Real-time Domain Adaptation in Semantic Segmentation

Ali Behlooei Dolatsaraei
Politecnico di Torino
s334033@studenti.polito.it

Abstract

Semantic segmentation aims to classify each pixel of an image into predefined categories. This paper examines the performance differences between classical segmentation networks (DeepLabV2) and real-time networks (BiSeNet with ResNet-18 backbone) when applied to real-world urban scenes from the CityScapes dataset. Our analysis reveals that while classical networks achieve higher accuracy, real-time networks provide a favorable balance between speed and performance for practical applications. The core focus of our work addresses the domain shift problem, where we investigate the challenges of training BiSeNet on synthetic GTA5 images and deploying it on real-world CityScapes data. To bridge this synthetic-to-real domain gap, we explore two key approaches: data augmentation techniques and adversarial domain adaptation. Among various augmentation methods tested (Color Jitter, Random Horizontal Flip, and their combinations), Gaussian Blur emerges as the most effective technique, significantly improving the model's ability to generalize across domains. However, our most promising results are achieved through adversarial domain adaptation, which implements a competitive learning process between a feature extractor and a discriminator. This approach demonstrates superior performance in segmenting common urban scene elements while maintaining real-time processing capabilities. The adversarial method shows particular strength in identifying large-scale features like roads and buildings, though challenges persist in detecting smaller or less frequent objects in the scene. Github Repository link

Keywords: Semantic Segmentation, Real-time Segmentation, Domain Adaptation, BiSeNet, DeepLabV2, Synthetic-to-Real Transfer, Adversarial Learning

1. Introduction

Semantic segmentation is a crucial area of computer vision that aims to classify every pixel in an image into predefined categories, providing a detailed understanding of scene content. In recent years, deep learning has revolutionized this field, enabling significant improvements in terms of accuracy and efficiency. However, one of the most significant challenges remains the adaptation of semantic segmentation models to domains different from those on which they were originally trained. This problem, known as "domain shift," is particularly evident when moving from synthetic datasets to real-world datasets, where the cost and time required for annotation are substantial. This project focuses on domain adaptation in real-time semantic segmentation networks. Its main goal is to develop and implement techniques that allow a segmentation network to maintain high performance even when applied to different domains. The emphasis on real-time performance is crucial for practical applications such as autonomous driving. Our methodology follows a systematic approach. First, we evaluate two semantic segmentation networks: DeepLabV2 [1], representing classical architectures, and BiSeNet, designed for real-time applications. Both networks are initially trained on the Cityscapes [2] dataset to establish baseline performance metrics, including Mean Intersection over Union (MIoU), latency, Floating Point Operations per Second (FLOPs), and the number of parameters. To address the domain shift challenge, we then train BiSeNet on the synthetic GTA5 [4] dataset and evaluate its performance on real Cityscapes data. This setup allows us to quantify the impact of domain shift and observe the performance degradation when the model encounters real-world scenarios. As an initial solution, we implement various data augmentation techniques during training, with Gaussian Blur proving the most effective among the four methods tested. In the final phase, we implement a more sophisticated domain adaptation strategy using an adversarial approach. This method establishes a competitive contest between two components: a feature extractor that learns domain-invariant representations, and a discriminator that attempts to distinguish between source and target domain features. By combining this adversarial approach with our most effective data augmentation techniques, we significantly improve model robustness when applied to real-world scenarios, effectively bridging the gap between synthetic training data and real-world applications.

2. Related Works

Semantic segmentation has advanced significantly in recent years, with innovations ranging from architectural improvements to novel training strategies. This section examines the fundamental works that have shaped our approach to domain adaptation in real-time segmentation.

