Summer 2022

Cross Domain Semantic Segmentation

Xinyi Wu

Cross Domain Semantic Segmentation

by

Xinyi Wu

Bachelor of Engineering
Tianjin University, 2018

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Computer Science and Engineering

College of Engineering and Computing

University of South Carolina

2022

Accepted by:

Song Wang, Major Professor

Lili Ju, Major Professor

Michael N. Huhns, Committee Member

Yan Tong, Committee Member

Lannan Luo, Committee Member

Tracey L. Weldon, Vice Provost and Dean of the Graduate School

Acknowledgments

While I'm writing the dissertation, I realize that I'm in the last days of my doctoral pursuing at University of South Carolina. I feel loath to leave because of the unforgettable days I spent with the people here.

I would first like to thank my advisor, Prof. Song Wang, from whom I learned how to choose topics to work on and became self-motivated. His feedback usually pushed me to think more and improved my works to a higher level. And I also want to thank my co-advisor Prof. Lili Ju, who helped a lot on all of my paper submissions and supported me with the research assistant position. To Prof. Michael N. Huhns, Prof. Yan Tong, and Prof. Lanan Luo, I appreciate the time you spent on my dissertation and your helps on my job searching.

I would then thank my colleagues and collaborators including Dr. Dazhou Guo, Dr. Kang Zheng, Dr. Yang Mi, Dr. Hao Guo, Jun Zhou, Yuhang Lu, Lan Fu, Zhenyao Wu, Liang Zhao, Xiaoguang Li, Zhihao Liu, Jin Wan, Rabab Abdelfattah, Canyu Zhang, Ziyu Zhao, Pingping Cai, Mohammad Hassan Erfani and Yuankai Teng. I feel great to meet you and working with you during these years. I would also want to thank Prof. Hongkai Yu, Ying Meng and Xinyu Wang who also gave me useful suggestions on my job searching.

Last but not least, I want to thank my parents who supported me study aboard to see the world; my boyfriend, Zhenyao, who helped me go through a lot of difficult time in the four years; my friends (in alphabetical order), Danning Zhao, Feifei Niu, Shengnan Feng, Yi Mu, Yuehuan Wang and Yunxin Qiao who always share experiences with me; and finally to myself.

Abstract

As a long-standing computer vision task, semantic segmentation is still extensively researched till now because of its importance to visual understanding and analysis. The goal of semantic segmentation is to classify each pixel of images based on the pre-defined classes. In the era of deep learning, convolutional neural networks largely improve the accuracy and efficiency of semantic segmentation. However, this success is achieved with two limitations: 1) a large-scale labeled dataset is required for training while the labeling process for this task is quite labor-intensive and tedious; 2) the trained deep networks can get promising results when testing on the same domain (*i.e.*, intra-domain test) but might suffer from a large performance drop when testing on different domains (*i.e.*, cross-domain test). Therefore, developing algorithms that can transfer knowledge from labeled source domains to unlabeled target domains is highly desirable to address these two limitations.

In this research, we explore three settings of cross domain semantic segmentation conditioned on the use of different training data in the target domain: 1) the use of a sole unlabeled target image, 2) the use of multiple unlabeled target images, and 3) the use of unlabeled target videos, respectively.

At the first part, we tackle the problem of one-shot unsupervised domain adaptation (OSUDA) for semantic segmentation where the segmentors only use one unlabeled target image during training. In this case, traditional unsupervised domain adaptation models usually fail since they cannot adapt to the target domain with over-fitting to one (or few) unlabeled target samples. To address this problem, existing OSUDA methods usually integrate a style-transfer module to perform domain

randomization based on the unlabeled target sample, with which multiple domains around the target sample can be explored during training. However, such a style-transfer module relies on an additional set of images as style reference for pre-training and also increases the memory demand for domain adaptation. Here we propose a new OSUDA method that can effectively relieve such computational burden by making full use of the sole target image in two aspects: (1) implicitly stylizing the source domain in both image and feature levels; (2) softly selecting the source training pixels. Experimental results on two commonly-used synthetic-to-real scenarios demonstrate the effectiveness and efficiency of the proposed method.

Secondly, we work on the problem of nighttime semantic segmentation which plays an equally important role as that of daytime images in autonomous driving but is much more challenging and less studied due to poor illuminations and arduous human annotations. Our proposed solution employs an adversarial training with a labeled daytime dataset and an unlabeled dataset that contains coarsely aligned day-night image pairs. The unlabeled daytime images from the target dataset serve as an intermediate domain to mitigate the difficulty in day-to-night adaption since they share similarities with the source in illumination pattern and contain the same static-category objects as their nighttime counterparts. Extensive experiments on Dark Zurich and Nighttime Driving datasets show that our method achieves state-of-the-art performance for nighttime semantic segmentation.

Finally, we propose a domain adaptation method for video semantic segmentation, *i.e.*, the target is in video format. Before our work, other works were achieving this goal by transferring the knowledge from the source domain of self-labeled simulated videos to the target domain of unlabeled real-world videos. In our work, we argue that it is not necessary to use a labeled video dataset as the source since the temporal continuity of video segmentation in the target domain can be estimated and enforced without reference to videos in the source domain. This motivates a new framework of

Image-to-Video Domain Adaptive Semantic Segmentation (I2VDA), where the source domain is a set of images without temporal information. Under this setting, we bridge the domain gap via adversarial training based on only the spatial knowledge, and develop a novel temporal augmentation strategy, through which the temporal consistency in the target domain is well-exploited and learned. In addition, we introduce a new training scheme by leveraging a proxy network to produce pseudo-labels on-the-fly, which is very effective to improve the stability of adversarial training. Experimental results on two synthetic-to-real scenarios show that the proposed I2VDA method can achieve even better performance on video semantic segmentation than existing state-of-the-art video-to-video domain adaption approaches.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

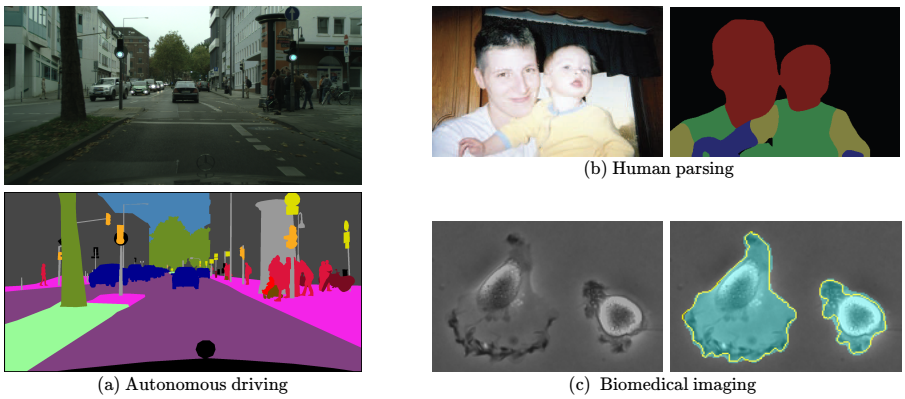
Computer vision is a field of study which helps computers see the world and understand the visual scenes as our humans. Based on different goals we want to achieve, plenty of understanding tasks are introduced such as image classification [26], object detection [73] and semantic segmentation [31]. As shown in Figure 1.1, an input image can be recognized at image level via image classification to answer the question "what's in the image?" and object detection makes one more step to locate the objects with bounding boxes. Furthermore, fine-grained pixel-level classification results can be achieved by semantic segmentation.

(a) Input (b) Image classification (c) Object detection (d) Semantic segmentation

Figure 1.1 Understanding a visual scene in different levels.

Among all three levels, semantic segmentation shows the most comprehensive results for scene understanding, *i.e.*, assigning every pixel of a given image with a pre-defined object category and indicating the location and shape of objects as well, and this dissertation is focused on semantic segmentation, which benefits tons of real-world applications such as autonomous driving [21], human parsing [40] and biomedical imaging [102] as shown in Figure 1.2.

Before deep learning was blooming, semantic segmentation techniques [63, 62, 78] are usually achieved by constructing Conditional Random Field models (CRF) over pixels or superpixels and leveraging features extracted/described using SIFT and RGB histograms. Several works [90, 32, 97] extract features with convolutions neural networks (ConvNets) and then achieve semantic segmentation via patchwise training



(a) Autonomous driving (b) Human parsing (c) Biomedical imaging

Figure 1.2 Semantic segmentation with different sets of pre-defined categories.

as stated in [77].

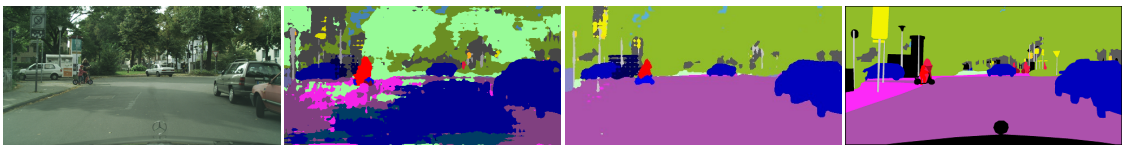
Since 2012, as convolutions neural networks (ConvNets) were allowed to go larger and deeper [61, 113, 115, 45] and showed impressive performance on image classification, researchers started to explore semantic segmentation with deep features extracted by ConvNets. In 2015, Long *et al.* [77] first adapted and extended the deep classification architectures [61, 113, 115] into semantic segmentation by supervised pre-training these deep architectures with image classification datasets and then fine-tuning full convolutionally on the segmentation task. This is achieved by simply replacing fully connected layers in the classification nets with convolutional layers which enables them to produce a map indicating the prediction probability of every pixel.

Afterward, ConvNets became a default choice for feature extraction from images in semantic segmentation. Following works further make improvements on feature representation and computational efficiency. The techniques to improve the feature representations include skip connection [102, 1, 71], atrous convolution [14], a.k.a. dilated convolutions [136], pyramid pooling/sampling modules [10, 148, 13, 11] and attention mechanisms [34, 54]. On the other hand, high efficiency can be achieved by 1) restricting input size [126, 147]; 2) designing light-weighted network [1, 94, 17,

134].

All of the above advanced ConvNets are supervised approaches that achieve high-quality results with the help of a substantial amount of dense pixel annotations. However, data collection and annotation, especially for a dense prediction task, is very time-consuming and tedious. Therefore, weakly and semi-supervised approaches [24, 49, 93, 58, 95] were also introduced to reduce the cost of dataset annotation. However, weak supervision such as bounding boxes still requires human effort to annotate.

Another direction is to generate synthetic datasets by rendering from video games, *e.g.*, GTA5 [100] and SYNTHIA [103], where the very first image is labeled with efforts and the rest are obtained via a partially automated label propagation based on mesh, texture and shader. However, the models trained on these synthetic datasets do not work well on the real-world images due to the difference of their data distributions as shown in Figure 1.3.



Input w/o adaptation w/ adaptation [119] GT

Figure 1.3 Semantic segmentation with and without domain adaptation.

Unsupervised domain adaptation (UDA) approaches, therefore, are proposed to bridge the domain gap, so that a model trained on one dataset (Source Domain) can generalize well to the others (Target Domains) without using any labels from the target datasets. The assumption here is that the source and the target domains should be very related and have common classes. In this research, we only focus on the single source domain adaptation problem, *i.e.*, only using one labeled dataset as the source domain. The first UDA framework for semantic segmentation was developed by Hoffman *et al.* [48] in 2016. As shown in Figure 1.4, it performed both global

and local alignments via adversarial training [41] and category-specific adaptation techniques [95].

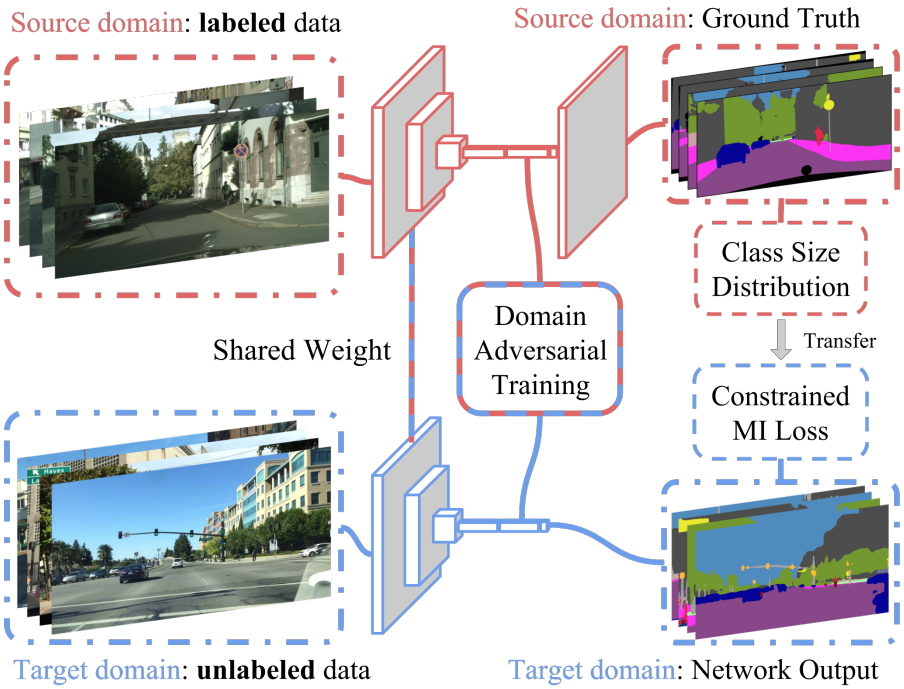


Figure 1.4 The framework of the UDA method proposed in [48].

After that, adversarial training becomes a major technique for domain adaptive semantic segmentation in a global view [117, 47, 119, 157, 91]. Adversarial training helps bridge the domain gap by training a classifier/discriminator to distinguish between the source and target domains, which the semantic segmentation network does not expect to see. Another useful strategy is self-training [158], which is carried out by alternately generating pseudo-labels for the target domain and re-training on the target domain for multiple rounds. Similarly, self-training is also a very effective way to improve the performance on the target domain and widely used in recent works [69, 157, 70, 59, 123, 133, 84, 144, 44]. More detailed introduction of these two techniques can be found in Section 2.3. In this research, we study domain adaptive or generalizable approaches for cross-domain semantic segmentation problems.

1.2 Challenges

Despite the great progress made by existing supervised methods and domain adaptation methods, there are still challenges to be addressed for the cross domain problems. These challenges mainly come from the data collection and domain gaps.

1.2.1 One/Few shot

The one or few-shot setting in UDA means using a small number of unlabeled target images to bridge the domain gap with the source. This is also a practical setting in the real world. For example, it could be hard to collect a large number of images under extreme weather conditions in the driving scenario. Most existing UDA methods fail in this setting, especially the ones that employ discriminators [117, 119, 91] to classify the source and target domains since they are prone to over-fit on the very few target samples.

1.2.2 Complicated Domain Gap

Domain gap is used to describe the distance between the source and target domains. Most of existing UDA approaches [48, 117, 158, 47, 119, 157, 69, 157, 70, 91, 59, 123, 133, 84, 144, 44] focus on the synthetic-to-real (Figure 1.5 (a)) scenario where the source is a labeled synthetic dataset and target is an unlabeled real-world dataset. The source dataset mainly contains daytime images and the target is fully daytime.

However, in the real world, there are hard domains due to the change of illumination and weather. For example, the nighttime domain is also commonly seen in autonomous driving scenario since people also drive cars in the evening. Compared with the daytime, obviously, the nighttime is a harder domain due to the low-light and glares. For the same reason, it is difficult to accurately annotate every pixel on the nighttime images (Figure 1.5 (b) bottom). Therefore, UDA can be considered as a solution for nighttime image semantic segmentation and we can pick fully la-

beled daytime dataset as the source. Because of the large domain gap between the day and night, existing works proposed for synthetic-to-real do not work well on this day-to-night setting.

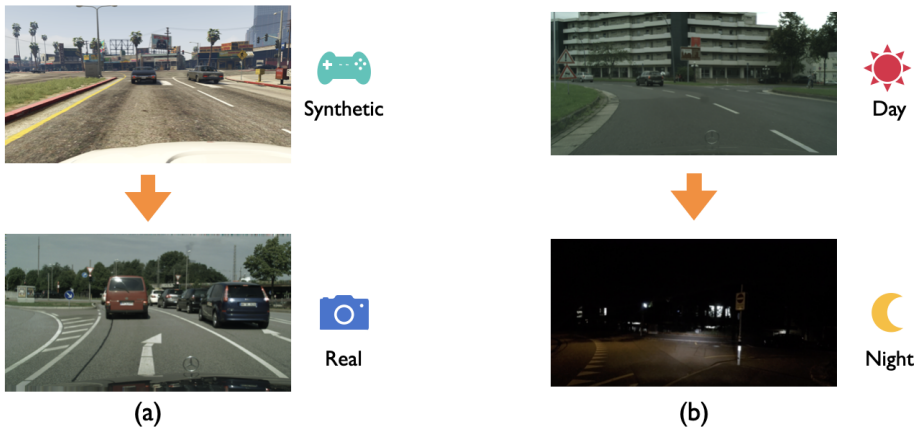


Figure 1.5 Comparison of domain gaps between synthetic-to-real and day-to-night.

Except for the nighttime domain, there are also other challenge domains that can be explored, as shown in Figure 1.6, which bring a challenge for the cross-domain semantic segmentation.

Figure 1.6 Different challenging domains in semantic segmentation from ACDC [107] dateset.

Video is also a common data format for real-world applications such as autonomous driving where a large number of videos are collected and processed for decision making. A very related research direction is video semantic segmentation [33, 36, 154, 75, 89] to perform semantic segmentation on every frame. The annotation burden of the video semantic segmentation is also very expensive where UDA

approaches can be considered, *i.e.*, using an unlabeled video dataset as the target. Unlike the image-level dataset, videos contain both temporal and spatial information. The challenge here includes the choice of the source dataset and the complicated domain gap.

1.3 Scope of the Proposed Research

To overcome the challenges, in this research, we explore three settings of cross-domain semantic segmentation conditioned on the target domain.They are 1) sole unlabeled target image is used; 2) multiple unlabeled target images are used; 3) unlabeled target videos are used, respectively. The first is to address the one/few shot setting in UDA (Section 1.2.1) and the second and third both are to address the complicated domain gap issue in UDA (Section 1.2.2). The three studies also cover different cross-domain scenarios including synthetic-to-real, cross time of day and cross weather conditions.

1.3.1 Adaptive Instance Normalization

As described in Section 1.2.1, using only one or few unlabeled target image(s) for domain adaptation is a practical but challenging setting since discriminators are not useful. In this study, we explore the usage of adaptive instance normalization (AdaIN) to make full use of the single unlabeled target image. Initially, AdaIN [53] was introduced to render a content image in the style of another one in real time. As shown in Figure 1.7, the generated image keeps the original content but its global style gets closer to the image that provides the style.

The key component of [53] is the AdaIN layer which is a simple extension to instance normalization layer. The computation of AdaIN layer is expressed as:

$$\text{AdaIN}(x, y) = \sigma(y) \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \mu(y), \tag{1.1}$$

where x and y represent the content and style feature, respectively, μ and σ denote the function for computing channel-wise mean and variance. Our study leverages

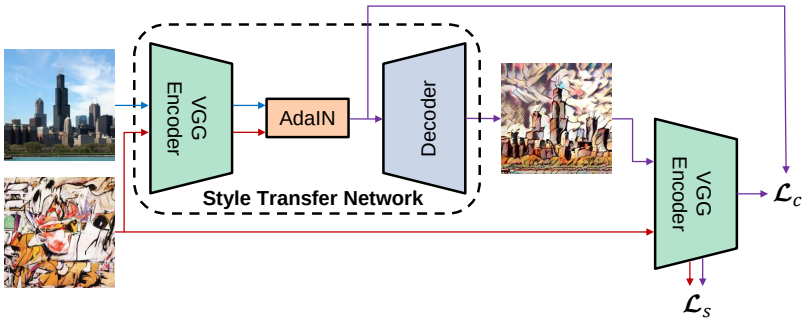


Figure 1.7 The style transfer pipeline of work [53].

this operation to align the source and target domains using very few unlabeled target samples.

1.3.2 Image Correspondence

In this study, we exploit coarse image correspondence to address the large domain gap challenge in UDA. Intuitively, an intermediate domain that shares similarities with both the source and target domain might help bridge the large domain gap. This idea of leveraging the intermediate domain and image correspondence is illustrated in Figure 1.8 based on the setting of day-to-night scenario, where the intermediate domain image (b) shares similarity with the source image (a) in illumination and has common objects with the target image (c).

Figure 1.8 The motivation of leveraging the image correspondence for nighttime semantic segmentation. (a) and (c) are source and target, respectively. (b) is an unlabeled daytime image serving as the intermediate domain which is taken at the same position as the target one (nighttime image) and aligned by GPS recordings.

This intermediate domain is selected because of the following two reasons: 1) It has similar illumination patterns as the source, thus it is easy to close the gap between source and the intermediate domain; 2) It shares the same content, *i.e.*, static objects such as trees, building, *e.t.c.*, so prediction of those aligned part should be same as well. Note that this design relies on the coarse aligned pair data collection by [106]. But compared with the pixel-wise data annotation, collecting unlabeled pair images is still easier. It is also worth mentioning that this idea is general and can be extended to other large domain gap cases given the corresponding reference for the target.

1.3.3 Temporal Consistency Learning

Employing UDA approaches for video-level tasks is usually more complicated than for image-level ones. For images, the domain gap can be observed from only the appearance. However, videos contain both spatial and temporal information which can lead to a domain gap with other videos. To achieve unsupervised video semantic segmentation, two recent works [42, 112] coincidentally suggest transferring knowledge from videos to videos by using a labeled video dataset as the source and an unlabeled video dataset as the target. In this way, both spatial and temporal information can be passed from the source to the target. However, should we transfer both of them to the target?

In this study, we make a fundamental hypothesis that it is needed to pass the spatial knowledge from the source domain to the target domain, not the temporal one, for video domain adaptation. The reasons are: 1) the between-frame continuity is the most important temporal knowledge for video semantic segmentation which can be well-exploited in target videos themselves, and 2) the temporal information between the source and target domains practically may not show a systematic domain gap that has to be filled by adaptation. Therefore, in this study, we address the unsupervised video semantic segmentation by introducing an image-to-video domain

adaptation framework, as illustrated in Figure 1.9.

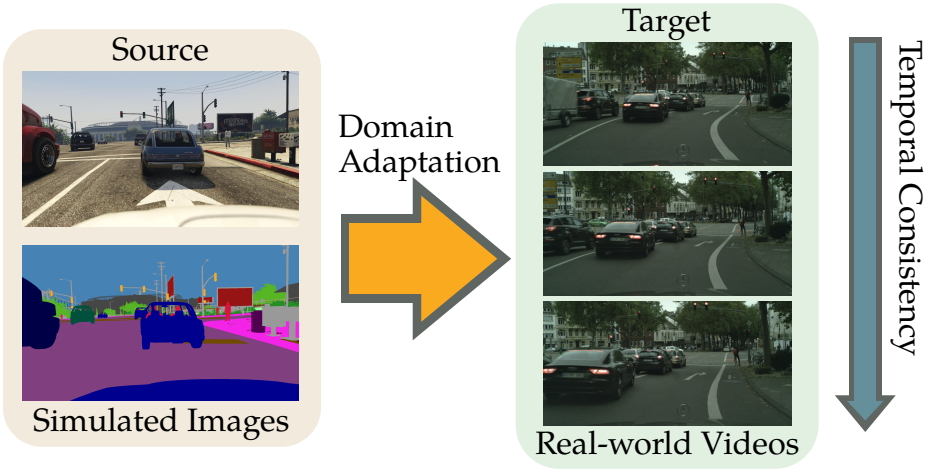


Figure 1.9 An illustration of the setting for the proposed image-to-video domain adaptive semantic segmentation. We use a labeled image dataset as the source and unlabeled videos as the target.

1.4 Proposed Approaches

In this section, the proposed approaches are introduced according to the above research scopes.

1.4.1 Style Mixing and Patchwise Prototypical Matching for One-Shot Unsupervised Domain Adaptive Semantic Segmentation

In the first study, we propose a new approach to address the one-shot setting in UDA as introduced in Section 1.3.1. Specifically, we integrate several style-mixing layers into the segmentor which play the role of a style-transfer module to stylize the source images without introducing any learned parameters. Moreover, we propose a patchwise prototypical matching (PPM) method to weighted consider the importance of source pixels during the supervised training to relieve the negative adaptation. Experimental results show that our method achieves new state-of-the-art performance

on two commonly used benchmarks for domain adaptive semantic segmentation under the one-shot setting and is more efficient than all comparison approaches.

1.4.2 A One-Stage Domain Adaptation Network for Unsupervised Nighttime Semantic Segmentation

In the second study, we propose a novel domain adaption network (DANNet) for nighttime semantic segmentation which leverages the image correspondence introduced in Section 1.3.2. It employs adversarial training using a labeled daytime dataset and an unlabeled dataset that contains coarsely aligned day-night image pairs. Specifically, for the unlabeled day-night image pairs, we use the pixel-level predictions of static object categories on a daytime image as a pseudo supervision to segment its counterpart nighttime image. We further design a re-weighting strategy to handle the inaccuracy caused by misalignment between day-night image pairs and wrong predictions of daytime images, as well as boost the prediction accuracy of small objects. The proposed DANNet is the first end-to-end method for nighttime semantic segmentation, which does not train additional day-night image transfer models as a separate pre-processing stage. Extensive experiments on Dark Zurich and Nighttime Driving datasets show that our method achieves state-of-the-art performance for nighttime semantic segmentation.

1.4.3 Image to Video domain adaptive semantic segmentation

In the last study, we propose and verify a new finding – for segmenting real videos, it is sufficient to perform domain adaptation from synthetic *images*, instead of synthetic *videos*, i.e., there is no need to adapt and transfer temporal information in practice. This finding is observed by introducing the setting of image-to-video domain adaptive semantic segmentation, *i.e.,* using labeled images as the source domain in domain adaptation for video semantic segmentation. There are also two novel designs in the

proposed domain adaptation framework including: 1) a temporal augmentation strategy to better exploit and learn diverse temporal consistency patterns in the target domain; and 2) a training scheme to achieve more stable adversarial training with the help of a proxy network. Experimental results on two synthetic-to-real scenarios demonstrate the effectiveness of the proposed method and verify our fundamental hypothesis. Without simulating/adapting temporal information in the source domain, our method still outperforms existing state-of-the-art video-to-video domain adaptation methods.

1.5 Structure of the Dissertation

The rest of this dissertation is organized as follows. In Chapter 2, the related background and techniques are introduced. In Chapter 3, a literature review for related works is conducted. In Chapter 4, the proposed one-shot unsupervised domain adaptation method is presented. In Chapter 5, a one-stage unsupervised domain adaptation method for nighttime semantic segmentation is presented. In Chapter 6, a novel image to video domain adaptive semantic segmentation method is presented. Finally, Chapter 7 concludes the dissertation and discusses the future directions.

Chapter 2

Background

In this chapter, we first introduce the background and concepts of deep ConvNets which serves as the cornerstone of existing semantic segmentation methods as well as this research. After that, several famous semantic segmentation methods and those used in this research are reviewed and discussed. Then, widely-used datasets for semantic segmentation are presented as well as common cross-domain scenarios. Finally, we summarize the evaluation metrics.

2.1 Background and Concepts in Deep ConvNets

2.1.1 History of Deep ConvNets

The history of Deep ConvNets can be dated back to 1980 when Neocognitron [35] is proposed. It introduced the concepts of layer-by-layer feature extraction, pooling and also used ReLU (Eq.(2) in [35]) to provide non-linearity and finally used for recognition tasks. In 1989, LeCun *et al.* [64, 66] first applied back-propagation learning [105] to the digit recognition task which is the next milestone in the development of Deep ConvNets. In 1998, the name of convolutional neural network came out alone with a new architecture dubbed LeNet5 [65] (Figure 2.1) which is the first modern ConvNets in the world.

Due to the limitation of the computational power and very strong competitors, *i.e.*, SVM, ConvNets were not very popular at that time. However, in 2012, a breakthrough was made by Krizhevsky *et al.* who designed the AlexNet [61] (Figure 2.1) which combined newly proposed dropout operation to avoid the over-fitting problem and ideas from prior works such as ReLU [35] and data augmentation [125]. Besides, the success also cannot be divorced from the large-scale ImageNet dataset [26] and the development of Graphics Processing Units (GPUs) [7, 18].

