

000  
001  
002  
003  
004  
005  
006  
007  
008  
009  
010  
011  
012  
013  
014  
015  
016  
017  
018  
019  
020  
021  
022  
023  
024  
025  
026  
027  
028  
029  
030  
031  
032  
033  
034  
035  
036  
037  
038  
039  
040  
041  
042  
043  
044  
045  
046  
047  
048  
049  
050  
051  
052  
053

054  
055  
056  
057  
058  
059  
060  
061  
062  
063  
064  
065  
066  
067  
068  
069  
070  
071  
072  
073  
074  
075  
076  
077  
078  
079  
080  
081  
082  
083  
084  
085  
086  
087  
088  
089  
090  
091  
092  
093  
094  
095  
096  
097  
098  
099  
100  
101  
102  
103  
104  
105  
106  
107

# Semantic Segmentation

Guillermo Jose' Gallucci  
331589

Nicola Bavaro  
328790

Lorenzo Gargasole  
331409

July 2, 2024

## Abstract

This project explores the use of synthetic images, derived from the game GTA5, to train a semantic segmentation model that can perform effectively on real images from the Cityscapes dataset. The use of synthetic images offers a significant advantage in terms of speed and simplicity in labeling, since, being generated from a video game, pixel-per-pixel annotations are inherently available, unlike real images that require laborious manual labeling. Initially, we performed training and validation of the DeepLabV2 model using only real images from Cityscapes, followed by similar experiments with the BiSeNet model. Subsequently, BiSeNet was trained and validated on synthetic images from GTA5, showing significant performance limitations when tested on real images. To overcome these difficulties, we implemented data augmentation techniques and subsequently applied domain shift methodologies, including Fourier Domain Adaptation for Semantic Segmentation (FDA) and Domain Adaptation via Cross-domain Mixed Sampling (DACS). These approaches enabled improved model performance on real data, bringing us closer to the goal of a robust semantic segmentation model trained on synthetic images but effective on real images.

**Keywords:** Real-time Semantic

Segmentation, Bilateral Segmentation Network

## 1. Introduction

Semantic segmentation plays a crucial role in many digital image processing applications. For example, in the area of object recognition, segmentation enables the isolation and identification of individual objects within an image, thereby improving the accuracy and reliability of recognition systems. Furthermore, in areas such as autonomous driving, semantic segmentation is critical for autonomous vehicles to identify and understand their surroundings. In the context of this project, the images used -sourced from the CityScapes and GTA5 datasets- are perfectly suited for this purpose, as they are captured from the perspective of

a car driver, reflecting exactly what a machine vision system needs to analyze and understand to support applications such as autonomous driving. In the field of computer vision, semantic segmentation poses a significant challenge due to the complexity of images and the need to process large amounts of data in real time. However, overcoming this challenge is crucial to developing intelligent systems that can understand and interact with their surroundings in a similar way to humans. Our research pipeline follows a well-defined time sequence. Initially, we proceed with the evaluation of DeepLabv2 and BiSeNet models, pre-trained on ImageNet, on Cityscapes datasets. We perform training and validation on Cityscapes for both models in order to evaluate their baseline performance on real data. Next, we proceed with training the BiSeNet model on the GTA5 dataset, followed by evaluating its performance on Cityscapes. This step allows us to explore the effects of domain shift between the two datasets and to identify any problems in generalizing the model. After identifying the limitations due to domain shift, we explore the effectiveness of augmentation techniques when training the BiSeNet model on GTA5. Next, we evaluate the performance of the improved model on Cityscapes to determine whether augmentations helped reduce the effect of domain shift and improve the generalization of the model to real data. Next, we select the augmentations that produced the best results and apply advanced domain adaptation techniques, such as Fourier Domain Adaptation (FDA) and Domain Adaptation via Cross-domain Mixed Sampling (DACS), to further refine the performance of the BiSeNet model when trained on GTA5 and validated on Cityscapes. This structured approach allows us to systematically address the challenges posed by domain shift, progressively improving the capabilities of the semantic segmentation model in moving from synthetic to real images.

## 2. Related Work

In the past decade, many effective semantic segmentation systems were based on manually designed features

paired with straightforward classifiers like **Boosting**, **Random Forests**, and **Support Vector Machines**. However, the performance of these systems was often constrained by the limited expressive power of hand-crafted features [1], it means that these features were not capable of capturing the complexity of the image data as needed and often lacked the ability to represent patterns and details in the images, which constrained the performance of the segmentation systems. Nevertheless, recently, the breakthroughs in Deep Learning for image classification have been rapidly applied to semantic segmentation. Given that semantic segmentation involves both segmentation and classification, a key challenge is effectively combining these tasks [1]. In order to achieve these objectives, we utilize BiSeNet, with an additional consideration given to DeepLabv2 to set an upper bound to image classification performance on the Cityscapes dataset.

