Real-Time Domain Adaptation in Semantic Segmentation

Leonardo Dardanello
Politecnico di Torino
Turin, Italy
s319060@studenti.polito.it

Lorenzo Fezza
Politecnico di Torino
Turin, Italy
s312689@studenti.polito.it

Paola Matassa
Politecnico di Torino
Turin, Italy
s317632@studenti.polito.it

Abstract

Semantic Segmentation is a Computer Vision task that finds its application in various fields such as: autonomous driving, medical analysis and robot navigation. It requires a large amount of pixel-wise annotations that are usually obtained through a very time-consuming human effort. In this paper we will discuss the topic of Domain Adaptation applied to Real-Time Semantic Segmentation,in particular we will employ a STDC-Seg first trained on source data and then used as a Generator in an Adversarial Domain Adaptation framework, furthermore we will exploit Data Augmentation to improve the performance of the setup. Next, we will adopt FDA, an image-to-image translation technique, to help cross the gap between the source domain and the target domain. Lastly, a Self-Supervised Learning(SSL) framework, that exploits pseudo-labels generated from source data via FDA, was implemented as further optimization. The results obtained show that the domain shift between the source data and the target data can be partially covered although additional work needs to be done to completely close it.

Code available at: https://github.com/LeoDardanello/DA_Semantic_Segmentation

1. Introduction

In recent years, Real-Time Semantic Segmentation has emerged as a crucial research area in computer vision, with applications ranging from autonomous driving, medical analysis and robot navigation. Unfortunately, the realisation of pixel-wise annotations necessary to train Deep Learning models requires a significant amount of time from specialised experts. To solve this problem, recent work has focused on the use of synthetic datasets (e.g. SYNTHIA, GTA5 [3]) in order to use computer-generated annotations and automate the process of label creation. Training models on synthetic datasets and testing them on real-word images introduce a Domain Shift between the two domains. During the exposure of our work we will refer to the domain of the synthetic images as "source domain" and as "target domain" for the domain of the real-world images. To cross the gap between the domains, Domain Adaptation (DA) techniques are used to train models on source images and subsequently test them on target images. One of the simplest and more common ways to reduce the Domain Shift is to perform Data Augmentation; changing some of the features (e.g light conditions, colors, zoom,etc..) of the source images can increase the variety and thus expose the model to a wider range of samples making it more robust. Other ways require to configure an Adversarial Domain Adaptation setup with the goal to map the features of the source domain as close as possible to the features of target domain, moreover this framework can be combined with Data Augmentation to further boost the results. Style Transfer via image-to-image translation is another option that usually does not require parameters to reduce the discrepancy between the Domains. It works by transforming the visual appearance of the source image to make it match the one of the target domain. Self-Supervised Learning (SSL) makes use of pseudo-labels obtained with a trained model to improve the performance of the DA framework by refining the alignment across the source domain and the target domain.

In this paper we propose:

- an implementation of a Real-Time Semantic Segmentation network, STDC-Seg [2] that uses a Short-Term Dense Concatenation (STDC) module that allows to extract deep features with a scalable receptive field.
- Three different types of transformation for Data Augmentation
- An implementation of an Unsupervised Adversarial Domain Adaptation framework.
- An image-to-image Fourier Domain Adaptation (FDA) based Style Transfer with three different values of beta (0.01, 0.05, 0.09) and Multi-Band Transfer (MBT) [5].
- A Self-Supervised Learning framework that uses pseudo-label to fine-tune the model trained with FDA.

2. Related Works

2.1. Real-time Semantic Segmentation.

The model design is crucial in tasks related to computer vision, especially when it comes to Real-Time tasks. Traditional models used to perform this task employ Fully Convolutional Networks that produce very significant results. This performances, however, entails a large computational cost and complicated network connections. The recent growth of Real-Time applications has led to the development of networks that implement several features to speed up the classification process. Some of these features include the use of lighter backbones and a multi-path architecture, an example of a network that implements these improvement is the BiSeNet introduced by Yu et al. [6]. Fan et al. [2] pointed out some of the problems that the BiSeNet had in terms of computational cost and used it as a starting point to develop the STDC-Seg network, a faster and lighter model that is well-suited for Real-Time Semantic Segmentation tasks.

