Real-time Domain Adaptation in Semantic Segmentation

Bálint Bujtor s310419
Boyan Cieutat s321171
Inaam ElHelwe s306979

Abstract

This project explores different domain adaptation techniques for real-time semantic segmentation. Achieving high accuracy in semantic segmentation requires large quantities of labeled training data which is expensive and time-consuming. Many attempted to apply models trained on a large-scale labelled source domain to another sparsely labelled or unlabeled target domain with limited success, due to the domain shift across the datasets. In this project, we study different domain adaptation techniques to mitigate this issue. We create a baseline and a paragon model with the pre-trained STDC network as the backbone, that we use to compare the other approaches. We implement data augmentation, adversarial training and Fourier domain adaptation to tackle the challenges of domain adaptation.

1. Introduction

Semantic Segmentation is a computer vision task in which the goal is to categorize each pixel in an image into a class or object. The goal is to produce a dense pixel-wise segmentation map of an image, where each pixel is assigned to a specific class or object, benefiting applications like self-driving vehicles, pedestrian detection, defect detection, therapy planning, and computer-aided diagnosis. Unlike other tasks, semantic segmentation offers pixel-level semantic information, enabling precise spatial understanding and critical judgments in various real-world scenarios. Driven by the rapidly growing demands of many real-time applications, researchers are motivated to explore effective and efficient segmentation networks. They propose designing low-latency, high-efficiency CNN models that achieve satisfactory segmentation accuracy. To enhance real-time inference speed, some approaches utilize lightweight backbones or reduce input image size. However, in our project, we introduce a handcrafted network, [3] based on the Short-Term Dense Concatenate (STDC) module which achieves faster inference speed, an explainable structure, and competitive performance compared to existing methods. However, this trained model may not generalize well to unseen images, especially when there is a domain gap between the training (source) and test (target) images For instance, the distribution of appearance for objects and scenes may vary in different cities, and even weather and lighting conditions can change significantly in the same city.

Domain adaptation, the main focus of our paper addresses this issue. Also known as domain transfer, Domain adaptation is a specialized form of transfer learning that focuses on training a model in a labelled source domain so that it can generalize effectively to a different (but related) unlabeled or sparsely labelled target domain. In this paper, we propose various domain adaptation techniques and compare their performances.
We begin by applying simple data augmentation techniques to the source data. Next, we leverage the adversarial domain adaptation technique proposed in [13]. This method is based on Generative Adversarial Networks (GANs) and involves a segmentation model that predicts output results and a discriminator designed to differentiate between inputs originating from either the source or target segmentation output. Using adversarial loss, the segmentation model is trained to fool the discriminator, thereby aiming to generate similar distributions in the output space for both source and target images.
Given the complexity of adversarial training, we also explored simpler techniques such as Fourier domain adaptation and compared their performance against adversarial training.

2. Related work

2.1. Semantic Segmentation

Many methods have achieved promising results in semantic segmentation by using deep neural networks. By feeding sufficient images and their pixel-wise labeling maps as training data, a deep neural network can learn a mapping between a semantic label and its diversified visual appearances. Some commonly used deep architectures include:

VGG [12]: Proposed by the Visual Geometry Group at

Oxford University, VGG networks paved the way for designing deeper structures for better performance. VGG has been adopted as the backbone of several semantic segmentation models.

ReNet [15]: This network replaced convolutional layers with multi-direction recurrent neural networks (RNNs).

ResNet [4]: ResNet is chosen as the backbone for many semantic segmentation methods. Its key contribution lies in modeling the residual representation into the CNN network structure, solving the difficulty of training very deep networks.

DenseNet [8]: DenseNet connects every layer to each other, providing fewer parameters, more reuse of features, and a better training process. This alleviates the vanishing gradient and model degeneration issues.

As the importance of real-time semantic segmentation arises, **MobileNet** [7] balances accuracy and computational costs by incorporating attention mechanisms.

2.1.1 Real-time Semantic Segmentation

Real-time semantic segmentation algorithms require a fast way to generate high-quality predictions. Although deep-learning-based semantic segmentation methods achieve high accuracy, their computational costs hinder the application in some real-time situations. The key challenge is enhancing the model efficiency while keeping the segmentation accuracy level. Recent real-time semantic segmentation approaches included:

- Cropping and Resizing to restrict the input size to reduce the computation complexity. However, the loss of spatial details corrupts the prediction, especially around boundaries leading to the accuracy decrease in both metrics and visualization.

