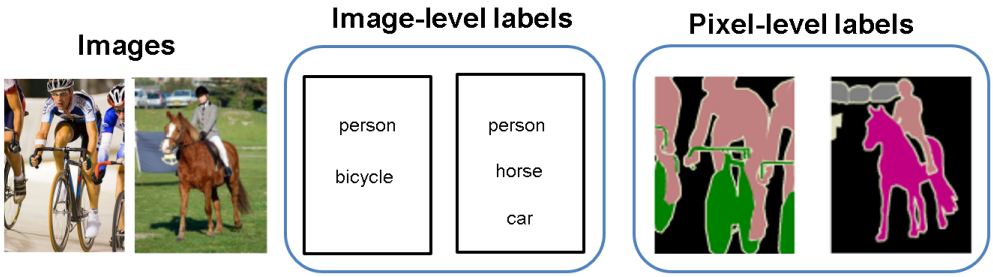

Figure 1.1: Sample images, corresponding image-level labels and pixel-level labels.

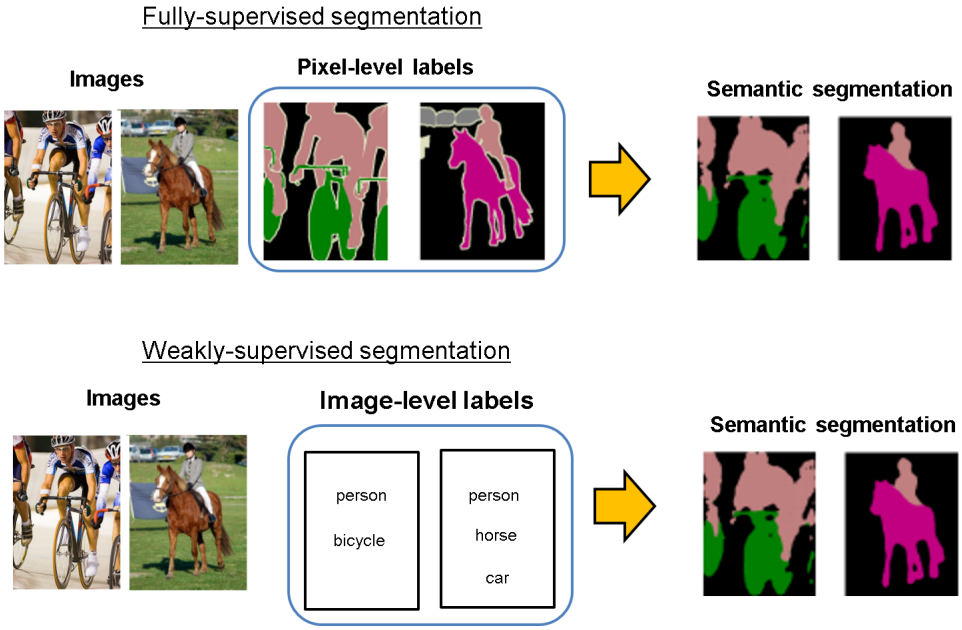

Figure 1.2: The concepts on fully-supervised and weakly-supervised segmentation.

ing the gap between them as narrow as possible. Recent weakly-supervised segmentation methods are roughly divided into two approaches: visualization of classification models and training segmentation models with pseudo pixel-level labels synthesized under the weakly-supervised setting. We tackled to improve the performance using the both of approaches with unique focuses, respectively. Though the ideas of the improvement for them are largely different, these approaches are not independent of each other. To be concrete, in general, the pseudo pixel-level labels are generated using visualization initially. Therefore improvement of the former approach would increase the performance of the latter approach. However, the final objective is the same in both of approaches. Therefore, we should consider the factor of improvement to combine them well. In particular, in the pseudo-label-based approach, we consider how to compensate insufficient components in the visualization approach. We show the illustration for the relationship between the two approaches in Figure 1.3. The improvement of the both of approaches are important and, in this thesis, we systematically explore the improvement of the performance of weakly-supervised segmentation methods through the two approaches.

Here, we introduce the both approaches: visualization of classification models and training segmentation models with synthesized pseudo pixel-level labels under the weakly-supervised setting. Visualization methods can highlight the regions whose pixels contribute recognition, We can extract object regions from visualization results because the contributed pixels have strong relationships with object regions. In this thesis, we focus on improving a backpropagation-based visualization method [2]. In general, visualization-based weakly-supervised segmentation methods tend to output ambiguous outlines but backpropagation-based visualization methods can extract information of outlines using guided backpropagation [3]. However, backpropagation-based visualization has difficulty for the images that include multiple-class objects. In backpropagation-based visualization, for the multiple-class objects, we visualize the targets of each of the classes by propagating a class signal from the top layer to the bottom layer using backpropagation. The class signal is gradually weakened through the layers, and the difference of the visualization would be getting smaller. It causes very similar visualization if we propagate different signals. The propagated signals in middle layers also have similar values for different signals. However, there are differences and we found that we can extract clear class-specific responses from the

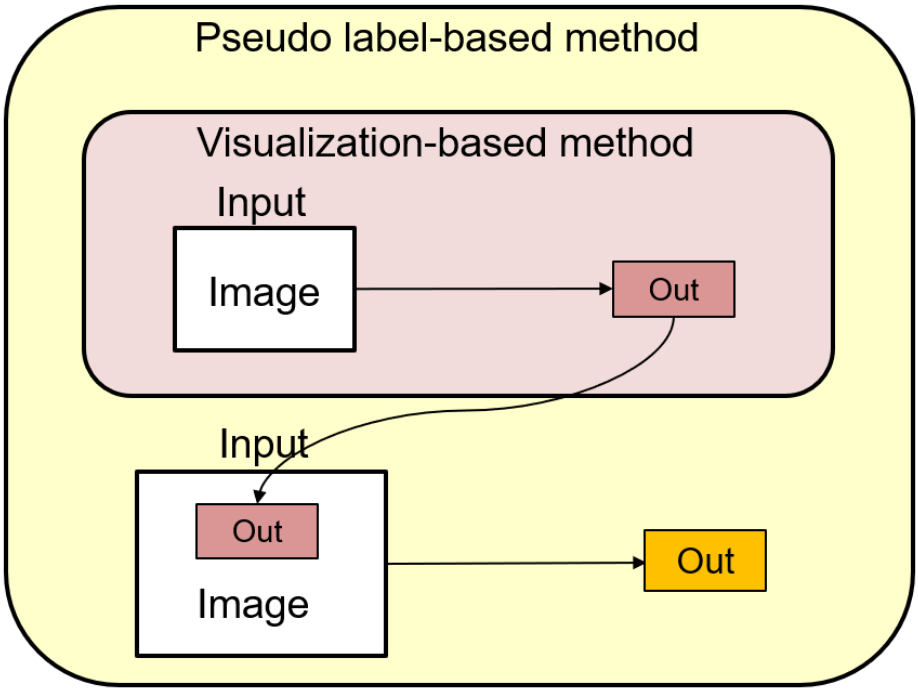

Figure 1.3: The illustration for the relationship between the two approaches: the visualization-based method and the psuedo pixel-level labels-based method.

difference between visualized signals. We demonstrated that we can obtain class-specific regions from the difference of the propagated signals, and we can boost backpropagation-based weakly-supervised segmentation accuracy using the difference of the propagated signals.

In this thesis, we also investigated weakly-supervised segmentation methods by training segmentation models with pseudo pixel-level labels synthesized under the weakly-supervised setting. This approach is not independent of visualization-based weakly-supervised segmentation because we synthesize the pseudo pixel-level labels by visualization in general. However, the focus of the method for the approach is different. Many existing works focused on how to make good pseudo pixel-level labels because the improvement of pseudo pixel-level labels would make better training for segmentation models. For examples, many works [4, 5, 6] have been reported that color information could assist to refine visualization-based weakly supervised segmentation, some researches [7] have demostrated that a re-training process could generate better pixel-level labels by updating pseudo pixel-level labels using a segmentation model with the pseudo pixel-level labels. In this thesis, being different from the existing works, we consider that the pseudo pixel-level labels can be regarded as training data containing many noisy labels and the noisy training data cause significant performance drop. Then we explored the methods that aim how to estimate noisy training data from pseudo pixel-level labels. We proposed two approaches to estimate noisy training data. First, we proposed a method to estimate noisy training data in image-level and to exclude noisy training data from training. Second, we proposed a method to estimate noisy training data in pixel-level and to interpolate noisy training data to estimated better training data. While the image-level noise reduction would have the problem of the trade-off between the quality and quantity of the training data, the pixel-level noise reduction approach is free from the problem. However, to interpolate noisy training data, we have to estimate not only noisy training data but also better labels for the data, that is a challenging task. We tackled the problem by estimating good regions and bad regions from two candidate masks using self-supervised difference detection.

In addition, towards real-world applications of weakly-supervised segmentation, we treat a food image domain. Food image segmentation is expected to be useful for computation of calorie estimation because we can estimate the relative amount of foods from food image regions, and there are strong relationships between calorie of foods and amount of foods. However, preparing

pixel-wise label of various foods is very time-consuming and costly. In fact, there exists no large-scale food image dataset with pixel-wise annotation. If weakly-supervised food segmentation methods become enough accurate, it would be beneficial for food image domain. Then we consider weakly-supervised food segmentation and we adapted weakly-supervised segmentation methods from the general image domain to the food image domain.

In this thesis, we verified the effectiveness and the limitation of the weakly-supervised segmentation on food images. We demonstrated that if a method of weakly-supervised segmentation achieved the good performance on general object datasets, it often dropped the performance on the food domain. However, we demonstrated that we could recovery the performance by several ways specified for the food image domain.

In particular, we verified the effectiveness of three weakly-supervised segmentation methods. In chapter 8, we adapted backpropagation-based visualization method [2] to food images, and combined it with a proposal-based detection approach to keep the accuracy. In chapter 9, we adapted the improved backpropagation-based visualization method [2] to food images , and used it as a foodness proposal method to keep the accuracy and computational cost. In chapter 10, we adapted the pseudo label-based method [2] to food images , and we also proposed visualization-based food plate estimation method. We demonstrated that the performance of weakly-supervised food segmentation can be boosted by utilizing pseudo food plate segmentation.

To summarize the above, in this thesis, we investigated weakly-supervised segmentation methods through the two approaches. We also adapted weakly-supervised segmentation methods to food images and explored the methods to keep the performance on food images.

1.2 Structure of thesis

In this section, we explain the structure of this thesis. First, in Chapter 2, we introduce the related works on fully-supervised segmentation and weakly-supervised segmentation. In Chapters 3 and 4, we describe the studies for improving of visualization-based weakly-supervised segmentation. In Chapter 3, we propose a method to combine forward-based visualization and backward-based visualization, which robustly captures pixel-level target objects. In Chapter 4, we propose a novel visualization method, which is based on only backward-based class-specific saliency maps. Though the

backward-based class-specific saliency maps can capture more clearer outlines of the target objects than forward-based visualization, however, they tend to confuse class information and it causes problems for images that include multiple-class objects. In Chapter 4, to alleviate the problem, we propose a novel approach to use subtraction of the class-specific saliency maps that are obtained from the different class signals. The research of Chapter 3 is published as [C4], and the research of Chapter 4 is published as [8] and [9], respectively.

CNN-based visualization methods of weakly-supervised segmentation greatly improved the performance compared to non CNN-based weakly-supervised segmentation methods. However, in general, visualization is not equal to segmentation, and the performance gap between fully supervised segmentation and weakly-supervised segmentation remains still large. Therefore, we also explored the methods to adapt visualization methods to segmentation efficiently. Wei et al [7] proposed a method to generate pseudo pixel-level labels in the weakly-supervised setting, and use the pseudo pixel-level labels for training of the fully-supervised segmentation models. This approach has gathered large attention, and we proposed a method to improve the approach in Chapter 5. In Chapter 5, we consider the pseudo pixel-level labels generated in the weakly-supervised settings include many noise, and we focus on reduction of the noise from the generated pseudo pixel-level labels. To be concrete, we estimated "easiness" of the segmentation for each image and retrieved "good seeds", and we used the "good seeds" for effective data augmentation for the training of the segmentation models. In Chapter 6, we propose a novel method to integrate two types of the pseudo pixel-level labels by self-supervised difference detection based confidence maps. While the method of Chapter 5 estimates noise of the pseudo pixel-level labels in image-level, the method of the Chapter 6 estimates noise of the pseudo pixel-level labels in pixel-level and interpolate the noise to better labels. The research of Chapter 5 is presented in [10] and [9], and the research of Chapter 6 is presented in [11] and [12], respectively.

We consider weakly-supervised food image segmentation is one of a promising task as applications of the weakly-supervised segmentation, and we explore the methods of adaptation of the weakly-supervised segmentation methods to weakly-supervised food segmentation. In Chapter 7, we describe related works on food image recognition. In Chapters 8, 9, and 10 we describe studies on weakly-supervised food detection and segmentation. In Chapter 8,

we proposed a weakly-supervised food segmentation method based on proposals and backward-based saliency maps. As our best knowledge, this is the first work on weakly-supervised food detection and segmentation. In Chapter 9, we proposed a novel method to generate proposals that include many food objects. This method is an application of the Chapter 4. We compute food-specific saliency maps from food category signals, and generate many food specific proposals. In Chapter 10, we proposed a novel method to estimate food plate regions in weakly-supervised settings. In this work, we found that we can estimate food plate regions without any annotation about the food plate. Only from image-level food class labels, we estimate food plate regions by the difference of visualization between a food category classifier and a food/non-food classifier. The weakly-supervised segmentation model of this method is based on the work of Chapter 6. The research of Chapter 8 is presented in [13], the research of Chapter 9 is presented in [14] and [15], and the research of Chapter 10 is presented in [16], respectively. Finally, in Chapter 11.2, we summarize this paper and discusses future issues. Figure 1.4 shows the structure of this thesis.

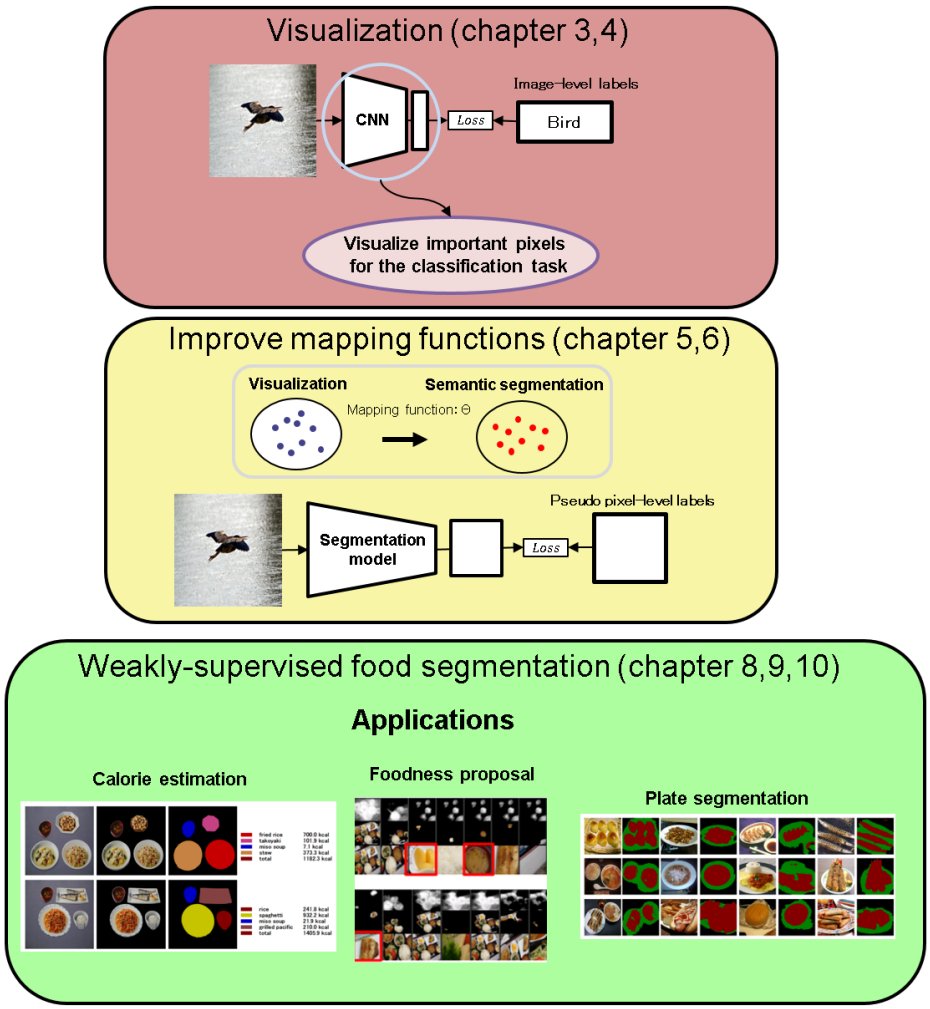

Figure 1.4: The structure of this thesis.

Chapter 2

Related work of weakly supervised segmentation

Recently, semantic segmentation using deep learning is being explored actively, and the accuracy was much improved compared to the conventional methods based on hand-crafted features. In this chapter, first, we introduce related works for fully-supervised semantic segmentation, and next we explain related works for weakly-supervised segmentation.

2.1 Fully-supervised semantic segmentation

Girshick et al. [17] and Hariharan et al. [18] proposed region proposal based semantic segmentation methods utilizing the ability of CNN. The authors generated around two thousand region proposals using Selective Search [19], and applied a classification model to each of the proposals. After the classification step, they integrated all the classification results and generated the final object regions. Though these methods outperformed the conventional methods, this approach caused long processing time for CNN-based image classification of many regions proposals.

While Girshick et al. [17] and Hariharan et al. [18] utilized of the ability CNN on the classification for semantic segmentation, Long et al. [20] and Mostajabi et al. [21] achieved robust and accurate semantic segmentation using CNN in a hierarchical way. A CNN is much different from methods based on hand-crafted features because it has a multi-layered structure consisting of convolutional and pooling layers. The pooling layers decrease location

information gradually. Therefore, it is difficult to keep location information in the upper layers. Long et al. [20] and Mostajabi et al. [21] showed that spatial information can be complemented in the upper activations by simple linear interpolation in the middle layers and integration. Mostajabi et al. [21] proposed Zoom Out Features(ZOF). ZOF used super-pixels for averaging feed-forward activations and upsampled the averaged activations in middle layers for integration. Long et al. [20] proposed a fully convolutional network (FCN). FCN replaced class score vectors with class score maps as outputs of a CNN. This idea was originally proposed by Sermanet et al. [22], which plays important roles to raise performance on CNN-based segmentation. We show the figure for the FCN in Figure 2.1. This can be used as unary priors of CRF [23, 24, 25].

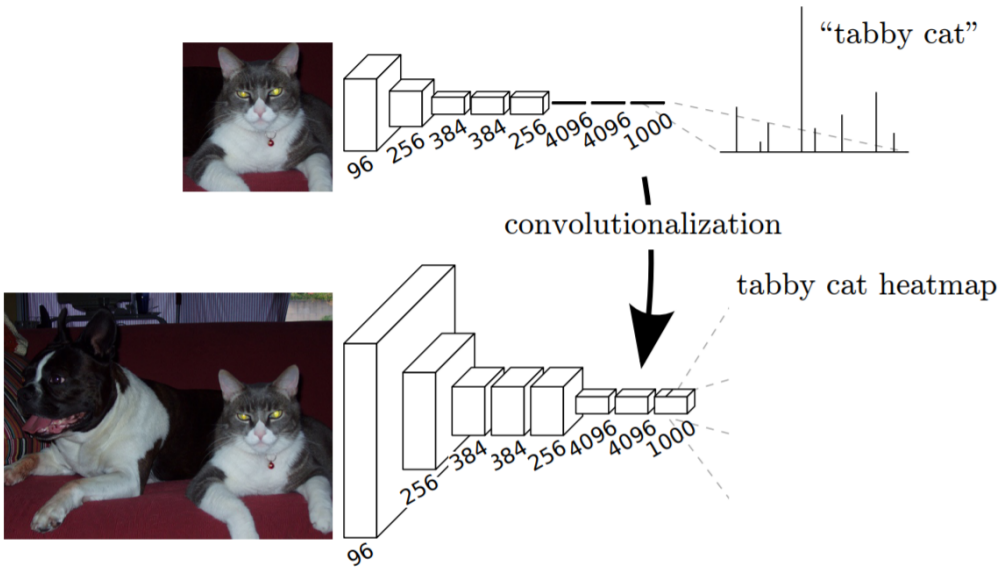

Figure 2.1: The illustration of FCN. This figure is cited from [20].

Chen et al. [26] and Yu et al. [27] showed that large receptive fields are very important for raising semantic segmentation. Both of the works used dilated convolutional layers. Note that Chen et al. called them as atrous convolution. The algorithm convoluted around pixels sparsely, then unify distant pixel information at the inference of each pixel category. We show the figure for the dilation in Figure 2.2. Zhao et al. [28] showed another approach for expanding the field of view. They used spatial pyramid pooling [29],

which concatenates multi-scale pooled features. There exists a more easier approach to change the field of view. Chen et al. [30] showed that the scale of input images can easily change the field of view. Chen et al. [31] proposed DeeplabV3, which utilizes Atrous Spatial Pyramid Pooling(ASPP). ASPP extended pyramid pooling in [28] by exploiting atrous convolution. Chen et al. also proposed DeeplabV3+ [32]. They demonstrated that Xception model [33] and a decoder network can boost the accuracy of semgnetation.

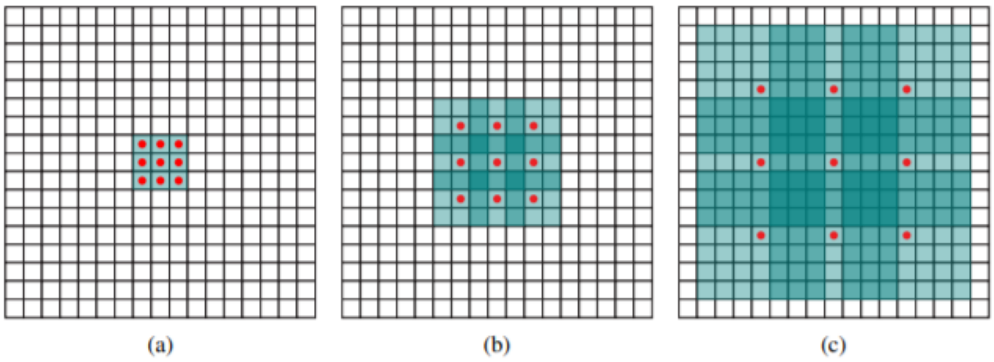

Figure 2.2: The illustration of dilated convolution. This figure is cited from [27].

2.1.1 Visualization Methods

FCN can accept input images of arbitrary sizes by replacing fully-connected layers with convolutional layers. Oquab et al. [34] proposed to use Global Max Pooling (GMP) before the last layer for classification in the training time, and took away the GMP in the test phase. Oquab et al. showed that FCN can be trained using image-level labels and the FCN trained with this approach outputs rough object location.

After that, some derived methods employing GMP were proposed by Pinheiro et al. [35]. As a CNN-based method, Zhou et al. [36] proposed Class Activation Map (CAM), which is a visualization method based on forwarding based activation. CAM is one of the current standard visualization method and this method is used various weakly supervised segmentation methods.

Generally, we optimize CNN parameters so as to minimize the loss between output values and ground truth values. The derivatives of the loss

function are propagated from the top layers to the lower layers by Back-Propagation(BP). Simonyan et al. [2] proposed a BP-based weakly supervised segmentation method using the derivatives. Springenberg et al. [3] produced Guided Back-Propagation(GBP), which is a modified method of BP-based method. They limit the derivatives to only positive values and showed that the limited derivatives capture accurate outlines.

In addition, Wei et al. proposed Adversarial erasing [37], which is a novel technique of visualization. In visualization, there are strong responses and weak responses that indicate a degree of contribution to the decision of the classification model. The regions have strong response are obtained easily, however, in segmentation, the regions have weak response are also important. To mine the regions have weak responses, the authors proposed an approach that erase regions have strong reponse by filling constant values and perform visualization again. The authors demonstrated that the regions have weak responses also can be mined by repeating visualization and erasing.

2.1.2 Mapping from visualization to segmentation

In weakly-supervised segmentation, many methods have been studied visualization-based approaches. Visualization highlight regions whose pixels contribute to recognition, and the highlighted regions correspond to target objects in generall. However, visualization often does not match actual segmentation. Therefore, to improve further from visualization, we need to consider the mapping from the visualization results to the semantic segmentation. Here, we introduce related works for mapping functions for visualization to segmentation.

Region refinement for weakly supervised segmentation results using CRF In general, outlines of outputs of FCN tend to be ambiguous under the weakly-supervised setting. CRF [38] is a method for refinement of the ambiguous outlines by considering strength of the connection in pixels using the color and location information. Chen et al. [4] and Pathak et al. [5] demonstrated that CRF can also be used as a post-processing method on weakly-supervised segmentation. Kolesnikov et al. [6] proposed a method considering CRF when optimizing the loss for the training of the parameters. Ahn et al. [39] focus on pixel-level similarity and propose a novel method: Pixel-level Semantic Affinity (PSA), which learns pixel-level similarity from CRF. We show the figure for the PSA in Figure 2.5.

Figure 2.3: The examples of the forwarding based visualization. This figure is cited from [36].

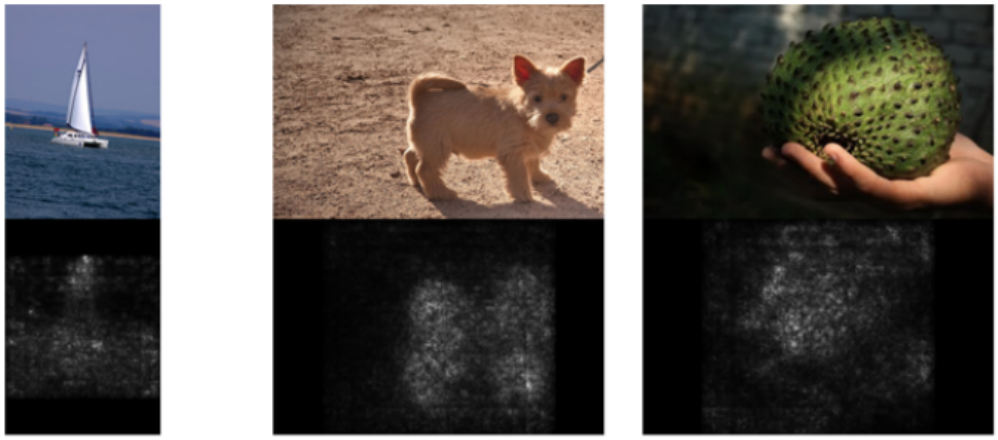

Figure 2.4: The examples of the backwading based visualization. This figure is cited from [2].

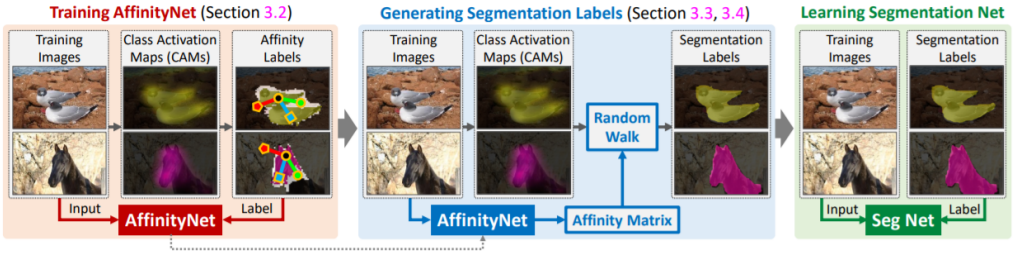

Figure 2.5: The illustration of PSA. This figure is cited from [39].

Furthermore, various researches employed CRF [8, 40, 41, 42, 7, 37, 43, 44]. As we stated, CRF plays an important role to improve the accuracy of weakly supervised segmentation.

Training fully supervised segmentation model under weakly supervised setting Recent years, an approah has been proposed to train fully-supervised semantic segmentation model under a weakly supervised setting, that have become key role to boost the accuracy of weakly-supervised segmentation. First, Papandreou et al. [45] proposed a method to train FCN under weakly-supervised setting using a global max pooling. Wei et al. [7] proposed a novel approach Simple To Complex (STC) framework. The authors trained a fully-supervised model using saliency-maps [46] based pixel-level labels. Wei et al. [7] also showed that re-training process can boost the accuracy of weakly-supervised segmentation model. In the re-training process, The authors generate new pixel-level labels from the trained segmentation model, and they re-train the segmentation model with the new pixel-level labels. We show the figure for the STC in Figure 2.6.

Generating pixel-level labels during the training of a fully supervised segmentation model Constrained convolutional neural network (CCNN) [5] and EM-adopt [4] trained FCN using generated pixel-level labels. Different from STC, these methods generated pixel-level labels during training in each step. Both of the methods adopted a similar approach that made constraint by setting the ratios of the foreground and the background in an image and changed pixel-level labels within the ratio. Wei et al. [37] proposed an online prohibitive segmentation learning (PSL). The authors generated pseudo pixel-level labels using visualization and trained semantic segmentation model using both of the pixel-level labels and the outputs of the semantic segmentation models. We consider that, in this approach, the

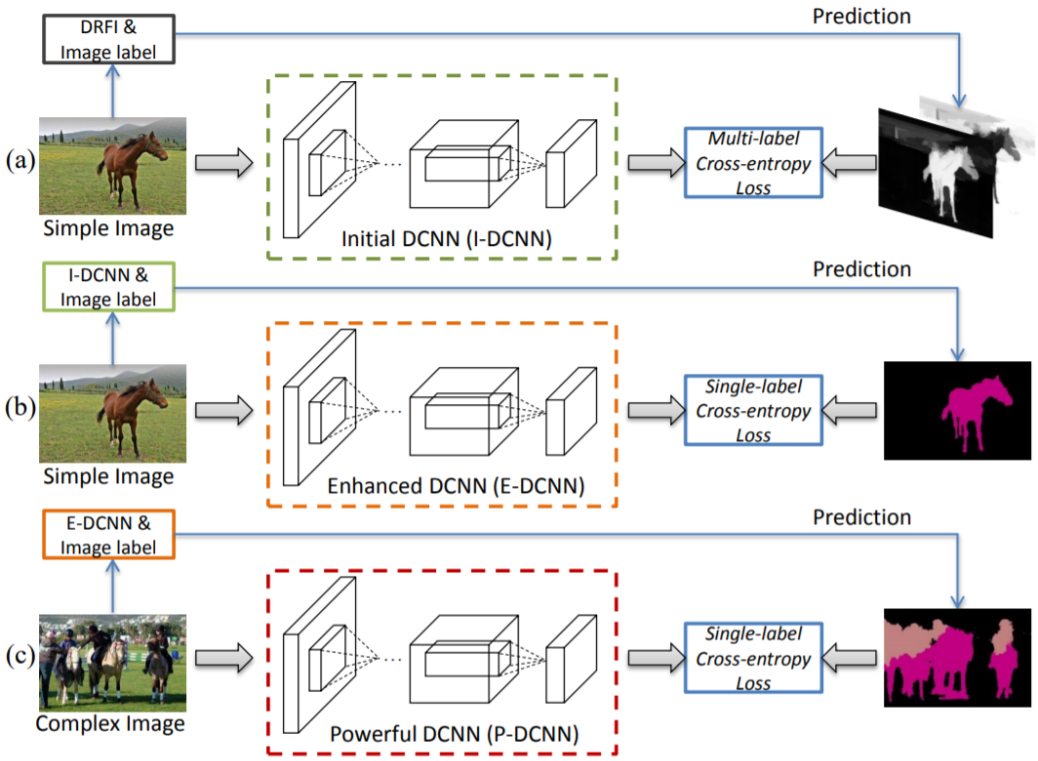

Figure 2.6: The illustration of the simple to complex framework. This figure is cited from [7].

pseudo pixel-level labels helped the training of segmentation model as constraint, which works in the similar way to the before constraint proposed by Pathak et al. [5] and Chen et al. [4]. Huang et al. [47] proposed deep seeded region growing (DSRG). DSRG is a method that expands seed regions during training. Before training, the authors prepared pixel-level seed labels that have unlabeled regions for unconsidered pixels.

2.1.3 Weakly supervised segmentation methods using additional information

Some researches demonstrated that additional supervision can improve the accuracy of weakly-supervised segmentation. Here, we introduced weakly-supervised segmentation using additional supervision.

Additional annotations Some researchers have proposed the bounding box annotation for weakly supervised segmentation [4], and they showed that the bounding box annotation substantially boosted performance. Chen et al. [4] showed that bounding box annotation can boost the performance of weakly-supervised segmentation with a simple approach. The bounding box annotation is less costly than pixel-wise annotation but still costly than only image-level annotation. As weaker additional annotation, Bearman et al. [48] proposed point annotation and scribble annotation were also proposed.

Additional data Methods that use web images for weakly supervised segmentation has also been proposed [35, 7, 49, 44]. There are also reports that web videos were helpful for improving the weakly supervised segmentation accuracy [50, 51]. Recently, saliency maps trained with other training data are widely used, and many works have reported this approach could substantially boost performance [52, 37, 53, 47, 43, 54, 55]. Hu et al. [56] showed that instance-level saliency maps for weakly supervised segmentation, though its cost higher than normal saliency maps. Saliency maps are helpful in various situations; however, fully-supervised saliency models would be affected by the domain of the training data, which may cause some problems on applications. Weakly supervised segmentation methods without saliency maps are also beneficial. In this thesis, we do not use any additional information. We use only images of PASCAL VOC with image-level labels and pre-trained models trained with ImageNet.

2.2 Dataset

We evaluated the proposed methods on PASCAL VOC 2012 segmentation benchmark[57]. We followed the common practice to augment the training data provided by [58]. There are 10,582 training images, 1,449 validation images and 1,456 test images. In this benchmark, although the PASCAL VOC dataset contains 20 classes, we need to classify 21 classes including the background class. Note that PASCAL VOC 2012 is the most common benchmark dataset in both fully-supervised segmentation and weakly-supervised segmentation.

Chapter 3

Visualization using forward and backward

In this chapter, we propose a method for CNN-based semantic segmentation method. Different from existing methods, the proposed method integrates both feed-forward activations and backpropagation (BP) based saliency maps. The feed-forward activations help to discriminate the object category and BP-based saliency maps help to detect background.

As feed-forward activations, we used and compared two types of methods: Zoom-Out Features(ZOF)[21] and Fully Convolutional Network(FCN)[20]. Furthermore, for BP-based saliency maps, we used a method proposed by Simonyan et al. [2]. But the BP-based saliency maps has several problems for using weakly-supervised segmentation, then we improved the method [2] to obtain denser and clearer saliency maps by up-sampling saliency maps of the intermediate layers and aggregating them. As a CNN, we use the VGG-16 model [2] pre-trained with 1000-class ILSVRC datasets and fine-tuned with multi-labeled training images in the PASCAL VOC dataset using only image-level labels. Figure 3.1 and Figure 3.2 show the processing flow of our methods.

To summarize our contribution in this chapter, they are as follows:

- We propose a new method which uses both feature maps generated from the forwarding of the CNN and back-propagation based object saliency maps.
- We show the effectiveness of the proposed method by the experiments with the Pascal VOC dataset.

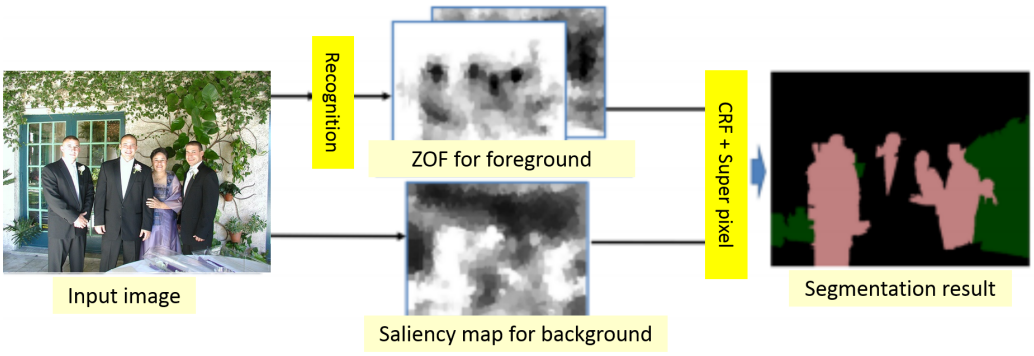

Figure 3.1: Processing flow of the ZOF base method.

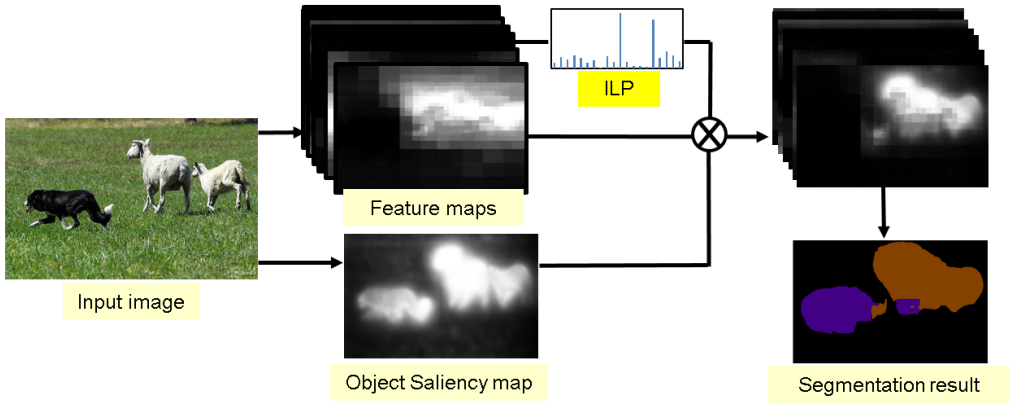

Figure 3.2: Processing flow of the FCN base method.

