Semantic segmentation with less annotation efforts

Zhang, Tianyi

2020

Zhang, T. (2020). Semantic segmentation with less annotation efforts. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/140292>

<https://doi.org/10.32657/10356/140292>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 20:59:39 SGT

SEMANTIC SEGMENTATION WITH LESS ANNOTATION EFFORTS

ZHANG TIANYI

Interdisciplinary Graduate School

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2020

SEMANTIC SEGMENTATION WITH LESS ANNOTATION EFFORTS

ZHANG TIANYI

Interdisciplinary Graduate School

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

May 26th, 2020

.....

Date

张天一

.....

ZHANG TIANYI

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

May 26th, 2020

.....

Date

.....

Asst Prof. Lin Guosheng

Authorship Attribution Statement

This thesis contains material from 4 papers published in the following peer-reviewed journal(s) / accepted (or submitted to) at conferences in which I am listed as an author.

Chapter 3 is published as [Tianyi Zhang, Guosheng Lin, Jianfei Cai, Tong Shen, Chunhua Shen and Alex C. Kot. Decoupled Spatial Neural Attention for Weakly Supervised Semantic Segmentation. IEEE Transactions on Multimedia, 2930 - 2941 \(2019\). DOI: 10.1109/TMM.2019.2914870.](#)

The contributions of the each co-author/collaborator are as follows:

- Prof Lin Guosheng provided the initial project direction.
- Prof Lin Guosheng advised on the details of the experiments.
- I prepared the manuscript drafts. The manuscript was revised by Prof Lin Guosheng, Prof Cai Jianfei, Prof Shen Chunhua and Prof Alex C. Kot.
- I designed and conducted most of the experiments.
- Tong Shen assisted to conduct parts of the experiments.

Chapter 4 is submitted as [Tianyi Zhang, Guosheng Lin, Weide Liu and Jianfei Cai. Splitting vs. Merging: Mining Object Regions with Discrepancy and Intersection Loss for Weakly Supervised Semantic Segmentation. IEEE International Conference on Computer Vision and Pattern Recognition \(2020\).](#)

The contributions of the each co-author/collaborator are as follows:

- I came up with the initial idea to start.
- Prof Lin Guosheng advised on the details of the experiments.
- I wrote the manuscript drafts. The manuscript was checked by Prof Lin Guosheng and Prof Cai Jianfei.
- I designed and conducted most of the experiments.
- Weide Liu assisted to conduct parts of the experiments.

Chapter 5 is published is composed of two publications:

[Tianyi Zhang, Guosheng Lin, Jianfei Cai and Alex Kot. Semantic Segmentation via Domain Adaptation with Global Structure Embedding. IEEE International Conference on Visual Communications and Image Processing \(2019\).](#)

The contributions of the each co-author/collaborator are as follows:

- I came up with the initial research direction to investigate.

- Prof Lin Guosheng advised the reference papers to refer to.
- I wrote the manuscript drafts. The manuscript was revised by Prof Lin Guosheng, Prof Cai Jianfei and Prof Alex C. Kot.
- I designed and conducted the experiments.

Tianyi Zhang, Jingyi Yang, Chuanxia Zheng, Guosheng Lin, Jianfei Cai and Alex C. Kot. Task-in-all Domain Adaptation for Semantic Segmentation. IEEE International Conference on Visual Communications and Image Processing (2019).

The contributions of the co-authors are as follows:

- Prof Cai and Zheng Chuanxia provided the initial project direction.
- I wrote the manuscript drafts. The draft was checked by Prof Lin Guosheng, Prof Cai Jianfei and Prof Alex C. Kot.
- I designed and conducted the experiments.
- Yang Jingyi prepared parts of the drafts.

May 26th, 2020

.....

Date

张天一

.....

ZHANG TIANYI

Acknowledgements

First of all, I want to thank my supervisor Prof Cai Jianfei who admitted me as a phd student five years ago. Through the five years no matter how much difficulty I was faced with in my research, he never gave me up and tried to help me go through the tough period. He always informed me of new research directions in the field and provided me with interesting ideas. He is also willing to share his experiences with me and give me advice in life. It has been a great honour to be his student.

I also want to thank my current supervisor Asst Prof Lin Guosheng who guides me in the field of semantic segmentation. He started to instruct me when I was totally at a loss about my research. I learned lots of methods in doing research and experiments from him. With his detailed instructions, I started from a initial learner to a relatively more experienced researcher in this field and gradually recovered my confidence in research. He gave me a lot of encouragement when I almost lost confidence and interest in my research and phd study. Under his instructions, I published my first transaction paper and finally complete the research works to formulate this degree thesis.

I also would like to appreciate my co-supervisor Prof Alex C. Kot. He taught me to take a patient attitude and steady step toward my research. He cared about my progress and situation when I was worried and anxious about my research. He gave me a lot of encouragement when I suffered from frustrated moments in my research experience.

I want to thank my other collaborators/instructors in research. Dr. Zhang Yu provided detailed advice when I started my research in NTU in the first year. He recognized my direction of research when I wasn't sure about it. With his help, I published my first paper and gained confidence in my research progress. Dr Niu Li also helped me a lot in revising the manuscripts and taught me how to

organize the logic structure of the manuscripts. I want to thank my friends and colleges in NTU, including but are not limited to Li Haoliang, Wan Renjie, Zhang Yabin, Zhang Xinfeng, Gao Guanyu, Shi Xiangxi, Wu Zhonghua, Chen Zhuo, Liu Weide, Tao Qingyi, Wang Anran, Cai Yujun, etc. They are also willing to help me when I came across the difficulties by sharing me with lots of useful suggestions or listening to my complaints about the research experience. With their help, I never felt helpless no matter what difficulties I came across. I also want to thank my other close friends, including but are not limited to Zhu Guanyu, Wu Yue, Li Wen, Wu Di, etc. No matter whether they are in Singapore or not, they are always willing to spend time and patience to my endless and repeated complaints. It is quite lucky to have these friends to support me.

It is a special and precious experience to pursue phd degree in NTU. From this experience I learned a lot about how to conduct research, although I am not doing well in this. More importantly, I gradually gained experience on how to analyse and solve the problems, how to balance between the pressure and life, how to collaborate with and learn from other people, etc. This experience benefits me a lot and helps me to have a better understanding about myself. Finally, I want to thank all the people, including myself, who helped me through this phd experience.

To my dear friends

Abstract

Semantic segmentation is a pixel-wise classification task, which is to predict class label to every pixel within an image. By interpreting images into the segments representing different semantic-level meanings, semantic segmentation is a fundamental step to many applications such as fashion parsing and autonomous driving. The training datasets for semantic segmentation are usually accompanied with pixel-wise groundtruth annotations. With the development of deep learning technology, the segmentation models trained with pixel-wise segmentation ground-truth labels can accurately predict segmentation masks.

However, one of the obstacles limiting the development of semantic segmentation is that the pixel-wise segmentation annotations for the real images are quite difficult and expensive to obtain. Thus, it is difficult to apply the segmentation methods to new datasets or semantic classes since it is labour-intensive to obtain new annotated data.

The target of this thesis is to reduce the annotation work load for semantic segmentation tasks on images. The approaches to tackle this problem are mainly classified into two categories: The first category is to rely on weaker level of supervisions within the datasets of same data domain in the training process. The second category is to transfer the pixel-wise knowledge from another domain, whose data and pixel-wise annotations are easier to generate, to the target data domain.

The first category is to utilize weaker level of supervisions within the datasets of same data domain. Such supervision formats include but are not limited to scribbles, boxes and image-level labels. The main challenge in the category is about how to recover the pixel-wise annotations from the weak labels. The supervision format of image-level labels for weakly supervised semantic segmentation is investigated in this thesis. In order to recover the pixel-wise annotations from image-level labels, region-mining models are trained to approximate the target object regions. The goal is to train the region-mining models which could highlight the integral object

regions instead of only the most discriminative regions. In this thesis, I investigate regularizing the region-mining model both in the forward pass and the backward pass of the training process.

The second category is to transfer the pixel-wise knowledge from another domain, whose data and pixel-wise annotations are easier to generate, to the target data domain. Although the pixel-wise annotations are difficult to obtain in the real image domain, it is easy to obtain such data and strong annotations from synthetic image domain. However, there is an obvious domain gap between different data domains. Thus, the main challenge in this category is about how to narrow the domain gap between source and target domain. In this thesis, I investigate the case of transferring from the synthetic source data with pixel-wise annotations to the real-world unlabeled target data. Adversarial learning approaches are applied both in pixel-level and feature-level in order to narrow the domain gap between the synthetic data and real-world data. Adversarial learning approaches usually suffer from the problem of content misalignment. To alleviate the content misalignment problem, two approaches are proposed in this thesis to regularize adversarial learning methods: the first is to embed the global structure knowledge into the feature-level adversarial learning step. The second is to back-propagate the final task loss into the pixel-wise adversarial learning step.

This thesis presents the methods of both the categories to alleviate the need of pixel-wise annotations for semantic image segmentation. The experiments show that the proposed methods could achieve promising segmentation performance without utilizing the pixel-wise annotations.

Contents

Acknowledgements	ix
Abstract	xiii
List of Figures	xix
List of Tables	xxiii
Symbols and Acronyms	xxvii
1 Introduction	1
1.1 Problem Background	1
1.2 Major Contributions	3
1.3 Thesis Outlines	4
2 Literature Review	7
2.1 Semantic Segmentation	7
2.2 Weakly Supervised Semantic Segmentation	8
2.2.1 Webly-Supervised Approaches	9
2.2.2 Region Mining Based approaches	10
2.2.2.1 Localization Model based	10
2.2.2.2 Post-Processing based	12
2.2.2.3 Training Process based	13
2.3 Unsupervised Domain Adaptation	13
2.3.1 Discrepancy based	14
2.3.2 Adversarial Learning based	15
2.3.2.1 Generative Adversarial Networks	15
2.3.2.2 Gradient Reversal Layer	16
2.3.3 Self-Training	17
2.3.4 UDA in Semantic Segmentation	18
3 Decoupled Spatial Neural Attention for Weakly Supervised Semantic Segmentation	21
3.1 Introduction	22

3.2	Methods	24
3.2.1	Conventional Spatial Neural Attention	24
3.2.2	Decoupled Attention Structure	26
3.2.3	Pseudo-annotation Generation	30
3.3	Experiments	31
3.3.1	Experimental Details	31
3.3.2	Ablation Study of Decoupled Attention Model	32
3.3.2.1	Properties of attention maps	32
3.3.2.2	Evaluation of drop-out layers	35
3.3.2.3	Evaluation of decoupled structure	37
3.3.2.4	Evaluation of merging parameters	37
3.3.3	Comparisons with Region-Mining Based Methods	39
3.3.4	Comparisons with State-of-the-arts	40
3.4	Summary	45
4	Splitting vs. Merging: Mining Object Regions with Discrepancy and Intersection Loss for Weakly Supervised Semantic Segmentation	47
4.1	Introduction	48
4.2	Methods	49
4.2.1	Revisiting CAM	50
4.2.2	Splitting vs. Merging	50
4.2.3	Mask Generation	53
4.3	Experiments	54
4.3.1	Datasets and Implementation Details	54
4.3.2	Ablation Study	55
4.3.2.1	Properties of Mining Region	55
4.3.3	Segmentation Results	62
4.4	Summary	64
5	Semantic Content Regularization for Unsupervised Domain Adaptation in Semantic Segmentation	67
5.1	Introduction	68
5.2	Global Structure Embedding	70
5.2.1	Image Translation	71
5.2.2	Feature-level DA with Structure Embedding	72
5.2.3	Self-training	74
5.3	Task-in-all Pipeline	74
5.3.1	Image Translation Stage	76
5.3.2	Segmentation Stage	76
5.3.3	Optimization	77
5.4	Experiments	78
5.4.1	Experimental Set-up	78
5.4.2	Global Structure Embedding	78

5.4.2.1	Ablation Study	78
5.4.2.2	Comparison with the state-of-the-art	80
5.4.3	Task-in-all Pipeline	81
5.4.3.1	Visualization Results	81
5.4.3.2	Segmentation Results	82
5.5	Comparison and Analysis	84
5.6	Summary	86
6	Conclusions and Future Works	87
6.1	Conclusions	87
6.2	Future Works	88
	List of Author's Publications	93
	Bibliography	95

List of Figures

3.1	A brief introduction of our proposed decoupled attention structure. D denotes the channel dimension and W, H denotes the spatial dimension of the feature space. It simultaneously generates two attention maps, namely Expansive Attention map and Discriminative Attention map. Expansive attention map is to identify object regions while the Discriminative attention map is to mine discriminative parts.	23
3.2	The structure of conventional spatial neural attention model. . . .	24
3.3	Our proposed decoupled attention structure. The heatmap generated by Expansive attention detector is named <i>Expansive Attention map</i> , which identifies the object regions. The heatmap generated by the Discriminative Attention detector is named <i>Discriminative Attention map</i> , which localizes the discriminative parts.	26
3.4	Visual examples of Expansive attention map and Discriminative attention map. Expansive attention map and Discriminative attention map have different properties which are suitable for different situations. The Expansive attention maps show superior performance for large objects (2nd and 4th rows) while the Discriminative attention maps are superior for small objects (1st and 3rd rows).	29
3.5	Visualization of attention maps. The fifth column shows the pseudo-annotations generated on the merged attention maps by hard-thresholding. The sixth column shows the pseudo-annotations generated on the merged attention maps after denseCRF refinement. The last column is the real ground-truth annotation for comparison. For the images with multiple classes we show the attention maps separately for each class. We observe that our merged attention maps combine the complementary information of Expansive attention map and Discriminative attention maps and the generated pseudo-annotations are close to the groundtruth.	33
3.6	The visual examples of different drop-out rates (DR). It shows that using a larger drop-out rate ($DR = 0.7$) leads to expansion effect over the using a smaller drop-out rate ($DR = 0.3$).	36
3.7	Visual Examples for the correlation between low classification score and small object range.	38

3.8	Qualitative results of PASCAL 2012 <i>val</i> images. We list the qualitative examples generated by Deeplab-largeFOV (Ours-VGG16) segmentation model and Resnet101 based (Ours-Resnet-101) [1] Deeplab-like segmentation model. It shows that our segmentation results are close to the groundtruth and the results predicted by the Resnet101-based segmentation model outperform the ones predicted by the VGG16-based model.	44
4.1	The brief revisiting of CAM [2] approach. It is a classification model with average pooling step to aggregate the patch-level score map into classification score. The resultant localization map S is obtained by training the model with classification loss.	50
4.2	The structure of our region-mining method. Our structure mainly consists of Reference Stream and Expanding Stream. Expanding Stream aims to expand the highlighted region of the localization maps. It follows an Splitting vs. Merging process, which is achieved by our Discrepancy loss (Disc loss) and Intersection loss (Cls loss inter). Discrepancy loss leads to a splitting effect on the localization maps while the Intersection loss leads to a merging effect. The combination of both losses helps to expand the highlighted regions in the union map output by Expansion Stream.	51
4.3	Visualization results of the input localization maps (S_A and S_B) of the Discrepancy loss with weights $\beta = 10$ and $\beta = 50$. It shows that with the larger weight of Discrepancy loss, the spatial patterns of the input localization maps are more visually different.	55
4.4	The visual examples of failure cases. The highlighted regions are likely to leak into the background region on the cases of frequent co-existence of foreground and background classes (the first row) and small foreground objects (the second row).	57
4.5	The visual examples of the union maps $U(S_A, S_B)$. It shows that with the increase of the weight of the Discrepancy loss (β) the expansion effect of the union map becomes more obvious.	58
4.6	The visually comparison of the union localization maps generated with/without the intersection loss. The weight for Discrepancy loss (β) is 50. It shows that without the intersection loss, the large Discrepancy loss mainly leads to split a single discriminative region instead of having obvious expansion effect.	62
4.7	The visually comparison of the union localization maps generated with/without the intersection loss. The weight for Discrepancy loss (β) is 20. It shows that without the intersection loss, the large Discrepancy loss mainly leads to split a single discriminative region instead of having obvious expansion effect.	63

4.8	Qualitative results of PASCAL 2012 <i>val</i> images. We list the qualitative examples generated by Deeplab-largeFOV (Ours-VGG16) segmentation model and Resnet101 based (Ours-Resnet-50) [1] Deeplab-like segmentation model. It shows that our segmentation results are close to the groundtruth and the results predicted by the Resnet50 based segmentation model outperform the ones predicted by the VGG16-based model.	65
5.1	Our proposed framework of <i>Global Structure Embedding</i> . Our framework is composed of three sequential steps: a) Image translation b) Feature-level domain adaptation. c) Self-training. In order to embed the global structure knowledge, we propose to add structure embedding loss in the feature-level adversarial training stage, besides the segmentation loss and local distribution loss. In the testing phase, we only utilize the final segmentation network, which is circled in orange. We delete details of Image translation step for brief illustration.	71
5.2	The pipeline of our framework of <i>Task-in-all Pipeline</i> . Our framework mainly consists of two parts: Image translation stage and Segmentation stage. The Image translation stage performs pixel-level domain adaptation and the segmentation stage performs high-level domain adaptation and task prediction. Such framework is optimized in an end-to-end manner, which utilizes the final task loss to regularize image translation.	75
5.3	The visualization of segmentation performance of <i>Global Structure Embedding</i>	81
5.4	The visualization of the translated images of <i>Task-in-all Pipeline</i> . The first column demonstrates the original source image, and the second column demonstrates the corresponding translated images of our pipeline. The third column demonstrates the translated images of CycleGAN. The last column shows the examples of the target images. We observe that there exist pixel-level content-misalignment problems in the translated images generated by CycleGAN. Our image translation model alleviate such pixel-level content-misalignment problems.	83

List of Tables

3.1	Quantitative evaluation of attention maps. mS_{prec} is the criteria about localizing the concentrated interior object parts, mS_{rec} is the criteria about covering over the object regions and mS_{IoU} is the criteria for identifying the whole object regions. The results show that Expansive attention map (expan-atten) performs well at covering over and identifying the whole object regions and Discriminative attention map (disc-atten) performs well at localizing interior partial object regions. The merged attention map (merged-atten) performs best at identifying the whole object regions.	34
3.2	Evaluation of different drop-out rates (DR). With the increase of drop-out rates, mS_{rec} constantly increases while mS_{prec} constantly decreases. It shows that large DR enhance the expansion effect of attention maps to some extent.	35
3.3	Evaluation of the number of dropout layers. We observe that adding dropout layers expansion effect (higher mS_{rec}). The results also show that if only using one drop-out layer, the dropout layer before the conv layer (DP_A) plays more important role over then one after conv layer (DP_B).	37
3.4	Comparisons with Region-Mining Based Methods. Both SEC [3] and AE [4] are based on the region-mining model of CAM [2]. We outperform SEC and AE w/o PSL with various settings. Our methods are also more efficient than SEC and AE since we only applies denseCRF in the post-processing step and we only train the region-mining model once.	39
3.5	Comparison between CAM and our attention model. We compare with two kinds of localization cues of CAM: The first one is generated by our re-implementation of VGG16-CAM model in [2] and the second one is the CAM localization cues provided by SEC [3]. We report the final segmentation results. Our attention model outperform CAM in estimating pseudo-annotations which lead to higher $mIoU$ segmentation score.	41
3.6	Our Segmentation results for each class on <i>val</i> and <i>test</i> images. We list the results of using Deeplab-largeFOV (vgg16) [5] and ResNet-101 based Deeplab-like model (resnet) [1] as segmentation DCNN model.	41

3.7	$mIoU$ of Weakly supervised semantic segmentation (WSSS) on <i>val</i> images. We divide the WSSS into three categories based on different levels of weak supervisions and whether additional information (data/supervision) is implicitly used in their pipeline. Our methods achieves the state-of-the-art results in the WSSS supervised only with image labels without external sources.	42
3.8	$mIoU$ of Weakly supervised semantic segmentation (WSSS) on <i>test</i> images. We divide the WSSS into three categories based on different levels of weak supervisions and whether additional information (data/supervision) is implicitly used in their pipeline. Our methods achieves the state-of-the-art results in the WSSS supervised only with image labels without external sources.	43
4.1	The Evaluation of the splitting effect of Discrepancy loss. With the increase of the Discrepancy loss weight β , the similarity score between the two maps regularized by Discrepancy loss constantly decreases, which shows that the Discrepancy loss helps to generate two score maps with different spatial patterns.	57
4.2	Evaluation of the union maps of the input maps of the Discrepancy loss on the <i>val</i> images. It shows that with the increase of the weight of the Discrepancy loss (β), the recall $mSoft_{rec}$ increases while the precision $mSoft_{pre}$ decreases, which shows the expansion effect of the Discrepancy loss. It also shows that adding the Intersection loss (Inter loss) helps the expansion effect more stable and obvious. . .	58
4.3	Our segmentation results for each class on <i>val</i> images. We list the condition whether addition training data (web data) are utilized and whether supervised saliency (S-Sal) are utilized. We achieve the state-of-the-art performance in WSSS.	59
4.4	Our segmentation results for each class on <i>test</i> images. We list the condition whether addition training data (web data) are utilized and whether supervised saliency (S-Sal) are utilized. We achieve the state-of-the-art performance in WSSS.	60
4.5	Our Segmentation results for each class on <i>val</i> images. We list the results of using Deeplab-largeFOV (vgg16) [5] and ResNet-50 based Deeplab-like model (resnet) [1] as segmentation DCNN model. We achieve the state-of-the-art performance in WSSS in most semantic classes.	61
4.6	Our Segmentation results for each class on <i>test</i> images. We list the results of using Deeplab-largeFOV (vgg16) [5] and ResNet-50 based Deeplab-like model (resnet) [1] as segmentation DCNN model. We achieve the state-of-the-art performance in WSSS in most semantic classes.	61
4.7	$mSoft_{overlap}$ of the union of different localization maps on the <i>val</i> images. It shows that the localization maps (S_A, S_B and S_C) are complimentary to each other and the union of all the three maps achives the highest $mSoft_{overlap}$ score.	64