From then on, more and more researchers started to explore the designs of ConvNets and the ImageNet Large Scale Visual Recognition Challenge (ILSRVC) became a very important tool to evaluate their designs. In 2013, ZFNet [141] became the win-

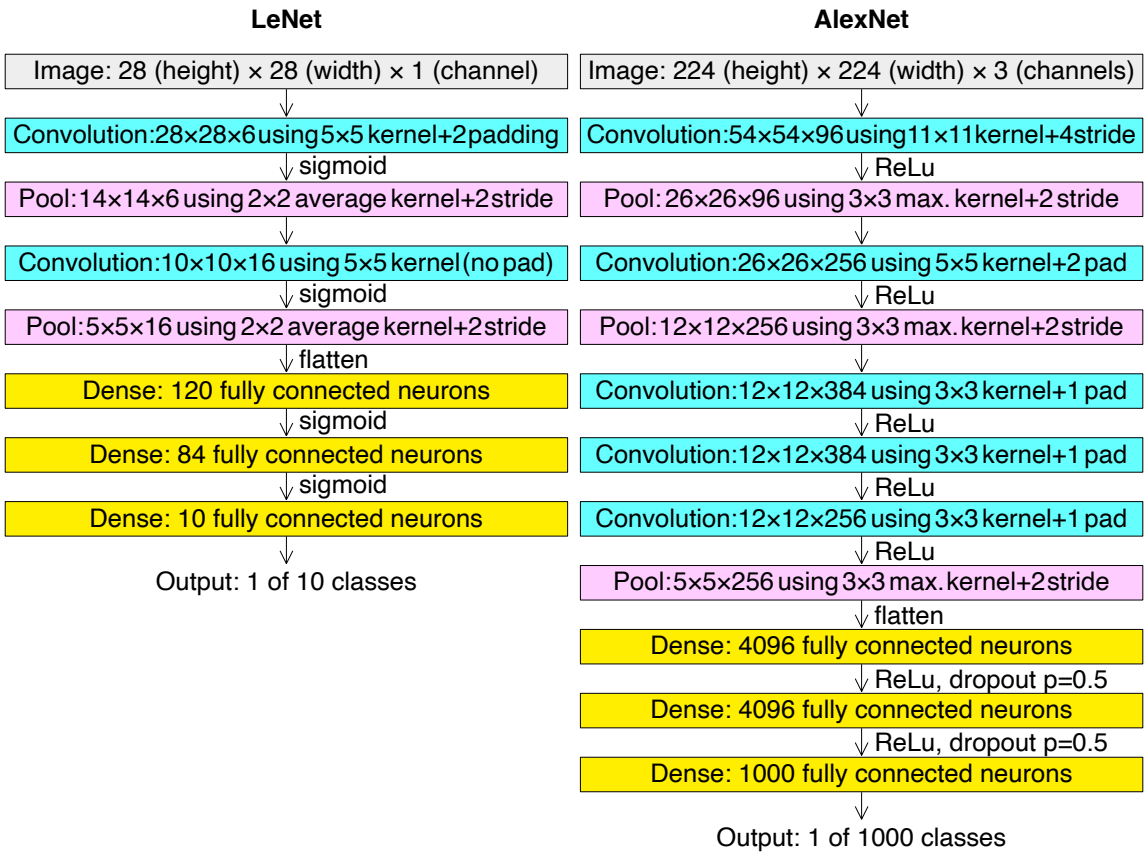


Figure 2.1 The architectures of LeNet and AlexNet from [19].

ner of the ILSRVC which employed the same architecture as AlexNet (8 layers) with different hyper-parameters choices. In 2014, VGGNet [113] (16-19 layers, Figure 2.2) was published which showed deep ConvNets can be achieved with small size convolutional kernels and GoogLeNet [115] (22 layers) increased both depth and width while keeping the computational budget constant with the inception module.

However, stacking layers to make ConvNets deeper may obtain even worse results than the shallow ones due to the vanishing gradient problem. In 2015, He *et al.* [45] (Figure 2.3) came up with the idea of residual learning to address this problem. Their proposed ResNet was built by stacking a large number of residual blocks with skip connections and at most resulted in a 152-layer ConvNet without compromising the performance. The simple yet very effective idea of residual learning later was extended

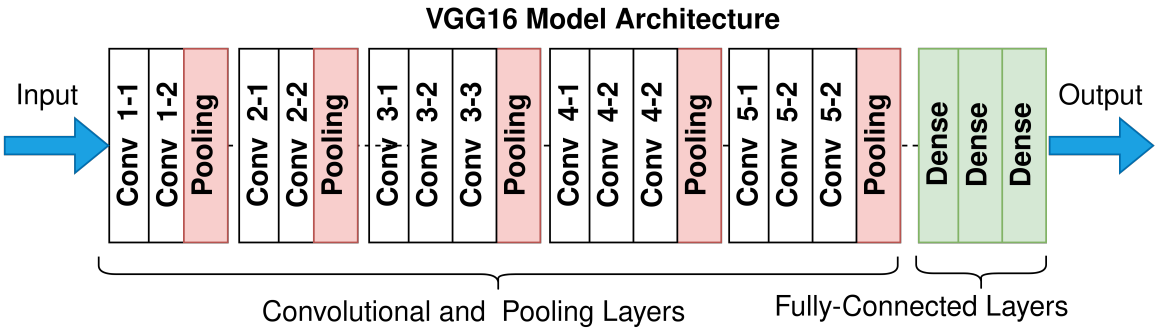


Figure 2.2 The architecture of VGG-16 from [20].

in several architectures such as WideResNet [140], ResNext [129] and DenseNet [51]. Until now, ResNet is still the most frequently used network architecture that benefits various applications for feature extraction.

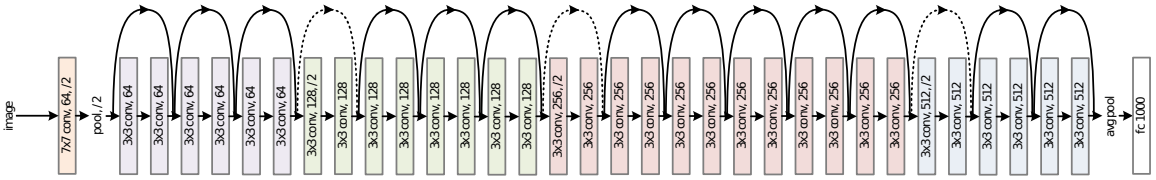


Figure 2.3 The architecture of ResNet-34 [45].

2.1.2 Components of Deep ConvNets

Most Deep ConvNets are usually a sequence of convolutional layers and pooling layers. Fully connected layers are not required, but sometimes they are used at the end of the ConvNets for high-level reasoning.

The main component of Deep ConvNets is convolutional layers which is customized with hyper-parameters including output channel, kernel size, padding and stride. Given the input in a shape of $c_{in} \times h_{in} \times w_{in}$ and kernel size, padding, and stride are denoted as $k \times k$, p and s, respectively. The number of trainable parameter in this convolutional layer will be $c_{in} \times c_{out} \times k \times k$ (without considering the bias), if

we want to obtain an output that has c_{out} channels. The resolution of the output is $h_{out} \times w_{out}$ can be computed by:

$$h_{out} = \lceil \frac{h_{in} + 2p - k}{s} \rceil + 1, \tag{2.1}$$

and

$$w_{out} = \lceil \frac{w_{in} + 2p - k}{s} \rceil + 1. \tag{2.2}$$

The computation of the convolution operation is illustrated in Figure 2.4.

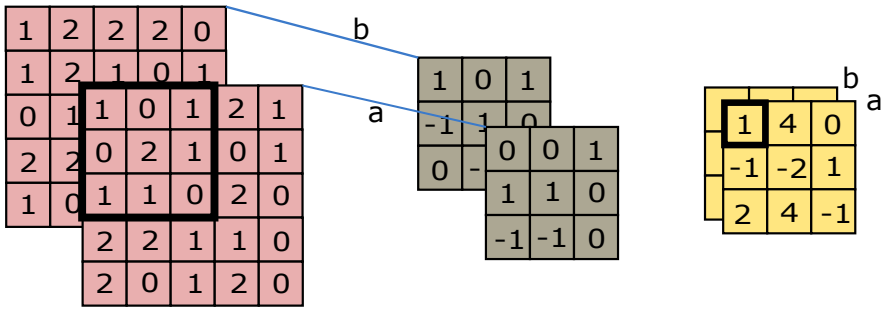


Figure 2.4 An illustration of computation in the convolutional layer. In this example, the shape of input is $2 \times 5 \times 5$, c_{out}, k, s and p is set to 1, 3, 1, and 0, respectively. So we require two kernels with each one taking care of each channel of the input. The resulted yellow feature maps a and b are summed up to get the output with a shape of $1 \times 3 \times 3$.

A ConvNet with only convolutional layers is a linear regression model which is hard to learn complex function mappings. Therefore, ConvNets require non-linear activation functions to help themselves learn complex mappings between network inputs and outputs. Generally, activation functions are applied to the output of the convolutional layers. The three commonly used non-linear activation functions are Sigmoid, TanH and ReLU, as shown in Figure 2.5.

Among the three functions, both the Sigmoid and TanH might cause the vanishing gradient problem. Because there is almost no change to the output when the function input value is very high or very low. Thus, the current most popular activation function is ReLU.

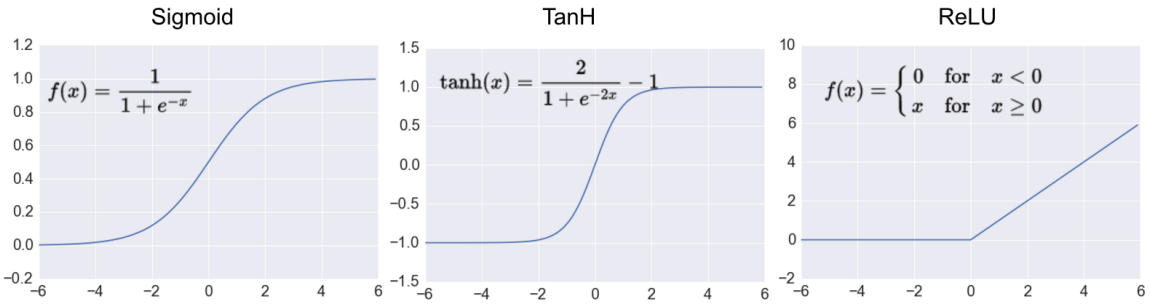


Figure 2.5 Commonly used activation functions in ConvNets from [86].

Pooling layers are also important parts of ConvNets which are used for feature map down-sampling. Usually, a pooling layer is inserted after an activation function (*e.g.*, ReLU). Average pooling and MAX pooling are two frequently used pooling layers in ConvNets as illustrated in Figure 2.6. Average pooling computes the average value for each window on the feature map and MAX pooling selects the maximum value in each window on the feature map. Same as convolutional layers, pooling layers are also controlled by hyper-parameters including kernel size, stride and padding. But they do not contain any trainable parameters at all.

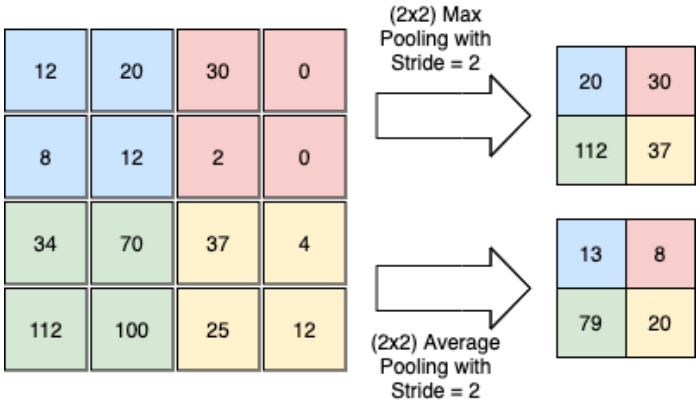


Figure 2.6 An illustration of average pooling and MAX pooling from [30].

Pooling operation helps ConvNets select important features and reduce the com-

putation burden as well. Besides, the feature selection also plays the role of data augmentation during training, thus it makes ConvNets become more robust to small input translations and relieves the over-fitting problem.

The fully connected layer is optional in the design of ConvNets where every single neuron has a connection to everyone in the next layer. It can help the network to make a prediction based on the whole input image globally. From the perspective of implementation, a fully connected layer can be replaced by a convolutional layer followed by a result reshaping to obtain the desired dimension. There is no fully connected layer in the fully convolutional networks [77].

2.2 Typical Deep Semantic Segmentation Networks

The development of the semantic segmentation is built upon the state-of-the-art deep ConvNets. In this section, we discuss the ones that played important roles in literature.

2.2.1 FCN

The first deep ConvNets for semantic segmentation is Fully Convolutional Network (FCN) [77] introduced by Long *et al.* . They adapted contemporary classification networks such as AlexNet [61], VGG [113] and GoogLeNet [115] into fully convolutional networks by replacing the fully connected layers into convolutional layers which makes the networks can efficiently perform pixel-wise prediction for arbitrary-sized inputs. The architecture of a FCN is shown in Figure 2.7.

It employed a sequence of convolutional layers and pooling layers to extract deep features and deconvolution operation to upscale the feature map. As a side effect, the deconvolution will lead to dissatisfyingly coarse results. This problem was relieved by adding links to fuse the prediction from higher layers and lower ones as shown in

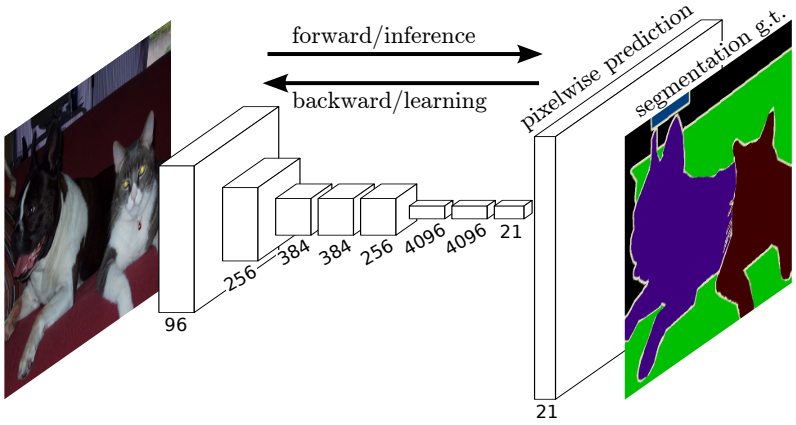


Figure 2.7 An illustration of FCN [77] built upon AlexNet.

Figure 2.8. As time goes by, more and more advanced approaches outperform FCN [77] on current benchmarks while keeping the idea of FCN in their network design.

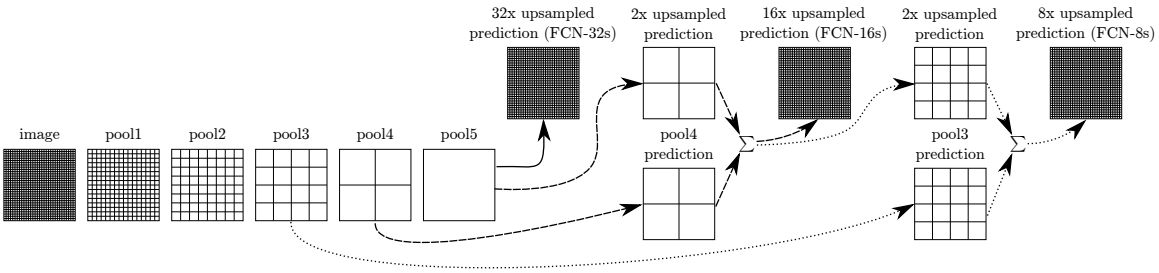


Figure 2.8 An illustration of the skip connection in FCN [77].

2.2.2 U-Net

Based on FCN [77], Ronneberger *et al.* designed U-Net [102] for biomedical image semantic segmentation. The major differences are: 1) The encoder and decoder are symmetric in U-Net, thus U-Net has a larger decoder. Instead, the decoder of FCN is a single deconvolution layer which will lead to a coarse prediction; 2) The skip connection in FCN is achieved by pixel-wise summation and concatenation in U-Net between the features from lower and higher layers. Compared with the summation,

concatenation can introduce more feature channels which allow the decoder to propagate context information to higher resolution layers.

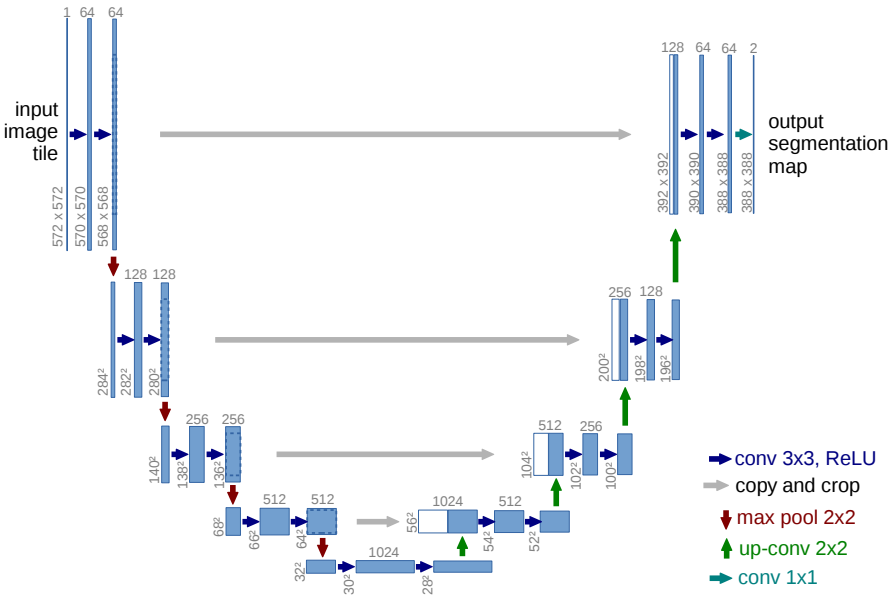


Figure 2.9 An illustration of the U-Net [102].

The feature concatenation also increases the memory cost. To relieve this burden, Badrinarayanan *et al.* [1] propose to record only the indices from the MAX pooling operation during feature encoding. In the decoder part, feature upsampling is achieved via unpooling operation, *i.e.*, filling a feature map based on the indices.

2.2.3 DeepLab Series

DeepLab series is another follower of FCN that contains four iterations. The DeepLab V1 [14] was published 2015. Chen *et al.* adapted the "atrous algorithm" into convolutional layers to efficiently compute dense feature maps at any target subsampling rate without introducing any approximations. The same operation was done in [136] dubbed "dilated convolution" where convolutional layers with multiple dilation rates were used for multi-scale context aggregation.

Then in 2017, the updated version (V2) [10] came out with the following major contributions: 1) Introducing the atrous spatial pyramid pooling (ASPP) layer to robustly segment objects at multiple scales; 2) Replacing the VGG-16 used in V1 with contemporary state-of-the-art classification network ResNet [45]. The ASPP module is an extension of the R-CNN spatial pyramid pooling (SPP) [46] which originally proposed for recognition and detection tasks to generate a fixed-length representation regardless of input image size/scale. With multiple pooling layers, SPP creates feature representations with multiple scales. Here, ASPP achieves the same goal by using multiple parallel atrous convolutional layers with different sampling rates instead of MAX pooling layers. An illustration of the ASPP module is shown in Figure 2.10.

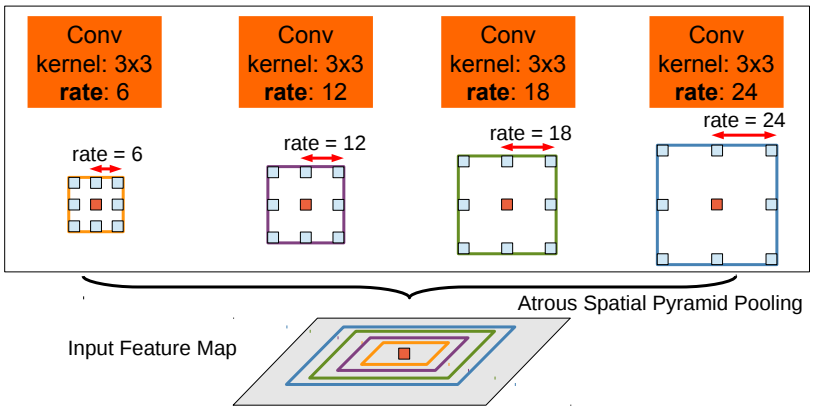


Figure 2.10 An illustration of the ASPP module proposed in [10].

Later in [13] (V3), Chen *et al.* had more discussions on the layout of the atrous convolution layers in both cascade and in parallel (*i.e.*, ASPP) and the experiments showed that the parallel structure was better. Besides, the ASPP in V3 was equipped with batch normalization layers and took global context information into consideration to overcome the problem introduced by large sampling rate. Following [25, 121], multi-grid method was also applied into the network backbone and achieved performance gains.

The last version of the DeepLab series is [11] (V3+) which employed V3 as a

powerful encoder and designed a simple yet effective decoder to form a novel encoder-decoder structure. The atrous convolution layers were still important components heavily used in their design. To improve the efficiency and accuracy, V3+ also adapted the Xception model [17] for the segmentation task and applied depthwise separable convolution to both ASPP and decoder to improve the efficiency and accuracy.

2.2.4 PSPNet

In 2017, Zhao *et al.* proposed a pyramid scene parsing network (PSPNet) [148] for semantic segmentation as illustrated in Figure 2.11. They adapted SPP module [46] into semantic segmentation as DeepLab V2 [10] dubbed Pyramid pooling module (PPM) in their network. Different from SPP, PPM upsamples and concatenates the multi-scale features maps as the final feature representation. They also proposed to train the network with an auxiliary loss for effective optimization.

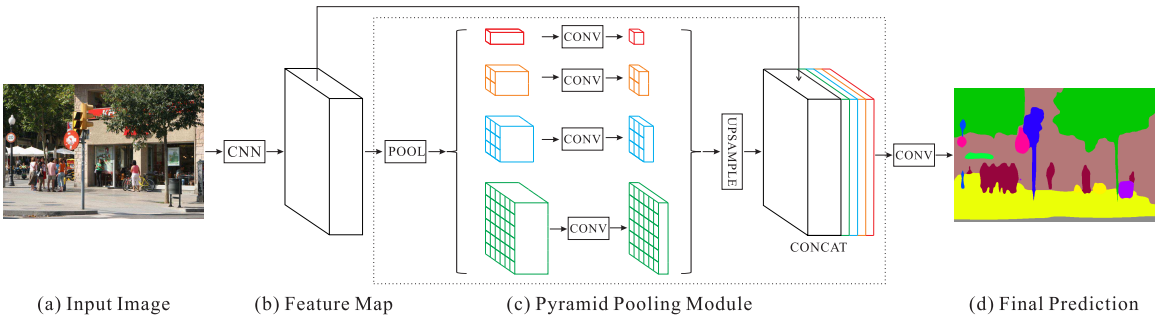


Figure 2.11 An illustration of PSPNet [148].

2.2.5 RefineNet

Multi-path refinement network, RefineNet [71], was also published in 2017 for semantic segmentation. It was proposed to address two problems: 1) Using ResNet suffers from the downscaling problem which is not suitable for semantic segmentation; 2) Dilated convolution can keep the feature size but increase the computational cost and memory requirement. As shown in Figure 2.12, RefineNet first extracts features

from layers of ResNet and refine the low-resolution features with fine-grained low-level features in a recursive manner to generate the final high-resolution semantic feature maps.

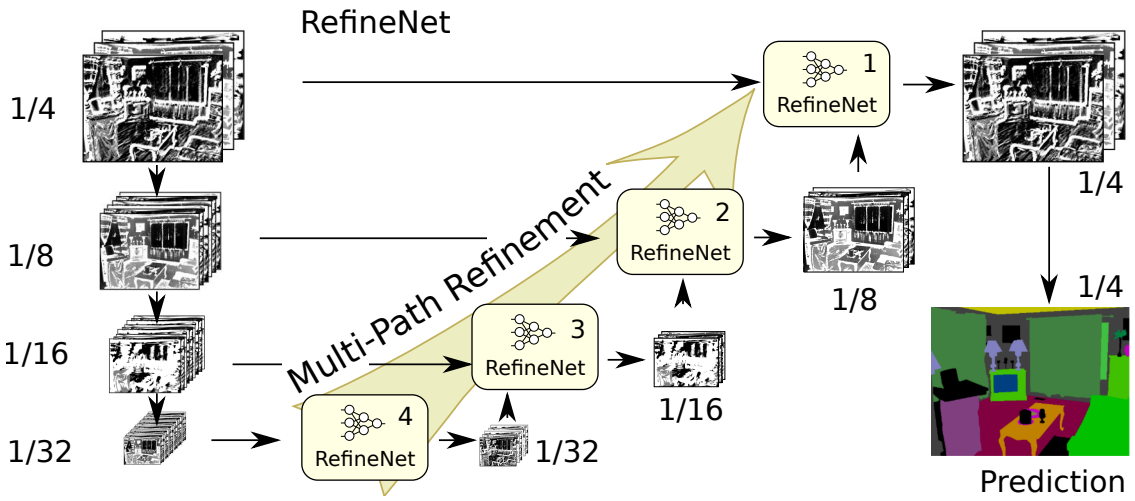


Figure 2.12 An illustration of RefineNet [71].

All the semantic segmentation networks discussed above are summarized in Table 2.1. Except for these manually designed architectures, researchers also learned the model architectures directly on the dataset of interest, *i.e.*, neural architecture search (NAS) [156]. In 2019, Liu *et al.* [74] made the first attempt to extend NAS to semantic segmentation and automatically build the network by searching both the structure of the cell and their connections.

All the above methods can be categorized as supervised methods that require large-scale datasets for network training which leads to a high cost in dataset collection for this dense prediction task. Besides, the trained models cannot generalize well to unseen scenarios.

Table 2.1 Summary of typical semantic segmentation methods and the backbone they used to achieve the best performance. * means the backbone is only used as encoder.

Backbone	Seg. Method	Year
VGG-16 [113]	FCN [77]	2014
VGG-16 [113]	DeepLab V1 [14]	2014
VGG-16* [113]	U-Net [102]	2015
ResNet-101 [45]	DeepLab V2 [10]	2017
ResNet-152 [45]	RefineNet [71]	2017
ResNet-269 [45]	PSPNet [148]	2017
ResNet-101 [45]	DeepLab V3 [13]	2017
ResNet-101 [45]	DeepLab V3+ [11]	2018

2.3 Domain Adaptive Semantic Segmentation Methods

Domain adaptive semantic segmentation, a.k.a. unsupervised domain adaptation (UDA), aims to transfer the knowledge from existing labeled datasets (Source Domain) to unlabeled datasets (Target Domain), *i.e.*, training a model for the target domain without using its label. Therefore, these UDA approaches can help relieve the pixel-wise annotation burden as well as improve the generalization ability to exist semantic segmentation methods. In general, existing domain adaptative semantic segmentation methods can be divided into two types: adversarial training and self-training. Table 2.2 summaries most of existing domain adaptive semantic segmentation methods.

2.3.1 Adversarial Training

A typical adversarial training-based approach is AdaptSegNet [117] whose framework is shown in Figure 2.13. During training, a weight-sharing semantic segmentation (DeepLab V2) serves as a generator to generate segmentation results for both the source and target images. The discriminator is a binary classification network to distinguish whether its inputs come from the source or target domain. Besides, the

Cross-Entropy loss is computed using the source output and its ground truth in each training iteration.

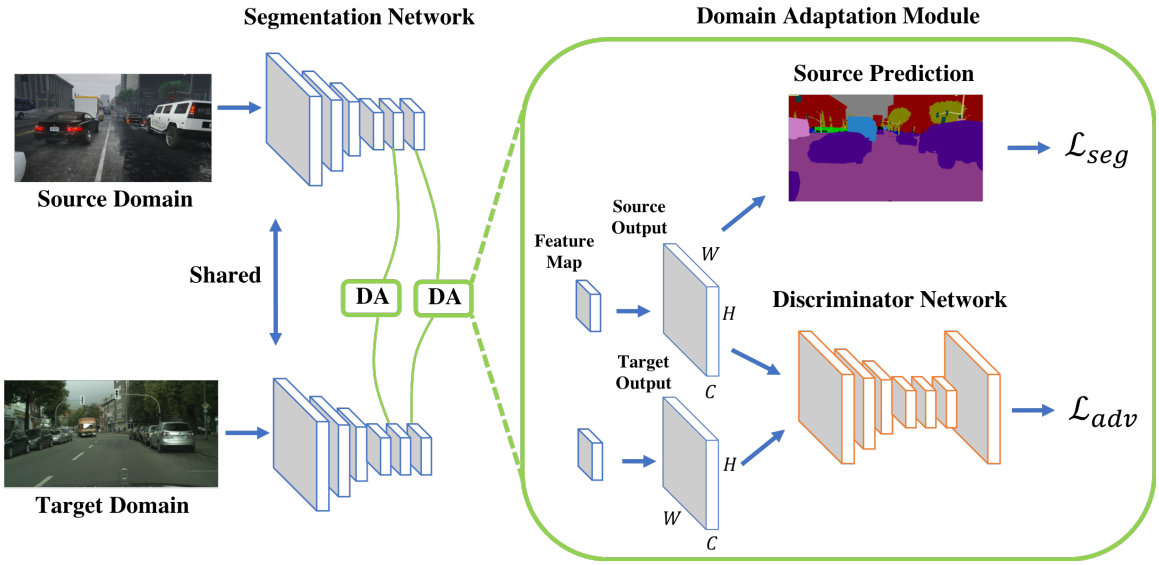


Figure 2.13 The domain adaptation framework proposed by Tsai *et al.* [117].

2.3.2 Self-Training

The self-training strategy currently almost becomes a default setting in recent UDA methods. Usually, it works together with the adversarial training for domain adaptation. Adversarial training is the first step for aligning the source and target domains and the trained model weights can be used to generate pseudo-label for the target images. Then, the confident pseudo-labels will be selected as the ground truth paired with the corresponding target image for network re-training which can be repeated multiple times for more performance gains.

2.4 Evaluation Metrics

There are in total four metrics for segmentation result evaluation, *i.e.*, pixel-wise accuracy (PA), mean pixel-wise accuracy (MPA), mean of class-wise intersection

Table 2.2 Summary of domain adaptive semantic segmentation methods. Seg. Network indicates the used segmentation network. AT and ST represent adversarial training and self-training, respectively.

