Real-time Domain Adaptation in Semantic Segmentation

Andrea Baccolo
Politecnico di Torino
Turin, Italy
s329570@studenti.polito.it

Fabio Daniele Diena
Politecnico di Torino
Turin, Italy
s332743@studenti.polito.it

Federico Olivero
Politecnico di Torino
Turin, Italy
s332808@studenti.polito.it

Abstract

One of the biggest challenge in Semantic Segmentation is labeling real-world datasets, since every pixel requires a label. In order to solve this problem we used a synthetic dataset from the famous videogame "GTA V", whose images have already been labeled by the creators of the game. Training deep learning models on this dataset and using them in real situations introduces the problem of domain shift. In this work, we firstly analyzed two different neural networks, Deeplabv2 and BiSeNet, on a real-world dataset, "Cityscapes", in order to evaluate which one would perform better in a real-time setting. Then we trained the BiSeNet network on the GTA V dataset and evaluated the training on Cityscapes. In conclusion , we evaluated the problem of domain shift and we tried to improve performances using data augmentations and the adversarial approach, which consists in trying to fool a neural network trained to distinguish real and synthetic images.

1. Introduction

Semantic Segmentation is a fundamental task in computer vision that involves partitioning an image into semantically meaningful regions [2]. Unlike traditional image classification, which assigns a single label to an entire image, semantic segmentation assigns a label to each pixel, thus providing a more detailed understanding of the scene. This pixel-level classification enables a wide range of applications, including autonomous driving, medical image analysis and scene understanding, but it also introduces new challenges. For instance, some of these applications require a model able to make instant predictions: in tasks such as autonomous driving, the model must quickly and accurately identify objects and obstacles in real-time to ensure a safe navigation. Unfortunately, there is often a trade-off between inference speed and accuracy. In our work we trained and evaluated two different neural networks, DeepLabV2 [1] and BiSeNet [7], on the Cityscapes dataset and we will empirically show how it is not necessary to sacrifice high-level segmentation in order to achieve a good performance in terms of speed. Then we used the selected network to deal with the other main challenge: the unlabeled datasets. Indeed, labeling large datasets in real-world situations is extremely computationally expensive and time-consuming: it takes around 90 minutes to label a single image from the "Cityscapes" dataset. An alternative solution is to train a model on another related large-scale source domain with labels, and apply it to the unlabeled target domain: video games datasets, such as GTA V, are widely used for this task, since they have already been labeled by developers. Although this approach reduces the time and the computational capacity needed, it introduces the problem of domain shift [8]: the discrepancy between the source domain and the target domain can significantly degrade the performance of models, as they often fail to generalize well to unseen data. In order to limit this discrepancy two main approaches have been taken into account: firstly we applied Data Augmentations to the synthetic dataset with the intent of making the two domains more similar; then we implemented an Adversarial Approach [4], where two different networks are used simultaneously: a generator model attempt to deceive a discriminator network, transforming the images and reducing the difference between the synthetic and the real ones.

2. Related Work

Representative methods towards domain adaptation and real-time requirements are discussed separately in this section.

2.1. PIDNet

A recent network applied in real-time semantic segmentation is PIDNet [5]. PIDNet is inspired by Proportional Integral-Derivative Controllers, which are widely employed in controlling specific resources that are continuously acquired and used. The main idea is to extend the usual two-branch network to a three-branch architecture containing details, context, and boundary information, respectively. The detail branch correspond to Proportional (P) and it pre-

serves details through high-resolution feature maps. The context branch correspond to the Integral (I). It considers both globally and locally information to parse long-range dependencies and it generally has the most semantic information among branches. Lastly the boundary branch correspond to Derivatives (D) and it marks boundary regions by extracting high-frequency features. To maintain a real-time implementation, all three branches are set to be moderate, deep and shallow. The I branch is crucial for both boundary detection and detail parsing: due to this argument it is a bit longer and it provides information through addiction for the D branch and through a PAG module for the P branch. The PAG module calculates similarity between P and I: an higher value correspond to a deeper consideration of the context branch, while a lower value correspond to a greater focus on details. Finally a BAG module merges the outputs from the three branches, aiming to consider more detail among boundaries.

2.2. Domain Adaptation

Regarding Unsupervised Domain Adaptation, two up to date techniques are Fourier Domain Adaptation (FDA [6]) and Domain Adaptation via Cross-domain Mixed Sampling (DACS [3]). FDA is motivated by the observation that amplitude considers domain information and it has a powerful impact on predictions without altering the perception of semantics, that is coded in the phase part. The authors exploited those features in the following way: firstly applying a Fourier transform to both target and source images, secondly swapping the low frequency component of the source image with the target one, lastly employing the inverse Fourier transform to obtain the new image used in training. It is worth noticing the simplicity of this technique, since it does not require any model implementation, unlike the adversarial approach in [4].