## DeepLabv2

Is a Deep Convolutional Neural Network (DCNN) [1] initially trained for image classification, now used for semantic segmentation by converting all fully connected layers to convolutional layers and increasing feature resolution through atrous convolutional layers that allow us to compute feature responses every 8 pixels instead of every 32 pixels in the original network. The model introduces three main innovations: atrous convolution, atrous spatial pyramid pooling (ASPP), and the use of a fully connected conditional random field (CRF) to refine segmentation results.

- **Atrous Convolution for Dense Feature Extraction and Field-of-View Enlargement:** Atrous convolution, or convolution with upsampled filters, allows control over the resolution at which feature responses are computed without increasing the number of parameters or the amount of computation. This is done by inserting spaces (or "trous") between filter weights, expanding the filter's field of view.

This method is crucial for obtaining high-resolution feature maps, particularly when the max-pooling layers in DCNNs do not perform downsampling. Once the feature maps are generated, they are upscaled to the original image resolution using bilinear interpolation. The mathematical formulation for atrous convolution is:

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

where  $y[i]$  is the output,  $x[i]$  is the input,  $w[k]$  is the filter,  $K$  is the filter size, and  $r$  is the expansion rate. The rate parameter  $r$  corresponds to the stride with which we sample the input signal; for example, standard convolution is a special case for rate  $r = 1$ . Atrous convolution also allows us to arbitrarily enlarge the field-of-view of filters at any DCNN layer. In fact, with rate

$r$ , it introduces  $r - 1$  zeros between consecutive filter values, effectively enlarging the kernel size of a  $k \times k$  filter to  $k_e = k + (k - 1)(r - 1)$  without increasing the number of parameters or the amount of computation [1].

- **Atrous Spatial Pyramid Pooling (ASPP):** ASPP tackles the problem of segmenting objects at various scales by utilizing atrous convolutions with different dilation rates on a feature map. This technique gathers information at multiple scales without the need to rescale the image repeatedly, enhancing computational efficiency. ASPP employs several parallel atrous convolutional layers, each with a distinct sampling rate, to capture objects and context across different scales.
- **Upsampling with Bilinear Interpolation:** Following feature extraction and the application of ASPP, the generated feature maps remain at a lower resolution than the original image. Bilinear interpolation is employed to upscale these maps back to the original image resolution. This process is essential to match the model's output size with the input image, maintaining the spatial details required for precise segmentation.
- **Fully Connected Conditional Random Field (CRF):** Combining DCNNs with CRFs improves boundary localization accuracy. Due to their invariant nature, DCNNs often lose spatial details. However, the fully connected CRF leverages Gaussian potentials to capture long-range dependencies and enhance edge details.

The main advantages are: Efficiency, thanks to the atrous convolutional layers and Accuracy: Excellent results on several datasets, including Cityscapes [1].

In this work, we employ a residual net variant of DeepLab adapted from ResNet and pre-trained with ImageNet weights. As previously mentioned, DeepLab establishes as a baseline for semantic segmentation on the Cityscapes dataset and allows performance comparison, particularly in terms of FLOPS and Latency, with another DCNN, BiSeNet.

## BiSeNet

It is used the Bilateral Segmentation Network with two parts: Spatial Path (SP) and Context Path (CP)

- **Spatial Path:** Manages spatial information within images, a critical aspect for accurately predicting detailed outputs. Preserves the spatial size of the original input image and encode spatial information with detailed convolution; contains three layers, each layer includes a convolutional with stride = 2 followed by batch normalization and ReLU. The three convolutional layers keep low the model's computational load and encodes

216 rich spatial information due to the large spatial size of  
 217 feature maps.  
 218

### RESNET 18

- 219 • Context Path: Contextual information enhances the  
 220 generation of high-quality results. Designed to provide  
 221 sufficient receptive field utilizing a lightweight model  
 222 and global average pooling, the model can downsample  
 223 rapidly the feature map fast to obtain a large receptive  
 224 field, which encodes high level semantic context  
 225 information. Then it was added a global average  
 226 pooling on the tail of the lightweight model which  
 227 can provide maximum receptive field with global  
 228 context information. The Context Path contains the  
 229 Attention Refinement Module (**ARM**) to refine features  
 230 at each stage. Utilizes global average-pooling to cap-  
 231 ture global context and computes an attention vector  
 232 to guide feature learning, thereby refining the output  
 233 features at each stage within the Context Path.  
 234