- Prune the channels of the network to boost the inference speed, nevertheless, it weaknesses the spatial capacity

- ENet [9] suggests dropping the last stage of the model in pursuit of an extremely tight framework. The drawback of this model is that ENet abandons the downsampling operation in the last stage and the receptive end of the model is not enough to cover large objects, resulting in a poor discriminative ability

- BiseNet [17] which is a Bilateral Segmentation Network with two parts: Spatial Path (SP) and Context Path (CP) decoupling the function of spatial information preservation and receptive field offering into two paths.

Our adopted method is an improvement of BiseNet, since adding an additional path to get low-level features is time-consuming, and the auxiliary path is always lack of low-level information guidance. We adopt a novel network based on the Short Term Dense Concatenate module (STDC module) for the purpose of faster inference speed, explainable structure and competitive performance, proposes an efficient lightweight backbone to provide a scalable receptive field. Furthermore, it has a single path decoder which uses detail information guidance to learn the low-level details, in Fig. 1 we can see how STDC network replaced the time-consuming path. Detail Guidance are adopted to guide the low-level layers for the learning of spatial details .

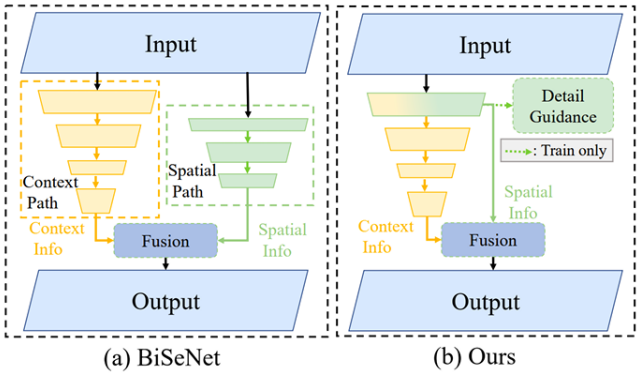


Figure 1. Illustration of architectures of BiSeNet [28] and our adopted approach. (a) presents Bilateral Segmentation Network (BiSeNet [17]), which use an extra Spatial Path to encode spatial information. (b) demonstrates our proposed method, which use a Detail Guidance module to encode spatial information in the low level features without an extra time-cosuming path.

2.2. Domain adaptation

The main challenge of domain adaptation (DA) is the difference between the source and target distributions, which can lead to unreliable predictions on the target domain. In our paper, we focus on two specific domain adaptation settings:

- Single-source, single-target unsupervised domain adaptation (UDA) [18]
- Single-source, single-target self-supervised domain adaptation (SSL)

For our experiments, we use a single source domain (GTA-5) and a single target domain (Cityscapes).

Unsupervised Domain Adaptation (UDA) does not require annotations for the target data. Instead, it relies on sufficient unlabeled target samples to train the model. For example, the source domain can consist of synthetic images and their corresponding pixel-level labels (semantic segmentation), and the target can be real images with no ground-truth annotations. The primary goal is to reduce the domain gap by learning from the labelled source data and the unlabeled target data, improving the model's performance on the target domain.

Self-Supervised Learning (SSL) SSL is a learning paradigm that captures input data's intrinsic patterns and

properties without using human-provided labels. The basic idea of SSL is to create auxiliary tasks based solely on the data itself. These tasks encourage the network to learn meaningful representations by performing well on these tasks, even in the absence of human-annotated labels.

By addressing both UDA and SSL, we aim to enhance the model's ability to adapt from the GTA-5 source domain to the Cityscapes target domain, leveraging the strengths of each approach to improve overall performance.

2.2.1 Adversarial Domain Adaptation

Adversarial domain adaptation methods employ adversarial loss to reduce domain shift. These methods aim to learn representations that effectively distinguish source labels while making it difficult to differentiate between the source and target domains. [14] proposed adding a domain classifier (a single fully connected layer) that predicts the binary domain label of the inputs and designed a domain confusion loss to encourage its prediction to be as close as possible to a uniform distribution over binary labels. CyCADA [5] uses CycleGAN [19] and transfers source domain images to the target domain with pixel alignment thus generating extra training data combined with feature space adversarial learning [6] which aligns the global statistics of our source and target data using a convolutional domain adversarial training technique. In this paper we will explore the Adversarial domain adaptation method proposed by [13] which consists of two modules: a segmentation network G and a fully-convolutional discriminator D using a cross-entropy loss Ld for the two classes (i.e., source and target)