2.1. Semantic Segmentation

Semantic segmentation [3] represents a fundamental computer vision task that involves partitioning an image into semantically meaningful regions . This process goes beyond simple object detection by requiring pixel-level classification, where each pixel in the image must be assigned to a specific semantic category. The challenge lies in maintaining precise boundaries between different objects while ensuring coherent segmentation across varying scales and contexts. In the realm of classical semantic segmentation architectures, DeepLabV2 stands as a seminal contribution. This architecture leverages deep convolutional neural networks (DCNNs) and introduces several key innovations to achieve high-accuracy pixel-level classification. DeepLabV2's success can be attributed to its use of atrous convolutions (also known as dilated convolutions), which expand the receptive field without increasing computational complexity, and its implementation of fully connected Conditional Random Fields (CRFs) for refined boundary localization. While classical approaches prioritize accuracy, real-time semantic segmentation introduces the additional challenge of balancing spatial precision with inference speed. The Bilateral Segmentation Network (BiSeNet) [8] addresses this challenge through an innovative dual-path architecture. BiSeNet's design comprises two complementary components: a Spatial Path that preserves spatial resolution and detailed information for precise boundary localization, and a Context Path that rapidly downsamples the input to capture broad contextual information. This architecture employs ResNet-18 as its backbone, pre-trained on ImageNet, striking an effective balance between model depth and computational efficiency. The relatively shallow 18-layer architecture, compared to deeper alternatives, enables faster inference while maintaining competitive segmentation accuracy through its specialized dual-path design.

2.2. Domain Adaptation

Domain adaptation [9] methods are essential for overcoming the domain-shift problem that occurs when deploying models across different visual domains. These techniques aim to adapt models trained on a source domain (typically synthetic data) to perform effectively on a target domain (typically real-world data) without requiring target domain labels. Two significant approaches have emerged in recent research. Fourier Domain Adaptation (FDA) operates in the frequency domain, exploiting the observation that image semantics are primarily encoded in the phase component while domain characteristics are captured in the amplitude spectrum. By swapping low-frequency components between source and target images, FDA achieves domain adaptation without complex model modifications. Domain Adaptation via Cross-domain Mixed Sampling (DACS) takes an alternative approach through cross-domain mixing. This method creates hybrid training samples by combining elements from both domains using ClassMix, where selected semantic classes from source images are transferred onto target domain images. While the resulting images may not maintain perfect visual realism, they effectively help the model learn domain-invariant features.Both approaches offer distinct advantages: FDA provides a lightweight, model-agnostic solution, while DACS enables more sophisticated domain mixing, each contributing valuable strategies for bridging the synthetic-to-real domain gap.

2.3. Adversarial Approach

The adversarial approach to domain adaptation implements a sophisticated competitive learning framework based on two competing networks. At its core, this method [6] establishes an adversarial game between a segmentation network (generator) that extracts features from images and a domain classifier (discriminator) that attempts to identify whether these features originate from the source or target domain. The key innovation lies in the opposing objectives of these networks: while the segmentation network strives to produce features that are indistinguishable between domains while maintaining semantic accuracy, the discriminator works to differentiate between source and target domain features. This adversarial dynamic creates a powerful learning mechanism where the segmentation network is forced to generate increasingly domain-invariant features to deceive the discriminator. Through this competitive process, the model learns robust feature representations that effectively bridge the domain gap. The power of this approach lies in its ability to discover complex, non-linear transformations that can adapt to various types of domain shifts, making it particularly effective for handling the challenging transition from synthetic to real-world data

3. Experimental Setup

The experimental framework of this study centers on two complementary semantic segmentation architectures: DeepLabV2, notable for its advanced field-of-view manipulation techniques, and BiSeNet, which achieves the challenging balance of real-time inference while preserving spatial resolution accuracy.

3.1. Data processing and Preparation

Our experimental framework relies on two complementary datasets that enable comprehensive evaluation of domain adaptation techniques in semantic segmentation. The Cityscapes dataset serves as our target domain, providing high-quality pixel-level annotations of real-world urban scenes. This dataset is particularly valuable for practical applications, offering detailed semantic labeling across 30 classes that are logically grouped into eight categories: flat surfaces, construction elements, nature, vehicles, sky, objects, humans, and void areas. For our specific implementation, we focus on 19 classes to maintain consistency across our experiments. Complementing the real-world data, we utilize the GTA5 dataset as our source domain. This synthetic dataset, generated from a sophisticated video game environment, provides an extensive collection of perfectly annotated images. The GTA5 dataset has been specifically structured to align with the same 19 classes used in our Cityscapes implementation, making it an ideal candidate for investigating domain adaptation techniques between synthetic and real-world environments.