235 It achieves high accuracy and efficiency in performance  
 236 because the paths are computed concurrently and they are  
 237 complementary to each other.  
 238

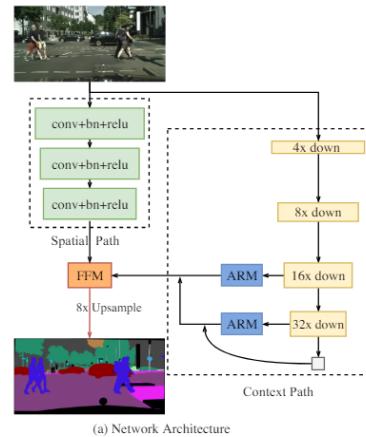
Finally, the output features of both paths are fused to  
 make the final prediction like it is shown in Fig. 1 and Fig.  
 2.

The feature information captured by the Spatial Path pri-  
 marily encodes rich detailed information, whereas the out-  
 put feature of the Context Path predominantly encodes con-  
 textual information. In simpler terms, the output feature of  
 the Spatial Path is low-level, while that of the Context Path  
 is high-level. This implies that directly summing these fea-  
 tures may not be appropriate or effective. Therefore, it is  
 proposed the Feature Fusion Module to fuse these features:  
 First, the concatenation of the output features from the Spa-  
 tial Path and Context Path. Following this, batch normaliza-  
 tion is applied to ensure the features are appropriately  
 scaled. Subsequently, average-pool of the concatenated fea-  
 ture into a feature vector and compute a weight vector. This  
 weight vector facilitates the re-weighting of features, effec-  
 tively performing feature selection and combination [2]

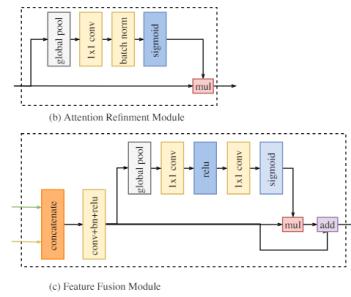
We also used techniques to improve the performance metric, like Data Augmentation, FDA and DACS.

## 2.1. Data Augmentation

We conducted several experiments to identify the optimal combination of data augmentation strategies. Each one aimed to diversify the training dataset, thus improving the model's predictions. Combining these different techniques, we selected the approach that yielded the most significant improvement. These augmentation techniques significantly enhanced the model's ability to generalize, leading to better performance on unseen data.



285 Figure 1. Proposed BiSeNet Architecture [2]  
 286



297 Figure 2. Proposed Attention Refinement Module and Feature Fusion Module [2]  
 298

## 2.2. FDA

Prior to delineating Fourier Domain Adaptation, it is per-  
 tinent to revisit fundamental concepts: Fourier transform  
 and Fast Fourier Transform (FFT).

The Fourier Transform is a mathematical transformation  
 that converts a function of time  $f(t)$  into a function of fre-  
 quency  $\mathcal{F}$ . The following describes the continuous Fourier  
 Transform

$$\mathcal{F}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

Where:

- $f(t)$  is the time-domain function
- $\mathcal{F}(\omega)$  is the frequency-domain function
- $\omega$  is the angular frequency in radiant per seconds  
 (rad/s). It can also be written as  $2\pi f$

The Fast Fourier Transform (FFT) is an efficient algo-  
 rithm for computing the Discrete Fourier Transform (DFT)  
 and its inverse. The DFT is the discrete version of the

324 Fourier Transform, applied to a finite sequence of sampled  
 325 values. The formula for the DFT is:  
 326

$$X_f = \sum_{n=0}^{N-1} x_n \cdot e^{-i \frac{2\pi}{N} f n}$$

330 Where:

- $x_n$  is the input sequence of length  $N$ , where  $N$  is the number of samples and  $X_f$  is the transformed sequence,  $f$  is the frequency index

335 The FFT reduces the time complexity from  $O(N^2)$  to  
 336  $O(N \log N)$

337 Now, we can proceed to explain how Fourier Domain  
 338 Adaptation works to mitigate domain shift problems.

339 Involves a simple process: it calculates the Fast Fourier  
 340 Transform (FFT) of each input, the GTA5 image. Then, it  
 341 swaps out the low-level frequencies from target Cityscapes  
 342 images and replaces them with those from the source images.  
 343 Finally, it reconstructs the image for training using  
 344 the inverse FFT (iFFT), while maintaining the original  
 345 annotations in the Grand Theft Auto 5 domain [3].