DA method	Backbone	Seg. Network	AT	ST	Year
FCNs in the Wild [48]	VGG-16 [113]	FCN [77]	✓		2016
CYCADA [47]	DRN-26 [137]	DRN-26 [137]	✓		2018
CDA [146]	VGG-19 [113]	FCN [77]	✓		2018
AdaptSegNet [117]	ResNet-101 [45]	DeepLab V2 [10]	✓		2018
CBST [158]	ResNet-38 [45]	ResNet-38 [45]		✓	2018
ADVENT [119]	ResNet-101 [45]	DeepLab V2 [10]	✓		2019
CRST [157]	ResNet-101 [45]	DeepLab V2 [10]		✓	2019
PyCDA [70]	ResNet-101 [45]	DeepLab V2 [10]		✓	2019
BDL [69]	ResNet-101 [45]	DeepLab V2 [10]	✓	✓	2019
IntraDA [91]	ResNet-101 [45]	DeepLab V2 [10]	✓	✓	2020
Stuff&Things [123]	ResNet-101 [45]	DeepLab V2 [10]	✓	✓	2020
FDA [133]	ResNet-101 [45]	DeepLab V2 [10]		✓	2020
IAST [84]	ResNet-101 [45]	DeepLab V2 [10]	✓	✓	2020
ProDA [144]	ResNet-101 [45]	DeepLab V2 [10]	✓	✓	2021
MetaCorrection [44]	ResNet-101 [45]	DeepLab V2 [10]		✓	2021
DAFormer [50]	Mix Transformer [128]	SegFormer [128]		✓	2022

over union ($MIoU$) and frequency weighted intersection over union ($FWIoU$). They are defined as follows:

$$PA = \frac{\sum_{i=0}^{k} n_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} n_{ij}}, \tag{2.3}$$

$$MPA = \frac{1}{k+1} \sum_{i=0}^{k} \frac{n_{ii}}{\sum_{j=0}^{k} n_{ij}}, \tag{2.4}$$

$$MIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{n_{ii}}{\sum_{i=0}^{k} n_{ij} + \sum_{j=0}^{k} n_{ji} - n_{ii}}, \tag{2.5}$$

$$FWIoU = \sum_{i=0}^{k} \frac{\sum_{j=0}^{k} n_{ij}}{\sum_{i=0}^{k} \sum_{j=0}^{k} n_{ij}} \frac{n_{ii}}{\sum_{i=0}^{k} n_{ij} + \sum_{j=0}^{k} n_{ji} - n_{ii}}, \tag{2.6}$$

where $k+1$ is the number of classes including the background, n_{ii} is the number of pixels that are correctly classified as the i-th class (True Positive), n_{ij} is the number

of pixels labeled with the i-th but are misclassified as the j-th class (False Negative), and n_{ji} is the number of pixels are misclassified as the i-th class (False Positive).

PA is easy to compute but its scores are usually dominated by the large size classes, thus cannot truly reflect the performance when we focus on the small size classes. This problem can be relieved by computing MPA. Among the four evaluation metrics, $MIoU$ is the most frequently used one. $FWIoU$ is a weighted average of IoUs by the frequency of each class.

Chapter 3

Literature review

This chapter provided a literature review for the related works of the three studies in this dissertation.

3.1 Style Mixing and Patchwise Prototypical Matching for One-Shot Unsupervised Domain Adaptive Semantic Segmentation

3.1.1 Unsupervised domain adaptation and domain generalization

OSUDA is developed from the general UDA setting. The first UDA approach for semantic segmentation was proposed in [48] using feature-level adversarial learning and category-specific adaptation. After that, adversarial learning has been applied to UDA in feature level [9, 47], output space [117, 91, 146, 96] and entropy of the prediction [119, 91] for alignment. Image translation is another approach for UDA [110, 69, 133] by exploiting advanced image-to-image translation networks, *i.e.*, CycleGAN [152], to reduce the domain discrepancy. Recently, multiple rounds of self-training with generated pseudo labels of the target domain samples was proved to be a powerful strategy to boost the adaptation performance [158, 69, 145]. However, these methods cannot be directly applied to the OSUDA setting due to the scarce of the target images.

Also related to OSUDA is the problem of domain generalization where the target domain is totally unknown. Based on the number of source domains involved during adaptive learning, existing DG approaches can be basically divided into multi-source DG [39, 29] and single-source DG [92, 139, 52]. For multi-source DG, Zhou *et al.* proposed a MixStyle [150] strategy to increase the domain diversity of the source domains. During training, two instances of different domains in a mini-batch are selected to synthesize novel domains leveraging the feature-level style statistics [53]. Single-source DG is more challenging since less labeled source data is accessible for adaptation. A typical solution is to perform domain randomization [116] on the source training samples via image stylization or translation which can also be treated

as a data/domain augmentation strategy. For example, several real-life images from ImageNet [26] are picked as randomization references [139] to adjust the source images or their domain invariant frequency components [52].

3.1.2 Prototypical Representation

Prototypes are defined as abstractions of essential semantic feature representations, which were popularly used in computer vision tasks recently. For example, Wang *et al.* design a prototype alignment regularization [120] for few-shot semantic segmentation, where the class-specific prototypes are computed via a masked average pooling. More recently, Zhang *et al.* [144] exploited the distances between the target features and the class-wise prototypes to re-weight the predicted probability for better self-training. In this study, we calculate the prototypes of the patches of the sole target image to weight the training pixels from the source domain.

3.1.3 Style-transfer

In [37], Gatys *et al.* proposed a neural style-transfer algorithm to generate high-quality artistic images by separating and recombining the content and style of arbitrary images. Later style-transfer has become an effective technique, which benefits several real-world applications such as makeup transfer and removal [5] and virtual try-on [132]. Our work is closely related to the adaptive instance normalization (AdaIN) proposed by Huang [53], which transfers the mean and variance in the feature space in real-time. The main difference is that we don't synthesize the image with a decoder.

3.2 A One-Stage Domain Adaptation Network for Unsupervised Nighttime Semantic Segmentation

3.2.1 Domain Adaptation for Semantic Segmentation

Domain adaptation methods are developed to transfer knowledge learned from source domains to target domains which share similar objects yet different data distributions. Recently, domain adaptation has been applied to help semantic segmentation. In [48], Hoffman *et al.* proposed a novel fully convolutional domain adversarial learning approach with category constraints [95] for semantic segmentation. Tsai *et al.* [117] later developed a multi-level adversarial network to perform domain adaptation in the output space.

Instead of using adversarial learning techniques, image translation and style transfer [152] from source images to target ones, or vice versa, have been widely used for domain adaptation [47, 127]. Previous works have shown that domain-invariant representations can be obtained in the process of image translation between the source and target domains [110, 153, 6]. Several recent works [69, 123, 59] made use of self-training strategies by iteratively predicting and fine-tuning a set of pseudo labels in multiple rounds of network training. Another line of researches [146, 70] adopted the curriculum-style learning by first learning easy properties in the target domain and then using it to regularize the semantic segmentation model. However, most of these general-purpose domain adaptation approaches cannot handle well the significant adaptation gap between the daytime and the nighttime images and therefore could not achieve satisfactory performance in nighttime semantic segmentation [106]. Specifically, all the above methods focus on the domain adaptation for synthetic-to-real (i.e., GTA5 [100] or SYNTHIA [103] to Cityscapes) or cross-city images (i.e., Cityscapes to Cross-City [9]), which are all daytime to daytime adaptations. In this study, we instead focus on the adaptation between the daytime and the nighttime

domains with significantly different illumination patterns [106].

3.2.2 Nighttime Semantic Segmentation

Recently, Dai *et al.* [23] leveraged an intermediate twilight domain to progressively adapt semantic models trained in daytime scenes to nighttime. Sakaridis *et al.* [106, 108] further extended it to a guided curriculum adaptation framework, which uses both the stylized synthetic images and the unlabeled real images to exploit the cross-time-of-day correspondence of the scene images. However, such gradual adaptation approaches usually need to train multiple semantic segmentation models, e.g., three models in [106] for three different domains respectively, which is highly inefficient. Following works along this line [101, 114, 88] also train some additional image transfer models, e.g., CycleGAN [152], to perform the day-to-night or night-to-day image transfer before training the semantic segmentation models. For these methods, the performance of later adaptation and semantic segmentation is highly dependent on the image transfer model pre-trained in the pre-processing stage.

Vertens *et al.* [118] proposed to leverage the thermal infrared images as a complementary input to the RGB images for nighttime semantic segmentation since thermal radiation is not very sensitive to the illumination changes. In [27], a two-stage adversarial training method was proposed for semantic segmentation of rainy night scenes by performing domain adaptation between day-night near scene pairs. Different from all the above methods, the DANNet proposed in this study performs a one-stage end-to-end adversarial learning for training the nighttime semantic segmentation network without using any other image modalities.

3.3 Image to Video domain adaptive semantic segmentation

3.3.1 Video semantic segmentation

Existing video semantic segmentation approaches can be categorized into accuracy-oriented and efficiency-oriented ones. Optical-flow-based representation warping and multi-frame prediction fusion have been employed to achieve more robust and accurate results [36, 154, 75]. An alternative solution is to use the gated recurrent layers to extract the temporal information [33] or propagate labels to unlabeled frames by means of optical flow [89]. Many strategies have been studied to improve efficiency. For example, features in each frame can be reused by adjacent frames to reduce the overall cost [111, 154, 130]. Li *et al.* [68] further proposed to reduce both of computational cost and maximum latency by adaptive feature propagation and key-frame allocation. More recently, Liu *et al.* [76] proposed to train a compact model via temporal knowledge distillation for real-time inference.

All of the above video semantic segmentation methods need the labeling on densely or sparsely sampled frames from the target domain for training. In this study, we instead use self-labeled simulated images for training and then adapt to the target domain for video semantic segmentation.

3.3.2 Domain adaptive image segmentation

In recent years, many domain adaptation approaches have been proposed for image semantic segmentation to relieve the burden of dense pixel-level labeling. Hoffman *et al.* [48] introduced the first unsupervised domain adaptation method for transferring segmentation FCNs [77] by applying adversarial learning on feature representations, which has become a standard strategy for domain adaptive semantic segmentation [9, 80]. More recently, the adversarial learning has been further extended to image level [47, 16], output level [117, 6] and entropy level [119, 91] for this task.

In [158], Zou *et al.* first suggested the self-training in the form of a self-paced curriculum learning scheme for segmentation by generating and selecting pseudo labels based on confidence scores. Following this, many alter works on semantic segmentation directly integrate self-training [69, 133, 59] or refine it by confidence regularization [157], self-motivated curriculum learning [70], uncertainty estimation [149], instance adaptive selection [84], prototypical pseudo label denoising [144] and domain-aware meta-learning [44].

As mentioned earlier, while these image-to-image domain adaptation methods can be applied to video segmentation by processing each frame independently [42], their performance is usually limited by ignoring the temporal information in the videos.

3.3.3 Domain adaptive video segmentation

Recently, Guan *et al.* [42] made the first attempt at video-to-video domain adaptive semantic segmentation, in which both cross-domain and intra-domain temporal consistencies are considered to regularize the learning. The former is achieved by the adversarial learning of the spatial-temporal information between the source and target domains and the latter by passing the confident part of the flow-propagated prediction between adjacent frames. Concurrently, Shin *et al.* [112] also introduced the concept of domain adaptive video segmentation and propose a two-stage solution – the adversarial learning at the clip level first, followed by the target-domain learning with the refined pseudo labels. As mentioned earlier, while our work also performs domain adaptive video segmentation, it differs from the above two works in terms of the source domain setting – they use videos but we instead use images.

Chapter 4

Style Mixing and Patchwise Prototypical Matching for One-Shot Unsupervised Domain Adaptive Semantic Segmentation

4.1 Overview

Using deep learning, state-of-the-art semantic segmentation can be obtained in the form of good prediction at each pixel by training well-designed segmentor network [10] on large-scale labeled datasets and testing on the same domain. However, constructing datasets for such dense prediction task is both very time-consuming and labor-intensive, which makes it often impossible to prepare a high-quality large-scale labeled training set for all different scenarios/domains, *i.e.*, different cities or different illumination conditions. As a result, the generalization ability of a trained model is limited, *i.e.*, it usually suffers from a drastic performance drop on an unseen testing domain due to the different data distributions from the training set.

Recently, some unsupervised domain adaptation approaches were proposed to overcome the domain discrepancy and reduce the demand of labeled data in new unseen test domains. Synthetic-to-real is a common setting in domain adaptive semantic segmentation, which was first proposed by Hoffman *et al.* [48]. In this setting, source domains with labeled synthetic data [100, 103] are constructed by using computer graphics techniques and in the meantime, sufficient number of samples in the target domain are also provided without labels. In many applications, such as medical imaging, large collection of unlabeled target data may be unavailable or difficult to obtain, which leads to the introduction of a new setting, the one-shot unsupervised domain adaptation (OSUDA) [79] for semantic segmentation. The difference of the general unsupervised domain adaptation (UDA), domain generalization (DG) and OSUDA is illustrated in Figure 4.1. In this work, we aim to tackle this challenging but practical setting of OSUDA.

Existing UDA approaches, especially those which employ discriminators to distinguish whether the content, *i.e.*, image feature [48], segmentation prediction [117] or entropy map [119], is from the source or target domains (Figure 4.1(a)), are prone to over-fitting on only one target sample – discriminators can easily distinguish the

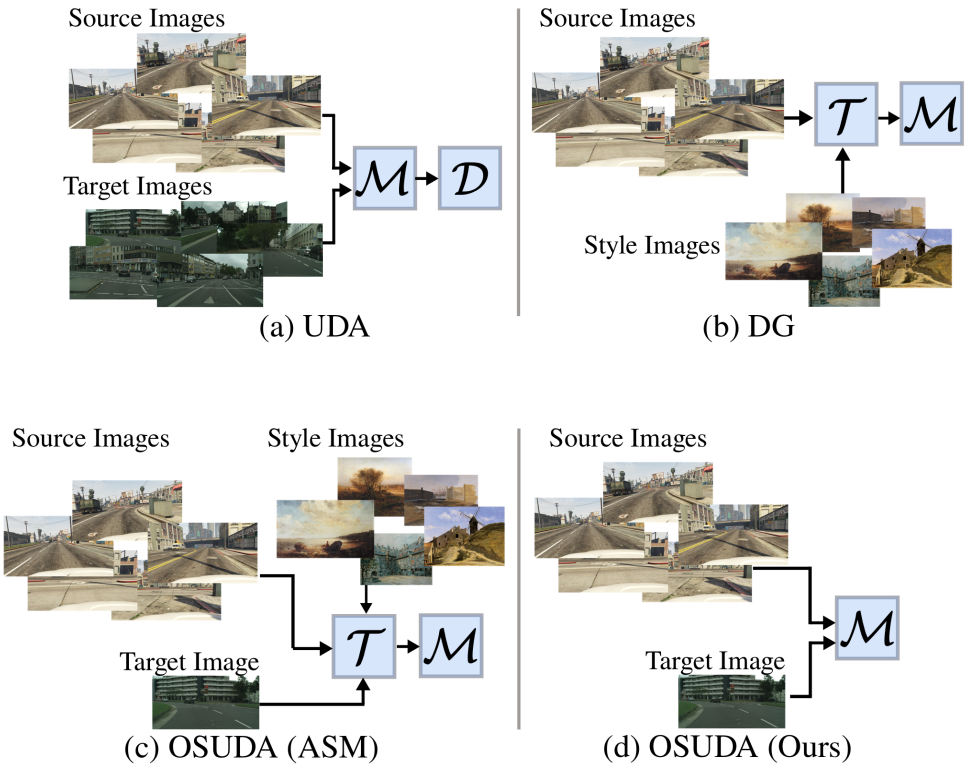


Figure 4.1 Illustration of general UDA, DG and OSUDA for semantic segmentation. The difference is mainly in the number of unlabeled target samples that are used for adaptation. Here, $\mathcal{M}$, $\mathcal{D}$ and $\mathcal{T}$ represent the segmentor, discriminator and style-transfer module, respectively.

over-fit target domain from the source domain. Other style-transfer based approaches cannot handle this one-shot setting either, since the source images can only be stylized by only one target sample. To solve this problem, Luo *et al.* proposed an adversarial style mining (ASM) algorithm [79], as illustrated in Figure 4.1(c), by mutually optimizing the style-transfer module and the semantic segmentation network via an adversarial regime. However, the style-transfer module itself requires additional data for pre-training and also increases the demand of GPU memory for adaptation.

In this work, we propose a new OSUDA approach, as illustrated in Figure 4.1(d), which does not require additional data to pre-train a style-transfer module and explicitly synthesizes stylized image for semantic segmentation. First, we design a

style-mixing segmentor which can simultaneously augment the source domain conditioned on feature statistics of the target sample and produce the semantic segmentation results. In addition, to relieve the negative adaptation [67], i.e., not all source samples/pixels have positive effect for domain adaptation, the source images are weightedly trained based on its similarity with patchwise prototypes of the sole target sample during domain adaptation.

The main contributions of this work are summarized as follows. We propose a simple and effective method for OSUDA semantic segmentation, which makes full use of the sole target image in two aspects: (1) implicitly stylizing the source domain in both image and feature levels; (2) softly selecting the source training pixels. No additional images and training parameters are introduced in the whole process. It is worth to mention that, with a pre-trained model on the source domain, our method only needs 20 minutes (500 iterations) to adapt to the target domain and obtains comparable results to the current best OSUDA approach (200k iterations without pretrain model, and additional training iterations for style-transfer model). Experimental results on two commonly-used synthetic-to-real scenarios demonstrate the effectiveness and efficiency of the proposed method.

4.2 Method

Given labeled samples $(X^{\mathtt{S}}, Y^{\mathtt{S}})$ from the source domain $\mathtt{S}$ and unlabeled samples $X^{\mathtt{T}}$ from the target domain $\mathtt{T}$, the goal of general UDA problem is to learn a mapping $\mathcal{G}$ formulated as

$$\mathcal{G}(X^{\mathtt{S}}) \rightarrow Y^{\mathtt{S}}; \;\; \mathcal{F}(\mathtt{S}) \rightarrow \mathcal{F}(\mathtt{T}), \tag{4.1}$$

where $\mathcal{F}$ is any function aligning the two domains based on either features or outputs. Different from general UDA, only one unlabeled target sample $x^{\mathtt{T}} \in X^{\mathtt{T}}$ is accessible in the OSUDA setting, which can be formulated as:

$$\mathcal{G}(X^{\mathtt{S}}|x^{\mathtt{T}}) \rightarrow Y^{\mathtt{S}}. \tag{4.2}$$

The network architecture of the proposed one-shot unsupervised domain adaptive semantic segmentation method is illustrated in Figure 4.2, which is composed of a style-mixing segmentor for both style-transfer and semantic segmentation, and a patchwise prototypical matching module for weighting the pixels of the source domain. The details of the two main components and the objective functions are discussed below.

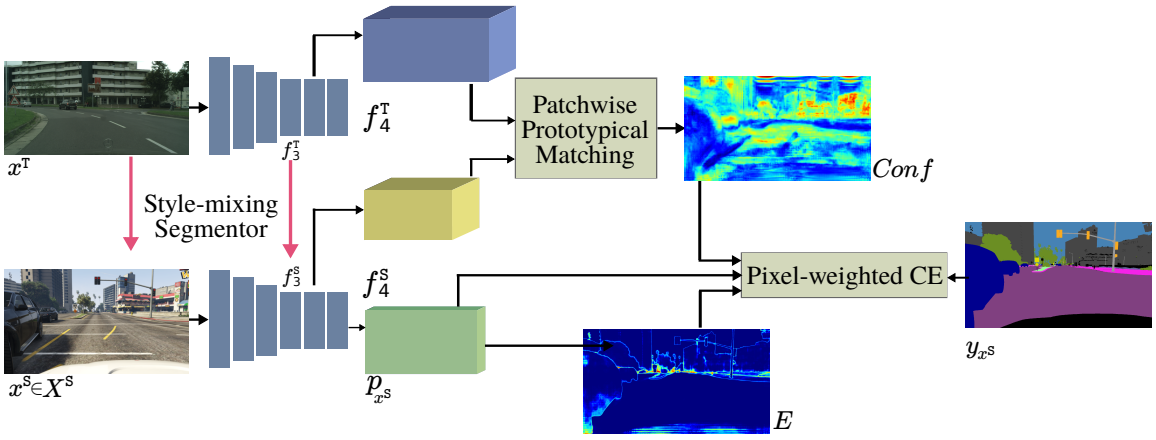


Figure 4.2 An illustration of the proposed method for OSUDA semantic segmentation. The pink arrows indicate the positions that style-mixing operation is performed.

4.2.1 Style-mixing segmentor

For each iteration, the style-mixing segmentor first takes the target sample x^{T} as the input in the evaluation mode to achieve target features using the current model parameters. Then, a sample x^{S} is randomly chosen from the source domain and fed into the segmentor in the training mode.

Inspired by [150], we propose to insert several AdaIN [53] layers into the segmentor to obtain intermediate domains by stylizing the source sample according to the target sample in both image and feature levels. Note that Luo *et al.* [79] also equipped AdaIN in their work, and the main difference is that they decouple the style-transfer

and the following semantic segmentation task and employ additional encoder and decoder to construct the stylized images, while ours combines these two tasks together. Following [53], we compute the spatial mean $\mu(.)$ and standard deviation $\sigma(.)$ of any given sample/feature $f \in \mathbb{R}^{C\times H\times W}$

$$\mu(f) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} f \tag{4.3}$$

and

$$\sigma(f) = \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(f-\mu(f))^2 + \epsilon}, \tag{4.4}$$

where C, H and W are channel, height and width of f, respectively. ϵ is set to 10^{-30}. Since the only one target sample is insufficient to describe the whole target feature distribution, we exploit more feature statistics centered around f^{T} as shown in Fig. 4.3.

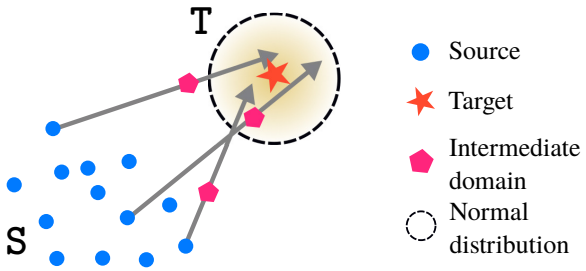


Figure 4.3 An illustration of the style-mixing operation. We first augment the target feature statistics by adding a perturbation sampled from normal distribution. Then some intermediate domains can be obtained by mixing the feature statistics of the source and the augmented target.

Then, we mix the statistics of source and target domains and calculate the intermediate channel-wise mean γ and standard deviation β by

$$\gamma = \lambda\sigma(f^{\mathrm{S}}) + (1-\lambda)\left(\sigma(f^{\mathrm{T}}) + r_\sigma\right), \tag{4.5}$$

$$\beta = \lambda\mu(f^{\mathrm{S}}) + (1-\lambda)\left(\mu(f^{\mathrm{T}}) + r_\mu\right), \tag{4.6}$$

where $\lambda \in \mathbb{R}^C$ are weights to balance the mixing operation which are randomly sampled from uniform distribution for each image/feature pair, $f^{\mathtt{S}} \in \{x^{\mathtt{S}}, f_3^{\mathtt{S}}\}$ and $f^{\mathtt{T}} \in \{x^{\mathtt{T}}, f_3^{\mathtt{T}}\}$ with $f_3^{\mathtt{S}}$ and $f_3^{\mathtt{T}}$ denoting the source and target features achieved from *layer3* respectively. Here we take $r_\sigma \sim N(0, \frac{|\sigma(f^{\mathtt{T}}) - \sigma(f^{\mathtt{S}})|}{10})$ and $r_\mu \sim N(0, \frac{|\mu(f^{\mathtt{T}}) - \mu(f^{\mathtt{S}})|}{10})$. The stylized source feature $\widehat{f^{\mathtt{S}}}$ is then produced by taking

$$\widehat{f^{\mathtt{S}}} = \gamma \left(\frac{f^{\mathtt{S}} - \mu(f^{\mathtt{S}})}{\sigma(f^{\mathtt{S}})} \right) + \beta. \tag{4.7}$$

4.2.2 Patchwise prototypical matching

In [67], Li *et al.* empirically found that some source samples could have negative effect on the adaptation. Based on this observation, they perform both image and pixel-level selections in the source domain to avoid the negative domain adaptation. However, both of their image and pixel-level selections are dependent on the distribution analysis of the target domain predictions, which are not applicable to our one-shot setting, i.e., the sole target image cannot correctly reflect the data distribution in the target domain and many categories are missing in this target sample. Inspired by [67], we propose a patchwise prototypical matching (PPM) by softly adjusting the weight of each source pixel during training according to their similarity with the target sample. We do not perform image-level selection since "negative" samples might also contain "positive" pixels for adaptation.

Specifically, we reshape the target image feature $f_4^{\mathtt{T}} \in \mathbb{R}^{C_4 \times H_4 \times W_4}$ obtained from *layer4* of the segmentor into the form of patches $p_4^{\mathtt{T}} \in \mathbb{R}^{N \times C_4 \times P^2}$ as shown in Fig. 4.4, where H_4, W_4, C_4 are the height, width and number of channels of $f_4^{\mathtt{T}}$, P is the patch size and N is the number of patches. There is no overlap between two patches.

Then, we compute the prototype for each patch via:

$$Proto_i = \frac{1}{P^2} \sum_{s=1}^{P} \sum_{t=1}^{P} Patch_i(s,t), \tag{4.8}$$

where $Proto_i \in \mathbb{R}^{C_4}$, $i \in [0, N-1]$ and (s,t) specifies each position in the patch. We compute the similarity between each prototype and the source features as a confidence

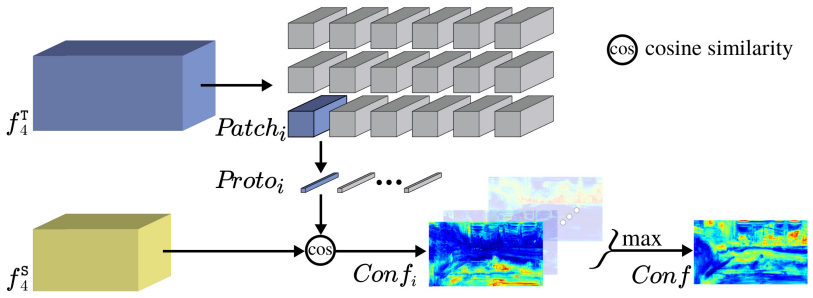


Figure 4.4 An illustration of the proposed patchwise prototypical matching.

map $Conf_i \in \mathbb{R}^{C_4 \times H_4 \times W_4}$ for adaptation by

$$Conf_i = \mathcal{F}\left(f_4^{\mathrm{S}}, Proto_i\right), \tag{4.9}$$

where $f_4^{\mathrm{S}} \in \mathbb{R}^{C_4 \times H_4 \times W_4}$ is the source image feature obtained from *layer4*, and we choose the cosine similarity as the distance function $\mathcal{F}$. We then perform a max operation across all prototypes to obtain the $Conf \in R^{H_4 \times W_4}$ for this source sample in this running iteration by

$$Conf = \max_{i \in [0, N-1]} Conf_i. \tag{4.10}$$

The reason for using the prototypical representation of the target features to compute the confidence maps include: 1) it is more efficient than pixel-wise similarity computation; 2) due to the domain gap, the pixel-level similarity usually contains much more noise which can be relieved by using patchwise prototypes. Finally, the confidence map is rectified based on the entropy of the source prediction. Given the source prediction $p_{x^{\mathrm{S}}}$, its entropy map $E \in \mathbb{R}^{H_4 \times W_4}$ can be achieved via:

$$E = -\frac{1}{\log(C)} \sum_{c=1}^{C} \left(p_{x^{\mathrm{S}}}^{(c)} \cdot \log(p_{x^{\mathrm{S}}}^{(c)}) \right), \tag{4.11}$$

where C is the number of classes. Through this way, the rectified confidence map is achieved by

$$\widehat{Conf} = Conf \cdot (1 - E). \tag{4.12}$$

High entropy indicates low confidence for the prediction, therefore, $(1-E)$ can highlight the confident region based on the prediction. The thought behind this design is that the source should be confident enough to help the adaptation to the target. A detailed pipeline for the proposed PPM is given in Algorithm 1.