The preparation of these datasets involves a carefully designed preprocessing pipeline to ensure optimal model performance. All images undergo a standardization process where they are resized to a consistent resolution of 512x1024 pixels. This resizing operation employs antialiasing techniques to maintain image quality and preserve the fine details crucial for accurate segmentation. Following the dimensional standardization, we implement a normalization process using established ImageNet-standard values. The normalization step is important because it aligns our input data with the pre-trained weights of our models, which were initially trained on the ImageNet dataset. Label processing requires equal attention to detail. The segmentation labels are carefully resized to match the input dimensions while ensuring that all class values remain within the valid range of 0 to 19. This clamping operation is essential for maintaining semantic consistency across our datasets. The final step in our preprocessing pipeline involves the organization of data for model training. We utilize DataLoader objects configured with a batch size of 4, maintaining separate loaders for training and validation sets to ensure proper model evaluation. This comprehensive preprocessing approach is consistently applied across both datasets, creating a standardized foundation for our experiments with both the classic DeepLabV2 architecture and the real-time BiSeNet model.

3.2. DeepLabV2

Deep Convolutional Neural Networks (DCNNs) have demonstrated remarkable capabilities in image classification tasks. DeepLabV2 advances these capabilities for semantic segmentation through its refined architecture, particularly focusing on three key components: atrous convolution, atrous spatial pyramid pooling with parallel atrous rates, and fully connected Conditional Random Fields (CRF).

The foundation of DeepLabV2's architecture is atrous convolution (also known as dilated convolution), which addresses the challenge of spatial resolution loss that typically occurs due to repeated max-pooling and striding operations in conventional CNNs. The output of an atrous convolution is defined as:

$$y[i] = \sum_{k=1}^{K} x[i + r \cdot k]w[k] \tag{1}$$

where r represents the atrous rate that determines the stride for sampling the input signal. This approach introduces controlled gaps between filter weights, enabling the network to sample input pixels at greater distances while maintaining the same number of parameters. The result is an expanded receptive field without increased computational cost.

A distinguishing feature of DeepLabV2 is its specific implementation of Atrous Spatial Pyramid Pooling (ASPP), which uses four parallel atrous convolutional layers with defined rates of 6, 12, 18, and 24. These parallel branches process the feature maps simultaneously with different atrous rates, allowing the network to effectively capture multi-scale context. This multi-scale processing is particularly crucial for segmenting objects of varying sizes within the same image.

The final component is a fully connected Conditional Random Field (CRF), which serves as a post-processing step to refine the segmentation output. This refinement stage helps recover detailed structure and precise boundaries in the final segmentation map by considering both local features and global context of the image.

3.3. BiSeNet Architecture

The Bilateral Segmentation Network (BiSeNet) represents an innovative approach to semantic segmentation, specifically designed to achieve both high accuracy and real-time performance through its distinctive dual-path architecture. This design addresses the fundamental challenge in semantic segmentation: maintaining spatial precision while capturing sufficient contextual information, all within the constraints of real-time processing requirements.

The architecture's primary innovation lies in its parallel processing streams, each serving a specialized function:

The Spatial Path is engineered to preserve spatial resolution and capture fine-grained details. It employs a series of convolution layers with strided operations, maintaining a careful balance between downsampling for efficiency and preserving spatial information. This path is crucial for

accurate boundary delineation and fine detail preservation in the segmentation output, particularly in complex urban scenes where precise object boundaries are essential. The Context Path operates in parallel, utilizing a lightweight backbone network (ResNet-18) [7]to extract contextual features efficiently. Through progressive downsampling and feature aggregation, this path captures semantic context at multiple scales. The lightweight design is intentional, optimizing the computational overhead while ensuring the capture of sufficient contextual information for accurate scene understanding. These parallel paths converge in the Feature Fusion Module, a sophisticated integration mechanism that combines the complementary information from both paths. This module employs adaptive weighting strategies to merge the high-resolution spatial details from the Spatial Path with the rich semantic context from the Context Path. The fusion process is designed to be computationally efficient while preserving the essential characteristics of both feature streams. The synergistic combination of these architectural elements enables BiSeNet to achieve real-time performance without compromising segmentation quality.