4.8	The evaluation of the generated pseudo-annotations with different Discrepancy weight β . We report $mIoU$ of the generated pseudo-annotations on the <i>train</i> images as the evaluation metrics of the pseudo-annotations.	64
5.1	Ablation Studies of <i>Global Structure Embedding</i> on Adapting GTA5 to Cityscapes. We show that our structure embedding strategy (STRUCT) can help improve performance. We also show the contribution of pixel-level/feature-level domain adaptation and self-training strategy.	79
5.2	Comparison <i>Global Structure Embedding</i> with other state-of-the-art methods on adapting GTA5 [6] to Cityscapes [7] datasets. We outperform other state-of-the-art methods with the Resnet101-based [8] segmentation networks. IT: Image-translation, Stru: Structure embedding loss, ST: Self-training.	80
5.3	Ablation Studies of <i>Task-in-all Pipeline</i> on the task of UDA semantic segmentation between GTA5 [6] and Cityscapes [7] dataset. We show that our Task-in-all pipeline outperforms sequential training pipeline in semantic segmentation in the target domain images. . . .	84
5.4	The comparison between <i>Global Structure Embedding</i> and <i>Task-in-all Pipeline</i> approaches. We compare the two regularization approaches from different perspectives such as training cost, target model to regularize and the final segmentation performance.	85

Symbols and Acronyms

Symbols

$\mathbb{R}^C$	the C -dimensional 1-D Euclidean space
$\mathbb{R}^{W \times H \times C}$	the $W \times H \times C$ -dimensional 3-D Euclidean space
$\mathbf{w}^\top \mathbf{x}$	the inner product of two vectors $\mathbf{w}$ and $\mathbf{x}$
$\mathcal{H}$	reproducing kernel Hilbert space
$\ \cdot\ _1$	the 1-norm in Euclidean space
$\mathbb{E}$	Expectation Value

Acronyms

AE	Adversarial Erasing
ASPP	Atrous Spatial Pyramid Pooling
CAM	Class Activation Mapping [2]
CRF	Conditional Ransom Field
DA	Domain Adaptation
DCNNs	Deep Convolutional Neural Networks
denseCRF	fully-connected CRF [9]
DR	drop-out rates
EM	Expectation-Maximization
FCN	Fully Convolutional Networks
FOV	Field-Of-View
GANs	Generative Adversarial Networks
GRL	Gradient Reversal Layer
MIL	multiple instance learning
mIoU	mean intersection-over-union
MMD	Maximum Mean Discrepancy
PSL	prohibitive segmentation learning

RKHS	reproducing kernel Hilbert space
SEC	Seed, Expand and Constrain [3]
SGD	Stochastic Gradient Descent
STC	Simple-to-Complex
TL	Transfer Learning
UDA	Unsupervised Domain Adaptation
WSSS	Weakly-supervised semantic segmentation supervised with image-level labels

Chapter 1

Introduction

1.1 Problem Background

Semantic segmentation is a pixel-wise classification task, which is to predict class label to every pixel within an image. By interpreting images into the segments representing different semantic-level meanings, semantic segmentation is a fundamental step to many computer vision applications. Recently Deep convolutional neural networks (DCNNs) have remarkably improved the performance in various applications that require a large amount of annotated data, such as image classification. The DCNNs structures have been widely extended to the pixel-wise prediction tasks including instance segmentation, saliency prediction and semantic segmentation with the emergence of Fully Convolutional Networks (FCN) structure. The DCNNs for the semantic segmentation are trained using the data with pixel-wise ground truth annotations. However, it is quite laborious or expensive to obtain a large amount of pixel-wise annotations for natural images. For example, it requires about 1 hour to pixel-wisely annotate an image from Cityscapes dataset. It is difficult to extend the segmentation methods to new training data or new semantic classes, hindering the developments and applications of semantic segmentation models in the real-world applications.

Thus, there is a growing interest towards reducing the data annotation effort for semantic segmentation tasks. From the view of data domain, the approaches could be roughly classified into intra-domain approaches and inter-domain approaches. The intra-domain approaches refer to the approaches that utilize weak supervision

formats instead of pixel-wise annotations for the natural images. Such supervision formats include but are not limited to scribbles, boxes and image-level labels, which are more efficient to annotate on natural images than pixel-wise annotations. The inter-domain approaches refer to the approaches which rely on the pixel-wise annotations, but from a different data domain whose data and annotations are more easily obtained than the original natural image domain. For example, the synthetic data and corresponding pixel-wise labels could be easily generated by computer graphics technology. The most intuitive solution is to train on the synthetic data with the same set of class labels to avoid the pixel-wise annotations on the natural images.

In our thesis, we investigate the approaches of both categories. For the intra-domain approaches, we study the case of weakly supervised semantic segmentation, which is supervised with image-level labels. Compared with other formats of supervisions, image-level labels are more efficient to annotate since it just indicates what semantic labels appear in the image. It is a hard setting since different from bounding box, point or scribble annotations, there is no clue of object localization and scale information provided by the image-level labels of the training data while the task is to estimate the pixel-level class prediction for the testing data. For the inter-domain approaches, we study the case of unsupervised domain adaptation (UDA) about semantic segmentation, whose setting is to train with the source domain data with pixel-wise annotations and target domain data without any label. We utilize synthetic images as the source domain data and the natural realworld images as the target domain data. It is a hard setting since there is no label information on the target domain, and there is obvious different domain distribution between the source and the target data.

One of the most important challenges in weakly semantic segmentation with image-level labels is how to estimate the locations and scales of target objects from image-level labels. On the cases that no external datasets are available, region-mining models are usually applied for this purpose. The region-mining model, or object localization model, is usually an image-classification model which could induce class-specific localization maps. The localization map is the heatmap that highlights the regions that are related to the specific image labels, which is an approximation of the target object localization and scales. However, region-mining methods usually only select the most discriminative parts, which deviate from our goal to estimate

the complete integral object regions. The main underlying reason is that the region mining models are trained solely from the classification loss thus targeting only the most discriminative parts is enough for the classification purpose. It is still an open problem on how to expand the highlighted region to the entire object regions.

Unsupervised domain adaptation (UDA) on semantic segmentation is a difficult task since there exists no clue of any label information in the target domain training data. In order to transfer the pixel-wise knowledge from source data to target data, the gap of the feature representation between the two domains should be bridged, which means that the feature representation should be generalized for different domains. However, in order to achieve classification task, the feature representation should be specific for each class label. Thus it is difficult to balance between the feature generalization and feature discrimination. Moreover, domain adaptation on pixel-level semantic segmentation is more challenging than the image-level image classification tasks since the pixel-level tasks face more complicated structure misalignment problems. Adversarial learning approaches are usually applied to UDA to mitigate the domain distribution gap between source and target datasets. Meanwhile, there may exist content-misalignment problems caused by the adversarial learning methods. The possible reasons for the content misalignment problems are as follows: First, the domain distribution of adversarial learning are calculated solely by aggregating the domain statistics over all the local patches, which neglects the global structure knowledge. Second, the pixel-level adversarial learning (*i.e.*, image translation) is not regularized by the semantic content loss (*e.g.*, the final segmentation loss). It is still an open problem how to alleviate the content misalignment problem of the UDA in semantic segmentation.

1.2 Major Contributions

The main contributions of the thesis are summarized as follows:

- *Decoupled Spatial Neural Attention for Weakly Supervised Semantic Segmentation*: In Chapter 3, for the task of weakly-supervised semantic segmentation we propose a region mining approach with decoupled attention structure for

generating pseudo-labels. Our decoupled attention model outputs two attention maps which focus on identifying object regions and mining the discriminative parts. These two attention maps are complimentary to each other to generate high-quality pseudo-annotations. This work has been published in the IEEE Transactions on Multimedia, 2019.

- *Splitting vs. Merging: Mining Object Regions with Discrepancy and Intersection Loss for Weakly Supervised Semantic Segmentation:* We aim to expand the highlighted region generated by the region mining model from a novel perspective of the optimization backward pass in Chapter 4. We propose a discrepancy loss which aims to mine out localization maps of different spatial patterns. It leads to a splitting effect of localization maps. We propose an intersection loss which aims to regularize the splitting phenomena caused by the discrepancy loss. It leads to a merging effect of localization maps. The combination of splitting and merging effect leads to expansion of the highlighted region of the localization maps. This work has been submitted to the IEEE International Conference on Computer Vision and Pattern Recognition, 2020.
- *Semantic Content regularization for Unsupervised Domain Adaptation in Semantic Segmentation:* In Chapter 5, we aim to regularize the adversarial learning for UDA in semantic segmentation. We propose two regularization approaches. The first is to embed the global structure knowledge into the feature-level adversarial learning step. The second is to back-propagate the final task loss into the pixel-level adversarial learning step. We further compare the pros and cons of the two regularization approaches in Chapter 5. This work has been published as two papers in the 2019 IEEE International Conference on Visual Communications and Image Processing (VCIP).

1.3 Thesis Outlines

Chapter 1 first introduces the definition of semantic segmentation and the need to reduce the pixel-wise annotation load for semantic segmentation. We introduce the goals and contributions of our approaches to alleviate the annotation load for semantic segmentation, including both the intra-domain and inter-domain approaches.

Chapter 2 reviews the related works, including the semantic segmentation, weakly-supervised semantic segmentation, unsupervised domain adaptation, etc.

Chapter 3 presents our work of *Decoupled Spatial Neural Attention*. It proposes a region-mining approach which regularizes the region mining model from the perspective of the forward pass.

Chapter 4 presents our work of *Splitting vs. Merging loss*. It proposes a region-mining approach which regularizes the region mining model from the perspective of the backward pass.

Chapter 5 presents our work of *Semantic Content regularization for UDA in semantic segmentation*. It aligns the segmentation content either by adding global structure loss in the adversarial training step, or back-propagating the task loss to the pixel-level adversarial training stage.

Chapter 6 summarizes and concludes the proposed methods in our thesis and discusses the potential research directions to investigate in the future works.

Chapter 2

Literature Review

2.1 Semantic Segmentation

Semantic segmentation is a pixel-wise prediction task whose goal is to estimate class label to every pixel in an image. DCNNs have remarkably improved the performance in various computer vision applications which require a large amount of training data with groundtruth labels, such as image classification. With the help of FCN structure, the DCNNs structures have been widely extended to the pixel-wise classification tasks including semantic segmentation. The semantic segmentation accuracy has been greatly improved with the help of development in DCNNs [10–17]. FCN [10] proposes to transform the fully connected layers in DCNNs to the convolutional layers for semantic segmentation tasks. Such FCN framework could be generalized to different pixel-wise prediction tasks including depth estimation or saliency detection.

One of the limitations of original FCNs is that the prediction score is high-level prediction of low spatial resolution, which will not lead to accurate high-resolution mask prediction. Some of the previous works focus on retaining the prediction masks of the high resolutions while still keeping its high-level semantic information. Deeplab-v2 [5] proposes to enlarge the dilated convolution rates to enlarge the respective fields. Noh *et al.* [11] propose deconvolution networks to upsample the prediction scores to recover the high resolutions.

Some previous works focus on aggregating context of different levels. Refinenet [12] incorporates information all different levels (from low level to high levels) into the proposed encoder-decoder structure to predict the segmentation masks. Deeplab [5] utilizes an atrous spatial pyramid pooling (ASPP) module to enclose context information of different scales.

2.2 Weakly Supervised Semantic Segmentation

Training DCNNs for semantic segmentation in a fully-supervised pipeline requires pixel-wise labels for training images, which is very difficult or label-intensive to obtain. Thus, weakly-supervised semantic segmentation receives much research attention to alleviate the workload of pixel-wise annotation for the training data.

In fully supervised semantic segmentation tasks, training images are accompanied with accurate pixel-wise annotations. Thus, the key challenge in fully supervised semantic segmentation is to design the segmentation structure which can completely mine out the relationship between the image and segmentation groundtruth. However, such accurate label formats do not exist in weakly supervised settings. Thus, the key challenges in weakly supervised semantic segmentation lie in the step of estimating the pixel-wise annotations from the weak labels and how to design the pipeline which is stable to the inaccurate estimated pixel-wise annotations. There are different formats of weak labels, such as scribbles [18], boxes [19], points [20] and image-level labels. Scribble and box annotations require the annotators to provide rough position and scales of the target objects and point annotations provides the center point of the objects. Image-label labels just provide the information of what semantic classes exist in a image without the need to provide any scale or position information of the groundtruth objects. Thus, among the weakly-supervised settings, image-level labels are the most label-saving and simplest to annotate. In this section we review the related works of Weakly Supervised Semantic Segmentation supervised image-level labels (WSSS).

Some early works [21, 22] treat WSSS task as multiple instance learning (MIL) setting which views each image being positive if at least one pixel/superpixel within it is positive, while negative if all the pixel/superpixel are negative. Other early works [23] apply Expectation-Maximization (EM) procedure, which alternates between

estimating pseudo-labels for the training images and optimizing DCNNs parameters. However, due to the lack of effective location cues, the performance of early works is not satisfactory enough. The performance of semantic segmentation with image-level labels was significantly improved after introducing location information to generate localization seeds/pseudo-annotations for segmentation DCNNs.

In the following section we revisit the previous works that incorporate localization seeds/pseudo-annotations. Based on whether external data sources are utilized, the approaches could be roughly classified into *Webly-Supervised Approaches* and *Region-mining Based approaches*.

2.2.1 Webly-Supervised Approaches

The approaches in this category utilize the additional easily obtained sources such as web sources into weakly supervised semantic segmentation tasks. Web images and videos are easily collected by indexing the category names. Web images always possess dominant foreground regions and clear background regions. The methods in this category assume that the pseudo-annotations of simple images (*e.g.*, web images) can be accurately estimated by saliency detection [24] or co-segmentation [25]. Thus, the Webly-Supervised Approaches usually follow the Simple-to-Complex (STC) strategy, in which the easy images (*i.e.* web images) could be utilized as the initialization point of training segmentation models/ estimating the pseudo-annotations. STC [24] predicts the masks for web images using saliency detection technique and trains the initial segmentation models using the web images and estimated masks. Then the initial segmentation model is applied on the target real images (*e.g.*, PSACAL VOC) to estimate the pseudo-masks for the target images. Shen *et al.* [26] proposes a bi-directional pipeline, in which the target image and web images help to compensate the segmentation results of each other. In this pipeline the target domain images remove the noisy labels in the web domains, while the web images help enhance the target segmentation results. Besides web images, other web sources could also be easily collected and assist the weakly supervised segmentation tasks. [27] crawls the web videos to incorporate their motion cues to accurately estimate the proxy-annotations for each video frame. Such motion cues help to accurately segment out the target moving objects which belong to the indexed labels in each frame, which improves the quality of

initial estimated pseudo-masks. Thus, the web sources could provide additional sources which could be utilized to improve the semantic segmentation results on the target datasets.

2.2.2 Region Mining Based approaches

Besides *Webly-Supervised Approaches* which utilize additional web image sources for estimating localization seed/pseudo-annotations, the seeds could also be estimated from the weakly labeled training images themselves. The region-mining methods are usually applied to estimate the pseudo-annotations on the basis that no additional data sources are available.

The region-mining model, or object localization model, is usually an image-classification model which could induce class-specific localization maps. The localization map is the heatmap that highlights the regions that are related to the specific image labels, which is an approximation of the target object localization and scales. However, region mining methods usually only select the most discriminative parts, which deviate from our goal to estimate the complete integral object regions. The main underlying reason is that the region mining models are trained solely from the classification loss thus targeting only the most discriminative parts is enough for the image classification purpose.

The previous weakly-supervised semantic segmentation focuses on different perspectives to cope with the problem of the in-complete localization seeds. We roughly classify the *Region Mining Based approaches* into the subcategories of *Localization Model based*, *Post-Processing based* and *Training Process based* approaches.

2.2.2.1 Localization Model based

The approaches in this category aims to train the region-mining model whose output localization maps could highlight the integral object regions instead of only the most discriminative regions. The early works [4, 28] usually follow the adversarial-erasing approaches, which needs to train region mining models in multiple sequential steps. Wei *et al.* [4] applies an adversarial-erasing approach which iterates between suppressing the most discriminative image region and training the region

mining model. It detects the most discriminative regions by the original region mining models, erases the most discriminative regions, and re-trains the region mining models to mine out the next discriminative regions. In this way, it gradually localizes the next most discriminative regions through multiple iterations and merges all the mined discriminative regions into final pseudo-annotations. Similarly Two-phase [28] captures the full extent of the objects by suppressing and mining processing the feature spaces in two phases. SeeNet [29] utilizes Conditionally Reversed Linear Units to reverse the signs of the feature maps according to the confident foreground/background region, which is another way to suppress the feature space following the similar adversarial erasing approach.

However, such multiple ad-hoc processing (*e.g.*, iterative training, adversarial erasing) approaches need to train the models in multiple times, which are difficult and time-consuming to implement. There are some works aiming to follow a more principal training pipeline, which is to train/inference the region mining model in a one step end-to-end manner. ACoL [30] and GAIN [31] enclose such erasing/suppressing operation in the training steps which switches the sequential pipeline into an end-to-end manner. Decoupled-net [32] proposes to enclose the spatial attention mechanism into the region-mining model and extend the regions by increasing the dropout rates of dropout layers. Among these methods [4, 29, 30], the common inherent idea is to suppress/contaminate the feature space to reduce its classification differentiation to force the classification model to highlight larger region to achieve classification purpose. Other works try to modify the structure of the deep model to highlight larger area. [33] enlarges the dilation rate of convolution layer to enlarge the respective field of the classification model to make the localization map highlight larger area.

Since the methods in this category are also closely related to the object localization/region mining tasks, in this section we also review some region-mining methods, which have been widely used in generating pseudo-annotations for semantic segmentation with image-level labels. Recent works of top-down neural saliency [2, 34, 35] perform well in weakly supervised localization tasks. Such works identify the discriminative regions respecting to each individual class based on image classification DCNNs. Zhang et al. [35] proposes Excitation Backprop to back-propagate in the network hierarchy to identify the discriminative regions. Zhou et al. [2] propose a region-mining or localization model named Class Activation

Mapping (CAM) for identifying discriminative parts by replacing fully-connected layers in image classification CNNs with convolutional layers and global average pooling layer. Grad-CAM [34] is a strictly generalization of CAM [2] without the need of modifying DCNN structure. Among the methods listed above, CAM [2] is the most widely used one in weakly-supervised tasks [3, 4, 28] for generating pseudo-annotations.

The methods in this category are also related to the spatial attention mechanism, since both the region mining methods or the spatial attention mechanism generate localization maps to highlight the feature regions for the final classification purpose. Spatial neural attention is a mechanism to assign corresponding weights to each spatial feature regions based on the feature content. It automatically outputs the soft heatmap to enhance the relevant feature grid and block the irrelevant feature grid during the training process for the purpose of training specific tasks. Spatial neural attention mechanism has been widely applied to many prediction tasks including image captioning [36, 37], machine translation [38], multi-label classification [39], human pose estimation [40] and saliency detection [41]. Intuitively, such weighted heat map could be applied to our purpose of pseudo-annotation generation.

2.2.2.2 Post-Processing based

The approaches in this category also aim to generate accurate pseudo-annotations, but focus on post-processing the output localization maps, which is independent of the training process of the region mining models. In this category, the output localization maps of the region mining models do not necessarily highlight the target objects but the post-processing step could compensate such inaccuracy.

The post-processing techniques usually rely on the low-level similarity cues to compensate the high-level feature discrimination of the localization maps. Decoupled-net [32] applies densecrf [9] on the coarse annotation masks to generate refined masks. Affinitynet [42] proposes to train an affinitynet to predict the inter-patch similarity of the localization seeds and applies random walk refinement technique on the localization seeds to make the annotation closer to object range.

2.2.2.3 Training Process based

The approaches in this category utilize other off-the-shelf region mining models to generate localization seeds/pseudo-annotations, which are usually inaccurate or incomplete to highlight the regions. To compensate such inaccuracy/incompleteness, the approaches in this category investigate training optimization processes for the segmentation model which is more stable to the inaccurate/incomplete supervision seeds. The most intuitive pipeline is to iterate between training the segmentation model and estimating the pseudo-annotations, which is also called self-training approach [4, 23, 33]. With such self-training strategy, it gradually predicts more accurate pseudo-annotations/localization seeds which leads to more accurate segmentation models. The improved segmentation model in-turn helps to generate more accurate pseudo-annotations, which is similar to EM (Expectation-Maximization) algorithms.

Besides the intuitive solution, there are other adaptive training pipelines which dynamically expand the seed supervisions in the stage of training segmentation model. SEC [3] proposes a pipeline which incorporates expansion loss, CRF constraints loss to the original segmentation loss in training the segmentation model. Similarly [43] dynamically grows the initial incomplete discriminative seeds into the larger object region in the training process. [3] and [43] utilize the low-level similarity cues for such seed expansion purpose in a dynamic pipeline.

2.3 Unsupervised Domain Adaptation

Domain Adaptation (DA) could be regarded as a subcategory of Transfer Learning (TL). We first briefly introduce Transfer Learning to highlight the role of domain adaptation in Transfer Learning. Transfer learning (TL) is a research problem which focuses on training models on source domain/task and evaluating with different but related domain or tasks. The term *domain* refers to the data feature space and its corresponding marginal probability distribution. The term *task* refers to the label sets and the corresponding objective predictive function. Domain Adaptation (DA) is a subcategory of TL. In Domain Adaptation, the source tasks (*i.e.*, features) are same as the target tasks (*i.e.*, features). However, there exist different marginal probability distributions between source/target domain settings.

Based on the different levels of supervision, domain adaptation could be further classified into *supervised DA*, *semi-supervised DA* and *unsupervised DA*. All the three subcategories are under the setting that the label for the source data are available, but differ in the settings of the target data. *Supervised DA* is defined as the setting that both source data and target data are fully labeled. *Semi-supervised DA* is defined as the settings in which the source data is completely labeled while the target data is only partly labeled. *Unsupervised DA* is defined as the setting that labeled data is available for the source data and the target data is totally unlabeled.

This thesis focus on the unsupervised domain adaptation (UDA) since it requires the least amount of annotation on the target domain. Next we review the parts of the methods that are widely used in the Unsupervised DA, including the subcategories of *Discrepancy-based*, *Adversarial-based* and *Self-training*.