2.2. Domain Adaptation.

Domain Adaptation tries to solve the problem of the Domain shift. Since large amount of pixel-level annotations can't be easily obtained, several techniques that attempt to transfer the knowledge across the domains have been developed [7]. Among those method, some of those exploits an Adversarial Training framework. This setup employs two different networks: a Generator and a Discriminator. The Generator predicts the semantic labels and tries to fool the Discriminator, while the discriminator tries to identify which domain the input images come from. Tsai et al. [4] developed a setup to perform Unsupervised Domain Adaptation on multiple levels. Additionally to Adversarial Training models, image-to-image translation can also be useful to bridge the Domain Shift with the benefit of not needing any parameter. Yang et al. [5] proposed an image-to-image translation based on Fourier Transform with the goal of aligning the domains via a Spectral Transfer of the low-end frequencies from the target image to the source image.

3. Method

3.1. STDC-Seg

The core network we will employ is a revised version of the traditional BiSeNet model: STDC-Seg. BiSeNet models adopts a multi-path architecture with two path: a Context Path [6] to learn high-level semantics and a Spatial Path [6] to extract low-level features. STDC-Seg uses a single-path architecture dropping the Spatial Path and deploying a Short-Term Dense Concatenation (STDC) Network as backbone for the Context Path; the learning of spatial information is managed by a custom-designed Detail Aggregation module [2]. The observations just described are shown in Figure 1.

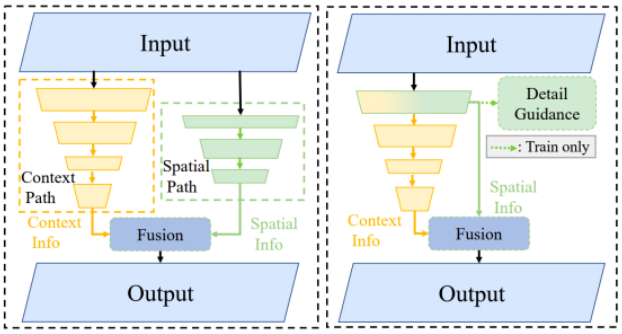


Figure 1. On the left the BiSeNet architecture, on the right the STDC-Seg architecture. Image taken from [2]

The STDC network is made of 6 stages as shown in 2 and its main component is the STDC module. Each module is composed of several blocks, following the work of Fan et al. [2] we will use 4 blocks for each STDC module. For every block a ConvX operation is performed: it incorporates a Convolutional layer, a Batch-Normalization layer and a ReLU activation layer. It can be expressed as:

$$x_i = ConvX_i(x_{i-1}, k_i) \tag{1}$$

where x_i stands for the *i-th* block, x_{i-1} for the preceding block and k_i for the convolution's kernel size. In particular the kernel size is 1 for the first layer and 3 for the remaining layers. The final output of STDC module is the concatenation, through skip-connection, of the output of each block as can be seen in 2. However, before the concatenation, an average pooling operation is applied on the output of different blocks to match the spatial dimension as shown in 2.

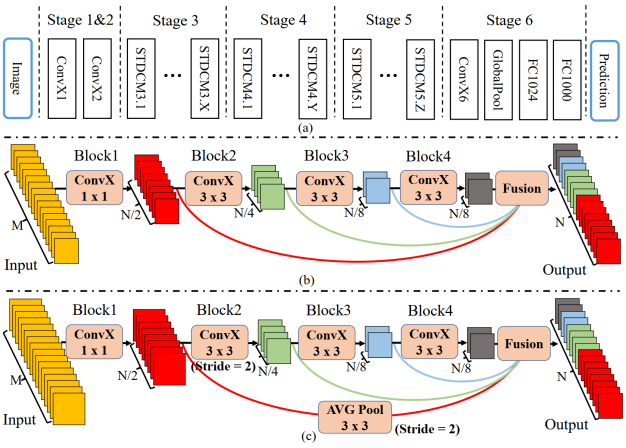


Figure 2. (a) STDC network architecture; (b) STDC module (c) STDC module with stride = 2. Image taken from [2]

Stages 1 and 2 implement a single ConvX block, while

stages 3, 4 and 5 employ multiple STDC modules and require careful fine-tuning to determine the appropriate quantity of said modules. Additionally, only for Stage 3, 4 and 5 the first blocks in the STDC modules down-sample the spatial resolution with a stride of 2.