3.4. Training and Validation Methodology

Both DeepLab and BiSeNet models were trained on the Cityscapes dataset for 50 epochs. The training process implements specific optimization strategies to ensure optimal convergence while maintaining consistent evaluation protocols. Both networks are optimized using the Adam optimizer with an initial learning rate of 0.001. The learning rate follows a polynomial decay schedule:

$$lr = lr_{initial} \times (1 - \frac{current\_iter}{max\_iter})^{power} \quad (2)$$

with the power parameter set to 0.9. The networks are trained using Cross-Entropy Loss:

$$\mathcal{L}_{seg}(I_s) = -\sum_{h,w}\sum_{c\in C} Y_s^{(h,w,c)} \log(P_s^{(h,w,c)}) \quad (3)$$

where Y_s represents the ground truth labels for source images and $P_s = G(I_s)$ denotes the output segmentation map.

The validation process evaluates model performance on unseen data from the Cityscapes validation dataset. This phase is crucial for assessing the models' generalization capabilities and preventing overfitting. Performance is primarily quantified through the mean Intersection over Union (mIoU) metric, which measures the overlap between predicted segmentation and ground truth.

3.4.1 Performance Analysis

The evaluation results demonstrate distinct characteristics for each model. DeepLab achieves higher segmentation accuracy with an mIoU of49.89%, but requires greater computational resources, evidenced by its 92.34ms latency and 43.901M parameters. BiSeNet, designed for real-time applications, maintains competitive accuracy (47.16% mIoU) while significantly reducing computational overhead, achieving a latency of 51.71ms with only 12.582M parameters. The FLOPs measurement (0.71T vs 0.58T) further confirms BiSeNet's efficiency in balancing accuracy with computational requirements.

Table 1. Performance Comparison of DeepLab and BiSeNet after 50 Epochs

Model	mIoU (%)	Latency (ms)	FLOPs (T)	Params (M)
DeepLab	49.89	92.34	0.712	43.901
BiSeNet	47.16	51.71	0.58	12.582

4. Domain Shift

Semantic segmentation networks typically begin their development phase in controlled, synthetic environments. These artificial environments are particularly advantageous as they facilitate the rapid generation of large-scale annotated datasets at significantly reduced costs compared to the resource-intensive process of collecting and manually annotating real-world data. This approach has become increasingly prevalent in the field of computer vision, especially for tasks requiring dense pixel-wise annotations. However, models trained exclusively on synthetic data frequently encounter performance degradation when deployed in real-world scenarios due to the phenomenon known as domain shift. This shift manifests as a significant divergence between the data distribution of the training dataset (source domain) and the operational environment (target domain). The visual disparities between synthetic and real images, including variations in texture, lighting conditions, object appearance, and scene composition, contribute to this performance gap. To quantitatively assess the impact of domain shift, we conducted experiments using BiSeNet, our chosen real-time segmentation architecture. The network was initially trained on the synthetic GTA5 dataset as the source domain, which provides perfectly annotated synthetic urban scenes. The model's generalization capability was then evaluated on the Cityscapes dataset, representing real-world urban environments. This experimental setup allows us to measure the direct impact of domain shift on segmentation performance and establishes a baseline for subsequent domain adaptation techniques. The choice of GTA5 and Cityscapes datasets for this evaluation is particularly relevant as both datasets share similar urban scene contexts and semantic categories, allowing for a focused analysis of the domain shift problem while minimizing the impact of semantic disparities. This controlled experimental setting en-

ables us to isolate and study the effects of visual domain shift on model performance, providing valuable insights for developing effective domain adaptation strategies.