2.3.1 Discrepancy based

The approaches in this category assume that domain shifts between two domains could be reduced by finetuning the models on the source and target data. The most commonly used method is to align the domain distributions calculated by Maximum Mean discrepancy (MMD) [44–46]. MMD is one of the metrics to quantify the domain distribution mismatch in the latent space. The definition is as Eq. 2.1.

$$MMD(s, t)^2 = \sup_{\|\phi\|_{\mathcal{H}} \leq 1} \left\| \mathbb{E}_{x^s \sim s}[\phi(x^s)] - \mathbb{E}_{x^t \sim t}[\phi(x^t)] \right\|_{\mathcal{H}}^2 \quad (2.1)$$

in which x indicates the feature space, s and t respectively denotes the source or target domain. ϕ is a map function that projects the attribute distribution into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. $\|\phi\|_{\mathcal{H}} \leq 1$ defines a set of unit ball in the RKHS $\mathcal{H}$. Empirical estimation of MMD is as Eq. 2.2:

$$MMD(D_s, D_t)^2 = \left\| \frac{1}{M} \sum_{i=1}^M \phi(x_i^s) - \frac{1}{N} \sum_{j=1}^N \phi(x_j^t) \right\|_{\mathcal{H}}^2, \quad (2.2)$$

The domain shifts could be reduced by minimizing $MMD(D_s, D_t)^2$. The task loss (*e.g.* classification loss) is calculated on the fully labeled source domain. By

minimizing the $MMD(D_s, D_t)^2$ loss and the task loss together we could mitigate the domain gap while achieving the supervision tasks.

2.3.2 Adversarial Learning based

2.3.2.1 Generative Adversarial Networks

Recently Generative Adversarial Networks (GAN) has been widely applied to Domain adaptation tasks. Goodfellow *et al.* [47] proposes GAN for the purpose of synthetic image generation. GAN mainly consists of two components: Generator G and a Discriminator D . The Discriminator aims to distinguish whether the image is sampled from the realworld training image set or is a fake image output by the Generator. The Generator aims to confuse the Discriminator by generating the fake images that are indistinguishable from the real images. Thus, GAN follows a min-max adversarial optimization process for distinguishing real/fake images, which is formulated as Eq. (2.3)

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}(s)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - G(z))] \quad (2.3)$$

in which x represents the real images and z denotes the input noise for the Generator G .

GAN could be applied to Domain Adaptation following the idea to generate the image or feature space that is indistinguishable by the Discriminator. The methods could be roughly classified into pixel-level DA or feature-level DA, based on whether the GAN is applied on the pixel-level or the feature-level, to generate domain-invariant images or features. The pixel-level DA [48, 49], or image-translation method, is the task to translate a source image to a translated image that closely resembles the target distribution. The translated images are paired with the labels from the source images which appeal or simulate the case of sampling the labeled image from the target data. The generated translated images play the role of data augmentation on the target data. Then the task model is trained on the labeled translated images, which acts as if training the task model directly on the target domain. The feature-level DA [50–52] learns the domain-invariant features by aligning features from different sources to a domain-invariant space. Since the

data from different domains share the same domain-invariant distributions, the task classifier trained with the features of source labeled data may generalize well for the target domain.

It is worth noting that some previous surveys [53, 54] regard the pixel-level DA and feature-level DA as two different subcategories, namely *Domain Mappings* (or *Generative Models* in [54]) and *Domain-Invariant Feature Learning* (or *Non-Generative Models* in [54]). They further regard the *Discrepancy-based* in Sec. 2.3.1 as a special case of the latter subcategory since it operates on the feature level. The reasons for such way of classification are as follows: The *Domain Mappings* need to train another generative model (*i.e.*, image translation) besides the task model while the *Domain-Invariant Feature Learning* directly operates on the inter-media feature space of the task model. On the contrary, we focus on the mechanism to close the domain distribution gap no matter which level of feature it operates on or whether it needs to train an additional generative model or not.

2.3.2.2 Gradient Reversal Layer

Besides GAN, Gradient Reversal Layer (GRL) [55] is another method for feature alignment among different domains. GRL model mainly consists of following components: a feature extractor, a task classifier and a domain classifier. The source/-target domain share the same weights in the feature extractor module. The task classifier is optimized on the source domain since the labeled data on only available for the source domain. Domain classifier is applied on the output of the feature extractor which predicts/distinguishes the binary label to differentiate source domain feature and target domain feature. Similar to GAN, feature extractor is trained to mitigate the domain discrepancy between the source and target feature, which is optimized in the opposite direction to the domain classifiers. Such opposite optimization direction is achieved by reversing the gradient of the domain classifier when back-propagating to the feature extractor. Thus, the domain classifier is optimized to distinguish the feature domain label while the feature extractor is optimized in the opposite direction to confuse the domain classifier. Similar to GAN, The feature extractor plays the role of Generator while the domain classifier plays the role of Discriminator. Compared with GAN, GRL follows a more principal optimization pipeline while GAN alternates between optimizing Generator and Discriminator.

Next we compare the Discrepancy-based methods and Adversarial-based methods. They both utilize a metric to measure the domain discrepancy between the two domains and the Generator (or feature extractor) is optimized to minimize the domain discrepancy. The Discrepancy-based methods and Adversarial-based methods are different in the following aspects: First, the domain discrepancy in Discrepancy-based methods is directly represented by a formulation, while in the Adversarial-based methods the domain discrepancy is implicitly formulated by the Discriminator Structure. Second, in Discrepancy-based methods the domain discrepancy loss is directly back-propagated to the feature extractor to reduce the domain gap. In the adversarial-based methods, it performs min-max optimization process which iterates between optimizing the Generator and Discriminator in an adversarial manner. Based on the optimization process, Discrepancy-based is easier to train since it has an explicit discrepancy loss formulation and could be optimized in an end-to-end manner. On the contrary, Adversarial-based methods are more difficult to train since the adversarial training process is hard to optimize and there is no policy of designing the Discriminator structure. Based on the UDA results, it is hard to tell which one is better since empirically it is a task-specific case and there is hard to propose a UDA pipeline which generalizes well to all the cases.

2.3.3 Self-Training

The methods listed in Sec. 2.3.1 and Sec. 2.3.2 follow the similar pipeline of aligning the feature/input spaces between source/target domains and relying the ground-truth labels paired with the source data to train the task loss. On the contrast, *Self-training* is the method that focuses on estimating the labels for the target training data. With the initially estimated labels, a task network could be directly fine-tuned on the target domain data, which may eventually adapt the task model to output the prediction results which are more accurate than the initial labels. Since the labels are only available for the source data, the initially estimated target domain labels are noisy and unreliable. Thus, a plausible solution is to only keep pseudo-labels that are easy and of high confidence and neglect the unsure pseudo-labels when training the target task network.

Empirically, Self-training strategy is a effective approach to improve the final results, since it assumes that the DCNN models could adaptly improve the performance in the previous round of iterations, similar to the spirit of EM. It is also widely applied to the weakly-supervised or semi-supervised tasks. However, in practice such alternating optimization is difficult to converge to the optimal solution.

Intrinsically, the self-training strategy follows the same target of pixel-level DA (*i.e.*, *Domain Mappings* in [53]), since both of these approaches aim to simulate the case of training the task models on the fully-labeled target datasets. In order to simulate the fully-labeled target datasets, pixel-level DA aims to generate the translated images to simulate the target data while self-training strategy aims to simulate the groundtruth labels.

2.3.4 UDA in Semantic Segmentation

In theory the UDA method is general to many different tasks (*e.g.*, image classification, image detection, semantic segmentation), where the differences mainly lie in choosing the corresponding task-specific loss and datasets. However, the semantic segmentation for UDA is difficult than other tasks since it contains more complicated structure information for pixel-wise prediction. There are some UDA methods that are specific for semantic segmentation tasks. Pixel-level DA approaches aims to translate the source domain images to the translated images which are visually similar to the target domain. It performs data augmentation on the target domain and final task model could be directly optimized using the translated images. For example, Cycada [56] relies on CycleGAN [49] to perform image-translation step. Among the feature-level DA approaches, [50] is the first work to perform feature-level domain adaptation for urban scene semantic segmentation task. Besides conventional features that are extracted from intermediate layers of the DCNN structure, output (prediction map) could be also considered as a kind of feature which is extracted at the last DCNN layer. AdaptSeg [51] demonstrates that performing adversarial training on the low-dimensional output space of segmentation model is more effective than on the high-dimensional feature space for semantic segmentation. A reasonable explanation is that softmax outputs of semantic segmentation space is of low-dimension and contains more dominant

structured property such as spatial layout and local context. Self-training is also applied to UDA in semantic segmentation, such as [57]. Besides transferring the easy and confident classes from source to target domain, [57] also proposed to class-wise weight the segmentation loss for the purpose of alleviating the class-imbalance problems in the training data.

Chapter 3

Decoupled Spatial Neural Attention for Weakly Supervised Semantic Segmentation

In this chapter, we tackle the problem of weakly supervised semantic segmentation with image label annotations which belongs to the intra-domain methods introduced in Chapter 1. We follow the most intuitive pipeline of sequentially estimating the pseudo-annotations and training the segmentation models with the pseudo-annotations. Thus, the final segmentation performance is largely dependent on the quality of generated pseudo-annotations in the first step.

We propose a region-mining method to predict pseudo-annotations. To regularize the training process of region mining model, we focus on designing model structure which incorporates the dropout layers to randomly block the feature grid in the forward training pass. Thus, the method introduced in the chapter aims to regularize the region mining model from the perspective of the forward pass.

Inspired by spatial neural-attention for image captioning, we propose a decoupled spatial neural attention network to estimate pixel-wise pseudo-labels. Our decoupled attention structure could simultaneously identify the object regions and localize the discriminative parts which generates high-quality pseudo-annotations in one forward path. The generated pseudo-annotations lead to the state-of-the-art weakly supervised semantic segmentation results.

3.1 Introduction

Semantic segmentation is a task to predict class label to each image pixels. It lays the basis for many applications such as part segmentation [58] and fashion parsing [59–61]. Recently Deep convolutional neural networks (DCNNs) [5, 10, 12, 62, 63] have brought great improvement in semantic segmentation performance. Training DCNNs in a fully-supervised setting with pixel-wise ground-truth annotation dramatically improve semantic segmentation accuracy. However, the main limitation in such fully-supervised setting is that it is labor-intensive to obtain tremendous amounts of pixel-level annotations for training images. Meanwhile, image-level annotations are much more efficient to annotate. Therefore, weakly-supervised semantic segmentation supervised with image-level annotations (WSSS) has received much attention.

The performance of WSSS has been remarkably improved by introducing efficient localization cues [3, 4, 64]. The most widely applied pipeline is to first estimate pseudo-annotations for the training images based on localization cues and then utilize the pseudo-annotations as the ground-truth for training the segmentation DCNNs. Clearly the quality of pseudo-annotations directly influences the performance of the final segmentation model. In our work, we follow the same pipeline and mainly focus on the first step, which is to estimate high-quality pseudo-annotations for the training data with only image-level labels. In recent years, top-down neural saliency [2, 34, 35] performs well in weakly-supervised localization tasks and consequently has been widely applied in generating pseudo-annotations for semantic segmentation supervised with image-level labels. However, as is pointed out by previous works [4], such top-down neural saliency is good at identifying the most discriminative regions of the objects instead of the whole extent of the objects. Thus the pseudo-annotations generated by these methods are far from the ground-truth annotations. To alleviate this problem, some works consist of multiple ad-hoc processing steps (*e.g.*, iterative training), which are difficult to implement. Some works introduce external information (*e.g.*, web data) to guide the supervision, which greatly increase data and computation load. On the contrary, our work proposes a principal pipeline which is simple and easy to implement.

Our aim is to generate pseudo-annotations for weakly supervised semantic segmentation efficiently and effectively. Following the spatial neural attention mechanism

FIGURE 3.1: A brief introduction of our proposed decoupled attention structure. D denotes the channel dimension and W, H denotes the spatial dimension of the feature space. It simultaneously generates two attention maps, namely Expansive Attention map and Discriminative Attention map. Expansive attention map is to identify object regions while the Discriminative attention map is to mine discriminative parts.

which has been investigated applied in VQA [65] and image captioning [37], we are inspired to introduce spatial neural attention into our pseudo-annotation generation pipeline and propose a decoupled spatial neural attention structure which simultaneously localizes the discriminative parts and estimates object regions in one end-to-end framework. Such structure helps to generate effective pseudo-annotations in one forward pass. The brief description of our decoupled attention structure is illustrated in Fig. 3.1.

We summarize our major contributions as follows:

- We introduce spatial neural attention and propose a decoupled attention structure for generating pseudo-annotations for weakly supervised semantic segmentation.
- Our decoupled attention model outputs two attention maps which focus on identifying object regions and mining the discriminative parts respectively. These two attention maps are suitable for identifying object regions under different circumstances and combination of the two maps will further improve the quality of attention maps.
- We employ a simple and effective pipeline without heuristic multi-step iterative training steps, which is different from most existing methods of weakly-supervised semantic segmentation.

FIGURE 3.2: The structure of conventional spatial neural attention model.

- We perform detailed ablation studies to verify the effectiveness of our decoupled attention structure. Finally our methods achieve state-of-the-art weakly supervised semantic segmentation results on Pascal VOC 2012 datasets.

3.2 Methods

First we give briefly review of the conventional spatial neural attention model in Sec. 3.2.1. Second we introduce our decoupled attention structure in Sec. 3.2.2. Then we introduce how to further generate pseudo-annotations based on this decoupled attention model in Sec. 3.2.3. Finally, the generated pseudo-annotations are utilized as the ground-truth annotation for training the final segmentation DCNN networks.

3.2.1 Conventional Spatial Neural Attention

In this section, we briefly introduce the conventional spatial neural attention mechanism. The conventional spatial neural attention mechanism is illustrated in Fig. 3.2. Attention structure mainly consists of two modules: the main branch modular and attention detector modular. The attention detector modular is jointly trained with the main stream modular end-to-end.

Formally we denote the features output by convolutional/pooling layers as $\mathbf{X} \in \mathbb{R}^{W \times H \times D}$. The attention detector takes $\mathbf{X} \in \mathbb{R}^{W \times H \times D}$ as the input feature and outputs spatial-normalized attention weights map $\mathbf{A} \in \mathbb{R}^{W \times H}$. $\mathbf{A}$ is applied on $\mathbf{X}$

to generate attended feature $\hat{\mathbf{X}} \in \mathbb{R}^{W \times H \times D}$. $\hat{\mathbf{X}}$ is fed into the classification modular to output image classification score vector $\mathbf{p} \in \mathbb{R}^C$, where C denotes the number of classes to classify.

For the purpose of notation efficiency, we denote the 3-D feature map of DCNNs with upper-case letters and denote the corresponding feature vector at one spatial location with its corresponding lower-case letters. For example, $\mathbf{x}_{ij} \in \mathbb{R}^D$ denotes the feature vector at the spatial position (i, j) of $\mathbf{X}$. Attention detector outputs attention weights map $\mathbf{A}$ which acts as a spatial regularizer to enhance the relevant regions and suppress the non-relevant regions for feature $\mathbf{X}$. Thus we are motivated to utilize the output of attention detector to generate pseudo-annotations for WSSS.

The details of the attention detector modular are as follows: Attention detector modular sequentially consists of a convolutional layer, a non-linear activation layer (Eq. (3.1)) and a spatial-normalization (Eq. (3.2)) as is shown in Fig. 3.2:

$$z_{ij} = F(\mathbf{w}^\top \mathbf{x}_{ij} + \mathbf{b}), \quad (3.1)$$

$$a_{ij} = \frac{z_{ij}}{\sum_{i,j} z_{ij}}, \quad (3.2)$$

where F is a non-linear activation function, such as exponential function in [37, 40]. $\mathbf{w} \in \mathbb{R}^D$ and $b \in \mathbb{R}^1$ denote the parameters of the 1×1 convolution layer in the attention detector modular. The attended feature $\hat{\mathbf{x}}_{ij} \in \mathbb{R}^D$ is calculated as

$$\hat{\mathbf{x}}_{ij} = \mathbf{x}_{ij} a_{ij}, \quad (3.3)$$

The classification modular consists of a spatial average pooling layer and 1×1 convolutional layer as the image classifier. We denote the $\mathbf{v}_c$ and h_c as the parameters of the classifier for class c , thus the classification score for class c is calculated as:

$$p_c = \left(\sum_{i,j} \mathbf{x}_{ij} a_{ij} \right)^\top \mathbf{v}_c + h_c, \quad (3.4)$$

where p_c is the score for c -th class.

FIGURE 3.3: Our proposed decoupled attention structure. The heatmap generated by Expansive attention detector is named *Expansive Attention map*, which identifies the object regions. The heatmap generated by the Discriminative Attention detector is named *Discriminative Attention map*, which localizes the discriminative parts.

3.2.2 Decoupled Attention Structure

As described in Sec. 3.2.1, the output of conventional spatial neural attention detector is class-agnostic. However, in semantic segmentation task each image may contain objects of multi-classes. Thus conventional class-agnostic attention map is not applicable to generate pseudo-annotations for such multi-class case since we need to predict pixel-wise label for each semantic class instead of just foreground/background. However, conventional spatial neural attention detector is aimed to assist the task of image classification which may not necessarily generate desired pseudo-annotations for the task of WSSS. Inspired by [39], we propose our decoupled attention structure specifically for the task of generating pseudo-annotations for weakly supervised semantic segmentation to alleviate these problems.

We illustrate our structure in Fig. 3.3. Such structure extends the conventional attention structure to multi-class cases. Moreover, it generates two different types of attention maps which identifies object regions and predicts the discriminative parts respectively. In Fig. 3.3, the attention map $\mathbf{A} \in \mathbb{R}^{W \times H \times C}$ generated by the Expansive attention detector in the top branch is named **Expansive attention map**, which aims at identifying object regions. The attention map $\mathbf{S} \in \mathbb{R}^{W \times H \times C}$ generated by the Discriminative attention detector in the bottom branch is named **Discriminative attention map**, which aims at mining the discriminative parts. Such two attention maps have different properties which are complementary to each

other to generate pseudo-annotations for weakly supervised semantic segmentation. The final attended map $\hat{\mathbf{S}}$ is generated by element-wise multiplication of these two attention maps and then spatially averaged to obtain image classification score.

Such attention structure is also similar to CAM [2] since it generates C channel pixel-wise prediction score (attended map) and then performs global average pooling to obtain $1 \times C$ image-level classification score in order to train with image-level label supervision. The c_{th} channel of the attended map is generated by element-wise multiplication of the c_{th} channels of class-specific attention maps, which are Expansive attention map and Discriminative attention map. In order to generate such two class-specific maps with C channels, 1×1 convolutional layers with C output channels are applied both in Discriminative and Expansive attention detectors.

The details of the structure in Fig. 3.3 are illustrated as follows:

Expansive Attention detector (E-A detector) modular consists of a convolutional layer, a non-linear activation layer (Eq. (3.5)) and a spatial-normalization step (Eq. (3.6)):

$$z_{ij}^c = F(\mathbf{w}_c^T \mathbf{x}_{ij} + b^c), \quad (3.5)$$

$$a_{ij}^c = \frac{z_{ij}^c}{\sum_{i,j} z_{ij}^c}, \quad (3.6)$$

where the superscript/subscript c denotes the value of c -th channel/class.

As we mentioned in Sec. 3.1, for the task of pseudo-annotation generation we aim to estimate the whole range of objects instead of only obtaining the most discriminative parts. Thus we design the E-A detector details as follows: we set

$$F(x) = \log(1 + \exp(x)) + \epsilon, \quad (3.7)$$

where $\epsilon = 0.1$, similar to [66]. Besides the 1×1 convolutional layer in the attention detector, we add one drop-out layer right before and after it respectively. The function of drop-out layers could be explained as follows: Drop-out layers randomly set the feature grids to zeros by a rate which consequently suppress the feature space by noise. Such zero-out process is operated directly on the CNN feature space at random spatial locations and feature dimensions. If the discriminative features are suppressed by drop-out process, the training process will mine other discriminative

features for classification purpose. Thus the attention model will adapt to highlight more relevant features instead of the most discriminative ones.

The Discriminative attention detector (D-A detector) consists of 1×1 convolutional layer whose parameters are denoted as $\mathbf{v}_c$ and h_c same as the classifier in Sec. 3.2.1. The D-A detector takes feature map $\mathbf{X}$ as the input and outputs class-specific attention map $\mathbf{S} \in \mathbb{R}^{W \times H \times C}$.

The attended map $\hat{\mathbf{S}} \in \mathbb{R}^{W \times H \times C}$ is calculated as:

$$\hat{s}_{ij}^c = s_{ij}^c a_{ij}^c, \quad (3.8)$$

$\hat{\mathbf{S}}$ is spatially average pooled to generate image classification score $\mathbf{p} \in \mathbb{R}^C$. Hence p_c is calculated as:

$$p_c = \sum_{i,j} (\mathbf{x}_{ij}^\top \mathbf{v}_c + h_c) a_{ij}^c. \quad (3.9)$$

Compared with Eq. (3.4), in our decoupled model the D-A detector is to predict the class score for dense pixel position instead of predicting image label score. Thus our model remains the spatial information for the classification map which is more suitable for semantic segmentation tasks. The multi-label classification loss for each image is formulated as:

$$L_{oss} = - \sum_c \left[y_c \log\left(\frac{1}{1 + e^{-p_c}}\right) + (1 - y_c) \log\left(\frac{e^{-p_c}}{1 + e^{-p_c}}\right) \right], \quad (3.10)$$

where y_c is the binary image label corresponding to c -th class.