3.1 CNN model

In this work, we use VGG-16 [59] as a basic CNN architecture. In our framework, we fine-tune a CNN with training images having no pixel-wise and bounding box information but image-level multi-label annotation. To carry out multi-label training of the CNN, we use Sigmoid cross entropy loss which is a standard loss function for multi-label annotation instead of soft-max loss. The Sigmoid cross entropy loss function is represented in the following equation:

$$\text{loss} = \sum_{k=1}[p_n \log \hat{p_n} - (1 - p_n)\log(1 - \hat{p_n})] \tag{3.1}$$

where K is the number of classes, $p_n = \{0, 1\}$ which represents the existence of the corresponding class label, and $\hat{p_n}$ means the output of Sigmoid function of the class score $f_k(x)$ represented in the following equation:

$$\hat{p_n} = \frac{1}{1 + e^{-f_k(x)}} \tag{3.2}$$

3.2 Zoom-out features

Zoom-out features is a method proposed by Mostajabi et al [21] as a fully-supervised semantic segmentation method. This method averages feed-forward activations in super-pixels and then obtains features for each super-pixel. They up-sampled all the feature maps so that their size becomes the same as a given image, and integrated up-sampled feature maps to estimate object locations more accurately.

In this chapter, we apply Zoom-out features (ZOF) [21] to weakly supervised segmentation. Note that we use super-pixels as region representation in the same way as [21]. To adapt ZOF to weakly-supervised learning, we estimate correspondence using multiple instance learning (MIL) which is one of the common methods to estimate regions corresponding to given labels, and we adopt mi-SVM [60] as a method of multiple instance learning which uses SVM iteratively. Given a certain class, we regard images having the label of the target class as positive bags, and images having no label of the target class as negative bags. Positive bags contain more positive regions, while negative bags contain no positive regions. Because MIL can estimate positive regions, we can estimate positive super-pixels by using MIL. As a feature representation of super-pixels, we use ZOF. We extract ZOF from each of super-pixel regions.

3.3 Fully convolutional network

Fully convolutinal network (FCN) is originally proposed by Sermanet et al. [22], and extended by Long et al. [20]. FCN can deal with an arbitrary size for input images because all the full connection layers are replaced with 1×1 convolutional layers. Using FCN, we can obtain a coarse object heatmap at the last convolutional layer. Therefore, in FCN, we can estimate object location directly without a second training step such as mi-SVM different from Section 3.2.

There are some works [45] [4] adapting FCN to weakly-superveised segmentation. They train FCN using not pixel-wise annotation as well as bounding box annotation but only image-level annotation with global-max-pooling for coarse object heatmaps at last layer of CNN.

3.4 Object saliency maps

Simonyan et al. regarded the derivatives of the class score with respect to the input image as class saliency maps. However, the position of an input image is the furthermost from the class score output on the deep CNN, which sometimes causes weakening or vanishing of gradients. Instead of the derivatives of the input image, we use the derivatives of relatively upper intermediate layers which are expected to retain more high-level semantic information. We select the maximum absolute values of the derivatives with respect to the feature maps at each location of feature maps across all the kernels, and up-sample them with bilinear interpolation so that their size becomes the same as an input image. Finally, we average them to obtain one saliency map. The idea of aggregating of information extracted from multiple feature layers was inspired by the work of [20], although they extracted not CNN derivatives but feature maps calculated by feed-forwarding.

The class score derivative v_i of the i-th layer is the derivative of class score Sc with respect to the layer L_i at the point (activation signal) L_i:

$$v_i = \left. \frac{\partial S_c}{\partial L} \right|_{L_i} \tag{3.3}$$

v_i can be computed by back-propagation. After obtained v_i, we up-sample it to w_i with bilinear interpolation so that the size of an 2-D map of v_i becomes the same as an input image. Next, the saliency map $m_{i,x,y}$ is computed as

$$\hat{w}^c_{i,h_i(x,y,k)} = \sum_{c \in candidate} w_{i,h_i(x,y,k)^c} - w_{i,h_i(x,y,k)^{c'}} \tag{3.4}$$

$$m_{i,x,y} = \max |w_{i,h_i(x,y,k)}| \tag{3.5}$$

where $h_i(x, y, k)$ is the index of the element of w_i, k represents kernel. Then, we aggregate $m_{i,x,y}$ for each target layer and obtaine a dense saliency maps $g_{x,y}$ are represented as:

$$g_{x,y} = \frac{1}{L} \sum \tanh(\alpha \cdot m_{i,x,y}) \tag{3.6}$$

where L is the number of layer to aggregate, α is scalar.

To estimate object saliency maps, we use guided back propagation (GBP) proposed by Springerberg et al. [3] instead of the normal back propagation (BP) used in the work on class saliency map estimation by Simonyan et al. [2]. Only the ways to back propagation through ReLUs (rectified linear units) are different. In the GBP, only positive loss values are propagated back to the previous layers through ReLUs as follows:

$$\text{BP} : \frac{dz^i}{dx^i} = \frac{dz^{i+1}}{dx^{i+1}} \cdot (conv^{i+1} > 0) \tag{3.7}$$

$$\text{GBP} : \frac{dz^i}{dx^i} = \left(\frac{dz^{i+1}}{dx^{i+1}} > 0\right) \cdot (conv^{i+1} > 0) \tag{3.8}$$

GBP can emphasize edges of objects, which is a desirable property for estimating object saliency maps. Figure 3.4 shows up-sampled saliency maps of "bicycle" in the image-level and four intermediate layers of VGG16 [59], w_i, $w_i^{conv2\_1}$, $w_i^{conv3\_1}$, $w_i^{conv4\_1}$, $w_i^{conv5\_1}$, obtained by both BP and guided BP as well as feature maps in case of back-propagating the "bicycle" signal to the network.

By aggregating saliency maps in the intermediate layers, we can obtain more clear object saliency maps. In this chapter, we use this saliency map for estimating background regions.

3.5 Integration feature maps with saliency maps

In this section, we denote for approaches to integrate feed-forward activations with BP-based saliency maps. We explore different integration methods for two types of feed-forward activations based on ZOF and FCN, respectively. In the ZOF based method, we adopt CRF using super-pixels. On the other hand, in the FCN based method, we directly use saliency maps for probability maps, which represent background. To be concrete, we use BP-based saliency maps in the similar manner to an approach proposed in [35] as smoothing prior.

3.5.1 CRF with Superpixel for ZOF

Since each Zoom-out features corresponds to each superpixel, we regard super-pixels are nodes in the CRF graph. We assume y_p is a label of su-

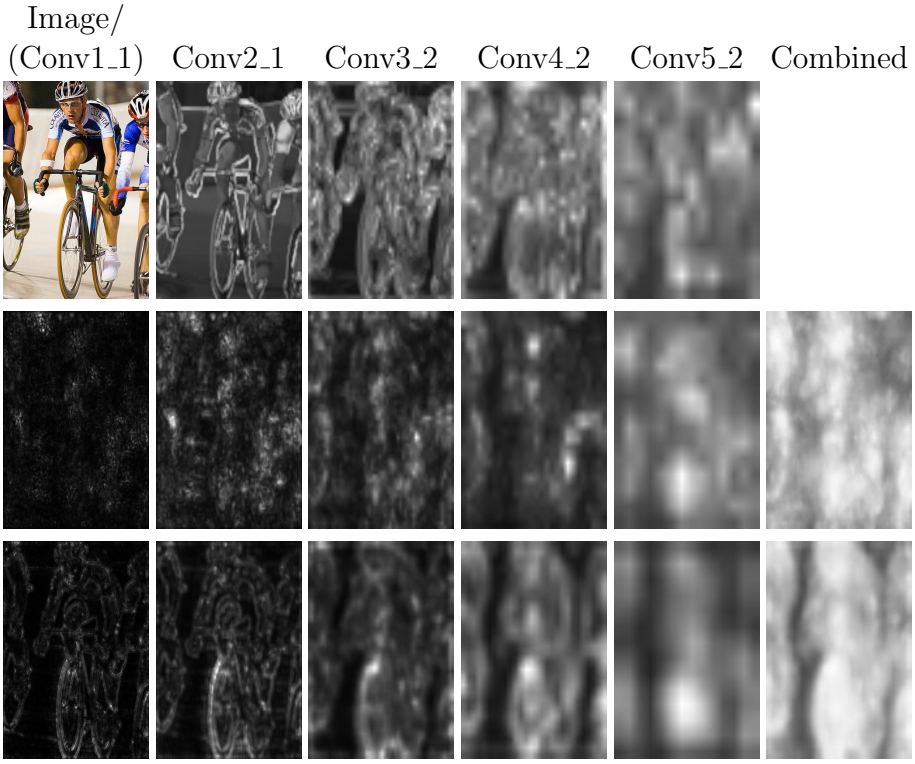

Figure 3.3: First row: feature maps (activations) (the given image itself at image-level), second row: saliency maps by back-propagation, third row: saliency maps by guided back-propagation, Columns: image-level, conv2_1, conv3_2, conv4_2, and conv5_2.

perpixel p in Image I, and $\mathbf{y}$ is a aggregated vector of all the y_p, energy function of CRF is defined as follows:

$$E(\mathbf{y}|I) = \sum_{p \in P} U(y_p|I) + \sum_{p,q \in N} V(y_p, y_q|I) \tag{3.9}$$

$U(\cdot)$ is a unary term, and $V(\cdot)$ is a pairwise term. We use as unary potential $U(y_p|I) = -\log Pr(y_p|I)$, where $Pr(y_p|I)$ is the label assignment probability at each super pixel y_p on image I. We obtained the label assignment probability of each object class in foreground by adapting linear SVM which is trained using mi-SVM [60] to zoom-out features. We use saliency maps obtained by backpropagation for background probability.

We define a pairwise term as follows referring to [61, 62, 63]:

$$V(y_p, y_q|I) = \left(\frac{L(p,q)}{1+\|p-q\|}\right)[y_p \neq y_q] \tag{3.10}$$

where $\|p-q\|$ is a distance between superpixel p and q regarding LUV color vectors, and $L(p,q)$ is the length of the boundaries shared by superpixel p and q.

3.5.2 Saliency maps for smoothing prior

We refine coarse object heatmaps obtained by FCN using BP-based saliency maps. We up-sampled coarse object heatmaps and saliency maps to unify sizes of height and width in advance. Here, $f_{x,y}$ represents coarse object heatmaps and $g_{x,y}$ represents saliency maps at pixel (x, y). The segmentation result $h_{x,y}$ is obtained as follows:

$$h_{x,y} = \begin{cases} k, & \text{if } \arg\max_{c \in C} f^c_{x,y} * g_{x,y} > \delta \\ k_{bg} & \text{otherwise} \end{cases} \tag{3.11}$$

where,C is set of object class, δ is a fix value threshold.

3.6 Experiments

3.6.1 Experimental setup

3.6.1.1 Training of CNN

We used 16-layered CNN, VGG-16 [2] pre-trained with ImageNet 1000 Categories as a basic CNN architecture. We fine-tuned VGG-16 using PASCAL VOC training dataset and train_aug by Hariharan et al [58] with Sigmoid entropy loss for multi-label training as described in Section 3.1 in batch size 16 and learning rate 1e-5, momentum 0.9 and weight decay 0.0005. For the first 30000 iterations, we fine-tuned only the upper layers of the modified VGG-16 than Pool_5, and for the next 20000 iterations, we fine-tuned all the layers.

3.6.1.2 Zoom-out features

We extracted about 500 super-pixels by the SLIC super-pixels [64] from all the training images, and calculated Zoom-out features (ZOF) [21]. While in [21] they extracted ZOF from all the layers of VGG-16, 13 convolutional layers and 3 fully connected layers, we extracted ZOF from 13 convolutional layers, pool5 and fc7. When applying mi-SVM [60] for each class, we used 500 images of the target class as positive samples and 1000 images of the other classes than the target class. We used the classification result of the CNN to limit object class for CRF.

3.6.1.3 Fully convolutional networks

In the FCN based method, we used image-level recognition results as image-level prior(ILP) for post processing which is noted by [35] to consider global context. Specifically, we adapted global-max-pooling to heatmaps in a manner similar to the training phase and multiplied each class pooled score and each class heatmap values.

Our approach differs from [35] on a method of correcting coarse heatmaps. Papandrew et al. [35] used MCG [65] which is known as region proposal for correcting heatmaps they call smoothing priors and made a foreground mask by aggregating objectness scores of about 2000 region candidates. On the other hand, we used saliency maps obtained by backpropagation and corrected coarse heatmaps in the similar way to MCG smoothing priors. Then, we compared our method with MCG smoothing priors.

3.6.1.4 Saliency maps

Backpropagation needs computational cost more than feed-forward processing, although there are little difference in derivatives obtained from signals of each class. Thus, we computed backward onece for an image even if there are several class objects. We predicted presence/absence of objects in the image by feed-forwarding, and made a signal which is the same dimension to CNN output. Simply, we prepare a vector, which has the same channnels to the number of the class for the signal and set 1 for presence classes and 0 for absence classes based on the image-level label. We propagated the signal by backpropagation from the top of the convolutional layer and extructed derivatives from layer conv3_2 and conv4_2 and conv5_2 and aggregated by

the Equation.3.6.

3.6.2 Results

Table 3.1 and Table 3.2 show the results on the proposed methods and some other state-of-the-art methods on the validation set and the test set. Note that although [35] showed the high performance, they used 700,000 additional training images selected from ImageNet which about 70 times as large as the common additional training data [58]. We report the results for three our methods and compare with other state-of-the-art weakly-supervesed segmentation methods. (ZOF with GBP) and (FCN with GBP) are methods integrating feature maps and saliency maps obtained by guided backpropagation. We also report (FCN with MCG) result to compare the effect of saliency maps with smoothing priors of (MIL-seg)[35] which is generated by MCG, due to unfair factors such as additional images for training and lack of deep-supervision of the CNN , [35] used overfeat-base segmentation network [66]. Therefore we compare saliency maps with smoothing priors [35] from the results of (FCN with GBP) and (FCN with MCG).

(ZOF with GBP) achieved better or comparable results. (FCN with GBP) outperformed MIL-FCN [45], EM-Adapt [4], CCNN [5] using only train_aug samples provided by Hariharan et al. [58] on validation set and test set. (FCN with GBP) also achieved MIL-ILP-seg [35] using additional images on test set in spite of fewer images for training. We show some example results in Figure 4.8.

We also compared (FCN with GBP) with (FCN with MCG), which is excluded unfair factors for verifing effect on saliency maps and smoothing priors using MCG. As a result, (FCN with GBP) outperformed (FCN with MCG) clearly , i.e., 33.8% vs. 41.4%(validation set),33.1% vs. 40.7%(test set). This indicates that combining saliency maps obtained by guided backpropagation and feature maps of CNN are effective for the weakly-supervised segmentation task. Figure 3.5 shows the comparison between saliency maps and MCG priors.

Table 3.1: Results on PASCAL VOC 2011 validation set. (Note that MIL-sppxl/bb/seg* used additional large training sets.)

Methods	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mean
MIL-FCN [45]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	25.7
EM-Adapt [4]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	38.2
CCNN [5]	65.9	23.8	17.6	22.8	19.4	36.2	47.3	46.9	47.0	16.3	36.1	22.2	43.2	33.7	44.9	39.8	29.9	33.4	22.2	**38.8**	36.3	34.5
MIL-sppxl* [35]	77.2	37.3	18.4	25.4	28.2	31.9	41.6	48.1	50.7	12.7	45.7	14.6	50.9	44.1	39.2	37.9	28.3	44.0	19.6	37.6	35.0	36.6
MIL-bb* [35]	78.6	46.9	18.6	27.9	30.7	38.4	44.0	49.6	49.8	11.6	44.7	14.6	50.4	44.7	40.8	38.5	26.0	45.0	20.5	36.9	34.8	37.8
MIL-seg* [35]	**79.6**	**50.2**	21.6	**40.6**	**34.9**	**40.5**	45.9	**51.5**	**60.6**	12.6	**51.2**	11.6	**56.8**	**52.9**	44.8	42.7	31.2	**55.4**	21.5	**38.8**	**36.9**	**42.0**
ZOF with GBP (ours)	70.6	44.4	24.7	37.5	16.4	33.3	**60.6**	35.5	58.8	5.5	45.5	15.9	53.4	41.1	**54.8**	39.6	24.2	52.1	18.4	38.6	27.5	38.1
FCN with MCG (ours)	71.0	21.9	18.5	22.0	12.8	34.6	37.5	43.3	47.1	17.7	38.5	29.4	40.9	43.3	40.7	38.7	29.0	35.6	22.8	36.6	27.5	33.8
FCN with GBP (ours)	76.8	40.0	**28.1**	38.6	24.6	39.7	37.3	50.2	51.4	**23.9**	47.2	**25.8**	53.6	49.1	53.9	**45.1**	**36.0**	48.1	**30.0**	35.8	33.9	41.4

Table 3.2: Results on PASCAL VOC 2012 test set. (Note that MIL-IPL-sppxl/bb/seg* used additional large training sets.)

Methods	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mean
MIL-FCN [45]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	24.9
EM-Adapt [4]	76.3	37.1	21.9	41.6	26.1	38.5	50.8	44.9	48.9	**16.7**	40.8	29.4	47.1	45.8	54.8	28.2	30.0	44.0	**29.2**	**34.3**	46.0	39.6
CCNN [5]	-	21.3	17.7	22.8	17.9	38.3	51.3	43.9	51.4	15.6	38.4	17.4	46.5	38.6	53.3	40.6	**34.3**	36.8	20.1	32.9	38.0	35.5
MIL-ILP-sppxl* [35]	74.7	38.8	19.8	27.5	21.7	32.8	40.0	50.1	47.1	7.2	44.8	15.8	49.4	47.3	36.6	36.4	24.3	44.5	21.0	31.5	41.3	35.8
MIL-ILP-bb* [35]	76.2	42.8	20.9	29.6	25.9	38.5	40.6	51.7	49.0	9.1	43.5	16.2	50.1	46.0	35.8	38.0	22.1	44.5	22.4	30.8	43.0	37.0
MIL-ILP-seg* [35]	**78.7**	48.0	21.2	31.1	**28.4**	35.1	51.4	**55.5**	52.8	7.8	**56.2**	19.9	**53.8**	50.3	40.0	38.6	27.8	51.8	24.7	33.3	**46.3**	40.6
ZOF with GBP (ours)	71.1	**48.4**	24.4	**48.5**	15.2	38.2	**65.6**	32.8	**57.9**	5.1	43.8	18.2	46.2	48.7	50.4	35.7	22.5	41.7	19.1	29.2	27.1	37.7
FCN with MCG (ours)	71.9	21.8	18.4	25.4	14.9	35.2	40.0	39.7	41.5	13.4	36.4	**29.9**	36.5	45.4	41.3	38.7	26.9	34.5	19.7	29.8	33.3	33.1
FCN with GBP (ours)	78.0	35.8	**28.5**	45.7	25.9	**43.1**	40.1	46.9	49.1	16.3	42.4	29.6	50.8	**51.3**	**57.2**	**44.4**	28.9	**44.8**	27.5	31.6	36.2	**40.7**

3.7 Summary

In this chapter, we proposed a novel weakly-supervised segmentation method based on feed-forward activations and BP-based object saliency maps [2] . In the proposed method, we showed that denser and clearer saliency maps can be obtained by up-sampling saliency maps of the intermediate layers and aggregating them. The proposed method showed better or comparable performance comparing the other state-of-the-art methods.

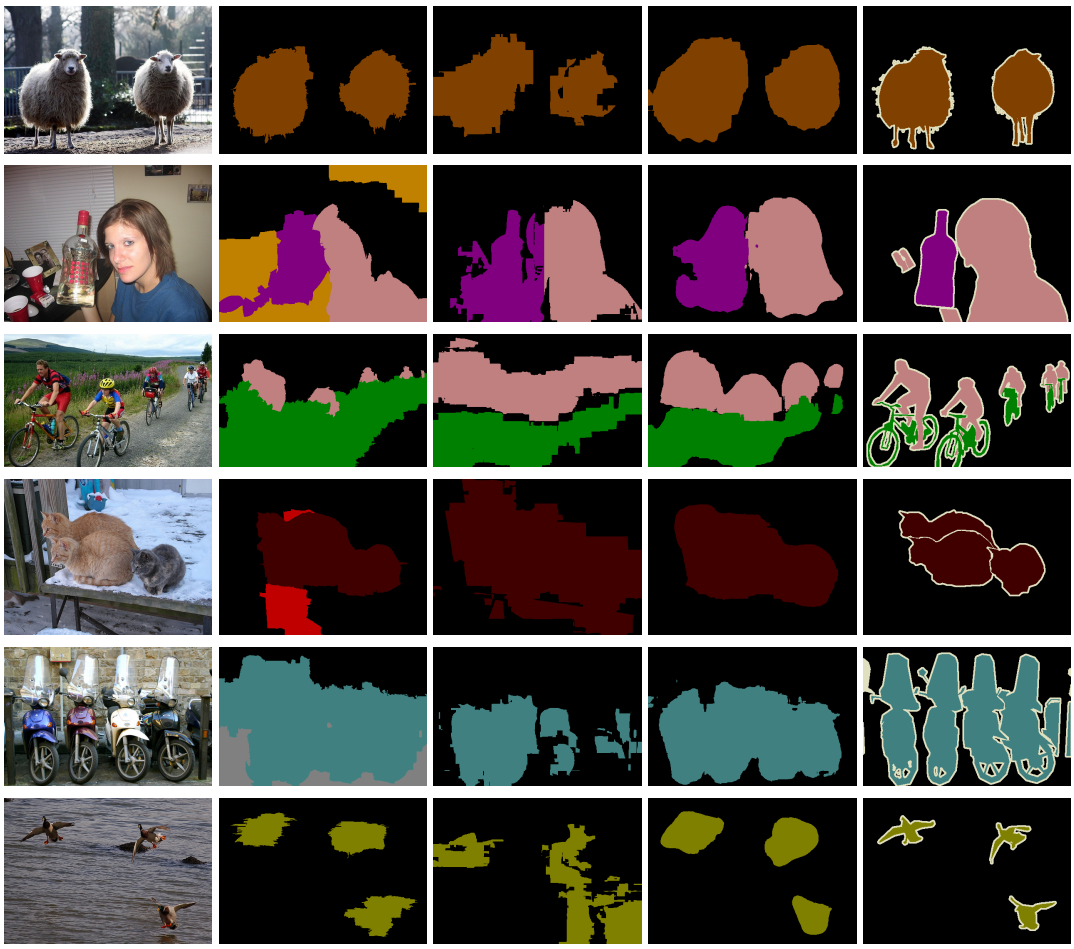

Figure 3.4: For each row, we show the input image, result of ZOF with GBP, and FCN with MCG, and FCN with GBP, and ground truth label.

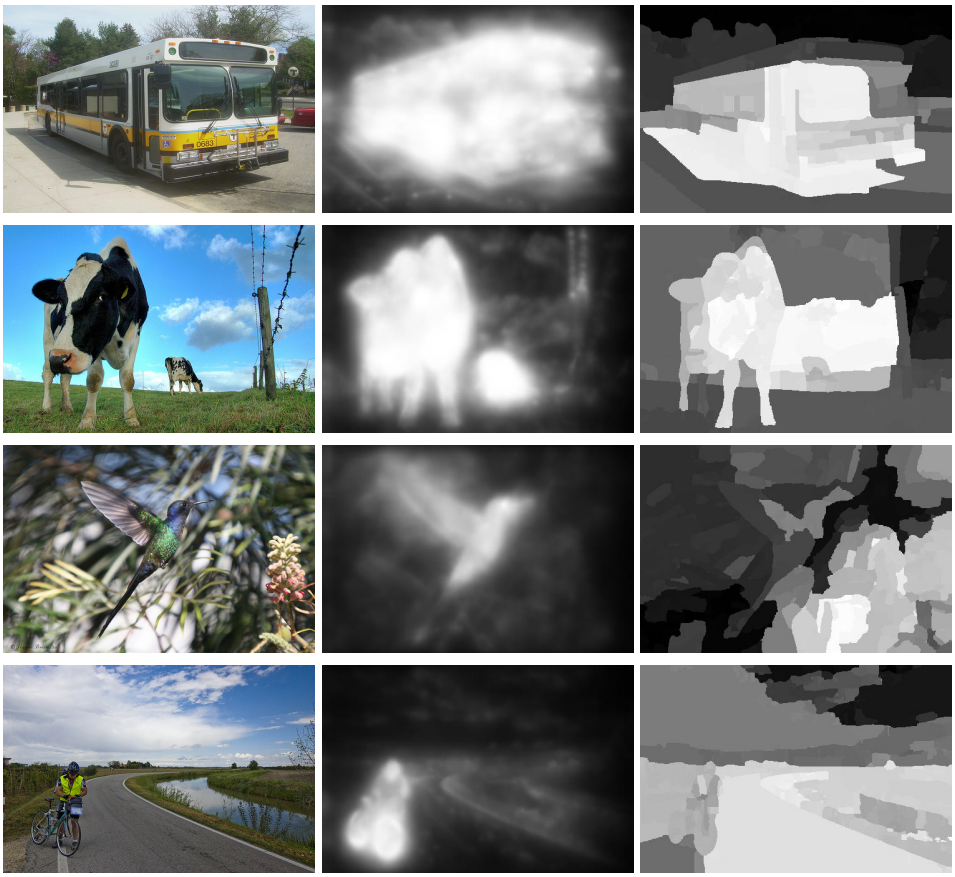

Figure 3.5: For each row, (left) image, (middle)saliency maps, (right)MCG priors.

Chapter 4

Visualization using only backward

In this chapter, we propose a CNN-based class saliency maps for weakly supervised semantic segmentation. We improved the CNN-based class-saliency maps method significantly and adapted them as unary potentials terms of fully-connected CRF ([38]).

Shimonyan et al. showed that class saliency maps could be obtained from the gradient of the class score which was calculated by back-propagation [2]. However, their class saliency maps are vague and not distinct (Figure 4.1(B)(C)). Furthermore, the saliency maps have another problem that it tends to respond to all foreground objects though the saliency maps is obtained by back-propagation for a class signal. Although Simonyan et al. citesim14 adopted GrabCut to convert segments from the class saliency maps in their paper, their method is unable to distinguish multiple object regions (Figure 4.1(D)(E)). We alleviate the problem of their method by some improvements and show the example of the improved result in (Figure 4.1(F)(G)). The examples of the results show that our saliency maps are more distinct and discriminative than the original saliency maps. The generated maps by the proposed method can be used as unary potentials of CRF as they are (Figure 4.1(H)). We call our new method for generating class saliency maps as "Distinct Class Saliency Maps (DCSM)".

To obtain more distinct class saliency maps, we propose three improvements over [2] (1) integrating derivatives with respect to the intermediate layers with up-sampling instead of the input image-level derivative; (2) subtracting the saliency maps of the other classes from the saliency maps of the

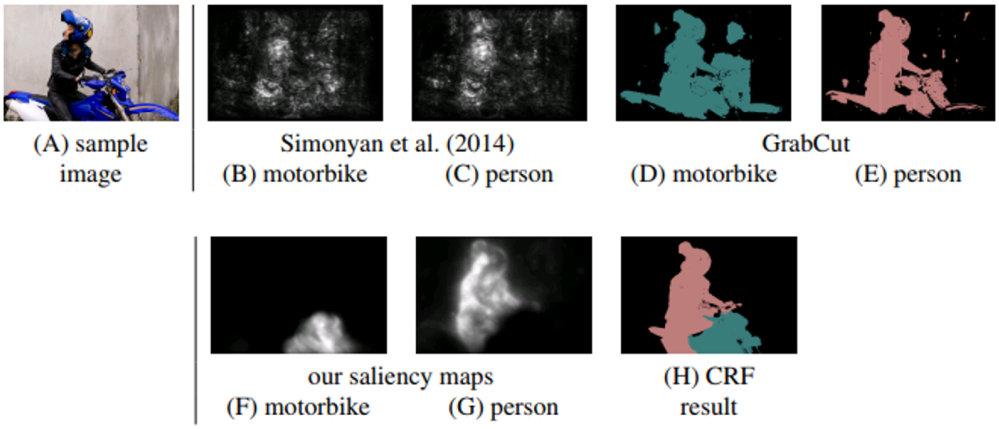

Figure 4.1: (From the left) (A)sample image, (B)(C)its class saliency maps with respect to "motorbike" and "person" by [2], (D)(E)estimated regions of them by GrabCut, (F)(G)class saliency maps by the proposed method, and (H) estimated regions by Dense CRF.

target class to differentiate target objects from other objects; (3) aggregating multiple-scale class saliency maps to combine higher resolution of the maps with the lower resolution of the maps. Finally, to convert the saliency maps to the segments, we apply fully-connected CRF ([38]) by using the distinct class saliency maps as unary potentials. In this chapter, we show that the proposed method has achieved comparable results on the PASCAL VOC 2012 dataset in the task of weakly-supervised semantic segmentation under the standard condition. works.

To summarize our contributions in this chapter, they are as follows:

- We propose a novel method to estimate distinct class saliency maps:
  - based on CNN derivatives with respect to feature maps of the intermediate convolutional layers.
  - subtracting class saliency maps from each other.
  - aggregating multiple-scale class saliency maps.

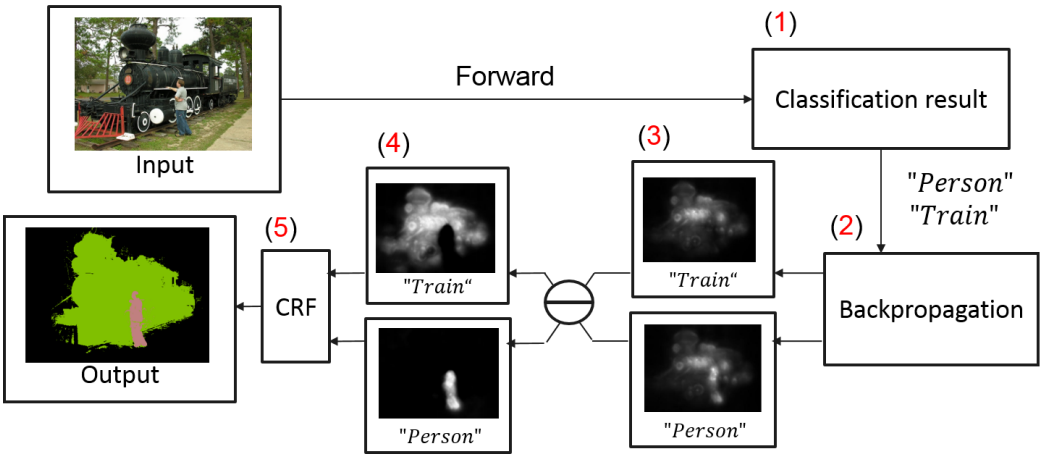

Figure 4.2: Processing flow of the proposed method: (1) multi-label classification (2) computation of back-propagation with respect to each of the detected class labels (3) generating raw class saliency maps (4) subtracting raw saliency maps of the other candidate classes from the saliency maps of the target class (5) applying Dense CRF with subtracted class saliency maps as unary potential

4.1 Distinct Class-specific Saliency Maps

In this section, we overview the proposed method and explain the detail of the method which consists of three elements: multi-label training of CNN, multi-class object saliency map estimation which was inspired by [2]. To achieve semantic segmentation for a given image, we (1) perform multi-label classification on a given image by feed-forward computation of the CNN, (2) calculate CNN derivatives with respect to feature maps of the intermediate convolutional layers with back-propagation by using each of the detected class labels as supervised signals in the loss function, (3) aggregate CNN derivatives of several intermediate layers with up-sampling to generate raw class saliency maps, (4) subtract raw saliency maps of the other candidate classes from the saliency maps of the target class, and (5) apply fully-connected CRF (Dense CRF) ([38]) with subtracted class saliency maps as unary potential. Finally, we obtain a segmentation result. The processing flow is shown in Figure 4.2.

4.1.1 Training CNN

For preparation, we train a CNN with a multi-label loss function. We use the VGG-16 ([59]) as a base CNN network pre-trained with ILSVRC datasets. In our work, we trained a CNN with image-level multi-label annotation.

Recently, fully convolutional networks (FCN), which accepts an image of arbitrary size are used widely. In this chapter, we also introduce FCN to enable the multi-scale generation of class saliency maps. We insert a global max pooling layer after the last convolutional layer for converting probability maps to probability vectors. We use images which are normalized to 500×500 by rescaling to have the largest size of the 500 pixels and zero-padding for training and testing in the same way as [67]. For multi-scale training, we resize training images randomly between the ratio 0.7 and 1.4 within a mini-batch.

To train the CNN with multi-label, we adopt a sigmoid cross entropy loss which is a standard loss function for multi-label annotation instead of a soft-max cross entropy loss, this approach is the same way as [67] and [45]. The Sigmoid cross entropy loss function is represented in the following equation:

$$\text{loss} = \sum_{n=1}^{K} [-p_n \log \hat{p_n} - (1 - p_n) \log(1 - \hat{p_n})] \tag{4.1}$$

where K is the number of classes, $p_n = \{0, 1\}$ which represents the existence of the corresponding class label, and $\hat{p_n}$ means the output of Sigmoid function of the class score $f_n(x)$ represented in the following equation:

$$\hat{p_n} = \frac{1}{1 + e^{-f_n(x)}} \tag{4.2}$$

4.1.2 Class Saliency Maps

We propose a new method to estimate class-specific saliency maps by enhancing the method proposed by [2] greatly. It consists of (1) extracting CNN derivatives with respect to feature maps of the intermediate convolutional layers, (2) subtracting class saliency maps between the target class and the other classes, and (3) aggregation of multi-scale saliency maps.

4.1.2.1 Extracting CNN derivatives

[2] regarded the derivatives of the class score with respect to the input image as class saliency maps. However, the class score output on the deep CNN is

the furthermost from the position of an input image, which sometimes causes weakening or vanishing of gradients. We use the derivatives with respect to feature maps of the relatively upper intermediate layers which are expected to retain more high-level semantic information instead of the derivatives of the class score with respect to the input image. We pick the maximum absolute values of the derivatives across all the kernels, and up-sample them by bilinear interpolation in order to adjust size of feature maps (Figure 4.3 (C)-(G)). Finally, we take mean of them to obtain one saliency map (Figure 4.3 (B)). The idea of aggregating of information extracted from multiple feature layers was inspired by the work of [20], although they extracted not CNN derivatives but feature maps calculated by feed-forwarding.

The class score derivative v_i^c of a feature map in the i-th layer is the derivative of class score Sc with respect to the layer L_i at the point (activation signal) L_i^0:

$$v_i^c = \left.\frac{\partial S_c}{\partial L_i}\right|_{L_i^0} \tag{4.3}$$

v_i^c can be computed by back-propagation. After obtained v_i^c, we up-sample it to w_i^c with bilinear interpolation so that the size of a 2-D map of v_i^c becomes the same as an input image. Next, the class saliency map $M_i^c \in \mathcal{R}^{m \times n}$ is computed as $M_{i,x,y}^c = \max_{k_i} |w_{i,h_i(x,y,k)}^c|$, where $h_i(x, y, k)$ is the index of the element of w_i^c. Note that each value of the saliency map is normalized by $\tanh(\alpha M_{i,x,y} / \max_{x,y} M_{i,x,y})$ for visualization in Figure 4.3 and all the other figures with $\alpha = 3$.

4.1.2.2 Subtracting raw class saliency maps

As shown in Figure 4.1(B)(C), the saliency maps of two or more different classes tend to be similar to each other especially in the image-level. The saliency map proposed by [2] sometimes matches foreground regions rather than a target class object. We relaxed this problem in the proposed methods by using saliency maps obtained from intermediate layers. However, the saliency regions of different classes are still overlapped with each other (Figure 4.4 (raw)). To resolve this problem, we subtract saliency maps of the other candidate classes from the saliency maps of the target class to differentiate target objects from other objects. Here, we assume that we use the CNN trained with multi-label loss, and select several candidate classes the class score of which exceed a pre-defined threshold with a pre-defined

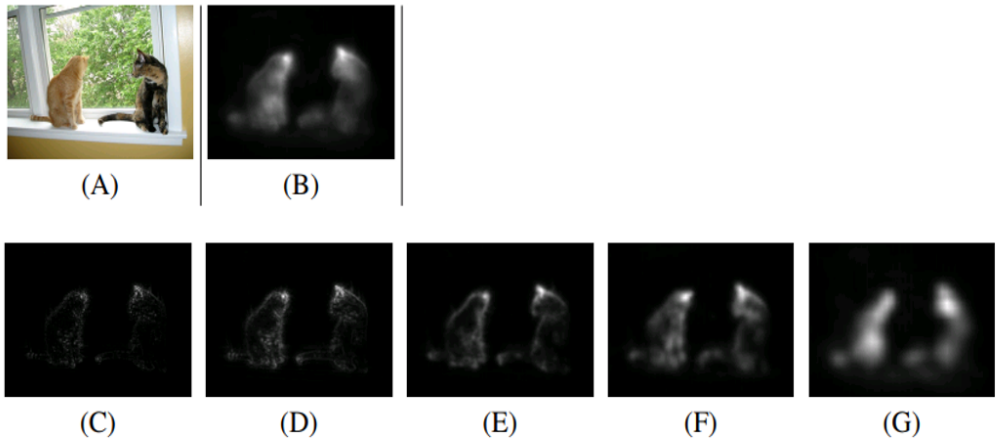

Figure 4.3: Class saliency maps obtained from the VGG16-net fine-tuned with the PASCAL VOC 2012 dataset. (A) an input image, (B) average of [(E)(F)(G)], (C) conv1_1, (D) conv2_1, (E) conv3_2, (F) conv4_2, (G) conv5_2

minimum number.