4.1. Data Preprocessing and Training

For both datasets, we implemented specific preprocessing pipelines to ensure optimal training conditions. The Cityscapes preprocessing maintains consistency with our previous experimental setup, using images resized to 512x1024 pixels. For the GTA5 dataset, input images are processed at 720x1280 resolution to preserve the detailed features of synthetic urban scenes. Both datasets undergo normalization using identical mean and standard deviation values to ensure consistent input distributions. The data loading process employs different configurations for training and validation. The GTA5 training data utilizes a batch size of 12 with shuffling enabled to enhance training randomization, while the Cityscapes validation set uses the same batch size without shuffling to maintain consistent evaluation conditions. This preprocessing pipeline ensures standardized input formats while preserving the unique characteristics of each dataset.

The training methodology employs BiSeNet with Cross-Entropy Loss and Adam optimizer, maintaining consistency with our previous experiments. The model is trained on the GTA5 dataset for 50 epochs to learn representations from synthetic urban scenes. For evaluation, we implement a comprehensive validation process that analyzes per-class performance on the Cityscapes dataset using confusion matrices. This enables the calculation of class-wise Intersection over Union (IoU) metrics, culminating in a mean IoU (mIoU) score that quantifies the model's overall segmentation capability across different urban scene elements.

Class	Average IoU (%)
mIoU	18.8
road	62.5
sidewalk	4.8
building	61.2
wall	2.8
fence	6.4
pole	8.4
light	2.9
sign	1.6
vegetation	68.0
terrain	6.4
sky	59.5
person	24.6
rider	0.1
car	39.0
truck	6.0
bus	1.8
train	0.7
motocycle	0.3
bicycle	0.0

Table 2. Performance Metrics for GTA5 to Cityscapes Domain Shift

5. Data augmentations

Data augmentation techniques play a crucial role in addressing the domain shift between synthetic and real datasets. By introducing controlled variations to the synthetic training images, these techniques enhance the model's ability to generalize across different domains. The augmentations are designed to simulate real-world variations that occur in natural environments, effectively expanding the diversity of the training data without requiring additional synthetic image generation. Our augmentation strategy focuses on three key techniques: Gaussian Blur, Color Jitter, and Random Horizontal Flip. Each technique addresses specific aspects of the domain gap between synthetic and real-world imagery. The Gaussian Blur transformation simulates the natural softness of real-world images by reducing the artificially sharp edges often present in synthetic data. This technique helps the model adapt to the inherent image quality variations found in real-world camera captures. Color Jitter augmentation modifies the brightness and contrast of synthetic images, simulating diverse lighting conditions encountered in real-world scenarios. This is particularly important as synthetic datasets often lack the natural lighting variations present in real environments. By exposing the model to different illumination conditions during training, we enhance its robustness to lighting changes in the target domain. Random Horizontal Flip introduces spatial variability into the training process, preventing the model from developing directional biases. This transformation is especially valuable for urban scene understanding, where objects and structures can appear in various orientations. Additionally, we experiment with combinations of these augmentation techniques to assess their complementary effects on reducing domain shift.

5.1. Data Preprocessing and Training

For our augmentation experiments, we implemented a comprehensive preprocessing pipeline for the GTA5 dataset. The Gaussian Blur augmentation employed a kernel size of 5×9 with sigma values ranging from 0.1 to 5, strategically chosen to reduce the artificial sharpness characteristic of synthetic images. For color space transformations, we applied ColorJitter with controlled parameters: brightness (0.2), contrast (0.2), saturation (0.2), and hue (0.1). These values were selected to introduce realistic lighting variations while maintaining the semantic integrity of the scenes. Random Horizontal Flip was implemented with a 0.5 probability to enhance spatial diversity

in the training data. We also explored the combined effect of augmentations, specifically implementing a dual transformation of Gaussian Blur and Random Horizontal Flip. Across all augmentation strategies, we maintained a consistent transformation probability of 0.5 and a batch size of 12 to ensure experimental consistency and fair comparison between different approaches. The training methodology remained consistent with our previous domain shift experiments, utilizing BiSeNet architecture with Cross-Entropy Loss and Adam optimizer across 50 epochs. This standardized approach enables direct comparison of the effectiveness of different augmentation strategies while maintaining experimental rigor.