346 Given a source dataset  $D^s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ , where

- $x^s \in \mathbb{R}^{H \times W \times 3}$  is the color image,  $y^s \in \mathbb{R}^{H \times W}$  is the semantic map associated with  $x^s$ .

347 And given the target dataset containing only color images  
 348 with no ground-truth labels  $D^t = \{x_i^t\}_{i=1}^{N_t}$

349 Let  $\mathcal{F}^A : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{H \times W \times 3}$  and  $\mathcal{F}^P : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{H \times W \times 3}$  be the amplitude and phase of the Fourier Transform  $\mathcal{F}$  of an RGB image, i.e., for a single channel image  $x$  we have:

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

350 Where:

- $\mathcal{F}(x)(m, n)$  is the Fourier Transform of image  $x$  at frequency coordinates  $(m, n)$
- $x(m, n)$  The pixel intensity of the image  $x$  at the spatial coordinates  $(m, n)$
- $h$ : The row index in the spatial domain (height coordinate).
- $w$ : The column index in the spatial domain (width coordinate).

371 This can be implemented using the Fast Fourier Transform [3].

372  $\mathcal{F}^{-1}$  is the inverse Fourier Transform that maps phase  
 373 and amplitude back to image space. Let  $M_\beta$  mask with

374 value zero except for the region where  $\beta \in (0, 1)$ . The  
 375 choice of  $\beta$  does not depend on image size or resolution.

376 Given two randomly images  $x^s \sim D^s$ ,  $x^t \sim D^t$ , FDA  
 377 can be formalized as:

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

381 The low-frequency components of the source image  
 382  $\mathcal{F}^A(x^s)$  are substituted with those of the target image  $x^t$ .  
 383 Subsequently, the modified spectral representation of  $x^s$  is  
 384 transformed back to the image domain, resulting in  $x^{s \rightarrow t}$ .  
 385 This new image retains the content of  $x^s$  but adopts the appearance  
 386 characteristic of a sample from  $D^t$  [3].

- The symbol  $\odot$  represents the Hadamard product,  
 387 which is element-wise multiplication

## 2.3. DACS

391 Domain Adaptation via Cross-domain Mixed Sampling  
 392 (DACS) is an innovative technique designed to tackle the  
 393 domain shift problem in semantic segmentation tasks. This  
 394 challenge often arises when moving from synthetic images,  
 395 to real-world images. DACS falls under the umbrella of  
 396 unsupervised domain adaptation (UDA), where the goal is  
 397 to train on labeled data from a source domain while learning  
 398 from unlabeled data in a target domain.

400 The essence of DACS lies in its approach to creating  
 401 new, augmented training samples. It cleverly combines images  
 402 from both the source and target domains. This involves  
 403 mixing segments of a source domain image, along with its  
 404 ground-truth semantic labels, onto an unlabeled target do-  
 405 main image. By doing this, DACS ensures that the resulting  
 406 mixed image contains features from both domains, making  
 407 the model more versatile.

408 Generating pseudo-labels for these mixed images is an-  
 409 other critical aspect of DACS. It combines the ground-truth  
 410 labels from the source domain with pseudo-labels from the  
 411 target domain. This method ensures that each semantic  
 412 class is well represented throughout the training process,  
 413 which helps to address the common problem of class im-  
 414 balance in UDA.

415 Training with these mixed samples allows the model to  
 416 be exposed to a wide variety of contexts. This exposure  
 417 is crucial for improving the model's ability to generalize  
 418 across different domains. By incorporating elements from  
 419 both the source and target domains, DACS enhances the  
 420 training signal and leverages the strengths of both datasets.