To visualize the multi-channel attention map of one image, we separately visualize each map channel corresponding to the image-level label. Specifically, to visualize each map channel we follow the same method of visualizing single channel attention maps. We show visualization examples of Expansive attention map and Discriminative attention map in Fig. 3.4, which illustrates that Expansive attention maps perform well at identifying large object regions while Discriminative attention maps perform well at mining the small discriminative parts. We observe that for large objects, the Expansive attention maps are likely to cover integral object regions while the discriminative attention maps only highlight parts of the objects. On the contrary, for the small objects, the Discriminative attention maps are likely to enclose the object region while the Expansive attention maps tends to leak into

FIGURE 3.4: Visual examples of Expansive attention map and Discriminative attention map. Expansive attention map and Discriminative attention map have different properties which are suitable for different situations. The Expansive attention maps show superior performance for large objects (2nd and 4th rows) while the Discriminative attention maps are superior for small objects (1st and 3rd rows).

the background parts. Such two attention maps show different and complementary properties since they show superior property of estimating object range in different situations. It is intuitive to weighted sum the two attention maps based on the approximation of the global object range and confidence. The classification score in the decoupled spatial attention network could be utilized as the approximation of object range and confidence. In the decoupled spatial attention network, the classification score is calculated as the summation of pixel-wise scores. Thus, the classification score is directly related to the number of high pixel-wise scores to sum, which include both the factor of object range (i.e. number to sum) and the factor of pixel-wise confidence (i.e., value to sum). Thus, large classification score is likely to be resulted from the large confidence objects. As for the small classification score, it is either resulted from small object range or low pixel-wise confidence, both two cases prefer large weights of Discriminative attention map for the purpose of preferring small object range or deleting the noisy low-confident regions. Thus,

we merge such two attention maps using Eq. (3.11)

$$\mathbf{T}_c = \hat{p}_c * \bar{\mathbf{A}}_c + (1 - \hat{p}_c) * \bar{\mathbf{S}}_c, \quad (3.11)$$

where $\bar{\mathbf{A}}_c$ is the min-max normalized result of the linear-rectified Expansive attention map and $\bar{\mathbf{S}}_c$ is the min-max normalized result of the linear-rectified Discriminative attention map corresponding to c -th class. $\mathbf{T}_c$ is the result merged attention map. $\hat{\mathbf{p}}$ is the softmax normalization result of the image classification score $\mathbf{p}$. Such weighted combination is intuitive: the small prediction score usually corresponds to the difficult objects of small size so more weights should be put on the task of mining the discriminative regions.

3.2.3 Pseudo-annotation Generation

In this section we describe our approaches to estimate pseudo-annotations according to the attention maps. We generate pseudo-annotations by simple thresholding following the similar practice used in [2–4]. Given an attention map of an image with image label set $\mathcal{L}$ (excluding the background), for each class c in $\mathcal{L}$, we generate foreground regions as follows: first we perform min-max spatial normalization on the linear-rectified attention map corresponding to class c to $[0,1]$ range. Then we threshold the normalized attention map by >0.2 following the popular settings of [3, 29, 33]. Inspired by [67], we sum the feature map $\mathbf{X}$ in the channel dimension and then perform min-max spatial normalization on it. Then we threshold this normalized map by <0.3 to generate background regions. Since the regions are generated independently for each class, there may exist label conflicts at some pixel positions. We choose foreground label with smallest region size for the conflict regions following the practice of [3]. We assign the unclassified pixels with void label denoted as *void* which represents the label is unclear at this position and may not be considered in the training process.

The generated hard annotations M are coarse and have much unclear area. We can further apply denseCRF [9] on M to generate refined annotations. We first describe how to generate the class probability vector for each spatial locations. For an image I , the labels which are present in I are denoted as $\mathcal{C}_I$. All the class labels in the target dataset is denoted as $\mathcal{C}$. m_i is the mask label of M for pixel i . We calculate the class probability vector $z_{i,c}$ for class $c \in \mathcal{C}$ at pixel i as follows:

If $m_i = \text{void}$,

$$z_{i,c} = \begin{cases} \frac{1}{|\mathcal{C}_I|}, & \text{if } c \in \mathcal{C}_I \\ 0, & \text{otherwise} \end{cases} \quad (3.12)$$

Else:

$$z_{i,c} = \begin{cases} \tau, & \text{if } c = m_i \\ \frac{1 - \tau}{|\mathcal{C}_I| - 1}, & \text{if } c \neq m_i \text{ and } c \in \mathcal{C}_I \\ 0, & \text{otherwise} \end{cases} \quad (3.13)$$

where $\tau \in (0.5, 1)$ is the manually fixed parameter. $|\mathcal{C}_I|$ represents the number of labels that are present in image I . The unary potential is calculated for the probability vector. We apply denseCRF [9] with the unary potential and take the result mask as the refined annotations.

3.3 Experiments

3.3.1 Experimental Details

Dataset and Metrics We utilize the PASCAL VOC 2012 image segmentation benchmark [68] to evaluate our methods. PASCAL VOC 2012 dataset has 21 semantic classes including background. The original dataset has 1464 training images (*train*), 1449 validation images (*val*) and 1456 testing images (*test*). We augment the datasets to 10582 training images by [69] following the common practice. We only utilize image-level class labels of the training images in our experiments. We perform ablation studies on the *val* images. We utilize the standard metric of mean intersection-over-union (mIoU) to evaluate our final segmentation results.

Training/Testing Settings We train the proposed decoupled attention network for pseudo-annotation estimation. Then We train the segmentation network based on the generated pseudo-annotations to predict the final segmentation result.

We build the proposed decoupled attention model based on *VGG-16* model. We transfer the layers from *VGG-16* from the first layer to layer *relu5_3* as the starting

convolutional layers which outputs $\mathbf{X} \in \mathbb{R}^{14 \times 14 \times 512}$ (as is shown in Fig. 3.3). We train our decoupled attention model using the mini-batch size as 15 images with the data augmentation methods such as random flip and random scale. We set the 0.01 as initial learning rate for the *VGG-16* transferred layers and 0.1 as the initial learning rate for the attention detector layers. The learning rate is decreased by a factor of 10 after 10 epochs. Training terminates after 20 epochs.

We train the DeepLab-LargeFOV (*VGG-16* based) of [5] as our segmentation model. The segmentation model tasks the image crop of size 321×321 and outputs the segmentation masks of size 41×41 . The initial base learning rate set as 0.001 and it is decreased by a factor of 10 after 8 epochs. Training terminates after 12 epochs. We use the public available pytorch implementation of Deeplab-largeFOV¹. In the inference phase of segmentation, we perform multi-scale inference to combine output score at different scales following the common practices in [12, 70]. The segmentation outputs are post-processed by denseCRF [9].

3.3.2 Ablation Study of Decoupled Attention Model

3.3.2.1 Properties of attention maps

As described in Sec. 3.2.2, our decoupled attention model outputs Expansive attention map to identify object regions and Discriminative attention map to localize the discriminative parts. We qualitatively and quantitatively compare these two attention maps generated by our decoupled attention structure and demonstrate their differences.

First we provide visual examples of the Expansive attention maps and Discriminative attention maps of *train* images in Fig. 3.5 (column 2 and column 3). In Fig. 3.5 we provide some examples including images with single/multiple objects and single/multiple class labels. We observed that for simple cases (*e.g.*, large objects), Expansive attention map is likely to cover whole region of objects, while Discriminative attention map is likely to locate the most discriminative part. We also observed that for difficult cases (*e.g.*, small objects), Expansive attention map is likely to identify a broad region, while Discriminative attention map is likely to precisely localize the object. Thus we draw the conclusion that these two attention

¹https://github.com/BardOfCodes/pytorch_deeplab_large_fov

FIGURE 3.5: Visualization of attention maps. The fifth column shows the pseudo-annotations generated on the merged attention maps by hard-thresholding. The sixth column shows the pseudo-annotations generated on the merged attention maps after denseCRF refinement. The last column is the real ground-truth annotation for comparison. For the images with multiple classes we show the attention maps separately for each class. We observe that our merged attention maps combine the complementary information of Expansive attention map and Discriminative attention maps and the generated pseudo-annotations are close to the groundtruth.

maps have different properties and are suitable for different situations. They are potentially complementary to each other. Thus, the combination of the two maps will result in attention maps that are applicable to different cases which lead to object annotations of high quality.

We also quantitatively compare these two attention maps. We apply our decoupled

TABLE 3.1: Quantitative evaluation of attention maps. mS_{prec} is the criteria about localizing the concentrated interior object parts, mS_{rec} is the criteria about covering over the object regions and mS_{IoU} is the criteria for identifying the whole object regions. The results show that Expansive attention map (expan-atten) performs well at covering over and identifying the whole object regions and Discriminative attention map (disc-atten) performs well at localizing interior partial object regions. The merged attention map (merged-atten) performs best at identifying the whole object regions.

	expan-atten	disc-atten	merged-atten
mS_{prec}	52.0	61.2	59.4
mS_{rec}	71.9	54.4	65.7
mS_{IoU}	42.1	39.4	44.0

attention model on *val* images and generate annotation masks without denseCRF refinement for Expansive attention map and Discriminative attention map. We evaluate the generated estimated masks regarding to the groundtruth of *val* images.

We propose three evaluation measures for the estimated masks to demonstrate the different properties of the two attention maps:

1) We use the commonly used IoU to evaluate the performance of identifying the whole objects, which are denoted as:

$$S_{IoU} = \frac{true\ pos.}{true\ pos. + false\ pos. + false\ neg.} \quad (3.14)$$

2) We propose a criteria to evaluate the localization precision within the object region, which are denoted as:

$$S_{prec} = \frac{true\ pos.}{true\ pos. + false\ pos.} \quad (3.15)$$

3) We propose a criteria to evaluate the localization recall over the object region, which are denoted as:

$$S_{rec} = \frac{true\ pos.}{true\ pos. + false\ neg.} \quad (3.16)$$

S_{prec} emphasizes localizing the concentrate and small regions within the objects which not necessarily cover the whole object regions. S_{rec} emphasizes the expansion of the highlighted regions by measuring whether it includes the whole range of the objects which not necessarily localize within object regions. S_{IoU} emphasizes the accuracy of identifying the whole object regions which is the final criteria to indicate the quality of the attention map for generating pseudo-annotations. We use the

average score over all classes, which are denoted as mS_{IoU} , mS_{prec} and mS_{rec} as our criteria.

The evaluation results are listed by the first two columns in Table 3.1. It shows that Expansive attention map gets higher mS_{rec} and mS_{IoU} score, while Discriminative attention map gets higher mS_{prec} . This verifies different properties of attention maps: Discriminative attention map is likely to localize partial interior object regions while Expansive attention map highlights the regions of larger expansion and is likely to identify the whole object region.

As is described in Sec. 3.2.2, we can merge these two attention maps to improve the quality of estimated pseudo-annotations. The two attention maps are merged as Eq. (3.11) and we use the mS_{IoU} criteria to evaluate. The results are listed in the last column in Table 3.1. We observe that mS_{IoU} of merged attention map is higher than only using single type of attention map, which demonstrates that Expansive attention map and Discriminative attention map are complementary to each other in localizing the whole object regions. The pseudo-annotations generated from the merged attention map are more similar to the ground-truth maps, as is shown in Fig. 3.5.

TABLE 3.2: Evaluation of different drop-out rates (DR). With the increase of drop-out rates, mS_{rec} constantly increases while mS_{prec} constantly decreases. It shows that large DR enhance the expansion effect of attention maps to some extent.

DR	0	0.3	0.4	0.5	0.6	0.7
mS_{prec}	57.3	57.1	56.7	56.2	55.1	51.8
mS_{rec}	64.5	65.7	67.6	69.6	71.6	74.1
mS_{IoU}	42.6	43.1	43.7	44.0	44.2	42.8

3.3.2.2 Evaluation of drop-out layers

As described in Sec. 3.2.2, we add dropout layers in the Expansive attention detector modular to have expansion effect. In this section we evaluate the effect of drop-out layers.

In our experiments we use the default drop-out rate of 0.5. We also evaluate other drop-out rates and follow the same criteria as Table 3.1 on the merged attention maps. The results are listed in Table 3.2. We observe that with the increase of

FIGURE 3.6: The visual examples of different drop-out rates (DR). It shows that using a larger drop-out rate ($DR = 0.7$) leads to expansion effect over the using a smaller drop-out rate ($DR = 0.3$).

drop-out rates, mS_{rec} constantly increases while mS_{prec} constantly decreases. This demonstrates that increasing the drop-out rates helps expand the identified region from the interior parts to the entire object scales. We also display the visualization examples of the attention maps trained with different drop-out rates in Fig. 3.6, which demonstrate the expansion effect results from larger drop-out rates.

As we pointed out in Sec. 3.2.2, the effects of drop-out layers could be explained as follows: Drop-out layers randomly set the feature grids to zeros by a rate which consequently suppress the feature space by noise. If the discriminative features are suppressed by drop-out process, the training process will mine other discriminative features for classification purpose. It is easy to implement and effective to expand the highlight regions.

Next we investigate the influence of the number of dropout layers. We make the ablation study as follows: we denote the dropout layer between the Feature $\mathbf{X}$ and 1×1 conv as DP_A , and the layer between 1×1 conv and Non-linear activation

TABLE 3.3: Evaluation of the number of dropout layers. We observe that adding dropout layers expansion effect (higher mS_{rec}). The results also show that if only using one drop-out layer, the dropout layer before the conv layer (DP_A) plays more important role over then one after conv layer (DP_B).

Dropout layers	Non	DP_B	DP_A	$DP_A + DP_B$
mS_{prec}	57.3	55.3	59.9	56.2
mS_{rec}	64.5	67.5	65.3	69.6
mS_{IoU}	42.6	42.5	44.1	44.0

function as DP_B . We experiment on different numbers of dropout layers and evaluate the merged attention map following the criteria in Sec. 3.3.2. The results are presented in Table 3.3. We come to the following observations: 1) If only using one drop-out layer, DP_A outperforms DP_B by a great margin in terms of mS_{IoU} . Thus DP_A plays more important role over DP_B in improving the annotation quality. 2) The highest mS_{IoU} is achieved by using two dropout layers ($DP_A + DP_B$) or using DP_A alone. We can choose either of the two settings to train our attention model. Considering using two dropout layers achieves better expansion effect (higher mS_{rec}), we choose two dropout layers in our case.

3.3.2.3 Evaluation of decoupled structure

In this section, we aim to show that our decoupled attention structure is effective. We implement the case of removing the Expansive attention detector modular and only train the remaining Discriminative attention detector modular. We add one drop-out layer before and after the convolutional layer of Discriminative attention modular respectively. Then we evaluate the Discriminative attention map following the same criteria as Table 3.1. The result of mS_{prec} , mS_{rec} and mS_{IoU} are 50.9%, 57.3% and 35.9%, respectively. These results are far lower than our decoupled attention in Table 3.1. It verifies that our decoupled attention structure is more effective than the single-stream attention.

3.3.2.4 Evaluation of merging parameters

To show the effect of the merging strategy of Eq. 3.11, we further reverse the weight of Expansive attention map and Discriminative attention map and report the evaluation results of the merged attention map following the same criteria as

Table 3.1. The result of mS_{prec} , mS_{rec} and mS_{IoU} are 54.1 % , 62.4% and 39.7%, respectively. The results are lower than our proposed merging strategy of Eq. 3.11.

To better illustrate the relationship between object range and classification score, we provide more visual examples with small object range and low classification score in Fig. 3.7. It shows that small classification score usually corresponds to small object range (1st to 3rd rows). There are some exceptions. For example, in the 4th row, the range of person class is large, but the classification score is still small, which may wrongly put more weights on the Discriminative attention map. However, it does not obviously reduce the quality of attention map since the difference between the two attention maps is not obvious. Based on the observation, we argue that the original weighted summation could help avoid highlighting the noisy background regions for tiny objects.

FIGURE 3.7: Visual Examples for the correlation between low classification score and small object range.

TABLE 3.4: Comparisons with Region-Mining Based Methods. Both SEC [3] and AE [4] are based on the region-mining model of CAM [2]. We outperform SEC and AE w/o PSL with various settings. Our methods are also more efficient than SEC and AE since we only applies denseCRF in the post-processing step and we only train the region-mining model once.

Methods	mIoU
SEC [3]	
L_{seed}	45.4
$L_{seed} + L_{constrain}$	50.4
$L_{seed} + L_{constrain} + L_{expand}$	50.7
AE w/o PSL [4]	
AE-step1	44.9
AE-step2	49.5
AE-step3	50.9
AE-step4	48.8
Ours	
Expansive attention map	54.1
Discriminative attention map	52.2
merged attention map	55.4

3.3.3 Comparisons with Region-Mining Based Methods

Our approach is also relevant to region-mining based weakly supervised semantic segmentation approaches. We compare with two region-mining based approaches, *i.e.*, SEC [3] and Adversarial-Erasing (AE) [4]. Both SEC and AE use region-mining methods CAM [2] to generate pseudo-annotations. CAM [2] is a method to mine the discriminate object regions by image-level labels, which is related to our attention based methods.

SEC [3] contains three losses to optimize: L_{seed} is the segmentation loss based on pseudo-annotations. L_{expand} is for expanding the localization seed to object scale the and $L_{constrain}$ is to make the segmentation results agree with the denseCRF [9] result. AE [4] iteratively mine the object regions through multiple iteration steps by erasing the most discriminative regions in the input images then re-train CAM localization network to mine the next most discriminative regions. The mined regions of all the iterations are combined as the final pseudo-annotations. We list segmentation results of the middle steps of SEC and AE in Table 3.4. For SEC, we show the segmentation results with different combination of losses. For AE,

we list the segmentation results using different number of erasing steps without prohibitive segmentation learning (PSL).

For our methods, we list the segmentation results using the pseudo-annotations generated by Expansive attention map, Discriminative attention map and merged attention map in Table 3.4. Since merged attention map achieves better performance over the other two attention maps, we will use the segmentation results generated by merged attention map for further experiments by default. We outperform SEC and AE w/o PSL with various settings, which indicates our attention based region mining approach are significantly more effective than the CAM based mining methods. Moreover, our method employed a simple pipeline without complicated iterative training in contrast with AE. It is also worth noting that SEC embed denseCRF [9] refinement into the training pipeline while we only applies denseCRF once for processing refinement. AE applies multiple erasing/mining iterations of training while we only trains the region-mining model and segmentation model one time. Thus our methods are much more efficient than the SEC [3] and AE [4].

For more fair comparison with CAM [2], we report the segmentation results using the pseudo-annotations generated by CAM in Table 3.5. We experiment on two kinds of localization cues of CAM: The first one is generated by our re-implementation of VGG16-CAM model in [2] and the second one is the CAM localization cues provided by SEC [3]. All the pseudo-annotations are generated by performing denseCRF on localization cues following the same setting as ours. We observe that our attention model outperform CAM in both cases in estimating pseudo-annotations which lead to higher *mIoU* segmentation score (55.4 *vs.* 47.8 & 47.5).

3.3.4 Comparisons with State-of-the-arts

We compare our segmentation results with other weakly supervised semantic segmentation works. The comparison results are presented in Table 3.7 and Table 3.8 for *val/test* images respectively. We divide weakly supervised semantic segmentation into three categories based on different levels of weak supervisions and whether additional information (data/supervision) is implicitly used in their pipeline:

TABLE 3.5: Comparison between CAM and our attention model. We compare with two kinds of localization cues of CAM: The first one is generated by our re-implementation of VGG16-CAM model in [2] and the second one is the CAM localization cues provided by SEC [3]. We report the final segmentation results. Our attention model outperform CAM in estimating pseudo-annotations which lead to higher $mIoU$ segmentation score.

localization cues	$mIoU$
CAM cues (our own implementation)	47.8
CAM cues (provided by SEC [10])	47.5
cues of merged attention map (ours)	55.4

TABLE 3.6: Our Segmentation results for each class on *val* and *test* images. We list the results of using Deeplab-largeFOV (vgg16) [5] and ResNet-101 based Deeplab-like model (resnet) [1] as segmentation DCNN model.

	background	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table	dog	horse	motorbike	person	potted plant	sheep	sofa	train	tv/monitor	$mIoU$
vgg16- <i>val</i>	84.5	73.3	23.4	66.4	42.7	48.7	76.9	61.0	66.0	21.8	65.2	37.1	70.0	62.5	62.0	58.9	41.1	71.2	32.4	58.7	39.0	55.4
vgg16- <i>test</i>	85.6	71.2	27.1	74.2	35.8	51.0	77.7	63.4	65.0	21.4	60.0	47.7	73.9	60.2	71.0	61.5	39.2	64.0	39.5	54.2	40.0	56.4
resnet- <i>val</i>	84.7	73.5	27.1	68.3	43.8	57.2	80.9	64.7	65.4	22.3	69.7	39.2	70.8	73.0	67.8	60.0	45.8	72.6	35.9	57.7	42.2	58.2
resnet- <i>test</i>	86.3	70.4	29.7	78.8	40.8	53.9	80.6	67.7	67.7	21.0	70.2	46.4	74.8	70.6	74.8	63.7	45.2	76.9	45.0	57.8	39.8	60.1

In the first category, the methods utilize interactive input such as scribble [18] or point [20] as supervision, which is relatively more precise indicator of object location and scale. Similar to our methods, ScribbleSup [18] and BoxSup [19] also follow the pipeline of first estimating pseudo-annotations and training segmentation model with the pseudo-annotations. Since scribble and box annotations provides relatively accurate information of object range at the cost of annotation time, the segmentation results outperform the methods supervised with image-level labels. The results in this category are listed in the first block of Table 3.7 and Table 3.8.

In the second category, the methods only use image-level label as supervision, but introduce information of external sources to improve segmentation results. The segmentation performance in this category are listed in the second block of Table 3.7 and Table 3.8. Some works use additional data to assist training. For example, STC [24] crawled 50K web images for initial training step and Crawled-Video [27] crawled large amount of online videos, which significantly increase the training data amount. Some work implicitly utilize the pixel-wise annotations in their training. For example, TransferNet [72] transfer the pixel-wise annotation from MSCOCO dataset [73] to PASCAL dataset. AF-MCG [74] utilize MCG proposal method

TABLE 3.7: $mIoU$ of Weakly supervised semantic segmentation (WSSS) on *val* images. We divide the WSSS into three categories based on different levels of weak supervisions and whether additional information (data/supervision) is implicitly used in their pipeline. Our methods achieves the state-of-the-art results in the WSSS supervised only with image labels without external sources.