The improved class saliency maps with respect to class c, $\tilde{M}_i^c$, are represented as:

$$\tilde{M}_{i,x,y}^c = \sum_{c' \in candidates} \max\left(M_{i,x,y}^c - M_{i,x,y}^{c'}, 0\right)[c \neq c'], \tag{4.4}$$

where *candidates* is a set of the selected candidate classes. Figure 4.4 shows results without subtraction in the left (raw) and ones with subtraction in the right (diff). As we can see, subtraction of saliency maps resolved overlapped regions among the maps of the different classes.

4.1.2.3 Aggregating multi-scale class saliency maps

We use fully convolutional networks (FCN) which accept an image of arbitrary size for generating multi-scale class saliency maps. FCN outputs will be class score maps if the input image size is larger than original input image size. We denote the class score maps as $h \times w \times C$, where C is the number of classes, and h and w are larger than 1. We simply back-propagate the target class score map in order to obtain CNN derivatives with respect to enlarged feature maps, which is define as $S_c(:,:,c) = 1$ (in the MATLAB notation) with 0 for all the other elements, where c is the target class index.

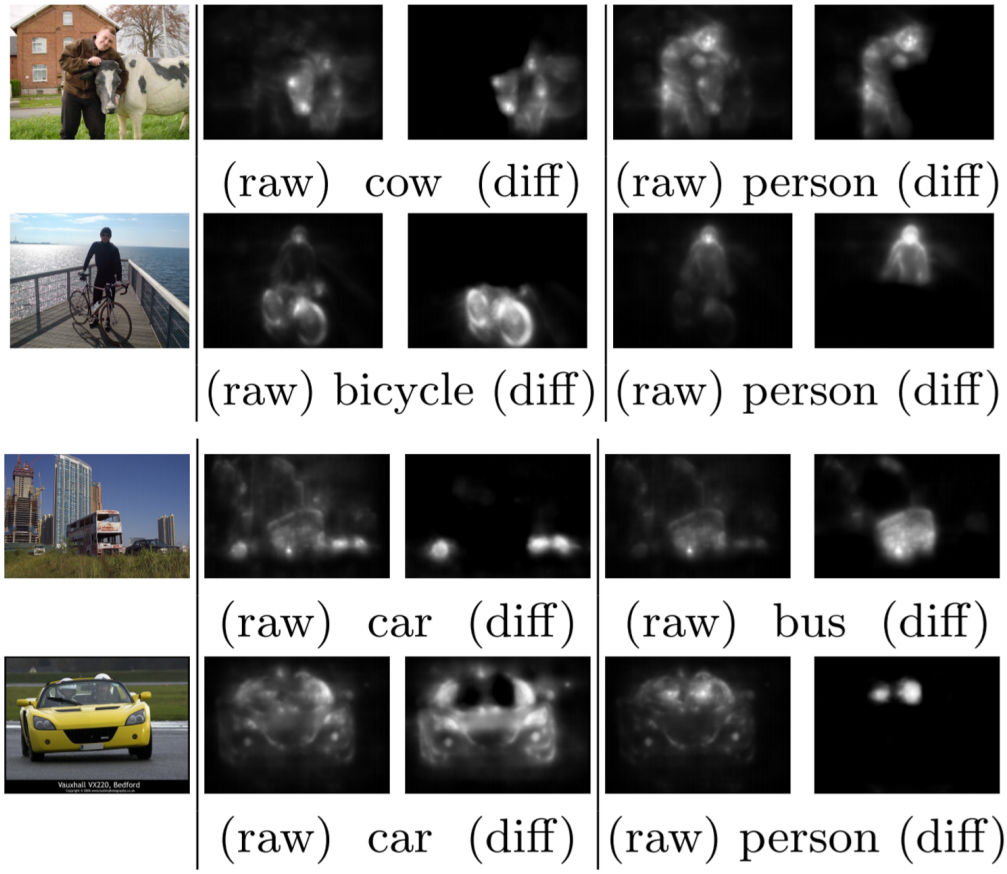

Figure 4.4: (raw) raw maps without subtraction (diff) maps with subtraction of other class maps.

The final class saliency map $\hat{M}^c$ averaged over the layers and the scales is obtained as follows:

$$\hat{M}^c_{x,y} = \frac{1}{|S||L|} \sum_{j \in S} \sum_{i \in L} \tanh(\alpha \tilde{M}^c_{j,i,x,y}), \tag{4.5}$$

where L is a set of the layers for which saliency maps are extracted, S is a set of the scale ratios, and α is a constant which we set to 3 in the experiments. Note that the size of $\tilde{M}_{j,i}$ for all the layers are normalized to the same size as an input image before taking average.

In the experiments, we also used guided back-propagation (GBP) proposed by [3] as the back-propagation method and compared with proposed

before by [2]. The difference between the two methods is computation through ReLU in only backward. GBP can reduce noise components from the derivatives than normal BP by limiting only positive values of CNN derivatives as shown in Figure 4.5. Figure 4.5 shows saliency map computed by each two methods.

$$\text{BP} : \frac{dz^i}{dx^i} = \frac{dz^{i+1}}{dx^{i+1}} \cdot \left(f^{i+1} > 0\right) \tag{4.6}$$

$$\text{GBP} : \frac{dz^i}{dx^i} = \frac{dz^{i+1}}{dx^{i+1}} \cdot \left(\frac{dz^{i+1}}{dx^{i+1}} > 0\right) \cdot \left(f^{i+1} > 0\right) \tag{4.7}$$

where f^i represents an activation at the i-th layer, and $(x > 0)$ means 1 if x is positive, or 0 if not.

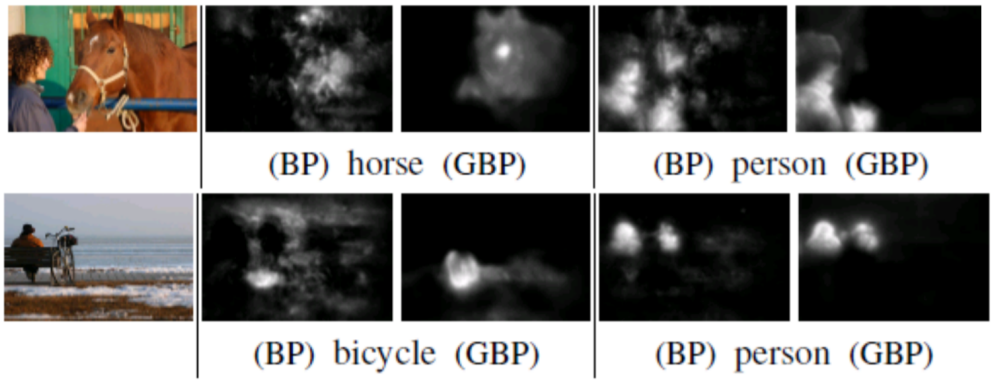

Figure 4.5: Obtained class saliency maps (Left) using BP (Right) using GBP.

4.1.3 Fully Connected CRF

Conditional Random Field (CRF) is a kind of probabilistic graphical model which considers both node priors and consistency between nodes. We apply CRF to estimate object boundaries, so that proposed class-specific saliency maps represent only probability of the target classes on each pixel and have no clear information on object boundaries, In this chapter, we use fully connected CRF [38] where every pixel is regarded as a node, and every node is connected to every other node. The energy function is defined as follows:

$$\text{E}(\mathbf{c}) = \sum_i \theta_i(c_i) + \sum_{i,j} \theta_{i,j}(c_i, c_j) \tag{4.8}$$

where c_i represents a class assignment on pixel i. The first unary term of the above equation is calculated from class saliency maps $\hat{M}_i^c$. We defined it as $\theta_i(c_i) = -\log(\hat{M}_{x,y}^c)$.

To obtain the background class maps directly by proposed method is difficult because background exists in most of images. To adapt CRF for image segmentation, a unary potential on the background class is needed as well as foreground potential. We define a unary potential on the background class from the maps of the candidate classes selected in the previous step by the following equation.

$$\hat{M}_{x,y}^{BG} = 1 - \max_{c \in target} \hat{M}_{x,y}^c \tag{4.9}$$

where $\hat{M}_{x,y}^{BG}$ is a saliency map of background class, and *target* represents a set of the selected candidate classes.

The pairwise term of Eqation 4.8 is represented by $\theta_{i,j}(c_i, c_j) = u(c_i, c_j)k(f_i, f_j)$ where and $k(\mathbf{f}_i, \mathbf{f}_j)$ is a Gaussian kernel. Note that $\mathbf{f}_i, \mathbf{f}_j$ represents some kinds of image features extracted from pixel i and j. Following [38], we adopt bilateral position and color terms, and the kernels are

$$k(\mathbf{f}_i, \mathbf{f}_j) = w_1 \exp\left(-\frac{|p_i - p_j|^2}{2\gamma_{\alpha^2}} - \frac{|I_i - I_j|^2}{2\gamma_{\beta^2}}\right) + w_2 \exp\left(-\frac{|p_i - p_j|^2}{2\gamma_{\gamma^2}}\right) \tag{4.10}$$

where the first kernel depends on both pixel positions (denoted as p) and pixel color intensities (denoted as I), and the second kernel only depends on pixel positions. The hyper parameters γ_α, γ_β, and γ_γ control the scale of the Gaussian kernels. This model is amenable to efficient approximate probabilistic inference proposed by [38].

4.2 Experiments

We evaluated the proposed methods using the PASCAL VOC 2012 data. We show that our methods have comparable ability with other state-of-the-arts segmentation methods on weakly-supervised setting. The PASCAL VOC 2012 segmentation dataset has 1464 training images, 1449 validation images, and 1456 test images including 20 class pixel-level labels as well as image-level labels. On training, we used the augmented PASCAL VOC training data provided by *train_aug* which training the image number is 10582. This training data is commonly used in the same way on the other works on

weakly-supervised segmentation such as MIL-FCN ([45]), EM-Adapt ([4]) and CCNN ([5]). For evaluation, we used a standard intersection over union (IoU) metric which is the official evaluation metric in the PASCAL VOC segmentation task.

4.2.1 Experimental Setup

We modified VGG-16 model ([59]) widely used in other weakly supervised semantic segmentation methods. We adopted Sigmoid cross entropy loss in order to train with multi-label annotation, resized input image randomly, converted an output of score map to a vector by global max pooling for multi-scale training and fine-tuned it with PASCAL VOC *train_aug* dataset. This training process follows the paper of [67]. As a training framework for CNN, we used Caffe ([68]). To train the CNN we used small batchsize 2 because of memory limitation which is caused by large training image size. We set learning rate 1e-5, momentum 0.9 and weight decay 0.0005. For the first 30000 iterations, we fine-tuned only the upper layers of the modified VGG-16 than Pool_5, and for the next 20000 iterations, we fine-tuned all the layers in order to avoid divergence. We trained the network by the process using a NVIDIA GeForce Titan-X GPU. On test phase we also used same GPU, it takes about 0.3 seconds to perform segmentation for a single image.

4.2.2 Evaluation on Class Saliency Maps

First, we compare the class saliency maps estimated by the proposed method, DCSM, with ones by Simonyan et al. [2] qualitatively. Figure 4.6 shows both the results by Simonyan et al. and our method for three multiple object images and one single object images. These experiments show that proposed method is much more effective for not only multiple object images but also single object images than the approach proposed by [2]. This figure shows our results are better than [2] greatly, because we aggregate gradients in the multiple intermediate layers and carry out subtraction of raw class saliency maps. Our results clearly discriminated multiple regions of the different classes.

Figure 4.7 shows the results for images containing three or more objects. In even such cases, all the class saliency maps except for "chair" in the top-right sample were estimated successfully.

Table 4.1: Results of the mean IoU by Simonyan et al. and ours on Pascal VOC 2012 *val set*

method \ α	2	2.5	3	4	5	6	7	8	9	10	15
Simonyan et al.	-	-	10.0	20.6	28.3	32.7	33.4	33.8	33.8	33.3	28.7
DCSM (ours)	40.0	44.0	44.1	40.6	36.4	-	-	-	-	-	

To compare the proposed method with the previous method proposed by [2] quantitatively, we carried out weakly-supervised segmentation by adapting fully-connected CRF to estimated class saliency maps. We obtained the class maps by Equation 9.4 which contains a hyper-parameter, α. As shown in Table 4.1, we searched for the best values of α for both of Simonyan et al. and proposed method. As results, our method achieved 44.1% as the best mean IoU with $\alpha = 3$, while Simonyan et al. achieved 33.8% with $\alpha = 8$ (or 9). This result proves that the proposed saliency maps have higher ability than the maps obtained by the method [2] as unary potentials of CRF for semantic segmentation.

4.2.3 Effects of Parameter Choices

Intermediate layers In the proposed method, we extracted CNN derivatives from intermediate layers of the VGG-16, and averaged them to estimate class saliency maps. We examined the effects on which layers we use to extract derivatives from. Table 4.2 shows the results evaluated with VOC *val set* varying the layer combinations. "Block1" in the Table means the average of conv1_1 and conv1_2 in VGG-16, and "average Block 3,4,5" means the average of Block 3, Block 4, and Block 5. Among the single blocks, Block 4 achieved the best result, and among the block combinations, the combination of Block 3,4,5 achieved the best. Although Block 5 itself was less effective, adding Block 5 to combinations was effective to boost performance. This shows that aggregation of CNN derivatives extracted from multiple upper layers is the better choice.

Size of input images We examined the effects on input image size and multi-scale combination of input images since we use fully convolutional CNN which can deal with arbitrary-sized input images. For up-scaling we used bilinear interpolation. Table 4.3 shows the results, which indicates 500×500 was the best, and the combination of 400×400, 500×500 and 600×600

was the best. This is partly because we used training images with random resizing from 350 to 700 pixels. From these results, multi-scale aggregation helped boost performance.

Minimum number of the raw class maps for subtraction We prepared the raw class saliency maps of the top-N classes which were predicted by multi-label classification to use for subtraction. Note that we limit the class for subtraction in which classification score is more than the pre-defined threshold, 0.5. Actually, the threshold is 0 before adapting sigmoid function the value of which is used on training multi-label prediction. We examined the changes on the differing N. We showed also the results on the case of $N = 0$ which meant that subtraction was never carried out, that is, the results without subtraction. Table 4.5 shows that using the top-4 ($N = 4$) raw class maps were the best[1]. The subtraction is always helpful to raise segmentation performance compared with the case of $N = 0$.

Guided BP vs. BP We compared normal backpropagation (BP) used in [2] with guided backpropagation (GBP) proposed [3]. Class maps obtained by GBP included less noise and The score was also better than normal BP as shown in Table 4.4.

4.2.4 Comparison with Other Methods

In the final subsubsection, we compare our results (DCSM) with other results by CNN-based methods quantitatively. Table 5.2 and Table 5.3 show the results for PASCAL VOC 2012 *val set* and *test set*, respectively.

While MIL-FCN ([45]), EM-Adapt ([4]), CCNN ([5]) and our methods used PASCAL VOC training data and augmented training data provided by [58], MIL-{sppxl,bb,seg} by [35] used their original additional training images which contains 700,000 images. Our method is different from other methods in terms of the way to use a CNN. While the existing methods employed only feed forward computation ([45, 35, 4, 5]), we use backward computation as well as feed forward computation. Although the way to train CNN is the same as MIL-FCN ([45]) and MIL-{sppxl,bb,seg} ([35]), the method to localize objects is essentially different. As shown in the tables, our results by DCSM with CRF outperformed the methods which not using iterative training approach.

In Table 5.3, we also compared DCSM with the fully supervised methods.

[1]We used $N = 3$ in all the other experiments to save computation.

Our result is close to the result by one of the best non-CNN-based fully supervised method, O2P ([69]). Their difference is only 2.5 points. We show qualitative results by the proposed method without/with Dense CRF in Figure 4.8.

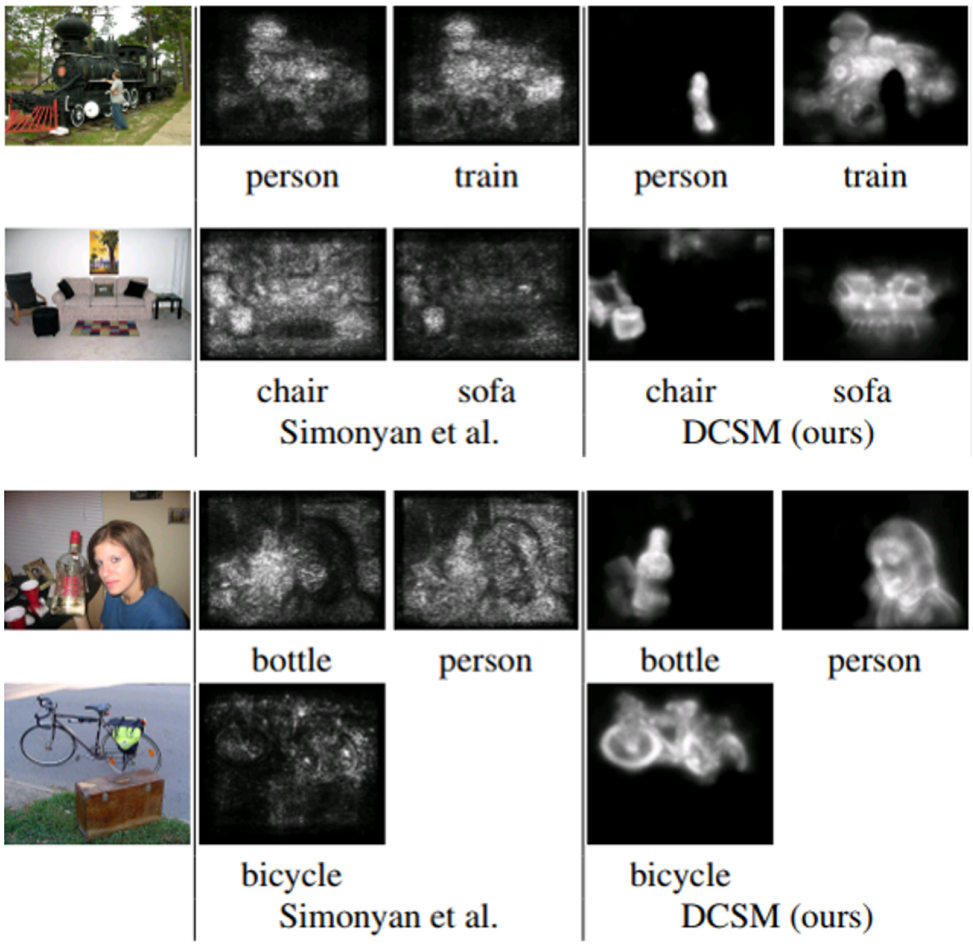

Figure 4.6: Obtained class saliency maps (Left) by [2] (Right) by the proposed method (DCSM).

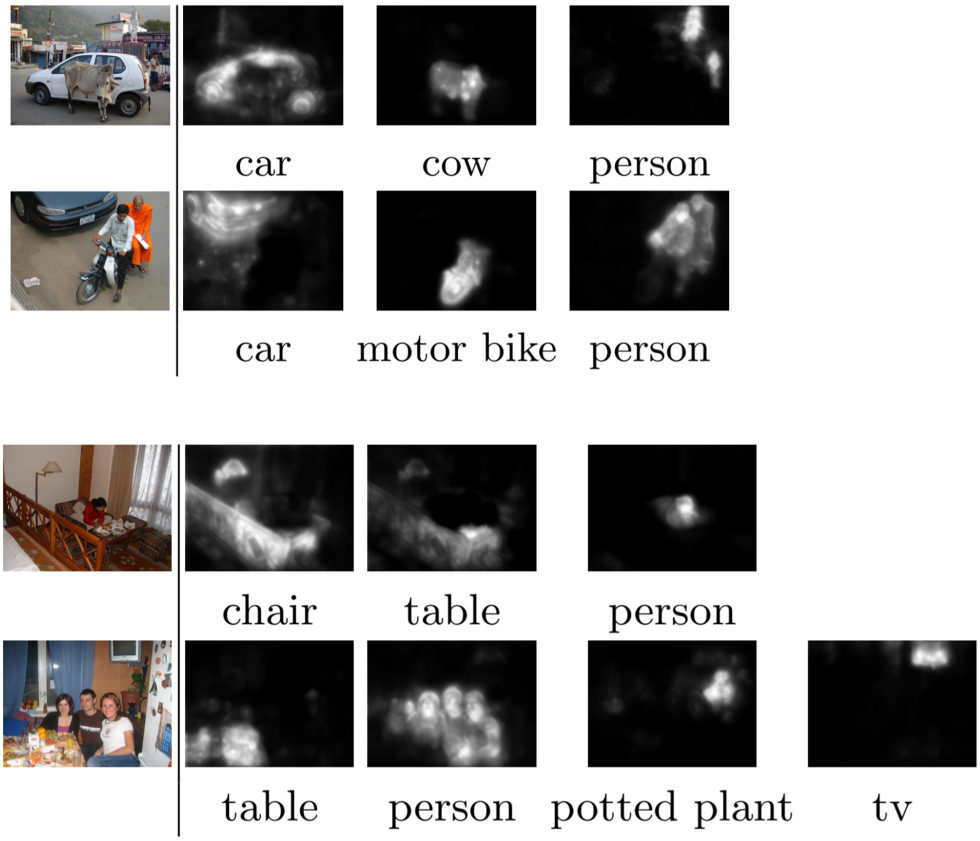

Figure 4.7: Obtained class saliency maps for images containing three or more classes.

Table 4.2: Effects by layers from which CNN derivatives are extracted.

layer	mean IoU
block1 (conv1_1, conv1_2)	5.5
block2 (conv2_1, conv2_2)	21.5
block3 (conv3_1, conv3_2, conv3_3)	32.5
block4 (conv4_1, conv4_2, conv4_3)	**40.3**
block5 (conv5_1, conv5_2, conv5_3)	26.3
average block 1,2,3,4,5	41.3
average block 2,3,4,5	42.2
average block 3,4,5	**42.8**
average block 4,5	42.5
average block 3,4	37.97

Table 4.3: Effects by input image size and multi-scale aggregation.

input image size	mean IoU
(1) 300 × 300	34.5
(2) 400 × 400	41.0
(3) **500 × 500**	**42.4**
(4) 600 × 600	41.8
(5) 700 × 700	40.0
(6) 800 × 800	34.5
average (1),(2),(3)	41.1
average (2),(3)	42.9
average (2),(3),(4)	**43.5**
average (3),(4)	42.9
average (3),(4),(5)	42.5
average (3),(4),(5),(6)	42.8

Table 4.4: Effects on the way of back-propagation.

method	BP	GBP
mean IoU	41.2	44.1

Table 4.5: Effects on the number of raw class maps for subtraction.

class N	0	1	2	3	4	5	10	15
mean IoU	38.2	42.2	43.5	44.1	44.2	44.0	43.7	43.3

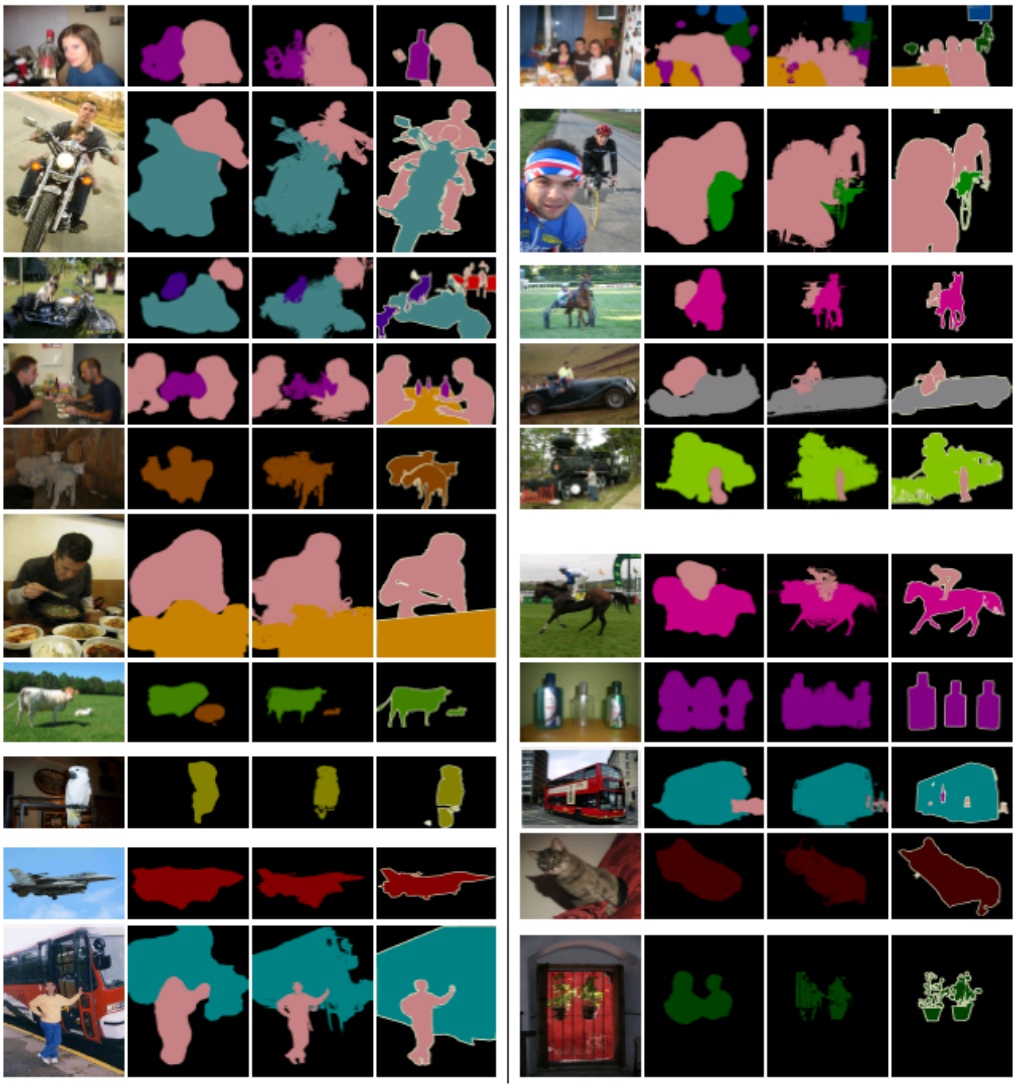

Figure 4.8: Qualitative results of DCSM on VOC 2012. Each row shows (left) input image, (middle left) results estimated from class maps, (middle right) results after applying FC-CRF, and (right) ground truth.

Table 4.6: Results on PASCAL VOC 2012 *val set*.

methods	images additional	approach iterative	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	**mIoU**
MIL-FCN [45]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	25.7
EM-Adapt [4]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	38.2
CCNN [5]	-	-	65.9	23.8	17.6	22.8	19.4	36.2	47.3	46.9	47.0	16.3	36.1	22.2	43.2	33.7	44.9	39.8	29.9	33.4	22.2	38.8	36.3	34.5
MIL-sppxl [35]	✓	-	77.2	37.3	18.4	25.4	28.2	31.9	41.6	48.1	50.7	12.7	45.7	14.6	50.9	44.1	39.2	37.9	28.3	44.0	19.6	37.6	35.0	36.6
MIL-bb [35]	✓	-	78.6	46.9	18.6	27.9	30.7	38.4	44.0	49.6	49.8	11.6	44.7	14.6	50.4	44.7	40.8	38.5	26.0	45.0	20.5	36.9	34.8	37.8
MIL-seg [35]	✓	-	79.6	50.2	21.6	40.6	34.9	40.5	45.9	51.5	60.6	12.6	51.2	11.6	56.8	52.9	44.8	42.7	31.2	55.4	21.5	38.8	36.9	42.0
F/B prior[40]	-	✓	79.2	60.1	20.4	50.7	41.2	46.3	62.6	49.2	62.3	13.3	49.7	38.1	58.4	49.0	57.0	48.2	27.8	55.1	29.6	54.6	26.6	46.6
STC[7]	✓	✓	84.5	68.0	19.5	60.5	42.5	44.8	68.4	64.0	64.8	14.5	52.0	22.8	58.0	55.3	57.8	60.5	40.6	56.7	23.0	57.1	31.2	49.8
SEC [6]	-	✓	82.4	62.9	26.4	61.6	27.6	38.1	66.6	62.7	75.2	22.1	53.5	28.3	65.8	57.8	62.3	52.5	32.5	62.6	32.1	45.4	45.3	50.7
DCSM (ours)	-	-	76.7	45.1	24.6	40.8	23.0	34.8	61.0	51.9	52.4	15.5	45.9	32.7	54.9	48.6	57.4	51.8	38.2	55.4	32.2	42.6	39.6	44.1

Table 4.7: Results on PASCAL VOC 2012 *test set*.

methods	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	**mIoU**
Fully Supervised:																						
O2P [69]	85.4	69.7	22.3	45.2	44.4	49.6	66.7	57.8	56.2	13.5	46.1	32.3	41.2	59.1	55.3	51.0	36.2	50.4	27.8	46.9	44.6	47.6
SDS [18]	86.3	63.3	25.7	63.0	39.8	59.2	70.9	61.4	54.9	16.8	45.0	48.2	50.5	51.0	57.7	63.3	31.8	58.7	31.2	55.7	48.5	51.6
FCN-8s [20]	-	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
Deeplab Large FOV [23]	92.6	83.5	36.6	82.5	62.3	66.5	85.4	78.5	83.7	30.4	72.9	60.4	78.5	75.5	82.1	79.7	58.2	82.0	48.8	73.7	63.3	70.3
Using Additional Supervision:																						
CCNN w/size [5]	-	42.3	24.5	56.0	30.6	39.0	58.8	52.7	54.8	14.6	48.4	34.2	52.7	46.9	61.1	44.8	37.4	48.8	30.6	47.7	41.7	45.1
One point[48]	80.6	50.2	23.9	38.4	33.1	38.5	52.0	50.9	55.4	18.3	38.2	37.7	51.0	46.1	54.7	43.2	35.4	45.1	33.0	49.6	40.0	43.6
F/B prior + CheckMask[40]	87.4	65.7	26.0	64.2	43.7	53.2	72.6	63.6	59.5	17.1	48.0	43.7	61.2	52.0	69.3	54.8	43.0	50.3	34.6	59.2	42.0	52.9
weakly-supervised:																						
MIL-FCN [45]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	24.9
EM-Adapt [4]	76.3	37.1	21.9	41.6	26.1	38.5	50.8	44.9	48.9	16.7	40.8	29.4	47.1	45.8	54.8	28.2	30.0	44.0	29.2	34.3	46.0	39.6
CCNN [5]	-	21.3	17.7	22.8	17.9	38.3	51.3	43.9	51.4	15.6	38.4	17.4	46.5	38.6	53.3	40.6	34.3	36.8	20.1	32.9	38.0	35.5
MIL-ILP-seg [35]	78.7	48.0	21.2	31.1	28.4	35.1	51.4	55.5	52.8	7.8	56.2	19.9	53.8	50.3	40.0	38.6	27.8	51.8	24.7	33.3	46.3	40.6
F/B prior[40]	80.3	57.5	24.1	66.9	31.7	43.0	67.5	48.6	56.7	12.6	50.9	42.6	59.4	52.9	65.0	44.8	41.3	51.1	33.7	44.4	33.2	48.0
STC [7]	85.2	62.7	21.1	58.0	31.4	55.0	68.8	63.9	63.7	14.2	57.6	28.3	63.0	59.8	67.6	61.7	42.9	61.0	23.2	52.4	33.1	51.2
SEC [6]	83.5	56.4	28.5	64.1	23.6	46.5	70.6	58.5	71.3	23.2	54.0	28.0	68.1	62.1	70.0	55.0	38.4	58.0	39.9	38.4	48.3	51.7
weakly-supervised:																						
DCSM (ours)	78.1	43.8	26.3	49.8	19.5	40.3	61.6	53.9	52.7	13.7	47.3	34.8	50.3	48.9	69.0	49.7	38.4	57.1	34.0	38.0	40.0	45.1

Table 5.2 and Table 5.3 show the comparison with the other weakly-supervised segmentation methods. Our method showed the comparable performance with the other state-of-the-art methods on the same condition using only image-level-label as training data. Especially our proposed method achieved the better result than F/B prior ([40]), STC ([7]), SEC ([6]) all of which employed (re)-trained DeepLab with the estimated initial masks on the weakly-supervised setting. Our approach also outperformed SDS which is based on the fully supervised method proposed by [18].

4.3 Summary

In this chapter, we proposed a new weakly-supervised semantic segmentation method consisting of a novel method of class saliency map estimation and Dense CRF. The proposed distinct class saliency maps (DCSM) outperformed the maps by Simonyan et al. citesim14 both qualitatively and quantitatively. The experimental results proved the effectiveness of the proposed method, which achieved the state-of-the-arts on the PASCAL VOC 2012 weakly-supervised segmentation.

Chapter 5

Noise data estimation and rejection by estimation of segmentation "Easiness"

In this chapter, we propose a novel algorithm to estimate "Easiness" of training data for weakly-supervised segmentation results. By this method we calculate scores for each training data and we retrieve "good seeds" based on the score. For the estimation of "Easiness", we consider "consistency" among the results with different conditions. To do that, we use two kinds of weakly-supervised segmentation method, a BP-based mask estimation method proposed by Simonyan et al [2] and an improved method proposed in the previous Chapter 4. By evaluating the consistency between the results by the two methods, we estimate segmentation "Easiness" of each of the training image, and select the easier ones as "seed images" and regards their estimated masks as "seed masks". In addition, we demonstrated that it is also effective to use retrieved images using the score of "Easiness" for data augmentation.

To summarize our contributions in this chapter, they are as follows:

- We propose a novel algorithm to estimate "Easiness" by consistency among results of different conditions.
- We show that the retrieved images by our proposed method is effective for data augmentation.

5.1 Method

In this chapter, we basically adopt an iterative approach of mask estimation and training of a fully-supervised semantic segmentation model in the similar way to [7, 6] for the weakly-supervised semantic segmentation tasks. To estimate initial masks which need training of a fully-supervised model in the weakly-supervised task, we use Distinct Class-specific Saliency Maps (DCSM) 4. In this chapter, in order to obtain better initial masks, we pay attention to " consistency" among the results of different processing for selecting "good seeds".

5.2 Estimation of "Easiness" of Training Images

In this chapter, we also propose a novel algorithm to estimate the accuracy of segmentation to use the results of DCSM as training data for the fully supervised segmentation model effectively. The proposed method can predict "Easiness" of segmentation and retrieve "good seeds". To estimate initial masks, we use DCSM. In order to obtain better initial masks, we pay attention on "consistency" among the results of different processing for selecting "good seeds". Especially, we paid attention to the following two points:

1. Correlation on "Easiness" between classification and segmentation.
2. Coherence on the segmentation results between one obtained by a sophisticated method and one obtained by a simpler method.

From the two assumptions, we select easier images from the training dataset, and give priority to them in the initial training phase.

5.2.1 Difference between DCSM and DCSM without subtraction

It is easy to imagine that the difficult images to be classified is hard to be segmented. However, the easy-classified images are not always easy for segmentation. Therefore, it is difficult to estimate "Easiness" on segmentation directly from classification results. Thus, in this chapter, we utilize the BP-based object-specific saliency map for estimation. While BP-based saliency

maps proposed by Simonyan et al. [2] are relatively vague and not distinct, DCSM generates more distinct class saliency maps that are discriminative for the regions of a target class from the regions of the other classes by taking subtraction for different class signals. Here, we consier the difference between the original DCSM and the DCSM without subtraction. If no difference appears in both results, the input images can be regarded as being simple images. On the other hand, if both results are largely different, the input images can be regarded as being complexed images.

For image x, let $V_o(x)$ be segmentation result without subtraction, $V_w(x)$ be segmentation result with subtraction. "Easiness" for subtraction $R_{sub}(x)$ is calculated as:

$$R_{sub}(x) = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \mathrm{IoU}(V_o^c(x), V_w^c(x)) \tag{5.1}$$

where IoU(.,.) is a function which returns the Intersection over Union (IoU) for two regions, and $\mathcal{C}$ is the set for the difference input image sizes.

5.2.2 Coherence on size change of input images

As an additional measurement on "Easiness", we consider consistency of the segmentation masks when varying the size of input images. On semantic segmentation task, the receptive field size of each pixel is important. While change of the receptive field size is related to segmentation accuracy, some images can be segmented accurately without the change. We consider that if we need the change of receptive field to obtain better results for segmentation, the difficulty of segmentation is high. In other words, if the same segmentation masks are obtained from various conditions in terms of image size, the given images can be regarded as an easy image to segment. In this chapter, we use this as the second measurement of "Easiness".

Actually, we used the three image sizes, $s_n = 320, 416, 512$ $(n = 0, 1, 2)$. We represent DCSM maps before adapting CRF as $M^{s_n}(x)$. We obtain aggregated maps, $M^b(x)$ with

$$M^b(x) = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} M^{s_c}(x)$$

$V^b(x)$ represents the CRF result of $M^b(x)$ after applying the dense CRF-based refinement. Then, we compute the coherence on size change, $R_{size}(x)$,

by the following equation:

$$R_{size}(x) = \frac{1}{2|\mathcal{C}|} \sum_{c \in \mathcal{C}} \mathrm{IoU}(V_o^b(x), V_o^c(x)) + \mathrm{IoU}(V_w^b(x), V_w^c(x)) \tag{5.2}$$

Finally we combine two kinds of the reliable scores in the following equation:

$$score = \lambda_1 \cdot R_{size} + \lambda_2 \cdot R_{sub} \tag{5.3}$$

where λ_1 and λ_2 are pre-defined constant values. In this chapter we simply set λ_1 and λ_2 to 0.5. Figure 5.1 shows the example of the results of the estimated mask in different conditions.