2.2.2 Fourier Domain Adaptation

As adversarial training is difficult, Fourier domain adaptation for semantic segmentation emerged as an alternative approach. Fourier Domain Adaptation (FDA) aligns the low-level statistics between the source and target distributions without training beyond the primary task of semantic segmentation. We will use the model described by [16] which computes the (Fast) Fourier Transform (FFT) of each input image, and swaps the low-level frequencies of the target images with the source images before reconstituting the image for training, via the inverse FFT (iFFT), using the original annotations in the source domain.

2.2.3 SSL Fourier Domain Adaptation

Domain adaptation and semi-supervised learning (SSL) are closely connected. When the domains are aligned, unsupervised domain adaptation becomes SSL. This involves training a model using labelled samples from the source domain and then using the model to generate artificial pseudo-labels for unlabeled images in the target domain. These pseudo-labels serve as ground truth for the target images and are used to retrain the model in subsequent iterations. We propose extending the Fourier domain adaptation model by averaging the output of 3 models trained with varying spectral domain sizes (a hyper-parameter of the model). This approach promotes multiband transfer, which will be discussed in more detail next.

3. Method

3.1. Short-Term Dense Concatenate Module

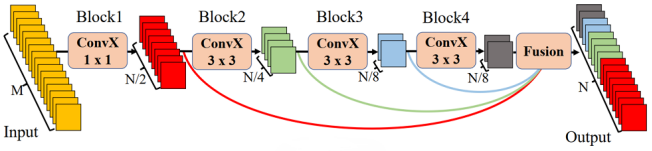


Figure 2. Short-Term Dense Concatenate module (STDC module) used in our network

Short Term Dense Concatenate module (STDC module) is illustrated in Fig. 2 is seperated into 4 blocks each having the formula :

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

Where x_i and x_{i-1} are the output and input of the ith Block respectively. ConvX includes one convolutional layer, one batch normalization layer and ReLU activation layer, and k_i is the kernel size of the convolutional layer. The number of channels decreases systematically from the first block of the STDC module to the last as in the semantic segmentation task we focus on scalable receptive field and multi-scale information. Low-level layers need enough channels to encode more fine-grained information with a small receptive field, while high-level layers with a large receptive field focus more on high-level information induction. We use multiple continuous layers, each encoding the input image or features at different scales and fields, which creates a multi-scale feature representation. The responses from these layers are combined using skip-paths, resulting in a multi-scale feature representation that identifies objects more accurately and efficiently. The final output of the STDC module is:

$$x_{output} = F(x_1, x_2, ..., x_n)$$

x_{output} is the STDC module output and F is the Fusion operation in our method and $x_1, x_2, ..., x_n$ are feature maps from all n blocks.

3.2. Data Augmentation

One of the employed approaches while training the semantic segmentation model is Data augmentation, aimed to

increase performance. It encompasses techniques that operate at the data level rather than modifying the model architecture. It enhances deep learning models by artificially creating diverse and balanced samples for the training dataset. When a dataset is sufficient in both quantity and quality, deep learning models perform more accurately and reliably. Thus, training data must fulfil two key requirements: adequate diversity and size. Data augmentation effectively achieves both these goals.

In our study, we explored some of the data augmentation techniques cited in [1]:

Geometric Transformations

- Flipping: This refers to flipping images horizontally, vertically, or both.

- Rotation: Images are rotated by specified angles to increase variability in orientation.

Photometric Transformations

- Hue: Adjusting the image colours by shifting the hue values.

- Saturation: Altering the colour intensity and lightness, which refers to how light or dark a colour is.

- Grayscale: Converting images to grayscale to focus on intensity variations.

- Brightness: Modifying the brightness to simulate different lighting conditions.

Kernel/Filter Transformations

- Blur: Applying blur filters to reduce image detail and sharpness, mimicking various focus levels.

3.3. Adversarial Domain Adaptation

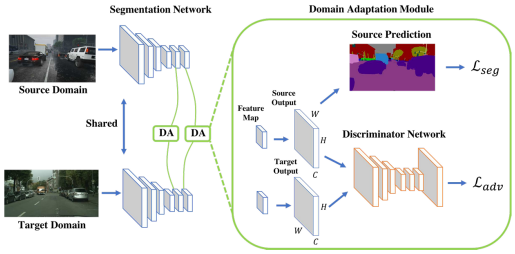


Figure 3. Overview of Adversarial Domain Adaptation Technique used. The illustrated module follows multi-level adversarial learning by adopting two adaptation modules at two different levels.