421 One of the standout benefits of DACS is its effective han-  
 422 dling of domain shift. By mixing images from different do-  
 423 mains, it bridges the gap between synthetic and real-world  
 424 data, leading to better model performance on the target do-  
 425 main. The cross-domain mixing ensures that all classes are  
 426 consistently represented during training, which helps miti-  
 427 gate issues related to class imbalance. Additionally, DACS  
 428

432 maintains a robust training signal by incorporating reliable  
 433 ground-truth labels from the source domain into the mixed  
 434 samples, unlike other methods that filter out low-confidence  
 435 pixels.  
 436

437 In summary, DACS is a forward-thinking approach that  
 438 leverages cross-domain image mixing to create robust training  
 439 samples for unsupervised domain adaptation. By blending  
 440 elements from both source and target domains, DACS  
 441 addresses the challenges of domain shifts, ensuring balanced  
 442 class representation and improved generalization.  
 443 This method has proven to be highly effective, setting new  
 444 standards for tasks involving significant domain differences  
 445 between training and application environments [4].  
 446

---

**Algorithm 1** DACS algorithm
 

---

**Require:** Source-domain and target-domain datasets  $\mathcal{D}_S$  and  $\mathcal{D}_T$ , segmentation network  $f_\theta$ .  
 1: Initialize network parameters  $\theta$  randomly.  
 2: **for**  $i = 1$  to  $N$  **do**  
 3:    $X_S, Y_S \sim \mathcal{D}_S$   
 4:    $X_T \sim \mathcal{D}_T$   
 5:    $\hat{Y}_T \leftarrow f_\theta(X_T)$   
 6:    $X_M, Y_M \leftarrow$  Augmentation and pseudo-label from  
    mixing  $X_S, Y_S, X_T$  and  $\hat{Y}_T$ .  
 7:    $\hat{Y}_S \leftarrow f_\theta(X_S), \hat{Y}_M \leftarrow f_\theta(X_M)$    ▷ Compute  
    predictions.  
 8:    $\ell \leftarrow L(\hat{Y}_S, Y_S, \hat{Y}_M, Y_M)$    ▷ Compute loss for the  
    batch.  
 9:   Compute  $\nabla_\theta \ell$  by backpropagation (treating  $Y_M$  as  
    constant.)  
 10:   Perform one step of stochastic gradient descent on  
     $\theta$ .  
 11: **end for**  
 12: **return**  $f_\theta$

---

Figure 3. Dacs Algorithm [4]

To conclude this section, we define the metric used to evaluate the performance of the models: The Mean Intersection over Union (mIoU) evaluates model performance by measuring the overlap between predicted and ground truth labels. It averages the Intersection over Union (IoU) scores across all classes, where IoU for a class is the intersection area of predicted and ground truth regions divided by their union area. Higher mIoU values indicate superior segmentation performance.

### 3. Experiments

The datasets we used are as follows:

- Cityscapes: It consists of 1572 images and corresponding labels for training (only to establish and upper bound performance limit for BiSeNet and DeepLab), each with a resolution of 1024x512 pixels. Additionally, there are 500 images and labels for validation, with the same dimension. These data are exclusively utilized to establish an upper performance limit for BiSeNet and DeepLabv2.

- GTA5: This dataset contains 2500 images and their corresponding labels for training, each with a resolution of 1280x512 pixels.

486  
 487  
 488  
 489

#### 3.1. Classic Semantic Segmentation network: DeepLabV2

In this experiment, we trained a classic segmentation network, DeepLabV2, using the Cityscapes dataset. The primary objective was to evaluate the model's performance in terms of segmentation accuracy and computational efficiency. The Cityscapes dataset was employed for both training and testing phases. We trained the model for 50 epochs, ensuring sufficient time for the model to converge. Both the training and testing resolutions were set at 1024x512, matching the native resolution of the Cityscapes images. We utilized the ResNet-101 (R101) backbone, which had been pre-trained on ImageNet, to extract Meaningful features from the images. The model was trained to recognize 19 semantic classes present in the Cityscapes dataset. The evaluation metrics included Mean Intersection over Union (mIoU), latency, Floating Point Operations per second (FLOPs), and the number of parameters. Table 1. shows the results



Figure 4. Image with predicted label of Cityscapes from DeepLabv2

Cityscapes	mIoU (%)	Latency(ms)	FLOPs	Params
DeeplabV2	56.99	246.17	0.375T	43.90M

Table 1. Performance metrics of DeepLabV2 on the Cityscapes dataset.

510  
 511  
 512  
 513  
 514  
 515

#### 3.2. Real-Time Semantic Segmentation Network: BiSeNet

We focused on training the BiSeNet segmentation network using the Cityscapes dataset. As before, the training process spanned 50 epochs, with both training and testing resolutions set to 1024x512. Table 2. illustrates the results.

Cityscapes	mIoU (%)	Latency(ms)	FLOPs	Params
BiSeNet	52.49	17.14	25.78G	12.58M

Table 2. Performance metrics of BiSeNet on the Cityscapes dataset.