On the other hand, DACS introduces a technique involving cross-domain mixed sampling. This approach aims to enhance the adaptation process by synthesizing new training samples that combine features from both the source and target domains. New, augmented samples are created by having one set of pixels coming from a source domain image, and one set of pixels coming from a target domain image, using the mixing strategy ClassMix: half of the classes in a source domain image are selected, and the corresponding pixels are cut out and pasted onto an image from the target domain. Note that the resulting mixed images are not necessarily realistic, since complete realism of the augmentations is not required for the functioning of the method.

3. Methods

In this section, we initially explain how Deeplabv2 and BiSeNet work, with a focus on how DeeplabV2 enlarge the field of view and how BiSeNet efficiently achieve real-time inference without renouncing to spatial resolution. Then we analyze the domain shift problem: we firstly applied some data augmentations on the GTA V images, trying to manually reduce the difference between the source and the target domain, and then we implemented the Adversarial approach.

3.1. DeeplabV2

Deeplabv2 is a deep convolutional neural network that introduced innovations in feature resolution. Traditional networks often reduce spatial resolutions through sequences of max pooling and down-sampling. Deeplabv2 addresses this issue with atrous convolution, which skips certain pixels during convolution to achieve a larger point of view. Mathematically is described in the following way:

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

where x is the input signal, y is the result of the operation, w is the filter and r is the rate that decide how many pixels are skipped. To detect objects at multiple scales without the computational expense of repeatedly resizing images, Deeplabv2 employs Atrous Spatial Pyramid Pooling (ASPP). ASPP utilizes multiple parallel atrous convolutional filters with different rates, providing varied visual perspectives. Furthermore, to improve predictions on borders, Deeplabv2 incorporates a fully-connected Conditional Random Field (CRF). This component minimizes an energy function that considers both the label assignment probability of pixel *i* from the previous part of the network and the difference between every pair of pixels (*i*,*j*) in terms of proximity and color.

3.2. BiSeNet

The Bilateral Segmentation Network (BiSeNet) has been developed for real-time Semantic Segmentation. BiSeNet consists of two main paths: a Spatial path, which captures spatial information, and a Context path, which provides adjustable receptive fields. The spatial path comprises three layers, each consisting of a convolution with a stride of two, a batch normalization and a ReLu activation. The Context path involves four down-sampling operations to obtain a large receptive field and incorporates an Attention Refinement Module (ARM) at each stage to enhance feature refinement, performing global average pooling to capture global context. Finally, the two paths converge through a Feature Fusion Module (FFM). This component concatenates the two inputs and performs convolution, batch normalization and ReLU to balance their scales. It then uses global average pooling with a sequence of convolution 1x1 and non-linearity functions.

Figure 1. Original image from GTA V

Figure 2. Color Jitter

Figure 3. Gaussian Blur

3.3. Data Augmentation

Data augmentations are highly effective regularization techniques for neural networks. They improve robustness against dataset biases by applying transformations to images, virtually expanding the dataset size or adjusting the visual appearance of synthetic data to better resemble real-world examples. After a deep inspection of the datasets, we noticed two main differences: the GTA V images seemed to be brighter and sharper than the Cityscapes ones. In order to reduce this gap, we firstly applied the Color Jitter technique: this transformation employ random filters which modify hue, saturation, brightness and contrast of the image. During training, it is applied as a preprocessing step to each batch in a randomized manner. This helps the model learn to generalize better by being exposed to a wider range of visual variations and it simulates different lighting conditions, camera settings, and environmental factors that images might encounter in real-world scenarios. We then used the Gaussian Blur transformation to reduce the high resolution of the game images, making them appear more realistic. The Gaussian Blur transformation perform a convolution with a Gaussian kernel to blur the image. Convolutions can be thought of as a type of moving weighted average, where the Gaussian kernel places more weight on the central pixel, diminishing the importance of pixels further away from the center. In parallel to these transformations, we implemented the Random Horizontal Flip technique: it consists in randomly (with a frequency of a given probability) flip some images, increasing the robustness of the dataset. We have not implemented a Random Vertical Flip because we assumed that in autonomous driving images turned upside down are not useful.

3.4. Adversarial Approach

Other approaches to reduce domain shift are Adversarial Discriminative Models, as the one described in [4]. Their main idea is to employ an adversarial objective with respect to a domain discriminator. There are two main network involved: the segmentation network and the discriminator. The discriminator is trained to predict from which domain a given image comes from. It is composed by 5 convolutional layers with kernel size 4, stride 2 and padding 1, a leaky ReLU with negative slope equal to 0.2, an upsample layer and a Sigmoid function. The segmentation network that we adopted is BiSeNet, the goal is to train the Segmentation network to generate an output that is difficult to distinguish for the discriminator. To this purpose, [4] divided the loss function in two part: a Segmentation loss ($\mathcal{L}_{seg}(I_s)$) that consider the segmentation task and an Adversarial loss ($\mathcal{L}_{adv}(I_t)$) weighted by the constant λ_{adv} in the following way:

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

where I_s is a generic image from the source and I_t from the target. Fixing $\mathcal{L}_{seg}$, the more the segmentation network is able to fool the discriminator, the less $\mathcal{L}_{adv}$ will be and consequently, it will produce a lower value of the final loss function.