Algorithm 1 Patchwise Prototypical Matching

Input: Source images $X^{\mathtt{S}}$; source labels $Y^{\mathtt{S}}$; one-shot target image $x^{\mathtt{T}}$; style-mixing segmentor $\mathcal{M}$ with the parameter θ and learning rate lr
Output: Optimal θ^*

for $x^{\mathtt{S}} \in X^{\mathtt{S}}$ **do**
  With no gradients:
    $(p_{x^{\mathtt{T}}}, f_3^{\mathtt{T}}, f_4^{\mathtt{T}}) = \mathcal{M}\left(x^{\mathtt{T}}, \text{style} = \text{None}\right)$;
    $Patch = \text{rearrange}(f_4^{\mathtt{T}})$;
    Compute $Proto$ via Eq.(4.8);
  $(p_{x^{\mathtt{S}}}, f_3^{\mathtt{S}}, f_4^{\mathtt{S}}) = \mathcal{M}\left(x^{\mathtt{S}}, \text{style} = (x^{\mathtt{T}}, f_3^{\mathtt{T}})\right)$;
  Compute the E using the $p_{x^{\mathtt{S}}}$ via Eq. (4.11);
  Compute the $\widehat{Conf}$ using $f_4^{\mathtt{S}}$ and $Patch$ via Eqs. (4.9), (4.10) and (4.12);
  Update the parameter: $\theta \leftarrow \theta - lr\nabla_\theta \mathcal{L}(p_{x^{\mathtt{S}}}, y_{x^{\mathtt{S}}}, \widehat{Conf}, E)$;
end for
Return θ as θ^*

4.2.3 Objective functions

In general, the semantic segmentation task applies the cross entropy as the loss function:

$$\mathcal{L}_{ce} = -\frac{1}{HW}\sum_{h,w}^{HW}\sum_{c=1}^{C}\Big(\ y_{x^{\mathtt{S}}}^{(h,w,c)} \cdot \log(p_{x^{\mathtt{S}}}^{(h,w,c)})\Big)\,, \tag{4.13}$$

where $y_{x^{\mathtt{S}}}^{(h,w,c)}$ represents the one-hot encoding of the ground-truth label at position (h,w) for the class c. In our approach, we employ the final confidence map $\widehat{Conf}$ to adjust the weight of each source sample in pixel-level via

$$\mathcal{L}_{pce} = -\frac{1}{HW}\sum_{h,w}^{HW}(\widehat{Conf}^{(h,w)} \cdot \sum_{c=1}^{C}(\ y_{x^{\mathtt{S}}}^{(h,w,c)} \cdot \log(p_{x^{\mathtt{S}}}^{(h,w,c)}))). \tag{4.14}$$

Finally, the whole network is trained with

$$\mathcal{L} = \alpha\mathcal{L}_{ce} + \mathcal{L}_{pce}, \tag{4.15}$$

where α is the balancing factor which is set to 0.5 in all experiments.

4.3 Experiment

4.3.1 Datasets and evaluation metric

We evaluate the proposed OSUDA semantic segmentation method in two synthetic-to-real scenarios, i.e., GTA5 [100] $\rightarrow$ Cityscapes [21] and SYNTHIA [103] $\rightarrow$ Cityscapes. Both GTA5 and SYNTHIA datasets are treated as the source domains, where the former contains 24,966 images with a resolution of $1,914 \times 1,052$ and the latter contains 9,400 images with a resolution of $1,280 \times 760$. We use Cityscapes as the target domain which is split into 2,975/500/1,525 images for training/validation/testing purpose. We follow the one-shot setting in [79] where only one unlabeled target image is used for domain adaptation. In GTA5 $\rightarrow$ Cityscapes, 19 common categories are evaluated and in SYNTHIA $\rightarrow$ Cityscapes, 16 common categories are evaluated. We apply the Intersection over Union (IoU) as the evaluation metric.

4.3.2 Implementation details

The proposed method is implemented using PyTorch trained on a single Nvidia 2080Ti GPU. We use the DeepLabV2-Res101 [10] initialized with the source-only trained weights provided by [117] as the segmentor. The source images are resized to $1,280 \times 760$ and the one-shot target sample keeps its original size.

We train the network using the SGD [3] optimizer with a momentum of 0.9 and a weight decay of 5×10^{-4}. The initial learning rate is set to 2.5×10^{-5} and it is decreased gradually following the poly learning rate policy in [117]. The batch size is set to 1 and the whole network is trained for 500 iterations. Note that we even don't get access to all source images during domain adaptation. Images from Cityscapes validation set are resized to $1,024 \times 512$ for performance evaluation. We run each OSUDA experiment with the same 5 images as [79] (one for each time) and 5 times for

each image. Finally, we report the average mIoU of the 25 runs computed using the model weights saved in the last running iteration. All the approaches are evaluated on the Cityscapes-val set.

4.3.3 Comparison results

In Table 4.1, we present the comparison results with other state-of-the-art approaches for the GTA5 → Cityscapes experiment. The compared method can be divided into three camps based on the data except for the source domain that are needed for adaptation: 1) one unlabeled target image only (denoted by O); 2) style image dataset (denoted by S); (3) both 1) and 2) (denoted by O+S). It can be observed that our method achieves the best performance in the first camp.

Obviously, the general UDA approaches [117, 81, 158, 158] are not working well in the one-shot setting and some even get worse results than the source only. Methods [152, 2] are proved to be more robust to this setting which indicates usefulness of the style transfer strategy. ASM [79] is the first method that tackle the OSUDA which is the most related one to ours. To make a fair comparison, we reproduce the results of ASM using the same backbone as us and the reported mIoU is also based on the model saved in the last iteration (not selecting the best one). Especially, our method does not need additional dataset to pre-train a style transfer model while ASM needs and runs only for 500 iterations for domain adaptation to achieve this comparable results.

We also find that domain generalization approaches [139, 52] using additional style image dataset also achieve comparable results or even better than the methods using one target image. This indicates that using more images than only one target image can be more helpful as expected. However, they need to spend more time to explore the desired domains and the style references also need to be properly chosen.

The results for SYNTHIA → Cityscapes experiment are reported in Table. 4.2,

Table 4.1 Quantitative comparison results for domain adaptation from GTA5 to Cityscapes. The per-category mIoU (%) of the Cityscapes-val set are reported. For all method with one-shot only setting denoted by O, the best results are presented in **bold**, with the second best results underlined.

Method	Extra Data	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain
Source only	-	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6
Adaptseg	O	77.7	19.2	75.5	11.7	6.4	16.8	18.2	15.4	77.1	34.0
CLAN	O	77.1	22.7	78.6	17.0	14.8	20.5	23.8	12.0	80.2	39.5
ADVENT	O	76.1	15.1	76.6	14.4	10.8	17.5	19.8	12.0	79.2	39.5
CBST	O	76.1	22.2	73.5	13.8	18.8	19.1	20.7	18.6	79.5	41.3
CycleGAN	O	80.3	23.8	76.7	17.3	18.2	18.1	21.3	17.5	81.5	40.1
OST	O	84.3	**27.6**	**80.9**	**24.1**	23.4	26.7	23.2	19.4	80.2	**42.0**
Ours	O	**85.0**	23.2	80.4	21.3	**24.5**	**30.0**	**32.0**	**26.7**	**83.2**	34.8
DRPC	S	-	-	-	-	-	-	-	-	-	-
FSDR	S	89.3	40.5	79.1	26.3	27.8	29.3	33.7	29.0	83.0	27.7
ASM	O+S	89.5	31.2	81.3	27.8	22.8	30.6	32.8	25.1	82.6	35.0

Method	Extra Data	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	**mIoU**
Source only	-	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6
Adaptseg	O	68.5	55.3	30.9	74.5	23.7	28.3	2.9	14.4	18.9	35.2
CLAN	O	74.3	56.6	25.2	78.1	29.3	31.2	0.0	19.4	16.7	37.7
ADVENT	O	71.3	55.7	25.2	76.7	28.3	30.5	0.0	23.6	14.4	36.1
CBST	O	74.8	57.4	19.9	78.7	21.3	28.5	0.0	28.0	13.2	37.1
CycleGAN	O	74.0	56.2	**38.3**	77.1	30.3	27.6	1.7	**30.0**	22.2	39.6
OST	O	**80.7**	**59.2**	20.3	**84.1**	**35.1**	**39.6**	1.0	29.1	23.2	42.3
Ours	O	74.0	57.3	29.0	77.7	27.3	36.5	**5.0**	28.2	**39.4**	**42.8**
DRPC	S	-	-	-	-	-	-	-	-	-	42.5
FSDR	S	76.0	57.8	27.5	81.0	32.3	42.4	16.8	21.0	30.2	44.8
ASM	O+S	76.7	59.2	26.6	82.3	27.7	34.1	0.9	25.6	29.6	43.2

where our method achieve the best performance across all of the three settings and surpass the second best by 4.5% mIoU in the one-shot only setting.

We also show qualitative results for GTA5 $\rightarrow$ Cityscapes and SYNTHIA $\rightarrow$

Table 4.2 Quantitative comparison results for domain adaptation from SYNTHIA to Cityscapes. The per-category mIoU (%) (13 categories) and mIoU* (%) (16 categories) of Cityscapes-val set are reported. For all method with one-shot only setting denoted by O, the best results are presented in **bold**, with the second best results underlined.

Method	Extra Data	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation
Source only	-	55.6	23.8	74.6	-	-	-	6.1	12.1	74.8
Adaptseg	O	64.1	25.6	75.3	-	-	-	4.7	2.7	77.0
CLAN	O	68.3	26.9	72.2	-	-	-	5.1	5.3	75.9
ADVENT	O	65.7	22.3	69.2	-	-	-	2.9	3.3	76.9
CBST	O	59.6	24.1	72.9	-	-	-	5.5	13.8	72.2
OST	O	75.3	31.6	72.1	-	-	-	12.3	9.3	76.1
Ours	O	**79.3**	**35.3**	**75.9**	5.6	16.6	29.83	**25.4**	**22.7**	**79.9**
DRPC	S	-	-	-	-	-	-	-	-	-
FSDR	S	69.3	34.9	77.6	7.9	0.2	29.4	16.3	19.2	72.3
ASM	O+S	85.7	39.7	77.1	1.1	0.0	24.2	2.1	9.2	76.9

Method	Extra Data	sky	person	rider	car	bus	motorcycle	bicycle	**mIoU**	**mIoU***
Source only	-	79.0	55.3	19.1	39.6	23.3	13.7	25.0	38.6	-
Adaptseg	O	70.0	52.2	20.6	51.3	22.4	19.9	22.3	39.1	-
CLAN	O	71.4	54.8	18.4	65.3	19.2	22.1	20.7	40.4	-
ADVENT	O	69.2	**55.4**	21.4	**77.3**	17.4	21.4	16.7	39.9	-
CBST	O	69.8	55.3	21.1	57.1	17.4	13.8	18.5	38.5	-
OST	O	71.1	51.1	17.7	68.9	19.0	**26.3**	25.4	42.8	-
Ours	O	**76.8**	54.6	**23.5**	60.2	**23.9**	21.2	**36.6**	**47.3**	**41.4**
DRPC	S	-	-	-	-	-	-	-	-	37.6
FSDR	S	76.3	56.7	22.1	80.6	41.5	19.1	29.3	47.3	40.8
ASM	O+S	81.7	43.4	11.4	63.9	15.8	1.6	20.3	40.7	34.6

Cityscapes each on 5 samples from the Cityscapes-val set in Figure 4.5 4.6 and Figure 4.74.8, respectively. It can be observed that our method achieves comparable visualization results as ASM in the two domain adaptation scenarios and even better on some categories such as train (Figure 4.7(a)), rider and bicycle (Figure 4.5(c)) and

truck (Figure 4.6(d)).

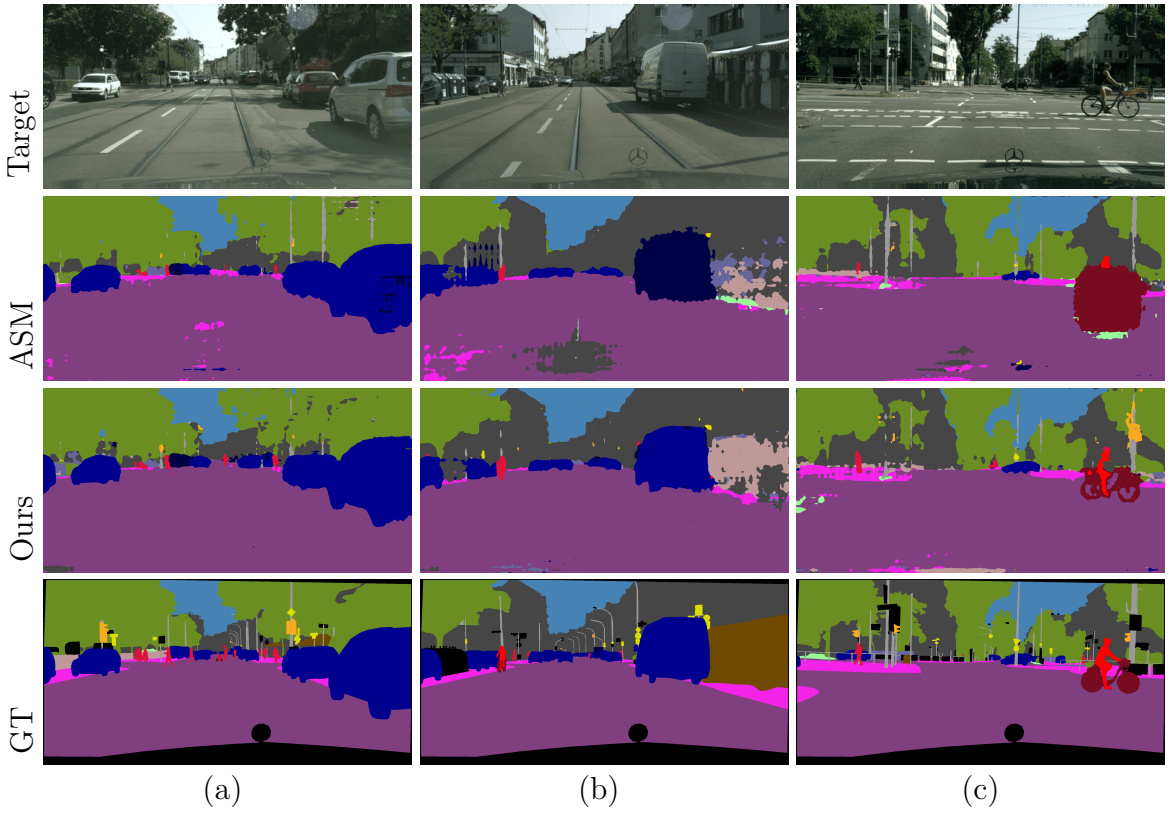


Figure 4.5 Some qualitative comparison results for domain adaptation from GTA5 → Cityscapes.

4.3.4 Ablation studies

Variants of the loss functions. We first investigate several variants of Eq. (4.12) as shown in Table. 4.3. We directly use the model weights provided by [117] and fine-tune it with the source data using the standard cross-entropy loss to obtain the source-only results. Note that all these variants are equipped with the original segmentor instead of the style-mixing one. We can observe that the confidence obtained via PPM is the most important component in this equation, without which the mIoU drops 2.66% on GTA5 → Cityscapes and 9.66% on SYNTHIA → Cityscapes. Compared with $\widehat{Conf}$, E has inconsistent effect on the two experiments.

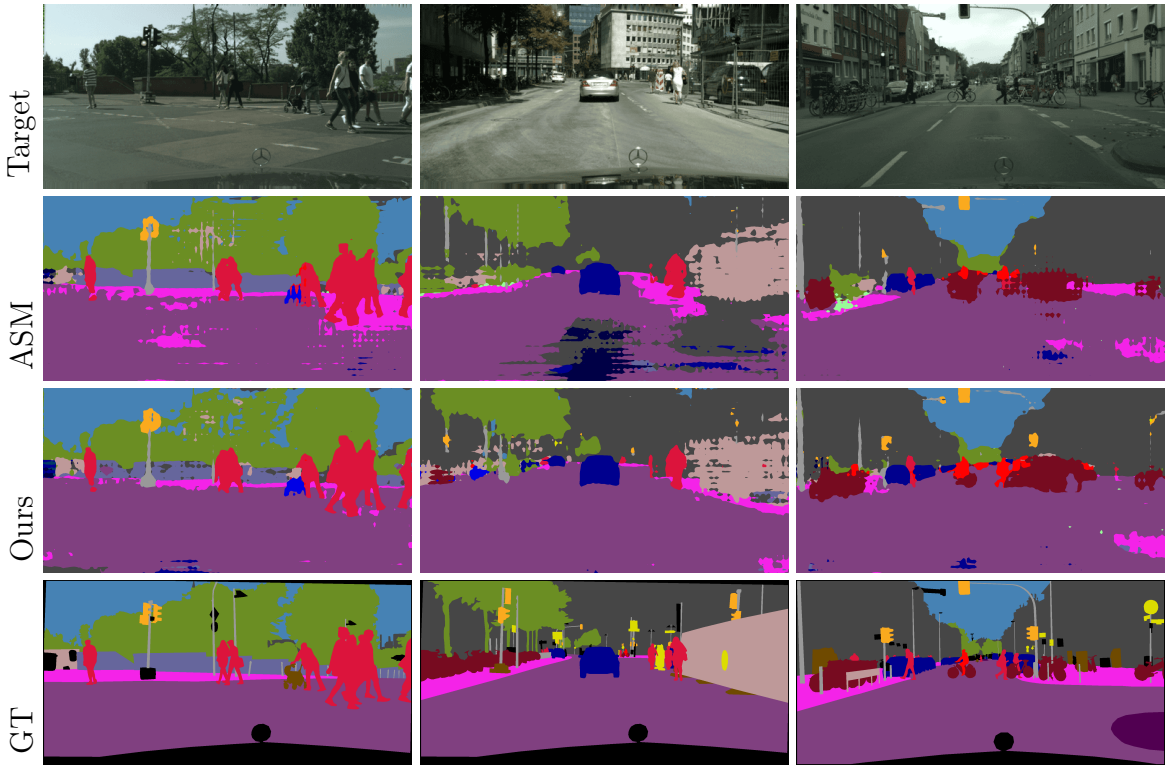


Figure 4.6 Some qualitative comparison results for domain adaptation from GTA5 → Cityscapes.

Table 4.3 Variants of Eq. (4.12) in both GTA5 → Cityscapes and SYNTHIA → Cityscapes scenarios. The mIoU (%) scores are reported.

Variants	Source-only	w/o $\widehat{Conf}$	w/o E	Eq. (4.12)
G → C	36.67	38.38	40.95	**41.04**
S → C	35.26	34.26	**44.14**	43.92

Variants of the style-mixing segmentor. We study several variants of the style-mixing segmentor as shown in Table. 4.4. The original Deeplab-V2-Res101 without style-mixing (with PPM) serves as the baseline which can obtain 41.04 mIoU. Applying the style-mixing layer in image-level (x^{s} only) can obtain 1.39% for GTA5 → Cityscapes and 3.04% for SYNTHIA → Cityscapes. In addition, we try different

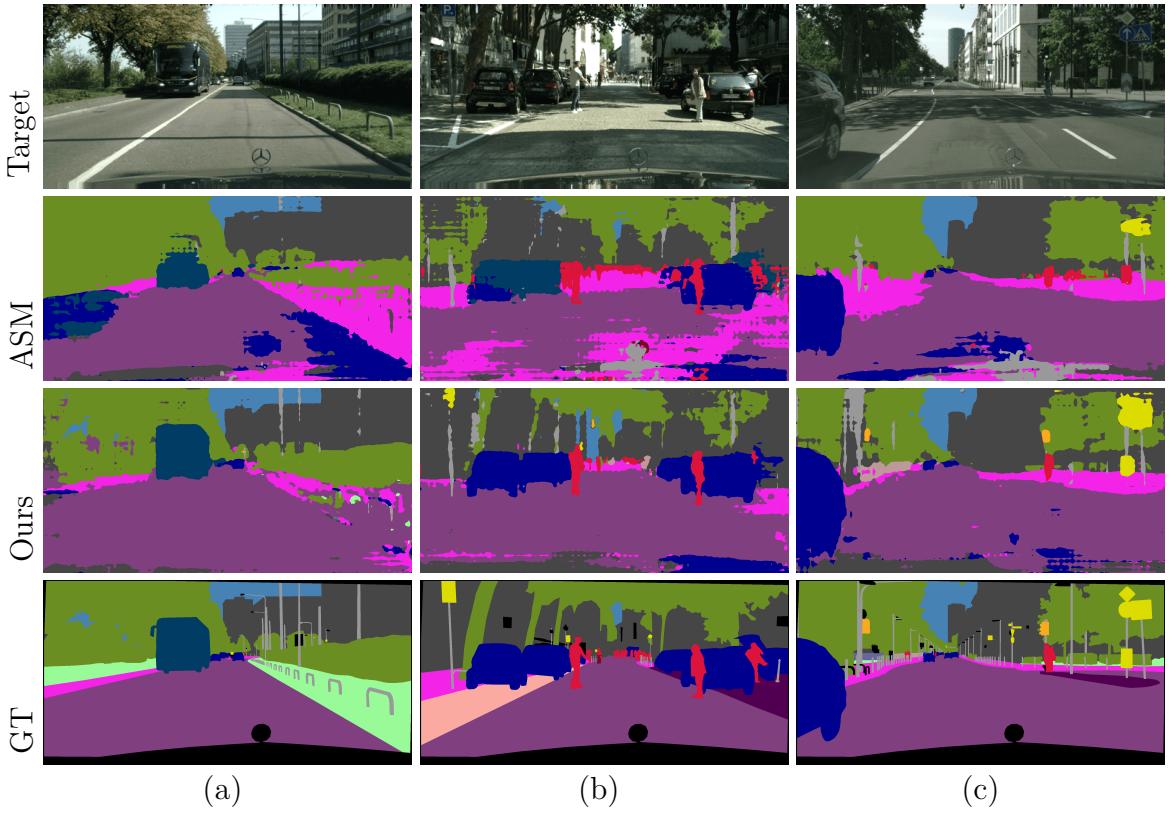


Figure 4.7 Some qualitative comparison results for domain adaptation from SYNTHIA → Cityscapes.

feature-level style-mixing and we observe that using f_3^{S} only is better than using other levels across the two adaptation settings. Therefore, we choose to apply the style-mixing layer to both x^{S} and f_3^{S} (Ours). Other combinations might result in similar performance. Compared with the original version of AdaIN, our modified version achieves better performance.

Table 4.4 Variants of the style-mixing segmentor in both GTA5 → Cityscapes and SYNTHIA → Cityscapes scenarios. The mIoU (%) scores are reported.

Variants	baseline	x^{S}	f_1^{S}	f_2^{S}	f_3^{S}	f_4^{S}	AdaIN	Ours
G → C	41.04	42.43	40.74	41.13	41.16	40.24	41.98	**42.77**
S → C	43.92	46.96	43.65	43.71	44.63	43.92	46.82	**47.33**

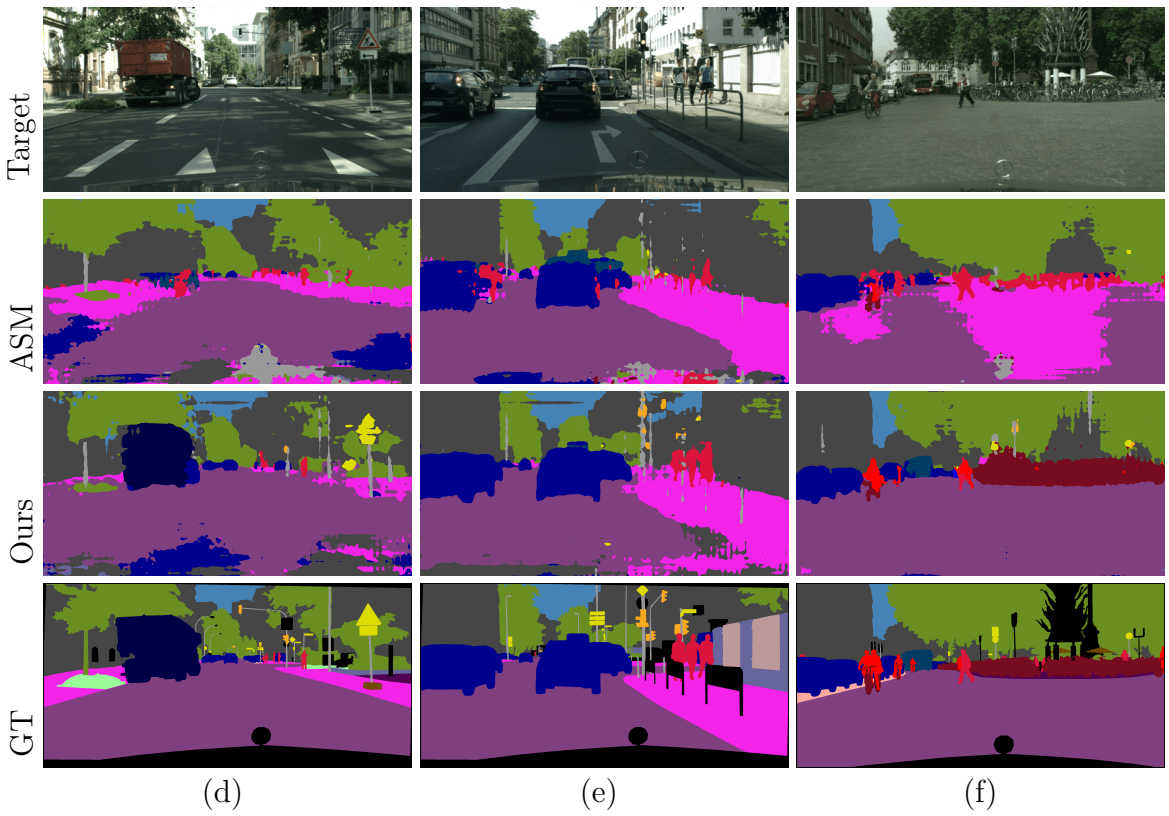


Figure 4.8 Some qualitative comparison results for domain adaptation from SYNTHIA → Cityscapes.

Variants of the patch size. We also study different choices of patch size as shown in Table. 4.5, where "no patch" means that we don't split the target image into patches and use prototype of the whole image to calculate the confidence map. We find that the patch size 32 performs the best size for both two domain adaptation experiments.

Table 4.5 Variants of the patch size for both GTA5 → Cityscapes and SYNTHIA → Cityscapes scenarios. The mIoU (%) scores are reported.

Patch size	8	16	32	64	no patch
G → C	42.43	42.30	**42.77**	42.38	42.26
S → C	44.83	45.91	**47.33**	45.52	46.03

Ablation Study on the pre-training model We study the effect of the usage of the pre-training model. From Figure 4.9, we find that our method can still obtain similar mIoU results without using the pre-training model for GTA5 → Cityscapes with more training iterations. Compared with ASM, our method does not need additional dataset and time to train a style-transfer model and uses fewer adaptation iterations with a pre-trained source-only model. And our method can save the memory usage, for example, it only needs about 10G GPU memory while ASM requires around 25G.

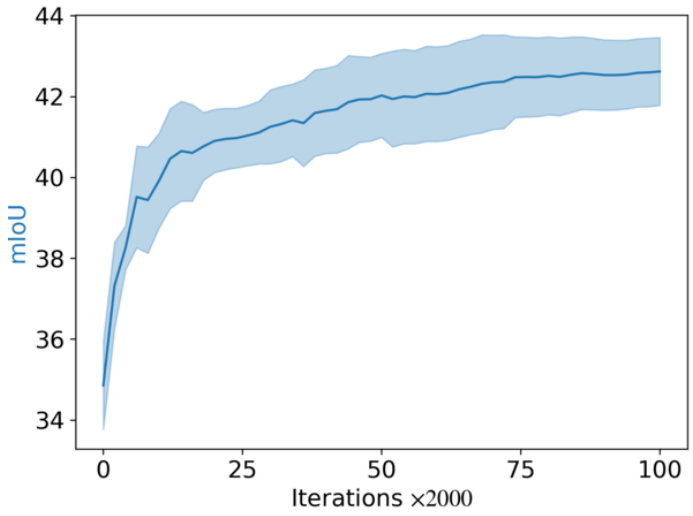


Figure 4.9 The mIoU (%) performance over varying adaptation iterations without using pertrained model for GTA5 → Cityscapes.

4.4 One-Shot Day-to-Night Domain Adaptation

We further evaluate the proposed method on the more challenging day-to-night setting. In this experiment, we pick Cityscapes [21] as the source and the Dark Zurich [106] as the target.

The Dark Zurich dataset is carefully collected by Sakaridis *et al.* for unsupervised nighttime semantic segmentation. It consists of 3,041 daytime, 2,920 twilight and 2,416 nighttime images that are all unlabeled and can be used for domain adaptation.

There are also 201 labeled nighttime images including 50 images for validation whose labels are provided and the rests serve as an online benchmark. The resolution of all images is 1,920 $\times$ 1,080.

In our experiments, only one nighttime image is used for domain adaptation and the Dark Zurich validation set is used for performance evaluation. We run this experiment with 4 images (one for each experiment) and 5 times for each image. Finally, we report the average mIoU of the 20 runs computed using the model weights saved in the last running iteration in Table 4.6. Here, the source-only model is obtained by training the DeepLabV2-Res101 [10] on the Cityscapes training set for 150K iterations. By applying the source-only model weights, our method can achieve 17.5% with additional 500 training iterations. We run ASM with the same 4 images for 50K iterations and compute the average of the 4 experiments as their results. It can be observed that both of the two OSUDA approaches obtain performance gains over the source-only results and our method get better results without training an explicit style transfer model with additional dataset. Some qualitative results are shown in Figure 4.10 where we can see that our method get better visualization results.

4.5 Chapter Summary

In this work, we have developed a novel method for the challenging one-shot setting in unsupervised domain adaptation. By inserting the new version of AdaIN layers into the segmentor, our method has the ability to explore more styles around the target sample and perform the semantic segmentation at the same time. We find that the implicit style-transfer based on feature-level statistics can significantly reduce the memory usage and improve the efficiency of domain adaptation. In addition, patch-wise prototypical matching, which is proposed for relieving the negative adaptation and weighting more the positive adaptation, is also shown to be very effective for this task. Various experiments demonstrate that our method can achieve better or

Table 4.6 Quantitative comparison results for domain adaptation from Cityscapes to Dark Zurich. The per-category mIoU (%) of the Dark Zurich validation set are reported.