Methods	<i>val</i>	Comments
Interactive Input		
ScribbleSup[18]	63.1	scribble annotation
ScribbleSup(point)[18]	51.6	point annotation
What't the point [20]	46.1	point annotation
BoxSup [19]	62.0	Box annotation
Image-level labels		
with external source		Additional Info
STC [24]	49.3	Web images
Co-segmentation [25]	56.4	Web images
Webly-supervised [71]	53.4	Web images
Crawled-Video [27]	58.1	Web videos
TransferNet [72]	52.1	MSCOCO [73] pixelwise label
AF-MCG [74]	54.3	MCG proposal
Joon <i>et al.</i> [64]	55.7	Supervised Saliency
DCSP-VGG16 [75]	58.6	Supervised Saliency
DCSP-ResNet-101 [75]	60.8	Supervised Saliency
Mining-pixels [76]	58.7	Supervised Saliency
AE w/o PSL [4]	50.9	Supervised Saliency
AE-PSL [4]	55.0	Supervised Saliency
Image-level label		
w/o external source		
MIL-FCN [21]	25.7	
EM-Adapt [23]	38.2	
DCSM [77]	44.1	
BFBP [67]	46.6	
SEC [3]	50.7	
Two-phase [28]	53.1	
AF-SS [74]	52.6	
Anirban <i>et al.</i> [78]	52.8	
Ours-VGG16	55.4	
Ours-ResNet-101	58.2	

which is trained with PASCAL VOC pixel-wise ground-truth. Some works utilize fully-supervised saliency detection methods in localization seed expansion [64] or foreground/background detection [4, 24], which implicitly utilize external saliency ground-truth data to train saliency detector.

TABLE 3.8: *mIoU* of Weakly supervised semantic segmentation (WSSS) on *test* images. We divide the WSSS into three categories based on different levels of weak supervisions and whether additional information (data/supervision) is implicitly used in their pipeline. Our methods achieves the state-of-the-art results in the WSSS supervised only with image labels without external sources.

Methods	<i>test</i>	Comments
Interactive Input		Annotation
BoxSup [19]	64.2	Box annotation
Image-level labels with external source		Additional Info
STC [24]	51.2	Web images
Co-segmentation [25]	56.9	Web images
Webly-supervised [71]	55.3	Web images
Crawled-Video [27]	58.7	Web videos
TransferNet [72]	51.2	MSCOCO [73] pixelwise label
AF-MCG [74]	55.5	MCG proposal
Joon <i>et al.</i> [64]	56.7	Supervised Saliency
DCSP-VGG16 [75]	59.2	Supervised Saliency
DCSP-ResNet-101 [75]	61.9	Supervised Saliency
Mining-pixels [76]	59.6	Supervised Saliency
AE w/o PSL [4]		Supervised Saliency
AE-PSL [4]	55.7	Supervised Saliency
Image-level label w/o external source		
EM-Adapt [23]	39.6	
MIL-FCN [21]	24.9	
DCSM [77]	45.1	
BFBP [67]	48.0	
SEC [3]	51.7	
Two-phase [28]	53.8	
AF-SS [74]	52.7	
Anirban <i>et al.</i> [78]	53.7	
Ours-VGG16	56.4	
Ours-ResNet-101	60.1	

In the third category, the methods only use image-level label as supervision without the information of external sources. Our methods belong to this category. The segmentation performance in this category are listed in the third block of Table 3.7 and Table 3.8.

We mainly focus on comparing with the methods in the third category. We achieve the mIoU of 55.4 and 56.4 respectively on *val* and *test* images, which significantly

FIGURE 3.8: Qualitative results of PASCAL 2012 *val* images. We list the qualitative examples generated by Deeplab-largeFOV (Ours-VGG16) segmentation model and Resnet101 based (Ours-Resnet-101) [1] Deeplab-like segmentation model. It shows that our segmentation results are close to the groundtruth and the results predicted by the Resnet101-based segmentation model outperform the ones predicted by the VGG16-based model.

outperforms other methods using the same supervision settings. In order to further improve the segmentation accuracy, we apply Deeplab-like model in [1] based on

Resnet-101 [8] as the segmentation model. We achieve the mIoU score of 58.2 on *val* images and 60.1 on *test* images, which outperforms all the existing WSSS methods using the same level of supervision setting up to date. We also list our segmentation results for each class in the Table 3.6 for reference. We display the qualitative results of the segmentation masks in Fig. 3.8.

Moreover, we also want to emphasize that we do not iteratively train the models in multiple steps. The whole pipeline only needs training decoupled attention structure and segmentation structure once on the PASCAL VOC *train* data. To our knowledge our approach has the most simple pipeline for weakly-supervised semantic segmentation.

3.4 Summary

We have presented a novel decoupled spatial neural attention work to generate pseudo-annotations for weakly-supervised semantic segmentation which only relies on the image-level annotations in training images. This decoupled attention model simultaneously outputs two class-specific attention maps with different properties and are effective for estimating pseudo-annotations. We perform detailed ablation experiments to verify the effect of our decoupled attention structure. Finally the generated pseudo-annotations lead to the state-of-the-art weakly supervised semantic segmentation results on Pascal VOC 2012.

Chapter 4

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

In this chapter, we also tackle the problem of weakly-supervised semantic segmentation supervised with image-level labels. Same as Chapter 3, we also follow the pipeline of sequentially estimating the pseudo-annotations and training the segmentation models with the pseudo-annotations. In this chapter we propose a region-mining model methods for accurately estimating the pseudo-annotations.

To regularize the training process of region mining model, we focus on adding the regularization losses to the original classification loss. Such regularization losses are back-propagated to force the region mining model to mine-out more regions that satisfy the classification purpose. Thus, different from Chapter 3 which regularizes the region mining model in the forward pass, the method introduced in the chapter aims to regularize the region mining model from the perspective of the backward pass.

We propose a Splitting vs. Merging optimization strategy, which is mainly composed of the Discrepancy loss and the Intersection loss. The proposed Discrepancy

loss aims at mining out regions of different spatial patterns instead of only the most discriminative region, which leads to the splitting effect. The Intersection loss aims at mining the common regions of the different maps, which leads to the merging effect. Our Splitting vs. Merging strategy helps to expand the output heatmap of the region mining model to the object scale. Finally, by training the segmentation model with the masks generated by our Splitting vs Merging strategy, we achieve the state-of-the-art weakly-supervised segmentation results on the Pascal VOC 2012 benchmark.

4.1 Introduction

In this chapter we investigate weakly-supervised semantic segmentation supervised with image-level labels (WSSS). However, semantic segmentation supervised with image-level labels is a difficult task, since there is no localization and scale information of the ground-truth objects provided by the training images.

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

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

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

We summarize the contributions in this chapter as follows:

- We propose to expand the highlighted regions generated by the region mining model from a novel perspective of the backward pass.
- We propose a Discrepancy loss which aims to mine out localization maps of different spatial patterns. It leads to a splitting effect of localization maps.
- We propose an Intersection loss which aims to regularize the splitting phenomena caused by the Discrepancy loss. It leads to a merging effect of localization maps.
- Training the segmentation network with the pseudo-masks generated by our splitting vs. merging process, we achieve state-of-the-art results on weakly supervised semantic segmentation on Pascal VOC 2012 segmentation benchmark.

4.2 Methods

In this section, first we briefly revisit CAM [2]. Next we introduce our region mining approach with the proposed Discrepancy loss and Intersection loss, which is a Splitting vs. Merging process to expand the highlighted region of the localization maps. Finally we normalize the resultant localization maps and generate pseudo-annotations to train the final segmentation model.

4.2.1 Revisiting CAM

FIGURE 4.1: The brief revisiting of CAM [2] approach. It is a classification model with average pooling step to aggregate the patch-level score map into classification score. The resultant localization map S is obtained by training the model with classification loss.

We briefly review CAM [2], which is one of the most widely used region mining techniques and serves as the basis model for our pipeline. It utilizes global average pooling to aggregate the pixel-level prediction to the image-level score. For simplicity, we introduce an equivalent variance of CAM utilized in ACol [30], which is a more delicate and simple formulation than the original definition [2]. The structure of CAM is illustrated in Fig. 4.1. The model is sequentially composed of fully convolutional feature extractor and patch-level classifier, which outputs feature $X \in \mathbb{R}^{W \times H \times D}$ and patch-level score map $S \in \mathbb{R}^{W \times H \times C}$, respectively. H and W denote the height and width in the spatial dimension. D denotes the feature dimension and C denotes the number of the classes to classify. Spatial Average pooling is applied on S to aggregate the patch-level score into the image-level prediction score $s \in \mathbb{R}^C$. The whole network is finetuned by calculating and back-propagating the image-classification loss $L_{cls}(s)$. For the multi-label case we utilize MultiLabel Soft Margin Loss, which is a multi-label one-versus-all loss based on max-entropy. Score map S is the resultant class-specific localization map which highlights the corresponding region for each individual class. [30] has proved theoretically that such variance is equivalent to the original CAM [2] and can directly generate the localization map during the forward pass, instead of a separate post-processing step for the map generation.

4.2.2 Splitting vs. Merging

One of the limitations of region-mining approaches, including but are not limited to CAM [2], is that it is likely to only highlight the most discriminative parts instead of the integral object region. In this section, we propose to alleviate this problem by our Splitting vs. Merging pipeline.

FIGURE 4.2: The structure of our region-mining method. Our structure mainly consists of Reference Stream and Expanding Stream. Expanding Stream aims to expand the highlighted region of the localization maps. It follows an Splitting vs. Merging process, which is achieved by our Discrepancy loss (Disc loss) and Intersection loss (Cls loss inter). Discrepancy loss leads to a splitting effect on the localization maps while the Intersection loss leads to a merging effect. The combination of both losses helps to expand the highlighted regions in the union map output by Expansion Stream.

The structure of our model is illustrated in Fig. 4.2. Our structure is mainly composed of two streams: Reference Stream and Expanding Stream. The Reference Stream is the original CAM structure which makes sure that the most discriminative region is always mined out. The Expanding Stream aims to expand the localization maps to a larger object scale by mining out localization maps of different spatial patterns. Our main contributions lie in the Expanding Stream, which contains the Splitting vs. Merging strategy formed by our proposed Discrepancy Loss and Intersection Loss.

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

Besides the classification loss, in order to enforce S_A and S_B to have difference spatial patterns, we propose a Discrepancy loss to regularize S_A and S_B . The Discrepancy loss is depicted as *Disc loss* in Fig. 4.2. The Discrepancy loss is

defined as:

$$\mathcal{L}_{disc} = -\frac{1}{HWC} \sum_{i,j,c} \|z_{ijc}^A - z_{ijc}^B\| \quad (4.1)$$

where

$$z_{ijc}^A = \frac{e^{s_{ijc}^A}}{\sum_{i,j} e^{s_{ijc}^A}}, \quad (4.2)$$

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

Discrepancy loss $\mathcal{L}_{disc}$ could effectively generate two maps with distinct spatial patterns. However, the optimization process is likely to fall into a trivial solution of splitting the original discriminative region, if only using the classification loss and the Discrepancy loss $\mathcal{L}_{disc}$. To avoid such trivial solution, we add an Intersection loss to regularize the optimization. The Intersection loss is denoted as *Cls loss inter* in Fig. 4.2. Our Intersection loss is defined as follows: we calculate S_{inter} as the element-wise minimum value between S_A and S_B . Average pooling operation is applied on S_{inter} to get the image classification score s_{inter} . The Intersection loss is calculated as the classification loss on s_{inter} , which is denoted as $\mathcal{L}_{cls}(s_{inter})$. By adding Intersection loss $\mathcal{L}_{cls}(s_{inter})$, we force the input maps of the Discrepancy loss (*i.e.*, S_A and S_B) to have large overlapping high-lighted area, which results in a merging effect of the localization maps. Optimizing with both the Discrepancy and the Intersection losses, we follow a Splitting vs. Merging pipeline which forces the Expansion Stream to mine out larger highlighted regions.

The Reference Stream is the original CAM with Classifier C which outputs the localization map S_C to highlight confident discriminative regions. It complements the Expanding Stream in case that the localization maps S_A and S_B highlight different but less discriminative regions in which the most discriminative part is neglected. Same as CAM, the Reference Stream is optimized by only the classification loss denoted as $\mathcal{L}_{cls}(S_C)$.

The final optimization objective is formulated as

$$f_T^* = \arg \min_{\mathbf{G}, \mathbf{w}^a, \mathbf{w}^b, \mathbf{w}^c} [\mathcal{L}_{cls}(S_A) + \mathcal{L}_{cls}(S_B) + \mathcal{L}_{cls}(S_C) + \mathcal{L}_{cls}(s_{inter})] + \beta * \arg \min_{\mathbf{w}^a, \mathbf{w}^b} \mathcal{L}_{disc} \quad (4.3)$$

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

4.2.3 Mask Generation

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

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

We utilize denseCRF [9] post-processing approach to estimate the hard annotations from normalized localization maps. The unary term of DenseCRF for each foreground class is the normalized localization maps M (*i.e.*, M_A , M_B and $U(S_A, S_B)$, etc.). Since the normalized map only indicates the probability for each foreground class, we need to estimate the probability of the background class. Similar to [42]. The background probability M_{bg} is calculated as $M_{bg} = (1 - M_{fore})^\alpha$, where M_{fore} is the foreground probability and α is the parameter to decide the weight of the background class. We utilize the normalized saliency score $M_{sal} \in (0, 1)$ and the normalized localization map M to calculate M_{fore} as $M_{fore}(i, j) = \max(\max_c M_{ijc}, M_{sal}^{ij})$, which is a channel-wise maximum operation on M followed by an element-wise maximum operation with M_{sal} . M_{bg} and M are concatenated as the unary term of the denseCRF to generate the hard pseudo-annotations for the training images. Then, we train the segmentation models using the pseudo-annotations.

In the testing stage, we directly apply the segmentation model on the validation/testing images to predict the segmentation masks.

4.3 Experiments

4.3.1 Datasets and Implementation Details

We perform experiments on the PASCAL VOC 2012 datasets [68] which contains 21 semantic classes in total. Following the common practice, we augment the dataset to 10582 training images with [69] datasets. We report the segmentation results on 1449 validation images (*val*) and 1456 testing images (*test*) using mean intersection-over-union (mIoU) as the segmentation criteria.

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

In our experiments we utilize the saliency model in PoolNet [79] using the resnet50 backbone w/o edge model.

For the segmentation model, we utilize the Deeplab-v2 like model in [26] which is based on vgg-16 or resnet50 model. It is similar to Deeplab-v2 structure, but with a global average pooling stream. The input image is of the size 320×320 . The initial learning rate is set as $16e-4$ and are diminished by rate 0.1 after 8 epoches. We utilize multi-scale merging technique in the reference stage following the common practice. The final segmentation output is post-processed by denseCRF [9] methods.

FIGURE 4.3: Visualization results of the input localization maps (S_A and S_B) of the Discrepancy loss with weights $\beta = 10$ and $\beta = 50$. It shows that with the larger weight of Discrepancy loss, the spatial patterns of the input localization maps are more visually different.

4.3.2 Ablation Study

4.3.2.1 Properties of Mining Region

In this section we perform detailed analysis on our generated localization maps to show the effect of our proposed Splitting vs. Merging pipeline. Unlike the previous work [32] which needs to evaluate the hard-annotations transferred from the soft localization maps, we aim to directly evaluate the properties of the soft localization maps in a more elegant way which neglects the influence of other post-processing parameters such as the hard threshold. Thus we propose three evaluation criteria for localization map evaluation: $Soft_{overlap}$, $Soft_{pre}$ and $Soft_{rec}$.

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

$$Soft_{overlap} = \frac{\sum_{i,j} \min(M_{ijc}, G_{ijc})}{\sum_{i,j} \max(M_{ijc}, G_{ijc})}, \quad (4.4)$$

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

$$Soft_{pre} = \frac{\sum_{i,j} \min(M_{ijc}, G_{ijc})}{\sum_{i,j} M_{ijc}}, \quad (4.5)$$

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

$$Soft_{rec} = \frac{\sum_{i,j} \min(M_{ijc}, G_{ijc})}{\sum_{i,j} G_{ijc}}. \quad (4.6)$$

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

First, we show that our Discrepancy loss helps to generate two localization maps with different spatial patterns. We calculate the similarity score between the input maps of the Discrepancy loss, which are S_A and S_B . To calculate the similarity of the maps we rely on the $mSoft_{overlap}$ defined as Eq. 4.4 with the input maps replaced by M_A and M_B . The results are shown in Table 4.1. It shows that with the increase of the Discrepancy loss weight β , the similarity score between the two maps constantly decreases, which shows that the Discrepancy loss helps to generate two localization maps with different spatial patterns. We also provide visualization results of the input maps (*i.e.*, S_A and S_B) with different weight of the Discrepancy loss in Fig. 4.3. It clearly shows that the localization maps regularized by larger weight of Discrepancy loss are more visually different.

Next we investigate the expansion effect of the Discrepancy loss over the union maps. We generate map $U(S_A, S_B)$, which is the union map of the input maps of the Discrepancy loss, and evaluate its variance with the increase of Discrepancy loss weight. The results are listed in the lower block of Table 4.2. It shows that with the increase of Discrepancy loss weight (β), the recall score $mSoft_{rec}$ constantly

TABLE 4.1: The Evaluation of the splitting effect of Discrepancy loss. With the increase of the Discrepancy loss weight β , the similarity score between the two maps regularized by Discrepancy loss constantly decreases, which shows that the Discrepancy loss helps to generate two score maps with different spatial patterns.

β	0	10	20	50
similarity score	93.54	78.88	77.64	64.31

increases, which means it is likely to cover more of the target objects. The precision score $mSoft_{pre}$ constantly decreases, which shows that it is more likely to leak out of the boundary of target object regions. It shows that larger Discrepancy loss helps to expand the highlighted regions of the localization maps. For more intuitive understanding of the expansion effect caused by the Discrepancy loss, we provide the visual examples of the union maps $U(S_A, S_B)$ in Fig. 4.5. It shows that with the increase of the weight of the Discrepancy loss the expansion effect of the union map becomes more obvious.

For better understanding of the expansion effect, we provide visual results of the failure cases in Fig. 4.4. There are some cases that the expanded highlighted regions are likely to leak into the background region instead of covering the target objects. One case is that the foreground classes usually co-exists with background classes. For example, the foreground class of boat usually co-exist with background class of water. The other case is that the target objects is of small scale and there is no need of expanding the regions.

FIGURE 4.4: The visual examples of failure cases. The highlighted regions are likely to leak into the background region on the cases of frequent co-existence of foreground and background classes (the first row) and small foreground objects (the second row).

FIGURE 4.5: The visual examples of the union maps $U(S_A, S_B)$. It shows that with the increase of the weight of the Discrepancy loss (β) the expansion effect of the union map becomes more obvious.

TABLE 4.2: Evaluation of the union maps of the input maps of the Discrepancy loss on the *val* images. It shows that with the increase of the weight of the Discrepancy loss (β), the recall $mSoft_{rec}$ increases while the precision $mSoft_{pre}$ decreases, which shows the expansion effect of the Discrepancy loss. It also shows that adding the Intersection loss (Inter loss) helps the expansion effect more stable and obvious.

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

Third we investigate the influence of the Intersection loss. The ablation study results are listed in Table. 4.2. The upper block shows the results without intersection loss while the lower block shows the results with intersection loss. We observe that in general the Discrepancy loss helps expand the target region whether with

TABLE 4.3: Our segmentation results for each class on *val* images. We list the condition whether addition training data (web data) are utilized and whether supervised saliency (S-Sal) are utilized. We achieve the state-of-the-art performance in WSSS.

Method	web data	S-Sal	<i>val</i>
EM-Adapt [23]	-	-	38.2
BFBP [67]	-	-	46.6
DCSM [77]	-	-	44.1
SEC [3]	-	-	50.7
AF-SS [74]	-	-	52.6
Two-phase [28]	-	-	53.1
Anirban <i>et al.</i> [78]	-	-	52.8
Decou-Net[32](vgg16)	-	-	55.4
Decou-Net[32](resnet101)	-	-	58.2
Affinity [42] Deeplab	-	-	58.4
Affinity [42] Resnet-34	-	-	61.7
STC [24]	✓	✓	49.3
Co-segmentation [25]	✓	-	56.4
Webly-supervised [71]	✓	-	53.4
Crawled-Video [27]	✓	-	58.1
BoostTrap [26](vgg16)	✓	-	58.8
BoostTrap [26](resnet50)	✓	-	63.0
Joon <i>et al.</i> [64]	-	✓	55.7
DCSP-VGG16 [75]	-	✓	58.6
DCSP-ResNet-101 [75]	-	✓	60.8
AE-PSL [4]	-	✓	55.0
DSRG (vgg16) [43]	-	✓	59.0
DSRG (resnet101) [43]	-	✓	61.4
FickleNet (vgg16)[80]	-	✓	61.2
FickleNet (resnet101)[80]	-	✓	64.9
MDC [33]	-	✓	60.4
GAIN [31]	-	✓	55.3
SeeNet (vgg16) [29]	-	✓	61.1
SeeNet (resnet101)[29]	-	✓	63.1
Ours (vgg16)	-	✓	63.7
Ours (resnet50)	-	✓	66.6

or without the Intersection loss. However, without Intersection loss, the expansion effect is not stable or obvious enough, especially with large Discrepancy loss weight. One reasonable explanation for this phenomena is that if the Discrepancy loss weight is too large, the optimization with only the Discrepancy loss may likely to fall into a tricky solution which simply splits the original discriminative region. Thus the Intersection loss helps regularize the optimization to make the

TABLE 4.4: Our segmentation results for each class on *test* images. We list the condition whether addition training data (web data) are utilized and whether supervised saliency (S-Sal) are utilized. We achieve the state-of-the-art performance in WSSS.

Method	web data	S-Sal	<i>test</i>
EM-Adapt [23]	-	-	39.6
BFBP [67]	-	-	48.0
DCSM [77]	-	-	45.1
SEC [3]	-	-	51.7
AF-SS [74]	-	-	52.7
Two-phase [28]	-	-	53.8
Anirban <i>et al.</i> [78]	-	-	53.7
Decou-Net[32](vgg16)	-	-	56.4
Decou-Net[32](resnet101)	-	-	60.1
Affinity [42] Deeplab	-	-	60.5
Affinity [42] Resnet-34	-	-	63.7
STC [24]	✓	✓	51.2
Co-segmentation [25]	✓	-	56.9
Webly-supervised [71]	✓	-	55.3
Crawled-Video [27]	✓	-	58.7
BoostTrap [26](vgg16)	✓	-	60.2
BoostTrap [26](resnet50)	✓	-	63.9
Joon <i>et al.</i> [64]	-	✓	56.7
DCSP-VGG16 [75]	-	✓	59.2
DCSP-ResNet-101 [75]	-	✓	61.9
AE-PSL [4]	-	✓	55.7
DSRG (vgg16) [43]	-	✓	60.4
DSRG (resnet101) [43]	-	✓	63.2
FickleNet (vgg16)[80]	-	✓	61.9
FickleNet (resnet101)[80]	-	✓	65.3
MDC [33]	-	✓	60.8
GAIN [31]	-	✓	56.8
SeeNet (vgg16) [29]	-	✓	60.7
SeeNet (resnet101)[29]	-	✓	62.8
Ours (vgg16)	-	✓	64.5
Ours (resnet50)	-	✓	66.7

Discrepancy loss have more stable effect of the region expansion. We visually compare the localization maps generated with/without the Intersection loss in Fig. 4.6 and Fig. 4.7. It shows that without the Intersection loss, the large Discrepancy loss weight mainly leads to split a single discriminative region instead of having obvious expansion effect.