5.2.3 Generating of segmentation mask

We generate segmentation masks of the training images with only image-level annotation by using DCSM. The final results are obtained after applying dense CRF. On the contrary to the original DCSM, we used single-class classifier CNNs as well, and we generate the final mask by integrating single-class classifier results with the multi-class classifier results. In the case of PASCAL dataset, we train each single-class CNNs with soft-max cross entropy loss. Figure 5.1 shows the examples of the generated mask. Note that we used only corresponding regions for results of different conditions as the training data such as localization cue used in [6].

5.3 Experiment

5.3.1 Experimental setup

For the setup of classification model, we used the same setup of Section 4.2.1. To train fully-supervised segmentation model with the estimated masks, we used DeepLab-CRF [23]. To optimize the model we used SGD for 10000 iteration, the batch size is 16, momentum parameter is 0.9 and a weight decay is 0.0005. We set the learning rate to 0.001 except for the last layer where learning rate is 0.01. We decrease the learning rate by 0.1 for every 2000 iterations. Each model is trained with 7-8 hours by a NVIDIA GeForce Titan-X GPU with 12GB memory. In the experiments of this section are conducted using DeepLab code [23], which is implemented based on the publicly available Caffe framework [70].

5.3.2 Evaluation for the estimation of "Easiness"

Figure 5.3 shows top5 retrieval results of each class obtained by the proposed algorithm "Easiness". Though "Easiness" is not fully-supervised, i.e. it does not uses any pixel-wise annotations, "good seeds" are retrieved in most cases. For example, in case of aeroplane, car, cow and dog, retrieved seeds are close to the ground truth. On the other hand, the results of sofa, chair and table include some noise. In general segmentation results of these class show low performance, hence the retrieved results are directly affected by the low quality of prediction.

Data augmentation is well known as the way for avoiding overfitting and improving accuracy on test data. However, it is expected that augmented data including noise will be not effective for improving accuracy. Therefore, we augmented training data for only "good seeds" retrieved by the proposed algorithm. As a data augmentation method, we referred approach of Liu et al. [71], which is mentioned on their poster in the conference. In fully supervised detection, they changed training data dynamically by data augmentation. We followed their approach [71] and dynamically augmented training data by random cropping and random padding. For cropped images, we recognized the images by the multi-class classification model and used only the images whose results corresponding to the class label. For each training data and each augmentation process we augmented 10 images. Table 10.2 shows the results of combination of "Easiness" with data augmentation where the image number is defined by the threshold of equation (Equation 5.2). "Base image N" represents the number of the images which was used as the training data without data augmentation, while "aug image N" indicates the number of image used for data augmentation. As the results, setting (c) achieved the best accuracy 51.3%, this setting limits both the number of base training images and the number of augmented images. Our proposed method improved the simple approach certainly, considering the result of setting (f) score is 48.8% which was trained with all the training images and all the augmented images, which was the lowest score in all settings. The Table also shows that training data selection for base images is effective constantly. In the setting (a) and (b), in order to collect the training data which has further quality, we limited augmented number of image to 780, but we obtained the worse results. In training of deep CNNs, there is a trade-off between the number of training image and training data quality. However, these results indicate

that the accuracy can be boosted by data selection. Even though the training data size was small, quality of data is important and higher-quality data can be used for the data augmentation effectively.

Table 5.1: Combination of "Easiness" with data augmentation

setting	Base image N	Aug image N	mIoU
(a)	8760 (th $\geq$ 0.3)	730 (th $\geq$ 0.8)	50.1
(b)	10582 (all)	730 (th$\geq$0.8)	48.9
(c)	8760 (th $\geq$ 0.3)	2105 (th $\geq$ 0.7)	**51.3**
(d)	10582 (all)	2105 (th$\geq$0.7)	49.9
(e)	8760 (th $\geq$ 0.3)	8760 (th $\geq$ 0.3)	49.7
(f)	10582 (all)	10582 (all)	48.8

Figure 5.1: Examples of visualization on the different conditions. In the top case, we define this sample as "a good seed" so that almost predicted regions have consistency. In the middle case, the visualization results have a large difference in the simple visualization and visualization with subtraction, hence we estimate the result of this sample as "a bad seed" for the (re-)training. In the bottom case, visualization results have the corresponding region, but some results have inconsistency in the results of varying input size, thus our proposed method generates generate a not good score for this sample.

Figure 5.2: (1) input image, (2) estimated mask by single-class model, (3) estimated mask by multi-class model, and (4) integrated mask.

Figure 5.3: Top5 retrieval results obtained by our proposed "Easiness" score on Pascal VOC 2012 train_aug dataset.

Table 5.2: Results on PASCAL VOC 2012 *val set.*

methods	images additional	approach iterative	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	**mIoU**
MIL-FCN [45]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	25.7
EM-Adapt [4]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	38.2
CCNN [5]	-	-	65.9	23.8	17.6	22.8	19.4	36.2	47.3	46.9	47.0	16.3	36.1	22.2	43.2	33.7	44.9	39.8	29.9	33.4	22.2	38.8	36.3	34.5
MIL-sppxl [35]	✓	-	77.2	37.3	18.4	25.4	28.2	31.9	41.6	48.1	50.7	12.7	45.7	14.6	50.9	44.1	39.2	37.9	28.3	44.0	19.6	37.6	35.0	36.6
MIL-bb [35]	✓	-	78.6	46.9	18.6	27.9	30.7	38.4	44.0	49.6	49.8	11.6	44.7	14.6	50.4	44.7	40.8	38.5	26.0	45.0	20.5	36.9	34.8	37.8
MIL-seg [35]	✓	-	79.6	50.2	21.6	40.6	34.9	40.5	45.9	51.5	60.6	12.6	51.2	11.6	56.8	52.9	44.8	42.7	31.2	55.4	21.5	38.8	36.9	42.0
F/B prior[40]	-	✓	79.2	60.1	20.4	50.7	41.2	46.3	62.6	49.2	62.3	13.3	49.7	38.1	58.4	49.0	57.0	48.2	27.8	55.1	29.6	54.6	26.6	46.6
STC[7]	✓	✓	84.5	68.0	19.5	60.5	42.5	44.8	68.4	64.0	64.8	14.5	52.0	22.8	58.0	55.3	57.8	60.5	40.6	56.7	23.0	57.1	31.2	49.8
SEC [6]	-	✓	82.4	62.9	26.4	61.6	27.6	38.1	66.6	62.7	75.2	22.1	53.5	28.3	65.8	57.8	62.3	52.5	32.5	62.6	32.1	45.4	45.3	50.7
AF-SS [72]	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	52.6
AE-PSL [37]	-	✓	83.4	71.1	30.5	72.9	41.6	55.9	63.1	60.2	74.0	18.0	66.5	32.4	71.7	56.3	64.8	52.4	37.4	69.1	31.4	58.9	43.9	55.0
DCSM (ours)	-	-	76.7	45.1	24.6	40.8	23.0	34.8	61.0	51.9	52.4	15.5	45.9	32.7	54.9	48.6	57.4	51.8	38.2	55.4	32.2	42.6	39.6	44.1
Easiness (ours)	-	✓	81.6	64.9	25.8	71.4	29.2	57.8	75.2	68.0	72.7	15.2	46.6	33.8	56.7	57.1	60.9	60.7	24.1	65.4	31.5	43.9	35.3	51.3

Table 5.3: Results on PASCAL VOC 2012 *test set*.

methods	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	**mIoU**
Fully Supervised:																						
O2P [69]	85.4	69.7	22.3	45.2	44.4	49.6	66.7	57.8	56.2	13.5	46.1	32.3	41.2	59.1	55.3	51.0	36.2	50.4	27.8	46.9	44.6	47.6
SDS [18]	86.3	63.3	25.7	63.0	39.8	59.2	70.9	61.4	54.9	16.8	45.0	48.2	50.5	51.0	57.7	63.3	31.8	58.7	31.2	55.7	48.5	51.6
FCN-8s [20]	-	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
Deeplab Large FOV [23]	92.6	83.5	36.6	82.5	62.3	66.5	85.4	78.5	83.7	30.4	72.9	60.4	78.5	75.5	82.1	79.7	58.2	82.0	48.8	73.7	63.3	70.3
Using Additional Supervision:																						
CCNN w/size [5]	-	42.3	24.5	56.0	30.6	39.0	58.8	52.7	54.8	14.6	48.4	34.2	52.7	46.9	61.1	44.8	37.4	48.8	30.6	47.7	41.7	45.1
One point[48]	80.6	50.2	23.9	38.4	33.1	38.5	52.0	50.9	55.4	18.3	38.2	37.7	51.0	46.1	54.7	43.2	35.4	45.1	33.0	49.6	40.0	43.6
F/B prior + CheckMask[40]	87.4	65.7	26.0	64.2	43.7	53.2	72.6	63.6	59.5	17.1	48.0	43.7	61.2	52.0	69.3	54.8	43.0	50.3	34.6	59.2	42.0	52.9
weakly-supervised:																						
MIL-FCN [45]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	24.9
EM-Adapt [4]	76.3	37.1	21.9	41.6	26.1	38.5	50.8	44.9	48.9	16.7	40.8	29.4	47.1	45.8	54.8	28.2	30.0	44.0	29.2	34.3	46.0	39.6
CCNN [5]	-	21.3	17.7	22.8	17.9	38.3	51.3	43.9	51.4	15.6	38.4	17.4	46.5	38.6	53.3	40.6	34.3	36.8	20.1	32.9	38.0	35.5
MIL-ILP-seg [35]	78.7	48.0	21.2	31.1	28.4	35.1	51.4	55.5	52.8	7.8	56.2	19.9	53.8	50.3	40.0	38.6	27.8	51.8	24.7	33.3	46.3	40.6
F/B prior[40]	80.3	57.5	24.1	66.9	31.7	43.0	67.5	48.6	56.7	12.6	50.9	42.6	59.4	52.9	65.0	44.8	41.3	51.1	33.7	44.4	33.2	48.0
STC [7]	85.2	62.7	21.1	58.0	31.4	55.0	68.8	63.9	63.7	14.2	57.6	28.3	63.0	59.8	67.6	61.7	42.9	61.0	23.2	52.4	33.1	51.2
SEC [6]	83.5	56.4	28.5	64.1	23.6	46.5	70.6	58.5	71.3	23.2	54.0	28.0	68.1	62.1	70.0	55.0	38.4	58.0	39.9	38.4	48.3	51.7
AF-SS [72]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	52.7
AE-PSL [37]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	55.7
weakly-supervised:																						
DCSM (ours)	78.1	43.8	26.3	49.8	19.5	40.3	61.6	53.9	52.7	13.7	47.3	34.8	50.3	48.9	69.0	49.7	38.4	57.1	34.0	38.0	40.0	45.1
Easiness (ours)	83.0	67.5	29.7	69.7	28.8	59.7	71.2	66.4	69.8	18.6	49.8	44.7	49.4	60.5	73.5	61.8	32.7	62.7	39.0	34.3	36.5	52.8

Table 5.2 and Table 5.3 show the comparison with the other weakly-supervised segmentation methods. Our method showed the comparable performance with the other state-of-the-art methods on the same condition using only image-level-label as training data. Especially our proposed method achieved better result than F/B prior [40], STC [7], SEC [6] all of which employed (re)-trained DeepLab with the estimated initial masks on the weakly-supervised setting. Our approach also outperformed SDS which is based on fully supervised method proposed by [18].

5.4 Summary

In this chapter, we trained fully supervised segmentation model with DCSM outputs, and we selected "good seeds" by our another proposed method "Easiness". We estimated "Easiness" of prediction from visualization results and retrieved prediction results by "Easiness". In the training of deep CNN, there exists a trade-off between the number of training samples and the quality of training samples. However, we showed that we could boost the segmentation accuracy by combining data selection with data augmentation.

Chapter 6

Noise data estimation and interpolation using self-supervised difference detection

Class Activation Map (CAM) [36] visualizes a trained classification model. However, visualization often does not match actual object regions. To improve weakly-supervised segmentation from visualization, we need to consider mapping functions for visualization to segmentation. As such mapping functions, Conditional Random Field (CRF) [38] is widely known. CRF can refine rough probability maps for object location by fitting to the edge of regions using color and location information. A re-training approach is also known as a versatile approach for such mapping functions. In the re-training process, we generate pseudo pixel-level labels and we re-train a segmentation model with the generated labels. Wei et al. [7] demonstrated repeating this approach can gradually improve the accuracy of weakly-supervised segmentation. Though these mapping functions can refine visualization, the mapping functions do not always improve input data, they sometimes causes performance dropping. Therefore, in this chapter, we propose a robust learning method for such noise.

In this chapter, we denote the information used as the inputs of the mapping functions as *knowledge*, and we consider the supervision containing the noise as *advice*. The supervision for fully supervised learning that allows one-to-one mapping is *teacher*. We assume that the *advice* provides supervision,

which includes some correct and incorrect information. To make effective use of the information obtained from this *advice*, it is necessary to select useful information. We regard the regions where opinions differ between *knowledge* and *advice* as *difference*. Since *difference* in the two segmentation masks can be obtained by simple processing without annotation, it is a kind of self-supervised learning to train a model, which predicts *difference*. Self-supervised learning is a pretext task as a form of indirect supervision. For example, as notable works, colorization [73] and predicting the patch ordering [74] have been proposed.

Inferring *difference* in *knowledge* and *advice* from *knowledge* leads to predicting the advisor's *advice* in advance. In predicting *advice*, there are predictable *advice* and unpredictable *advice*. Certain *advice* can be easily inferred because many similar samples are included during training. Here, we assumed that *advice* contains a sufficient number of good information, and predictable information can be considered to be useful information. Based on this idea, we propose a method for selecting information by finding the true information in *advice* that can be predicted from the inference results of difference detection. Figure 6.1 shows the concept of the proposed approach.

In this chapter, we demonstrate that the proposed Self-Supervised Difference Detection (SSDD) module can be used in both the seed generation stage and the training stage of fully supervised segmentation. In the seed generation stage, we refine the CRF results for pixel-level semantic affinity (PSA) [39] by using the SSDD module. In the training stage, we introduce two SSDD modules inside the training loop of a fully supervised segmentation network. In the experiments, we demonstrate the effectiveness of the SSDD modules in both stages. In particular, the SSDD modules greatly boosted the performance of the WSS on the PASCAL visual object classes (VOC) 2012 dataset, and achieved new state-of-the-art. To summarize it, our contributions are as follows:

- We propose an SSDD module, which estimates the noise of the mapping functions of the weakly-supervised segmentation and select useful information.
- We show that the SSDD modules can be effectively applied to both the seed generation stage and the training stage of a fully supervised segmentation model.

- We obtained the best results on the PASCAL VOC 2012 dataset with 64.9% mean IoU on the *val set* and 65.5% on the *test set*.

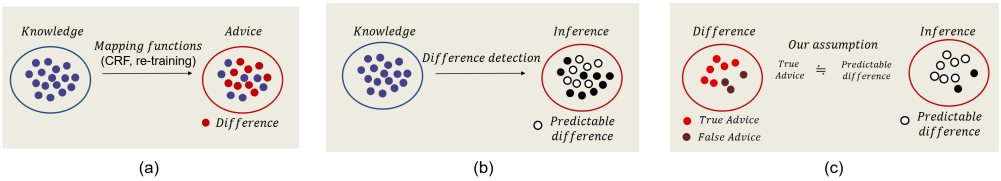

Figure 6.1: The concept of the proposed approach. (a) We denote the inputs of the mapping functions as *knowledge* and the outputs as *advice*. (b) The proposed difference detection network (DD-Net) estimates the *difference* between *knowledge* and *advice*. (c) In *difference*, the *advice* is divided into true *advice* and false *advice*. We assume that if the amount of true *advice* is larger than the amount of false *advice*, that is, if a set of false *advice* are outliers, then the predictable *advice* has a strong correlation with the true *advice*.

6.1 Method

There was no supervision for the mapping functions of segmentation in the weakly-supervised setting; therefore, it was necessary to consider a mapping for bringing the input close to the better segmentation results by using a method that incorporated human knowledge. We propose a method for selecting useful information from the results of the mapping functions by treating the results as supervision containing noise. We define the inputs of the mapping functions as *knowledge*, and the mapped results as *advice*. We predict the regions of *differences* between *knowledge* and *advice*, and we call this as the difference detection task. Using the inference results, we select the information of the *advice*.

6.1.1 Difference detection network

In this section, we formulate the difference detection task. In the proposed method, we predict the *difference* between *knowledge* and *advice*. Here, we define the segmentation mask of *knowledge* as m^K, the segmentation mask

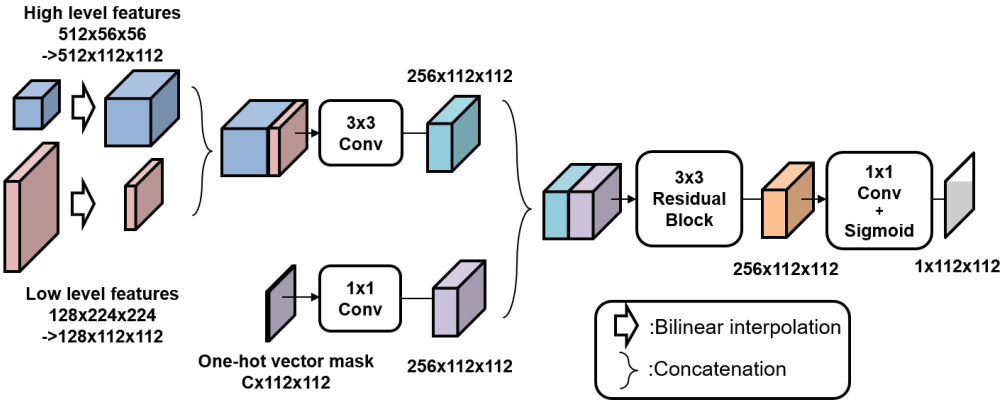

Figure 6.2: Difference Detection Network (DD-Net).

of *advice* as m^A, and their *difference* as $M^{K,A} \in \mathbb{R}^{H \times W}$.

$$M_u^{K,A} = \begin{cases} 1 & \text{if } (m_u^K = m_u^A) \\ 0 & \text{if } (m_u^K \neq m_u^A) \end{cases}, \tag{6.1}$$

where $u \in \{1, 2, .., n\}$ indicates a location of pixels, and n is the number of pixels. Next, we define a network of difference detection for deducing the *difference*. We use feature maps extracted from a trained CNN to assist the difference detection. In particular, we use high-level features $e^h(x; \theta_e)$ and low-level features $e^l(x; \theta_e)$ extracted from a backbone network, such as ResNet. Here, x is an input image, and e is an embedding function parameterized by θ_e. As shown in Figure 6.3, the confidence map of the input mask d is generated by difference detection network (DD-Net), $\text{DDnet}(e^h(x; \theta_e), e^l(x; \theta_e), \hat{m}; \theta_d), d \in \mathbb{R}^{H \times W}$, where $\hat{m}$ is a one-hot vector mask with the same number of channels to the target class number, θ_d is the parameter of the DD-Net, and $e(x) = (e^l(x), e^h(x))$. The architecture of DD-Net is shown in Figure 6.2; it consists of three convolutional layers and one Residual block with three inputs and one output. DD-Net takes either a raw mask or a processed mask as an input, and outputs the difference mask. This network performs learning using the following losses:

$$\begin{aligned} \mathcal{L}_{diff} = \frac{1}{|S|} \sum_{u \in S} (& J(M^{K,A}, d^K, u; \theta_d) \\ & + J(M^{K,A}, d^A, u; \theta_d)), \end{aligned} \tag{6.2}$$

where S is a set of pixels of the input spaces, and $J()$ is assumed to be a function that returns a loss for the binary cross entropy.

$$J(M, d, u) = M_u \log d_u + (1 - M_u) \log(1 - d_u).$$

Note that the parameters of the embedding function θ_e are independent of the optimization of θ_d. The training of DD-Net is self-supervised; therefore, neither special annotation nor additional data are needed.

6.1.2 Self-supervised difference detection module

In this section, we describe the details of the SSDD module shown in Figure 6.3, which integrates two masks adaptively according to the confidence maps. We denote a set of *advice* that are true in *difference* as $S^{A,T}$, and a set of *advice* that are false as $S^{A,F}$. The purpose of the method is to extract as many samples of $S^{A,T}$ as possible from the entire set of *advice* S^A. Let d^K be the inference results of *advice* from the given *knowledge*. The inference results are the probability distributions from 0 to 1, and the values have variations. The variations are caused by the difference in the difficulty of inference. The presence of similar patterns during training can have a strong influence on the difference in the difficulty of inference. Here, if there are a sufficient number of *advice* that are true values rather than false values, that is, if $|S^{A,T}| > |S^{A,F}|$, the larger values indicate that their *advice* most likely belong to $S^{A,T}$. However, for the values of d^K at a boundary, it is not clear whether *advice* belongs to $S^{A,T}$ or not; this should probably be different from sample to sample. Therefore, it is difficult to deduce a good *advice* directly from the size of the value of d^K. To alleviate the problem, we use the inference results about the state of *knowledge* for each *advice*. Although *advice*s have large variations in their distribution, these variations are less than the variations in the distribution of *knowledge* in general. Therefore, using *advice* to infer *knowledge* is assumed to be easier than using *knowledge* to *advice* inference. In this chapter, we consider the results of the inference of *knowledge* to *advice* for evaluating the difficulty of inference in each sample; we use the inferences for the thresholds for each sample. Specifically, we calculate the confidence scores of *advice* from the viewpoint of how close the values of d^K to d^A. The confidence score $w_u \in \mathbb{R}$ is defined by the following expression:

$$w_u = d_u^K - d_u^A + bias_u \tag{6.3}$$

Here, *bias* is a hyper parameter for a threshold of the selection obtained by the difference detection, and it is also an enhanced value for the categories in the presence labels of the input image. The refined masks m^D obtained from m^K and m^A are defined by the following expression:

$$m_u^D = \begin{cases} m_u^A & \text{if } (w_u \geq 0) \\ m_u^K & \text{if } (w_u < 0) \end{cases} \tag{6.4}$$

We denote this processing flow for generating new segmentation mask as an SSDD module in the after notation.

$$m^D = SSDD(e(x), m^K, m^A; \theta_d) \tag{6.5}$$

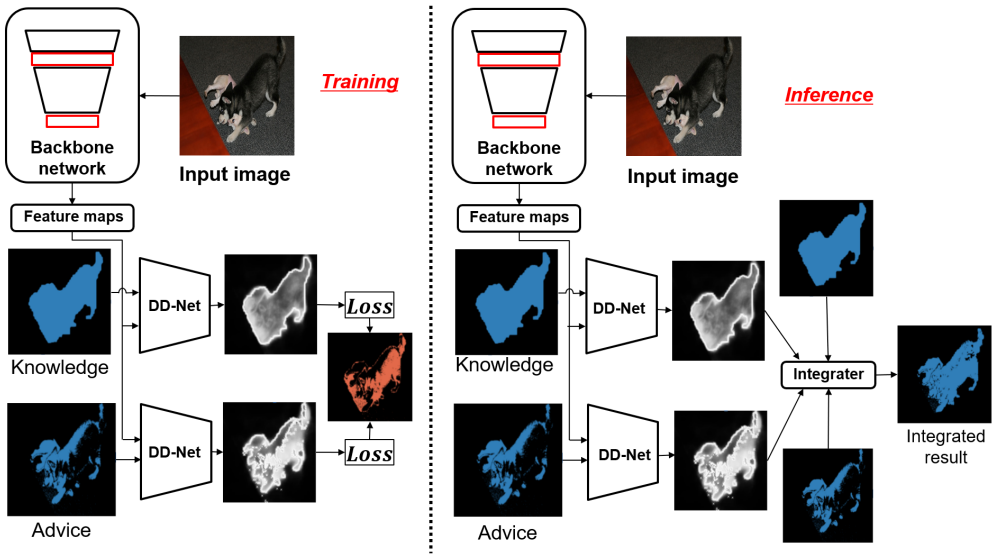

Figure 6.3: Overview of the DD-Net. The figure on the left shows the training of the DD-Net, and the right figure shows the processing of the integration using the results of difference detection.

6.2 Introducing SSDD modules into the processing flow of WSS

In this section, we explain how to use SSDD modules in the processing flow of WSS. The proposed method can be adapted to various cases by applying

inputs of the mapping function as *knowledge* and the results of the mapping function as *advice*. The processing flow that we adopted in this chapter consists of two stages: the seed generation stage with static region refinement and the training stage of a segmentation model with dynamic region refinement. In the first stage, we adapted the proposed method by applying the results of PSA as *knowledge* and its CRF results as *advice* (Section 6.2.1). In the second stage, we adapted the proposed method by applying the results of the first stage (Section 6.2.1) as *knowledge*, and the outputs of the segmentation models trained by the masks were applied as *advice* (Section 6.2.2).

6.2.1 Seed mask generation stage with static region refinement

PSA [39] is a method to propagate label responses to nearby areas that belong to the same semantic entity. Though PSA employs CRF for the refinement of the segmentation mask, CRF often fails to improve the segmentation masks; in fact, it degrades the masks. In this section, we refine the outputs of CRF in PSA by using the proposed SSDD module. We illustrate the processing flow of the first seed generation stage in Figure 6.4. Note that we omitted the input of the given image to an SSDD module for the sake of simplifying in the figure.

We denote an input image as x; the probability maps obtained by PSA are denoted as $p^{K0} = PSA(x; \theta_{psa})$, and its CRF results are denoted as p^{A0}. We obtain the segmentation masks (m^{K0}, m^{A0}) from the probability maps (p^{K0}, p^{A0}) by taking the argument of the maximum of the presence labels including a background category. We computed the loss of the DD-Net as follows:

$$\begin{aligned}\mathcal{L}_{diff0} = \frac{1}{|S|}\sum_{u\in S}(&J(M^{K0,A0}, d^{K0}, u; \theta_{d0}) \\ &+J(M^{K0,A0}, d^{A0}, u; \theta_{d0})),\end{aligned} \tag{6.6}$$

The proposed method is not effective when either of the segmentation masks or both of them do not have the correct labels. These cases are not only meaningless for the proposed refinement approach, but they may also harm the training of the DD-Net. We define the bad training samples by simple processing based on the difference in the number of the class-specific pixels, and we exclude them from the training.

In this work, we also train the embedding function by training a segmentation network with m^{K0} to obtain good representation for the inputs of high-level features and low-level features:

$$\mathcal{L}^{base} = \mathcal{L}^{seg}(x, m^{K0}; \theta_{e0}, \theta_{base}), \tag{6.7}$$

$$\mathcal{L}^{seg}(x, m; \theta) = -\frac{1}{\sum_{k \in K} |S_k^m|} \sum_{k \in K} \sum_{u \in |S_k^m|} \log(h_u^k(\theta)), \tag{6.8}$$

where S_k^m is a set of locations that belong to the class k on the mask m; h_u^k is the conditional probability of observing any label k at any location $u \in \{1, 2, ..., n\}$; and $\mathcal{C}$ is a set of class labels. θ_{e0} are parameters of embedding functions and θ_{base} are parameters for the segmentation branch. The training of θ_{e0} is independent of θ_{d0}.

The final loss function for the static region refinement using the difference detection is as follows:

$$\mathcal{L}_{static} = \mathcal{L}_{base} + \mathcal{L}_{diff0}. \tag{6.9}$$

After training, we integrate the masks (m^{K0}, m^{A0}) and obtain the integrated masks m^{D0} using the SSDD module with the trained parameter θ_{d0} as follows:

$$m^{D0} = SSDD(e(x), m^{K0}, m^{A0}; \theta_{d0}). \tag{6.10}$$

6.2.2 Training stage of a fully supervised segmentation model with a dynamic region refinement

When we train a fully supervised semantic segmentation model with pixel-level seed labels, the accuracy of the seed labels directly affects the performance of the segmentation. The performance gain is expected by replacing the seed labels to better the pixel-level labels during training. In this study, we propose a novel approach to constrain the interpolation of the seed labels during the training of a segmentation model. The idea of the constraint is to limit the interpolation of seed labels only to predictable regions of difference detection between newly generated pixel-level labels and seed labels.

In practice, we interpolate the pixel-level seed labels in two steps of each iteration as shown in Figure 6.5. Note that "SegNet" in the figure does

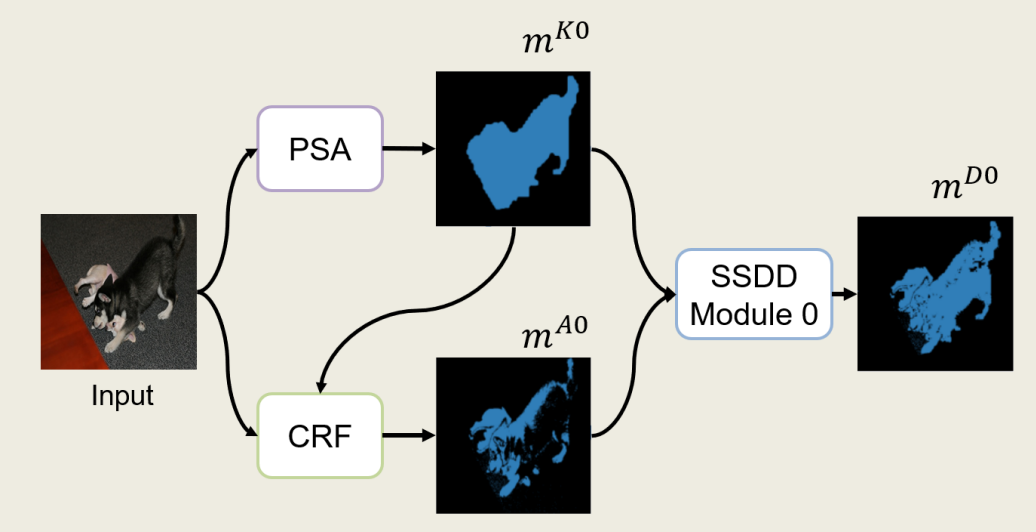

Figure 6.4: Processing flow at the seed mask generation stage with static region refinement.

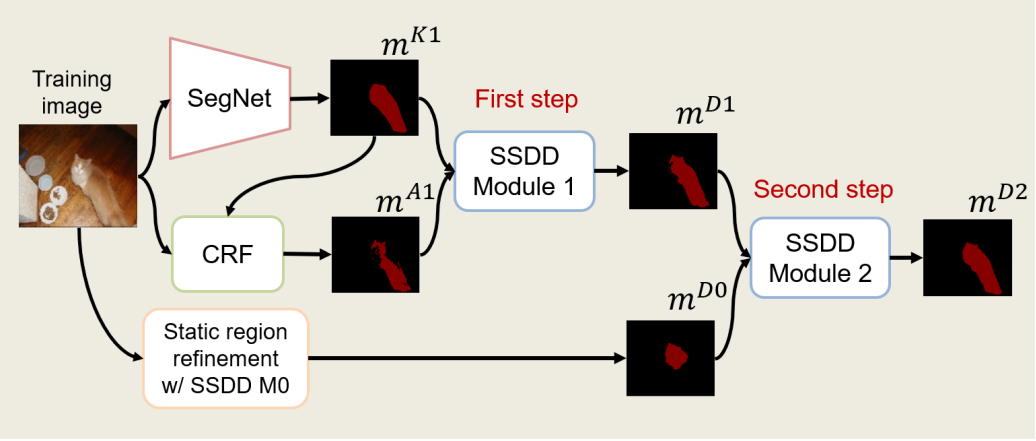

Figure 6.5: Illustration of the processing flow for the dynamic region refinement. ("SegNet" does not represent any specific network but represents any kind of network for fully supervised semantic segmentation.)

not represent a specific segmentation network; it represents any fully supervised segmentation network. In the first step, for an input image x, we obtain the outputs of the segmentation model $p^{K1} = Seg(e(x); \theta_{main})$ and its CRF outputs p^{A1}. We obtain the segmentation masks (m^{K1}, m^{A1}) from the probability maps (p^{K1}, p^{A1}) by taking the argument of the maximum of the presence labels including a background category. Then, we obtain the refined pixel-level labels m^{D1} by applying the proposed refinement method as follows: $m^{D1} = SSDD(e(x), m^{K1}, m^{A1}; \theta_{d1})$. In the second step, we apply the proposed method to the seed labels m^{D0} and to the mask m^{D1} obtained in the first step. The further refined mask m^{D2} is obtained by $m^{D2} = SSDD(e(x), m^{D0}, m^{D1}; \theta_{d2})$. We generate the mask m^{D2} in each iteration and train the segmentation model using the generated mask m^{D2}. We train the semantic segmentation model with the generated mask m^{D2} as follows:

$$\mathcal{L}_{main} = \mathcal{L}_{seg}(x, m^{D2}; \theta_{e1}, \theta_{main}), \tag{6.11}$$

The loss of DD-Net for m^{A1} and m^{K1} is as follows:

$$\begin{aligned} \mathcal{L}_{diff1} = \frac{1}{|S|} \sum_{u \in S} (& J(M^{K1,A1}, d^{K1}, u; \theta_{d1}) \\ & + J(M^{K1,A1}, d^{A1}, u; \theta_{d1})), \end{aligned} \tag{6.12}$$

In the second stage, we also exclude the bad samples (as done in Section static) based on the change ratio of pixels because the proposed method is not effective if the input segmentation masks do not have correct regions.

We explain how to train the DD-Net for (m^{D0},m^{D1}). The masks (m^{K1}, m^{A1}, m^{D1}) depend on the outputs of the segmentation model $Seg(e(x), \theta_{main})$. Therefore, if the learning of the segmentation model falls into a local minimum, the masks will become meaningless; all the pixels become background pixels or single foreground pixels. In this case, the inference results of the difference detection is also always constant, that is, $(D^K = 1, d^A = 1, d^A = d^K)$, and Equation 10.3 becomes $w = bias$. To escape from this local minimum, we create a new branch of a segmentation model and use it for learning the difference detection between m^{D0} and m^{D1}. Assume that the mask m^{sub} was obtained from outputs of the branch of the new segmentation model $p^{sub} = Seg(e(x); \theta_{sub})$. In the training of difference detection, we trained the network to learn the differences among (m^{D0}, m^{sub}) and (m^{sub}, m^{D1}) as

follows:

$$\mathcal{L}_{diff2} = \frac{1}{|S|}\sum_{u\in S}(J(M^{D0,sub}, d^{D0}, u; \theta_{d2}) + J(M^{sub,D1}, d^{D1}, u; \theta_{d2})), \tag{6.13}$$

If m^{sub} is the output, which is halfway between m^{D0} and m^{D1}, the replacement of the training samples will let the segmentation model exit from the situation ($d^K = 1, d^A = 1, d^A = d^K$), and the inference results of the difference detection will predict the regions that correlate with the *difference* between m^{D0} and m^{D1}. We train the parameters θ_{sub} from the following loss to achieve the outputs that are halfway between m^{D0} and m^{D1}.

$$\mathcal{L}_{sub} = \alpha\mathcal{L}_{seg}(x, m^{D0}; \theta_{e1}, \theta_{sub}) + (1-\alpha)\mathcal{L}_{seg}(x, m^{D1}; \theta_{e1}, \theta_{sub}), \tag{6.14}$$

where α is a hyper parameter of the mixing ratio of m^{D0} and m^{D1} .

The final loss function of the proposed dynamic region refinement method is calculated as follows:

$$\mathcal{L}_{dynamic} = \mathcal{L}_{main} + \mathcal{L}_{sub} + \mathcal{L}_{diff1} + \mathcal{L}_{diff2} \tag{6.15}$$

The range of the losses is not far from each other, then we do not leverage them in this formula.

6.3 Experiments

We evaluated the proposed methods using the PASCAL VOC 2012 data. The PASCAL VOC 2012 segmentation dataset has 1464 training images, 1449 validation images, and 1456 test images including 20 class pixel-level labels and image-level labels. Similar to the methodology followed by [35, 4, 6], we used the augmented PASCAL VOC training data provided by [18] as well, wherein the training image number was 10,582. For evaluation, we used an IoU metric, which is the official evaluation metric in the PASCAL VOC segmentation task. For calculating the mean IoU on the val and test sets, we used the official evaluation server. We compared the best performance of our method with the state-of-the-art methods on both the val and test sets.

6.3.1 Implementation details

Our experiments are heavily based on the previous research [39]. For the generating results of PSA results, we used implementations and trained parameters provided by the authors that are publicly available. We followed the methodology of [39] and set hyperparameters that gave the best performance. For the CRF parameters, we used the default settings provided by [38]. For the semantic segmentation model, we used a ResNet-38 model, which had almost the same architecture as that in [39]. The only difference was in the last upsampling rate; in the paper on PSA, the authors set the upsampling rate to 8, while we set the rate to 2 for reducing the computational cost of CRF. The input image size was 448 for training, and the test images and the output feature map size before the upsampling was 56. In the DD-Net, we used features obtained from the segmentation model before the last layer as the high-level features e^h and the features obtained before the second pooling layer as the low level features e^l. These feature map sizes were adjusted to 112 by 112 using the simple linear interpolation approach. We initialized the parameters of the segmentation models by using parameters trained with the PASCAL VOC images and their image-level labels with a pre-trained model using ImageNet, which was also provided in [39]. The codes provided by [39] did not include the training and test code for the segmentation models; therefore, we implemented our own codes. In the original paper on PSA, though the authors optimized the segmentation models by Adam; however, the performance was unstable in our re-implementation, and there were several unclear settings. Therefore, we used SGD for training the entire networks. We set an initial learning rate to 1e-3 (1e-2 for initialization without the pre-trained model), and we decreased learning rate with cosine LR ramp down [75]. For the static region refinement, we trained the network with batch sizes of 16 and 10 epochs. For the dynamic region refinement, we trained the network with batch sizes of 8 and 30 epochs. For the data augmentation and inference technique, we carefully followed the methodology used in [39]. We implemented the proposed method using PyTorch. All the networks are trained using four NVIDIA Titan X PASCAL. We will open the results of the proposed method and training codes.