Method	Extra Data	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain
Source only	-	69.5	12.4	44.9	3.0	18.5	17.8	15.6	7.0	34.2	7.4
ASM [79]	O+S	75.2	31.7	38.6	7.7	17.6	16.9	12.6	4.9	24.4	8.0
Ours	O	63.3	16.6	46.5	5.1	22.9	9.0	15.5	7.7	39.7	10.1

Method	Extra Data	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	**mIoU**
Source only	-	8.4	12.3	0.8	22.8	0.0	0.0	0.0	3.4	0.2	14.6
ASM [79]	O+S	9.8	16.7	1.4	42.9	0.0	0.0	0.0	7.7	2.1	16.7
Ours	O	31.5	10.2	0.7	38.7	0.0	0.0	0.0	11.0	4.2	17.5

comparable results to the current state-of-the-arts in the one-shot setting with much fewer iterations.

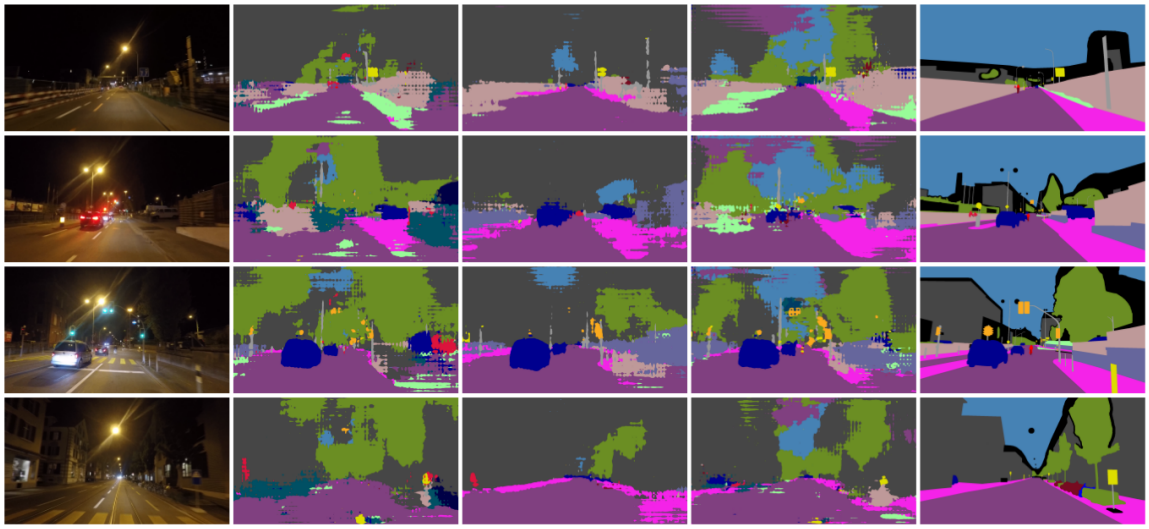


Figure 4.10 Some qualitative comparison results for domain adaptation from Cityscapes to Dark-Zurich.

Chapter 5

A One-Stage Domain Adaptation Network for Unsupervised Nighttime Semantic Segmentation

5.1 Overview

With the advancement of deep learning and computing power, the state-of-the-art performance of semantic segmentation for natural scene images taken at the daytime has been significantly improved in recent years [34, 54]. Many researchers have started to segment more challenging images under various kinds of degradations, such as those taken in foggy weather [109] or at the nighttime [106]. In this work, we focus on semantic segmentation of nighttime images, which has wide and important applications in autonomous driving.

With many indiscernible regions and visual hazards [142], *e.g.*, under/over exposure and motion blur, it is usually difficult even for human to build high-quality pixel-level annotations of the nighttime scene images as ground truth, which, however, is a prerequisite for training many deep neural networks for semantic image segmentation. To handle this problem, several domain adaptation methods have been proposed to transfer the semantic segmentation models from daytime to nighttime without using labels in the nighttime domain. For example, in [23, 106, 108], an intermediate twilight domain is taken as a bridge to build the adaptation between daytime to nighttime. In [106, 101, 114, 88, 108], an image transferring network is trained to stylize nighttime or daytime images and construct synthetic datasets. All these methods require an additional pre-processing stage of training an image transfer model between daytime and nighttime. This is not only time-consuming but also making the second stage closely rely on the first one. Especially, it is difficult to generate a transferred image that shares the exactly same semantic information with the original images when the domain gap is large.

In this work, we propose a novel one-stage domain adaptation network (DANNet) based on adversarial learning for nighttime semantic segmentation by using the newly released Dark Zurich dataset [106], which contains unlabeled day-night scene image pairs that are coarsely aligned using GPS recordings. The proposed DANNet performs

a multi-target adaptation from Cityscapes data to Dark Zurich daytime (Dark Zurich-D) and nighttime data (Dark Zurich-N). Specifically, we first adapt the model from Cityscapes, which contains large-scale training data with labels, to Dark Zurich-D since they are all taken at the daytime. Then, the prediction of Dark Zurich-D is used as a pseudo supervision for Dark Zurich-N in the network training. We apply an image relighting subnetwork to make the intensity distribution of the images from different domains to be close. Following [117], we incorporate a weight-sharing semantic segmentation network to make predictions for the relighted images and perform an adversarial learning in the output space to ensure very close layout across different domains. We further design a re-weighting strategy to handle the inaccuracy caused by misalignment between day-night image pairs and wrong predictions of daytime images, as well as boost the prediction accuracy of small objects. We conduct extensive experiments on Dark Zurich and Nighttime Driving datasets to justify the effectiveness of the proposed DANNet for nighttime semantic segmentation.

5.2 Method

5.2.1 Framework overview

Our method involves a source domain S and two target domains T_d and T_n, where S, T_d, and T_n represent Cityscapes (daytime), Dark Zurich-D (daytime), and Dark Zurich-N (nighttime), respectively. Note that only the source domain S of Cityscapes has ground-truth semantic segmentation in training. The proposed DANNet proceeds the domain adaptation from S to T_d and S to T_n simultaneously and it consists of three different modules: an image relighting network, a semantic segmentation network, and two discriminators, as illustrated in Figure 5.1. Three input images I_s, I_{td}, and I_{tn} are from the source domain S (Cityscapes) and two target domains T_d and T_n (Dark Zurich-D and Dark Zurich-N), respectively. They go through a weight-sharing image relighting network which can make their distributions to be close to each other

using the light loss L_{light}. All the outputs are fed into a weight-sharing segmentation network to obtain the predictions. For the predictions from I_s, a semantic segmentation loss L_{seg} is computed using the ground truth from the source dataset. Besides, the predictions from I_{td} for the categories of static objects provide weak supervision for the corresponding categories from I_{tn}, reflected by a static loss L_{static}. Note that the composition of the relighting network and the semantic segmentation network forms the generator G. Two discriminators D_d and D_n are proposed to distinguish outputs from the source domain S or the target domains T_d and from the source domain S or the target domains T_n, respectively. All modules of the proposed DANNet are elaborated in detail below.

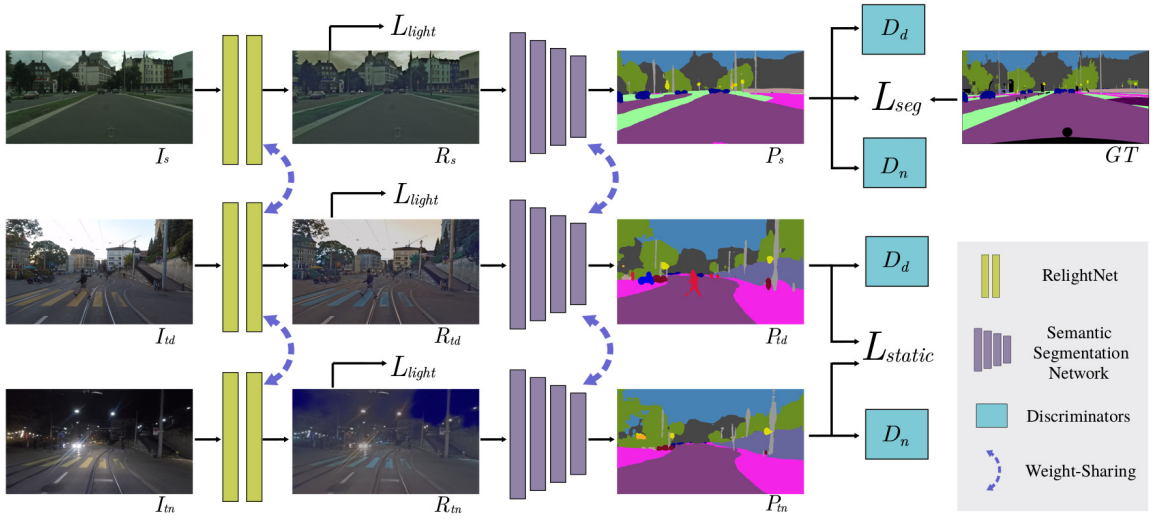


Figure 5.1 The architecture of the proposed DANNet.

Image relighting network Inspired by [56], we design an image relighting network to make the intensity distributions of the images from different domains to be close such that the later semantic segmentation network is less sensitive to illumination changes. The relighting network takes the scene images I_s, I_{td} and I_{tn} from the three domains, and generates the relighted images R_s, R_{td} and R_{tn}, respectively. The relighting network shares weights for all input images from the three domains, see

Figure 5.2 for the detailed structure of this network. It consists of four convolutional layers, three residual blocks and two transposed convolutional layers, and each convolutional layer is followed by a batch normalization layer. The output from the last layer is then added to the input images to obtain the relighted image.

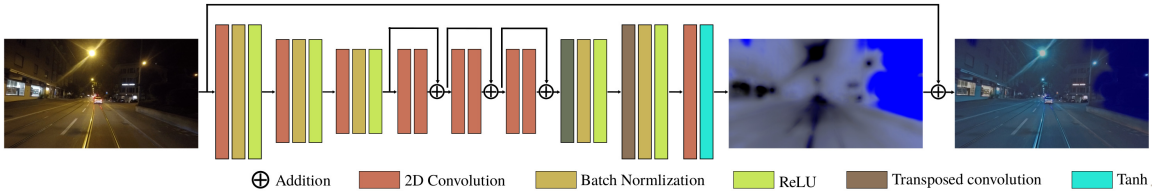


Figure 5.2 The structure of the image relighting network.

Semantic segmentation network We select and test three popular semantic segmentation networks in our method: Deeplab-v2 [10], RefineNet [71] and PSP-Net [148]. Note that the common backbone is ResNet-101 [45] in all of them. For this module, we share weights for all the input images from the three domains. The semantic segmentation network takes R_s, R_{td} and R_{tn} as the inputs and produces segmentation predictions (category-likelihood map) P_s, P_{td} and P_{tn} for the three domains, respectively. The composition of the image relighting network and the semantic segmentation network forms the generator G of the proposed DANNet.

Discriminators As done in [117], the discriminators are designed to distinguish whether the segmentation prediction comes from the source domain or either of the target domains by performing adversarial learning in the output space. We modified the architecture in [98] following [117] by utilizing all fully convolutional layers. Particularly, it includes 5 convolutional layers with the channel numbers of $\{64, 128, 256, 256, 1\}$, and a kernel size of 4×4. The stride is 2 for the first two convolutional layers and 1 for the rest. Since we have two target domains T_d and T_n, we design two discriminators D_d and D_n to distinguish whether the output is from S or T_d and from S or T_n, respectively. The two discriminators share the same structures

yet the weights and are jointly trained.

5.2.2 Probability re-weighting

Due to the fact that the numbers of pixels for different object categories are imbalanced in the source domain, network training can usually converge more easily by predicting a pixel to be a category of large-size object, such as road, building, and tree, in training discriminators. In this case, it is quite difficult to correctly predict the pixels of small objects which have relatively fewer annotations in the dataset, such as pole, sign, and light. To address this problem, we propose a re-weighting strategy to the predicted category-likelihood maps. Specifically, for each category $k \in \mathbb{C}$, we first define a weight

$$w'_k = -\log(a_k), \tag{5.1}$$

where a_k is the proportion of all the valid pixels that are labeled as category k in the source domain. Clearly the smaller the value of a_k, the larger the value of w'_k and the use of such a weight can help segment the categories of smaller-size objects. We use the logarithm to prevent from overweighting small-size object categories. In our experiment, we further normalize this weight by

$$w_k = \frac{w'_k - \overline{w}}{\sigma(w)} \cdot std + avg, \tag{5.2}$$

where $\overline{w}$ and $\sigma(w)$ are the mean and standard deviation of $w'_k, k \in \mathbb{C}$, respectively. The parameters std and avg are two positive constants we pre-select to shift the value range of w_k to be mainly positive. During training, we set $std = 0.05$ and $avg = 1.0$ empirically. We then multiply each normalized weight w_k with the corresponding category channel of the predicted likelihood map P, where $P \in \{P_{td}, P_{tn}\}$. Thus, the final semantic segmentation result F is obtained by employing an argmax operation on the multiplication result.

5.2.3 Objective functions

In this subsection, we introduce all the objective functions involved in the proposed end-to-end DANNet training, including the light loss, the semantic segmentation loss, the static loss, and the adversarial loss.

Light Loss The light loss is proposed to ensure that the intensity distributions of the outputs R_s, R_{td} and R_{tn} after the image relighting network are close to each other. The light loss is a combination of three loss functions: the total variation loss L_{tv}, the exposure control loss L_{exp}, and the structural similarity loss L_{ssim}.

The total variation loss L_{tv} [104] is widely used in image denoising [143] and image synthesis [122] to make images smoother. In this work, we apply such a loss function to remove rough textures such as noises to facilitate the semantic segmentation. The loss L_{tv} is defined by

$$L_{tv} = \frac{1}{N}\|(\nabla_x(I-R))^2 + (\nabla_y(I-R))^2\|_1, \tag{5.3}$$

where $I \in \{I_s, I_{td}, I_{tn}\}$ represents the input images, $R \in \{R_s, R_{td}, R_{tn}\}$ is the output of the relighting network, N is the number of pixels in I, ∇_x and ∇_y represent intensity gradients between neighboring pixels along the x and y directions, respectively, and $\|\cdot\|_1$ is the L_1 norm that sums up over all the pixels.

To obtain the similar lighting effects in the day and night scenarios, we apply the following exposure loss L_{exp} proposed in [43] to control the exposure level:

$$L_{exp} = \frac{1}{M}\|\varphi(R) - E\|_1, \tag{5.4}$$

where φ is a 32×32 average pooling function and M represents the number of pixels in $\varphi(R)$. Different from [43], the value of E is dynamically set to be the average intensity value of the nighttime image for each training iteration.

The structural similarity loss L_{ssim} [124] is widely used for image reconstruction [38, 8]. Here we apply this loss function to ensure that the generated relighted

images R could maintain the structure of original images I. The loss L_{ssim} is defined by

$$L_{ssim} = \frac{1}{2N}\|1 - SSIM(I, R)\|_1. \tag{5.5}$$

As in [38], we use a simplified SSIM (structural similarity index measure) with a 3×3 block filter in this loss function.

Finally, by combining all the three loss terms, our light loss L_{light} is defined by

$$L_{light} = \alpha_{tv} L_{tv} + \alpha_{exp} L_{exp} + \alpha_{ssim} L_{ssim}, \tag{5.6}$$

where $\alpha_{tv}, \alpha_{exp}$, and α_{ssim} are set to 10, 1, and 1, respectively in all experiments.

Semantic segmentation loss We adopt the widely used weighted cross-entropy loss for training the semantic image segmentation in the source domain:

$$L_{seg} = -\frac{1}{N|\mathbb{C}|} \sum_{k \in \mathbb{C}} \|w_k GT^{(k)} \cdot log(P_s^{(k)})\|_1, \tag{5.7}$$

where $P_s^{(k)}$ is the k-th channel of the prediction P_s from the source images, w_k is the weight defined in Eq. (5.2), and $GT^{(k)}$ is the one-hot encoding of the ground truth for the k-th category.

Static loss Based on the fact that the daytime image share similarities with its corresponding nighttime counterpart when considering only the static object categories, we here introduce a static loss to provide pixel-level pseudo supervision for the static object categories, *e.g.*, road, sidewalk, wall, fence, pole, light, sign, vegetation, terrain and sky, in the nighttime images.

Given the segmentation predictions $P_{td} \in \mathbb{R}^{H \times W \times C}$ and $P_{tn} \in \mathbb{R}^{H \times W \times C}$, we only consider the channels corresponding to the static categories for calculating this loss. Let us denote $C^{\mathcal{S}}$ as the total number of the categories of static objects, then it holds that $P_{td}^{\mathcal{S}} \in \mathbb{R}^{H \times W \times C^{\mathcal{S}}}$ and $P_{tn}^{\mathcal{S}} \in \mathbb{R}^{H \times W \times C^{\mathcal{S}}}$.

We first apply Eq. (5.2) to calculate the re-weighted prediction F_{td} as the pseudo label. Following [135, 15], we then employ the focal loss [72] to remedy the imbalance

among different categories of training samples. Finally, the static loss L_{static} is defined by

$$L_{static} = -\frac{1}{N}\|(1 - P_{tn}^{\mathcal{S}})^{\gamma} log(p)\|_1, \tag{5.8}$$

where N is the total number of valid pixels in the segmentation ground truth, γ is the focusing parameter (set to 1 in all experiments), and p is the likelihood map for the correct category. Different from the focal loss in [72], we compute p at each pixel i in a 3×3 local region for category c by

$$p(c, i) = \max_j (o(c, j) \cdot P_{tn}^{\mathcal{S}}(c, i)), \tag{5.9}$$

where o is the one-hot encoding of the semantic pseudo ground truth F_{td}, and j represents each position of the 3×3 region centered at i.

Adversarial loss We employ two discriminators for adversarial learning, which are used to distinguish whether the output is from the source domain or one of the two target domains, *i.e.*, S or T_d and S or T_n. We adopt the least-squares loss function [83] to make both predictions P_{td} and P_{tn} to be close to P_s. Specifically, we define the combination of these two adversarial losses (L_{adv}) as:

$$L_{adv} = (D_d(P_{td}) - r)^2 + (D_n(P_{tn}) - r)^2, \tag{5.10}$$

where $P_{td} = G(I_{td})$, $P_{tn} = G(I_{tn})$, and r is the label for the source domain which has the same resolution as the output of discriminators. Thus, the total loss L_{total} of the generator (G) is defined by combining L_{light}, L_{seg}, L_{static} and L_{adv}:

$$\min_G L_{total} = \beta_1 L_{light} + \beta_2 L_{seg} + \beta_3 L_{static} + \beta_4 L_{adv}, \tag{5.11}$$

where $\beta_1, \beta_2, \beta_3$, and β_4 are set to 0.01, 1, 1 and 0.01 respectively in all experiments.

The generator and the corresponding discriminators are trained alternatively and the objective functions of the discriminators D_s and D_n are defined respectively by:

$$\min_{D_d} L_d = \frac{1}{2}(D_d(P_s) - r)^2 + \frac{1}{2}(D_d(P_{td}) - f)^2, \tag{5.12}$$

$$\min_{D_n} L_n = \frac{1}{2}(D_n(P_s) - r)^2 + \frac{1}{2}(D_n(P_{tn}) - f)^2, \tag{5.13}$$

where f is the label for the target domains with the same resolution as the output of discriminators.

5.3 Experiment

5.3.1 Datasets and evaluation metrics

For all experiments, we use the mean of category-wise intersection-over-union (mIoU) as the evaluation metric, and the higher the better. The following datasets are used for model training and performance evaluation:

Cityscapes [21] The Cityscapes dataset contains 5,000 frames taken in street scenes with pixel-level annotations of a total of 19 categories, and both the original images and annotations have a resolution of $2,048 \times 1,024$ pixels. In total, there are 2,975 images for training, 500 images for validation and 1,525 images for testing. In this work, we use the Cityscapes training set in the training stage of the proposed DANNet for adversarial learning.

Dark Zurich [106] The Dark Zurich dataset consists of 2,416 nighttime images, 2,920 twilight images and 3,041 daytime images for training, which are all unlabeled with a resolution of $1,920 \times 1,080$. Images in these three domains can be coarsely aligned by using GPS-based nearest neighbor assignment to compensate the translation in each direction and the zoom in/out factors. In this work, we only use 2,416 night-day image pairs in training of the proposed DANNet (without using the twilight images). The Dark Zurich dataset also contains another 201 annotated nighttime images including 50 for validation (Dark Zurich-val) and 151 for testing (Dark Zurich-test), for quantitative evaluation. Note that the Dark Zurich-test serves as an online benchmark whose ground truth are not publicly available. In our experiments, by submitting the segmentation results to the online evaluation website we get the

performance of the proposed DANNet on Dark Zurich-test against the annotated ground truths.

Nighttime Driving [23] The Nighttime Driving test set contains 50 nighttime images of resolution $1,920 \times 1,080$ from diverse visual scenes. All these 50 images have been annotated at the pixel level using the same 19 Cityscapes category labels. In our experiments, we only use Nighttime Driving test set for method evaluation.

5.3.2 Experimental settings

We implement the proposed DANNet using PyTorch on a single Nvidia 2080Ti GPU. Following [10], we train our network using the Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9 and a weight decay of 5×10^{-4}. The base learning rate is set to 2.5×10^{-4} and then we employ the poly learning rate policy to decrease it with a power of 0.9. The batch size is set to 2. We use Adam optimizer [60] for training the discriminators with β being set to $(0.9, 0.99)$. The learning rate of the discriminators is set to 2.5×10^{-4} and follows the same decay strategy as for the generator. In addition, we apply random cropping with the crop size of 512 on the scale between 0.5 and 1.0 for Cityscapes dataset, with the crop size of 960 on the scale between 0.9 and 1.1 on Dark Zurich dataset, and random horizontal flipping in the training. To make the training easier to converge, we use the semantic segmentation models that are pre-trained on Cityscapes for 150,000 epochs and report the performance of different segmentation models on the validation set of Cityscapes and Dark Zurich in Table 5.1.

5.3.3 Comparison with state-of-the-art methods

Comparison on Dark Zurich-test We first compare our DANNet with some existing state-of-the-art methods, including MGCDA [108], GCMA [106], DMAda [23] and several other domain adaptation approaches [117, 119, 69] on Dark Zurich-test,

Table 5.1 The mIoU performance of the pre-trained semantic segmentation models on the validation set of Cityscapes and Dark Zurich.

Method	Cityscapes-val	Dark Zurich-val
RefineNet [71]	65.20	15.16
DeepLab-v2 [10]	65.67	12.14
PSPNet [148]	63.37	12.28

and the results on the mIoU performance are reported in Table 5.2. Among these methods, MGCDA, GCMA, and DMAda share the same baseline RefineNet while the rest are based on Deeplab-v2 and they use the common ResNet-101 backbone [45] and the nighttime images in Dark Zurich-test as inputs during testing. Our DANNets with either DeepLab-v2, RefineNet or PSPNet all perform better than or tie to existing methods on this dataset, and the one with PSPNet achieves the best performance among all, with a 2.7% improvement of the overall mIoU over the highest score obtained by all existing methods (by MGCDA). We also observe that our DANNet significantly outperforms other methods on quite a few categories, such as road, sidewalk, and sky, which indicates that our method handles the large day-to-night domain gap very well even in discernible regions. Sample visualization results on Dark Zurich-val in Figure 5.3 also verify such observation.

Comparison on Night Driving We report the performance of the proposed DANNet and the same set of comparison methods on Night Driving test set in Table 5.3, with sample visualization results presented in Figure 5.4. It is worth to mention that Night Driving dataset is not labeled as elaborately as Dark Zurich-test as shown in Figure 5.4, and many categories that our DANNet predicts well (see Table 5.2), such as building and vegetation, are not annotated in this test set. We also notice that the category of sky is only labeled in 2 out of the 50 images in Night Driving test set. Even with these issues, our DANNet with PSPNet still achieves the second best performance (MGCDA gets the best) on this dataset.

Table 5.2 The per-category results on Dark Zurich-test by current state-of-the-art methods and our DANNet.

Method	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain
RefineNet [71]-Cityscapes	68.8	23.2	46.8	20.8	12.6	29.8	30.4	26.9	43.1	14.3
DeepLab-v2 [10]-Cityscapes	79.0	21.8	53.0	13.3	11.2	22.5	20.2	22.1	43.5	10.4
PSPNet [148]-Cityscapes	78.2	19.0	51.2	15.5	10.6	30.3	28.9	22.0	56.7	13.3
AdaptSegNet-Cityscapes→DZ-night [117]	86.1	44.2	55.1	22.2	4.8	21.1	5.6	16.7	37.2	8.4
ADVENT-Cityscapes→DZ-night [119]	85.8	37.9	55.5	27.7	14.5	23.1	14.0	21.1	32.1	8.7
BDL-Cityscapes→DZ-night [69]	85.3	41.1	61.9	32.7	17.4	20.6	11.4	21.3	29.4	8.9
DMAda [23]	75.5	29.1	48.6	21.3	14.3	34.3	36.8	29.9	49.4	13.8
GCMA [106]	81.7	46.9	58.8	22.0	20.0	41.2	**40.5**	**41.6**	64.8	31.0
MGCDA [108]	80.3	49.3	66.2	7.8	11.0	**41.4**	38.9	39.0	64.1	18.0
DANNet (DeepLab-v2)	88.6	53.4	69.8	34.0	20.0	25.0	31.5	35.9	69.5	**32.2**
DANNet (RefineNet)	90.0	54.0	**74.8**	**41.0**	21.1	25.0	26.8	30.2	**72.0**	26.2
DANNet (PSPNet)	**90.4**	**60.1**	71.0	33.6	**22.9**	30.6	34.3	33.7	70.5	31.8

Method	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	**mIoU**
RefineNet [71]-Cityscapes	0.3	36.9	49.7	63.6	6.8	0.2	24.0	33.6	9.3	28.5
DeepLab-v2 [10]-Cityscapes	18.0	37.4	33.8	64.1	6.4	0.0	52.3	30.4	7.4	28.8
PSPNet [148]-Cityscapes	20.8	38.2	21.8	52.1	1.6	0.0	53.2	23.2	10.7	28.8
AdaptSegNet-Cityscapes→DZ-night [117]	1.2	35.9	26.7	68.2	45.1	0.0	50.1	33.9	15.6	30.4
ADVENT-Cityscapes→DZ-night [119]	2.0	39.9	16.6	64.0	13.8	0.0	58.8	28.5	20.7	29.7
BDL-Cityscapes→DZ-night [69]	1.1	37.4	22.1	63.2	28.2	0.0	47.7	**39.4**	15.7	30.8
DMAda [23]	0.4	43.3	50.2	69.4	18.4	0.0	27.6	34.9	11.9	32.1
GCMA [106]	32.1	**53.5**	47.5	**75.5**	39.2	0.0	49.6	30.7	21.0	42.0
MGCDA [108]	55.8	52.1	**53.5**	74.7	**66.0**	0.0	37.5	29.1	22.7	42.5
DANNet (DeepLab-v2)	82.3	44.2	43.7	54.1	22.0	0.1	40.9	36.0	**24.1**	42.5
DANNet (RefineNet)	**8**4.0	47.0	33.9	68.2	19.0	**0.3**	66.4	38.3	23.6	44.3
DANNet (PSPNet)	80.2	45.7	41.6	67.4	16.8	0.0	**73.0**	31.6	22.9	**45.2**

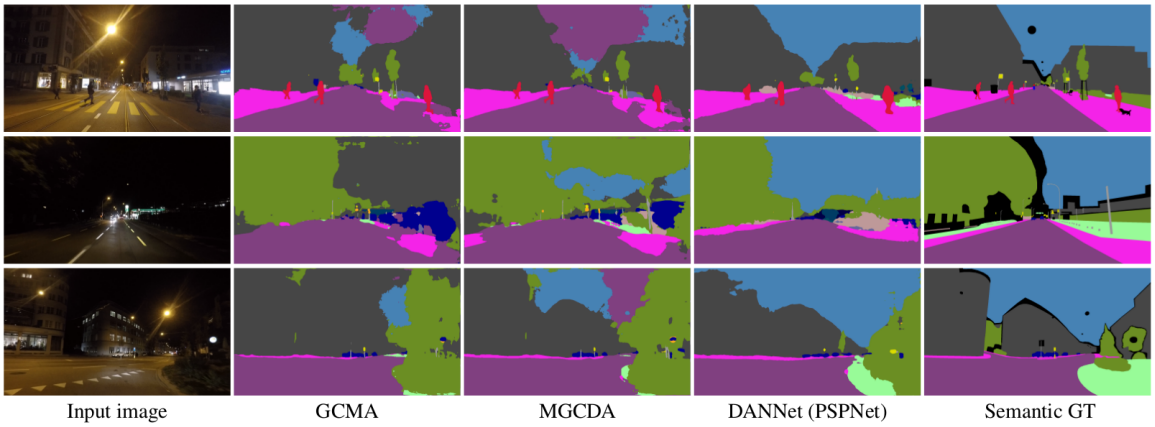


Figure 5.3 Visualization comparison of our DANNet with some existing state-of-the-art methods on three samples from Dark Zurich-val.

Table 5.3 Comparison of our DANNet with some existing state-of-the-art methods on Nighttime Driving test set [23].