Alongside the STDC network the other main component of the STDC-Seg is the Detail Aggregation module. It has the task to provide detail guidance (boundaries, corners, etc...) for the low-level layers (Stage 4 and 5) through the generation of the Detail Ground Truth.

As illustrated in figure 3 the generation of the Detail Ground Truth can be modeled as a binary classification task using as input the Segmentation Ground Truth and the Detail Aggregation module [2] (blue dashed box in 3) as trainable model. This module is composed by convolution operation exploiting the Laplacian Kernel [2] and a 1x1 convolution layer. The computed Detail GT is compared against the feature map extracted by the Detail Head [2] from Stage 3 and the loss used to optimize the process is expressed as:

$$\mathcal{L}_{Detail}(p_d, g_d) = \mathcal{L}_{Dice}(p_d, g_d) + \mathcal{L}_{bce}(p_d, g_d) \quad (2)$$

where H and W are the image dimensions, p_d is the predicted detail, g_d is the corresponding ground-truth and $\mathcal{L}_{bce}$ represents the binary cross-entropy loss. $\mathcal{L}_{Dice}$ represents the dice loss and is computed as:

$$\mathcal{L}_{Dice}(p_d, g_d) = 1 - \frac{2\sum_i^{HxW} p_d^i g_d^i + \varepsilon}{\sum_i^{HxW} \left(p_d^i\right)^2 + \sum_i^{HxW} \left(g_d^i\right)^2 + \varepsilon} \quad (3)$$

where ε is a Laplace smoothing factor.

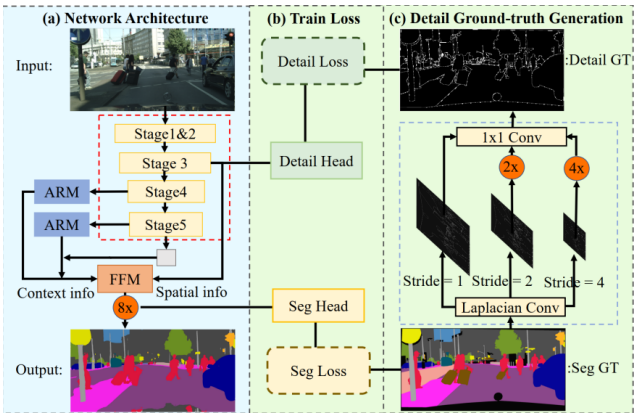


Figure 3. (a) STDC-Seg network architecture; (b) Detail Loss and Segmentation Loss computation; (c) Detail Ground Truth Generation and Detail Aggregation module training; image from [2]

Although STDC-Seg is a revised variation of the BiSeNet model, some of its modules are still employed, like the Attention Refinement Module (ARM) [6] used to refine the features of the last two stages 3 and the Feature Fusion Module (FFM) [6] used to fuse the high-level features from Stage 3 and the low-level features coming from the low-level layers 3.

3.2. Data Augumentation

Data Augmentation works through the generation of synthetic images reflecting the variations and characteristics of the target domain. Using these artifacts the diversity and characterization of the dataset can be increased, providing the model with a more solid basis for learning. Our Data Augmentation implementation consists in a set of three different transformations. For each training epoch each source image has a 50% probability of being subjected to a random chosen transformation among the three adopted.

In particular, the Augmentation performed are :

- *Random Crop :* A random clipping is applied to the image to allow focusing on particular close-up of the image 4a;
- *Horizontal Flip :* An inversion along the horizontal axis is applied to enhance the recognition of scenarios where the elements are arranged symmetrically 4b;
- *Jitter :* Several modification of the visual aspects of the source image (brightness, contrast, saturation and hue) are applied in order to introduce more color variability 4c;

Each transformation result is shown in the figure below.