6.3.2 Analysis of static region refinement

In the proposed method, we used fully connected CRF [38] with the same parameter settings as those for PSA [39], ($w_g = 3$, $w_{rgb} = 10$,$\theta_\alpha = 80$, $\theta_\beta = 13$, $\theta_\gamma = 3$) in the following kernel potentials: $k(f_i, f_j) = w_g exp\left(-\frac{|p_i - p_j|}{2\theta_\alpha^2} - \frac{|I_i - I_j|}{2\theta_\beta^2}\right) + w_{rbg} exp\left(-\frac{|p_i - pj|^2}{2\theta_\gamma^2}\right)$. To examine the relationship between the CRF params and results, we changed the values of (w_g, w_{rgb}) and evaluated the accuracy. Figure 6.6 shows a comparison of the proposed static region refinement with the PSA [39] and its CRF results on the training set. The weakening of w_{rgb} decreases the difference only between the CRF and the SSDD+CRF results; therefore the effectiveness of the proposed method reduces. However, the proposed method always indicates a high accuracy. The optimal weights are different for each image, and it is expected to be difficult to search them for each image. We consider that the proposed method realized the improvement of CRF by correcting the partial failure of CRF.

Figure 6.7 shows the difference detection results and their refined segmentation masks. In the fourth and fifth rows of Figure 6.7, we show the typical failure cases of the proposed method. The regions of small objects tend to vanish in the CRF, and the DD-Net also learns such tendencies, which causes the failure of the proposed re-refinement method. In the fifth row, both of the input segmentation masks fail to provide segmentation. In such cases, the proposed method is also not effective.

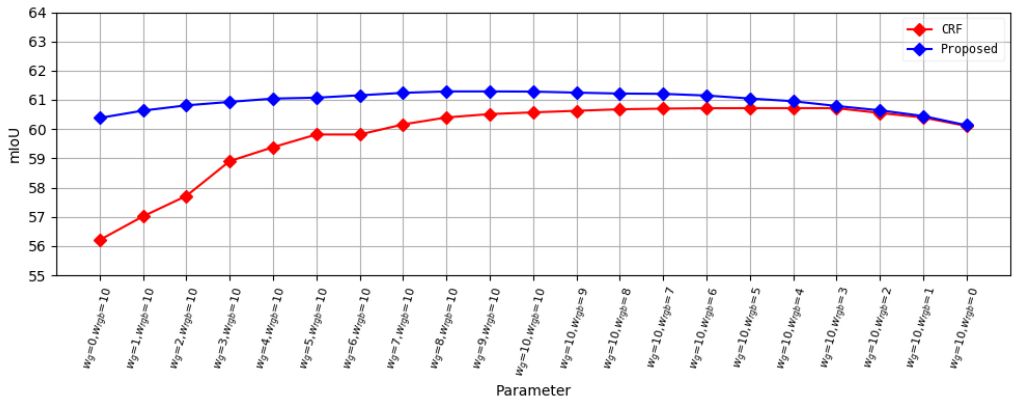

Figure 6.6: mIoU of the seed masks of the training images with different params values with only CRF and with SSDD and CRF.

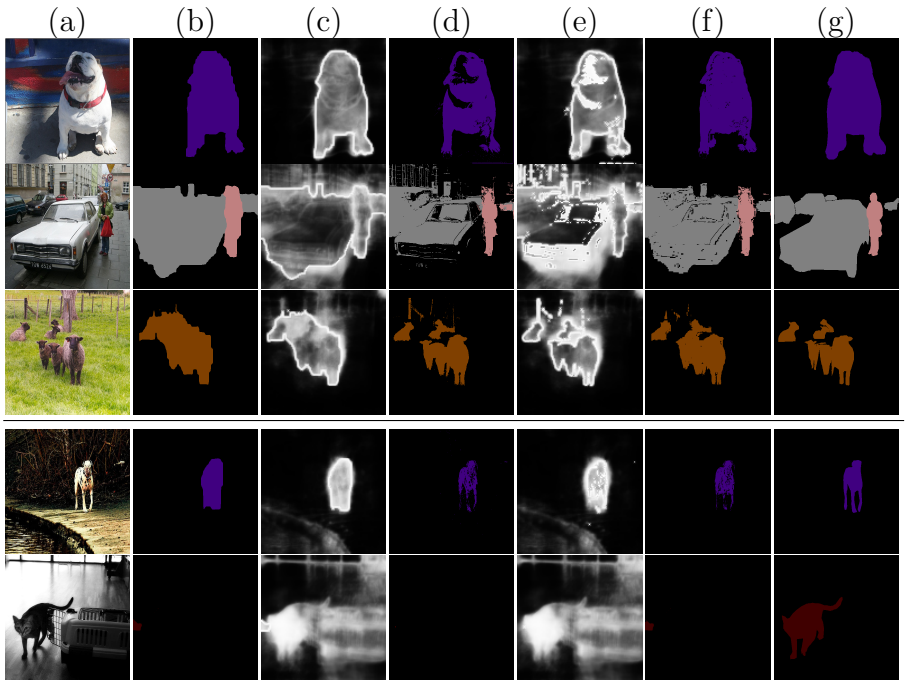

Figure 6.7: Each row shows (a) input images, (b) raw PSA segmentation masks, (c) difference detection maps of (b), (d) CRF masks of (b), (e) difference detection maps of (d), (f) refined segmentation masks by the proposed method, and (g) ground truth masks.

Table 6.1: Results on PASCAL VOC 2012 *val set.*

Methods	Bg	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Motor	Person	Plant	Sheep	Sofa	Train	Tv	**mIoU**
PSA [39]	88.2	68.2	30.6	81.1	49.6	61.0	77.8	66.1	75.1	29.0	66.0	40.2	80.4	62.0	70.4	73.7	42.5	70.7	42.6	68.1	51.6	61.7
SSDD	89.0	62.5	28.9	83.7	52.9	59.5	77.6	73.7	87.0	34.0	83.7	47.6	84.1	77.0	73.9	69.6	29.8	84.0	43.2	68.0	53.4	64.9
Gain	*+0.8*	*-5.7*	*-1.7*	*+2.6*	*+3.3*	*-1.5*	*-0.2*	*+7.6*	*+11.9*	*+5.0*	*+17.7*	*+7.4*	*+3.7*	*+15.0*	*+3.5*	*-4.1*	*-12.7*	*+13.3*	*+0.6*	*-0.1*	*+1.8*	*+3.2*

6.3.3 Analysis of the whole proposed method

We denote the dynamic region refinement as "SSDD" in all the tables. The score of the SSDD is with the CRF with parameters ($w_g = 3$, $w_{rgb} = 10$) that are default values from the author's public implementation. We also used the parameters for the CRF during training.

Comparison with PSA Table 6.1 shows the comparison of the dynamic region refinement method with the PSA. We observe that the proposed method outperforms PSA by more than 3.2 point margins. This clearly proves the effectiveness of the interpolation for the seed labels with the novel constraint by difference detection. The accuracy is greatly improved as compared with the results of the static region refinement because of the increase in the number of good *advice* by end-to-end learning of the segmentation model, that is, $|S^{A1,T}| > |S^{A0,T}|$.

In Table 6.1, we also show the gains between the proposed method and PSA for detailed analysis. We obtain over 10% gain on the cat, cow, horse, and sheep classes. Interestingly, all the classes that gave the large gain belonged to the animal category. However, in the potted plant, airplane, and person class objects, it was hard to improve the segmentation mask by using the proposed method. In the proposed method, we considered the precondition that *advise*, which is a true value, was larger than the value that was not a true value($|S^{A,T}| > |S^{A,F}|$). When this precondition was satisfied, the accuracy of the classes improved. If the precondition was not satisfied, the accuracy did not improve or the accuracy decreased.

Figure 6.8 shows the examples of the results of the re-implementation of PSA, the static region refinement, and the dynamic region refinement. Dynamic region refinement shows more accurate predictions on object location and boundary. The results of the static region refinement are outputs of a segmentation model re-trained with the masks in case of ($w_g = 3$, $w_{rgb} = 10$) in Figure 6.6. Note that we show the results before the CRF for detailed comparisons.

Comparison with the state-of-the-art methods Table 6.2 shows the results of the proposed method and the recent weakly-supervised segmentation methods that do not use additional supervisions on the PASCAL VOC 2012 validation data and PASCAL VOC 2012 test data. We observed that our method achieves the highest score as compared with all the existing methods, which use the same types of supervision [5, 4, 8, 40, 6, 42, 41, 76, 39]. The

Table 6.2: Comparison with the WSS methods without additional supervision.

Method	Val	Test
FCN-MIL [45]ICLR2015	25.7	24.9
CCNN [5]ICCV2015	35.3	35.6
EM-Adapt [4]ICCV2015	38.2	39.6
DCSM [8]ECCV2016	44.1	45.1
BFBP [40]ECCV2016	46.6	48.0
SEC [6]ECCV2016	50.7	51.7
CBTS [41]CVPR2017	52.8	53.7
TPL [42]ICCV2017	53.1	53.8
MEFF [76]CVPR2018	-	55.6
PSA [39]CVPR2018	61.7	63.7
SSDD	**64.9**	**65.5**

Table 6.3: Comparison of the WSS methods with additional supervision.

Method	Additional supervision	Val	Test
MIL-seg [35]CVPR2015	Saliency mask + Imagenet images	42.0	40.6
MCNN [50]ICCV2015	Web videos	38.1	39.8
AFF [72]ECCV2016	Saliency mask	54.3	55.5
STC [7]PAMI2017	Saliency mask + Web images	49.8	51.2
Oh et al. [52]CVPR2017	Saliency mask	55.7	56.7
AE-PSL [37]CVPR2017	Saliency mask	55.0	55.7
Hong et al. [51]CVPR2017	Web videos	58.1	58.7
WebS-i2 [49]CVPR2017	Web images	53.4	55.3
DCSP [55]BMVC2017	Saliency mask	60.8	61.9
GAIN [77]CVPR2018	Saliency mask	55.3	56.8
MDC [53]CVPR2018	Saliency mask	60.4	60.8
MCOF [54]CVPR2018	Saliency mask	60.3	61.2
DSRG [47]CVPR2018	Saliency mask	61.4	63.2
Shen et al. [44]CVPR2018	Web images	63.0	63.9
SeeNet [43]NIPS2018	Saliency mask	63.1	62.8
AISI [56]ECCV2018	Instance saliency mask	63.6	64.5
SSDD	-	**64.9**	**65.5**

proposed method outperforms the recent previous works on MEFF and TPL by large margins. As discussed earlier, the proposed method also outperforms the current state-of-the-art methods [39]. This result clearly indicates the effectiveness of the proposed method.

Table 6.3 shows the comparison of the proposed method with a few weakly-supervised segmentation methods that employ relatively cheap additional information. Surprisingly, the proposed method also outperforms all the listed weakly-supervised segmentation methods. The proposed methods outperformed the following methods: SeeNet [40], DSRG [7], MDC [6], GAIN [77], and MCOF [54] that employed fully supervised saliency methods. In addition, the score of the proposed method was also better than the results of AISC [56], which used instance-level saliency map methods. Note that AISC achieved 64.5% on the val set and 65.6% on the test set using an additional 24,000 ImageNet images for training. The score of the proposed method was also higher than the score of Shen et al. [44], which used 76.7k web images for training. It is not possible to have a completely fair comparison for them because of the difference of the network model, the augmentation technique, the number of iteration epochs, and so on. However, the proposed method demonstrates comparable performance or better performance without any additional training information.

Details of the simple decision In the proposed method, we select *advice* by inference results of difference detection. The confidence score is calculated from the viewpoint of how close the value of d^K to d^A. In the proposed method, if this difference is large enough, we ignore the *advice*. Therefore, if the inferences of the difference detection are too easy, the values of d^K for *advice* that is not true become close to d^A, and the proposed method does not work effectively. In particular, if the inference results of the difference detection are $(d^K = 1, d^A = 1, d^A = d^K)$, we cannot distinguish whether the *advice* belongs to the set of true values $|S^{A,T}|$ or the set of false values $|S^{A,F}|$ based on the results of the difference detection. Therefore, we judge the typical failure examples of *advice* and excluded them from the training sample so that the differences between d^K and d^A were large in the inference of the bad *advice*. To be concrete, when the number of differences in the pixels in each class of mask is obviously large, we assume that the *advice* has failed. We define the bad training samples as the pair of the masks for the

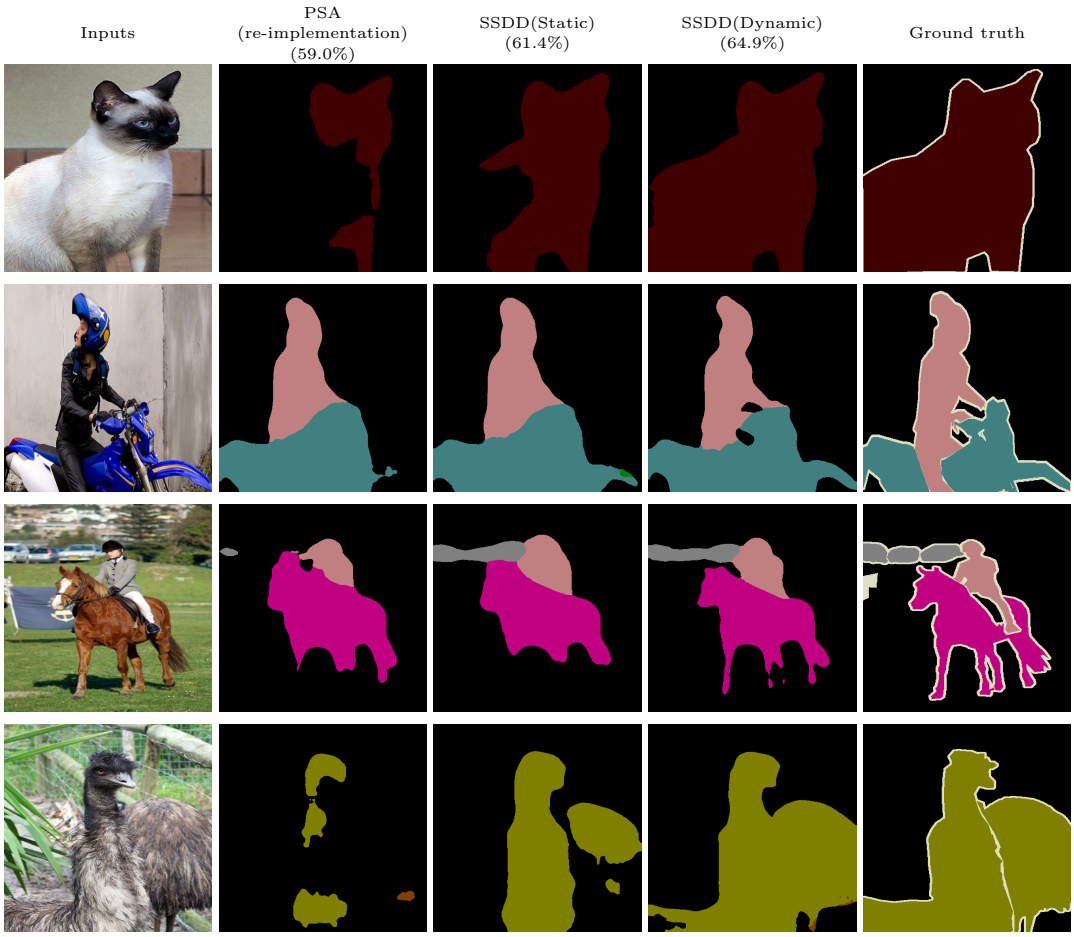

Figure 6.8: Segmentation examples of results on PASCAL VOC 2012.

difference detection that satisfies the following equation:

$$\forall c \in \mathcal{C}, \frac{|S_c^{m^A}|}{|S_c^{m^K}|} < 0.5, \tag{6.16}$$

where $\mathcal{C}$ is a set of image-level label of the input image. We decide the threshold 0.5 empirically.

Details of the bias in Equation 10.3 In Equation 10.3, we use *bias*, which is a kind of hyperparameter. In this section, we discuss this *bias*. We define the *bias* as follows:

$$bias_u = \begin{cases} b_{dd} \pm b_{class} & \text{if } m_u^A \text{ or } m_u^k \text{ belongs to } \hat{\mathcal{C}} \\ b_{dd} & \text{if } otherwise \end{cases}, \tag{6.17}$$

where $\forall c \in \hat{\mathcal{C}}$ satisfy $\frac{|S_c^{m^A}|}{|S_c^{m^K}|} < 0.5$ and $c \in \mathcal{C}$. b_{dd} is a bias for the difference

between *knowledge* and *advice*, and b_{class} is a bias for the class category. When the number of differences in the pixels in each class of mask is obviously large, it is assumed that the *advice* has failed, and to prioritize the label of that class over the results of the difference detection, we use the bias b_{class}. We defined the values of b_{dd} and b_{class} by using the grid search.

Values of hyperparameters We explore good hyperparameters by a grid search and verify the effect of the hyperparameters. We change the values of the hyperparameters and measure the mean IoU scores. Table 6.4 shows the hyper parameter values and the mean IoU scores. The hyperparameters (b_{dd}, b_{class}) are used in Equation 6.17 as the bias values. In $b_{dd} = 0.4$, the mean IoU score becomes the maximum value. We also set the bias b_{class} for the missing categories. We observe that the setting $b_{class} = 1.0$ achieved a maximum mean IoU. It is expected that the class biases for the missing categories help to the train for robustness. In addition, we also verify the effect of hyperparameters for coefficients of losses in Equation 6.11. Though we had expected that the value of α would affect the performance, the hyper parameter was not critical for the change of the mean IoU. The balanced setting, that is, $\alpha = 0.5$ showed the best score.

Table 6.4: Experimental results with different parameters.

b_{dd}	0.0	0.1	0.2	0.3	**0.4**	0.5
mIoU	62.2	63.9	64.6	64.2	64.9	62.7
b_{class}	0.0	0.5	**1.0**	1.5	2.0	
mIoU	64.3	63.0	64.9	64.5	63.7	
α	1.0	0.75	**0.5**	0.25	0.0	
mIoU	63.1	64.4	64.9	64.3	63.2	

Detailed comparison with existing works We show the detailed comparison with existing works on the PASCAL VOC 2012 *val* and *test* sets in Table 6.5, Table 6.6 and Table 6.7.

Table 6.5: Results on PASCAL VOC 2012 *val set* without additional supervision.

methods	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	**mIoU**
MIL-FCN [45]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	24.9
CCNN [5]	68.5	25.5	18.0	25.4	20.2	36.3	46.8	47.1	48.0	15.8	37.9	21.0	44.5	34.5	46.2	40.7	30.4	36.3	22.2	38.8	36.9	35.3
EM-Adapt [4]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	38.2
DCSM [8]	76.7	45.1	24.6	40.8	23.0	34.8	61.0	51.9	52.4	15.5	45.9	32.7	54.9	48.6	57.4	51.8	38.2	55.4	32.2	42.6	39.6	44.1
BFBP [40]	79.2	60.1	20.4	50.7	41.2	46.3	62.6	49.2	62.3	13.3	49.7	38.1	58.4	49.0	57.0	48.2	27.8	55.1	29.6	54.6	26.6	46.6
SEC [6]	82.4	62.9	26.4	61.6	27.6	38.1	66.6	62.7	75.2	22.1	53.5	28.3	65.8	57.8	62.3	52.5	32.5	62.6	32.1	45.4	45.3	50.7
CBTS [41]	85.8	65.2	29.4	63.8	31.2	37.2	69.6	64.3	76.2	21.4	56.3	29.8	68.2	60.6	66.2	55.8	30.8	66.1	34.9	48.8	47.1	52.8
TPL [42]	82.8	62.2	23.1	65.8	21.1	43.1	71.1	66.2	76.1	21.3	59.6	35.1	70.2	58.8	62.3	66.1	35.8	69.9	33.4	45.9	45.6	53.1
MEFF [76]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PSA [39]	88.2	68.2	30.6	81.1	49.6	61.0	77.8	66.1	75.1	29.0	66.0	40.2	80.4	62.0	70.4	73.7	42.5	70.7	42.6	68.1	51.6	61.7
SSDD (ours)	89.0	62.5	28.9	83.7	52.9	59.5	77.6	73.7	87.0	34.0	83.7	47.6	84.1	77.0	73.9	69.6	29.8	84.0	43.2	68.0	53.4	64.9

Table 6.6: Results on PASCAL VOC 2012 *test set* without additional supervision.

methods	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	**mIoU**
MIL-FCN [45]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	25.7
CCNN [5]	68.5	25.5	18.0	25.4	20.2	36.3	46.8	47.1	48.0	15.8	37.9	21.0	44.5	34.5	46.2	40.7	30.4	36.3	22.2	38.8	36.9	35.3
EM-Adapt [4]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	39.6
DCSM [8]	78.1	43.8	26.3	49.8	19.5	40.3	61.6	53.9	52.7	13.7	47.3	34.8	50.3	48.9	69.0	49.7	38.4	57.1	34.0	38.0	40.0	45.1
BFBP [40]	80.3	57.5	24.1	66.9	31.7	43.0	67.5	48.6	56.7	12.6	50.9	42.6	59.4	52.9	65.0	44.8	41.3	51.1	33.7	44.4	33.2	48.0
SEC [6]	83.5	56.4	28.5	64.1	23.6	46.5	70.6	58.5	71.3	23.2	54.0	28.0	68.1	62.1	70.0	55.0	38.4	58.0	39.9	38.4	48.3	51.7
CBTS [41]	85.7	58.8	30.5	67.6	24.7	44.7	74.8	61.8	73.7	22.9	57.4	27.5	71.3	64.8	72.4	57.3	37.0	60.4	42.8	42.2	50.6	53.7
TPL [42]	83.4	62.2	26.4	71.8	18.2	49.5	66.5	63.8	73.4	19.0	56.6	35.7	69.3	61.3	71.7	69.2	39.1	66.3	44.8	35.9	45.5	53.8
MEFF [76]	86.6	72.0	30.6	68.0	44.8	46.2	73.4	56.6	73.0	18.9	63.3	32.0	70.1	72.2	68.2	56.1	34.5	67.5	29.6	60.2	43.6	55.6
PSA [39]	89.1	70.6	31.6	77.2	42.2	68.9	79.1	66.5	74.9	29.6	68.7	56.1	82.1	64.8	78.6	73.5	50.8	70.7	47.7	63.9	51.1	63.7
SSDD (ours)	89.5	71.8	31.4	79.3	47.3	64.2	79.9	74.6	84.9	30.8	73.5	58.2	82.7	73.4	76.4	69.9	37.4	80.5	54.5	65.7	50.3	65.5

Table 6.7: Results on PASCAL VOC 2012 *val set* with additional supervision.

methods	info type†	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	**mIoU**
MIL-seg [35]	S	79.6	50.2	21.6	40.6	34.9	40.5	45.9	51.5	60.6	12.6	51.2	11.6	56.8	52.9	44.8	42.7	31.2	55.4	21.5	38.8	36.9	42.0
MCNN [50]	WV	77.5	47.9	17.2	39.4	28.0	25.6	52.7	47.0	57.8	10.4	38.0	24.3	49.9	40.8	48.2	42.0	21.6	35.2	19.6	52.5	24.7	38.1
AFF [72]	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	54.3
STC [7]	S	84.5	68.0	19.5	60.5	42.5	44.8	68.4	64.0	64.8	14.5	52.0	22.8	58.0	55.3	57.8	60.5	40.6	56.7	23.0	57.1	31.2	49.8
Oh et al. [52]	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	55.7
AE-PSL [37]	S	83.4	71.1	30.5	72.9	41.6	55.9	63.1	60.2	74.0	18.0	66.5	32.4	71.7	56.3	64.8	52.4	37.4	69.1	31.4	58.9	43.9	55.0
Hong et al. [51]	WV	87.0	69.3	32.2	70.2	31.2	58.4	73.6	68.5	76.5	26.8	63.8	29.1	73.5	69.5	66.5	70.4	46.8	72.1	27.3	57.4	50.2	58.1
WebS-i2 [49]	WI	84.3	65.3	27.4	65.4	53.9	46.3	70.1	69.8	79.4	13.8	61.1	17.4	73.8	58.1	57.8	56.2	35.7	66.5	22.0	50.1	46.2	53.4
DCSP [55]	S	88.9	77.7	31.3	73.2	59.8	71.0	79.2	74.5	80.0	15.1	73.3	10.2	76.1	72.2	69.1	72.1	39.9	73.9	14.6	70.3	53.1	60.8
GAIN [77]	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	56.8
MDC [53]	S	89.5	85.6	34.6	75.8	61.9	65.8	67.1	73.3	80.2	15.1	69.9	8.1	75.0	68.4	70.9	71.5	32.6	74.9	24.8	73.2	50.8	60.4
MCOF [54]	S	87.0	78.4	29.4	68.0	44.0	67.3	80.3	74.1	82.2	21.1	70.7	28.2	73.2	71.5	67.2	53.0	47.7	74.5	32.4	71.0	45.8	60.3
DSRG [47]	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	61.4
Shen et al. [44]	WI	86.8	71.2	32.4	77.0	24.4	69.8	85.3	71.9	86.5	27.6	78.9	40.7	78.5	79.1	72.7	73.1	49.6	74.8	36.1	48.1	59.2	63.0
SeeNet [43]	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	63.1
AISI [56]	IS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	64.5
SSDD (ours)	-	89.0	62.5	28.9	83.7	52.9	59.5	77.6	73.7	87.0	34.0	83.7	47.6	84.1	77.0	73.9	69.6	29.8	84.0	43.2	68.0	53.4	64.9

(† AS:Saliency mask, WV:web videos. WI Web images. IS Instance saliency mask.)

6.4 Summary

In this chapter, we proposed a novel method to refine a segmentation mask from a pair of segmentation masks before and after the refinement process such as the CRF by using the proposed SSDD module. We demonstrated that the proposed method could be used effectively in two stages: the static region refinement in the seed generation stage and the dynamic region refinement in the training stage. In the first stage, we refined the CRF results of PSA [39] by using the SSDD module. In the second stage, we refined the generated semantic segmentation masks by using a fully supervised segmentation model and CRF during the training. We demonstrated that three SSDD modules could greatly boost the performance of WSS and achieve the best results on the PASCAL VOC 2012 dataset over all the weakly-supervised methods with and without additional supervision.

Chapter 7

Related works of food image recognition

Recently, many peoples record daily foods using smart devices. The recorded information can provide numerical data for the number of calories and nutritional values. These data are beneficial for promoting healthy-eating habits. However, the process of recording is a burden to users.

Food image recognition has the potential to reduce the labor on the food recordings by replacing the manual procedure of taking a picture to food image recognition. Considering the recent trends to upload food images to SNS, it is important to simplify food recording also matches In terms of technical aspects, food image recognition also matches the recent trends in fashion, owing to recent significant advances in deep neural networks.

In food recognition, object detection and semantic segmentation are also important. From the food position in images, we can estimate the size of the food at bounding-box-level or pixel-level using object detection and semantic segmentation. The information of the food size are related to the amount of food, and we can utilize it for food calorie estimation. Food calorie estimation is one of a promising task in food recognition because it would be useful on health management. We can estimate food calorie based on not only the food category but also the amount of food using object detection and segmentation.

However, most CNN-based object detection and semantic segmentation methods require bounding-box-level labels or pixel-level labels. These annotations are very costly comparing with image-level labels, because many images with attached tags are available on hand-crafted open image data

sets such as ImageNet and on the web. In this study, we focus on weakly-supervised semantic segmentation, which requires neither pixel-wise annotation nor bounding box annotation but only image-level annotation.

In general, object detection and semantic segmentation with bounding-box annotation or pixel-wise annotation are referred to as fully-supervised methods, while object detection and semantic segmentation with only image-level annotation are referred to as weakly supervised methods.

In this thesis, we focus on image recognition in the domain of food. Our study is also related to object detection and semantic segmentation. In terms of related works, we discuss previous food recognition studies, including food detection and segmentation, and recent CNN-based detection and segmentation work for generic images.

Food image recognition is a promising application of visual object recognition, owing to its potential in estimating food calories and analyzing the eating habits of people for their general well-being. There have been numerous studies on food image recognition that have been published [78, 79, 80, 81, 82, 83, 84].

Moreover, the effectiveness of convolutional neural networks (CNN) has been recently demonstrated for large-scale object recognition at the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2012. Krizhevsky et al. [85] won the ILSVRC 2012, outperforming all other teams who employed conventional hand-crafted feature approaches. In the CNN approach, input data consist of a resized image, and the output is a class-label probability. In other words, CNN includes all the object recognition steps such as local feature extraction, feature coding, and learning. In general, the advantages of CNN includes the adaptive estimation of optimal feature representations for datasets, which is not possible using conventional hand-crafted feature approaches. In conventional approaches, we first extract local features such as SIFT and SURF and then code them into bag-of-feature or Fisher Vector representations. In the context of food image recognition, classification accuracy based on the UEC-FOOD100 dataset [81] improved from 59.6% [80] to 72.26% [86] by replacing the Fisher Vector and linear SVM with CNN.

However, most studies assume that one food image represents only one food item. The approaches presented in these studies cannot handle an image that contains two or more food items such as an image of a hamburger and French fries. To list all food items in a given image of food and to estimate the calories associated with the food, the segmentation of food is needed.

Some studies attempted food region segmentation [81, 87, 88, 89].

Matsuda et al. [81] proposed the use of multiple methods to detect food regions, including Felzenszwalb's deformable part model (DPM) [90], a circle detector, and the JSEG region segmentation method [91].

He et al. [89] employed local variation [92] to segment food regions for estimating the total calories associated with the food in a given food photo. In some studies on mobile food recognition [87, 88], users were asked to point to the rough locations of each food item in an image of food and to perform GrabCut [93] for extracting food item segments.

In addition, there have been several studies on the estimation of calories using computer vision techniques. Kong et al.[94] reconstructed 3D food models using multi-angle pictures and estimated the calories associated with the food using the cubic volume of 3D models. Chen et al.[95] recognized an image and computed the cubic volume using depth information. It must be noted that they obtained depth information using a sensor. 3D base calorie estimation methods tend to be laborious for users. On the other hand, Myers et al.[96] proposed a calorie estimation application called"im2calorie." They obtained each pixel depth information through deep learning prediction and estimated the food calories. However Myers et al. have not achieved practical use.

Pouladzadeh et al.[97] estimated food calories from the segmentation results of an image. They defined a thumb as the base food area and estimated food volumes and calories from the area ratios of the thumb and the food. While we can always take a picture of food using our thumbs, this method can potentially distort the image and taking a picture with only one hand can be difficult. As more recent study, Myers et al. [96] proposed calorie estimation application which called "im2calorie". They obtained each pixel depth information by prediction of deep learning and estimated calories. Ege et al. [98] estimated calorie by logistic regression and demonstrated that multi-task learning of dish detection and calorie estimation can enhance the accuracy of calorie estimation. Ege et al. [99] also proposed an approach to estimate the real food size from grains of rice, which is useful for calorie estimation. We show the figures for the examples of im2calorie [96] and the concept of the method for estimation of the real food size [99] in Figure 7.1 and Figure 7.2, respectively.

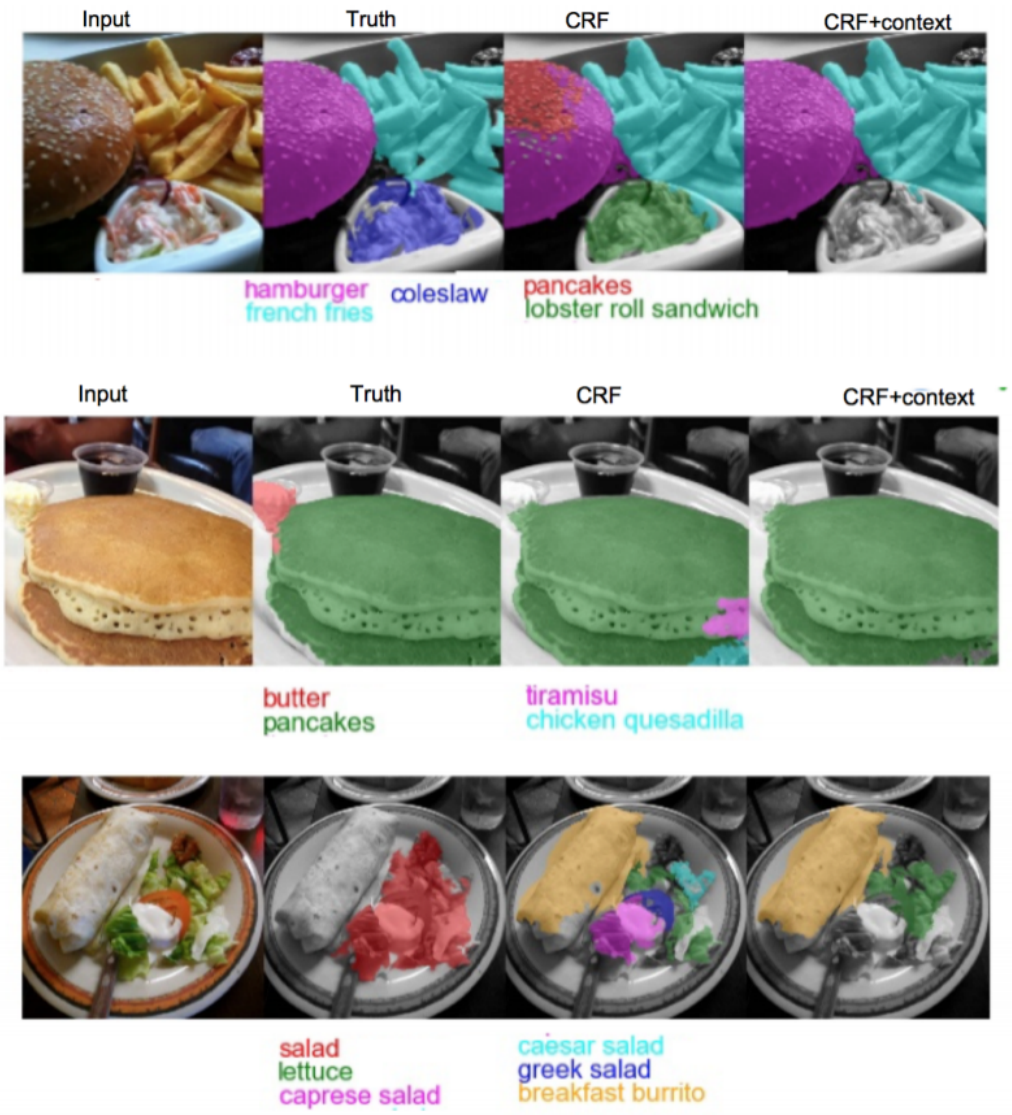

Figure 7.1: The concept of the examples of segmentation of im2calorie. This figure is cited from [96].

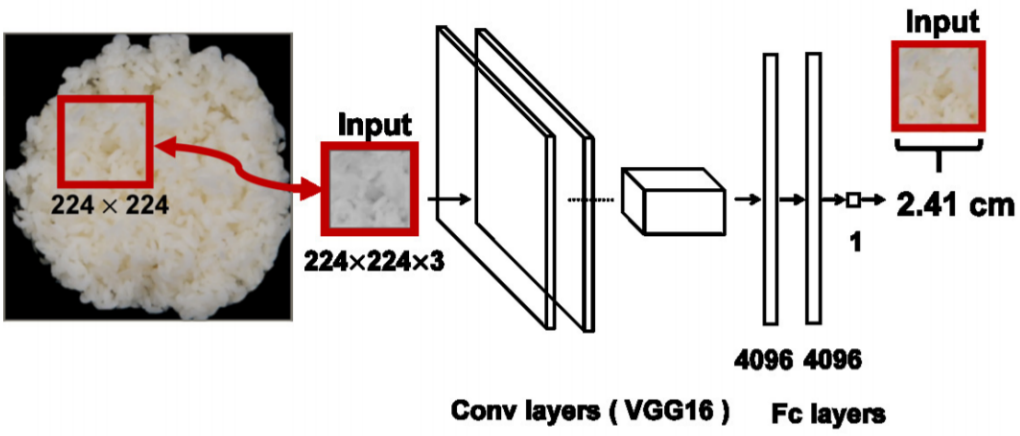

Figure 4: Real scale estimation network.

Figure 7.2: The concept of the method for estimation of the real food size [99]. This figure is cited from [99].

Chapter 8

Backward-based weakly-supervised food segmentation

In this chapter, we propose a new region segmentation method which combines the ideas of RCNN [100] and Simonyan et al. [2]. In RCNN, firstly, region proposals were generated by selective search [19], then extracted CNN activation features from all the proposal, applied SVM to evaluate proposals and integrated them by non-maximum suppression to produce object bounding boxes. They fine-tuned CNN pre-trained with ImageNet 1000 categories using the PASCAL VOC dataset having 20 categories.