The Adversarial Domain Adaptation model proposed by [13] consists of two modules: a segmentation network **G** that in our case will be based on the STDC module explained in the previous section and a discriminator **D**. Given 2 sets of images having size H x W from source and target domains I_S and I_T respectively. The labelled source image I_S is first forwarded to the segmentation network for optimizing **G** and we receive P_T the output prediction and a segmentation loss computed based on the source ground truth. We also pass I_T to the segmentation model to predict the segmentation softmax output. To make the target predictions closer to the source ones we use these two predictions (P_T and P_S) as the input to the discriminator D to distinguish whether the input is from the source or target domain, by calculating an adversarial loss on the target prediction. This loss is back-propagated to the segmentation network G which encourages it to generate similar segmentation distributions in the target domain to the source prediction.

The loss computed is given by :

$$L(I_s, I_t) = L_{seg}(I_s) + \lambda_{adv} L_{adv}(I_t)$$

Let L_{seg} represent the cross-entropy loss calculated using the ground truth annotations for the source domain. L_{adv} is the adversarial loss that adapts the predicted segmentation of target images to the distribution of source predictions. λ_{adv} is the weight used to balance the two losses.

3.4. Fourier Domain Adaptation

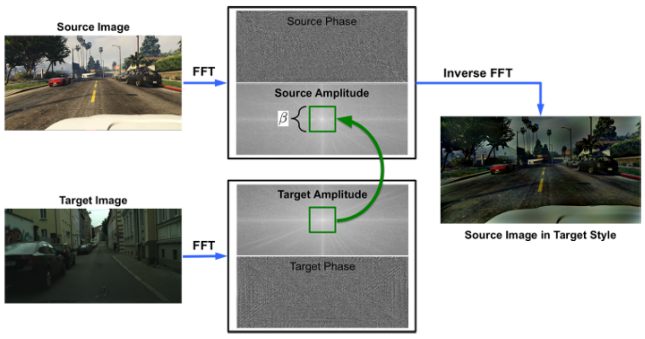


Figure 4. Overview of the Fourier Domain Adaptation for Semantic Segmentation technique proposed by [16].

The idea of the Fourier Domain adaptation model proposed by [16] arises from the observation that the low-level spectrum (amplitude) can vary significantly without affecting the perception of high-level semantics. Whether something is a vehicle or a person should not depend on the characteristics of the sensor, or the illumination, or other low-level sources of variability. Yet such variability has a significant impact on the spectrum, forcing a learning-based model to "learn it away" along with other nuisance variability. If this variability is not represented in the training set, the models fail to generalize. The method, as illustrated in Fig. 4 consists of computing the (Fast) Fourier Transform (FFT) of each input image, and replacing the low-level frequencies of the target images with those from the source images before reconstituting the image for training via the inverse FFT (iFFT) using the original annotations in

the source domain. Fourier domain adaptation requires selecting one free parameter which is the size of the spectral neighborhood to be swapped. Given the Fourier transform F and F^{-1} the inverse Fourier transform that maps spectral signals (phase and amplitude) back to image space, FDA can be formalized as:

$$x^{s\to t} = F^{-1}([M_\beta \odot F^A(x^t) + (1-M_\beta)\odot F^A(x^s), F^P(x^s)])$$

where M_β is a mask, whose value is zero except for the center region where $\beta \in (0, 1)$.

3.5. Self-Supervised Training

We exploited the self-supervised learning method proposed by [16]. This method relies on Multi-band transfer (MBT) which consists of first averaging over the predictions of different FDA models (these models are trained on different βs, to achieve better predictions when validating, To regularize better, we also apply a thresholding on the confidence values of each prediction. For each semantic class, we accept the predictions with confidence within the top 66% or above 0.9. The obtained predictions are then used as ground truth labels for the self-learning, keeping the average a probability distribution over K categories. Using the pseudo-labels generated by M models, we can train the segmentation network to get further improvement using the following self-supervised training loss:

$$\mathcal{L}_{sst}(\phi^w; D^{s\to t}, \hat{D}^t) = \mathcal{L}_{ce}(\phi^w; D^{s\to t}) + \lambda_{ent}\mathcal{L}_{ent}(\phi^w; D^t) + \mathcal{L}_{ce}(\phi^w; \hat{D}^t) \quad (1)$$

where $\hat{D}^t$ is D^t augmented with pseudo labels $\hat{y}_i^t$'s.