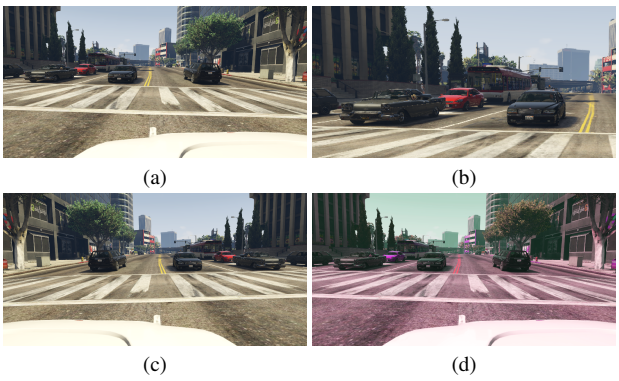


Figure 4. (a) Original image (b) Random Crop (c) Horizontal Flip (d) Jitter

Regarding the labels for the augmented images, for the *Random Crop* and the *Horizontal Flip* augmentation the corresponding label is also subject to the same augmentation, while for the *Jitter* augmentation the label is left untouched.

3.3. Unsupervised Adversarial Domain Adaptation

When trying to overcome the Domain Shift issue between source and target domains, employing an Adversarial Domain Adaptation framework enables the transfer of

learned information across different domains. Tsai et al. [4] proposed a multi-level adversarial setup 5 to effectively perform Domain Adaptation on different feature levels 5.

This framework requires two different components:

- a Generator G, with the objective of classify the image and learn how to produce prediction that are more similar to the target domain;
- a Discriminator, with the task of distinguish the domain of the image received as input;

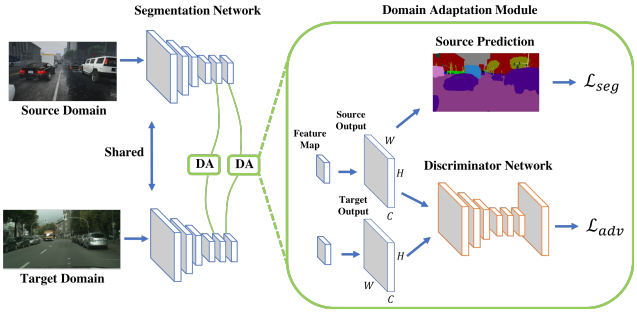


Figure 5. Overview of the Domain Adaptation framework proposed by Tsai et al. [4], image from [4].

First an image X_s from the source domain is passed through the Generator G. Then the segmentation softmax P_t for the target image X_t is predicted. Since the goal of this Domain Adaptation framework is to make the segmentation prediction P_s close to P_t, both of these predictions are passed as input to the Discriminator D. An Adversarial loss $\mathcal{L}_{adv}$ is used to propagate the gradient from D to G, in order to encourage G to generate segmentation distribution closer to the target domain. The joint loss for a single-layer adaptation can be expressed as:

$$\mathcal{L}(X_s, X_t) = \mathcal{L}_{seg}(X_s, Y_s) + \lambda_{adv}\mathcal{L}_{adv}(X_t) \quad (4)$$

where Y_s stands for the source image's label. $\mathcal{L}_{seg}$ represents, for the whole STDC network concatenated output 2, the cross-entropy loss expressed as:

$$\mathcal{L}_{seg}(X_s, Y_s) = -\sum_{h,w}\sum_{c\in C} Y_s^{h,w,c}\log(P_S^{(h,w,c)}) \quad (5)$$

where C denotes the number of semantic classes. $\mathcal{L}_{adv}$ is the adversarial loss weighted by a factor of λ_{adv}. In particular $\mathcal{L}_{adv}$ is formulated as:

$$\mathcal{L}_{adv}(X_t) = -\sum_{h,w}\log(D(P_t)^{(h,w,1)}) \quad (6)$$

This loss is custom-designed to train the Generator into fooling the Discriminator by maximizing the probability of the target prediction being considerate as the source prediction.

Discriminator training is performed using the segmentation softmax output P as input and employing the following binary cross-entropy loss function for the two classes (i.e source and target).

$$\mathcal{L}_d = -\sum_{h,w}(1-z)\log(D(P)^{(h,w,0)}) + z\log(D(P)^{h,w,1}) \quad (7)$$

where $z = 0$ if the sample belong to the target domain and $z = 1$ is the sample is from the source domain.