5.2. Results

The experimental results presented in The experimental results presented in Table 3 demonstrate varying effectiveness of augmentation techniques in addressing domain shift between GTA5 and Cityscapes datasets. Gaussian Blur proves most effective with a mIoU of 19.39%, particularly excelling in structural elements like buildings (62.6%) and vegetation (67.3%). This success suggests that reducing synthetic sharpness effectively bridges the visual gap with real-world imagery. Color Jitter shows limited effectiveness with a mIoU of 14.88%, struggling particularly with fine-grained object categories (showing 0% IoU for many classes) and indicating that simple color space transformations inadequately address domain shift challenges. The combination of augmentation techniques achieves an intermediate performance of 16.03% mIoU, but fails to surpass the effectiveness of Gaussian Blur alone, suggesting that additional transformations may interfere with the benefits of edge softening. This is evident in the reduced performance across key categories, with the combination approach achieving lower scores in both building (58.6%) and vegetation (62.4%) compared to Gaussian Blur alone.

6. Adversarial Domain Adaptation

To address the domain shift challenge [5] in real-time semantic segmentation, we propose an Adversarial Domain Adaptation framework integrated with the BiSeNet architecture. This approach aims to learn domain-invariant features that effectively generalize from the source domain (GTA5) to the target domain (Cityscapes) while maintaining BiSeNet's real-time performance capabilities.

Following the methodology outlined in, our framework comprises two primary components working in an adversarial setting:

- **Generator (G):** A real-time segmentation network (BiSeNet) that processes input images from both source and target domains ($I_s, I_t \in \mathbb{R}^{H\times W\times 3}$). This network aims to produce domain-agnostic segmentation maps while maintaining computational efficiency. Copy

- **Discriminator (D):** A neural network that analyzes the generator's output features, attempting to distinguish between source and target domains. This network promotes the learning of domain-invariant features through adversarial training.

The domain adaptation scenario presents a unique challenge: while the source domain (GTA5) provides labeled data for supervised learning, the target domain (Cityscapes) remains unsupervised during training. This setting requires careful consideration, as target domain labels are strictly reserved for evaluation purposes and must not influence the training process.

In domain adaptation, the goal is to align the feature distributions of the source and target domains to improve the performance of the model on the target domain. The key loss functions used in this process are described below.

6.1. Domain Adaptation Losses

Domain adaptation aims to align the feature distributions of the source and target domains to enhance the model's performance on the target domain. The key loss functions involved in this process are as follows:

6.1.1 Segmentation Loss

The segmentation loss ensures that the generator produces accurate segmentation maps for the source domain. It is computed by comparing the predicted segmentation map with the ground truth labels using a cross-entropy loss function. This loss helps the model learn to extract meaningful features from the source domain, improving its ability to generalize.

6.1.2 Discriminator Loss

The discriminator is trained to distinguish between segmentation predictions from the source and target domains. By learning to differentiate between the two, the discriminator plays a crucial role in domain alignment. The loss for the discriminator is computed using a cross-entropy function, ensuring that it can effectively classify whether an input belongs to the source or target domain.

6.1.3 Adversarial Loss

Adversarial loss is used to align the feature distributions of the target domain with those of the source domain. This loss encourages the generator to produce predictions for the

Table 3. Performance comparison of different data augmentations on the GTA5 to Cityscapes dataset

Augmentations	mIoU (%)	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	truck	bus	train	motorbike	bicycle	
Gaussian Blur	19.39	0.566	0.008	0.626	0.030	0.058	0.130	0.026	0.028	0.673	0.042	0.633	0.303	0.002	0.487	0.061	0.005	0.009	0.005	0.000
Color Jitter	14.88	0.614	0.000	0.547	0.038	0.000	0.000	0.000	0.000	0.538	0.048	0.528	0.000	0.000	0.374	0.019	0.000	0.000	0.000	0.000
Combination	16.03	0.557	0.019	0.586	0.010	0.019	0.048	0.000	0.000	0.624	0.086	0.509	0.202	0.010	0.240	0.029	0.058	0.000	0.000	0.000

target domain that are indistinguishable from the source domain predictions. By minimizing this loss, the model reduces the domain gap, allowing it to perform well on the target domain despite domain shifts.