516  
 517  
 518  
 519  
 520  
 521  
 522  
 523  
 524  
 525  
 526  
 527  
 528  
 529  
 530  
 531  
 532  
 533  
 534  
 535  
 536  
 537  
 538  
 539



Figure 5. Image with predicted label of Cityscapes from BiSeNet

### 3.3. Evaluating the Domain Shift problem

Our experiments addressed the domain shift problem in semantic segmentation by training a real-time segmentation network on synthetic images from GTA5 and evaluating its performance on real images from Cityscapes. Training on synthetic images like those from GTA5 is advantageous due to several reasons. Firstly, synthetic datasets offer a cost-effective alternative to manually annotated real-world images, reducing the financial burden associated with data acquisition. Secondly, synthetic datasets provide control over various factors such as lighting conditions, object appearances, and environmental settings, enabling more systematic experimentation and model tuning. Lastly, training on synthetic data allows for the creation of diverse and abundant annotated datasets, which are essential for training deep learning models effectively. During training on GTA5, the network learned to segment objects within simulated environments, adapting to the synthetic characteristics of the dataset. However, when evaluated on real images from Cityscapes, the network faced challenges due to differences in lighting, weather conditions, and object appearances not encountered during training. Despite these disparities, the network exhibited commendable adaptability, albeit with a slight performance degradation compared to intra-domain training on Cityscapes alone. We trained our model for 50 epochs as per our standard practice. Training utilized images from GTA5 at a resolution of 1280x720, while testing was conducted on Cityscapes images at a resolution of 1024x512. We employed a ResNet18 backbone pre-trained on ImageNet, consistent with previous training setups.

### 3.4. Data Augmentations to reduce the domain shift

To mitigate the domain shift problem and enhance the generalization capability of our segmentation network trained on synthetic data, we explore the utilization of data augmentations during training. These augmentations serve two main purposes: virtually expanding the dataset size, and modifying the visual appearance of synthetic images to make them more akin to real-world counterparts. In our approach, we replicate the previous experiment while incorporating data augmentations during training. We devised three augmentation strategies:

- Horizontal Flip (Aug1): This basic augmentation tech-

nique involves flipping images horizontally. By mirroring images with a probability of 0.5, we aim to increase dataset variability and enhance the network's ability to generalize across different orientations.

- Color Jitter, Gaussian Blur, and Gamma Correction (Aug2): combines multiple augmentation techniques, including color jitter, Gaussian blur, and Gamma correction. Color jitter introduces random color variations to images, mimicking real-world lighting conditions and color shifts. Gaussian blur simulates the effects of camera focus and motion blur, contributing to a more realistic appearance. Gamma correction adjusts image brightness, further diversifying the dataset.
- Aug1+2 (Augmentation Union): represents the combination of Aug1 and Aug2.

We introduce these augmentations with a probability of 0.5 during training, in order to diversify the training dataset and simulate variations encountered in real-world images. This approach seeks to improve the network's robustness to domain shift by exposing it to a broader range of visual scenarios during training. Looking at the results in the table below, it becomes evident that the incorporation of each data augmentation technique has led to an enhancement in the mIoU of the classes. This indicates that the utilization of augmentation strategies during training has positively influenced the model's ability to generalize to images from the target domain. Particularly noteworthy is the substantial increase in mIoU, reaching 23.13%, achieved by the combination of both the augmentation strategies.

### 3.5. Domain Adaptation

Despite the significant improvement achieved by the data augmentation techniques, the model trained on synthetic images from the GTA5 dataset continued to show difficulties in performing on real images from the Cityscapes dataset. This highlighted the existence of a domain shift between the two datasets that augmentation techniques alone were not sufficient to completely bridge. To address this problem, we decided to implement domain adaptation techniques, focusing on an image-to-image approach. Our choice fell on two distinct techniques: Fourier Domain Adaptation (FDA) and Domain Adaptation via Cross-domain Mixed Sampling (DACS). FDA relies on transforming synthetic images in the Fourier domain to fit the statistical characteristics of real images, thus reducing the gap between the two domains. DACS, on the other hand, combines samples from both domains during training to create a mixed dataset that facilitates model fitting. In the following sections, we will describe their basic principles, the implementation process, and the results obtained.