3.4. Fourier Domain Adaptation

Usually the source and target domain images have differences in visual appearance that can make the Domain Adaptation task more difficult to solve. These discrepancies can be reduced through the implementation of effective style transfer techniques. Fourier Domain Adaptation (FDA) [5] is an image-to-image translation technique that is able to transfer the style from an image to another through the computation of the Fast Fourier Transform (FFT). In particular it replaces the low-level frequencies of the source image with those of the target image. For a single channel image x of height H and width W, its relative FFT can be expressed as:

$$\mathcal{F}(x)(m,n) = \sum_{h,w} x(h,w)e^{-j2\pi\left(\frac{h}{H}m + \frac{w}{W}n\right)}, j^2 = -1 \quad (8)$$

We denote the amplitude component and the phase component of the FFT respectively as $\mathcal{F}^{\mathcal{A}}$ and $\mathcal{F}^{\mathcal{P}}$, both having dimensions $H \times W \times 3$. Furthermore, $\mathcal{F}^{-1}$ denotes the inverse Fourier Transform (iFFT), which inverts the mapping of spectral signals ($\mathcal{F}^{\mathcal{A}}$ and $\mathcal{F}^{\mathcal{P}}$) back into image space. If we assume that the center of the image has coordinates (0,0), we can define M_b as a mask, whose value is zero except for the center region where $\beta \in (0,1)$.

$$M_\beta(h,w) = \mathbb{1}_{(h,w)\in[-\beta H:\beta H, -\beta H:\beta H]} \quad (9)$$

Where β is used for adjusting the size of the spectral neighbourhood to be swapped. Yang et al. [5] uses values of $\beta <= 0.15$ but the choice is task-dependant. Using the definitions just introduced (8, 9) the whole FDA process can be formalized as:

$$x^{s\longrightarrow t} = \mathcal{F}^{-1}\left(\left[M_\beta \circ \mathcal{F}^{\mathcal{A}}(x^t) + (1-M_\beta)\circ\mathcal{F}^{\mathcal{A}}(x^s), \mathcal{F}^{\mathcal{P}}(x^s)\right]\right) \quad (10)$$

where x^s and x^t are two images respectively from the source and the target domain. Below it is reported an example of the FDA process with a model using $\beta = 0.01$.

Figure 6. Example of an image produced by using the Fourier Domain Adaptation process. Top left image from GTA5 [3]; top right image from Cityscapes [1]; on the bottom image is applied the FDA transformation with $\beta = 0.01$

3.5. Self-Supervised Learning

Self-supervised learning (SSL) exploits the inherent patterns of unlabelled data to extract meaningful representations and autonomously learn useful features. This property is particularly useful in tasks where it is not easy to retrieve a large amount of pixel-annotated labels, such as Semantic Segmentation. Therefore, we will exploit this property and generate pseudo-labels for the target domain. Following the approach of Yang et al. [5], we instantiate three different models ϕ_{β_m}, with $m = 1, 2, 3$ trained from scratch using an Adversarial Domain Adaptation setup. For a given target image x^t, the prediction is provided by the mean across the $M = 3$ models:

$$\widehat{y^t} = \arg\max_K \frac{1}{M} \sum_m \phi_{\beta_m}(x^t) \tag{11}$$

where K is the number of semantic classes. In order to have reliable pseudo-label, for each semantic class is applied a threshold: a prediction is accepted only if its confidence is within the top 66% or greater than 0.9, otherwise the class void is assigned.

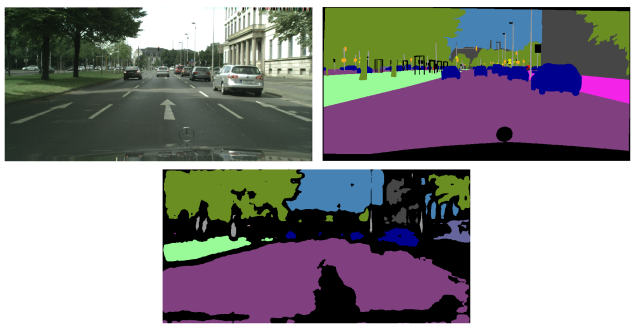


Figure 7. Top left image: Cityscapes [1] original; top right image: Cityscapes [1]label; bottom image: generated pseudo-label

Having obtained the pseudo-label, a SSL framework can be used to fine-tune ϕ_{β_m}. For the fine-tuning we will adopt an Adversarial Domain Adaptation setup, where appropriate changes have been done to the loss function:

$$\mathcal{L}(X_s, X_t, Y_s, \widehat{Y_t}) = \mathcal{L}_{seg}(X_s, Y_s) + \lambda_{adv}\mathcal{L}_{adv}(X_t) + \mathcal{L}_{seg}(X_t, \widehat{Y_t}) \tag{12}$$

where X_s and X_t are respectively the source and the target image, while Y_s and $\widehat{Y_t}$ denotes the source label and the target pseudo-label.