6.2. Training and Validation

6.2.1 Training Setup

The training setup for the generator follows the same configuration as BiSeNet Model. However, we obtained the best results using the Stochastic Gradient Descent (SGD) optimizer with a weight decay of 0.0001 and a learning rate decay power of 0.05.

For the discriminator, we achieved optimal performance using the Adam optimizer with a learning rate of 10^{-5}, following the same learning rate scheduling as the generator but with a learning rate decay power of 0.9. Additionally, we controlled the balance of the adversarial loss using a weighting parameter $\lambda = 0.01$.

6.2.2 Computational Considerations

Due to the computational complexity and resource constraints, training was highly time-consuming. To mitigate these challenges, we achieved our best results by using a reduced batch size of 10 for both models. Moreover, we limited the number of iterations to 70, which is significantly lower compared to the augmentation phase, where training duration matched the length of the training dataset.

6.3. Adversarial Domain Adaptation Results

Table 4 presents the results of adversarial domain adaptation, with an overall **mIoU of 21.64%**. The model demonstrates strong performance on the **road** class, achieving an IoU of **0.807**, and similarly performs well on the **building** and **vegetation** classes, with IoUs of **0.671** and **0.705**, respectively. These results indicate effective adaptation for certain structural and environmental features.

However, the model struggles with several classes, particularly those related to humans (e.g., **person**, **rider**) and vehicles (e.g., **truck**, **bus**, **motorbike**), where the IoU values are close to zero. This suggests that further improvements are needed to better adapt the model to these challenging categories.

Table 4. Performance of Adversarial Domain Adaptation

Metric	**Value**
mIoU (%)	21.64
road	0.807
sidewalk	0.189
building	0.671
wall	0.112
fence	0.068
pole	0.087
traffic light	0.002
traffic sign	0.000
vegetation	0.705
terrain	0.169
sky	0.622
person	0.000
rider	0.000
car	0.596
truck	0.080
bus	0.003
train	0.000
motorbike	0.000
bicycle	0.000

7. Conclusion

In this work, we conducted a comprehensive investigation of domain adaptation techniques for real-time semantic segmentation, addressing the challenge of bridging synthetic and real-world urban scenes. Through systematic evaluation of both classical (DeepLabV2) and real-time (BiSeNet) architectures, we established benchmarks and quantified the impact of domain shift between GTA5 and Cityscapes datasets. Our analysis revealed that carefully designed data augmentation strategies can significantly mitigate domain shift effects, with Gaussian Blur emerging as the most effective technique, achieving a mIoU of 19.39%. Furthermore, our implementation of adversarial domain adaptation demonstrates the potential of learning domain-invariant features while maintaining real-time performance. These insights contribute to the broader understanding of domain adaptation in semantic segmentation and provide practical strategies for improving the robustness of real-time segmentation models in real-world applications.

References

[1] Liang-Chieh Chen, George Papandreou, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. In *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, 2018. 1

[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*, pages 3213–3223, 2016. 1

[3] Shijie Hao, Yuan Zhou, and Yanrong Guo. A Brief Survey on Semantic Segmentation with Deep Learning. *arXiv preprint*, 2023. 2

[4] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for Data: Ground Truth from Computer Games. In *European Conference on Computer Vision*, pages 102–118, 2016. 1

[5] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. DACS: Domain Adaptation via Cross-domain Mixed Sampling. *arXiv preprint*, 2021. 6

[6] 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. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[7] Yanchao Yang and Stefano Soatto. FDA: Fourier Domain Adaptation for Semantic Segmentation. *arXiv preprint*, 2020. 4

[8] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation. In *European Conference on Computer Vision*, 2018. 2

[9] 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