594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647

648

### 3.5.1 Fourier Domain Adaptation (FDA)

650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668

After noticing the challenges in adapting the model trained on the synthetic images of GTA5 to the context of real images of Cityscapes, we decided to implement Domain Adaptation via Fourier (FDA) as part of our adaptation strategy. Using FDA in combination with Aug1+2 augmentation techniques during training, we worked on the GTA5 images while maintaining a constant resolution of 1280x720 pixels. This approach allowed us to make the GTA5 images much more similar to Cityscapes images, creating a training environment that better reflects the characteristics of the target domain. After 50 training epochs, we observed a significant improvement in the mIoU (Mean Intersection over Union) of the model. The mIoU increased to about 26.29%, registering a 3.16% increase compared to the initial phase of training. This result demonstrates the effectiveness of the combined approach of FDA and augmentation techniques in facilitating the adaptation of the model to the target domain images, enabling better generalization and superior performance in semantic segmentation of Cityscapes images.

669  
670  
671  
672

### 3.5.2 Domain Adaptation via Cross-domain Mixed Sampling (DACS)

673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697

To integrate Domain Adaptation via Cross-domain Mixed Sampling (DACS) into our approach, we adopted an innovative strategy to further improve the performance of our semantic segmentation model. Using DACS, we mixed images from different datasets to create a mixed dataset that incorporates features from both the synthetic GTA5 dataset and the real Cityscapes dataset. With DACS, we leveraged the real semantic maps from the GTA5 dataset to select specific classes and add them to unlabeled images from Cityscapes. Next, we created pseudo-labels for these new images by combining GTA5 labels with pseudo-labels generated from Cityscapes images. The results obtained from the application of DACS were remarkable. Using this technique, we were able to achieve a significant improvement in the Mean Intersection over Union (mIoU) of our model. Specifically, we went from an mIoU of 23.13% obtained using only Augmentation 1 and 2 (Aug1+2) techniques to an mIoU of 32.12%. This represents an increase of almost 10%, demonstrating the effectiveness of DACS in facilitating model fitting to target domain images, enabling better generalization and superior performance in semantic segmentation of Cityscapes images.

698  
699  
700  
701

## 4. Results

Our exploration into semantic segmentation across synthetic and real-world datasets has yielded significant insights and advancements. Initially, training our DeepLabV2 model on the Cityscapes dataset, using a batch\_size=2,

GPU P100 and the Adam Optimizer with a starting learning\_rate=1e-3, we achieved a baseline Mean Intersection over Union (mIoU) of 56.9%. This served as a robust starting point, showcasing the model's capabilities in a controlled, real-world context with well-annotated data.

On the other hand, BiSeNet, when trained on Cityscapes with a batch\_size of 4 and the same initial learning\_rate, achieves a mIoU of 52.49, demonstrating its potential efficiency in a real-world environment similar to its training environment.

Transitioning to the synthetic GTA5 dataset, using BiSeNet with a batch\_size=4, GPU P100 and the Adam Optimizer with a starting learning\_rate=1e-3, presented a shift in domain characteristics, leading to a notable decrease in performance when evaluated on Cityscapes images, with the mIoU dropping to 15.12% as it is demonstrated in Fig. 6. This stark contrast highlighted the challenges posed by domain discrepancies, such as variations in lighting, object appearances, and environmental settings between synthetic and real data.

To address these challenges, we employed innovative approaches aimed at bridging the domain gap and enhancing model adaptability. Introducing data augmentation techniques (Aug1+2), this time using a batch\_size=8, thank to the double GPU T4, provided an initial boost, improving the mIoU to 23.13% by diversifying the training data and simulating real-world image variations, Fig. 8. illustrates this result.

Further enhancements were achieved through Fourier Domain Adaptation (FDA), using a batch\_size=4 and double GPU T4, which leveraged frequency domain transformations to align synthetic GTA5 images more closely with the characteristics of Cityscapes, a graphical representation of FDA is visible in Fig. 9. This approach resulted in a noticeable performance increase, pushing the mIoU to 26.42% and demonstrating the effectiveness of domain adaptation strategies in improving model generalization just as Fig. 10 shows.