Meanwhile, Simonyan et al. [2] proposed a method to generate object saliency maps by back propagation (BP) over a pre-trained CNN, and showed it enabled semantic object segmentation by applying GrabCut [93] using saliency maps as seeds.

In this chapter, we firstly obtain region proposals by selective search [19], secondly estimate saliency maps with BP-based methods over the pre-trained CNN for each of the region proposals after aggregation of overlapped proposals, thirdly apply GrabCut using the obtained saliency maps as seeds of GrabCut, and finally apply non-maximum suppression to obtain final region results.

In the experiments, we examined food segmentation with UEC-FOOD100 [81] and compared the proposed method and RCNN [100] regarding food detection performance in the bounding box level. In addition, we used PASCAL VOC 2007 as well. Our method outperformed RCNN by both of the dataset.

Although CNN [86, 79] has been applied to food image classification problem so far, no work tackled food image segmentation problems with CNN-based methods. As long as we know, this is the first work to apply a CNN-based segmentation method to food image segmentation task.

In addition, we estimate calories using the segmentation results. We obtained food area sizes in an image form segmentation results with pixel level food location. However it varies from the actual food size, since the food size in an image is relative.

Some previous works estimate food size from the area ratio of a base object and foods. There are some objects which is comparable size with food in an image and cab be took at food time such as e.g. cash cards and a thumb. We can compute the actual food size from an actual base object size and area ratio of a base object and foods. However we need to take a picture of the base object such a cards or thumb with foods and these base objects have the potential to make looking worse. Considering the recent trend uploading food images to SNS, the problem will become a large obstacle. Previous methods also have another problem, if forgot taking a picture with a base object we can't estimate calories and someones may feel the procedure taking a base object at food time as labor.

In contrast to previous works, we present a calorie estimation method using only food segmentation results by limiting target to a multiple food image. In short words, we decide a food as a base object in multiple foods. We estimate a calorie from the segmentation results by food area ratios without preparing a base object in advance.

We summarize the contribution as below:

- We initially achieved CNN-based food segmentation without pixel-wise annotation.
- We initially estimated food calories with CNN and evaluated results in practice.
- We proposed a novel calorie estimation method which is based on area ratios without non-food specific item.

8.1 Proposed Method

The proposed method on CNN-based region detection consists of the following steps as shown in Figure 8.1:

- Apply selective search and obtain 2000 bounding box proposals at most.
- Group them and select bounding boxes.
- Perform backpropagation over the pre-trained CNN regarding all the selected bounding boxes.
- Obtain saliency maps by averaging BP outputs within each group.
- Extract segments based on the saliency maps with GrabCut.
- Apply non-maximum suppression (NMS) to obtain final region results.

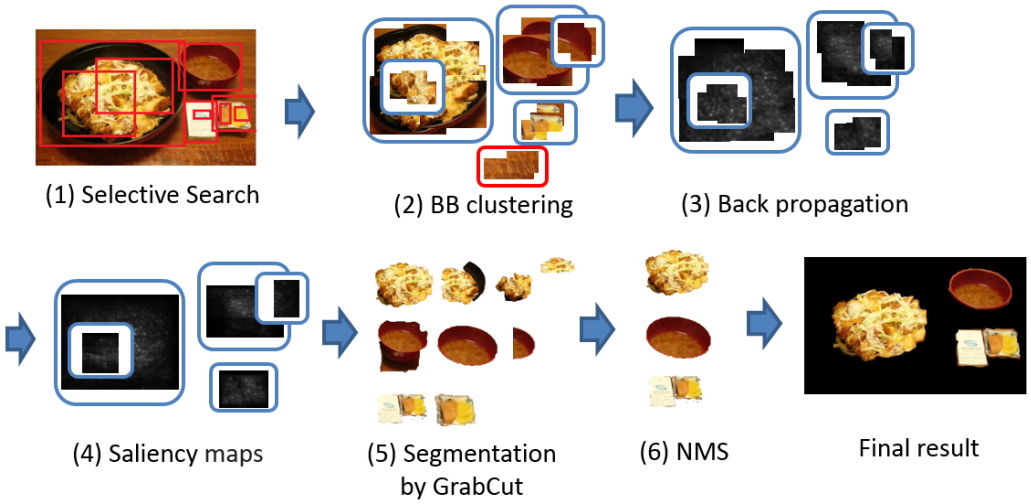

Figure 8.1: The processing flow of the proposed method.

8.1.1 Selective Search

In the work by Simonyan et al. [2], they applied their method to a whole image. This brings accepTable results for images containing only one prominent object, while it difficult to handle images containing many objects.

Especially, in case that a target image includes multiple same-class objects, Simonyan et al.'s method sometimes extracts multiple objects as one large object region and fails to extract individual object regions, since they employed GrabCut is a generic region segmentation method.

Then, first, we apply selective search [19] to obtain food region candidates where we perform estimation of saliency maps and region segmentation, which is inspired by RCNN [100]. We obtain 2000 region proposals represented by bounding boxes at most from the selective search implementation. [1]

8.1.2 Bounding Box Grouping

2000 bounding boxes (BB) are too many to perform estimation of BP-based saliency maps and GrabCut within each of them. Therefore, we perform bounding box clustering to reduce the number of bounding boxes. We group the bounding boxes based on the ratio of intersection over union (IOU) into 20 BB groups at most, and we removed the groups the number of the members of which is less than 15 BBs. The rest BB groups are regarded as food region candidates. Note that BB groups sometimes contain other BB groups inside them, as shown in Figure 8.1(2), because we cluster BBs according to the ratio of intersection over union (IOU).

8.1.3 Saliency Maps by Back Propagation over Trained CNN

According to Simonyan et al. [2], we estimate food saliency maps which represents rough position of target objects employing back propagation (BP) over the trained CNN. In general, BP is used for training of CNNs, which propagates errors between estimated values and ground truth values in an output layer from an output layer to an input layer in the backward direction. In case of training, the weights of CNNs are modified so that total errors are reduced. Reducing errors is equivalent to increasing the output scores of given classes. If propagating errors to an input image, we can obtain a map indicating which pixels need to be changed to increase the scores of given classes. Such pixels are expected to correspond to the object location in the images. This is the explanation why BP can be used for object region

[1] Downloaded from http://koen.me/research/selectivesearch/

estimation. The advantage of this method is that it does not need neither pixel-wise annotation or bounding box annotation as training data. The only thing needed is a trained CNN with labeled images.

To obtain a BP base saliency map, first of all, we compute the derivative w of a class score vector Sc with respect to the layer I at the point (activation signal) I_0:

$$w = \left.\frac{\partial S_c}{\partial I}\right|_{I_0} \tag{8.1}$$

The size of a saliency map is the same as a given image which is height $\times$ width $\times$ channel. In [2], they obtained a saliency map by select max values regarding channels as follows: the saliency map $m_{i,x,y}$ is computed as

$$m_{i,x,y} = \max |w_{i,h_i(x,y,k)}| \tag{8.2}$$

To perform BP, both forward pass and backward pass computation are needed. Forward pass computation is equivalent to classification by CNN. We provide a region cropped within each selected BB to CNN in the forwarding direction, and obtain soft-max scores of all the categories. Then, we select the top five categories, and provide the vector as S_c where only the elements corresponding to the top five categories are 1 and the rest elements are 0 into the backward pass. Note that the size of an input image is fixed to 227×227 in case of using AlexNet. We resize (shrink or enlarge) cropped regions to fit the fixed size.

In the next place, we unify saliency maps in each bounding box groups. Simonyan et al. [2] prepared 10 sub-images by cropping and reflecting and averaged them to obtain a saliency maps. Because obtained saliency maps from a derivative is sparse since. We follow this process in slightly different approaches. We prepare sub-images from bounding box groups.over lapped regions. Bounding box groups can be dealt as sub-images so that they are overlapped regions. Especially, We obtain saliency maps from each bounding box groups and average them after normalization.

Besides, we consider that there are two other methods than the BP-based method proposed by Simonyan et al. [2]. One is deconvolution (deconv) proposed by Zeiler et al. [101], the other is guided back propagation (guided BP) proposed by Springerberg et al. [102]. Basic ideas of the three method are the same. Only the ways to back propagation through ReLUs (rectified

linear units) are different as below:

$$\text{BP} : \frac{dz^i}{dx^i} = \frac{dz^{i+1}}{dx^{i+1}} \cdot (conv^{i+1} > 0) \tag{8.3}$$

$$\text{Deconv} : \frac{dz^i}{dx^i} = \frac{dz^{i+1}}{dx^{i+1}} \cdot \left(\frac{dz^{i+1}}{dx^{i+1}} > 0\right) \tag{8.4}$$

$$\text{GBP} : \frac{dz^i}{dx^i} = \frac{dz^{i+1}}{dx^{i+1}} \cdot \left(\frac{dz^{i+1}}{dx^{i+1}} > 0\right) \cdot (conv^{i+1} > 0) \tag{8.5}$$

Originally, guided BP and deconv were proposed as visualizing methods of the inside of a CNN which was regarded as a black box for analysis and understanding of it. Guided BP can emphasis edges of objects, which is good for visualizing trained filters inside a CNN. objects but not good for region extraction. Figure 8.2 shows saliency maps, and GrabCut results obtained by the three methods.

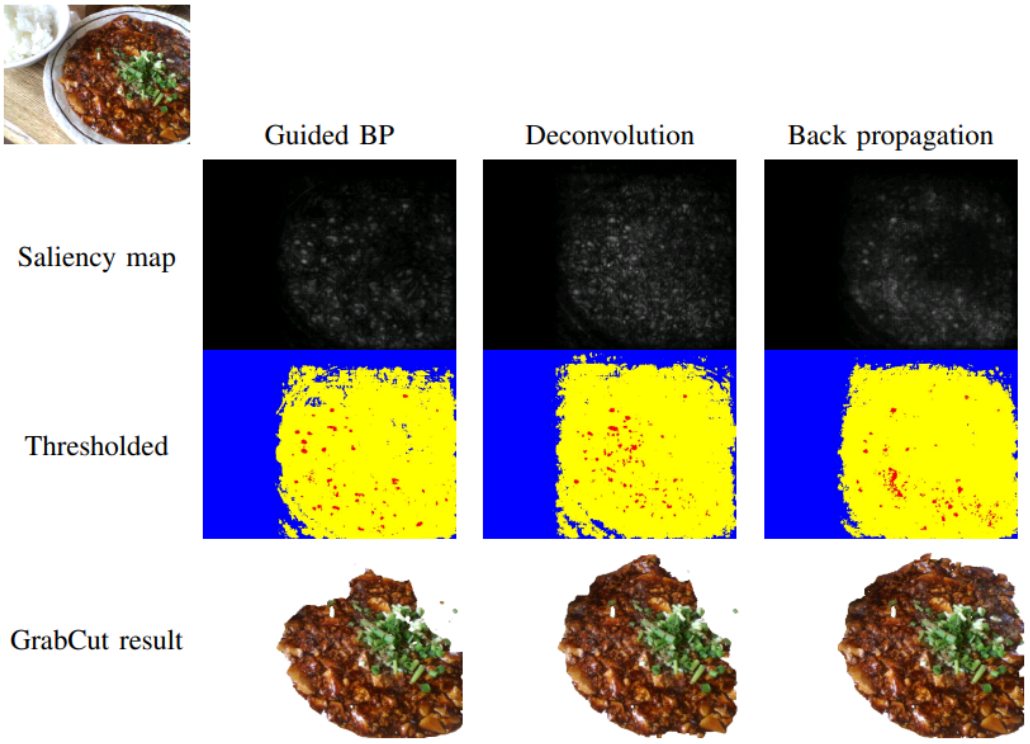

Figure 8.2: Saliency maps, thresholded saliency maps and GrabCut results generated by three kinds of BP-variant methods: Guided BP, Deconvolution, Back Propagation.

After obtaining saliency maps of BBs, we average them within each BB group and obtain saliency maps of BB groups as shown in Figure 8.1(4). The pixels with higher values are expected to correspond to objects.

8.1.4 Segmentation by GrabCut

In this step, we apply GrabCut [93] to each BB group region to extract whole object regions, because BP can estimate only the most discriminative parts of objects. To use GrabCut, both foreground and background color models are needed. In the similar way as Simonyan et al. [2], the foreground regions are estimated from the pixels with the top 3% saliency, while the background regions are estimeted from the lower 40% saliency. The red regions and the blue regions represent the foreground and the background regions in the thresholded images in Figure 8.2. Because we apply GrabCut to each BB group independently, we obtain several regions for one objects as shown in Figure 8.1 (5).

To integrate overlapped regions, we apply non-maximum suppression (NMS), and we obtain non-overlapped regions as shown in Figure 8.1 (6). Finally, we estimate rectangular regions bounding obtained segmented regions, and provide them to the trained CNN to obtained labels for each of the segmented regions. In addition, in the experiments, we use the extracted bounding boxes for evaluation.

8.2 Calorie estimation

We estimate calories from the segmentation results with area ratios of multiple foods. We can estimate a calorie without changing the picture view and some labors so that we don't use a non-food specific item for the base area. On the other hand, we need to define following two additional visions because of differing from previous works:

- To choose a base food in multiple foods for caluculating the calorie in practice.
- To investigate each food area ratio.

We choose a base area from segmentation results of multiple foods. Former work with a specific item for a base area size need to investigate only

the area ratio of the item and foods, while we need each food area ratio to calculate a calorie. In this section, we show how we get these additional visions and how we calculate a calorie in practice.

8.2.1 A choice of a base food

We don't fix a base food so that there are a lot of patterns in food combinations. Therefore we define each food priority for choosing a base food class at each time. We decide the priority based on a tendency of unchanging food volumes. Some food volumes change frequently , while some foods volume rarely change. For example, in "Teishoku" which is Japanese traditional multiple foods menu, we can often change "rice" volume as options, while we can't change "miso-soup" volume. There are differences in the tendency of unchanging food volumes and we want to define food which volume is sTable as a high priority class for choosing a base food class.

We explore foods which volume is sTable such like not rice but miso-soup from multiple-food data on UECFOOD100. UECFOOD100 multiple food data has 1500 multiple food images including a variety kind of foods such as "Teishoku","Bargar set" and so on. UECFOOD100 multiple food data also has bounding box information on food category included UECFOOD100. We compute each area ratios of category k against category to limitation k' on UECFOOD100 multiple images. We approximate region area ratios with BB area ratios due to the limitation of annotation in uec-multiple food. We denote the space of images by I. For any image $i \in \mathrm{I}$, T^i is a collection , $(t_1^i, \cdots, t_p^i)$ of each bounding box annotation of i at size p. The bounding box annotations belong to a set K of category labels. A area ratio of $r(k_{t_p}, k_{t_q})$ is formulated as below:

$$r(t_p^i|k, t_q^i|k') = \frac{s(t_p^i|k)}{s(t_q^i|k')} \tag{8.6}$$

Where $s(k|t_p^i)$ is area of annotation t_p^i which belong to category k. A mean area ratio $\hat{r}(k, k')$ is computed as below:

$$\hat{r}(k, k') = \sum_{i \in \mathrm{I}} \sum_{p} \sum_{q} r(t_p^i|k, t_q^i|k')[k \neq k', p \neq q] \tag{8.7}$$

We can obtain standard deviation $\sigma(k, k')$ from $\hat{r}(k, k')$ and each r. Each mean value size varies in each class k. Therefore we compute a relative

standard variation distribution of k for k' $rsd(k, k')$.

$$rsd(k, k') = \frac{\sigma(k, k')}{\hat{r}(k, k')} \tag{8.8}$$

A relative standard variation is normalized by mean value. However a tendency remains, small mean values lead relative standard variation small. Especially large food item tends to hold small relative standard variation value. There should be no relevance between absolute size and size variation distribution. To solve this problem, furthermore, we add a normalization procedure. Simply we take a mean for pair class. We define a variation distribution of k for k' as $v(k, k')$.

$$v(k, k') = \left(\frac{\sigma(k, k')}{\hat{r}(k, k')} + \frac{\sigma(k', k)}{\hat{r}(k', k)} \right) /2 \tag{8.9}$$

This normalization make $v(k, k') = v(k', k)$ and restrict unbalanced variation for an absolute food size. We integrate a variation distribution of k by summing up in each other class k'.

$$v(k) = \frac{1}{|\mathrm{K}|} \sum_{k' \in \mathrm{K}} \hat{v}(k, k')[k \neq k'] \tag{8.10}$$

We assume that a small variance mean that the food class area ratio in other food classes is stable. Therefore , We define priorities based on the mean variation distributions $v(k)$. Table 8.1 shows priorities and each class variation distribution values. In Table 8.1, we exclude some foods on UECFOOD100 due to lack of enough number items for evaluation of variation.

8.2.2 Calorie estimation from area ratios

Our goal is to estimate calories for each food classes in an image. We choose a food from multiple foods in an image based on priorities which is defined in Sec 8.2.1. An absolute food area ratio $R(k)$, for class k is computed as follows:

$$R(k) = \hat{r}(k, k_b) \frac{s(k_b)}{s(k)} \tag{8.11}$$

Where k_b is a base food class.

Simply We compute calorie C_k of class k using an absolute food area ratio $R(k)$

$$C_k = R(k) * c_k \tag{8.12}$$

Where c_k is standard calorie.

Briefly, we focus on food category and food volume as calorie decision factors and we achieve it by segmentation. Especially, we compute food volume from only food area ratios by choosing a base food class from multiple foods in an image.

8.3 Inplementation details

We trained CNN by two steps. First of all, we pre-trained AlexNet [85] CNN with 2000 categories in the ImageNet including 1000 food-related categories. Secondly, we fine-tuned AlexNet [85] with the UEC-FOOD100 dataset [81] and used it to estimate food categories and saliency maps. Figure 8.3 shows detail of fine-tuned AlexNet model construction. We trained this network using the Caffe [68] toolbox.

Note that, in calculating saliency mpas, we used MatConvNet [103]. MatConvNet can compute backward easily since this tool is provided by a lab which Simonyan has belonged to.

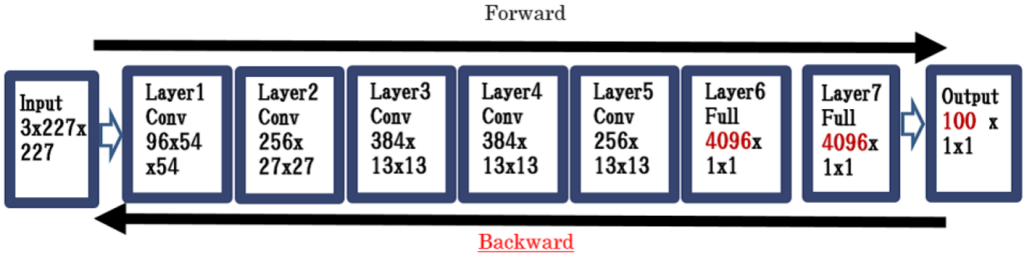

Figure 8.3: For detail of AlexNet construction.

8.4 Experiments

In the experiments, we used the UEC-FOOD100 dataset [81] and the PASCAL VOC 2007 detection dataset, both of which have bounding box information as well as class labels.

8.4.1 Food detection evaluation

The UEC-FOOD100 dataset [81] contains one hundred kinds of food photos. The total number of the food photos are 12740 including 1174 multiple-food photos. In the experiment, we used 1174 multiple-food photos including 3045 food items for testing, while we used the rest 11566 photos for fine-tuning a CNN pre-trained with the ImageNet 1000 dataset.

For evaluation, we use mean average precision. We count it as a correct result only if the ratio of intersection over union (IOU) exceeds 50% between the detected bounding box and the ground truth bounding box. Note that we evaluated results regarding not segmentation but only bounding boxes, since UEC-FOOD has no pixel-wise annotation.

Figure 8.4 shows some examples of the detected BB and food regions. The red letters with yellow backgrounds represent food IDs and corresponding output scores from the CNN. Most of the food items were correctly detected. In the top row, "[93] kinpira-style salad" was correctly detected, although it was not annotated in the ground truth data. In the bottom row, "[24] beef noodle" was detected as only half of the ground truth region due to the failure of GrabCut.

Next, we compared three kinds of BP-variant methods which are used for estimating saliency maps. Table 8.2 shows mean average precisions by three methods regarding estimated bounding boxes. Although the results by BP were better than the results by the other methods, the difference was not so large.

We compared our results with the results by RCNN. For RCNN as well as the proposed method, we used the same CNN fine-tuned with the single food images of UEC-FOOD 100. Table 8.3 shows the results. Unexpectedly, the mean AP by RCNN was much lower than the proposed method. Figure 8.5 shows some example results. Compared to the bounding boxes estimated by the proposed methods, RCNN detected too small bounding boxes which cannot be counted as correct bounding boxes. Although our method is based on RCNN, there are some differences in procedure. Especially, RCNN recognizes bounding box proposals while our method recognizes GrabCut results. In general GrabCut results don't include food texture patch such as detected small region on RCNN results since GrabCut expands a seed region with the low level feature. We consider this difference causes that our method superior to RCNN on AP of UEC-FOOD100. This means that CNN trained

with food images recognize food from textures.

8.4.2 Evaluation on Pascal VOC 2007 Detection Task

For more fair comparison with RCNN [100], we also applied our method to Pascal VOC 2007 detection dataset [57]. The PASCAL VOC dataset consists of 20 general object classes. There are also many multi-class object images in Pascal VOC dataset. We used the pre-trained model on PASCAL VOC 2007 included in the RCNN package [2]. In the same way as UEC-FOOD, we compare both performances in mean average precision. The results are shown in Figure 8.4. Our method outperformed RCNN by 4.5 points.

8.4.3 Calorie estimation of UECFOOD100

We estimate a calorie from the segmentation results. Table 8.1 shows food priorities with variation distribution of food area ratio on UEC multiple food datas. We choose a base food class based on these priorities and calculate calories in practice. Figure 8.7 shows some examples of successful calorie estimation results.

Though we computed valid calories from the segmentation results at UECFOOD100, we should report that there are also failure cases. We show failure cases on Figure 8.8. Needless to say, if we choose a mis-segmented-food region as a base area, a whole calorie estimation result will be terrible. Though we consider only food volume for calorie estimating, the fact is clear that there are many factors affect on calories besides food volume. These are subjects for future analysis.

8.4.4 Calorie estimation for FOOD panel

We also tested for evaluation of calorie estimation on the food panel dataset. These datas are not real foods, but veiwing is similar to real food because of the same angle and reproducing actual food size rate. Importantly, each panel has a food calorie data. Hence, we can evaluate calorie estimation performance using these panels. We took 34 pictures using these panels and estimate calories for each one using the same flow to Section 8.4.3.

[2]Downloaded from https://github.com/rbgirshick/rcnn

Figure 8.9 shows results. Note that to segment panel iamges relatively easy more than UECFOOD images due to some factors such as less light reflectance. However there are some failure cases like a thrid example in Figure 8.9. We mapped esitimated calorie and truth data relation in Figure 8.4.4. Mapped points along the red-line mean performance is better. Note that we ignored some outliers which over 5000 Kcal.

Table 8.5 shows numerical results for calorie estimation of some categories including total result. In some food-items, Mean error and mean standard deviation results are very large due to outliers. However mean related error and mean related standard deviation restrict effect of outlier factors. Since we can evaluate the calorie estimation performance of each food-item from these results. Mean related standard deviation of stew is 0.24, while fried chicken is 1.11, hamburger is 1.24. In terms of mean related standard deviation, calorie estimation for a stew is easier than a chicken and a hamburger.

In the evaluation of total calorie estimation, we ignore some outliers in the same flow as graph Figure 8.4.4. We achieved 0.41 in terms of the correlation coefficient for total calorie estimation. Considering the difficulty of calorie estimation only using multiple foods in an image, this result not bad.

8.5 Summary

In this chapter, we proposed a CNN-based food image segmentation which requires no pixel-wise annotation. The proposed method consists of food region proposals by selective search and bounding box clustering, backpropagation based saliency map estimation with the CNN fine-tuned with the UEC-FOOD100 dataset, GrabCut guided by the estimated saliency maps and region integration by non-maximum suppression. In the experiments, the proposed method outperformed RCNN regarding food region detection as well as the PASCAL VOC detection task. We also estimate calories using segmentation results without a non-food specific object. We focus on the food category and food volume as an important calorie decision factors and we achieve it by segmentation. Especially, we compute food calories from only food area ratios by choosing a base food class from multiple foods in an image.

Table 8.1: This Table shows pair of food items and variation values.These pairs are sorted based on variation value.Small variation value mean holding a high priority.

food item	variation distribution
ramen noodle	0.298
beef bowl	0.358
kinpira-style sauteed burdock	0.365
sashimi bowl	0.368
pork miso soup	0.423
fried fish	0.427
fried rice	0.431
pork cutlet on rice	0.444
sirloin cutlet	0.465
jiaozi	0.485
green salad	0.512
potato salad	0.553
beef curry	0.556
egg sunny-side up	0.558
hambarg steak	0.569
grilled pacific saury	0.570
croquette	0.585
rice	0.598
omelet	0.599
miso soup	0.601
grilled salmon	0.604
cold tofu	0.630
sauteed vegetables	0.633
french fries	0.642
natto	0.664
chinese soup	0.670
Japanese tofu and vegeTable chowder	0.676
fried chicken	0.693
tempura bowl	0.694
teriyaki grilled fish	0.716
sweet and sour pork	0.731
ginger pork saute	0.760
mixed rice	0.844
hamburger	0.969

Figure 8.4: The results of food region segmentation for UEC-FOOD100. (1) original food photo, (2) detected BB, (3) estimated food segments, (4) ground truth BB. ([] represents food ID: [01] rice, [05] pork cutlet, [17] humberger, [24] beef noodle, [36] miso soup, [39] oden, [93] kinpira-style salad, [94] rice ball, [98] french fries.)

Table 8.2: Mean average precision over all the 100 categories, 53 categories (more than 10 items of which are included in the test data), and 11 categories (more than 50 items of which are included in the test data).

UEC-FOOD100 mAP	100class (all)	53class (#item $\geq$ 10)	11class (#item $\geq$ 50)
guided back propagation (GBP)	**50.7**	52.5	51.4
deconvolution (deconv)	48.0	54.1	**55.4**
back propagation (BP)	49.9	**55.3**	**55.4**

Table 8.3: The results by RCNN and the proposed methods.

UEC-FOOD100 mAP	100class (all)	53class (#item ≥ 10)	11class (#item ≥ 50)
R-CNN	26.0	21.8	25.7
proposed method	**49.9**	55.3	**55.4**

Table 8.4: The results for the PASCAL VOC 2007 detection dataset.

	aero	bike	bird	boat	btl	bus	car	cat	chair	cow	dTable	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	**54.2**
proposed	81.5	70.2	65.2	39.7	37.8	63.9	83.2	67.8	27.0	65.3	39.5	63.6	63.2	73.2	61.2	37.3	63.5	39.8	70.0	60.8	**58.7**

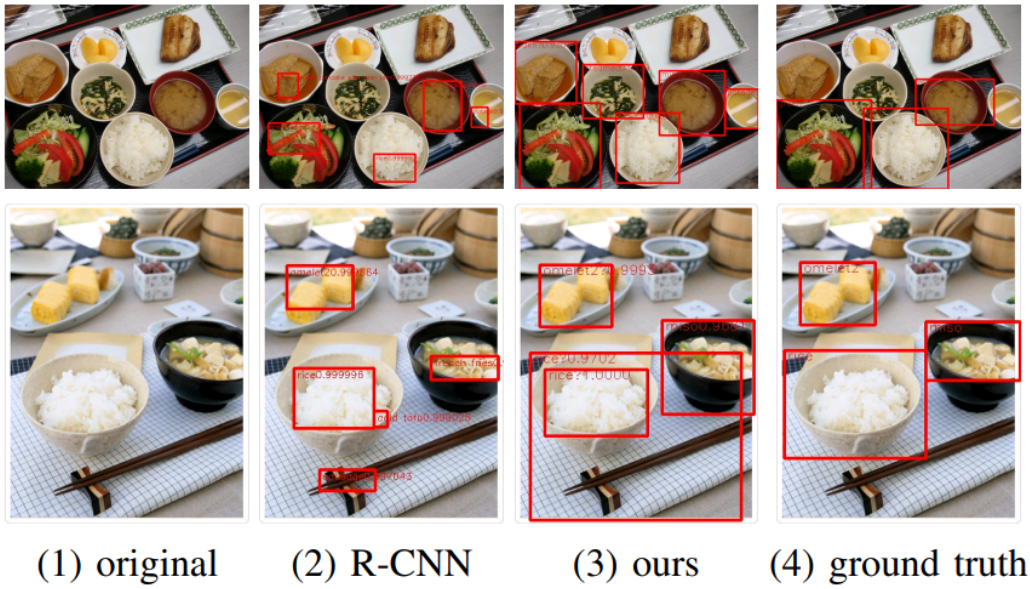

Figure 8.5: Examples of the detection results by R-CNN and the proposed method.

Table 8.5: The Evalution for calorie estimation on food-panel-datas.

Food	Mean error	Mean standard deviation	Mean related error	Mean related standard deviation	Correlation//coefficient
miso-soup	**9.64**	**25.77**	0.43	0.72	-
rice	216.35	362.56	0.19	0.67	-
fried chicken	33.06	373.91	1.77	1.11	-
spaghetti	371.03	405.16	**0.08**	0.64	-
stew	111.72	182.15	0.10	**0.24**	-
hamburger	2129.47	3107.62	0.59	1.24	-
total	237.05	569.50	0.08	0.46	0.41

Figure 8.6: Example for general object segmentation results on PASCAL VOC 2012 (1)original image, (2) segmentation result, (3)label

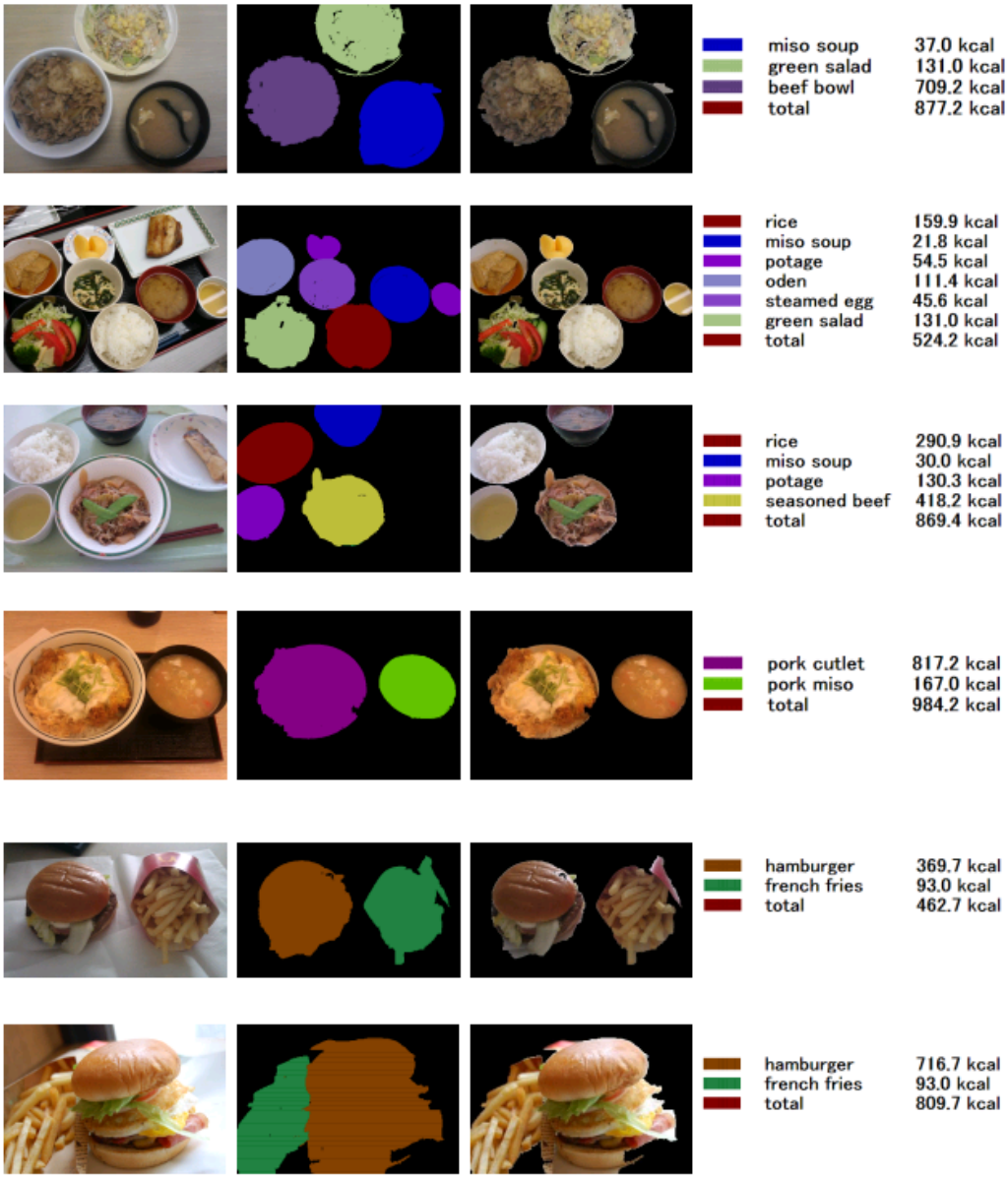

Figure 8.7: Examples for successful results of calorie estimation on UEC-FOOD100. First cols show input images. Second cols show segmentation masks. Third col shows image regions link with the segmentation masks. Forth cols show estimated calories of each food item and whole results. Colors link with the segmentation mask colors.

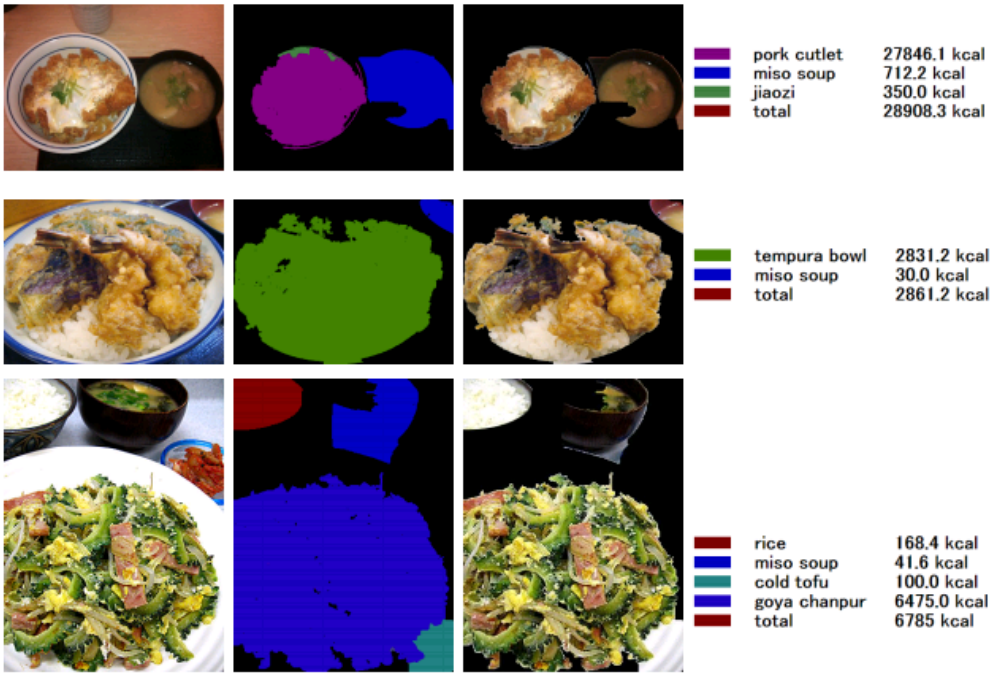

Figure 8.8: Examples for failar cases of calorie estimation on UECFOOD100.

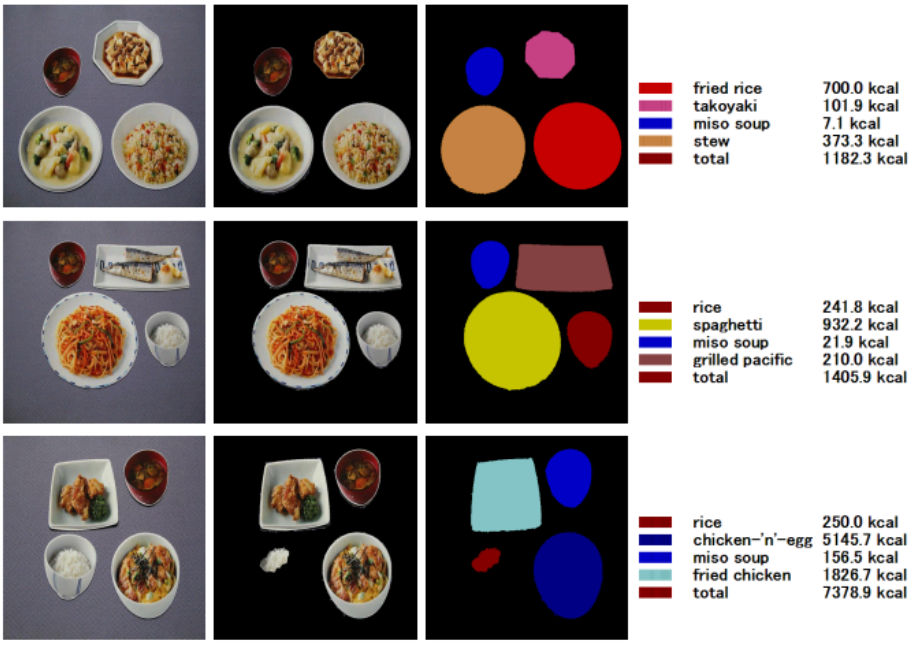

Fig. 9. Examples for calorie estimation on food panel.