4. Experimental Results

4.1. Datasets

The datasets employed to conduct our analysis and evaluation are **Cityscapes** [1] and **GTA5** [3]. In particular:

- **Cityscapes:** The Cityscapes dataset consists in a collection of 25 000 inner-city street scenes taken from 50 different cities; 5000 of these images are have high quality pixel-level annotations while 20 000 have coarse annotation [1]. The dataset contains 30 classes, grouped by 8 different categories, excluding the more rare classes only 19 were used.
- **GTA5:** it is a synthetic dataset that contains 25 000 images with pixel-level annotation extrapolated with the detouring technique [3] directly from the game. It contains 19 semantic classes compatible with the Cityscapes ones.

Regarding Cityscapes a subset of 1572 images was used, while for GTA5 1875 images were employed. The validation and test folds both consists of 500 images for Cityscapes and 625 images for GTA5.

4.2. Configuration & Baseline

We use STDC-Seg as our main model, where no differently specified, our setting will be the following:

- number of epoch: 50
- optimizer: SGD (Stochastic Gradient Setup) with momentum 0.9 and weight decay $1e^{-4}$
- batch size: 8
- initial learing rate of 0.001 with a polynomial decrease expressed by: $(\frac{1-iter}{maxiter})^{power}$ where $power = 0.9$
- image cropping: (1024, 512)
- ImageNet pre-training [2], with mean normalization of (0.485, 0.456, 0.406) and standard deviation of (0.229, 0.224, 0.225)

Upper & Lower Bounds																					
Src→Trg	Accuracy	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motocycle	bicycle	mIOU
CS → CS	81.0	96.8	75.9	86.9	34.4	34.9	35.3	35.3	54.5	87.7	52.8	90.4	63.9	37.3	88.8	32.6	23.4	16.2	22.0	60.3	54.2
GTA5 → CS	49.0	38.9	7.7	49.9	5.1	1.9	8.4	3.6	1.8	69.2	5.0	61.4	24.0	0.6	31.6	3.7	0.0	0.0	0.9	0.0	16.5

Table 1. Results in terms of accuracy, per class IoU and overall mIoU . The upper bound is using the Cityscapes train fold and Cityscapes val fold. The lower bound instead is given by the Cityscapes validation partition and the GTA5 training split.

We define a lower bound and an upper bound, as indicated in 1, in order to evaluate the performances obtained by our network in solving the Semantic Segmentation task.

We highlight that for the results obtained in 1 an Adam optimizer was used.

4.3. Data Augmentation & Unsupervised Adversarial Domain Adaptation

In order to reduce the Domain Shift and improve our Lower Bound 1, we adopted the Data Augmentation transformations described in 2.2. Then, an Unsupervised Adversarial Domain Adaptation (UDA) framework is implemented to also reduce the Domain Shift. A STDC-Seg is deployed as Generator, and the Fully-Convolutional Discriminator (FCD) proposed in [4] is used as Discriminator. Its architecture is made by 5 convolution layers with a 4×4 kernel and a stride of 2, with respectively $\{64, 128, 256, 512, 1\}$ channels. Except for the last layer, each convolution layer is followed by a leaky ReLU parametrized by 0.2. First, a setup with only augmentation was tested with Adam optimizer, then, using the standard configuration 4.2, Data Augmentation transformations were adopted on the source images before being passed through the Generator, in order to enhance the source data variability. In 8 it is reported the different evolution per epoch of the mIoU metric for the covered configuration alongside the Lower Bound evolution.