Fourth we show that our pipeline outputs the complementary localization maps.

TABLE 4.5: Our Segmentation results for each class on *val* images. We list the results of using Deeplab-largeFOV (vgg16) [5] and ResNet-50 based Deeplab-like model (resnet) [1] as segmentation DCNN model. We achieve the state-of-the-art performance in WSSS in most semantic classes.

	background	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table	dog	horse	motorbike	person	potted plant	sheep	sofa	train	tv/monitor	mIoU
SEC [3]	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
STC [24]	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
Video [27]	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
AE [4]	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
Boost [26]	86.8	71.2	32.4	77.0	24.4	69.8	85.3	71.9	86.5	27.6	78.940.7	78.5	79.1	72.7	73.1	49.6	74.8	36.1	48.1	59.2	63.0	
Weby [71]	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
Decou	84.7	73.5	27.1	68.3	43.8	57.2	80.9	64.7	65.4	22.3	69.7	39.2	70.8	73.0	67.8	60.0	45.8	72.6	35.9	57.7	42.2	58.2
Affinity [42]	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
vgg16-Ours	89.9	84.6	36.0	79.7	57.7	65.5	81.2	75.5	82.3	23.2	65.5	31.5	78.0	72.1	72.4	74.8	36.8	75.5	31.6	70.7	52.3	63.7
resnet50-Ours	90.485.638.9	78.9	62.073.483.7	74.3	82.9	25.8	77.8	30.1	81.179.376.1	73.9	38.6	85.0	32.7	72.8	55.7	66.6						

TABLE 4.6: Our Segmentation results for each class on *test* images. We list the results of using Deeplab-largeFOV (vgg16) [5] and ResNet-50 based Deeplab-like model (resnet) [1] as segmentation DCNN model. We achieve the state-of-the-art performance in WSSS in most semantic classes.

	background	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table	dog	horse	motorbike	person	potted plant	sheep	sofa	train	tv/monitor	mIoU
SEC [3]	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
STC [24]	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
Weby [71]	85.8	66.1	30.0	64.1	47.9	58.6	70.7	68.5	75.2	11.3	62.6	19.0	75.6	67.2	72.8	61.4	44.7	71.5	23.1	42.3	43.6	55.3
Video [27]	87.2	63.9	32.8	72.4	26.7	64.0	72.1	70.5	77.8	23.9	63.6	32.1	77.2	75.3	76.2	71.5	45.0	68.8	35.5	46.2	49.3	58.7
Boost [26]	87.2	76.8	31.6	72.9	19.1	64.9	86.7	75.4	86.830.0	76.6	48.5	80.5	79.9	79.7	72.6	50.1	83.5	48.3	39.6	52.2	63.9	
Decou [32]	86.3	70.4	29.7	78.8	40.8	53.9	80.6	67.7	67.7	21.0	70.2	46.4	74.8	70.6	74.8	63.7	45.2	76.9	45.0	57.8	39.8	60.1
Affinity [42]	89.1	70.6	31.6	77.2	42.2	68.9	79.1	66.5	74.9	29.6	68.7	56.182.1	64.8	78.6	73.5	50.8	70.7	47.7	63.9	51.1	63.7	
vgg16-Ours	90.4	78.7	34.5	82.2	50.7	63.8	76.5	74.1	80.1	24.6	69.7	35.4	77.7	77.1	78.3	74.9	46.6	78.5	34.8	70.5	54.4	64.5
resnet50-Ours	90.785.937.382.5	50.5	64.8	83.1	77.6	82.8	28.4	76.8	34.6	81.2	82.980.5	73.6	43.9	85.7	32.0	71.755.2	66.7					

The results are listed in Table. 4.7. We report the $mSoft_{overlap}$ score over different localization maps, such as S_A , S_B and S_C . It shows that the union of map S_A and S_B has higher $mSoft_{overlap}$ score than using single map alone. The union of all the three maps S_A , S_B and S_C shows the highest $mSoft_{overlap}$ score. Thus, we utilize the union of the three maps for pseudo-mask generation.

Finally, we generate pseudo-annotations following the practice in Sec. 4.2.3. We set the background parameter $\alpha=2$ and utilize the public model provided by [79] to generate saliency maps. We utilize the traditional intersection-over-union criteria $mIoU$ for hard masks as the final criteria to evaluate our generated annotations. The results are listed in Table. 4.8, which show that we achieve better quality of the

FIGURE 4.6: The visually comparison of the union localization maps generated with/without the intersection loss. The weight for Discrepancy loss (β) is 50. It shows that without the intersection loss, the large Discrepancy loss mainly leads to split a single discriminative region instead of having obvious expansion effect.

pseudo-masks over the case without Discrepancy loss. We utilize the pseudo-masks of the highest quality ($\beta=20$) to train the final segmentation models.

4.3.3 Segmentation Results

Finally, we train the segmentation model with the pseudo-masks generated by our localization maps and report our segmentation results in Table.4.3 and Table.4.4. We list the extra information knowledge that may improve the segmentation results

FIGURE 4.7: The visually comparison of the union localization maps generated with/without the intersection loss. The weight for Discrepancy loss (β) is 20. It shows that without the intersection loss, the large Discrepancy loss mainly leads to split a single discriminative region instead of having obvious expansion effect.

for the purpose of clear and fair comparison, such as whether extra training images are listed and whether supervised saliency methods are utilized. We achieve the state-of-the-art results in weakly supervised semantic segmentation. We also list the segmentation results in each category in Table. 4.5 and Table. 4.6, which show that we outperform other state-of-the-art performance in most of the semantic categories. We provide the visualization of result segmentation masks in Fig. 4.8, which show that our predicted segmentation masks are visually close to the segmentation groundtruth. We also provide the upperline results of training segmentation model

TABLE 4.7: $mSoft_{overlap}$ of the union of different localization maps on the *val* images. It shows that the localization maps (S_A, S_B and S_C) are complimentary to each other and the union of all the three maps achieves the highest $mSoft_{overlap}$ score.

β	0	10	20	50
S_c	22.25	23.65	23.56	24.36
S_A	20.96	22.48	22.8	24.47
S_B	20.93	22.71	23.33	24.96
$U(S_A, S_B)$	21.22	23.7	24.17	27.35
$U(S_A, S_B, S_C)$	24.57	26.58	27.07	28.57

TABLE 4.8: The evaluation of the generated pseudo-annotations with different Discrepancy weight β . We report $mIoU$ of the generated pseudo-annotations on the *train* images as the evaluation metrics of the pseudo-annotations.

β	0	10	20	50
$mIoU$	59.65	60.92	61.27	57.22

with pixel-wise groundtruth for comparison. The upperline $mIoU$ results on *val* data is 72.1/75.7 using vgg16/resnet50 base model, which shows that our WSSS results are competitive with the fully-supervised upperline results.

4.4 Summary

In this chapter, our goal is to propose a region-mining methods in order to generate pseudo-masks for weakly supervised semantic segmentation. We aim to propose the region mining approach to identify the integral object regions instead of only the most discriminative parts. In order to achieve this goal, different from the previous methods which contaminate the forward feature map of the region mining model, we tackle it from a novel perspective of the backward optimization pass. We propose a Splitting vs. Merging approach, which is mainly composed of a Discrepancy loss and an Intersection loss. Our Splitting vs. Merging pipeline effectively and efficiently expands the highlighted regions from the most discriminative parts to the integral object region. With the pseudo labels generated from our region mining models, we achieve the state-of-the-art weakly supervised semantic segmentation results on the PASCAL VOC12 benchmarks.

FIGURE 4.8: Qualitative results of PASCAL 2012 *val* images. We list the qualitative examples generated by Deeplab-largeFOV (Ours-VGG16) segmentation model and Resnet101 based (Ours-Resnet-50) [1] Deeplab-like segmentation model. It shows that our segmentation results are close to the groundtruth and the results predicted by the Resnet50 based segmentation model outperform the ones predicted by the VGG16-based model.

Chapter 5

Semantic Content Regularization for Unsupervised Domain Adaptation in Semantic Segmentation

Chapter 3 and Chapter 4 tackle the weakly-supervised semantic segmentation problems, which alleviate the need of pixel-wise annotation by utilizing the image-level annotations in the same data domain. Different from Chapter 3 and Chapter 4, this chapter focuses on the problem of semantic segmentation with unsupervised domain adaptation (UDA). UDA alleviates the pixel-wise annotation effort by utilizing the pixel-wise annotations of another data domain whose data/labels are much easier to generate.

In unsupervised domain adaptation, to narrow domain gap from different data domains, we follow the common pipeline which is based on the adversarial learning approaches, either in the pixel-level or feature-level. However, there may exist some content-misalignment problems in the adversarial learning pipeline due to the possible reasons. First, the result of pixel-level adversarial learning does not necessarily remain the semantic consistency of the input image. Second, global structure knowledge is not embedded to regularize the feature-level adversarial learning.

This chapter proposes two regularization approaches to regularize/align the content of the UDA semantic segmentation. The first approach is to embed the global structure knowledge into the feature-level adversarial training step to tackle the problem of structure misalignment. The second approach is to propose a task-in-all pipeline which back-propagates the final task loss into the pixel-wise adversarial learning step. Finally we compare and discuss the pros and cons of the two content regularization approaches.

5.1 Introduction

In this chapter we also aim to reduce the data annotation load for semantic segmentation tasks. Different from the previous Chapter 3 and Chapter 4 which utilize the image-level annotations instead of the pixel-level annotations in the real-image domain, in this chapter we utilize the pixel-level annotations, but from synthetic-image domain. The data and label of the synthetic domain could be easily obtained through computer graphics technology, which do not need time-consuming pixel-wise annotations. However, there exists obvious different domain distribution gap between source synthetic data and target real-world data. Thus the task model trained on the source labeled data may not perform well on the target data. Domain adaptation is one solution to mitigate the domain distribution difference between source/target domain data. In this work, we focus on the task of semantic segmentation with unsupervised domain adaptation. The setting is to train on the synthetic data with pixel-wise annotation (*i.e.*, source data) and unlabeled real-world data (*i.e.*, target data).

One of the widely-used pipelines to align the distribution between two domains is to apply domain adaptation in either pixel-level or feature-level. Both of the domain adaptation methods are effective for distribution alignment between the two domains. The most intuitive solution is a sequential training pipeline, which is to separately train the image-translation model and the task model for segmentation. Image translation model usually performs pixel-level adversarial learning, while the task model performs feature-level adversarial learning and task-supervised training. However, there are some content-misalignment problems of this sequential training pipeline. We refer the content-misalignment problem as the phenomena that the feature belonging to one semantic content wrongly possess the properties

of other semantic labels. Such misalignment problems could happen to different features levels, including the pixel-level and output-level. The pixel-level content misalignment problems refer to cases that the region of the output images shows the color/texture that obviously not belonging to its semantic labels. For example, the sky shows the color of the ground region. The output-level content misalignment problems refer to the cases of erroneous prediction of the segmentation masks.

As for our semantic segmentation task, we pay more attention to the output-level content-misalignment problems since our final goal is to accurately predict the segmentation masks for the target-domain images. Such content misalignment phenomena is mainly due to the following reasons: First, the domain distribution in adversarial learning is usually mainly calculated by aggregating the distribution statistics over all the local patches [50–52], which neglects the global structure knowledge between them. However, such structure knowledge is an important characteristic of images, especially for the urban scene datasets (e.g., GTA5 [6] and Cityscapes [7]). For example, sky will often appear above the road semantic label, buildings tend to exist besides roads. Such global structure knowledge is shared between source/target data and will potentially help to regularize the semantic content of the prediction. Second, some unsupervised pixel-level domain adaptation (image translation or style transfer) methods do not necessarily possess semantic content, as is stated in [56]. One of the reasons is that the aim of image translation is to generate visually satisfactory translated images, instead of facilitating the final tasks, not to mention that it is difficult to choose the most visually satisfactory results just based on subjective judgement. The training process of image translation is unrelated to the final semantic tasks. Consequently, the pixel-level content-misalignment error is propagated to the subsequent task model, which result in the low segmentation accuracy.

In the chapter, we propose two regularization approaches to alleviate the content misalignment problems in UDA of semantic segmentation. We name our two approaches as *Global Structure Embedding* and *Task-in-all Pipeline* respectively. *Global Structure Embedding* regularizes the content by embedding the global structure knowledge into the feature-level adversarial learning step. *Task-in-all Pipeline* proposes an end-to-end optimization pipeline to back-propagate the final task loss into the pixel-wise adversarial learning step.

The contributions of this chapter are listed as follows:

- In *Global Structure Embedding*, we propose a framework of domain adaptation in semantic segmentation which effectively combines image translation, feature-level domain adaptation and self-training strategy. We effectively embed global structure knowledge into the adversarial learning by adding structure embedding loss into the domain adaptation framework. Such structure embedding loss regularizes the feature-level adversarial learning step.
- In *Task-in-all Pipeline*, we propose our Task-in-all pipeline for unsupervised domain adaptation on semantic segmentation, which incorporates image-translation and final segmentation task into an end-to-end training pipeline. Our Task-in-all pipeline outperforms the sequential training pipeline using the image translation model and the final task model. Such Task-in-all pipeline regularizes the pixel-level adversarial learning step by back-propagating the semantic task-loss into the pixel-level adversarial learning step.

We introduce our *Global Structure Embedding* and *Task-in-all Pipeline* in Section 5.2 and Section 5.3 respectively. Then we demonstrate the experiments performance of each method and make comments on them.

5.2 Global Structure Embedding

In this section, we introduce our first UDA regularization approach *Global Structure Embedding*. It adds a structure embedding loss to alleviate the content misalignment problem, either propagated by the image-translation results or the local aggregation of domain distribution in the feature-level adversarial learning.

Our methods are targeted on training on the fully-labeled synthetic source domain images and unlabeled target domain images. Let $X_s = \{x_s, s = 1, 2, \dots, N_s\}$ and $Y_s = \{y_s, s = 1, 2, \dots, N_s\}$ denote the set of the source-domain images x_s and its corresponding segmentation ground-truth y_s . $X_t = \{x_t, t = 1, 2, \dots, N_t\}$ denotes the set of unlabeled target-domain images. Our aim is to rely on X_s , Y_s and X_t to predict the segmentation map for X_t .

Our pipeline is briefly described as follows: we first apply image translation to transform the source-domain images X_s to the translated images $\tilde{X}_s$ which are visually inseparable from the target domain. Then we apply domain adaptation on

FIGURE 5.1: Our proposed framework of *Global Structure Embedding*. Our framework is composed of three sequential steps: a) Image translation b) Feature-level domain adaptation. c) Self-training. In order to embed the global structure knowledge, we propose to add structure embedding loss in the feature-level adversarial training stage, besides the segmentation loss and local distribution loss. In the testing phase, we only utilize the final segmentation network, which is circled in orange. We delete details of Image translation step for brief illustration.

feature-level with our Structure Embedding technique on the $\tilde{X}_s$ to train the Mask model which aims to generate pseudo-annotations $\hat{Y}_t$ for X_t . Finally we utilize $\hat{Y}_t = \{\hat{y}_t, t = 1, 2, \dots, N_t\}$ and X_t to train the final segmentation model. The pipeline is depicted in Fig. 5.1.

5.2.1 Image Translation

In this section, we apply image-translation on the source-domain images X_s to output the translated images $\tilde{X}_s = \{\tilde{x}_s, s = 1, 2, \dots, N_s\}$ which are visually inseparable from the target data domain. For image-translation models we choose the state-of-the-art methods, such as CycleGAN [49] and UNIT [48]. Please refer to [48, 49] for detailed implementation.

5.2.2 Feature-level DA with Structure Embedding

In this section, we perform feature-level domain adaptation between the target-domain images and translated source-domain images in order to train a mask model to estimate pseudo-annotations for the target-domain images. Our structure is built on AdaptSeg [51], which is to perform domain adaptation on output space of the segmentation network. We first briefly review AdaptSeg [51], then describe how to embed structure knowledge in it. To avoid confusion, in this section we refer to the segmentation network/mask model as Generator.

Different from AdaptSeg [51] which takes X_s , Y_s and X_t as input, we apply image translation prior to feature-level adaptation. Thus, we replace X_s with $\tilde{X}_s$ as the input into the Generator. $\tilde{x}_s$ (*resp.* x_t) is input into a Generator to generate softmax output segmentation map $\tilde{p}_s$ (*resp.* p_t) $\in \mathbb{R}^{W \times H \times C}$, where C denotes the number of classes to predict. Same as AdaptSeg, a fully-convolutional discriminator network D with binary value output is trained to distinguish source (label 0) and target domain (label 1). The batch-size of source/target images is denoted as n_s and n_t respectively. The optimization loss for Generator is formulated as Eq. 5.1

$$\mathcal{L} = \mathcal{L}_{seg}(\tilde{x}_s, y_s) + \lambda_{adv} \mathcal{L}_{adv}(x_t) \quad (5.1)$$

where

$$\mathcal{L}_{seg}(\tilde{x}_s, y_s) = -\frac{1}{|n_s|} \sum_{h,w} \sum_{c=1}^C y_s^{(h,w,c)} \log \tilde{p}_s^{(h,w,c)} \quad (5.2)$$

and

$$\mathcal{L}_{adv}(x_t) = -\frac{1}{|n_t|} \sum_{h,w} \log(D(p_t)^{(h,w,0)}) \quad (5.3)$$

The optimization loss for Discriminator D is formulated as Eq. 5.4

$$\mathcal{L}_D = -\frac{1}{|n_s|} \sum_{h,w} \log(D(\tilde{p}_s)^{(h,w,0)}) - \frac{1}{|n_t|} \sum_{h,w} \log(D(p_t)^{(h,w,1)}) \quad (5.4)$$

in which h and w denote the spatial patch indices. The training process is to iteratively optimize Eq. 5.1 and Eq. 5.4. Although AdaptSeg [51] claims that output space contains structure information of an image, we observe in Eq. 5.3

and 5.4 that [51] only aggregates (sums) the local structure statistics over the all the patches, which neglects the global structure knowledge within an image. Thus we refer Eq. 5.3 and Eq. 5.4 as Local Distribution Loss in Fig. 5.1. Besides, as is stated in Sec. 5.1 the translated source images will also lose the global structure knowledge to some extent. Thus, we propose our structure embedding method to embed the such global structure knowledge into the training process. Specifically we built our structure embedding strategy as follows:

First, we resize $p_t \in \mathbb{R}^{W \times H \times C}$ into a smaller spatial size (W', H') and flatten them into a vector $p_t^* \in \mathbb{R}^{W'H'C}$. We transfer y_s into one-hot matrix and following the same operation to obtain $y_s^* \in \mathbb{R}^{W'H'C}$. Such vectorization operation preserves the global structure information of an image. Our goal is to minimize the distribution distance between y_s^* and p_t^* in order to transfer the embedded the global structure knowledge from source to target domain. It also aligns the distribution between the output space and groundtruth space.

To calculate the distribution distance, we utilize Maximum Mean Discrepancy (MMD) measure [44–46]. Specifically, our structure embedding loss is formulated as Eq. 5.5:

$$\mathcal{L}_{stru} = MMD(y_s^*, p_t^*)^2 = \left\| \frac{1}{n_s} \sum_{s=1}^{n_s} \phi(y_s^*) - \frac{1}{n_t} \sum_{t=1}^{n_t} \phi(p_t^*) \right\|_{\mathcal{H}}^2, \quad (5.5)$$

where ϕ is a map function which projects the attribute distribution into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. The arbitrary distribution can be represented using the kernel embedding technique [81]. Therefore, we have a kernel function $k(y_s^*, p_t^*) = \phi(y_s^*)\phi(p_t^*)$ induced by ϕ . Thus Eq. 5.5 can be expressed as Eq. 5.6

$$\mathcal{L}_{stru} = \frac{1}{(n_s)^2} \sum_{s=1}^{n_s} \sum_{s'=1}^{n_s'} k(y_s^*, y_{s'}^*) + \frac{1}{(n_t)^2} \sum_{t=1}^{n_t} \sum_{t'=1}^{n_t} k(p_t^*, p_{t'}^*) - \frac{2}{n_s n_t} \sum_{s=1}^{n_s} \sum_{t=1}^{n_t} k(y_s^*, p_t^*) \quad (5.6)$$

where we use well-know RBF characteristic kernel with bandwidth α .

$$k(y_s^*, p_t^*) = \exp\left(-\frac{1}{2\alpha}(y_s^* - p_t^*)^2\right) \quad (5.7)$$

Finally, the optimization loss for Generator model, namely the mask model in Fig. 5.1, is updated as Eq. 5.8

$$\mathcal{L} = \mathcal{L}_{seg}(\tilde{x}_s, y_s) + \lambda_{adv}\mathcal{L}_{adv}(x_t) + \lambda_{stru}\mathcal{L}_{stru} \quad (5.8)$$

5.2.3 Self-training

In this section, we utilize the mask model generated in Sec. 5.2.2 to generate pseudo-annotations $\hat{Y}_t = \{\hat{y}_t, t = 1, 2, \dots, N_t\}$ for the target image X_t to train the final segmentation model for the target data. $\hat{y}_t$ is estimated by $\hat{y}_t = \max_c p_t^{(h,w,c)}$ for all target training images, where p_t is the output of mask estimation model in Sec. 5.2.2. The training loss for segmentation model is formulated as Eq. 5.9, where $\hat{p}_t$ is the output of the final segmentation model.

$$\mathcal{L}_{seg}(x_t, \hat{y}_t) = -\frac{1}{|n_t|} \sum_{h,w} \sum_{c=1}^C \hat{y}_t^{(h,w,c)} \log \hat{p}_t^{(h,w,c)} \quad (5.9)$$

In the testing stage, we apply the final segmentation model on the target domain images to predict segmentation masks.