Figure 8.9: Examples for calorie estimation on food panel.

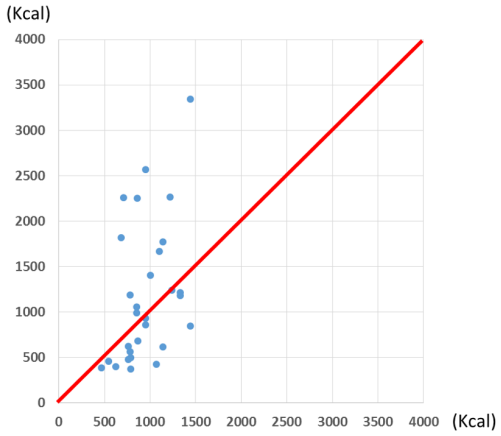

Figure 8.10: Horizontal axis means true calorie. Vertical axis means estimated calorie.

Chapter 9

Visualization based food region proposal for boosting computation

To minimize the annotation costs associated with training semantic segmentation models and object detection models, weakly-supervised detection and weakly-supervised segmentation approaches have been extensively studied. However most of these approaches assume that the domain between training and testing is the same, which at times results in considerable performance drops. For example, if we train an object detection network using only web images showing a large object at the center, it can be difficult for the network to detect multiple small objects. In this chapter, we focus on training a CNN with only web images and achieve object detection in the wild.

A proposal-based approach can address the problem associated with differences in domains because web images are similar to images of the proposal. In both domains, the target object is located at the center of the image and the ratio of the size of the target object to the size of the image is large. Several proposal methods have been proposed to detect regions with high "object-ness."

However, many of these proposals generate a large number of candidates to increase the recall rate. Considering the recent advent of deep CNNs, methods that generate a large number of proposals exhibit problems in terms of processing time for practical use. Therefore, we propose a CNN-based "food-ness" proposal method in this chapter that requires neither pixel-wise annotation nor bounding box annotation. Our method generates proposals

through backpropagation and most of these proposals focus only on food objects. In addition, we can easily control the number of proposals. Through experiments, we trained a network model using only web images and tested the model on the UEC FOOD 100 dataset. We demonstrate that the proposed method achieves high performance compared to traditional proposal methods in terms of the trade-off between accuracy and computational cost.

Therefore, in this chapter, we propose an intermediate approach between the traditional proposal approach and the fully convolutional approach. In particular, we propose a novel proposal method that generates high"food-ness" regions using fully convolutional networks based on the backward approach by training food images gathered from the web.

In particular, we consider "Distinct Class-specific Saliency Maps (DCSM)" 4 to be weakly-supervised detection and segmentation methods. These methods demonstrate high performance in weakly-supervised tasks and can be easily used to adapt to other targets. However DCSM is ineffective for Webly supervised methods because of the change in domain at the time of training and testing. In Webly supervised approaches, most training images are single labeled images and we assume that the targets are multiple-food images, consisting of multiple foods in the test phase. The differences in the domain can cause considerable performance drops. However, we determined that we can obtain the rough food regions from the outputs of the DCSM, even though it is difficult to directly obtain detailed food regions and the correct class of food for the region. We consider the rough food regions to be a type of proposal for food objects and we define "food-ness" as a representation that reflects how likely a pixel belongs to a region of any food category. In this chapter, we used "food-ness" as a proposal for foods and we apply it to a proposal-based method for foods by following traditional detection or segmentation methods such as RCNN and SDS. For this proposal method, we primarily discuss the computational costs and the methods of generating a small number of effective region candidates.

We summarize the contributions as below:

- We achieved Webly supervised food-detection and food-segmentation for the first time.
- We proposed a novel proposal method for food images.

9.1 Proposed Method

We propose a new method of generating "food-ness" regions with weakly supervised annotation. Our method is based on distinct class-specific saliency maps (DCSM) 4, which is an extension of Simonyan et al.[2]. In this section, we discuss the DCSM and the manner in which DCSM has been adopted for "food-ness" proposal.

9.1.1 Overall Architecture

We follow traditional detection methods by using proposals. We first generate proposals based on DCSM. We then identify each candidate region. Finally, we unify overlapped candidates by Non Maximum Suppression(NMS). In this study, we prepare two CNNs for proposal and recognition. We illustrate an overview in Figure 9.1. Details of the proposed method process is as follows:

- Recognize an image.
- Sort each food class based on the softmax output.
- Backpropagate upper rank class scores.
- Subtract each class derivative value.
- Obtain "food-ness" proposals.
- Recognize each "food-ness" candidate.
- Unify overlapped candidates by NMS.

9.1.2 DCSM

In [2], the authors considered the derivatives of the class score with respect to the input image as class saliency maps. However, the position of an input image is the furthest from the class score output on the deep CNN, which sometimes causes weakening or vanishing of gradients. Instead of the derivatives of the class score with respect to the input image, Shimoda et al. 4 used the derivatives with respect to feature maps of the relatively upper intermediate layers that are expected to retain more high-level semantic information. In addition, they applied some techniques that are known to

be effective in semantic segmentation through a backward approach. They selected the maximum absolute values of the derivatives with respect to the feature maps at each location of the feature maps across all the kernels and up-sampled them with bi-linear interpolation so that their size becomes the same as an input image.

The class score derivative v_i^c of a feature map layer is the derivative of the class score Sc with respect to the layerL_i at the point (activation signal) L_i^0:

$$v_i^c = \left.\frac{\partial S_c}{\partial L_i}\right|_{L_i^0} \tag{9.1}$$

v_i^c can be computed by back-propagation. After obtaining v_i^c, Shimoda et al. up-sampled it to w_i^c through bi-linear interpolation so that the size of a 2-D map of v_i^c becomes the same as an input image. Next, the class saliency map $M_i^c \in \mathcal{R}^{m \times n}$ is computed as

$$M_{i,x,y}^c = \max_{k_i} |w_{i,h_i(x,y,k)}^c|, \tag{9.2}$$

where $h_i(x, y, k)$ is the index of the element of w_i^c.

The saliency maps of two or more different classes tend to be similar, particularly at the image level. The saliency maps by[2] are likely to correspond to foreground regions rather than specific class regions. To address this, Shimoda et al. 4 proposed to subtracte saliency maps of the other candidate classes from the saliency maps of the target class to different target objects from other objects. They selected several candidate classes with a pre-defined threshold and a pre-defined minimum number.

The improved class saliency maps with respect to class c, $\tilde{M}_i^c$, are represented as:

$$\tilde{M}_{i,x,y}^c = \sum_{c' \in candidates} \max\left(M_{i,x,y}^c - M_{i,x,y}^{c'}, 0\right)[c \neq c'], \tag{9.3}$$

where *candidates* is a set of selected candidate classes. Subtraction of saliency maps resolved the overlapped regions among the maps of the different classes.

Shimoda et al. 4 used fully convolutional networks (FCN) that accept arbitrary-sized inputs for multi-scale generation of class saliency maps. If an input image that is larger than the one used in the original CNN is given to the fully-convolutional CNN, class score maps represented as $h \times w \times C$ are outputted, where C is the number of classes, and h and w are larger than 1.

To obtain CNN derivatives with respect to enlarged feature maps, Shimoda et al. 4 simply back-propagated the target class score map defined as$S_c(:,:,c) = 1$ (in the MATLAB notation) with 0 for all other elements, where c is the target class index.

The final class saliency map $\hat{M}^c$ averaged over the layers and the scales is obtained as follows:

$$\hat{M}^c_{x,y} = \frac{1}{|S||L|} \sum_{j \in S} \sum_{i \in L} \tanh(\alpha \tilde{M}^c_{j,i,x,y}), \tag{9.4}$$

where L is a set of the layers for which saliency maps are extracted, S is a set of the scale ratios, and α is a constant which we set to 3 in the experiments. Note that we assume the size of $\tilde{M}_{j,i}$ for all the layers are normalized to the same size as an input image before calculation of Equation 9.4.

In [3], guided back-propagation (GBP) [3] was adopted as a back-propagation method instead of normal back-propagation (BP) used in [2]. The difference between the two methods is the backward computation through ReLU. GBP can visualize saliency maps with fewer noise components than normal BP by back-propagating only the positive values of CNN derivatives through ReLU [3].

9.1.3 "Food-ness" Proposal

In this chapter, we focus on training models with single-food images and on testing multiple-foods images. In general, domain changes from training time to testing time results in performance degradation. This problem is referred to as one of the cross-domain problems or the domain adaptation problems. Using the DCSM, this problem was also observed and accuracy degraded significantly. We illustrate our situation using this domain adaptation problem and an example at Figure 9.2 in food images.

In this study, we avoid this domain adaptation problem using region proposals. Proposal methods generate object region candidates and these candidates must include target objects. When recognizing target objects in the candidates, we obtain better results than in the case of recognizing raw images without proposals. Because, in our situation, test images include multiple food images, some candidate regions can be considered single food images. Therefore, the condition with some candidate regions is closer to the training condition than the raw test images condition.

RCNN [100] and SDS [18] are typical methods of detection and segmentation using proposals based on CNN. They use selective search[19] and MCG [65] as proposal methods. These proposal methods also typically generate a considerable number of candidates, approximately 2000 with local features. A considerable number of candidates rise recall but pay off computational costs. We consider the number of candidates, approximately 2000, to be too large and there can be several inefficient processes for food recognition. Therefore, we propose a novel proposal method for foods with CNN.

"Objectness" is a value that reflects the likelihood that a region or bounding box in an image covers an object of any category. In this study, we define "food-ness" as a representation that reflects the likelihood that a pixel belongs to a region of any food category. In this study, we adapt DCSM for calculating "food-ness."

The original DCSM approach is ineffective because of the problem of domain changes as we mentioned above. In fact, the estimated regions by DCSM trained with only Web images are not precise. However, we observed that most regions belonged to any food items in an image. Interestingly, the estimated regions for food classes that are not included in a given image still belong to other existing objects, and some regions fit food regions as shown in Figure 9.3. This means that CNN trained with different domain images could not precisely transfer knowledge related to the category of food but could learn rough food conception.

In practice, to adapt DCSM for "food-ness" we increase the number of *candidates* in Equation 9.3. It must be noted that we do not aggregate multi-input-scale results because of increasing computational costs. We obtain the probability maps for each signal of a class using backpropagation as follows:

$$\mathrm{P^c} = \frac{1}{|\mathrm{L}|} \sum_{\mathrm{i} \in \mathrm{L}} \tanh(\alpha \mathrm{M_i^c}), \tag{9.5}$$

where P^c denotes probability maps such as saliency maps. We convert the probability maps P^c to masks M^c through thresholding. In this study, we set the threshold to 0.5. When the mask M^c contains multiple food items, the probability maps often include several peaks. Therefore, to obtain better proposals, we divide each mask M^c into several masks M_k^c by separating the isolated regions using a binary tracing method.$k \in \{1, 2, \cdots, K\}$ represents the elements of regions and K is the number of regions. For binary tracing, we used *bwconncomp*, which is a MATLAB function. We finally integrated

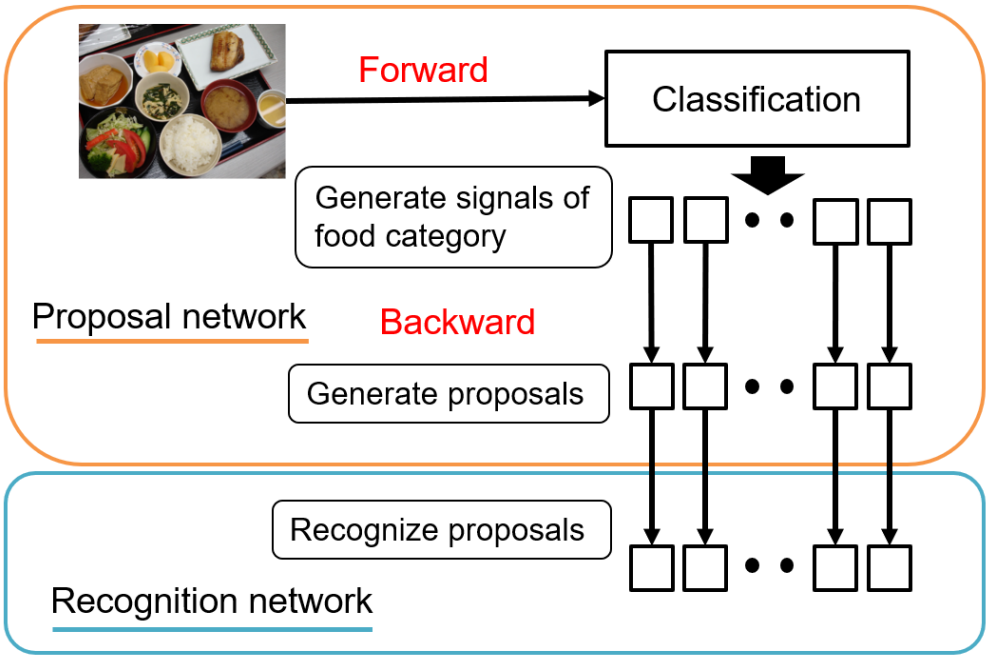

Figure 9.1: Processing flow of our method.

the masks by ignoring the category of signals for backpropagation and we used the integrated masks $\hat{M}_{k'}, (K' = C \times K)$ as food-ness proposal. To summarize this, we increase the number of candidate classes in the DCSM method and obtain regions from the output probabilities with DCSM. We can increase the number of candidate classes as far as the maximum target class number, which in our case is 100 for the UEC-FOOD100 dataset. We will discuss the manner in which the number of classes in Section 9.2.2 are chosen. For each input image $x \in \mathcal{X}$, we compute the proposals $\hat{M}_{k'}$ using the above process. We then obtain bounding boxes $\hat{B}_{k'}$ by extracting the maximum and minimum values of the coordinate from the pixels, which belong to the food region on each mask $\hat{M}_{k'}$. For each bounding box $\hat{B}_{k'}$, we cropped the images $x^p_{k'}$and identified these cropped images using recognition networks. The training of recognition networks is independent of the proposal network. We train the proposal network and recognition network separately. Details of the training are presented in Section 9.2.

Figure 9.2: Example of our cross domain situation.

9.2 CNN training

In this study, we adopt VGG16 as a base convolutional network for fine-tuning food images. Although there are common factors, we separate the proposal network and recognition network because of differences in applications. We fine-tune VGG16 as a proposal network with a fully convolutional technique. We also fine-tune VGG16 for recognition networks in a traditional way. In this section, we present the details concerning these two networks.

9.2.1 Proposal Network

As an off-the-shelf basic CNN architecture, we use the VGG-16 [59] pre-trained with 1000-class ILSVRC datasets. In our framework, we fine-tune a CNN with training images with only image-level annotation. Fully convolutional networks (FCN) that accept arbitrary-sized inputs have been recently used in studies on CNN-based detection and segmentation such as [67] and[20]. The fully connected layers in these studies with n units were replaced with equivalent convolutional layers having n 1×1 filters. Following these studies, we introduce FCN for multi-scale generation of class saliency

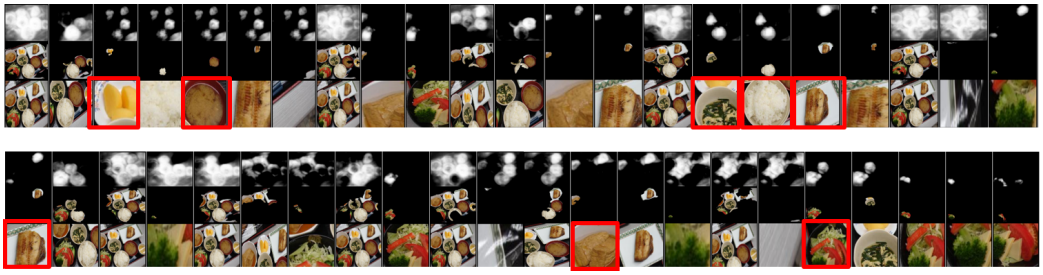

Figure 9.3: Proposal results . The first row presents the saliency obtained from DCSM. The second row indicates the regions obtained from saliency maps. The third row indicates the bounding boxes that we recognize. The red rectangle indicates a good candidate.

maps. When training, we insert global max pooling before the final loss function layer to handle input images that are larger than the images used for pre-training of the VGG-16. Global max pooling is an operation that has been adopted in several weakly-supervised segmentation methods. The purpose of this operation is to convert the last output to a vector from a matrix. Therefore, we can train FCN with usual image-level-label and soft-max loss.

In particular, we replace a fully connected layer with a convolution layer for the VGG16-model and train the network on the UECFOOD-100 dataset, which consists of 100 types of food classes with global max pooling.

9.2.2 Recognition Network

For recognition, although we change only the last layer for food category outputs, we prepare additional categories for training. The purpose of a recognition network is to discriminate candidates obtained from the proposal network. The conditions for recognizing candidates vary from the training phase in terms of including non-target-category-object images and small-food-patch images. In RCNN and SDS, they consider only non-target-category-object images as the background so that RCNN and SDS can be tested on a general object detection dataset. However, food recognition is different for general object recognition. Food recognition has the similarity to the texture recognition, namely, food patches can be discriminated as food with a high score by CNN. For example, in the case of dog recognition with CNN, the recognition results for the proposals of legs and skin will include low scores in

dog probability, while, in the case of food recognition, the patches of rice images will indicate high scores in rice probability. To sum up, CNN cannot discriminate general objects with limited parts but can discriminate foods with minimum patch information. Therefore we create additional classes for food patch class.

Furthermore we add low-resolution images because we determined that a low-resolution image was discriminated as food patch category. We assume that this is the reason for which a small-food-patch image tends to be a low-resolution image. Therefore, we add low-resolution images to each food class. Our intuition is that if we consider low-resolution images to be training images, the low-resolution images will not be recognized as small-food-patch images.

We augment the training images for cropped images and expand the category to 202 from 101 to address some problems for each candidate recognition in food images. Practically, we cropped three images from each training image as a food path using random positions with random sizes. The minimum size of the cropped image is 50 and the maximum size is 150. It must be noted that the original image size is 256, i.e. the rate of each cropped image size for the original image is approximately 0.2 and 0.6. We also prepare three images as low-resolution images by down-sampling and rescaling. We randomly defined down-sample sizes. The minimum downscaled size is 10 and the maximum size is 256, which is equal to the original image size. We finally obtain augmented training images that are seven times larger than the original training images.

9.3 Experiments

In the experiments, we used the UEC-FOOD100 dataset [81] and web food images. The UEC-FOOD100 dataset [81] consist of 100 class food categories and each category includes 100 images. It should be noted that each food item is an annotated bounding box. On the other hand, although the web food images have the same category as in the UEC-FOOD100 dataset, each category includes 1000 images without bounding box annotation. Most of these web food images are obtained from twitter streams and some images are obtained from the Bing API. We use multiple-food images from the UEC-FOOD100 dataset as a test dataset for object detection. All detection evaluations based on mean average precision are also considered for Pascal

VOC detection evaluation.

9.3.1 Food Detection Evaluation

We prepared two datasets, one dataset consist of UECFOOD-100 and web Images. Another dataset consist of only web images.

9.3.1.1 Additional Classes for Recognition Network

We first evaluate three cases of recognition networks with two datasets using a fixed proposal network setting. Table 9.1 presents the average precision (AP) of the three models trained under different conditions with two training data."Foodness 2" demonstrated higher performance than "Foodness 1". This means that adding a small patch class is effective. On the other hand, "Foodness" 3 achieved better results than Foodness 2". We can observe that adding low-resolution images is also effective for the recognition network."Foodness 4", "Foodness 5" and "Foodness 6" are trained with only Web images. The AP of "Foodness 6" is higher than "Foodness 4" and "Foodness 5". In Webly supervised, additional classes are also effective."Foodness 6" exhibits a drop in AP compared with "Foodness 3";while overcoming the AP of "Foodness 2". Based on the results above, we can state that additional classes are effective and Webly supervised learning possesses reasonable capabilities.

Table 9.1: Mean average precision over all the 100 categories, 53 categories (more than 10 items of which are included in the test data), and 11 categories (more than 50 items of which are included in the test data) for the results in the different conditions and models.

method	small-patch class	low-resolution images	training with only web images	100class (all)	53class (#item ≥ 10)	11class (#item ≥ 50)
"Foodness 1"	-	-	-	30.0	29.3	31.9
"Foodness 2"	✓	-	-	33.7	39.0	33.6
"Foodness 3"	✓	✓	-	39.5	46.0	38.9
"Foodness 4"	-	-	✓	33.5	35.1	33.3
"Foodness 5"	✓	-	✓	32.2	34.8	31.8
"Foodness 6"	✓	✓	✓	36.4	39.9	36.3

9.3.1.2 Global Pooling for Proposal Network

We then compare two general global-pooling operations, global average pooling, and global max pooling. Table 9.2 presents a comparison of final pooling operations for two datasets.

Table 9.2: Comparison of global pooling operations for "food-ness".

method	training with only web images	100class (all)	53class (#item ≥ 10)	11class (#item ≥ 50)
"Foodness" (average pooling)	-	39.5	46.0	38.9
"Foodness" (average pooling)	✓	36.4	39.9	36.3
"Foodness" (max pooling)	-	39.9	48.3	37.6
"Foodness" (max pooling)	✓	38.9	42.5	38.1

9.3.1.3 Comparison with Other Traditional Proposal Methods

Next, we compare the quality of our proposal method with that of other traditional proposal methods. We evaluate our methods in terms of mean AP and speed factors. We prepare two traditional proposal methods as baselines. Selective search (SS) [19] is a bounding-box proposal method and Multiscale Combinatorial Grouping (MCG) [65] involves a segmentation region proposal method. Both methods generate a large number candidates, approximately 2000. To assess our proposal quality, we changed the candidate class number. Small candidate class results in smaller computational costs so that the time of backward computation can be reduced. Table 9.3 presents the comparison results. It must be noted that recognition speed includes theoretical values computed from candidate numbers and the computational cost of an image. AP of "Foodness" with 30 candidate classes outperforms SS [19] and MCG [65] even though it has 40 times lesser number of candidates. In addition, even if we reduced the candidate class number, the mean AP is still held by 30%. This shows that our proposal exhibits sufficient quality for "food-ness" detection.

9.4 Summary

We proposed a CNN-based "food-ness" proposal method that requires no pixel-wise annotation even in the case of bounding box annotation. We fo-

Table 9.3: Comparison with other traditional proposal method.

method	100class (all)	53class (#item ≥ 10)	11class (#item ≥ 50)	proposal speed[s]	Recognition speed for candidates[s]
Selective Search [19]	38.3	39.1	35.7	7.6	35.0
Multiscale Combinatorial Grouping [65]	33.9	43.7	33.4	2.5	35.0
"Foodness" with 10 candidate classes	33.1	33.0	33.2	0.5	1.1
"Foodness" with 20 candidate classes	36.5	40.1	37.7	1.0	2.6
"Foodness" with 30 candidate classes	38.9	42.5	38.1	1.4	3.8

cused on an intermediate approach involving traditional proposal approaches and fully convolutional approaches. In particular, we proposed a novel proposal method that generates "food-ness" regions through a fully convolutional network-based backward approach by training web food images. Therefore, we achieved a reduction in computational costs and ensured quality food detection.

Figure 9.4: Examples of results. Left images are input images. Center images are detection results. Right images are ground truth images.

Chapter 10

Weakly-supervised estimation of food plate regions

Though recent weakly-supervised segmentation methods achieve high accuracy on a benchmark of weakly-supervised segmentation of general objects, we should consider the difference between general objects and food objects to apply the methods for food images. For example, most of food images include plate regions and it is important that whether or not food segmentation should include plate regions. The solution will vary depending on applications. While, in the case of calorie estimation, it is desirable that the plate regions are excluded from food segmentation, if the aim of food segmentation is inpainting it would be desirable that the plate regions are included in food segmentation. If the plate regions can be inferred, either case can be accommodated. In addition, the information on the plate regions may be beneficial for the refinement of food segmentation. In this chapter, we propose a novel method to synthesize plate segmentation masks without any pixel-wise annotation and we utilize the plate segmentation for improvement of the weakly-supervised food segmentation. In Fig.10.1, we show the motivation and the concept of the proposed approach.

To deduce plate regions without pixel-wise annotation, we train not only a food category classifier but also a food/non-food classifier. In the visualization of the food category classifier, plate regions will not respond because plates are included in most of food images. Therefore, plate regions are not expected to contribute to the recognition of the food category. On the other hand, in the visualization of the food/non-food classifier, plate regions will respond because plates are not included in most of non-food images. Thus, the

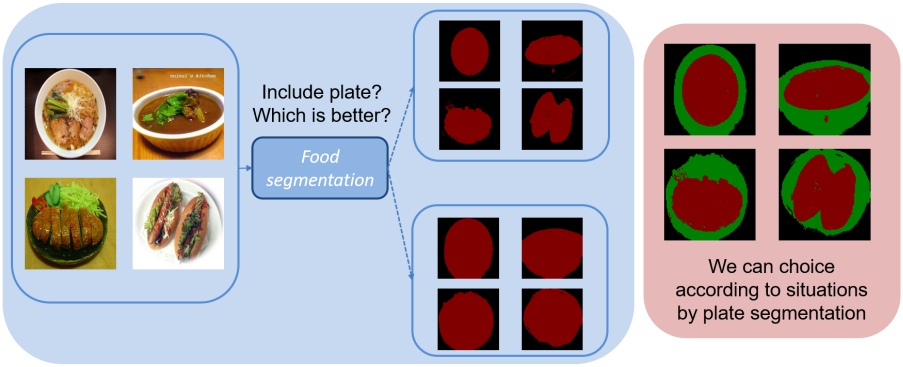

Figure 10.1: The motivation and the concept of our proposed approach.

presence of plate regions is expected to assist recognition by the food/non-food classifier. As we stated, there is a difference in the visualization of plate regions between the food category classifier and the food/non-food classifier. We utilize the difference between the visualization of the two classifiers for prediction of plate regions, and synthesize plate segmentation masks. In this chapter, we also propose approaches to boost weakly-supervised food segmentation accuracy using the plate segmentation masks. Especially, we make consistency between a food segmentation model and a plate segmentation model in food regions and background regions. We demonstrate that the proposed approaches can improve a generic weakly-supervised segmentation method in the food domain, and we assess the quality of the plate segmentation by the improvement of the weakly-supervised segmentation method, which utilizes inference of the plate segmentation. To the best of our knowledge and belief, both of the works are the first attempt to extract plate regions from food images without any pixel-wise annotation using visualization techniques, and to boost the accuracy of food segmentation using plate segmentation.

In order to deduce plate areas without pixel-wise annotation, in this study, we train not only food class classifiers but also food/non-food classifiers. In the recognition of food/non-food, plate areas will respond because those have a strong co-occurrence with foods. On the other hand, in the recognition result of the food category, plates are included in most of food images, so the contribution to the recognition of the food category is not large. That is, in the visualization of the food class classifier and the food/non-food classifier

is different and the difference may have correlation with plate areas. In this study, the difference between the visualization results of the dish area in these two classifiers is used to infer the dish area without the area level annotation.

10.1 Plate segmentation with visualization of food classifiers

In this chapter, we synthesize plate segmentation masks for learning a plate segmentation model that infers plate regions of food images. To generate plate regions, we use visualization of a food category classifier and a food/non-food classifier. Fig.10.2 shows the illustration on the idea of the proposed approach.

We assume that $v_L = CAM(x; \theta_L) \in \mathbb{R}^{C \times H \times W}$ is a visualization of the C-class food classifier for input image x generated by Class Activation Mapping (CAM) [36]. In the similar manner, the visualization of the food/non-food classifier is represented by $v_F = CAM(x; \theta_F) \in \mathbb{R}^{2 \times H \times W}$, where θ_L and θ_F are the parameters for the classifiers. Both v_F and v_L should respond to food regions. However, the results of visualization are expected to be different. In particular, while the visualization of the food/non-food classifier returns clear responses in plate regions, the visualization of the food category classifier returns weak responses in plate regions. This is because the plate regions have strong co-occurrence with food images. In this chapter, we assume that the difference in v_F and v_L corresponds to plate regions and we synthesize plate segmentation masks by utilizing the difference.

Here, we denote the steps on synthesizing of plate segmentation masks. First, from v_F, we obtain binary segmentation masks $m_{F,cam}$ whose pixels represent belonging to foods or non-food objects. Secondly, we obtain segmentation masks $m^y_{L,cam}$ for category labels y assigned to images from v_L. If $m_{F,cam}$ and $m^y_{L,cam}$ are able to be extracted correctly, the difference in the masks would be plate regions based on the above assumption. However, the visualization of food category classifier is unreliable because of the difficulty of food classification. Therefore, in this work, in addition to the visualization for the class label, we define unreliable regions obtained from the visualization of the top K classes of the recognition result. In practice, we define unreliable regions $m^{r^K}_{L,cam}$ whose pixels do not overlap with $m^y_{L,cam}$, and just ignore the pixels when training of a plate segmentation model. We set K to

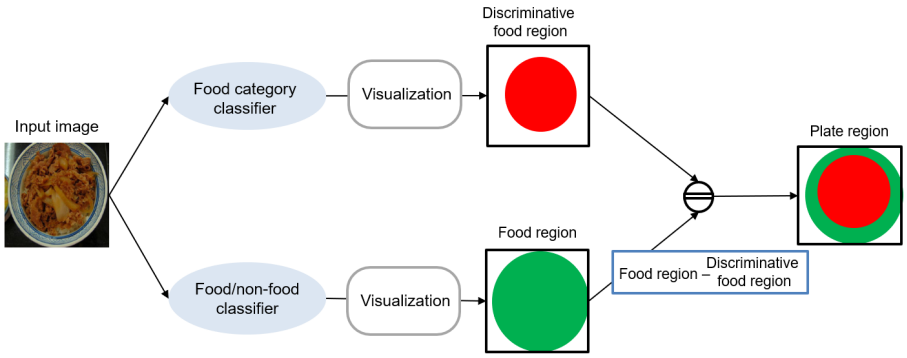

Figure 10.2: An illustration of the proposed approach for synthesizing plate segmentation masks using the visualization technique.

30. We empirically decided this value. We denote the segmentation masks synthesized by the above processing as $m_{P,cam}$. Here, we define a set of pixels for $m_{P,cam}$ as $S_{P,cam}$. This set can be represented by $S_{P,cam} = S^{fg}_{F,cam} - S^{fg}_{L,cam}$, where $S^{fg}_{L,cam}$ is a set of the foreground of the categorical food regions and $S^{fg}_{F,cam}$ is a set of the foreground of the whole food regions. We train the plate segmentation model by the synthesized ternary masks $m_{P,cam}$, which category consists of background, plate regions and food regions. The loss of the plate segmentation model is as follows:

$$\mathcal{L}_{plate} = -\frac{1}{\sum_{k=(0,1,2)} |S^k_{P,cam}|} \sum_{k=(0,1,2)} \sum_{u \in S^k_{P,cam}} \log(h^k_u(x;\theta_P)), \tag{10.1}$$

where h^k_u is conditional probability of observing any label k at any location u. $S_{P,k}$ is a set of pixels for a class k of the mask $m_{P,cam}$. We apply CRF [38] to the probability map of the plate segmentation model and used the CRF applied results as the final plate segmentation $m_{P,out}$.

10.2 Improving weakly-supervised food segmentation using plate segmentation

In general, the inside of plate regions are food regions and the outside of plate regions are non-food regions. In this research, we aim to improve the accu-

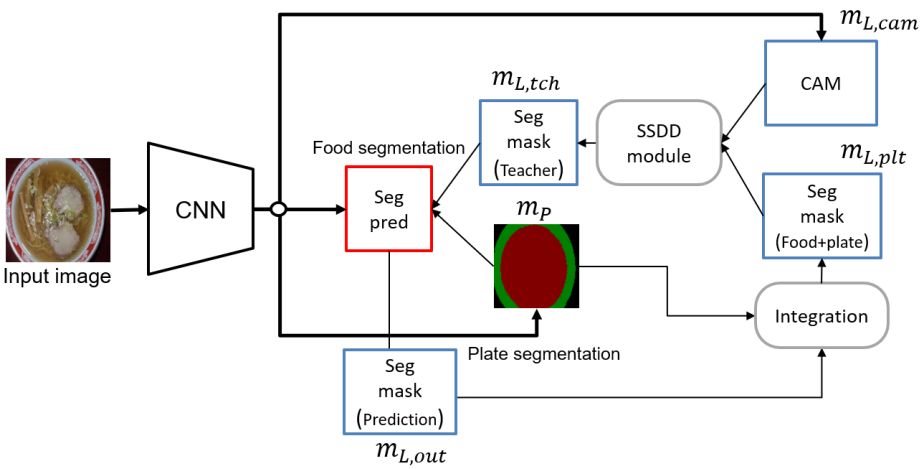

Figure 10.3: An overview of the proposed method for refinement of weakly-supervised food segmentation methods.

racy of weakly-supervised food segmentation by utilizing the relationship between the plate regions and the food regions. To perform weakly-supervised segmentation, we use a method that utilizes Self-Supervised Difference Detection (SSDD) [11]. To improve this further, we propose a new approach which utilizes estimated plate regions. In this section, we describe the details of the approach for making consistency between a food segmentation model and a plate segmentation model. Fig.10.3 shows an overview of the proposed approach.

10.2.1 Self-Supervised Difference Detection (SSDD) module

In this chapter, we use SSDD [11] as a base weakly-supervised segmentation method that integrates two candidate segmentation masks using difference detection. The method proposed a SSDD module, which takes two segmentation masks as inputs and outputs one integrated mask. To be concrete, here, we denote the two segmentation masks as m^K and m^A that has a role of *knowledge* and *advise*, respectively. The module synthesizes a new segmentation mask m^D by integration of m^K and m^A using inference of difference detection. Difference detection is a task to estimate differences of two segmentation mask. A mask for the difference $M^{K,A} \in \mathbb{R}^{H \times W}$ is defined as

following:

$$M_u^{K,A} = \begin{cases} 1 & \text{if } (m_u^K = m_u^A) \\ 0 & \text{if } (m_u^K \neq m_u^A), \end{cases} \tag{10.2}$$

where $u \in \{1, 2, .., n\}$ indicates a location of pixels, and n is the number of pixels. In the module, we use the Difference Detection network for inference of the difference, $\text{DDnet}(e^h(x;\theta_e), e^l(x;\theta_e), \hat{m}; \theta_d), d \in \mathbb{R}^{H \times W}$, where $\hat{m}$ is a one-hot tensor with the same number of channels to the target class number, θ_d is parameters of DD-Net and $e^h(x;\theta_e)$ is high level features and $e^l(x;\theta_e)$ is low level features extracted from a backbone network such as ResNet. DD-Net takes either of segmentation mask as an input, and outputs the estimation of the difference. We calculate a confidence score $w_u \in \mathbb{R}$ from inferences of the DD-Net d^K and d^A for the masks m^K and m^A:

$$w_u = d_u^K - d_u^A + bias_u, \tag{10.3}$$

where *bias* is a hyper parameter for a border of the selection. The refined masks m^D obtained from m^K and m^A are defined by the following expression.

$$m_u^D = \begin{cases} m_u^A & \text{if } (w_u \geq 0) \\ m_u^K & \text{if } (w_u < 0) \end{cases} \tag{10.4}$$

In this chapter, we use mask of CAM $m_{L,cam}$ as *knowledge* and a synthesized mask using the plate segmentation model $m_{L,plt}$ as *advice*. From these masks, we generate $m^{L,tch}$ and use it for the training of a segmentation model. We describe the detail of $m_{L,plt}$ in the next section.

10.2.2 Constrain of food regions by plate regions

In standard weakly-supervised food segmentation methods, the food and plate regions may be mixed and it would cause problems in some food-specific applications. In this study, to prevent this we make consistency between the food segmentation model and the plate segmentation model in the food regions. As we stated in Section 10.2.1, we integrate two segmentation masks $m_{L,cam}$ and $m_{L,plt}$ using the SSDD module. Since the accuracy of the integrated segmentation mask $m_{L,tch}$ depends on the accuracy of the two segmentation masks used for the inputs, the improvement of these inputs would lead better accuracy. Here, we refine the one of the input segmentation mask, which has a role of *advice*. Specifically, we refine the outputs of the

food segmentation model $m_{L,out}$ using the outputs of the plate segmentation model $m_{P,out}$, and generate a mask $m_{L,plt}$. To avoid mixing of the food regions and the plate regions we constrain the food regions by below processing:

$$m_{L,plt} = \begin{cases} m_{L,out} & \text{if } (m_{P,out} = food\ class) \\ BG\ class & \text{if } (m_{P,out} = BG\ or\ plate\ class) \end{cases} \tag{10.5}$$

It is expected that the outputs near by the boundary in food regions and plate regions would be refined by this processing.

10.2.3 Penalizing background prediction using Plate segmentation

Since food segmentation is a kind of fine-grained classification, the degree of difficulty is high compared to general object segmentation. Actually, the food segmentation model tends to output background class in regions that are difficult to inference an appropriate category. Therefore, in this section, we limit the outputs of background by making consistency in the inference of the food segmentation model and the plate segmentation model. To limit the outputs of background, we constrain the outputs of the food segmentation model on the background class using a penalty loss. The penalty loss minimizes the cross entropy loss for the inverse conditional probability on pixels that belongs to inconsistency regions between the food segmentation model and the plate segmentation model. We denote the outputs of the food segmentation model as $h(;\theta_s)$ and a set of the pixels that are classified as food regions by the plate segmentation model as $S^{food}_{P,out}$. We define penalty loss for the background class as following:

$$\mathcal{L}_{penalty} = -\frac{1}{|S^{food}_{P,out}|} \sum_{u \in S^{food}_{P,out}} \log(softmax(-h^{bg}_u(x;\theta_{seg}))), \tag{10.6}$$

where $h^{bg}_u(x;\theta_s)$ is conditional probability maps of *background* class.