4.4. Fourier Domain Adaptation

Fourier Domain Adaptation (FDA) is an image-to-image transformation used to align the source and target domain and thus bridge the gap between the domains. FDA requires a parameter β, accordingly, following [5], we tried different values of β, in particular: 0.01, 0.05, 0.09. Each β was tested using the same UDA framework discussed in 3.3. As observed by Yang et al. [5] the best performing entries by semantic class are equally distributed among models using different values of β. In order to combine the positive results from each model, an average among the three different models is performed. This procedure is defined as Multi-Band Transfer (MBT) [5].

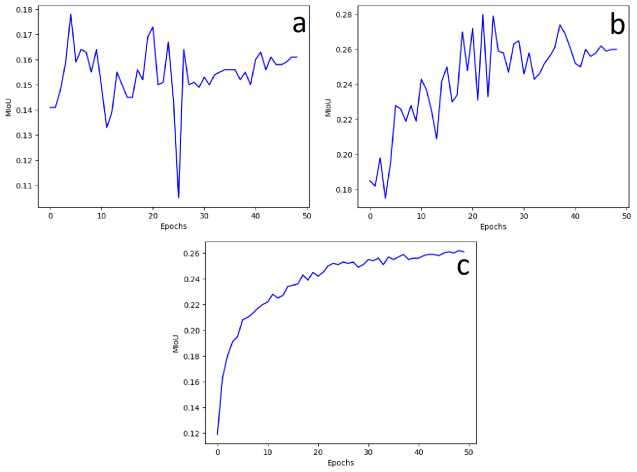
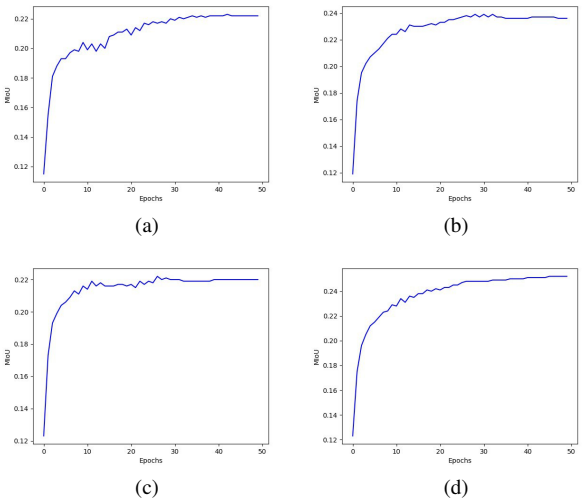


Figure 8. a) Lower Bound; b) Data Augmentation; C) Data Augmentation + UDA

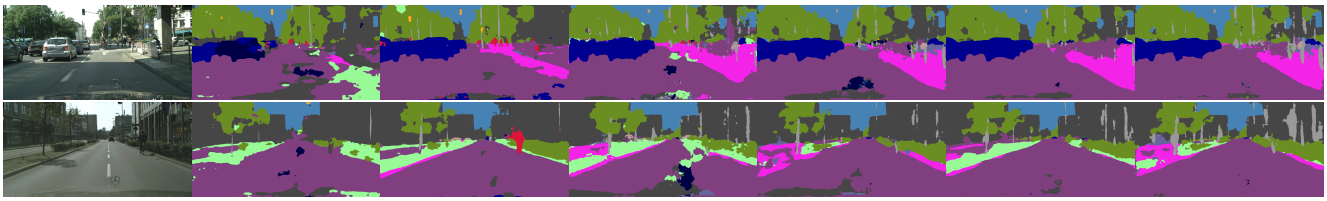


(a) (b) (c) (d)

Figure 9. Learning curve with FDA with: (a) $\beta = 0.01$ (b) $\beta = 0.05$ (c) $\beta = 0.09$ (d) MTB