4. Experiments

In this section, we present experimental results to evaluate the proposed domain adaptation methods for semantic segmentation under various settings.
In our study, we use **GTA-5** [10] as our synthetic source dataset and **Cityscapes** [2] as the real target dataset.

4.1. Datesets

Cityscapes

Cityscapes [2] is a semantic scene parsing dataset, which is taken from a car perspective. It contains 5,000 fine annotated images and split into training, validation and test sets, with 2,975, 500 and 1,525 images respectively. The annotation includes 30 classes, 19 of which are used for semantic segmentation task. The images have a high resolution of 2048x1024, thus it is challenging for the real-time semantic segmentation. For fair comparison, we only use the fine annotated images in our experiments.

GTA5

The GTA5 dataset [10] consists of 24966 images with the resolution of 1914×1052 synthesized from the video game based on the city of Los Angeles. There are 24 semantic classes present in the dataset, whose ground truth annotations are compatible with the Cityscapes dataset [2] that contains 19 categories. To achieve compatibility a semantic class remapping technique was implemented, following a commonly accepted method [11]. We use the full set of GTA5 and adapt the model to the Cityscapes training set with 2975 images. During testing, we evaluate on the Cityscapes validation set with 500 images.

For the Cityscapes and GTA-5 transformations, we applied a series of preprocessing steps to standardize both datasets. Initially, the Cityscapes images are resized to a resolution of 512x1024, while the GTA-5 images are resized to 526x957. In the case of the FDA training, both images were resized to 512x1024 to allow the correct execution of the FFT and iFFT operations. The resized images are then converted to image format and normalized using the mean and standard deviation provided by the ImageNet dataset. This ensures consistency in pixel intensity values and maintains uniform pre-processing across different datasets.

4.2. Metrics

To evaluate the performance of our models, we used two metrics: **Pixel Accuracy** and **Intersection over Union (IoU)**.

Pixel Accuracy is calculated by summing the true positives (correctly predicted pixels) and dividing by the sum of true positives and false positives (incorrectly predicted pixels). This ratio quantifies the model's pixel-level accuracy across all images in the dataloader. However, this metric can sometimes be misleading when the class representation is small within the image, as it tends to be biased towards reporting how well negative cases are identified.

IoU measures the number of pixels common between the target and prediction masks divided by the total number of pixels present across both masks. This metric provides a more balanced evaluation of the model's performance, especially in cases with imbalanced class distributions.

4.3. Simple Training

In all subsequent training sessions, we set the number of epochs to 50 and the learning rate to 0.01. We iterated between batch sizes 4 and 8, and the number of workers 2 and 4, utilizing two different optimizers: ADAM with a batch size of 4 and SGD with a batch size of 8.

We begin by evaluating the performance of the chosen

semantic segmentation model without accounting for domain shift. Initially, we train the BiSeNet neural network with a pre-trained STDC backbone.

4.3.1 Experiment Cityscapes

In the initial step, we establish the upper bound for the domain adaptation phase. For simplicity, we assume an ideal scenario where the validation dataset is identical to the training dataset, so we train and test both on the Cityscapes dataset, we used the training settings mentioned above. The obtained accuracies are presented in 1

4.3.2 Experiment GTA-5

As the GTA-5 dataset directory was not pre-split between training and validation, we had to partition it ourselves. The dataset is divided into training and validation sets using a validation split ratio of 0.2, which is a common practice. This ratio ensures that a sufficient portion of the dataset is reserved for validation while still retaining a substantial portion for training. Then we applied the same network used before to train on Cityscapes.1

4.3.3 Domain Shift

Before delving into domain adaptation techniques, we conducted a preliminary experiment: using our model trained on the GTA-5 dataset and validating it on Cityscapes without employing any domain adaptation methods. As expected, this approach resulted in a significant drop in accuracies compared to training and validating within the same dataset. This initial analysis highlighted the impact of domain shift between these datasets, emphasizing the need for effective domain adaptation strategies to address such challenges. The results are shown in 1

Optimizer	Simple Training	Pixel Accuracy (%)	mIoU (%)
ADAM	Cityscapes	78.7	44.4
	GTA-5	78.2	49.5
	Domain Shift	52.1	11.5
SGD	Cityscapes	80.6	63.5
	GTA-5	80.0	64.0
	Domain Shift	43.5	14.2

Table 1. Experimental results of training and validating a simple BisNet model for semantic segmentation on Cityscapes, training on GTA-5, then validating the model trained on GTA-5 on Cityscapes.