Method	mIoU
RefineNet [71]-Cityscapes	32.75
DeepLab-v2 [10]-Cityscapes	25.44
PSPNet [148]-Cityscapes	27.65
AdaptSegNet-Cityscapes→DZ-night [117]	34.5
ADVENT-Cityscapes→DZ-night [119]	34.7
BDL-Cityscapes→DZ-night [69]	34.7
DMAda [23]	36.1
GCMA [106]	45.6
MGCDA [108]	**49.4**
DANNet (RefineNet)	42.36
DANNet (DeepLab-v2)	44.98
DANNet (PSPNet)	<u>47.70</u>

5.3.4 Ablation study

To demonstrate the effectiveness of different components of the proposed DANNet, we train several model variants for 35,000 epochs and test them on Dark Zurich-val. The performance results are reported in Table 5.4. Adaptation to Dark Zurich-N using AdaptSegNet [117] serves as the baseline and DANNet is the full model.

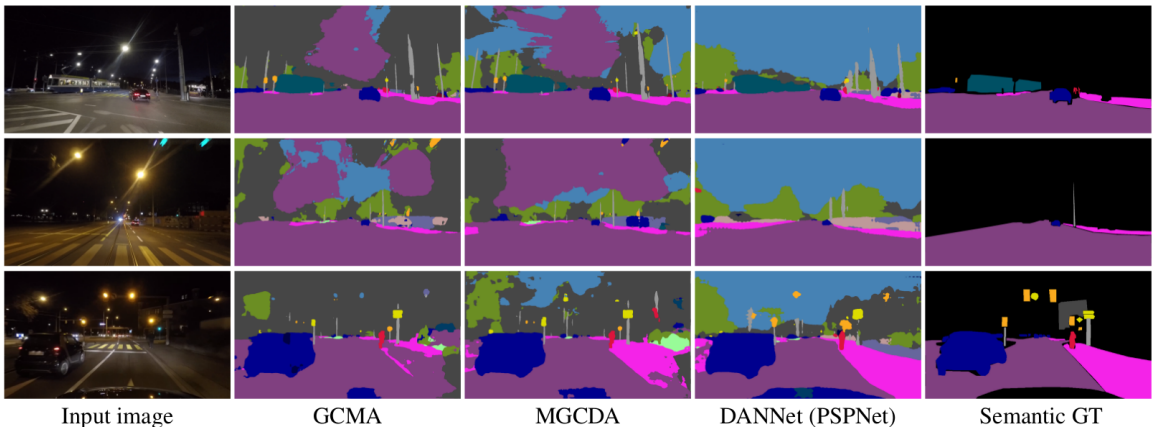


Figure 5.4 Visualization comparison of our DANNet with some existing state-of-the-art methods on three samples from Night Driving-test.

We observe that coarsely aligned Dark Zurich-D is quite important although it is unlabeled, and the pseudo labels drawn from the predictions on Dark Zurich-D also play a key role in our network, without which the mIoU decreases by 13.78%. Both the image relighting network and the corresponding loss L_{light} can enhance the performance. We also see that the specially designed loss L_{static} is better than directly applying the cross entropy or focal loss to calculate the static loss.

In addition, the re-weighting strategy is verified to be useful and can further boost the performance. As shown in Figure 5.5, this strategy helps segment the small objects. We find that the selection of the value *std* is also important in applying the re-weighting strategy.

We test different *std* values and the performance curve of the proposed DANNet on Dark Zurich-val is shown in Figure 5.6. The optimal performance is achieved when setting $std = 0.16$ during testing. By directly applying the commonly-used weights provided by OCNet [138], it only achieves 35.05 mIoU on DZ-val dataset, which is less than that of our DANNet. In general, the full settings of our DANNet bring about an additional 10% performance increase over the state-of-the-art approaches on Dark Zurich-val.

Table 5.4 Ablation study on several model variants of our DANNet (PSPNet) on Dark Zurich-val.

Method	mIoU
GCMA [106]	26.65
MGCDA [108]	26.10
AdaptSegNet-Cityscapes→DZ-night [117]	20.19
w/o Dark Zurich-D	22.78
w/o relighting network & L_{light}	34.14
w/o L_{light}	35.05
w/o L_{static}	20.48
w/ Cross Entropy Loss in L_{static}	33.61
w/ Focal Loss in L_{static}	36.49
w/o re-weighting on pseudo labels	32.71
w/o re-weighting on prediction	32.22
w/o pretrained segmentation model	30.74
DANNet	**36.76**

Figure 5.7 shows the confusion matrix computed using Dark Zurich-val. We can observe that pole is easily to be misclassified as vegetation, traffic sign usually is misclassifed as building.

5.3.5 More quantitative results

We first report the quantitative comparison results of the re-weighting strategy for each category by our DANNet on Dark Zurich-val in Table 5.5. It is easy to see that most of the classes are predicted more accurately by applying the proposed re-weighting strategy. We also provide the per-category results for the ablation study on several model variants of our DANNet (supplement to Table 4 in the main paper) in Table 5.6.

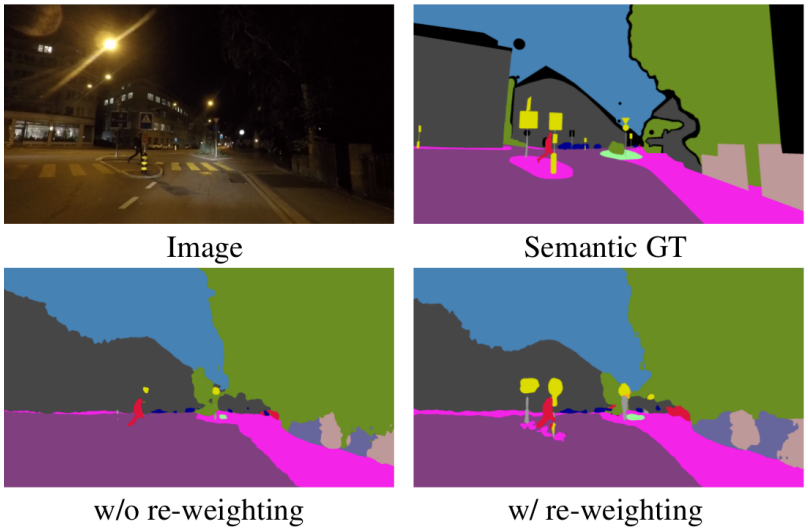


Figure 5.5 Visualization results of w/ and w/o the re-weighting strategy on a sample from Dark Zurich-val by our DANNet (PSPNet).

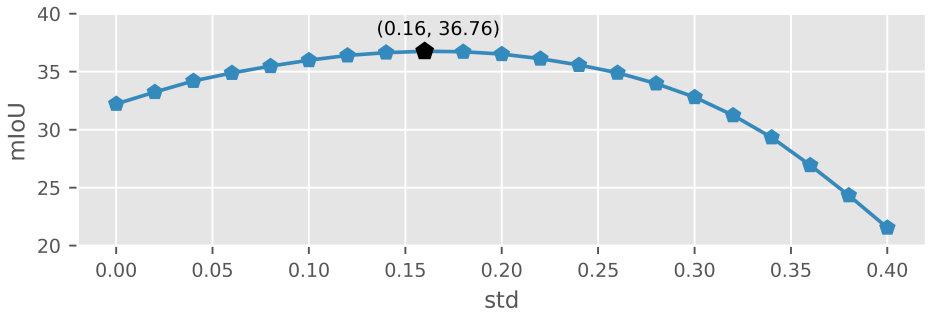


Figure 5.6 Ablation study on the value of *std* in the re-weighting strategy on Dark Zurich-val by our DANNet (PSPNet).

5.4 Chapter Summary

In this work, we have proposed a novel end-to-end neural network DANNet for unsupervised nighttime semantic segmentation, which performs an adaptation from a labeled daytime dataset to unlabeled day-night image pairs. In our DANNet, an image relighting network with a special light loss function is first used to make the intensity distributions of the images from different domains to be close to each other.

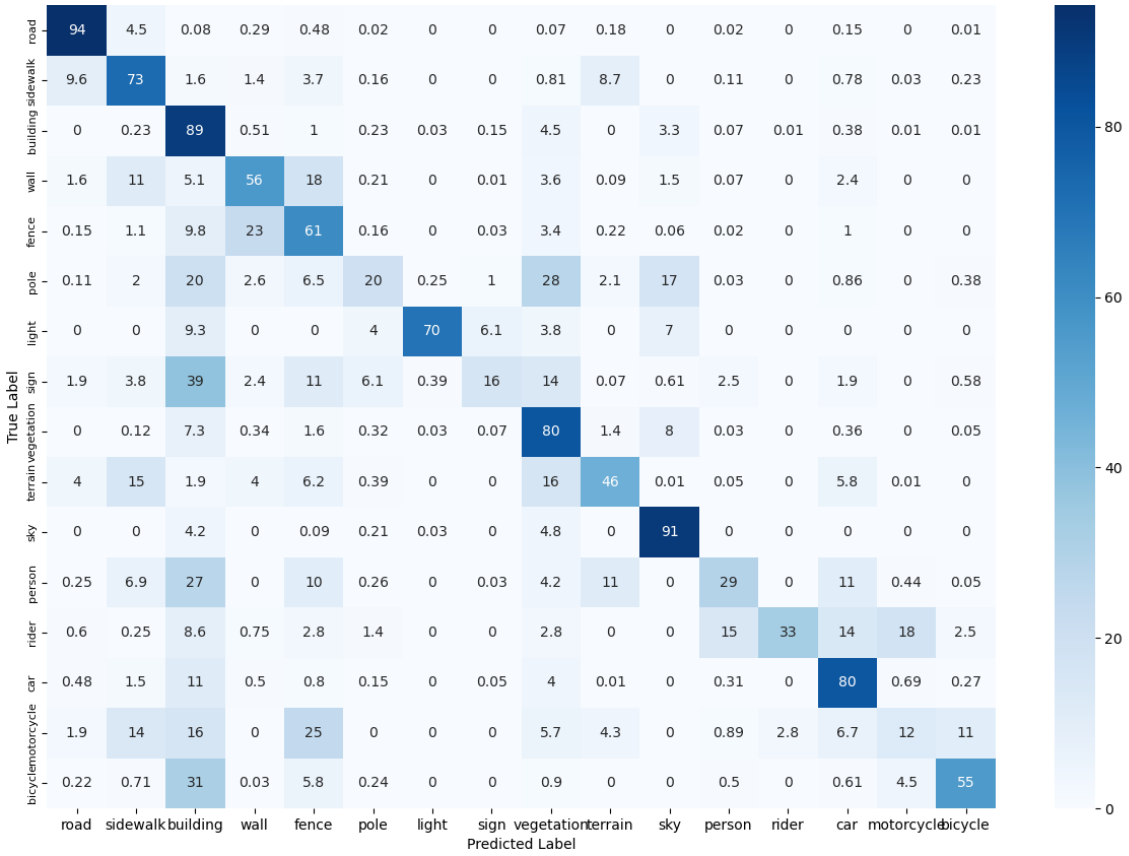


Figure 5.7 Normalized confusion matrix (%) of semantic segmentation in the Dark Zurich-val.

Then the unlabeled Dark Zurich-D data is used to bridge the domain gap between the labeled daytime images (Cityscapes) and the unlabeled nighttime images (Dark Zurich-N). By leveraging the similar illumination patterns between Dark Zurich-D and Cityscapes and coarse alignment of static categories between Dark Zurich-D and Dark Zurich-N, our DANNet performs multi-target domain adaptation as well as a re-weighting strategy to boost the performance for small objects. Experimental results demonstrated the effectiveness of each of the designed components and showed that our DANNet achieves the state-of-the-art performance on Dark-Zurich and Night Driving test datasets.

Table 5.5 The per-category results on Dark Zurich-val by our DANNet with or without the proposed re-weighting strategy. The higher results are presented in **b**old.

Method	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain
w/o re-weighting	88.89	55.68	74.19	26.78	**4**1.73	7.48	20.93	4.42	67.31	19.01
w/ re-weighting	**9**0.93	**5**9.35	**7**7.08	**3**7.79	40.35	**1**4.08	**3**9.95	**1**4.15	**6**8.26	**2**8.61

Method	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	**m**IoU
w/o re-weighting	80.22	19.5	24.69	**4**8.55	0.0	0.0	0.0	4.27	28.51	32.22
w/ re-weighting	**8**2.91	**2**1.01	**2**5.44	46.99	0.0	0.0	0.0	**1**0.21	**4**1.33	**3**6.76

Table 5.6 The per-category results on Dark Zurich-val for the ablation study on several model variants of our DANNet.

Method	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain
GCMA [106]	73.4	38.4	48.1	12.3	27.9	23.9	47.4	15.1	55.6	27.4
MGCDA [108]	59.0	46.1	64.4	6.0	11.5	26.0	40.1	14.8	56.6	27.3
AdaptSegNet-Cityscapes→DZ-night [117]	68.7	12.7	54.5	6.7	37.8	17.8	19.8	10.9	43.0	14.8
w/o Dark Zurich-D	75.3	25.0	48.7	15.2	32.6	18.6	13.2	13.0	41.0	25.8
w/o relighting network & L_{light}	89.3	56.9	79.0	36.2	30.5	9.7	32.4	8.1	67.4	33.9
w/o L_{light}	90.9	53.2	78.6	39.4	39.4	12.8	37.7	7.36	65.9	27.2
w/o L_{static}	70.8	20.7	42.8	11.8	28.7	17.6	23.0	12.5	38.9	24.1
w/ Cross Entropy Loss in L_{static}	89.9	59.1	77.2	38.6	33.0	12.7	28.3	10.2	66.2	30.1
w/ Focal Loss in L_{static}	88.4	60.5	80.0	36.8	33.3	13.9	41.6	6.6	67.5	35.1
w/o re-weighting on pseudo labels	90.2	58.4	76.2	32.0	32.7	9.8	29.2	7.4	62.4	30.1
w/o re-weighting on prediction	88.9	55.7	74.2	26.8	41.7	7.45	20.9	4.4	67.3	19.0
w/o pretrained segmentation model	88.1	56.5	70.1	30.6	31.6	11.7	22.2	9.4	58.1	23.1
DANNet (PSPNet)	90.9	59.4	77.1	37.8	40.4	14.1	40.0	14.2	68.3	28.6

Method	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	**mIoU**
GCMA [106]	25.8	27.1	1.4	54.3	0.0	0.0	0.0	13.9	14.4	26.7
MGCDA [108]	31.4	23.3	1.2	61.6	0.0	0.0	0.0	15.0	12.5	26.1
AdaptSegNet-Cityscapes→DZ-night [117]	8.07	14.2	0.0	41.4	0.0	0.0	0.0	8.3	24.8	20.2
w/o Dark Zurich-D	1.5	18.2	26.3	42.2	0.0	0.0	0.0	6.1	30.7	22.8
w/o relighting network & L_{light}	83.7	21.8	0.7	52.6	0.0	0.0	0.0	14.1	32.3	34.1
w/o L_{light}	82.3	18.9	9.7	55.5	0.0	0.0	0.0	14.6	32.5	35.1
w/o L_{static}	4.1	16.3	23.6	40.1	0.0	0.0	0.0	2.6	13.6	20.5
w/ Cross Entropy Loss in L_{static}	81.7	17.1	0.4	55.0	0.0	0.0	0.0	12.1	27.4	33.6
w/ Focal Loss in L_{static}	83.4	19.2	20.0	56.6	0.0	0.0	0.0	19.1	31.5	36.5
w/o re-weighting on pseudo labels	83.5	15.4	7.1	52.8	0.0	0.0	0.0	11.4	23.0	32.7
w/o re-weighting on prediction	80.2	19.5	24.7	48.6	0.0	0.0	0.0	4.3	28.5	32.2
w/o pretrained segmentation model	76.1	16.2	7.8	46.9	0.0	0.0	0.0	14.8	20.9	30.7
DANNet (PSPNet)	82.9	21.0	25.4	47.0	0.0	0.0	0.0	10.2	41.3	36.8

Chapter 6

Image to Video Domain Adaptive Semantic Segmentation

6.1 Overview

Generating a dense prediction map for each frame to indicate specific class of each pixel, video semantic segmentation is a fundamental task in computer vision with important applications in autonomous driving and robotics [22, 21]. Just like image semantic segmentation [77, 10, 148], state-of-the-art supervised learning methods for video semantic segmentation require large-scale labeled training data, which is costly and laborious to annotate manually [33, 36, 154, 75, 89]. Semi-supervised training [87, 89, 155, 12] can help relieve the manual-annotation burden but still requires to annotate sparsely sampled video frames from the same domain.

One way to avoid completely manual annotation is to train segmentation models on simulated data that are easily rendered by video game engines and therefore self-annotated, and then transfer the learned knowledge into real-world video data for improving semantic segmentation. Underlying this is actually an important concept of domain adaptation – from the source domain of simulated data to the target domain of real-world data – which was initially studied for image semantic segmentation [48, 9, 117, 119, 133, 144], *e.g.*from GTA5 [100] to Cityscapes [21] and from SYNTHIA [103] to Cityscapes with much success. This concept of domain adaptation also has been extended to tackle video semantic segmentation – a straightforward approach is to treat each video frame as an image and directly perform image-to-image domain adaptation to segment each frame independently [42]. By ignoring the temporal information along the videos, these approaches usually exhibit limited performance on video semantic segmentation.

Recent progress on video semantic segmentation witnesses two inspirational works [42, 112] that coincidentally suggest video-to-video domain adaptation. Both of them employ adversarial learning of the video predictions between the source and target domains and therefore consider spatial-temporal information in both domains. While we can generate large-scale simulated videos to well reflect the source domain, it may

lead to high complexity of the network and its training in the source domain. Motivated by such observation and with the goal to reduce the cost, we aim to develop a new concept of *image-to-video domain adaptive semantic segmentation* (I2VDA), where the source domain only contains simulated images and the target domain real-world videos.

Videos contain spatial-temporal information and video-to-video domain adaption can exploit and pass both spatial and temporal knowledge from the source domain to the target domain. The fundamental hypothesis of the proposed image-to-video domain adaptation for semantic segmentation is that we only need to pass the spatial knowledge from the source domain to the target domain, not the temporal one. In principle, we have two major arguments for this hypothesis: 1) the between-frame continuity is the most important temporal knowledge for video semantic segmentation and such continuity can be well exploited from videos in the target domain, *e.g.*, the optical flow along each video; and 2) the temporal information between the source and target domains practically may not show a systematic domain gap that has to be filled by adaptation. On the other hand, using images, instead of videos, in the source domain can significantly reduce the required training-data size and the network complexity.

In this paper, we verify the above fundamental hypothesis by developing a new image-to-video domain adaptive semantic segmentation method. In our method, we propose a novel temporal augmentation strategy to make use of the temporal consistency in the target domain and improve the target predictions. Moreover, the domain gap is bridged by the widely-used adversarial learning strategy which only considers the spatial features in the two domains. To relieve the instability of the adversarial learning, we further introduce a new training scheme that leverages a proxy network to generate pseudo labels for target predictions on-the-fly. We conduct extensive experiments to demonstrate the effectiveness of the proposed method and

each of its strategy. The main contributions of this paper are summarized as follows:

- We propose and verify a new finding – for segmenting real videos, it is sufficient to perform domain adaptation from synthetic *images*, instead of synthetic *videos*, i.e., there is no need to adapt and transfer temporal information in practice.
- We introduce for the first time the setting of image-to-video domain adaptive semantic segmentation, *i.e.*, which uses labeled images as the source domain in domain adaptation for video semantic segmentation.
- We successfully develop an I2VDA method with two novel designs: 1) a temporal augmentation strategy to better exploit and learn diverse temporal consistency patterns in the target domain; and 2) a training scheme to achieve more stable adversarial training with the help of a proxy network.
- Experimental results on two synthetic-to-real scenarios demonstrate the effectiveness of the proposed method and verify our fundamental hypothesis. Without simulating/adapting temporal information in the source domain, our method still outperforms existing state-of-the-art video-to-video domain adaptation methods.

6.2 Proposed Method

6.2.1 Problem setting

The goal of image-to-video domain adaptive semantic segmentation is to transfer only spatial knowledge from a labeled source domain $\mathtt{S}$ to an unlabeled target domain $\mathtt{T}$. Same as the setting of domain adaptive video segmentation [42, 112], the target domain is in the format of video sequences $I^{\mathtt{T}} := \{I_0^{\mathtt{T}}, I_1^{\mathtt{T}}, ..., I_n^{\mathtt{T}}, ...\}$ with $I^{\mathtt{T}} \in \mathtt{T}$. In

contrast, the source domain consists of a set of image-label pairs that are not in chronological order, $(I^{\mathtt{S}}, GT^{\mathtt{S}}) \in \mathtt{S}$.

6.2.2 Framework overview

Our work bridges the spatial domain gap between the source and the target via adversarial learning and further considers the augmented temporal consistency in the target domain to achieve accurate predictions for the videos. In addition, a novel training scheme is introduced to improve the stability of the adversarial training. The proposed image-to-video domain adaptive semantic segmentation framework is illustrated in Figure 6.1. The main components include flow estimation network $\mathcal{F}$ (for temporal augmentation and consistency learning), semantic segmentation network $\mathcal{M}$ and its proxy $\mathcal{M}'$, and discriminator $\mathcal{D}$.

Flow estimation network In our work, the flow estimation network $\mathcal{F}$ is used to obtain the optical flow between two consecutive frames and the computed optical flow is used for two purposes: 1) synthesizing an intermediate frame I_t given two consecutive target frames I_0 and I_1; and 2) warping the predictions to ensure temporal consistency in the target domain. Here, we use pre-trained FlowNet2[1] [55] as $\mathcal{F}$ to estimate the optical flow.

Semantic segmentation network We adopt the widely-used Deeplab-v2 [10] with a backbone of ResNet-101 [45] (pre-trained on ImageNet [26]) as the semantic segmentation network $\mathcal{M}$. During training, $\mathcal{M}$ is used in a training mode to generate the predictions for $I^{\mathtt{S}}$, I_0 and I_t, which are denoted as $P^{\mathtt{S}}$, P_0 and P_t, respectively. Note that these predictions are upsampled to the same resolution as the input images. In addition, the proxy network $\mathcal{M}'$ has the same architecture as $\mathcal{M}$, which instead is used in an evaluation mode to generate pseudo labels given I_1 as the input. The parameters of $\mathcal{M}'$ are updated via a copy from $\mathcal{M}$ at a certain frequency.

[1] https://github.com/NVIDIA/flownet2-pytorch

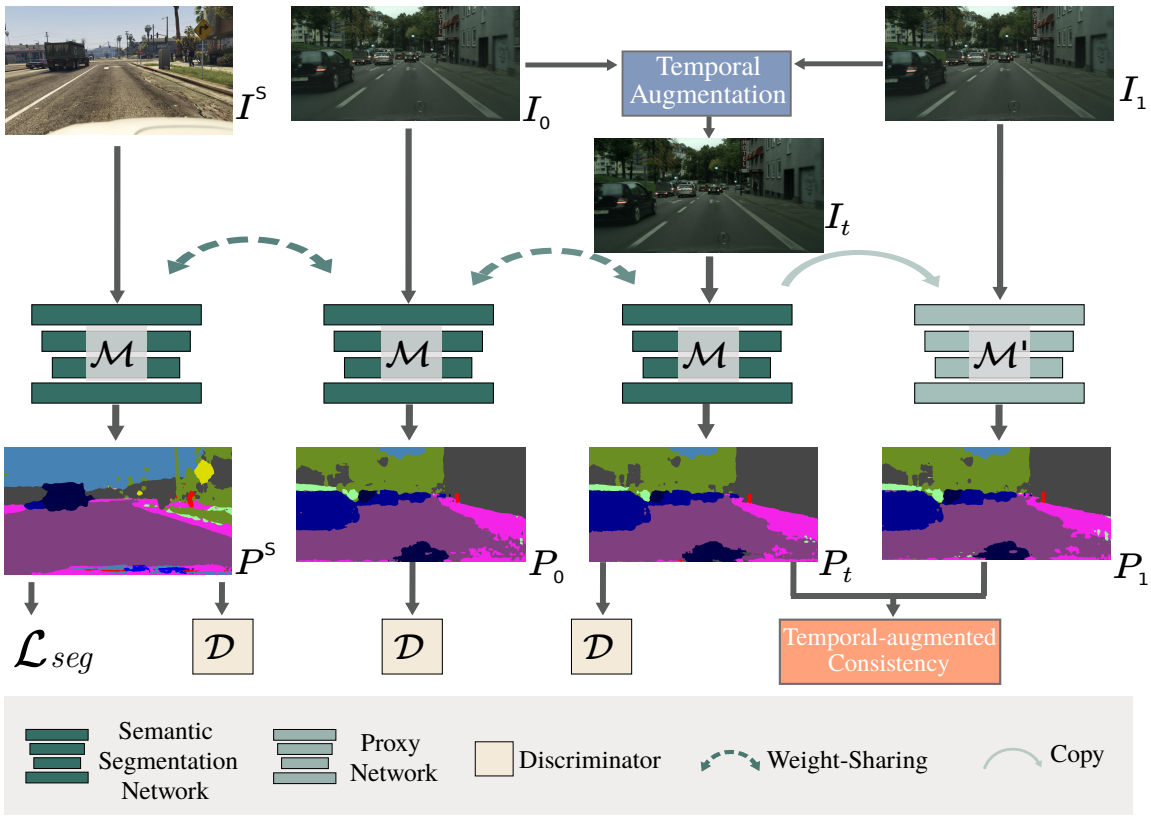


Figure 6.1 An illustration of the proposed image-to-video domain adaptive semantic segmentation framework. During training, the framework requires three inputs including a source image $I^{\mathtt{S}}$ and two consecutive frames I_0 and I_1 from a target video $I^{\mathtt{T}}$. First, an intermediate target frame I_t $(0 < t < 1)$ is synthesized using I_0 and I_1 via a frame interpolation with temporal augmentation. Then, $I^{\mathtt{S}}$, I_0 and I_t are fed into a weight-sharing semantic segmentation network $\mathcal{M}$ to obtain the corresponding predictions. A semantic segmentation loss $\mathcal{L}_{seg}$ is computed using the prediction of $I^{\mathtt{S}}$ and its label $GT^{\mathtt{S}}$. A discriminator $\mathcal{D}$ is employed to distinguish outputs from the source domain $\mathtt{S}$ and target domain $\mathtt{T}$. Besides, a proxy network $\mathcal{M}'$ takes I_1 as the input to generate its pseudo label which is used for ensuring the temporal consistency of the target predictions. Note that the parameters of $\mathcal{M}'$ are updated via copying from $\mathcal{M}$ instead of back propagation.

Discriminator To perform the adversarial learning, we employ the discriminator $\mathcal{D}$ to distinguish whether the prediction is from the source domain or the target one by following [117].

6.2.3 The temporal augmentation strategy

From our perspective, the source does not require to be an ordered video sequence, but the temporal patterns such as frame rate and the speed of the ego-vehicle in the target domain do matter for performance improvements. As stated in [82], the temporal constraint is sensitive to object occlusions and lost frames, *e.t.c.*. Here we propose a novel temporal augmentation strategy to achieve robust temporal consistency in the target domain, which is implemented based on a well-studied task – video frame interpolation [57].

Different from images, videos have the unique temporal dimension where more choices on data augmentation strategies can be applied other than those only focusing on the spatial dimension, *e.g.*, random flipping and rotation. In [155], Zhu *et al.* proposed to synthesize more image-label pairs by transforming a past frame and its corresponding label via video prediction technique for video semantic segmentation. This method can tackle the general video semantic segmentation task where only sparsely sampled video frames are labeled – the labels can be propagated to the unlabeled or synthesized frames. However, it is not applicable to our setting because of no labels in the target videos.

We carefully design a temporal augmentation strategy that is suitable for robust unlabeled video representation to improve the diversity of temporal consistency in the target domain. Specifically, given two consecutive target frames I_0 and I_1, we first extract the bi-directional optical flows using the pre-trained $\mathcal{F}$ as follows:

$$F_{0\to1} = \mathcal{F}(I_0, I_1), \quad F_{1\to0} = \mathcal{F}(I_1, I_0). \tag{6.1}$$

By assuming that the optical flow field is locally smooth as [57], $F_{t\to0}$ and $F_{t\to1}$, for some $t \in (0, 1)$ randomly generated in each training iteration, can be approximated by:

$$F_{t\to0} \approx tF_{1\to0}, \quad F_{t\to1} \approx (1-t)F_{0\to1}. \tag{6.2}$$

Then, an intermediate frame I_t can be formulated as:

$$I_t = \alpha \mathcal{W}(I_0, F_{t\to 0}) + (1-\alpha)\mathcal{W}(I_1, F_{t\to 1}), \tag{6.3}$$

where the parameter α controls the contribution of I_0 and I_1 and is set to 0.5 in all experiments, and $\mathcal{W}(\cdot,\cdot)$ is a backward warping function implemented using the bilinear interpolation [151, 57].

The blue region of Figure 6.2 illustrated the process of the proposed temporal augmentation strategy. Next, we will show how to use the produced synthesized frame to achieve better temporal-augmented consistency learning in the target domain.