Domain Adaptation Experiments																					
Method	Accuracy	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motocycle	bicycle	mIOU
GTA5 →CS (Lower Bound)	49.0	38.9	7.7	49.9	5.1	1.9	8.4	3.6	1.8	69.2	5.0	61.4	24.0	0.6	31.6	3.7	0.0	0.0	0.9	0.0	16.5
Data Augmentation	69.1	77.9	13.4	73.1	17.1	5.9	17.6	18.3	3.8	77.8	21.6	59.4	36.8	1.8	57.3	8.7	1.7	0.0	0.7	0.2	26.0
Adv. DA with Data Aug.	68.6	76.7	28.5	72.7	12.8	8.5	21.3	0.5	0.2	72.1	16.1	67.4	1.4	0.0	68.1	7.9	27.6	13.3	0.0	0.0	26.1
Adv. DA with FDA (β=0.01)	64.7	70.7	19.0	63.8	11.9	3.6	18.4	0.0	0.0	64.2	9.9	64.2	0.0	0.0	64.4	15.0	15.9	0.1	0.0	0.0	22.2
Adv. DA with FDA (β=0.05)	67.4	77.5	23.9	65.7	8.6	6.6	17.6	0.2	0.0	70.5	11.3	62.6	0.2	0.0	66.7	10.5	12.2	13.8	0.0	0.0	23.6
Adv. DA with FDA (β=0.09)	64.9	74.5	29.6	59.5	0.7	0.7	15.9	0.0	0.0	66.7	9.2	65.1	0.0	0.0	63.4	11.2	4.5	2.5	0.0	0.0	22.0
Adv. DA with FDA (MBT)	69.3	78.2	24.9	71.1	12.0	5.2	20.6	0.0	0.0	71.7	13.3	67.6	0.0	0.0	72.9	17.2	13.4	9.7	0.0	0.0	25.2
Adv. DA with FDA (β=0.05) SSL	69.3	82.0	27.0	68.0	11.2	9.5	17.2	0.0	0.0	75.5	16.3	61.5	0.2	0.0	70.8	14.1	12.6	15.5	0.0	0.0	25.3

Table 2. Experimental results for different configurations



(a) (b) (c) (d) (e) (f) (g)

Figure 10. a) Original Cityscapes image; b) Lower Bound, c) Data Augmentation; d) Adversarial Domain Adaptation with Data Augmentation; e) Adversarial Domain Adaptation + best FDA; f) MBT; g) SSL

4.5. Self Supervised Learning

In addition, we tried to further improve the performances of the FDA models by using pseudo-labels for the target domain. For the generation of the pseudo-label the MBT model was used in order to obtain the best entries for each semantic class. To obtain high-fidelity annotation the thresholding described in 3.5 was applied. To avoid self-referencing we carried out 10 extra epochs of fine-tuning on our pre-trained best-performing FDA model: $\beta = 0.05$. The learning curve of these epochs is represented in Figure 11 and the final results for all the methods adopted for the domain shift are shown in Table 2.

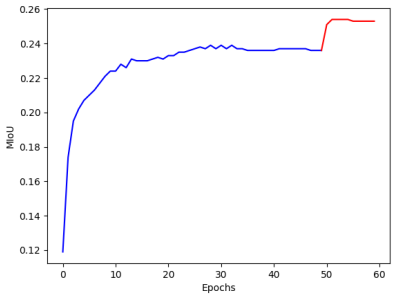


Figure 11. Learning curve of training performed with $\beta = 0.05$ over 50 epochs (represented in blue), with 10 additional epochs of fine-tuning with pseudo-labels (in red).

5. Conclusions

In our work we have implemented several techniques to reduce the Domain Shift between a synthetic dataset, GTA5, and a dataset containing real-word images, Cityscapes, in order to improve the performances of the Real-Time Semantic Segmentation task. Empirical experiments results have shown a significant increment in terms of mIoU, with a peak of +10% respect to the Lower Bound 1. During the experiments performed, we were limited by the GPU hardware in terms of parameters choice, this inevitably lead to sub-optimal performance. More work still needs to be done, especially in fine-tuning the parameters in order to find their optimal values. We hope that in the future more powerful hardware will be employed alongside a more comprehensive dataset to obtain better segmentation predictions.

6. Acknowledgements

We would like to thanks Claudia Cuttano for her guidance and support.

References

[1] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[2] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. 2021. 1, 2, 3, 5

[3] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. *Computer Vision - ECCV 2016*, 2016. 1, 5

[4] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 2, 4, 6

[5] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 4, 5, 6

[6] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. *Computer Vision - ECCV 2018*, 2018. 2, 3

[7] Sicheng Zhao, Xiangyu Yue, Shanghang Zhang, Bo Li, Han Zhao, Bichen Wu, Ravi Krishna, Joseph E. Gonzalez, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia, and Kurt Keutzer. A review of single-source deep unsupervised visual domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 2020. 2