4.4. Domain Adaptation

4.4.1 Data augmentation

The initial approach to enhance performance involves applying data augmentation techniques. This process increases data variability, enabling the model to generalize more effectively. Specifically, we applied contrast adjustment with a factor ranging from 2 to 6, set the hue to 0.3, and horizontally flipped the images. These augmentation methods were applied with a probability of 50%, ensuring a balanced mix of augmented and original images. This balance helps the model learn to recognize features in diverse conditions. As expected, the performance increased significantly, particularly when using SGD as an optimizer for domain shift, improving mIoU from 14.2% without data augmentation to 15.6% with data augmentation. It's interesting noting that applying data augmentation when training in the same domain didn't play a role in improving performance when ADAM is used as an optimizer.

Optimizer	Simple Training	Pixel Accuracy (%)	mIoU (%)
ADAM	Cityscapes	76.3	43.9
	GTA-5	73.7	46.7
	Domain Shift	43.2	14.2
SGD	Cityscapes	80.9	57.7
	GTA-5	80.6	62.5
	Domain Shift	44.7	15.6

Table 2. Experimental results of training and validating a simple BisNet model for semantic segmentation on Cityscapes, training on GTA-5, then validating the model trained on GTA-5 on Cityscapes with data augmentation applied.

4.4.2 Unsupervised Adversarial Domain Adaptation

Adversarial domain adaptation involves two models: the generator and the discriminator. In our approach, we use the STDC model from the previous section as the generator and the FCD discriminator proposed by [13]. Balancing the weight between segmentation and adversarial losses during optimization is crucial, making the selection of an appropriate λ_{adv} essential. Given our limited computational resources and the scope of our research not including extensive hyper-parameter tuning, we opted to use the λ_{adv} value that performed best in [13]. Additionally, we used the same learning rate they employed, which is 0.0001.

As data augmentation proved to enhance performance for domain shift we conducted all following experiments with data augmentation. We also used two different optimizers, ADAM and SGD, to determine if performance

could be further enhanced, even though previous experiments consistently showed better performance with SGD. The results are shown in the table below.

Optimizer	Pixel Accuracy (%)	mIoU (%)
ADAM	59.4	19.3
SGD	69.7	22.5

Table 3. Experimental results of training and validating ADA with and without data augmentation and with 2 different optimizers.

The results clearly indicate a significant performance improvement when applying adversarial domain adaptation compared to the simple domain adaptation training discussed in the previous section. This enhancement is particularly noticeable when using SGD as the optimizer increasing from 15.6% to 22.5%.

4.4.3 Fourier Domain Adaptation

As an extension of our research, we employed the proposed Fourier Domain Adaptation (FDA) method. FDA involves selecting a free parameter, β , which determines the size of the spectral neighbourhood to be swapped. We tested our model using three different values of $\beta(0.01, 0.05, 0.09)$. In all experiments, we set the entropy regularization weight λ_{ent} to 0.05. For the cross-entropy loss, we set η to 2.0 to impose a greater penalty on high-entropy predictions, thereby encouraging the model to make more confident predictions.
We observe a significant improvement in performance compared to all experiments held so far specially when applying the SGD optimizer.

Optimizer	β	Pixel Accuracy (%)	mIoU (%)
ADAM	0.01	67.6	21.5
	0.05	68.9	22.9
	0.09	65.1	18.4
SGD	0.01	71.5	30.4
	0.05	73.2	30.9
	0.09	72.0	29.8

Table 4. Experimental results of training and validating FDA with different values of β

As observed from the experimental results reported in Table 5, FDA outperforms all previously trained models, achieving a pixel accuracy of 72.8% when trained with the SGD optimizer and $\beta = 0.05$, and an mIoU of 31.1% for $\beta = 0.01$ the results can be observed in the visualisation plotted in 5.