However, the most substantial improvement came with Domain Adaptation via Cross-domain Mixed Sampling (DACS). By integrating features from both synthetic GTA5 and real Cityscapes datasets, DACS (batch\_size=4 and double GPU T4) facilitated a significant boost in performance. The mIoU rose to 32.12%, marking a nearly 10% increase compared to the Aug1+2 augmentation strategy alone. This approach leveraged pseudo-labeling and cross-domain feature mixing to enhance the model's ability to generalize across diverse datasets. Fig. 11. illustrates an example of the outputs of the DACS method, utilizing Cityscapes target images along with their corresponding ground truth labels. Conversely, Fig. 12. presents our outputs of the BiSeNet model after being trained with GTA5 images and labels, subsequent to the application of the DACS technique

756	Method	mIoU (%)	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	MC	Bike	810
757	No Augmentation	15.12	8.35	1.28	37.40	1.79	6.50	12.80	13.94	2.23	73.50	12.75	57.97	23.61	0.00	30.02	2.13	3.02	0.01	0.00	0.00	811
758	Augmentation 1	17.57	34.04	4.02	44.98	9.63	4.14	13.80	10.46	1.90	70.68	6.90	61.60	29.78	0.41	36.46	3.95	0.42	0.00	0.64	0.00	812
759	Augmentation 2	21.00	31.06	16.28	61.84	9.90	5.27	20.37	19.45	4.87	72.82	5.76	80.61	30.03	0.14	28.84	10.32	0.00	0.00	1.52	0.00	813
760	Augmentation 1+2	23.13	42.58	13.03	71.36	16.10	10.04	17.33	16.68	4.92	77.31	17.52	80.69	33.45	1.21	20.41	9.58	2.81	0.00	4.60	0.00	814
761	FDA	26.42	77.65	21.51	73.23	19.93	6.20	21.46	12.25	5.03	79.66	20.03	80.60	29.22	0.45	41.41	7.24	2.88	0.00	3.32	0.04	815
762	DACS	32.12	88.29	30.86	76.98	21.58	13.70	25.79	13.77	14.96	70.64	13.24	64.43	48.12	0.69	80.59	14.54	21.45	7.02	3.67	0.00	816

Table 3. Operations to solve domain shift problem, all results are in percentage

with only Cityscapes images .

All the results, including mIoU and percentage for each class can be seen in Table 3.



Figure 6. BiSeNet prediction on Cityscapes after training with GTA5 images

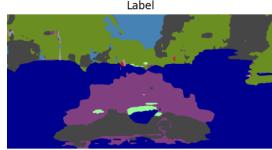


Figure 7. BiSeNet prediction on Cityscapes after training with GTA5-augmentation 12

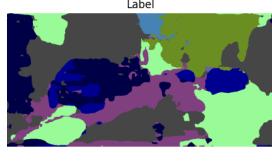
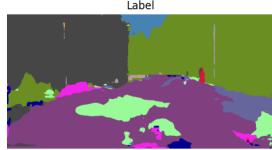


Figure 8. GTA5 Image with and without FDA applied, and the Cityscapes image from which frequencies were taken.



Figure 9. Prediction of Cityscapes using BiSeNet trained on GTA5 images after the FDA application



## 5. Conclusion

In this report, we explored various methods to enhance semantic segmentation performance, particularly addressing the challenges posed by domain shift between synthetic and real-world data. We addressed the domain shift



Figure 10. Application of DACS using Cityscapes ground truth labels

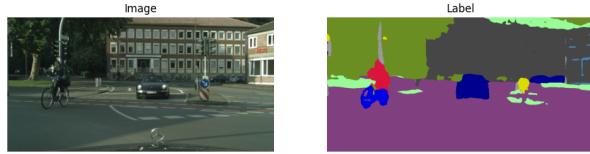


Figure 11. Prediction of Cityscapes using BiSeNet trained on GTA5 images after the DACS application

problem by training on synthetic GTA5 data and testing on Cityscapes. Through our experiments, we found that combining advanced data augmentation and domain adaptation techniques significantly improves the model's ability to generalize across different domains. Data augmentation techniques significantly improved the model's generalization capabilities. FDA and DACS further reduced the domain gap, with DACS achieving the highest improvement in mIoU. These results highlight the efficacy of integrating different datasets and applying adaptive techniques to enhance the robustness and generalization capability of segmentation models.

## References

- [1] “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFS”, Liang-Chieh Chen, George Papandreou, Kevin Murphy, Alan L. Yuille <https://arxiv.org/pdf/1606.00915.pdf>
- [2] “BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation”, Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, Nong Sang <https://arxiv.org/pdf/1808.00897.pdf>
- [3] “FDA: Fourier Domain Adaptation for Semantic Segmentation”, Yanchao Yang, Stefano Soatto, <https://arxiv.org/pdf/2004.05498.pdf>
- [4] “DACS: Domain Adaptation via Cross-domain Mixed Sampling”, Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, Lennart Svensson, <https://arxiv.org/pdf/2007.08702.pdf>