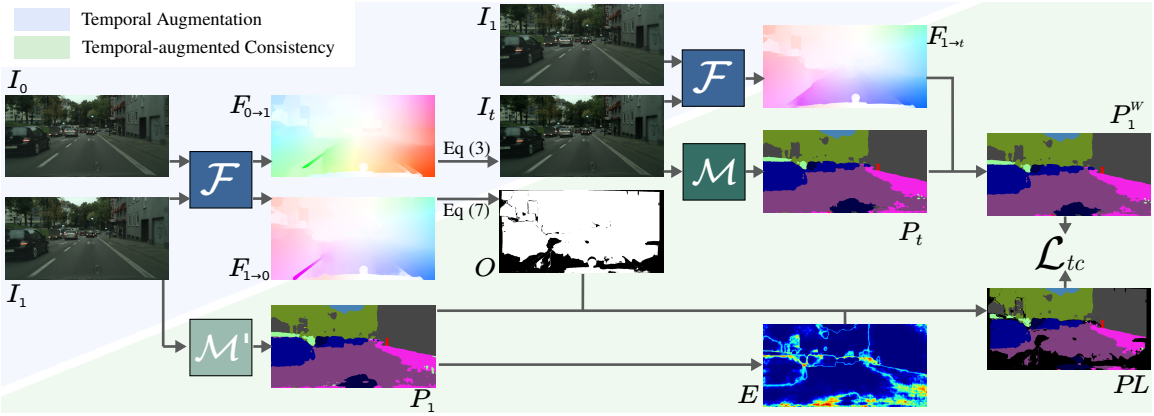


Figure 6.2 An illustration of the proposed temporal augmentation strategy (Sec. 6.2.3) and temporal-augmented consistency learning (Sec. 6.2.4) in the target domain.

6.2.4 The temporal-augmented consistency learning

Temporal consistency learning is a commonly-used constraint for video-level tasks [85, 89, 76, 131, 42]. In this work, we extend this idea and propose the temporal-augmented consistency learning leveraging the synthesized frame I_t obtained via Eq. (6.3). The goal of this operation is to not only improve the prediction consistency between consecutive frames, but more importantly, fulfil the on-the-fly self-training to stablize the adversarial training. As illustrated in the green part of Figure 6.2,

the temporal-augmented consistency loss computed between a propagated prediction P_1^W of I_t and a corresponding pseudo label PL. Below we detail how to achieve this temporal-augmented consistency learning.

Firstly, the target frame I_0 and the synthesized frame I_t are fed into the segmentation network $\mathcal{M}$ to obtain the corresponding segmentation predictions P_0, $P_t \in \mathbb{R}^{C\times H\times W}$, where C, H and W are the number of categories, the height and the width of the input image, respectively.

The prediction P_t is then propagated forward to the moment 1 to generate

$$P_1^W = \mathcal{W}(P_t, F_{1\to t}), \tag{6.4}$$

where $F_{1\to t}$ denotes the optical flow from the moment 1 to moment t and is computed by:

$$F_{1\to t} = \mathcal{F}(I_1, I_t). \tag{6.5}$$

Simultaneously, the pseudo label of P_1^W is generated via another path. The proxy network $\mathcal{M}'$ first takes the other target frame I_1 as input and output the prediction P_1 (More details related to the usage of $\mathcal{M}'$ are introduced later in Sec. 6.2.5). Then the prediction P_1 is rectified according to its own confidence and only the predictions with the high confidence will be kept as the pseudo labels. Following [119], we first compute the entropy map $E \in [0,1]^{H\times W}$ via:

$$E = -\frac{1}{\log(C)} \sum_{k=1}^{C} \Big(P_1^{(k)} \cdot \log(P_1^{(k)}) \Big). \tag{6.6}$$

Since the synthesized frame I_t is not perfect, especially in the occlusion region, we further exclude the occlusion region in P_1 during the temporal-augmented consistency learning. Specifically, the occlusion region $O \in \mathbb{R}^{H\times W}$ is defined as:

$$O = \begin{cases} 1, & \text{if } \mathcal{W}(F_{1\to 0}, F_{0\to 1}) + F_{0\to 1} < \eta. \\ 0, & \text{otherwise,} \end{cases} \tag{6.7}$$

where η is a hyper-parameter (set to 1 in all experiments). The final rectified pseudo label PL is then given by:

$$PL = \begin{cases} \text{ArgMax}(P_1), & \text{if } E < \delta \text{ and } O = 1. \\ i, & \text{otherwise,} \end{cases} \tag{6.8}$$

where the threshold $\delta = 0.8$, and i is the ignored class label which is not considered during training. The rectified pseudo label PL is used to guide the prediction P_1^W which is achieved by minimizing the following temporal-augmented consistency loss $\mathcal{L}_{tc}$:

$$\mathcal{L}_{tc} = CE(P_1^W, PL). \tag{6.9}$$

Different from [76, 42] which compute the L1 distance for temporal consistency, we employ the cross-entropy (CE) instead. Note that this is a non-trivial design, since Eq. (6.9) is also used to achieve the on-the-fly self-training. The CE loss is a common choice for self-training-based approaches [69, 133, 59] in domain adaptive semantic segmentation.

6.2.5 Proxy network for on-the-fly self-training

The usage of the proxy network is motivated by two observations: 1) the instability of the adversarial training strategy in existing domain adaptation approaches [117, 119]; and 2) the self-training technique requires multiple training stages but is not able to improve the performance on the target domain. Therefore, in this paper we propose to employ a proxy network $\mathcal{M}'$ to implicitly generate the pseudo labels for P_t on-the-fly. Specifically, $\mathcal{M}'$ gets starting to work after a few training iterations and it is used only in an evaluation mode. The parameters of $\mathcal{M}'$ will be updated via copying from M at every a fixed number of iterations.

6.2.6 Pipeline and other training objectives

In summary, we describe the whole training pipeline in Algorithm 2 with the involved loss functions listed and discussed below.

Algorithm 2 – I2VDA

Input: Source images $\{I^{\mathrm{S}}\}$, source labels $\{GT^{\mathrm{S}}\}$, two consecutive frames $\{I_0, I_1\}$ from target videos, the base segmentor $\mathcal{M}$ with parameter $\theta_{\mathcal{M}}$, the proxy network $\mathcal{M}'$ with parameter $\theta_{\mathcal{M}'}$ and discriminator $\mathcal{D}$ with parameter $\theta_{\mathcal{D}}$, the max number of training iterations MAX_ITER, the copying frequency ITER_COPY, the training iterations ITER_LAUNCH before launching $\mathcal{M}'$.

Output: Optimal $\theta_{\mathcal{M}}$

iter = 0
for iter<MAX_ITER **do**
    Synthesize I_t via Eq. (6.3);
    Feed I^{S}, I_0 and I_t into $\mathcal{M}$ to obtain the predictions;
    if iter % ITER_COPY **then**
        Update the parameters: $\theta_{\mathcal{M}'} \leftarrow \theta_{\mathcal{M}}$;
    end if
    if iter < ITER_LAUNCH **then**
        Update $\theta_{\mathcal{M}}$ using $(\mathcal{L}_{seg} + 0.01\mathcal{L}_{adv})$;
    else
        Feed I_1 into $\mathcal{M}'$ and obtain PL;
        Compute $\mathcal{L}_{tc}$ using Eq. (6.9);
        Update $\theta_{\mathcal{M}}$ using $(\mathcal{L}_{seg} + 0.01\mathcal{L}_{adv} + \mathcal{L}_{tc})$;
    end if
    Update $\theta_{\mathcal{D}}$ using $\mathcal{L}_d$ defined in Eq. (6.12);
    iter += 1;
end for
Return $\theta_{\mathcal{M}}$;

We compute the semantic segmentation loss based on CE to train $\mathcal{M}$ to learn knowledge from the source domain:

$$\mathcal{L}_{seg} = CE(P^{\mathrm{S}}, GT^{\mathrm{S}}). \tag{6.10}$$

Minimizing the adversarial loss can close the gap between the source and target predictions so that the target prediction can fool the discriminator. The adversarial

loss $\mathcal{L}_{adv}$ is defined as:

$$\mathcal{L}_{adv} = (\mathcal{D}(P_0) - r)^2 + (\mathcal{D}(P_t) - r)^2, \tag{6.11}$$

where r is the label indicating the source domain which has the same resolution as the output of the discriminator. The final loss of semantic segmentation network can be expressed as $\mathcal{L} = \mathcal{L}_{seg} + \mathcal{L}_{tc} + 0.01\mathcal{L}_{adv}$. Besides, the goal of the discriminator is to distinguish between the source and target predictions which is trained with the following objective function:

$$\mathcal{L}_d = (\mathcal{D}(P^{\mathtt{S}}) - r)^2 + \frac{1}{2}(\mathcal{D}(P_0) - f)^2 + \frac{1}{2}(\mathcal{D}(P_t) - f)^2, \tag{6.12}$$

where f is the label indicating the target domain with the same resolution as the output of discriminator.

6.3 Experimental results

6.3.1 Datasets

VIPER [99] dataset comprises 254,064 fully annotated video frames for training, validation and testing rendered from a computer game. We use 13,367 images marked as *0[2] with their labels as one of our source datasets. The frame resolution is $1,920 \times 1,080$. Following [42], 15 classes are considered for adaptation.

SYNTHIA [103] dataset is a synthetic dataset that consists of photo-realistic video frames rendered from a virtual city. It contains 8,000 labeled frames with a resolution of $1,280 \times 720$. We use the 850 labeled images from SYNTHIA-SEQS-04-DAWN[3] as another source dataset. Note that we remove the temporal constraint by randomly shuffling the frames in time. Following [42], 11 classes are considered for adaptation.

[2]https://playing-for-benchmarks.org/download/

[3]https://synthia-dataset.net/downloads/

Cityscapes [21] dataset focuses on semantic understanding of real urban street scenes. It contains 5,000 images with fine annotations that are split into 2,975/ 500/1,525 for training/validation/testing. Each annotated image is the 20^{th} image from a 30 frame video snippets. The resolution of each image is $2,048 \times 1,024$. We use it as the target domain in this work.

6.3.2 Experimental settings

We implement the proposed I2VDA method using Pytorch. Following [117], our semantic segmentation network $\mathcal{M}$ is trained using Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9 and its initial learning rate is 2.5×10^{-4}. The discriminator $\mathcal{D}$ is optimized using Adam with a β of $(0.9, 0.99)$ and its initial learning rate is 1.0×10^{-4}. We employ the polynomial decay with a power of 0.9 on the learning rates of both $\mathcal{M}$ and $\mathcal{D}$.

The images in VIPER [99], SYNTHIA [103], Cityscapes [21] are resized to 896×512, 1280×768 and 1024×512, respectively. We don't perform any spatial-level data augmentation strategy during training and testing. Each experiment in this paper is run for 50,000 iterations with a batch size of 2 on two Tesla V100 GPUs. Especially for testing, we only feed each frame independently into $\mathcal{M}$ to achieve the prediction without using optical flow. The mean intersection-over-union (mIoU) is used as the main evaluation metric, for which the higher the better. We also report video-specific metric of "Temporal Consistency" (TC) [76], which is again the higher the better.

6.3.3 Comparison with state-of-the-art methods

VIPER $\rightarrow$ Cityscapes We first compare our I2VDA method with the existing state-of-the-art methods, including [42, 133, 157, 91, 158, 158, 119] as in [42], for the VIPER $\rightarrow$ Cityscapes scenario. The quantitative results are reported in Table 6.1.

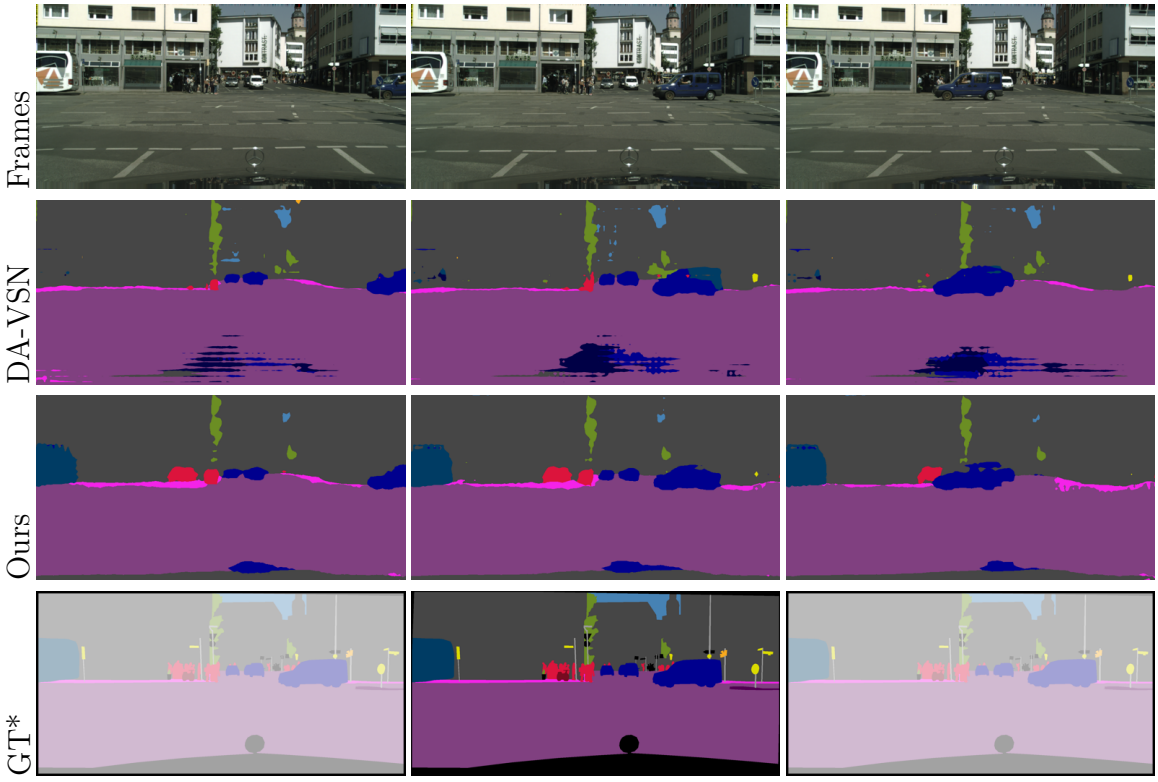
We find our method significantly outperforms (50.6% mIoU) all the others that

Table 6.1 Quantitative comparison results on the VIPER → Cityscapes domain adaptive video segmentation task. The best results are presented in **bold**, with the second best results underlined.

Methods	road	sidewalk	building	fence	traffic light	traffic sign	vegetation	terrain	sky	person	car	truck	bus	motorcycle	bicycle	mIoU (%)
Source only	56.7	18.7	78.7	6.0	22.0	15.6	81.6	18.3	80.4	59.9	66.3	4.5	16.8	20.4	10.3	37.1
AdvEnt [119]	78.5	31.0	81.5	22.1	29.2	26.6	81.8	13.7	80.5	58.3	64.0	6.9	38.4	4.6	1.3	41.2
CBST [158]	48.1	20.2	**84.8**	12.0	20.6	19.2	83.8	18.4	**84.9**	59.2	71.5	3.2	38.0	23.8	**37.7**	41.7
IDA [91]	78.7	33.9	82.3	22.7	28.5	26.7	82.5	15.6	79.7	58.1	64.2	6.4	41.2	6.2	3.1	42.0
CRST [157]	56.0	23.1	82.1	11.6	18.7	17.2	**85.5**	17.5	82.3	60.8	73.6	3.6	38.9	30.5	35.0	42.4
FDA [133]	70.3	27.7	81.3	17.6	25.8	20.0	83.7	**31.3**	82.9	57.1	72.2	22.4	49.0	17.2	7.5	44.4
DA-VSN [42]	**86.8**	**36.7**	83.5	22.9	30.2	27.7	83.6	26.7	80.3	60.0	79.1	20.3	47.2	21.2	11.4	47.8
I2VDA	80.0	34.2	83.6	**26.0**	**33.0**	**31.2**	**85.6**	**34.0**	78.3	**62.3**	**81.1**	**38.3**	**54.7**	**36.0**	3.2	**50.6**

are trained with VIPER videos, *i.e.*, use additional unlabeled frames that are adjacent to the labeled one for temporal modeling. Besides, these video-to-video domain adaptation approaches require two images and a pre-computed optical flow as inputs during testing, while our method performs only a per-frame inference without optical-flow computation. On the video-level evaluation, our method achieves 66.01% on TC metric, while DAVSN [42] obtain 63.82%. This shows that our proposed method can generate more consistent prediction across frames. The video samples that we provide in the supplemental materials also verify this conclusion. We also present sample qualitative results for VIPER → Cityscapes scenario in Figure 6.3 and 6.4. It can be observed that our method visually achieves better performance than the second best approach DA-VSN (but the best among all existing ones). Although our method does not contain temporal modeling during testing, our predictions in Figure 6.3 (a) still show better temporal consistency than those by DA-VSN. In the spatial level, our segmentation results look also more accurate, *e.g.*, bus in (a), person in (a) and (c), car in (a)-(d).

SYNTHIA → Cityscapes The quantitative comparison results for SYNTHIA → Cityscapes scenario are reported in Table 6.2, where our method still achieves the best performance and surpasses the second best (DA-VSN) by 2.6% mIoU.



(a)

Figure 6.3 Qualitative comparison results on the VIPER → Cityscapes domain adaptive video segmentation task. (a) The first three columns show the predictions of three consecutive frames. *Only one frame has ground truth in each video (30 frames).

Sample qualitative comparison results for this adaptation scenario are shown in Figure 6.5 and 6.6, and our method still achieves more consistent and accurate segmentation results.

6.3.4 Ablation studies

On the framework design To verify the effectiveness of each component of the I2VDA framework, we conduct a comprehensive ablation study with several model variants. The results under the VIPER → Cityscapes scenario are reported in Table 6.3. The variant in the first row serves as the baseline which trains the semantic

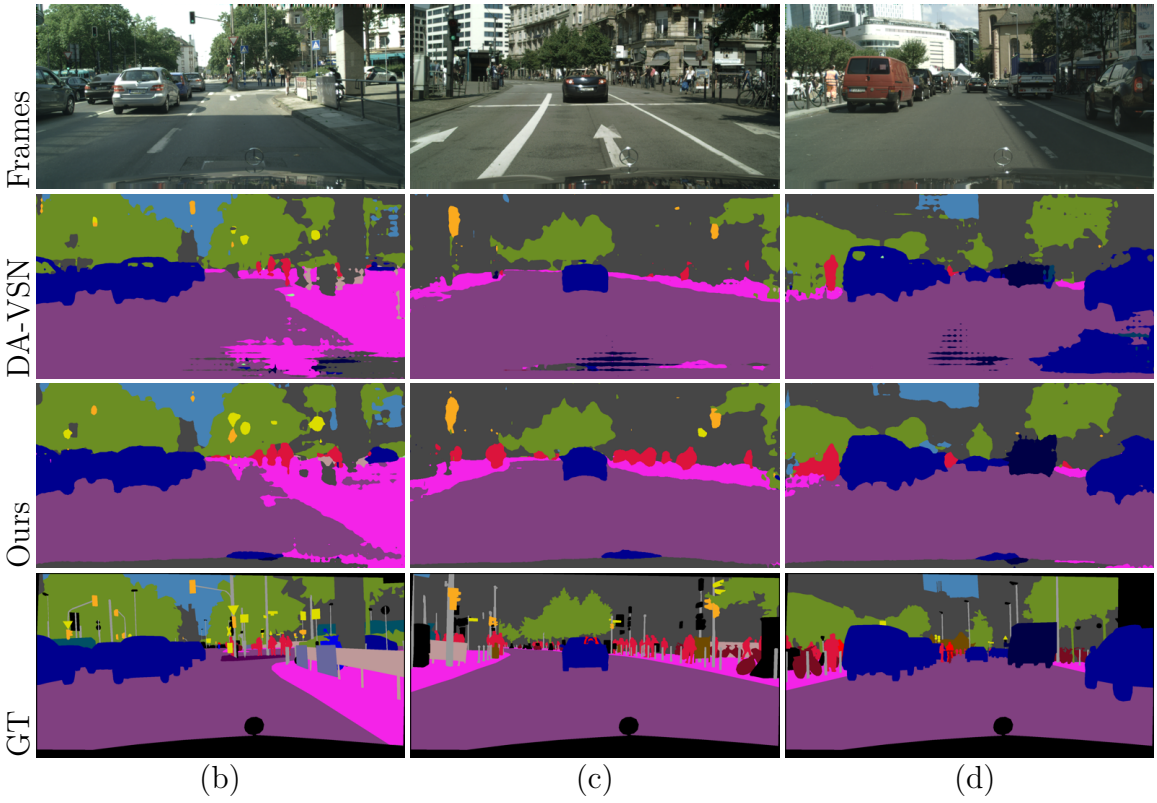


Figure 6.4 Qualitative comparison results on the VIPER → Cityscapes domain adaptive video segmentation task. (b)-(d) show three other independent results from the Cityscapes validation set.

segmentation network with the labeled source domain and employs the adversarial learning to close the domain gap [117], and the last row is the I2VDA method with full settings. We find that the proposed temporal augmentation strategy and temporal consistency learning are both very effective and can achieve 3.6% and 5.0% gains, respectively, over the baseline. Another observation is that the temporal augmentation strategy only obtain 1.6% mIoU gain on its own, but it will play a much greater role (50.6% vs. 44.0%) when combined with the temporal consistency learning. In addition, rows 4-5 show the effectiveness of some designs inside the temporal consistency learning including the consideration of occlusion and entropy.

On the proxy network The proxy network also plays an important role in the

Table 6.2 Quantitative comparison results on the SYNTHIA → Cityscapes domain adaptive video segmentation task.

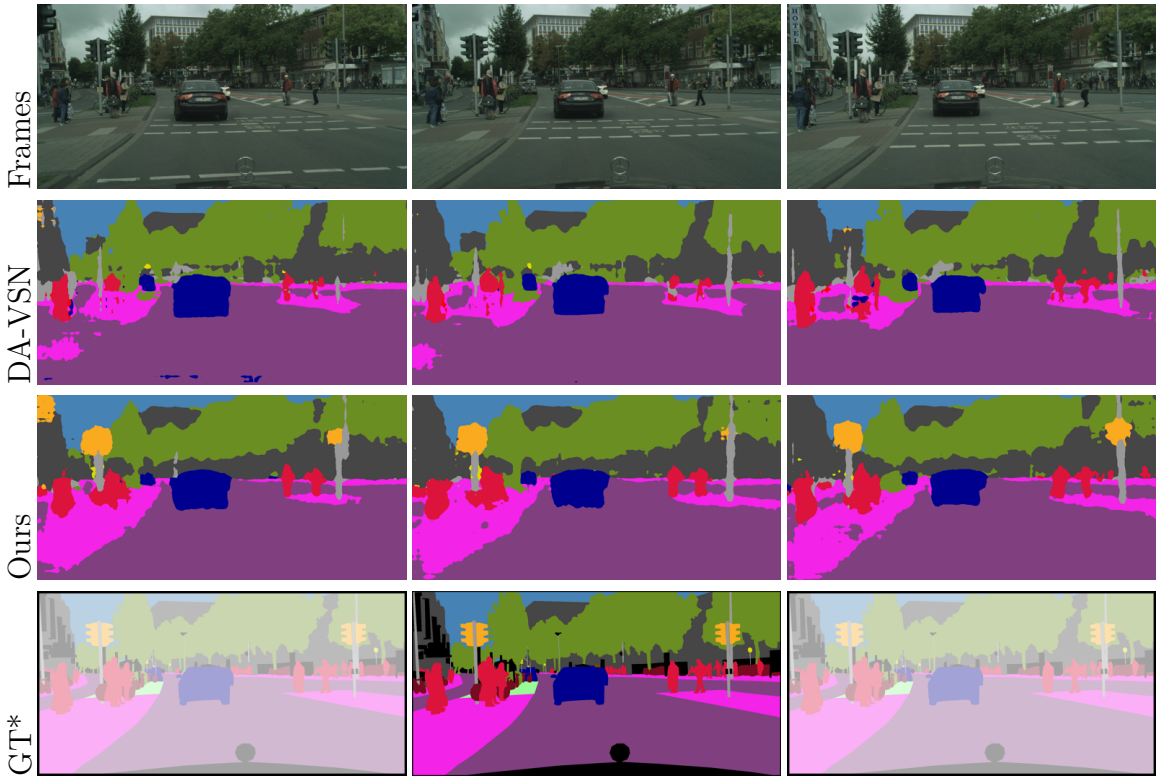
Methods	road	sidewalk	building	pole	traffic light	traffic sign	vegetation	sky	person	rider	car	**m**IoU (%)
Source only	56.3	26.6	75.6	25.5	5.7	15.6	71.0	58.5	41.7	17.1	27.9	38.3
AdvEnt [119]	85.7	21.3	70.9	21.8	4.8	15.3	59.5	62.4	46.8	16.3	64.6	42.7
CBST [158]	64.1	30.5	78.2	28.9	14.3	21.3	75.8	62.6	46.9	20.2	33.9	43.3
IDA [91]	87.0	23.2	71.3	22.1	4.1	14.9	58.8	67.5	45.2	17.0	73.4	44.0
CRST [157]	70.4	31.4	**79.1**	27.6	11.5	20.7	**78.0**	67.2	**49.5**	17.1	39.6	44.7
FDA [133]	84.1	32.8	67.6	28.1	5.5	20.3	61.1	64.8	43.1	19.0	70.6	45.2
DA-VSN [42]	89.4	31.0	77.4	26.1	9.1	20.4	75.4	74.6	42.9	16.1	82.4	49.5
I2VDA (Ours)	**90.3**	**40.8**	74.9	26.7	**20.3**	**22.0**	72.2	75.2	47.2	**20.3**	**82.6**	**52.1**

Table 6.3 Ablation study on the I2VDA framework designs under the VIPER → Cityscapes scenario.

Variants	mIoU (%)
Baseline	44.0
w/o Temporal Augmentation	47.0
w/o Temporal-augmented Consistency	45.6
w/o Occlusion O in Eq. (6.8)	49.7
w/o Entropy Map E in Eq. (6.8)	49.6
Full I2VDA settings	**50.6**

temporal-augmented consistency learning. We conduct experiments on the choice of copying frequency (ITER_COPY) and the training iterations before launching (ITER_LAUNCH). From Table 6.4, we find that copying every 8,000 iterations and launching the proxy network after 8,000 iterations achieves the best performance. In addition, as shown in Figure 6.7, the use of the proxy network does improve the training stability effectively. The baseline here is the same as the one in Table 6.3.

On some hyper-parameters We also conduct experiments to explore the choice of hyper-parameters involved in the temporal-augmented consistency learning



(a)

Figure 6.5 Qualitative comparison results on the SYNTHIA → Cityscapes domain adaptive video segmentation task. (a) The first three columns show the predictions of three consecutive frames. *Only one frame has ground truth in each video (30 frames).

Table 6.4 Ablation study on ITER_COPY and ITER_LAUNCH for the proxy network under the VIPER → Cityscapes scenario. The ITER_LAUNCH is fixed to 8,000 for the first sub-table and the ITER_LAUNCH is fixed to 8,000 for the second sub-table.

ITER_COPY	1k	8k	15k
mIoU(%)	48.9	**50.6**	49.6

ITER_LAUNCH	1k	8k	15k
mIoU(%)	48.3	**50.6**	50.2

including the threshold η in Eq.(6.7), δ in Eq.(6.8), t in Eq.(6.2) and α in Eq.(6.3). The results are reported in Table 6.5, 6.6, and 6.7 where we find that our method achieves better performance when $\eta = 1.0$, $\delta = 0.3$ and $\alpha = 0.5$ and using randomly generated t.

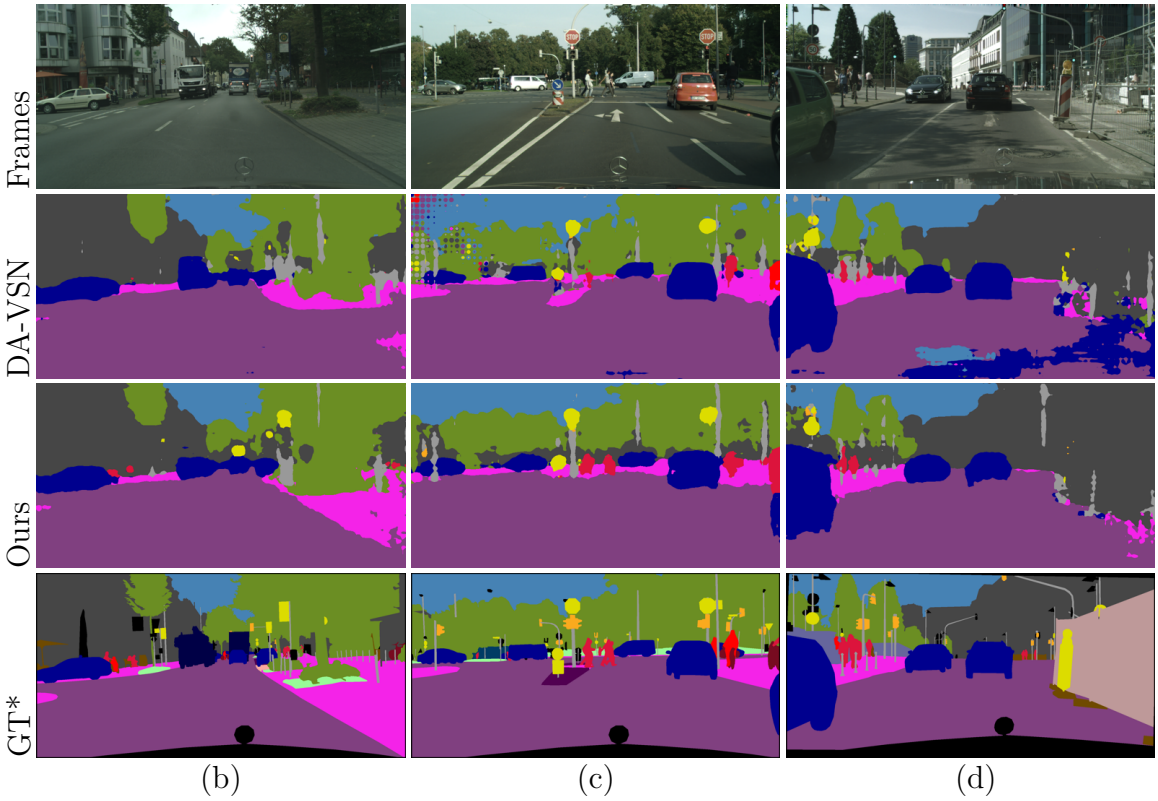


Figure 6.6 Qualitative comparison results on the SYNTHIA → Cityscapes domain adaptive video segmentation task. (b)-(d) show three other independent results from the Cityscapes validation set.