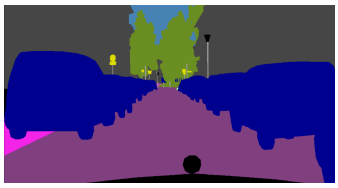
4.4.4 SSL FDA

We employed the three previously trained FDA models, corresponding to $\beta = (0.01, 0.05, 0.09)$, to perform self-supervised learning. By simply averaging prediction of the three models (MBT), we get a relatively good performance (70% pixel accuracy with SGD) but still simple FDA has performed better. Then the pseudo labels generated by MBT are treated as the ground truth labels to the self-supervised learning model discussed before.

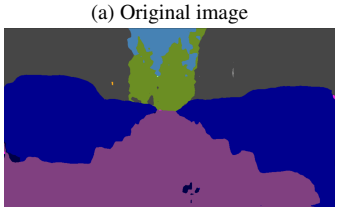
Optimizer	Pixel Accuracy (%)	mIoU (%)
ADAM	63.7	11.2
SGD	66.3	17.3

Table 5. Experimental results of training and validating SSL FDA with $\beta = 0.01$ and different optimizers

Despite expectations of outperforming simple FDA trainnings, the results of SSL FDA did not meet anticipated performance levels. Further analysis, experimentation, and adjustments to the SSL FDA approach may help improve the model's performance and achieve the intended outcomes.



(a) Original image



(b) Predicted image with FDA with $\beta = 0.01$ and SGD

Figure 5. Visualisation of original labels and predictions

5. Conclusion

In our study aimed at enhancing the performance of a semantic segmentation model capable of handling domain shift, we trained Adversarial Domain Adaptation (ADA), Fourier Domain Adaptation (FDA), and Self-Supervised Fourier Domain Adaptation (SSL FDA) models with and without data augmentation. These models were trained using the STDC backbone.
Our experiments yielded several key observations. FDA

with a β value of 0.05 achieved the highest pixel-level accuracy, highlighting its effectiveness in accurately predicting individual pixel classes. While, FDA with a β value of 0.01 attained the best mean Intersection over Union (mIoU), demonstrating its superior performance in capturing overall segmentation quality.

Furthermore, the application of data augmentation techniques significantly enhanced the model's performance. The increased data variability from augmentation enabled the models to generalize better to diverse conditions, resulting in improved outcomes. Among the two optimizers tested, SGD consistently outperformed ADAM, establishing it as the optimal choice for real-time semantic segmentation tasks.

Future work could focus on extensive hyperparameter tuning to further improve model performance. This could involve fine-tuning λ_{adv} values, learning rates, and other critical parameters to push the boundaries of model accuracy and robustness in handling domain shift.

You can find the corresponding GitHub repository for our project at the following link: GitHub Repository.

References

[1] Khaled Alomar, Halil Ibrahim Aysel, Xiaohao Cai, Jérôme Gilles, and Luminiţa Moraru. Data augmentation in classification and segmentation: A survey and new strategies. 2023. 4

[2] 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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5

[3] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. 2022. 1

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. 2016. 2

[5] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *CoRR*, abs/1711.03213, 2017. 3

[6] J. Hoffman, D. Wang, F. Yu, and T. Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*, abs/1612.02649, 2016. 3

[7] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[8] Zhihua Huang, Jie Zhang, Peidong Li, Lanjian Xu, Xiaomin Zhang, Yangyang Yuan, and Lei Xu. Tert-butylation of naphthalene by tertiary butanol over hy zeolite and cerium-modified hy catalysts. 2017. 2

[9] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016. 2

[10] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games, 2016. 5

[11] Sarrrrry. Pytorchdl_gta5. https://github.com/sarrrrry/PyTorchDL_GTA5, 2024. Accessed: 2024-06-09. 5

[12] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2014. 1

[13] 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. 1, 3, 4, 6

[14] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. *arXiv preprint arXiv:1510.02192*, 2015. Submitted on 8 Oct 2015. 3

[15] Francesco Visin, Kyle Kastner, Kyunghyun Cho, Matteo Matteucci, Aaron Courville, and Yoshua Bengio. Renet: A recurrent neural network based alternative to convolutional networks. 2015. 2

[16] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. 2020. 3, 4, 5

[17] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. *arXiv preprint arXiv:1808.00897*, 2018. 2

[18] 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. *arXiv preprint arXiv:2002.05735*, 2020. 2

[19] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 3