5.3 Task-in-all Pipeline

In this section, we introduce our second UDA regularization approach *Task-in-all Pipeline*. It is also based on the pipeline of applying both pixel-level and feature-level adversarial learning. *Task-in-all Pipeline* regularizes the pixel-level adversarial learning by back-propagating the task-specific loss into the pixel-level adversarial learning step through an end-to-end training pipeline.

We train with fully-labeled source domain synthesis images (X_s, Y_s) and unlabeled target real-world domain images X_r . $X_s = \{x_s \in \mathbb{R}^{H \times W \times 3}\}$, $Y_s = \{y_s \in \mathbb{R}^{H \times W \times C}\}$ and $X_r = \{x_r \in \mathbb{R}^{H \times W \times 3}\}$. s and r denote the domain indices of source and target data respectively. C denotes the number of categories. H and W denote the spatial sizes of the input images. y_s is the ground-truth mask of the source images.

FIGURE 5.2: The pipeline of our framework of *Task-in-all Pipeline*. Our framework mainly consists of two parts: Image translation stage and Segmentation stage. The Image translation stage performs pixel-level domain adaptation and the segmentation stage performs high-level domain adaptation and task prediction. Such framework is optimized in an end-to-end manner, which utilizes the final task loss to regularize image translation.

Our main goal is to train a semantic segmentation network f_T to predict the semantic segmentation map for the input target real-world images. We show our Task-in-all pipeline in Fig. 5.2. Our framework is mainly composed of two stages: Image translation stage and Segmentation stage. Image translation stage performs pixel-level domain adaptation and the segmentation stage performs high-level domain adaptation and task prediction. Our frameworks is inspired by T2Net [82] which introduces a wide-spectrum translation network for pixel-level domain adaptation. Instead of purely encouraging the translated images to be similar in image domain, we also train an auxiliary discriminator on semantic domain to encourage the predicted segmentation maps of two domains to a same distribution. In Fig. 5.2, the upper stream illustrates the training pipeline for the source domain data while the bottom steam illustrates the training pipeline for the target domain data. Such two streams share the same weights in both the translation network and task network. Specifically:

- The upper stream illustrates the training pipeline for source domain image-label pairs (x_s, y_s) . We translate the source image x_s to the target domain using GAN (Generative adversarial network) loss $\mathcal{L}_{GAN_R}$ to make it visually similar to the target domain. Then we input the translated image into the

task network f_T for the purpose of semantic segmentation. The task loss $\mathcal{L}_{seg}$ is calculated by comparing the segmentation map with the ground-truth y_s .

- The bottom stream illustrates the training pipeline for target domain x_r . In the image translation stage, the reconstruction loss $\mathcal{L}_r$ is calculated between the original image and the translated target image in order to recover the image in the target domain. In the Segmentation stage, GAN loss $\mathcal{L}_{GAN_P}$ is utilized to encourage the predicted semantic maps of the two domains to belong to same distribution.

The details of the framework are described as follows:

5.3.1 Image Translation Stage

We use a generator $G_{S \rightarrow R}$ to transform a source x_s into an image $\hat{x}_s = G_{S \rightarrow R}(x_s)$ that is indistinguishable from the target images X_r . The generator is followed by a discriminator D_R to differentiate the generated image $\hat{x}_s$ from target images x_r . Following the typical Generative adversarial network (GAN), the GAN loss could be written as Eq. 5.10

$$\mathcal{L}_{GAN_R}(G_{S \rightarrow R}, D_R) = \mathbb{E}_{x_r \sim X_r}[\log(D_R(x_r))] + \mathbb{E}_{x_s \sim X_s}[\log(1 - D_R(\hat{x}_s))] \quad (5.10)$$

As we use $G_{S \rightarrow R}$ as a wide-spectrum image generator, we also pass real target images x_r through it for image reconstruction. We use a reconstruction loss on the generator output $\hat{x}_r = G_{S \rightarrow R}(x_r)$, which could be written as Eq. 5.11.

$$\mathcal{L}_r(G_{S \rightarrow R}) = \mathbb{E}_{x_r \sim X_r} \|\hat{x}_r - x_r\|_1 \quad (5.11)$$

5.3.2 Segmentation Stage

In this stage we aim to train the task model, which is a semantic segmentation network. We train the segmentation network using that translated image-label pairs $(\hat{x}_s, y_s)$. We input $\hat{x}_s$ into the segmentation network to get the segmentation

softmax map p_s , and calculate the conventional per-pixel cross entropy loss using the segmentation label map y_s as Eq. 5.12

$$\mathcal{L}_{seg}(f_T) = -\mathbb{E}_{p_s \sim P_s} \sum_{h,w} \sum_{c \in C} y_s^{(h,w,c)} \log(p_s^{(h,w,c)}) \quad (5.12)$$

To further regularize the domain adaption process, we perform output space adaption by introducing another GAN loss function based on the prediction softmax map of both domain images through adversarial learning. Another discriminator D_P , which learns to differentiate source image segmentation map p_s from target image segmentation map p_r , is used for this purpose. As is stated in [51], the segmentation maps of source and target domain images contain similar context and layout information which is more suitable for domain adaptation than other high-level feature. The GAN loss function is formulated as Eq. 5.13

$$\mathcal{L}_{GAN_P}(f_T, D_P) = \mathbb{E}_{p_r \sim P_r} [\log(D_P(p_r))] + \mathbb{E}_{p_s \sim P_s} [\log(1 - D_P(p_s))] \quad (5.13)$$

where P_s and P_r denote the set of segmentation/prediction map of source and target domain.

5.3.3 Optimization

We follow an end-to-end training pipeline. Our full optimization objective is formulated as Eq. 5.14:

$$\begin{aligned} \mathcal{L}(G_{S \rightarrow R}, D_R, f_T, D_P) = & \mathcal{L}_{GAN_R}(G_{S \rightarrow R}, D_R) + \\ & \alpha_r \mathcal{L}_r(G_{S \rightarrow R}) + \alpha_t \mathcal{L}_{seg}(f_T) + \alpha_p \mathcal{L}_{GAN_P}(f_T, D_P) \end{aligned} \quad (5.14)$$

The optimal task network parameter f_T is calculated by:

$$f_T^* = \arg \min_{f_T} \min_{G_{S \rightarrow R}} \max_{D_R, D_P} \mathcal{L}(G_{S \rightarrow R}, D_R, f_T, D_P) \quad (5.15)$$

5.4 Experiments

5.4.1 Experimental Set-up

In our experiment, we take fully-supervised synthetic dataset of GTA5 [6] as the source domain data and unlabeled real-world dataset of Cityscapes [7] as the target domain data. Results are evaluated on the target domain validation data. The GTA5 [6] dataset contains 24966 frames synthesized from a video game, each frame is of high-resolution of 1914×1052 . Cityscapes [7] contains 2975 training images and 500 validation images, which are of high-resolution of 2048×1024 . The groundtruth of the two datasets are compatible with each other which contain 19 categories. We use the whole dataset of GTA5 to adapt to the training images of Cityscapes for training, and evaluate on the validation images of Cityscapes. We evaluate the segmentation results using the standard mean intersection-over-union (mIoU) metrics.

Following [51], our Segmentation model (*i.e.*, mask model and segmentation model in *Global Structure Embedding*, task network in *Task-in-all Pipeline*) is based on Deeplab-v2 [5] of Resnet101 [8] architecture with ASPP layers. We also utilize the same Discriminator model as in [51] with 5 convolution layers.

For the image-translation models in *Task-in-all Pipeline*, we follow the model of [49] and [82]. However, we utilize fewer parameters than [49] and [82] for the purpose for more efficient training. Specifically, we decrease the channel number of the intermediate layers from 64 to 32, and reduce the number of layers from 9 to 6. We use the same image-level discriminator as [82].

5.4.2 Global Structure Embedding

5.4.2.1 Ablation Study

We perform ablation studies to evaluate the contributions of each components of *Global Structure Embedding*. The ablation experiments are listed in Table. 5.1. The first block lists the baseline methods [51] without image-translation (pixel-DA) step. We refer to [51] as Local-Distri since it only utilizes local distribution

TABLE 5.1: Ablation Studies of *Global Structure Embedding* on Adapting GTA5 to Cityscapes. We show that our structure embedding strategy (STRUCT) can help improve performance. We also show the contribution of pixel-level/feature-level domain adaptation and self-training strategy.

Methods	mIoU
Baseline (without adaptation)	35.0
Local-Distri (feat-DA) [51]	41.5
CycleGAN [49] (pixel-DA) + Local-Distri (feat-DA)	42.8
CycleGAN [49] + Local-Distri + STRUCT	44.8
CycleGAN [49] + Local-Distri + STRUCT +SelfTrain	45.4
UNIT [48] + Local-Distri	44.7
UNIT [48] + Local-Distri + STRUCT	45.7
UNIT [48] + Local-Distri + STRUCT +SelfTrain	47.4

loss. The second and third blocks list the ablations studies using image translation methods (CycleGAN [49] or UNIT [48]). We came to the following conclusions:

First, we evaluate the contribution of image-translation methods. We evaluate two widely-applied image translation methods, namely CycleGAN [49] and UNIT [48]. We observe that performing image translation on the source images before feature-level (output) adaptation outperforms using feature-level adaptation alone (42.8 vs 41.5, 44.7 vs 41.5).

Second, we evaluate the contribution of Structure Embedding strategy (STRUCT). We observe that adding structure embedding loss helps improve the segmentation results (45.7 vs 44.7, 44.8 vs 42.8), since it compensates the structure misalignment problem in the original adversarial learning pipeline.

Third, we evaluate the contribution of self-training. The results show that adding self-training could further improve the segmentation results (47.4 vs 45.7, 45.4 vs 44.8), which achieves the best performance.

We list our segmentation performance for each semantic label in the lower block of Table 5.2. Compared with the baseline of AdaptSeg [51], we come to the following conclusions: 1) Adding the Image-translation stage (UNIT [48]) helps to improve the IoU score in most semantic classes over AdaptSeg. 2) Adding structure embedding loss helps to improve the IoU score in the classes with strong spatial prior and large number of training samples, such as road, terrain, car, etc. The reason is that the MMD-based structure loss encodes the class probability for each spatial

TABLE 5.2: Comparison *Global Structure Embedding* with other state-of-the-art methods on adapting GTA5 [6] to Cityscapes [7] datasets. We outperform other state-of-the-art methods with the Resnet101-based [8] segmentation networks. IT: Image-translation, Stru: Structure embedding loss, ST: Self-training.

Method	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU
AdaptSeg [51]	86.5	25.9	79.8	22.1	20.0	23.6	33.1	21.8	81.8	25.9	75.9	57.3	26.2	76.3	29.8	32.1	7.2	29.5	32.5	41.4
Advent [52]	89.4	33.1	81.0	26.6	26.8	27.2	33.5	24.7	83.9	36.7	78.8	58.7	30.5	84.8	38.5	44.5	1.7	31.6	32.4	45.5
ROAD [83]	76.3	36.1	69.6	28.6	22.4	28.6	29.3	14.8	82.3	35.3	72.9	54.4	17.8	78.9	27.7	30.3	4.0	24.9	12.6	39.4
CyCADA [56]	79.1	33.1	77.9	23.4	17.3	32.1	33.3	31.8	81.5	26.7	69.0	62.8	14.7	74.5	20.9	25.6	6.9	18.8	20.4	39.5
Yu et al. 2019 [84]	89.2	38.4	80.4	24.4	21.0	27.7	32.9	16.1	83.1	34.1	77.8	57.4	27.6	78.6	31.2	40.2	4.7	27.6	27.6	43.2
Ours (+IT)	88.3	42.5	83.7	32.8	23.8	29.0	31.9	27.5	84.0	38.2	80.7	56.8	26.1	73.4	28.3	37.7	4.9	28.3	31.6	44.7
Ours (+IT/Stru)	89.0	41.4	83.4	33.9	22.8	28.7	31.5	25.2	83.7	39.5	80.3	57.5	27.6	77.7	33.5	40.9	4.9	28.6	39.1	45.7
Ours (+IT/Stru/ST)	89.8	41.7	84.1	33.2	23.4	28.7	31.2	29.1	84.6	43.4	81.4	59.6	30.0	79.1	35.3	51.8	0.23	32.2	41.7	47.4

location. 3) Whether adding self-training stage can help improve the IoU score is dependent on the segmentation performance in the last previous stage. If the IoU score is high in the last iteration (+IT/STRU), self-training step will help boost the performance, such as road, veg, sky, car, etc. On the contrary, if the IoU score is low in the last iteration (+IT/STRU), self-training step will further decrease the performance, such as the class of ‘train’.

5.4.2.2 Comparison with the state-of-the-art

Finally, we compare our segmentation results with other state-of-the-art methods. For fair comparison, we only list the results which utilize segmentation models based on Resnet101 [8] model. The comparison results are listed in Table 5.2. The results show that our *mIoU* performance outperform other state-of-the-art approaches. The upperline *mIoU* result of training with target domain pixel-wise annotation is 71.0. Our segmentation results are competitive with the upperline result considering that no label information is available in the training stage for the target domain data. We also provide visualization of the segmentation results in Fig. 5.3. It shows that the segmentation results in bulky classes such as building and sky are relatively smooth and accurate. However, for the classes of small scale such as pole, light, many details are likely to be neglected in the segmentation masks.

FIGURE 5.3: The visualization of segmentation performance of *Global Structure Embedding*.

5.4.3 Task-in-all Pipeline

5.4.3.1 Visualization Results

We list some examples of image-translation results of *Task-in-all Pipeline* in Fig. 5.4. We come to the following observations.

First, we observe that with the help of our image-translation model (second column) or CycleGAN [49] (third column), we could translate the source images to the color style which is similar to the target domain images.

Second, CycleGAN [49] achieves more visually satisfactory results compared with our image translation model in terms of surface texture. Taking the road surface for example, the results of CycleGAN is of more smooth texture, which is more similar to the target domain images. On the contrary, the road surface of our image translation results are more similar to the source domain which are of more spotted texture. However, it is still a open question whether the more visually satisfactory results will necessarily lead to the better final task performance. We will further investigate this question in Sec. 5.4.3.2.

It is worth noting that there exists the case of pixel-wise content-misalignment problems in the translated images generated by CycleGAN. For example, in the last row of Fig. 5.4, the sky (light grey) shows the texture/color which belongs to the ground label (dark grey) in the translated images generated by CycleGAN. As we stated in the Sec. 5.1, such pixel-level content misalignment problem may be propagated to the subsequent task model. Finally it leads to the output-level content misalignment problem, which means the low segmentation accuracy. The pixel-level misalignment problem is the main target problem we aim to tackle in this *Task-in-all Pipeline* approach. Visualization results of our image-translation stage in Fig. 5.4 shows that we alleviate the mis-alignment problems in the translated stages.

5.4.3.2 Segmentation Results

We list our segmentation results of *Task-in-all Pipeline* in Table 5.3. For fair comparison, we compare with the experiments on CycleGAN [49], which shares the similar model image generator structure but of different training pipeline. We directly utilize the off-the-shelf result of GTA5-to-Cityscapes image-translated results provided by [56] as the source data, and apply high-level domain adaptation [51] on it. Such experiments could be utilized to simulate the case of sequentially training the image-translation network and task segmentation network.

We make the following conclusions from Table 5.3.

First, we observe that performing image translation on the source images before high-level (output) adaptation outperforms using high-level adaptation alone, no matter using separate training or end-to-end Task-in-all training pipeline.

Second, our Task-in-all pipeline outperforms separate-training pipeline, with less-complicated image-translation model. Thus, with such end-2-end training, our pipeline could generate the transferred images which better facilitate the tasks of semantic segmentation. It is worth mentioning that we utilize the image-translation models of the similar structure but with fewer channel numbers and layers, without the complicated cycle training pipeline.

Based on the observations in Sec. 5.4.3.1, although we did not achieve visually better translated image over CycleGAN, we still outperform it as for the final

FIGURE 5.4: The visualization of the translated images of *Task-in-all Pipeline*. The first column demonstrates the original source image, and the second column demonstrates the corresponding translated images of our pipeline. The third column demonstrates the translated images of CycleGAN. The last column shows the examples of the target images. We observe that there exist pixel-level content-misalignment problems in the translated images generated by CycleGAN. Our image translation model alleviate such pixel-level content-misalignment problems.

segmentation results. We conclude that the more visually satisfactory results may not necessarily be more suitable for the final tasks.

TABLE 5.3: Ablation Studies of *Task-in-all Pipeline* on the task of UDA semantic segmentation between GTA5 [6] and Cityscapes [7] dataset. We show that our Task-in-all pipeline outperforms sequential training pipeline in semantic segmentation in the target domain images.

Methods	mIoU
Baseline (only train on the source images)	35.0
Adapt on output [51]	41.5
CycleGAN [49] + Adapt on output (separate training)	42.8
Task-in-all (end2end) (ours)	43.4

5.5 Comparison and Analysis

In this section, we compare and analyze the pros and cons of the two proposed regularization approaches. The two approaches are similar in the sense that they perform both pixel-level and feature-level adversarial learning approach and they utilize the similar model structures. We compare the two regularization approaches from different perspectives such as training cost, target model to regularize and the final segmentation performance. The comparisons are listed in Table 5.4. Our analysis of the comparison results is as follows:

First, as for the training cost, the main limitation of *Global Structure Embedding* is that the computational cost is high since the MMD computational cost is proportional to the input vector dimension and the dimension of the flattened vector is quite high. One solution to alleviate this problem is to spatially downsample the input maps before the vectorization operation, since the global structure knowledge is mainly a spatial distribution which does not need to be represented by the high resolution. The main limitation of *Task-in-all Pipeline* is that the training process consumes large memory since it needs to load and optimizes the network parameters in an end-to-end pipeline. One solution is to downsample the spatial size of the output of the image translation model. Another solution is to reduce the intermedia channels/layers of image translation. However, the total memory is still over 12GB, which means it requires large GPU memory capacity.

TABLE 5.4: The comparison between *Global Structure Embedding* and *Task-in-all Pipeline* approaches. We compare the two regularization approaches from different perspectives such as training cost, target model to regularize and the final segmentation performance.

method	<i>Global Structure Embedding</i>		<i>Task-in-all Pipeline</i>	
Regularization perspective	Add regularization loss for the feature-level DA		perform regularization pipeline to regularize the pixel-level DA	
Computation Cost	The computation cost of MMD is $\mathcal{O}(n^2d)$, where n denotes the batch size and d denotes the dimension of the input vector		No additional training cost since there is no additional loss/parameters to optimize.	
Memory Consumption	The additional memory cost is neglect-able since no additional parameters need to be optimized		The memory cost is large since we need to optimize all the parameters in an end-to-end pipeline, which means we need to load all the model parameters at one time.	
Regularization Performance	CycleGAN+AdaptSeg	42.8	CycleGAN+AdaptSeg	42.8
	CycleGAN+AdaptSeg+STRUCT	44.8	CycleGAN+AdaptSeg (Task in all)	43.4
Final Performance	$mIoU$ of 47.4		$mIoU$ of 43.4	

Second, the final segmentation results of *Global Structure Embedding* outperforms the result of *Task-in-all Pipeline*. The reasons are as follows: 1) The additional self-training step improves the results of solely applying adversarial learning steps. 2) The image translation model of UNIT [48] is more advanced over CycleGAN [49]. 3) We observe that on the basis of same image translation method (*i.e.*, CycleGAN), the segmentation results of *Global Structure Embedding* still outperforms *Task-in-all Pipeline* (*i.e.*, 44.8 vs. 43.4), which shows the advantage of the regularization method of *Global Structure Embedding*.

Third, *Global Structure Embedding* and *Task-in-all Pipeline* are the regularization approaches which focus on different perspectives. *Global Structure Embedding* focuses on adding regularization loss while *Task-in-all Pipeline* focuses on a optimizing pipeline. *Global Structure Embedding* focuses on regularizing the feature-level DA while *Task-in-all Pipeline* focuses on regularizing the pixel-level DA. Thus, such two approaches could be complementary to each other. Combining the two

approaches may further improve the segmentation performance. We will investigate it in our future work.

5.6 Summary

We propose two regularization approaches for UDA in semantic segmentation in the Chapter: *Global Structure Embedding* and *Task-in-all Pipeline*. *Global Structure Embedding* focuses on adding a global structure embedding loss to regularize the feature-level DA stage. *Task-in-all Pipeline* focuses on switching the separate training pipeline into an end-to-end training process, which regularizes the pixel-level DA (*i.e.*, image-translation) stage. We perform ablation studies on each regularization approach to show their effect of alleviating content misalignment problems. Finally we achieve the state-of-the-art segmentation results in adapting GTA5 synthetic datasets to Cityscapes real-world datasets in semantic segmentation. We compare and analyze the pros and cons of the two regularization approaches and the combination of the two approaches may potentially further improve the segmentation results.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

The goal of this thesis is to reduce the data annotation work load for semantic image segmentation. We tackle this problem from two perspectives: intra-domain view and inter-domain view. The intra-domain view aims to utilize weaker-level supervision format over the pixel-wise supervision in the real-world dataset, namely weakly-supervised semantic segmentation (WSSS). The inter-domain view aims to train with the pixel-wise annotations from a synthetic data domain and unlabeled target domain data, namely unsupervised domain adaptation (UDA) for semantic segmentation.

As for WSSS tasks, we follow the pipeline of sequentially estimating the pseudo-annotations and training segmentation model with the estimated pseudo-annotations. For the purpose of accurately estimating the pseudo-annotations, we propose two region-mining approaches: The first work *Decoupled Spatial Neural Attention for Weakly Supervised Semantic Segmentation* proposes a decoupled attention structure which could simultaneously mine out the integral object regions and discriminative object parts. The second work *Splitting vs. Merging: Mining Object Regions with Discrepancy and Intersection Loss for Weakly Supervised Semantic Segmentation* proposes to combine the Discrepancy loss and Intersection loss to expand the highlighted region from the discriminative parts to other informative object regions.