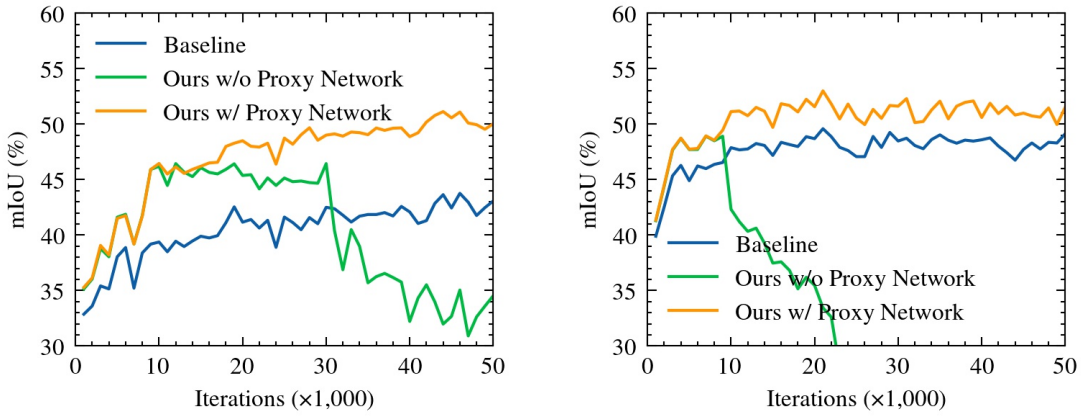
Table 6.5 Ablation study on η in Eq. (6.7) and δ in Eq. (6.8) under the VIPER → Cityscapes scenario. δ is fixed to 0.3 for the first sub-table and η is fixed to 1.0 for the second sub-table.

η	0.1	1.0	2.0
mIoU(%)	50.4	**50.6**	50.2

δ	0.1	0.3	0.5
mIoU(%)	48.5	**50.6**	49.7

6.3.5 Quantitative results

The impact of optical-flow computation for testing. In the main paper, our reported results are based on per-frame inference without optical-flow computation. Here, we further study the impact of optical-flow computation for testing. The comparison results under the VIPER → Cityscapes scenario and the SYN-



(a) VIPER → Cityscapes (b) SYNTHIA → Cityscapes

Figure 6.7 The mIoU performance vs. varying adaptation iterations.

Table 6.6 Ablation study on t in Eq.(6.2).

t	0	0.25	0.5	0.75	1	random
mIoU	47.0	47.0	48.4	48.1	47.4	**50.6**

Table 6.7 Ablation study on α in Eq.(6.3).

α	0.1	0.3	0.5	0.7	0.9
mIoU	49.2	49.7	**50.6**	49.6	48.8

THIA → Cityscapes scenario are shown in Table 6.8 and Table 6.9, respectively. Note that the results are achieved by testing with the best model after 90k training iterations[4]. To be specific, the "I2VDA (two-frame)" is implemented by a non-parametric fusion defined as:

$$P = \mathcal{M}(I_f) + \gamma \mathcal{M}(\mathcal{W}(I_{f-1}, F)), \tag{6.13}$$

where $\mathcal{W}(\cdot,\cdot)$ is the warping function and $\mathcal{M}$ is the segmentation network that have been defined in the main paper, I_f is the current frame, I_{f-1} represents its previous

[4]The results are different from those from the main paper which are based on 50k training iterations.

frame, F is the optical flow between I_{f-1} and I_f, and γ is set to 0.5 to balance the fusion. We observe that the improvements for both scenarios are not very obvious by further using two frames and computing optical flow for testing. These results, to some extent, indicate that our proposed temporal augmentation strategy is effective to help learn diverse temporal patterns during training thus there is no need to explicitly consider the temporal consistency during testing.

Table 6.8 Ablation study on the impact of optical-flow computation for testing under the VIPER → Cityscapes scenario.

Methods	road	sidewalk	building	fence	traffic light	traffic sign	vegetation	terrain
single	84.78	36.09	84.02	28.02	36.46	36.02	85.89	32.48
two-frame	85.13	36.66	84.11	26.36	36.18	35.88	86.10	33.13

Methods	sky	person	car	truck	bus	motorcycle	bicycle	mIoU (%)
single	73.97	63.18	81.87	33.02	51.75	39.94	0.17	51.18
two-frame	74.29	63.33	82.12	32.80	52.38	39.42	0.22	51.21

Table 6.9 Ablation study on the impact of optical-flow computation for testing under the SYNTHIA → Cityscapes scenario.

Methods	road	sidewalk	building	pole	traffic light	traffic sign	vegetation	sky	person	rider	car	mIoU (%)
single	89.88	40.54	77.58	27.26	18.69	23.60	76.07	76.34	48.48	22.39	82.13	53.00
two-frame	89.95	40.74	77.58	27.53	17.92	23.12	76.22	76.60	48.47	22.42	82.20	53.00

6.4 Limitation and discussion

An accurate pretrained optical flow estimator (*e.g.*, FlowNet [55]) is the cornerstone of our method. The estimated flow is employed to synthesize the intermediate frames for temporal augmentation and to warp the prediction for ensuring target temporal consistency as well. Since the flow estimator is trained on additional synthetic datasets such as Sintel [4], employing the trained model weights in the target domain for optical flow estimation also suffers from a domain gap. An unsatisfactory optical flow estimation will mislead the training of the domain adaptive semantic segmentation which is a limitation of our method. Pretraining the flow estimator directly on the target domain (*e.g.*, Cityscapes) in an unsupervised manner might be a good future direction to bridge the domain gap caused by optical flow estimation.

6.5 Conclusion

In this paper, we found that it is not necessary to transfer temporal knowlege for domain-adaptive video semantic segmentation and have introduced for the first time the setting of image-to-video domain adaptive semantic segmentation which transfers knowledge from simulated images to real-world videos. Our I2VDA method reduces the domain gap between the source and target via adversarial training on only spatial knowledge. On the other hand, our method enhances the temporal consistency learning in the target domain by performing the temporal augmentation via frame interpolation to explore more temporal patterns and leveraging the proxy network to provide the pseudo labels on-the-fly to improve the stability of adversarial training. Experimental results on two synthetic-to-real scenarios showed that our method can outperform existing state-of-the-art video-to-video domain adaptation methods.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

To conclude, this dissertation aims to overcome the challenges in data collection and large domain gaps by exploring three different settings for cross-domain semantic segmentation conditioned on different target domains including 1) using one or few unlabeled target images; 2) using multiple unlabeled target images; 3) using unlabeled videos as target, respectively. In the first work, we achieve domain adaptation on a target domain by only using one unlabeled target sample. To tackle this problem, we integrate several style-mixing layers into the segmentor which play the role of a style-transfer module to stylize the source images without introducing any learned parameters. Moreover, we propose a patchwise prototypical matching (PPM) method to weighted consider the importance of source pixels during the supervised training to relieve the negative adaptation.

Then in the second work, our proposed method employs an adversarial training with a labeled daytime dataset and an unlabeled dataset that contains coarsely aligned day-night image pairs. Specifically, for the unlabeled day-night image pairs, we use the pixel-level predictions of static object categories on a daytime image as pseudo supervision to segment its counterpart nighttime image. We further design a re-weighting strategy to handle the inaccuracy caused by a misalignment between day-night image pairs and wrong predictions of daytime images, as well as boost the prediction accuracy of small objects. The proposed DANNet is the first one-stage adaptation framework for nighttime semantic segmentation, which does not train additional day-night image transfer models as a separate pre-processing stage.

Finally, we introduce an image-to-video domain adaptation method to segment videos without using labels. Via this research, we found that it is not necessary to transfer temporal knowledge for domain-adaptive video semantic segmentation and have introduced the setting of image-to-video domain adaptive semantic segmentation which transfers knowledge from simulated images to real-world videos. Our I2VDA

method reduces the domain gap between the source and target via adversarial training on only spatial knowledge. On the other hand, our method enhances the temporal consistency learning in the target domain by performing the temporal augmentation via frame interpolation to explore more temporal patterns and leveraging the proxy network to provide the pseudo labels on the fly to improve the stability of adversarial training. Experimental results on two synthetic-to-real scenarios showed that our method can outperform existing state-of-the-art video-to-video domain adaptation methods.

7.2 Future works

Based on the studies presented in the dissertation, there are a few directions/challenges that can be explored as shown below.

7.2.1 Advanced Segmentation Networks

Based on the literature, we can find that large progress is always made by employing a stronger backbone. Currently, ResNet [45] is the major backbone for UDA. But it has been proved by a recent work [50] that using transformers as backbone [28, 128] achieves more performance gains than ResNet. Therefore, it is worth exploring the strong backbone of UDA.

7.2.2 Negative Transfer

Although domain adaptation helps improve the mIoU performance for all the classes of the target, you can find the IoU of a certain class is even worse than without the adaptation. This phenomenon is called negative transfer – the performance of a certain class is destroyed by domain adaptation. With a carefully designed network and losses, we may better selectively perform the domain adaptation on a certain class to relieve this problem.

7.2.3 Efficient UDA

Most existing UDA approaches focus on the accuracy of the target domain. Except for the accuracy, efficiency is also important. A light-weighted model can speed up the offline domain adaptation process as well as the online prediction. Therefore, effective and efficient cross-domain networks are also highly desirable in the future.

7.2.4 Image Restoration and Enhancement

Image restoration and enhancement methods might help improve the quality of the images and reduce the domain gap. For example, developing a low-light image enhancement model may help improve the visibility of the nighttime images. Moreover, for the foggy/rainy images, an effective image de-hazing/de-raining module might be incorporated to help bridge the gap to the normal domain.

Bibliography

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pp. 2481–2495.

[2] Sagie Benaim and Lior Wolf. "One-Shot Unsupervised Cross Domain Translation". In: *Advances in Neural Information Processing Systems*. 2018.

[3] Léon Bottou. "Large-scale machine learning with stochastic gradient descent". In: *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[4] Daniel J Butler et al. "A naturalistic open source movie for optical flow evaluation". In: *European Conference on Computer Vision (ECCV)*. Springer. 2012, pp. 611–625.

[5] Huiwen Chang et al. "Pairedcyclegan: Asymmetric style transfer for applying and removing makeup". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 40–48.

[6] Wei-Lun Chang et al. "All about structure: Adapting structural information across domains for boosting semantic segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 1900–1909.

[7] Kumar Chellapilla, Sidd Puri, and Patrice Simard. "High performance convolutional neural networks for document processing". In: *Tenth international workshop on frontiers in handwriting recognition*. Suvisoft. 2006.

[8] Chen Chen et al. "Learning to see in the dark". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 3291–3300.

[9] Yi-Hsin Chen et al. "No more discrimination: Cross city adaptation of road scene segmenters". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017, pp. 1992–2001.

[10] Liang-Chieh Chen et al. “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2017), pp. 834–848.

[11] Liang-Chieh Chen et al. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. In: *European Conference on Computer Vision (ECCV)*. 2018.

[12] Liang-Chieh Chen et al. “Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 695–714.

[13] Liang-Chieh Chen et al. “Rethinking atrous convolution for semantic image segmentation”. In: *arXiv preprint arXiv:1706.05587* (2017).

[14] Liang-Chieh Chen et al. “Semantic image segmentation with deep convolutional nets and fully connected crfs”. In: (2015).

[15] Minghao Chen, Hongyang Xue, and Deng Cai. “Domain adaptation for semantic segmentation with maximum squares loss”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 2090–2099.

[16] Yun-Chun Chen et al. “Crdoco: Pixel-level domain transfer with cross-domain consistency”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 1791–1800.

[17] François Chollet. “Xception: Deep learning with depthwise separable convolutions”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1251–1258.

[18] Dan Claudiu Ciresan et al. “Flexible, high performance convolutional neural networks for image classification”. In: *Twenty-second international joint conference on artificial intelligence*. 2011.

[19] Wikimedia Commons. *LeNet and AlexNet*. 2022. URL: `https://commons.wikimedia.org/w/index.php?title=File:Comparison_image_neural_networks.svg&oldid=645651867`.

[20] Wikimedia Commons. *VGG16*. 2022. URL: `https://commons.wikimedia.org/w/index.php?title=File:VGG16.png&oldid=639031035`.

[21] Marius Cordts et al. “The cityscapes dataset for semantic urban scene understanding”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3213–3223.

[22] Camille Couprie et al. “Indoor semantic segmentation using depth information”. In: *arXiv preprint arXiv:1301.3572* (2013).

[23] Dengxin Dai and Luc Van Gool. “Dark model adaptation: Semantic image segmentation from daytime to nighttime”. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2018, pp. 3819–3824.

[24] Jifeng Dai, Kaiming He, and Jian Sun. “Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2015, pp. 1635–1643.

[25] Jifeng Dai et al. “Deformable convolutional networks”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017, pp. 764–773.

[26] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Ieee. 2009, pp. 248–255.

[27] Shuai Di et al. “Rainy Night Scene Understanding With Near Scene Semantic Adaptation”. In: *IEEE Transactions on Intelligent Transportation Systems* (2020).

[28] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.

[29] Qi Dou et al. “Domain Generalization via Model-Agnostic Learning of Semantic Features”. In: *Advances in Neural Information Processing Systems*. 2019.

[30] Kheirie Elhariri. *Convolutional Neural Networks for Image Classification*. 2022. URL: https://medium.com/analytics-vidhya/convolutional-neural-networks-for-image-classification-c403159e81af.

[31] Mark Everingham et al. “The pascal visual object classes (voc) challenge”. In: *International Journal of Computer Vision* 88.2 (2010), pp. 303–338.

[32] Clement Farabet et al. “Learning hierarchical features for scene labeling”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2012), pp. 1915–1929.

[33] Mohsen Fayyaz et al. “STFCN: spatio-temporal FCN for semantic video segmentation”. In: *arXiv preprint arXiv:1608.05971* (2016).

[34] Jun Fu et al. “Dual attention network for scene segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 3146–3154.

[35] Kunihiko Fukushima and Sei Miyake. “Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition”. In: *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

[36] Raghudeep Gadde, Varun Jampani, and Peter V Gehler. “Semantic video cnns through representation warping”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017, pp. 4453–4462.

[37] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. “A neural algorithm of artistic style”. In: *arXiv preprint arXiv:1508.06576* (2015).

[38] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. “Unsupervised monocular depth estimation with left-right consistency”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 270–279.

[39] Boqing Gong, Kristen Grauman, and Fei Sha. “Reshaping visual datasets for domain adaptation”. In: *Advances in Neural Information Processing Systems* 26 (2013), pp. 1286–1294.

[40] Ke Gong et al. “Instance-level human parsing via part grouping network”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 770–785.

[41] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in Neural Information Processing Systems* 27 (2014).

[42] Dayan Guan et al. “Domain Adaptive Video Segmentation via Temporal Consistency Regularization”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 8053–8064.

[43] Chunle Guo Guo et al. “Zero-reference deep curve estimation for low-light image enhancement”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 1780–1789.

[44] Xiaoqing Guo et al. “MetaCorrection: Domain-aware Meta Loss Correction for Unsupervised Domain Adaptation in Semantic Segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 3927–3936.

[45] Kaiming He et al. “Deep residual learning for image recognition”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[46] Kaiming He et al. “Spatial pyramid pooling in deep convolutional networks for visual recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.9 (2015), pp. 1904–1916.

[47] Judy Hoffman et al. “Cycada: Cycle-consistent adversarial domain adaptation”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1989–1998.

[48] Judy Hoffman et al. “Fcns in the wild: Pixel-level adversarial and constraint-based adaptation”. In: *arXiv preprint arXiv:1612.02649* (2016).

[49] Seunghoon Hong, Hyeonwoo Noh, and Bohyung Han. “Decoupled deep neural network for semi-supervised semantic segmentation”. In: *Advances in Neural Information Processing Systems* 28 (2015).

[50] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. “DAFormer: Improving Network Architectures and Training Strategies for Domain-Adaptive Semantic Segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

[51] Gao Huang et al. “Densely connected convolutional networks”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 4700–4708.

[52] Jiaxing Huang et al. “FSDR: Frequency Space Domain Randomization for Domain Generalization”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[53] Xun Huang and Serge Belongie. “Arbitrary style transfer in real-time with adaptive instance normalization”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017, pp. 1501–1510.

[54] Zilong Huang et al. “Ccnet: Criss-cross attention for semantic segmentation”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 603–612.

[55] Eddy Ilg et al. “Flownet 2.0: Evolution of optical flow estimation with deep networks”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2462–2470.

[56] Tomas Jenicek and Ondrej Chum. “No Fear of the Dark: Image Retrieval under Varying Illumination Conditions”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 9696–9704.

[57] Huaizu Jiang et al. “Super slomo: High quality estimation of multiple intermediate frames for video interpolation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 9000–9008.

[58] Anna Khoreva et al. “Simple does it: Weakly supervised instance and semantic segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 876–885.

[59] Myeongjin Kim and Hyeran Byun. “Learning texture invariant representation for domain adaptation of semantic segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 12975–12984.

[60] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *International Conference on Learning Representations* (2014).

[61] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in Neural Information Processing Systems* 25 (2012).

[62] L’ubor Ladickỳ et al. “Associative hierarchical crfs for object class image segmentation”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE. 2009, pp. 739–746.

[63] John Lafferty, Andrew McCallum, and Fernando CN Pereira. “Conditional random fields: Probabilistic models for segmenting and labeling sequence data”. In: (2001).

[64] Yann LeCun et al. “Backpropagation applied to handwritten zip code recognition”. In: *Neural computation* 1.4 (1989), pp. 541–551.

[65] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[66] Yann LeCun et al. “Handwritten digit recognition with a back-propagation network”. In: *Advances in Neural Information Processing Systems* 2 (1989).

[67] Guangrui Li et al. “Content-consistent matching for domain adaptive semantic segmentation”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 440–456.

[68] Yule Li, Jianping Shi, and Dahua Lin. "Low-latency video semantic segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 5997–6005.

[69] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. "Bidirectional learning for domain adaptation of semantic segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 6936–6945.

[70] Qing Lian et al. "Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 6758–6767.

[71] G. Lin et al. "RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[72] Tsung-Yi Lin et al. "Focal loss for dense object detection". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017, pp. 2980–2988.

[73] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European Conference on Computer Vision (ECCV)*. Springer. 2014, pp. 740–755.

[74] Chenxi Liu et al. "Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 82–92.

[75] Si Liu et al. "Surveillance video parsing with single frame supervision". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 413–421.

[76] Yifan Liu et al. "Efficient semantic video segmentation with per-frame inference". In: *European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 352–368.

[77] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3431–3440.

[78] Aurelien Lucchi et al. "Are spatial and global constraints really necessary for segmentation?" In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE. 2011, pp. 9–16.

[79] Yawei Luo et al. "Adversarial Style Mining for One-Shot Unsupervised Domain Adaptation". In: *Advances in Neural Information Processing Systems*. 2020.

[80] Yawei Luo et al. “Significance-aware information bottleneck for domain adaptive semantic segmentation”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 6778–6787.

[81] Yawei Luo et al. “Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 2507–2516.

[82] K-K Maninis et al. “Video object segmentation without temporal information”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.6 (2018), pp. 1515–1530.

[83] Xudong Mao et al. “Least squares generative adversarial networks”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2794–2802.

[84] Ke Mei et al. “Instance adaptive self-training for unsupervised domain adaptation”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 415–430.

[85] Ondrej Miksik et al. “Efficient temporal consistency for streaming video scene analysis”. In: *International Conference on Robotics and Automation*. IEEE. 2013, pp. 133–139.

[86] Adil Moujahid. *A Practical Introduction to Deep Learning with Caffe and Python*. 2022. URL: `http://adilmoujahid.com/posts/2016/06/introduction-deep-learning-python-caffe/`.

[87] Siva Karthik Mustikovela, Michael Ying Yang, and Carsten Rother. “Can ground truth label propagation from video help semantic segmentation?” In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 804–820.

[88] Sauradip Nag, Saptakatha Adak, and Sukhendu Das. “What’s There in the Dark”. In: *IEEE International Conference on Image Processing*. IEEE. 2019, pp. 2996–3000.

[89] David Nilsson and Cristian Sminchisescu. “Semantic video segmentation by gated recurrent flow propagation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 6819–6828.

[90] Feng Ning et al. “Toward automatic phenotyping of developing embryos from videos”. In: *IEEE Transactions on Image Processing* 14.9 (2005), pp. 1360–1371.

[91] Fei Pan et al. “Unsupervised intra-domain adaptation for semantic segmentation through self-supervision”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 3764–3773.

[92] Xingang Pan et al. “Two at once: Enhancing learning and generalization capacities via ibn-net”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 464–479.

[93] George Papandreou et al. “Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2015, pp. 1742–1750.

[94] Adam Paszke et al. “Enet: A deep neural network architecture for real-time semantic segmentation”. In: *arXiv preprint arXiv:1606.02147* (2016).

[95] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. “Constrained convolutional neural networks for weakly supervised segmentation”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2015, pp. 1796–1804.

[96] Christian S Perone et al. “Unsupervised domain adaptation for medical imaging segmentation with self-ensembling”. In: *NeuroImage* 194 (2019), pp. 1–11.

[97] Pedro Pinheiro and Ronan Collobert. “Recurrent convolutional neural networks for scene labeling”. In: *International Conference on Machine Learning*. PMLR. 2014, pp. 82–90.

[98] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks”. In: *International Conference on Learning Representations* (2015).

[99] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. “Playing for benchmarks”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017, pp. 2213–2222.

[100] Stephan R Richter et al. “Playing for data: Ground truth from computer games”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 102–118.

[101] Eduardo Romera et al. “Bridging the day and night domain gap for semantic segmentation”. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2019, pp. 1312–1318.

[102] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on*

Medical image computing and computer-assisted intervention. Springer. 2015, pp. 234–241.

[103] German Ros et al. "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3234–3243.

[104] Leonid I Rudin, Stanley Osher, and Emad Fatemi. "Nonlinear total variation based noise removal algorithms". In: *Physica D: nonlinear phenomena* 60.1-4 (1992), pp. 259–268.

[105] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[106] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. "Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 7374–7383.

[107] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. "ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 10765–10775.

[108] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. "Map-Guided Curriculum Domain Adaptation and Uncertainty-Aware Evaluation for Semantic Nighttime Image Segmentation". In: *arXiv preprint arXiv:2005.14553* (2020).

[109] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. "Semantic foggy scene understanding with synthetic data". In: *International Journal of Computer Vision* 126.9 (2018), pp. 973–992.

[110] Swami Sankaranarayanan et al. "Learning from synthetic data: Addressing domain shift for semantic segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 3752–3761.

[111] Evan Shelhamer et al. "Clockwork convnets for video semantic segmentation". In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 852–868.

[112] Inkyu Shin et al. "Unsupervised Domain Adaptation for Video Semantic Segmentation". In: *arXiv preprint arXiv:2107.11052* (2021).

[113] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *International Conference on Learning Representations*. 2015.

[114] Lei Sun et al. “See clearer at night: towards robust nighttime semantic segmentation through day-night image conversion”. In: *Artificial Intelligence and Machine Learning in Defense Applications*. Vol. 11169. International Society for Optics and Photonics. 2019, 111690A.

[115] Christian Szegedy et al. “Going deeper with convolutions”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1–9.

[116] Josh Tobin et al. “Domain randomization for transferring deep neural networks from simulation to the real world”. In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2017, pp. 23–30.

[117] Yi-Hsuan Tsai et al. “Learning to adapt structured output space for semantic segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 7472–7481.

[118] Johan Vertens, Jannik Zürn, and Wolfram Burgard. “HeatNet: Bridging the Day-Night Domain Gap in Semantic Segmentation with Thermal Images”. In: *arXiv preprint arXiv:2003.04645* (2020).

[119] Tuan-Hung Vu et al. “Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 2517–2526.

[120] Kaixin Wang et al. “Panet: Few-shot image semantic segmentation with prototype alignment”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 9197–9206.

[121] Panqu Wang et al. “Understanding convolution for semantic segmentation”. In: *2018 IEEE winter conference on applications of computer vision (WACV)*. Ieee. 2018, pp. 1451–1460.

[122] Ting-Chun Wang et al. “High-resolution image synthesis and semantic manipulation with conditional gans”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 8798–8807.

[123] Zhonghao Wang et al. “Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 12635–12644.

[124] Zhou Wang et al. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612.

[125] Juyang Weng, Narendra Ahuja, and Thomas S Huang. “Cresceptron: a self-organizing neural network which grows adaptively”. In: *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*. Vol. 1. IEEE. 1992, pp. 576–581.

[126] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. “Real-time semantic image segmentation via spatial sparsity”. In: *arXiv preprint arXiv:1712.00213* (2017).

[127] Zuxuan Wu et al. “Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 518–534.

[128] Enze Xie et al. “SegFormer: Simple and efficient design for semantic segmentation with transformers”. In: *Advances in Neural Information Processing Systems* 34 (2021).

[129] Saining Xie et al. “Aggregated residual transformations for deep neural networks”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1492–1500.

[130] Yu-Syuan Xu et al. “Dynamic video segmentation network”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 6556–6565.

[131] Tianyang Xu et al. “Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking”. In: *IEEE Transactions on Image Processing* 28.11 (2019), pp. 5596–5609.

[132] Han Yang et al. “Towards photo-realistic virtual try-on by adaptively generating-preserving image content”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 7850–7859.

[133] Yanchao Yang and Stefano Soatto. “Fda: Fourier domain adaptation for semantic segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 4085–4095.

[134] Changqian Yu et al. “Bisenet: Bilateral segmentation network for real-time semantic segmentation”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 325–341.

[135] Changqian Yu et al. “Learning a discriminative feature network for semantic segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 1857–1866.

[136] Fisher Yu and Vladlen Koltun. “Multi-Scale Context Aggregation by Dilated Convolutions”. In: *ICLR*. May 2016.

[137] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. “Dilated residual networks”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 472–480.

[138] Yuhui Yuan and Jingdong Wang. “Ocnet: Object context network for scene parsing”. In: *arXiv preprint arXiv:1809.00916* (2018).

[139] Xiangyu Yue et al. “Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 2100–2110.

[140] Sergey Zagoruyko and Nikos Komodakis. “Wide Residual Networks”. In: *British Machine Vision Conference*. 2016.

[141] Matthew D Zeiler and Rob Fergus. “Visualizing and understanding convolutional networks”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2014, pp. 818–833.

[142] Oliver Zendel et al. “How good is my test data? Introducing safety analysis for computer vision”. In: *International Journal of Computer Vision* 125.1-3 (2017), pp. 95–109.

[143] Kai Zhang et al. “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising”. In: *IEEE Transactions on Image Processing* 26.7 (2017), pp. 3142–3155.

[144] Pan Zhang et al. “Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 12414–12424.

[145] Qiming Zhang et al. “Category Anchor-Guided Unsupervised Domain Adaptation for Semantic Segmentation”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 433–443.

[146] Yang Zhang, Philip David, and Boqing Gong. “Curriculum domain adaptation for semantic segmentation of urban scenes”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017, pp. 2020–2030.

[147] Hengshuang Zhao et al. “Icnet for real-time semantic segmentation on high-resolution images”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 405–420.

[148] Hengshuang Zhao et al. “Pyramid scene parsing network”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2881–2890.

[149] Zhedong Zheng and Yi Yang. “Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation”. In: *International Journal of Computer Vision* 129.4 (2021), pp. 1106–1120.

[150] Kaiyang Zhou et al. “Domain Generalization with MixStyle”. In: *International Conference on Learning Representations*. 2021.

[151] Tinghui Zhou et al. “View synthesis by appearance flow”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 286–301.

[152] Jun-Yan Zhu et al. “Unpaired image-to-image translation using cycle-consistent adversarial networks”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017, pp. 2223–2232.

[153] Xinge Zhu et al. “Penalizing top performers: Conservative loss for semantic segmentation adaptation”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 568–583.

[154] Xizhou Zhu et al. “Deep feature flow for video recognition”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2349–2358.

[155] Yi Zhu et al. “Improving semantic segmentation via video propagation and label relaxation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 8856–8865.

[156] Barret Zoph and Quoc V. Le. “Neural Architecture Search with Reinforcement Learning”. In: *International Conference on Learning Representations*. 2017.

[157] Yang Zou et al. “Confidence regularized self-training”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 5982–5991.

[158] Yang Zou et al. “Unsupervised domain adaptation for semantic segmentation via class-balanced self-training”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 289–305.