10.2.4 Final loss for the food semantic segmentation model

Here, we explain about a final loss function for training the food segmentation model. The parameters θ_{seg} of the food segmentation model are trained using

the outputs of the SSDD module $m_{L,tch}$ by below equation:

$$\mathcal{L}_{main} = -\frac{1}{\sum_{k\in\hat{y}} |S^k_{L,tch}|} \sum_{k\in\hat{y}} \sum_{u\in S^k_{L,tch}} \log(h^k_u(x;\theta_{seg})). \tag{10.7}$$

In addition, we also use the loss of $\mathcal{L}_{penalty}$ we stated in Section 10.2.3 for training the segmentation model. The final loss of the segmentation model is as following:

$$\mathcal{L}_{seg} = \mathcal{L}_{main} + 0.1\mathcal{L}_{penalty} + \mathcal{L}_{plate}. \tag{10.8}$$

We empirically decided the coefficient of the $\mathcal{L}_{penalty}$.

10.3 Experiments

In the experiments, we used the UEC-FOOD100 dataset [81]. The UEC-FOOD100 dataset [81] consists of 100 class food categories, and each category includes 100 images. Each food item has bounding box annotation, although they have no annotation for segmentation masks. Then, we add new semantic segmentation masks to 10% of UEC-FOOD100 dataset, and used them for the evaluation of weakly-supervised segmentation. In addition, we have collected 8155 non-food images from the Web and Twitter, and we use them for the training of the food/non-food classifier. We train the proposed model using only image-level labels. The training data does not include bounding information. For training of the classifier models and food segmentation model, we used the 90% of the UEC-FOOD100 dataset.

We evaluate the accuracy of the weakly-supervised segmentation using mean Intersection over Union (mIoU) and Pixel accuracy (Pix acc). mIoU is a standard measurement for semantic segmentation that evaluates the overlap and the union in inference and ground truth. Pix acc is a simpler measurement that is the accuracy for the all pixels.

10.3.1 Implementation details

As semantic segmentation model we used a ResNet-38 model, which is the same architecture used in [11]. The input image size is 448x448 for training and test images and the output feature map size before upsampling is 56x56. These feature map sizes are adjusted to 112 by 112 using simple linear interpolation. Before training of the segmentation model, we trained the food

category classifiers with initialization using a pre-trained model of ImageNet. After training of the food category classifiers, we initialized the parameters of backbone models with the food category classifier. The backbone network of the food segmentation model, plate segmentation model and food classifier models are shared, and we trained them in an end-to-end manner. Note that we also continued training of the classifier models. We set an initial learning rate to 1e-3 (1e-2 for initialization without the pre-trained model) and we decreased learning rate with cosine warm up [75]. The batch size for training is 2. For data augmentation and inference technique, we followed the paper [11]. We implemented the proposed method using PyTorch.

10.3.2 Qualitative result of plate segmentation and discussion

In this work, we propose a method to synthesize plate segmentation masks of food images without pixel-wise annotation and we train a plate segmentation model with the synthesized masks. Fig.10.4 shows some successful examples of plate segmentation. The proposed plate segmentation model excels on inference of plates that have the round shape, but, in several cases, the model can also successfully infer plate regions whose shape is not round such as the case of the middle row in Fig.10.4. This indicates that the proposed method infers various types of plate regions and the inference does not fall trivial solutions. Fig.10.5 shows some failure cases. While the proposed plate segmentation model can predict the boundaries between food regions and plate regions, it often fails to capture boundaries between plate regions and background regions. The proposed plate segmentation model also goes wrong on inference for big plates that extend toward the outside of the image such as the example of the bottom of the left in Fig.10.5. We consider that both of the failure cases are caused by limitations of visualization, that is the whole plate regions do not contribute to the recognition of the food/non-food classifier in these cases. There is also another problem in the plate segmentation model, the plate segmentation model attempts to predict plate regions if there are no plates in images. These problems do not harm the accuracy of weakly-supervised segmentation, however, it would be problems on some other applications. There is still room for improvement in this approach.

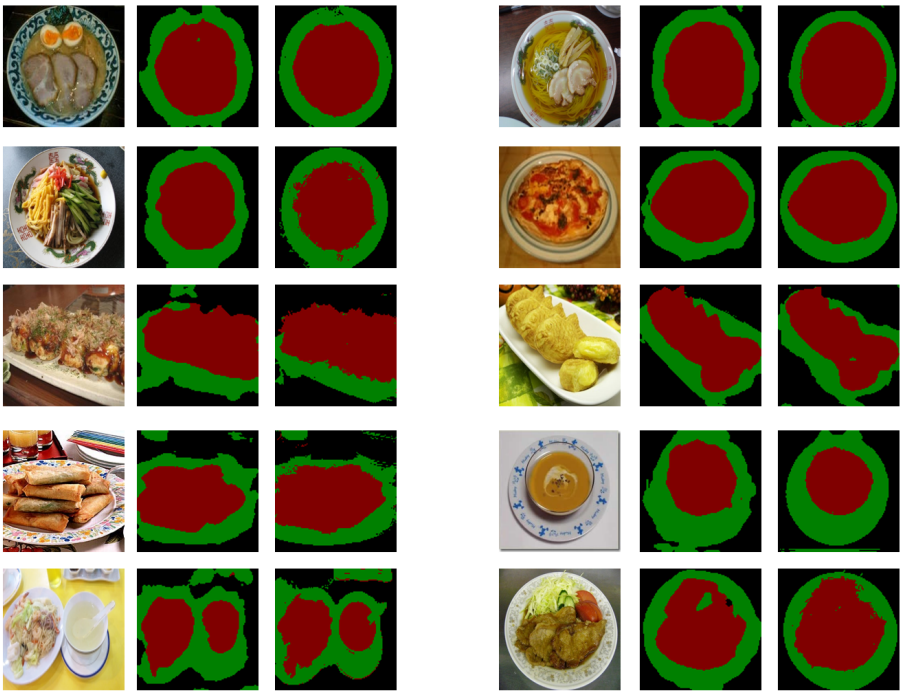

Figure 10.4: The examples of the plate segmentation model for the successive results. From left to right, input images, raw plate segmentation masks and CRF applied plate segmentation masks.

10.3.3 Ablation study

Here, we study how each of the parts of the proposed approach influences the overall performance. Table 10.1 shows the improvement of the accuracy of weakly-supervised segmentation by the proposed approaches. **Constrain** is the approach proposed in Section 10.2.2 for reducing overflowed food regions and **Penalizing** is the approach proposed in Section 10.2.3 for enhancing outputs of food regions on pixels that are often classified as background. The constraint of the food regions using plate segmentation causes large performance dropping because the constraint is too strong and makes the unbalance on inference of the background class though we expected it would be helpful to capture the boundary of the food regions. Penalizing background regions using plate segmentation boosts up the accuracy from 49.7% to 52.6%. This gives evidence that SSDD tends to misclassify on pixels that estimated as

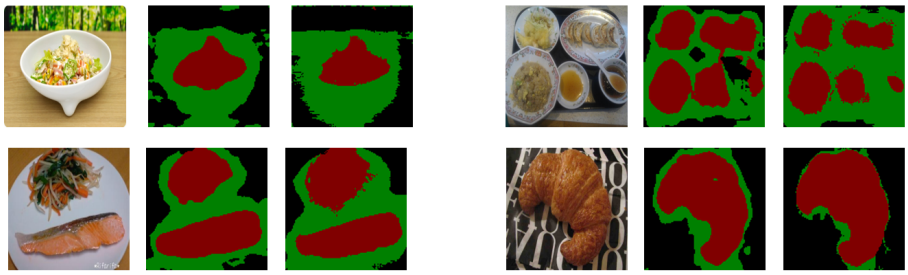

Figure 10.5: The examples of the plate segmentation model for the failure cases. From left to right, input images, raw plate segmentation masks and CRF applied plate segmentation masks.

Table 10.1: Ablation study for the approaches to refine food segmentation by plate segmentation masks.

Method	**Constrain**	**Penalizing**	mIoU	Pix acc
(I)	-	-	49.7	78.3
(II)	✓	-	42.9	75.4
(III)	-	✓	52.6	81.0
(IV)	✓	✓	**55.4**	**82.6**

background, and plate segmentation can assist to reduce the misclassification on such pixels. When we incorporate both of the approaches, constraint of the food regions further leads to the performance boost of 2.8%. These results indicate that both of approaches help weakly-supervised food segmentation. The balance on the food regions and background regions is important, and plate segmentation is effective on making the balance. We show the qualitative results in Fig.10.6.

10.3.4 Comparison with existing weakly-supervised segmentation methods

We compare with three existing weakly-supervised segmentation methods. Class Activation Mapping (CAM) [36] is a popular weakly-supervised segmentation method that roughly outputs object location with the ambiguous boundary. SSDD [11] is one of the state-of-the-art method among the current

Table 10.2: Comparison with existing methods.

Method	mIoU	Pix acc
CAM [36]	30.7	65.1
SSDD (base method) [11]	49.7	78.3
Simple Does It [104]	51.1	81.9
PFSeg (Proposed)	**55.4**	**82.6**

works of weakly-supervised segmentation that greatly improves CAM using CRF and the self-supervised difference detection module. We used SSDD as a base method, and combine the proposed approaches that use plate segmentation. To assess the effectiveness of the proposed approach, we also compare the proposed approach with "Simple Does It" [104]. "Simple Does It" [104] is a well-known bounding box-based weakly-supervised segmentation. While CAM, SSDD and the proposed method are trained with only image-level labels, "Simple Does It" requires bounding boxes for training, i.e. it uses additional supervision. We compare the proposed method with the simplest way using GrabCut [93] proposed in the paper [104]. More concretely, the method generates pseudo pixel-level labels from each bounding box by applying GrabCut [93] and extracting foreground masks. After extracting foreground masks, the method gives the category labels to the foreground masks using the labels of the bounding boxes, then the method trains a segmentation model with the generated segmentation masks. This approach is simple, but a powerful baseline considering the advantage of bounding box information. The performance comparisons are summarized in Table 10.2. We denoted the proposed method as Plate-based Food Segmentation (PFSeg).

As shown in Table 10.2, the proposed method achieved 55.4% on mIoU and 82.6% on Pix acc. Compared with the base method, the gain are 5.7 points and 4.3 points on mIoU and Pix acc, respectively. They are also higher than "Simple Does It", which uses bounding boxes as additional training information. These results indicate that the proposed method is efficient and plate segmentation model trained without pixel-wise annotation is beneficial for improving weakly-supervised food segmentation. Fig.10.7 shows the examples of the weakly-supervised food segmentation methods.

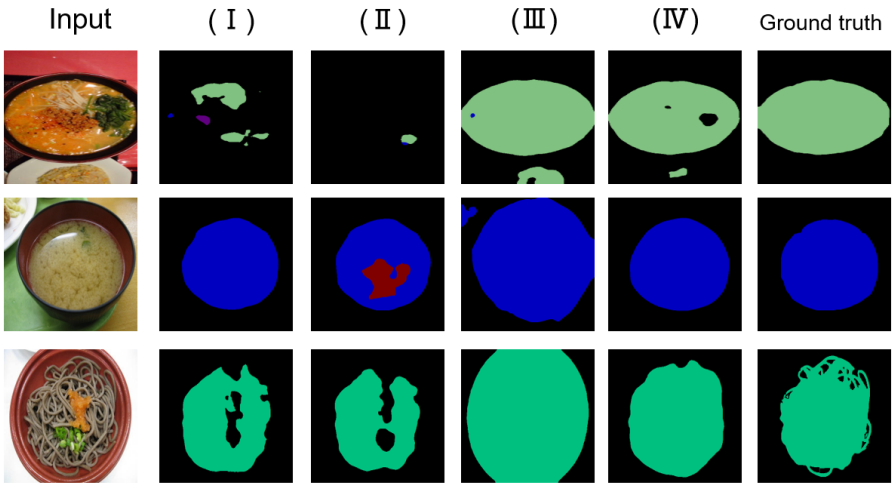

Figure 10.6: Examples of the weakly-supervised food segmentation results. (I), (II), (III) and (IV) correspond to Table 10.1 of the method. From the results, we can observe that the both of the proposed approaches make large effects on the balance of inference for the food regions and background regions, and we can make the good balance by using both of them together.

10.4 Summary

In this chapter, we proposed a method to synthesize segmentation masks for food plate regions by visualization. We used a food category classifier and a food/non-food classifier for visualization and extracted plate regions from the difference in the visualization of the two types of the classifiers. In addition, we also proposed the approach to make consistency between a food segmentation model and a plate segmentation model, and demonstrated that we boosted the accuracy of weakly-supervised food segmentation using the proposed approach.

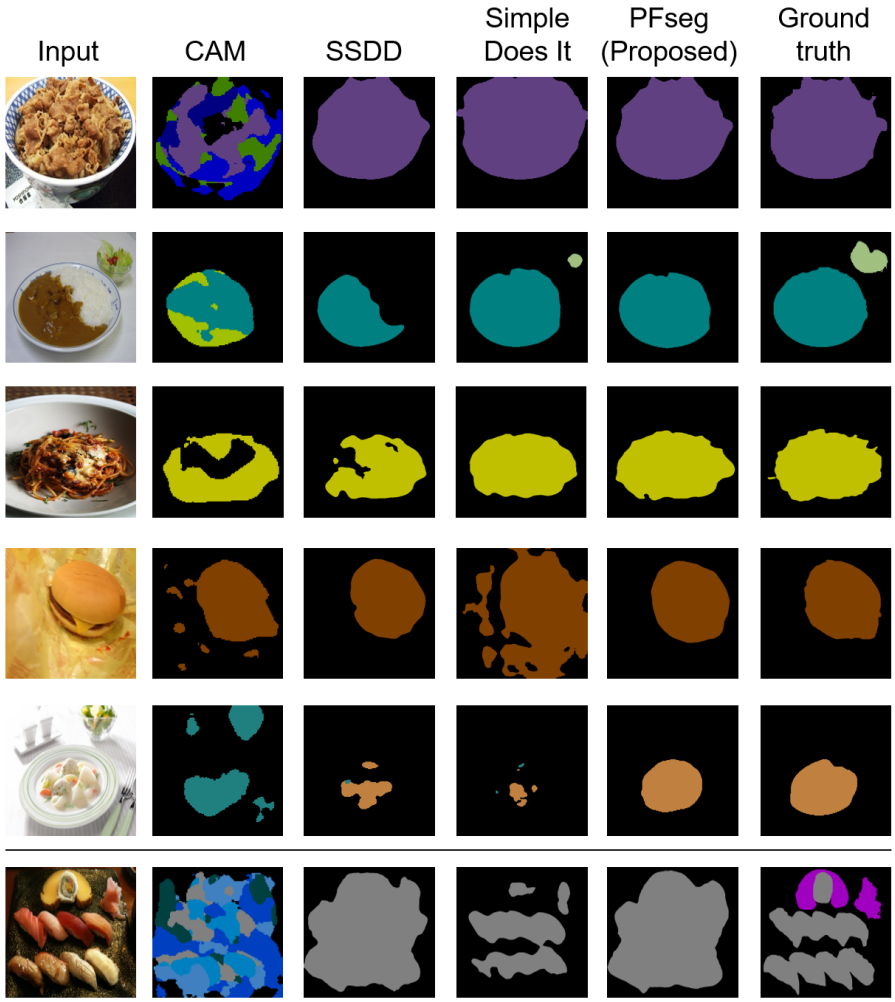

Figure 10.7: Examples of the comparison with existing weakly-supervised food segmentation methods. Simple does it [104] often fails when the background color is similar to the object color because the method is based on GrabCut, which uses color information for extracting foreground masks. The bottom row shows the typical failure case. Simple Does It is better than the proposed method on the inference of the small target objects so that Simple Does It uses bounding box annotations that make big advantages for such targets.

Chapter 11

Conclusion

11.1 Conclusion

In this thesis, we aimed to improve the accuracy of weakly-supervised segmentation for reduction of the annotation cost of CNN-based semantic segmentation. Especially, we focused on a setting as weakly-supervised segmentation that allows only image-level labels for training data. If we can train semantic segmentation models with image-level labels instead of pixel-level labels, it would be a good option as a cost-effective method. However, there is a large gap between semantic segmentation models trained with image-level labels and trained with pixel-level labels. To make the gap as narrow as possible, we attempted to improve the accuracy of weakly-supervised segmentation. To imporove the weakly-supervised segmentation, we explored weakly-supervised segmentation methodology from the two perspectives: improving visualization-based methods and pseudo pixel-level labels-based methods. Different from the existing works, we investigated backward-based visualization techniques (Chapter 3, Chapter 4) and estimation methodology of noise reduction of pseudo pixel-level labels (Chapter 5, Chapter 6), respectively.

For improving visualization techniques, though forward-based visualization have been actively studied, we focused on backward-based visualization. General backward-based visualization techniques have problems for the multiple-category targets in an image. We found that the problem can be alleviated by taking the difference of visualization between different category signals, and demonstrated that backward-based visualization would also be a good option as a weakly-supervised segmentation method.

Though, in the visualization approach, we found that the backward-based

approach is effective, the performance of visualization as segmentation is limited because visualization is not equal to the segmentation. Therefore, we also investigated weakly-supervised segmentation methods using pseudo pixel-level labels. In this approach, we compensated insufficient components in the visualization using color information. However, refinement using color information is not always accurate. Differing from existing weakly-supervised segmentation methods, in this thesis, we aimed to estimate noisy training data from pseudo pixel-level labels using the visualization and the color information, and proposed two methodologies that estimate noisy training data in image-level and in pixel-level. In the former approach, we estimated noisy training data in image-level and rejected them from training data. In the latter approach, we estimated noisy training data in pixel-level and interpolated them to better labels for the training data. We verified that both of the approaches improved the accuracy of weakly-supervised segmentation. However, the latter approach is better than the former approach with a large margin. We analyzed the reason is that there is a trade-off in the former approach for the quality of the training data and the quantity of the training data. While the former approach decreases the amount of the training data for ignoring the bad pseudo pixel-level labels, the latter approach is free from the problem for the amount of the training data. We consider that this is because the latter approach outperformed the former approach.

As we stated, we explored effective weakly-supervised segmentation methods in the different perspectives comparing existing works through the two approaches. In the visualization-based approach, we demonstrated that backpropagation-based methods can effectively capture outlines of objects. In the pseudo pixel-level labels-based approach, we proposed the methods that estimate noisy training data from the pseudo pixel-level labels that are synthesized to compensate for the insufficient components in the visualization using the color information. We proposed four weakly-supervised segmentation methods in this thesis and two of four methods achieved the state-of-the-art performances at that time (Chapter 4, Chapter 6). In recent years, the accuracy of weakly-supervised segmentation has been improved greatly. Actually, over 20% points are increased on the benchmark of Pascal VOC dataset from when we have started to explore the methods of weakly-supervised segmentation. However, there is a still gap between weakly-supervised segmentation and fully-supervised segmentation. We believe that our works have contributed to this great progress of weakly-supervised segmentation,

and would be helpful for following researches on weakly-supervised segmentation.

In addition, as an application of weakly-supervised segmentation, we also have studied weakly-supervised food segmentation. We adapted weakly-supervised segmentation methods for general objects to food images and investigated the effects of that. As results, the weakly-supervised segmentation methods for general objects often drop the performance in the food domain. However, we also showed that performance dropping can be recovered by considering approaches utilizing food-specific characteristics. We believe that these results are beneficial to know the tendency of the versatile weakly-supervised methods.

11.2 Future work

In this thesis, we proposed several methods for improving weakly-supervised segmentation. As future works, we consider two directions. The first direction is further improvements for weakly-supervised segmentation. The second one is adaption and integration for not only segmentation but also other tasks. We explain the detail of both future directions below.

As a reasonable future work, we consider further improvement of the performance of weakly-supervised segmentation. In this thesis, we proposed the four methods for the improvement of the performance, and totally more than 20% improvement was achieved through the four methods. However, recently, the performance of fully-supervised segmentation as well as the performance of weakly-supervised segmentation greatly improved, then the gap of the performance between them is still large. If the gap between them would be narrow, the range of applications would be expanded. There would be many factors of the problems of the current weakly-supervised segmentation. Especially, we consider that one of the largest problem is that there are categories that color information are ineffective on. Recent weakly-supervised segmentation methods depend on refinement methods using color information such as Conditional Random fields (CRF). However, these methods are not effective in some categories. Concretely, in animal categories, color information would work well because the color of their skins and furs are consistent in general. On the other hand, in the category of potted plant, chair and table, color information is not effective because some of their parts are thin, such as the stem of the plants and legs of chairs. Actually, the method we proposed

in Chapter 6, which utilizes CRF to the utmost degree, greatly improved the performance on animal categories, made counter-productive on the potted plant class. As we stated, the recent weakly-supervised segmentation methods depend on color information and to solve the problem considering other approaches would be important. We consider keypoint detection has potential to solve this problem. Keypoint detection is a task to extract corresponding regions from a pair of images. If we adapt keypoint detection to a pair of images including the same category objects, the extracted regions using keypoint detection would be the regions of the same category objects. We consider the regions extracted by this techniques would not dependent on color information, and further improvement is expected by this approach.

As second future work, we consider adaption of weakly-supervised learning on other tasks and integration of weakly-supervised learning on multiple tasks. As weakly-supervised learning on other tasks, there exist weakly-supervised detection, weakly-supervised event detection, weakly-supervised pose estimation and so on. Recently, weakly-supervised instance segmentation has been also actively studied. Instance segmentation is a kind of multiple-task learning of semantic segmentation and object detection. Therefore, instance segmentation requires both of annotation cost for semantic segmentation and object detection, then the cost becomes quite large. The multiple-task based method such as instance segmentation might draw further attention, but multiple-task requires additional annotation costs for the number of tasks. If it happened, the annotation cost of one image would become large and the demand for weakly-supervised learning would increase. Weakly-supervised segmentation is one of the earliest research on weakly-supervised learning and has been actively studied. Therefore, the knowledge of weakly-supervised segmentation could help other weakly-supervised learning. We also consider that the multiple-task on weakly-supervised learning might be well combined. In promising multiple-task such as instance segmentation, there would be large correlations between its supervisions, and it might be inefficient to annotate all supervisions for all multiple-tasks, respectively. Actually, Khoreva et al. [104] reported that high performance of semantic segmentation can be achieved trained with the supervision of object detection. When training models of multiple-tasks, to consider the correlation between the tasks might be beneficial. Though several methods of weakly-supervised instance segmentation have been proposed, the methods aim to keep the performance of both tasks: semantic segmentation and

object detection and these methods do not intend to utilize the correlations between the tasks. In fact, the performance of each task of weakly-supervised instance segmentation is lower than the performance of the single task. If we can utilize the correlation of the multiple-tasks, the performance of each single task also might be improved.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2009.

[2] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proc. of International Conference on Learning Representations Workshop Track (ICLRWS)*, 2014.

[3] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *Proc. of International Conference on Learning Representations (ICLR)*, 2015.

[4] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2015.

[5] D. Pathak, P. Krahenbuhl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2015.

[6] A. Kolesnikov and C. H.Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *Proc. of European Conference on Computer Vision (ECCV)*, 2016.

[7] Y. Wei, X. Liang, Y. Chen, X. Shen, M. Cheng, Y. Zhao, and S. Yan. STC: A simple to complex framework for weakly-supervised semantic segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[8] W. Shimoda and K. Yanai. Distinct class saliency maps for weakly supervised semantic segmentation. In *Proc. of European Conference on Computer Vision (ECCV)*, 2016.

[9] W. Shimoda and K. Yanai. Weakly supervised semantic segmentation using distinct class specific saliency maps. *Computer Vision and Image Understanding*, 2019.

[10] W. Shimoda and K. Yanai. Predicting segmentation "easines" from the consistency for weakly-supervised segmentation. In *Proc. of Asian Conference on Pattern Recognition (ACPR)*, 2017.

[11] W. Shimoda and K. Yanai. Self-supervised difference detection for weakly-supervised semantic segmentation. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2019.

[12] W. Shimoda and K. Yanai. Self-supervised difference detection for refinement crf and seed interpolation. In *Proc. of CVPR Workshop on Weakly Supervised Learning for Real-World Computer Vision Applications*, 2017.

[13] W. Shimoda and K. Yanai. Cnn-based food image segmentation without pixel-wise annotation. 2015.

[14] W. Shimoda and K. Yanai. Webly-supervised food detection with foodness proposal. *IEICE Transactions on Information and Systems*, 2019.

[15] Shimoda. W and Yanai. K. Foodness proposal for webly-supervised food detection. 2016.

[16] W. Shimoda and K. Yanai. Zero-annotation plate segmentation using a food category classifier and a food/non-food classifier. In *Proc. of ICCV Workshop on Multi-Discipline Approach for Learning Concepts (MDALC)*, 2019.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2014.

[18] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *Proc. of European Conference on Computer Vision (ECCV)*, 2014.

[19] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2015.

[21] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2015.

[22] P. Sermanet, D. Eigen, X. Zhang, M.l Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *Proc. of International Conference on Learning Representations (ICLR)*, 2014.

[23] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and Yuille A. L. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *Proc. of International Conference on Learning Representations (ICLR)*, 2015.

[24] S. Zheng, S. Jayasumana, B. R. Paredes, V. Vineet, and Z. Su. Conditional random fields as recurrent neural networks. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2015.

[25] S. Bell, N. Upchurch, P. Snavely, and K. Bala. Material recognition in the wild with the materials in context database. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2015.

[26] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille. Semantic iamge segmentation with deep convolutional nets and fully connected crfs. In *Proc. of International Conference on Learning Representations (ICLR)*, 2015.

[27] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *Proc. of International Conference on Learning Representations (ICLR)*, 2016.

[28] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017.

[29] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 346–361, 2014.

[30] L. Chen, Y. Yang, J. Wang, Wei. Xu, and A. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2016.

[31] L. C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[32] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. of European Conference on Computer Vision (ECCV)*, 2018.

[33] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017.

[34] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2014.

[35] P. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2015.

[36] B. Zhou, A. Khosla, A. Lapedriza, and A. Oliva, A. Torralba. Learning deep features for discriminative localization. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2016.

[37] Y. Wei, J. Feng, X. Liang, M. Cheng, Y. Zhao, and S. Yan. Object region mining with adversarial erasing: A simple classification to se-

mantic segmentation approach. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017.

[38] P. Krahenbuhl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[39] J. Ahn and S. Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

[40] F. Saleh, M. Akbarian, M. Salzmann, L. Petersson, S. Gould, and J. M.Alvares. Built-in foreground/background prior for weakly-supervised semantic segmentation. In *Proc. of European Conference on Computer Vision (ECCV)*, 2016.

[41] A. Roy and S. Todorovic. Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017.

[42] D. Kim, D. Cho, D. Yoo, and I. Kweon. Two-phase learning for weakly supervised object localization. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2017.

[43] Q. Hou, P. T. Jiang, Y. Wei, and M. M. Cheng. Self-erasing network for integral object attention. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[44] T. Shen, G. Lin, C. Shen, and R. Ian. Bootstrapping the performance of webly supervised semantic segmentation. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

[45] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully convolutional multi-class multiple instance learning. In *Proc. of International Conference on Learning Representations (ICLR)*, 2015.

[46] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2013.

[47] Z. Huang, W. Xinggang, J. Wang, W. Liu, and W. Jingdong. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

[48] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision. In *Proc. of European Conference on Computer Vision (ECCV)*, 2016.

[49] B. Jin, M. Segovia, and S. Susstrunk. Webly supervised semantic segmentation. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

[50] P. Tokmakov, K. Alahari, and C. Schmid. Weakly-supervised semantic segmentation using motion cues. In *Proc. of European Conference on Computer Vision (ECCV)*, 2016.

[51] S. Hong, D. Yeo, S. Kwak, H. Lee, and B. Han. Weakly supervised semantic segmentation using web-crawled videos. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017.

[52] O. Seong, B. Rodrigo, K. Anna, A. Zeynep, F. Mario, and S. Bernt. Exploiting saliency for object segmentation from image level labels. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017.

[53] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. Huang. Revisiting dilated convolution: A simple approach for weakly- and semisupervised semantic segmentation. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

[54] X. Wang, S. You, X. Li, and H. Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

[55] A. Chaudhry, K. P. Dokania, and P. H. Torr. Discovering class-specific pixels for weakly-supervised semantic segmentation. In *Proc. of British Machine Vision Conference (BMVC)*, 2017.

[56] R. Fan, Q. Hou, M. M. Cheng, G. Yu, R. R. Martin, and S. M. Hu. Associating inter-image salient instances for weakly supervised semantic

segmentation. In *Proc. of European Conference on Computer Vision (ECCV)*, 2018.

[57] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.

[58] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and Malik. J. Semantic contours from inverse detectors. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2011.

[59] K. Simonyan, A. Vedaldi, and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of International Conference on Learning Representations (ICLR)*, 2015.

[60] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[61] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimality boundary & region segmentation of objects in n-d images. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2001.

[62] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proc. of European Conference on Computer Vision (ECCV)*, 2006.

[63] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2009.

[64] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. Slic superpixels compared to state-of-the-art superpixel methods. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[65] J. Pont-Tuset, A. Arbelaez, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2014.

[66] P. Sermanet, D. Eigen, X. Zhang, M.l Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *Proc. of International Conference on Learning Representations (ICLR)*, 2014.

[67] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? -weakly-supervised learning with convolutional neural networks. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2015.

[68] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[69] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *Proc. of European Conference on Computer Vision (ECCV)*, 2012.

[70] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013.

[71] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *Proc. of European Conference on Computer Vision (ECCV)*, 2016.

[72] X. Qi, Z. Liu, J. Shi, H. Zhao, and J. Jia. Augmented feedback in semantic segmentation under image level supervision. In *Proc. of European Conference on Computer Vision (ECCV)*, 2016.

[73] Z. Cheng, Q. Yang, and B. Sheng. Deep colorization. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2015.

[74] C. Doersch, A. Gupta, and A. Efros. Unsupervised visual representation learning by context prediction. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2015.

[75] L. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *Proc. of International Conference on Learning Representations (ICLR)*, 2017.

[76] W. Ge, S. Yang, and Y. Yu. Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

[77] K. Li, Z. Wu, K.-C. Peng, J. Ernest, and Y. Fu. Tell me where to look: Guided attention inference network. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

[78] L. Bossard, M. Guillaumin, and L. V. Gool. Food-101 - mining discriminative components with random forests. In *Proc. of European Conference on Computer Vision (ECCV)*, 2014.

[79] H. Kagaya, K. Aizawa, and M. Ogawa. Food detection and recognition using convolutional neural network. In *Proc. of ACM International Conference on Multimedia (ACMMM)*, pages 1085–1088, 2014.

[80] Y. Kawano and K. Yanai. Foodcam: A real-time food recognition system on a smartphone. *Multimedia Tools and Applications*, pages 1–25, 2014.

[81] Y. Matsuda, H. Hoashi, and K. Yanai. Recognition of multiple-food images by detecting candidate regions. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, pages 1554–1564, 2012.

[82] M. Chen, Y. Yang, C. Ho, S. Wang, S. Liu, E. Chang, C. Yeh, and M. Ouhyoung. Automatic chinese food identification and quantity estimation. In *Proc. of SIGGRAPH Asia*, 2012.

[83] M. Bosch, F. Zhu, N. Khanna, C. J. Boushey, and E. J. Delp. Combining global and local features for food identification in dietary assessment. In *Proc. of IEEE International Conference on Image Processing (ICIP)*, 2011.

[84] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar. Food recognition using statistics of pairwise local features. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2010.

[85] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[86] Y. Kawano and K. Yanai. Food image recognition with deep convolutional features. In *Proc. of ACM UbiComp Workshop on Workshop on Smart Technology for Cooking and Eating Activities (CEA)*, 2014.

[87] Chamin Morikawa, Haruki Sugiyama, and Kiyoharu Aizawa. Food region segmentation in meal images using touch points. In *Proc. of ACM MM WS on Multimedia for Cooking and Eating Activities (CEA)*, pages 7–12, 2012.

[88] Y. Kawano and K. Yanai. Real-time mobile food recognition system. In *Proc. of IEEE CVPR International Workshop on Mobile Vision (IWMV)*, 2013.

[89] Y. He, C. Xu, N. Khanna, C. J. Boushey, and E. J. Delp. Food image analysis: Segmentation, identification and weight estimation. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2013.

[90] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[91] Y. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):800–810, 2001.

[92] P. F. Felzenszwalb and D. P. Huttenlocher. Image segmentation using local variation. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 98–104, 1998.

[93] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004.

[94] F. Kong and J. Tan. DietCam: Automatic dietary assessment with mobile camera phones. In *Proc. of ACM International Conference on Pervasive and Mobile Computing*, pages 147–163, 2012.

[95] M. Chen, Y. Yang, C. Ho, S. Wang, S. Liu, E. Chang, C. Yeh, and M. Ouhyoung. Automatic chinese food identification and quantity estimation. In *Proc. of ACM SIGGRAPH Asia Technical Briefs*, page 29, 2012.

[96] A. Meyers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. P. Murphy. Im2calories: Towards an automated mobile vision food diary. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[97] P. Pouladzadeh, S. Shirmohammadi, and R. Almaghrabi. Measuring calorie and nutrition from food image. *IEEE Transactions on Instrumentation and Measurement*, 63(8):1947–1956, 2014.

[98] T Ege and K. Yanai. Image-based food calorie estimation using knowledge on food categories, ingredients and cooking directions. In *Proc. of ACM International Conference on Multimedia (ACMMM)*, 2017.

[99] T Ege, W. Shimoda, and K. Yanai. A new large-scale food image segmentation dataset and its application to food calorie estimation based on grains of rice. 2019.

[100] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[101] M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Proc. of European Conference on Computer Vision (ECCV)*, 2014.

[102] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *Proc. of International Conference on Learning Representations Workshop Track (ICLRWS)*, 2015.

[103] A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. *arXiv preprint arXiv:1412.4564*, 2014.

[104] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele. Simple does it: Weakly supervised instance and semantic segmentation.

In *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017.

Journal Publications

[J1] W. Shimoda and K. Yanai, "Weakly Supervised Semantic Segmentation Using Distinct Class Specific Saliency Maps", Computer Vision and Image Understanding, Elsevier, 2019. (in press) (Chapter 5) (https://doi.org/10.1016/j.cviu.2018.08.006)

[J2] W. Shimoda and K. Yanai, "Webly-Supervised Food Detection with Foodness Proposal", IEICE Transactions on Information and Systems, Vol. E102-D, No.7, pp. 1230-1239, 2019. (Chapter 9) (https://doi.org/10.1587/transinf.2018CEP0001)

International Conference Publications

[C1] W. Shimoda and K. Yanai, "CNN-Based Food Image Segmentation without Pixel-Wise Annotation", Proc. of International Workshop on Multimedia Assisted Dietary Management (MADiMa), 2015. (Chapter 8)

[C2] W. Shimoda and K. Yanai, "Distinct Class-specific Saliency Maps for Weakly Supervised Semantic Segmentation", Proc. of European Conference on Computer Vision (ECCV), 2016. (Chapter 4)

[C3] W. Shimoda and K. Yanai, "Foodness proposal for Webly-Supervised Food Detection", Proc. of International Workshop on Multimedia Assisted Dietary Management (MADiMa), 2016. (Chapter 9)

[C4] W. Shimoda and K. Yanai, "Weakly-Supervised Segmentation by Combining CNN Feature Maps and Object Saliency Maps", Proc. of International Conference on Pattern Recognition (ICPR), 2016. (Chapter 3)

[C5] W. Shimoda and K. Yanai, "Predicting Segmentation "Easines" from the Consistency for Weakly-Supervised Segmentation", Proc. of Asian Conference on Pattern Recognition (ACPR), 2017. (Chapter 5)

[C6] W. Shimoda and K. Yanai, "Self-supervised Difference Detection for Refinement CRF and Seed Interpolation, Proc. of CVPR Workshop on Weakly Supervised Learning for Real-World Computer Vision Applications, 2019. (Chapter 6)

[C7] W. Shimoda and K. Yanai, "Self-supervised Difference Detection for Weakly-supervised Semantic Segmentation", Proc. of IEEE/CVF International Conference on Computer Vision (ICCV), 2019. (Chapter 6)

[C8] W. Shimoda and K. Yanai, "Zero-Annotation Plate Segmentation Using a Food Category Classifier and a Food/Non-Food Classifier", Proc. of ICCV Workshop on Multi-Discipline Approach for Learning Concepts (MDALC), 2019. (Chapter 10)

Invited Talks

[T1] “画像を生成する深層学習ネットワーク —領域分割と画像生成・変換—”, 日本画像学会 第 31 回フリートーキング Imaging Today (2017/07).

[T2] “Font Image Conversion Using Style Transfer and Cross Domain Transfer Learning”, International Display Workshops (2018/11).

[T3] “深層学習による質感画像の認識・変換”, 日本鉄鋼協会秋季講演大会 (2019/09).

[T4] “ICCV 採択論文紹介 (Self-supervised Difference Detection for Weakly-supervised Semantic Segmentation)”, 電子情報通信学会パターン認識・メディア理解研究会（PRMU） (2019/10).