As for UDA in semantic segmentation tasks, we follow the pipeline of applying pixel-level and feature-level adversarial learning methods. In our third work *Semantic Content Regularization for UDA in Semantic Segmentation*, we propose and compare two approaches, namely *Global Structure Embedding* and *Task-in-all Pipeline*, to regularize the content-misalignment problems in adversarial learning. *Global Structure Embedding* proposes a structure embedding loss to regularize the feature-level adversarial learning step. *Task-in-all Pipeline* proposes to back-propagate the final task loss to regularize the pixel-level adversarial learning step.

In summary, both the WSSS and UDA approaches could reduce work load of pixel-wise annotating the target real-world data. WSSS utilizes the less labor-intensive label format while UDA utilize the data source whose labels are easier to generate. WSSS focuses on recovering the pixel-wise information from the weak annotation format while UDA focuses on narrowing the domain distribution gap between the different data sources. Both of the two approaches achieve satisfactory semantic segmentation results on the basis that no pixel-wise annotation is required on the target datasets. With the help of WSSS and UDA methods, DCNNs for semantic segmentation could be easily extended for new emerging classes and datasets.

6.2 Future Works

Although our approaches achieve satisfactory semantic segmentation results without pixel-wise annotations for the training images, the performance still obviously under-perform the fully-supervised approaches. There still exist large amounts of problems to explore. In this section, we point out some directions to investigate.

Context Relationship for Mining Regions. This direction focuses on the pseudo-annotation estimation steps of WSSS tasks. In order to rely on the low-level context relationship cues to compensate the high-level feature discrimination, most existing works applies the post-processing refinement step over the result heatmap generated by the region mining models. However, if the result heatmap only highlights parts of the objects, the post-processing refinement can not effectively expand the highlighted region to the integral objects. Thus, we intend to enclose the low-level context relationship in the training process of the region mining models, which adaptively expand the highlighted regions in the training

process. In order to enclose the low-level content feature, denseCRF [9] utilizes fully connected CRF which is quite time-consuming for accurate refinement. Such refinement mechanism is not applicable to be embedded in the training process since it is quite difficult and inefficient to optimize. Thus, one of the key obstacles to tackle is to increase the efficiency of context relationship estimation to make it applicable in an online manner.

Expand vs. Shrink for Mining Regions. Currently most region mining approaches focus on the problem of expanding the highlighted region generated by the region mining model to the integral object scales. There is seldom works that work in the other direction of how to shrink from the entire image region to the target object scale. We came up with direction based on the following observations: First, with the increase of training epochs of region mining models, the highlighted feature range automatically shrinks from the entire image to the most discriminative parts. We could rely on such training process to decide when to stop the training iterations to stop the shrinking process. Second, currently the highlighted regions are mainly calculated on the higher-level features. Such features have strong discriminative property which are not necessarily our desired feature. On the contrary, the low-level features are likely to highlight larger range of regions. The low-level features only suppress the features where there exist no complicated texture or boundaries, such as water or sky. We empirically observe that the highlighted region of the high-level feature is the subset of the highlighted region of the low-level feature space, which forms a shrinking phenomena through the structure of the deep models. We could also utilize the fact to determine which level to choose to stop the shrinking process.

Segmentation Structure for WSSS. Currently most of the WSSS approaches directly apply the segmentation model of the fully-supervised semantic segmentation tasks on the estimated pseudo-annotations. There exist some potential problems. First, fully-supervised semantic segmentation model often focuses on mining out the high-resolution details (*eg.*, object boundaries) of the prediction masks, while such high-resolution groundtruth details are not available for the weakly-supervised settings. Second, fully-supervised semantic segmentation model tend to completely fit the ground-truth annotations since the annotations are completely reliable. However, the estimated pseudo-annotations are noisy and training the traditional fully-supervised model can easily overfit to the noisy labels. Thus, it is

a promising direction to investigate the segmentation model for weakly-supervised setting. Inspired by [1, 26], one promising direction is to encode global information to help filter out the noisy local prediction. It is difficult to segment out the target objects of tiny sizes just based on the segmentation model. However, such small sizes of the target objects usually lead to the low classification score calculated by average pooling the pixel-wise prediction score. Thus, the global information could remove the noisy predictions of the small size objects and enhance the segmentation of large-scale objects. The other directions include proposing a segmentation model that can adaptively set the training weights for the certain and uncertain pixel-wise labels. Such direction tend to be effective for weakly supervised semantic segmentation since the estimated pseudo-labels are noisy and overfitting the segmentation model to all the pseudo-labels will dramatically decrease the final segmentation performance.

Web Sources for WSSS. Although it is a common practice to utilize web sources to assist the WSSS tasks, there still exist promising directions to explore. Besides the images/videos, the web sources contain lots of additional information which could be utilized. For example, instead of just the image-level label, we could also utilize more informative descriptions for the image which could help to better localize the target objects. For example, with the text descriptions, we may have more information about the target objects, such as color, position, scales, etc. With the help of image captioning methods, we would mine-out the attention maps which corresponds to the target objects described by the caption, which better facilitate the task of initial pseudo-annotations for WSSS tasks. Another possible direction to investigate is to extend the attention structure to video motion cues. Currently, the WSSS work which utilizes the web crawled videos is still based on the localization seeds generated by the image region mining models. It post-processes the localization seeds with the motion cues for the purpose of pseudo-annotation generations. Different from the previous works we focus on incorporating the motion cues directly in the structure of region mining models to output the localization maps which enclose the integral object regions.

Feature Representation for UDA in semantic segmentation. [51] proves that performing adversarial learning on the output space is more effective on the high-dimensional feature space. [52] shows that aligning the entropy feature space

shows promising performance for UDA semantic segmentation tasks. Thus, it is interesting to investigate how to design the feature space that the UDA approaches operate on for the semantic segmentation tasks. In general, such feature space should be of low-dimension property while mining out the key structure information for segmentation tasks. How to project the original high-dimensional feature space to low-dimension feature and which feature is suitable for UDA semantic segmentation is still an open problem to investigate.

List of Author's Publications

Journal Articles

- **Tianyi Zhang**, Guosheng Lin, Jianfei Cai, Tong Shen, Chunhua Shen and Alex C. Kot “Decoupled Spatial Neural Attention for Weakly Supervised Semantic Segmentation,” *IEEE Transactions on Multimedia*, 2019.

Conference Proceedings

- **Tianyi Zhang**, Yu Zhang, Jianfei Cai and Alex C. Kot, “Efficient object feature selection for action recognition,” in Proceedings of *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.
- **Tianyi Zhang**, Li Niu, Jianfei Cai and Alex C. Kot, “Action proposals using hierarchical clustering of super-trajectories,” in Proceedings of *IEEE Visual Communications and Image Processing (VCIP)*, 2017.
- **Tianyi Zhang**, Guosheng Lin, Jianfei Cai and Alex C. Kot, “Semantic Segmentation via Domain Adaptation with Global Structure Embedding,” in Proceedings of *IEEE Visual Communications and Image Processing (VCIP)*, 2019.
- **Tianyi Zhang**, Jingyi Yang, Chuanxia Zheng, Guosheng Lin, Jianfei Cai and Alex C. Kot, “Task-in-all Domain Adaptation for Semantic Segmentation,” in Proceedings of *IEEE Visual Communications and Image Processing (VCIP)*, 2019.

- **Tianyi Zhang**, Guosheng Lin, Weide Liu and Jianfei Cai, “Splitting vs. Merging: Mining Object Regions with Discrepancy and Intersection Loss for Weakly Supervised Semantic Segmentation,” submitted to *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Bibliography

- [1] T. Shen, G. Lin, C. Shen, and I. Reid, “Learning multi-level region consistency with dense multi-label networks for semantic segmentation,” in *IJCAI*, 2017. [xx](#), [xxi](#), [xxiii](#), [xxiv](#), [41](#), [44](#), [61](#), [65](#), [90](#)
- [2] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *CVPR*, 2016. [xx](#), [xxiii](#), [xxvii](#), [11](#), [12](#), [22](#), [27](#), [30](#), [39](#), [40](#), [41](#), [49](#), [50](#)
- [3] A. Kolesnikov and C. H. Lampert, “Seed, expand and constrain: Three principles for weakly-supervised image segmentation,” in *ECCV*, 2016. [xxiii](#), [xxviii](#), [12](#), [13](#), [22](#), [30](#), [39](#), [40](#), [41](#), [42](#), [43](#), [59](#), [60](#), [61](#)
- [4] Y. Wei, J. Feng, X. Liang, M. Cheng, Y. Zhao, and S. Yan, “Object region mining with adversarial erasing: A simple classification to semantic segmentation approach,” in *CVPR*, 2017. [xxiii](#), [10](#), [11](#), [12](#), [13](#), [22](#), [30](#), [39](#), [40](#), [42](#), [43](#), [59](#), [60](#), [61](#)
- [5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected crfs,” in *ICLR*, 2015. [xxiii](#), [xxiv](#), [7](#), [8](#), [22](#), [32](#), [41](#), [61](#), [78](#)
- [6] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, “Playing for data: Ground truth from computer games,” in *ECCV*, 2016. [xxv](#), [69](#), [78](#), [80](#), [84](#)
- [7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *CVPR*, 2016. [xxv](#), [69](#), [78](#), [80](#), [84](#)
- [8] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016. [xxv](#), [45](#), [78](#), [80](#)
- [9] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected crfs with gaussian edge potentials,” in *NIPS*, 2011. [xxvii](#), [12](#), [30](#), [31](#), [32](#), [39](#), [40](#), [53](#), [54](#), [89](#)
- [10] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, 2015. [7](#), [22](#)
- [11] H. Noh, S. Hong, and B. Han, “Learning deconvolution network for semantic segmentation,” in *ICCV*, 2015. [7](#)

- [12] G. Lin, A. Milan, C. Shen, and I. Reid, “RefineNet: Multi-path refinement networks for high-resolution semantic segmentation,” in *CVPR*, 2016. [8](#), [22](#), [32](#)
- [13] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *CVPR*, 2017.
- [14] L. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, “Attention to scale: Scale-aware semantic image segmentation,” in *CVPR*, 2016.
- [15] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, “Conditional random fields as recurrent neural networks,” in *ICCV*, 2015.
- [16] X. Li, Z. Liu, P. Luo, C. C. Loy, and X. Tang, “Not all pixels are equal: difficulty-aware semantic segmentation via deep layer cascade,” in *CVPR*, 2017.
- [17] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid, “Efficient piecewise training of deep structured models for semantic segmentation,” in *CVPR*, 2016. [7](#)
- [18] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, “Scribblesup: Scribble-supervised convolutional networks for semantic segmentation,” in *CVPR*, 2016. [8](#), [41](#), [42](#)
- [19] J. Dai, K. He, and J. Sun, “Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation,” in *ICCV*, 2015. [8](#), [41](#), [42](#), [43](#)
- [20] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, “What’s the point: Semantic segmentation with point supervision,” in *ECCV*, 2016. [8](#), [41](#), [42](#)
- [21] D. Pathak, E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional multi-class multiple instance learning,” in *ICLR Workshop*, 2015. [8](#), [42](#), [43](#)
- [22] P. O. Pinheiro and R. Collobert, “From image-level to pixel-level labeling with convolutional networks,” in *CVPR*, 2015. [8](#)
- [23] G. Papandreou, L. Chen, K. Murphy, and A. L. Yuille, “Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation,” in *ICCV*, 2015. [8](#), [13](#), [42](#), [43](#), [59](#), [60](#)
- [24] 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,” *TPAMI*, vol. 39, no. 11, pp. 2314–2320, 2017. [9](#), [41](#), [42](#), [43](#), [59](#), [60](#), [61](#)
- [25] T. Shen, G. Lin, L. Liu, C. Shen, and I. Reid, “Weakly supervised semantic segmentation based on co-segmentation,” 2017. [9](#), [42](#), [43](#), [59](#), [60](#)
- [26] T. Shen, G. Lin, C. Shen, and I. Reid, “Bootstrapping the performance of weakly supervised semantic segmentation,” in *CVPR*, 2018. [9](#), [54](#), [59](#), [60](#), [61](#), [90](#)

- [27] S. Hong, D. Yeo, S. Kwak, H. Lee, and B. Han, “Weakly supervised semantic segmentation using web-crawled videos,” 2017. [9](#), [41](#), [42](#), [43](#), [59](#), [60](#), [61](#)
- [28] D. Kim, D. Yoo, I. S. Kweon, *et al.*, “Two-phase learning for weakly supervised object localization,” in *ICCV*, 2017. [10](#), [11](#), [12](#), [42](#), [43](#), [59](#), [60](#)
- [29] Q. Hou, P. Jiang, Y. Wei, and M. Cheng, “Self-erasing network for integral object attention,” in *NeurIPS*, 2018. [11](#), [30](#), [59](#), [60](#)
- [30] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T. S. Huang, “Adversarial complementary learning for weakly supervised object localization,” in *CVPR*, 2018. [11](#), [50](#)
- [31] K. Li, Z. Wu, K. Peng, J. Ernst, and Y. Fu, “Tell me where to look: Guided attention inference network,” in *CVPR*, 2018. [11](#), [59](#), [60](#)
- [32] T. Zhang, G. Lin, J. Cai, T. Shen, C. Shen, and A. C. Kot, “Decoupled spatial neural attention for weakly supervised semantic segmentation,” *TMM*, 2019. [11](#), [12](#), [55](#), [59](#), [60](#), [61](#)
- [33] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, “Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation,” in *CVPR*, 2018. [11](#), [13](#), [30](#), [59](#), [60](#)
- [34] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, “Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization,” in *ICCV*, 2017. [11](#), [12](#), [22](#)
- [35] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, “Top-down neural attention by excitation backprop,” in *ECCV*, 2016. [11](#), [22](#)
- [36] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, “Image captioning with semantic attention,” in *CVPR*, 2016. [12](#)
- [37] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *ICML*, 2015. [12](#), [23](#), [25](#)
- [38] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *ICLR*, 2015. [12](#)
- [39] F. Zhu, H. Li, W. Ouyang, N. Yu, and X. Wang, “Learning spatial regularization with image-level supervisions for multi-label image classification,” in *CVPR*, 2017. [12](#), [26](#)
- [40] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, “Multi-context attention for human pose estimation,” in *CVPR*, 2017. [12](#), [25](#)
- [41] J. Kuen, Z. Wang, and G. Wang, “Recurrent attentional networks for saliency detection,” in *CVPR*, 2016. [12](#)

- [42] J. Ahn and S. Kwak, “Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation,” in *CVPR*, 2018. [12](#), [53](#), [59](#), [60](#), [61](#)
- [43] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, “Weakly-supervised semantic segmentation network with deep seeded region growing,” in *CVPR*, 2018. [13](#), [59](#), [60](#)
- [44] A. Gretton, K. Fukumizu, Z. Harchaoui, and B. K. Sriperumbudur, “A fast, consistent kernel two-sample test,” in *NIPS*, 2009. [14](#), [73](#)
- [45] H. Li, S. J. Pan, R. Wan, and A. C. Kot, “Heterogeneous transfer learning via deep matrix completion with adversarial kernel embedding,” in *AAAI*, 2019.
- [46] H. Li, S. J. Pan, S. Wang, and A. C. Kot, “Domain generalization with adversarial feature learning,” in *CVPR*, 2018. [14](#), [73](#)
- [47] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *NIPS*, 2014. [15](#)
- [48] M. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” in *NIPS*, 2017. [15](#), [71](#), [79](#), [85](#)
- [49] J. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *ICCV*, 2017. [15](#), [18](#), [71](#), [78](#), [79](#), [81](#), [82](#), [83](#), [84](#), [85](#)
- [50] J. Hoffman, D. Wang, F. Yu, and T. Darrell, “Fcns in the wild: Pixel-level adversarial and constraint-based adaptation,” *arXiv preprint arXiv:1612.02649*, 2016. [15](#), [18](#), [69](#)
- [51] Y.-H. Tsai, W.-C. Hung, S. Schuler, K. Sohn, M.-H. Yang, and M. Chandraker, “Learning to adapt structured output space for semantic segmentation,” in *CVPR*, 2018. [18](#), [72](#), [73](#), [77](#), [78](#), [79](#), [80](#), [81](#), [82](#), [84](#), [90](#)
- [52] T. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, “Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation,” *arXiv preprint arXiv:1811.12833*, 2018. [15](#), [69](#), [80](#), [90](#)
- [53] G. Wilson and D. J. Cook, “A survey of unsupervised deep domain adaptation,” *arXiv preprint arXiv:1812.02849*, 2019. [16](#), [18](#)
- [54] M. Wang and W. Deng, “Deep visual domain adaptation: A survey,” *Neuro-computing*, vol. 312, pp. 135–153, 2018. [16](#)
- [55] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *ICML*, 2015. [16](#)
- [56] J. Hoffman, E. Tzeng, T. Park, J. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, “Cycada: Cycle consistent adversarial domain adaptation,” in *ICML*, 2018. [18](#), [69](#), [80](#), [82](#)

- [57] Y. Zou, Z. Yu, B. Vijaya Kumar, and J. Wang, “Unsupervised domain adaptation for semantic segmentation via class-balanced self-training,” in *ECCV*, 2018. [19](#)
- [58] F. Xia, P. Wang, X. Chen, and A. Yuille, “Joint multi-person pose estimation and semantic part segmentation,” in *CVPR*, 2017. [22](#)
- [59] S. Liu, X. Liang, L. Liu, K. Lu, L. Lin, X. Cao, and S. Yan, “Fashion parsing with video context,” *TMM*, vol. 17, no. 8, pp. 1347–1358, 2015. [22](#)
- [60] X. Liang, L. Lin, W. Yang, P. Luo, J. Huang, and S. Yan, “Clothes co-parsing via joint image segmentation and labeling with application to clothing retrieval,” *TMM*, vol. 18, no. 6, pp. 1175–1186, 2016.
- [61] S. Liu, J. Feng, C. Domokos, H. Xu, J. Huang, Z. Hu, and S. Yan, “Fashion parsing with weak color-category labels,” *TMM*, vol. 16, no. 1, pp. 253–265, 2014. [22](#)
- [62] H. Shi, H. Li, F. Meng, Q. Wu, L. Xu, and K. N. Ngan, “Hierarchical parsing net: Semantic scene parsing from global scene to objects,” *TMM*, vol. PP, no. 99, pp. 1–1, 2018. [22](#)
- [63] Z. Qiu, T. Yao, and T. Mei, “Learning deep spatio-temporal dependency for semantic video segmentation,” *TMM*, vol. PP, no. 99, pp. 1–1, 2017. [22](#)
- [64] S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele, “Exploiting saliency for object segmentation from image level labels,” in *CVPR*, 2017. [22](#), [42](#), [43](#), [59](#), [60](#)
- [65] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, “Stacked attention networks for image question answering,” in *CVPR*, 2016. [23](#)
- [66] B. Zhuang, L. Liu, Y. Li, C. Shen, and I. Reid, “Attend in groups: a weakly-supervised deep learning framework for learning from web data,” in *CVPR*, 2017. [27](#)
- [67] F. Saleh, M. S. A. Akbarian, M. Salzmann, L. Petersson, S. Gould, and J. M. Alvarez, “Built-in foreground/background prior for weakly-supervised semantic segmentation,” in *ECCV*, 2016. [30](#), [42](#), [43](#), [59](#), [60](#)
- [68] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *IJCV*, vol. 88, no. 2, 2010. [31](#), [54](#)
- [69] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, “Semantic contours from inverse detectors,” in *ICCV*, 2011. [31](#), [54](#)
- [70] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, “Large kernel matters—improve semantic segmentation by global convolutional network,” in *CVPR*, 2017. [32](#)

- [71] B. Jin, M. V. Ortiz Segovia, and S. Susstrunk, “Webly supervised semantic segmentation,” in *CVPR*, 2017. [42](#), [43](#), [59](#), [60](#), [61](#)
- [72] S. Hong, J. Oh, B. Han, and H. Lee, “Learning transferrable knowledge for semantic segmentation with deep convolutional neural network,” in *CVPR*, 2015. [41](#), [42](#), [43](#)
- [73] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick, “Microsoft coco: Common objects in context,” in *ECCV*, 2014. [41](#), [42](#), [43](#)
- [74] X. Qi, Z. Liu, J. Shi, H. Zhao, and J. Jia, “Augmented feedback in semantic segmentation under image level supervision,” in *ECCV*, 2016. [41](#), [42](#), [43](#), [59](#), [60](#)
- [75] A. Chaudhry, P. K. Dokania, and P. H. Torr, “Discovering class-specific pixels for weakly-supervised semantic segmentation,” in *BMVC*, 2017. [42](#), [43](#), [59](#), [60](#)
- [76] Q. Hou, P. K. Dokania, D. Massiceti, Y. Wei, M. Cheng, and P. Torr, “Mining pixels: Weakly supervised semantic segmentation using image labels,” *arXiv preprint arXiv:1612.02101*, 2016. [42](#), [43](#)
- [77] W. Shimoda and K. Yanai, “Distinct class-specific saliency maps for weakly supervised semantic segmentation,” in *ECCV*, 2016. [42](#), [43](#), [59](#), [60](#)
- [78] A. Roy and S. Todorovic, “Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation,” in *CVPR*, 2017. [42](#), [43](#), [59](#), [60](#)
- [79] J. Liu, Q. Hou, M. Cheng, J. Feng, and J. Jiang, “A simple pooling-based design for real-time salient object detection,” *arXiv preprint arXiv:1904.09569*, 2019. [54](#), [61](#)
- [80] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon, “Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference,” in *CVPR*, 2019. [59](#), [60](#)
- [81] A. Smola, A. Gretton, L. Song, and B. Schölkopf, “A hilbert space embedding for distributions,” in *ICALT*, 2007. [73](#)
- [82] C. Zheng, T. Cham, and J. Cai, “T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks,” in *ECCV*, 2018. [75](#), [78](#)
- [83] Y. Chen, W. Li, and L. Van Gool, “Road: Reality oriented adaptation for semantic segmentation of urban scenes,” in *CVPR*, 2018. [80](#)
- [84] Y. Tsai, K. Sohn, S. Schuler, and M. Chandraker, “Domain adaptation for structured output via discriminative representations,” *arXiv preprint arXiv:1901.05427*, 2019. [80](#)