4. Experimental results

We did our experiments using batches with size equal to 4 and considering a number of training epochs equal to 50. We considered two datasets: Cityscapes with resolution 1024×512 and GTAV with resolution 1280×720. As optimizer we used Stochastic Gradient Descent except for the discriminator in Adversarial approach which adopts Adam optimizer.

We decreased the learning rate value linearly across epochs by raising a coefficient dependent on the current epoch and the maximum number of epochs to the power of 0.9. We used DeeplabV2 with backbone ResNet 101 and BiSeNet with backbone ResNet 18. The backbones were pre-trained on ImageNet. Moreover we considered different initial learning rate for BiSeNet and DeeplabV2: 2.5e-4 and 1e-4.

The code and model are available at https://github.com/federico2879/MLDL2024_semantic_segmentation.git

4.1. Network Comparison

We used Cityscapes dataset for both training and test.

Classic	mIoU	Latency	FLOPs	Parameters
DeepLabv2	36.8	243	0.375 T	43.901 M

Table 1. DeepLabV2

Real-time	mIoU	Latency	FLOPs	Parameters
BiSeNet	41.8	16	25.78 G	12.582 M

Table 2. BiSeNet

We can observe in Table 1 and in Table 2 that BiSeNet has a better IOU than DeepLab. This happens in spite of the fact that BiSeNet is faster, indeed the latency and the number of parameters are lower. On the contrary Deeplab has a value of FLOPs higher, so a greater processing capacity.

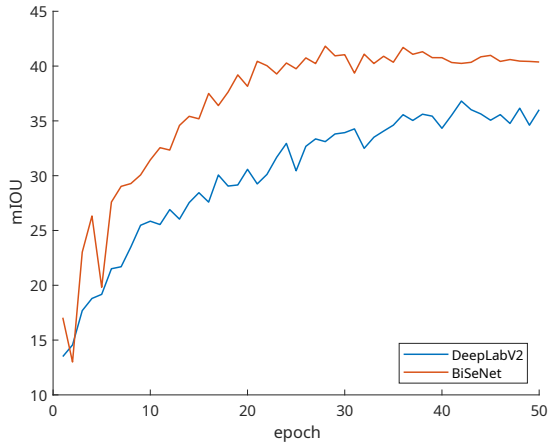

Figure 4. Comparison in terms of test mIOU between DeepLabV2 and BiSeNet

In Figure 4 we can see that test mIOU of both DeepLabV2 and BiSeNet has a positive trend. After the first few epochs, where the variance is higher, BiSeNet start to outperform DeepLabV2. However, the image shows how as the number of epochs continues to grow, the slope of the lines starts to decrease. The curves, which were initially steep, begin to flatten out: performances of both models seem to stabilize around values of 40 for BiSeNet and 35 for DeepLab.

4.2. Evaluating Domain Shift

We used GTAV for training and Cityscapes for testing.

As we can see in Table 3 the mean IOU is approximately half of the same network without domain shift. So we can see why this situation is problematic. Moreover, we observe that some classes, such as road, building, and sky, are predicted quite well: these classes are highly represented in the dataset. In contrast, rare and small objects, such as bicycles and riders, are completely overlooked.

Domain Shift	BiSeNet
mIoU	20.8
road	68.0
sidewalk	8.4
building	67.2
wall	4.1
fence	8.7
pole	7.4
light	5.2
sign	2.3
vegetation	72.3
terrain	7.4
sky	66.0
person	23.8
rider	0.0
car	40.2
truck	9.4
bus	3.6
train	1.2
motocycle	0.6
bicycle	0.0

Table 3. BiSeNet GTA V

4.3. Improving Domain Shift

Working with data augmentation we used a random horizontal flip with a percentage equal to 0.15 as basic transformation in all following runs. After little trials on down-sample datasets, we chose the combinations of parameters that had the best results: regarding the Color Jitter we picked 0.5 for brightness, contrast and saturation, and 0.05 for hue; for the Gaussian Blur we chose 3 as kernel size and a range for the variance between 0.2 and 0.8.

In Table 4 are represented performances of Color Jitter, Gaussian Blur and a mixture of the two. The only application of the Color Jitter on the source domain allowed to reach an improvement in terms of mIOU. Although the Gaussian Blur filter alone got worse performances, the combination of the two augmentations achieved an even better result: a final mIOU of 24.2, with an increase of 16% from the previous performances of BiSeNet. Table 4 shows also where these improvements come from. The most represented classes all manifested a positive increase in performance, while the less represented classes obtained conflicting results: some of them performed well, others showed a drop greater than 80% in terms of IOU.

It is worth noticing the overfitting phenomenon that shows off in training with and without data augmentations (Figure 5). Bisenet training on GTA V reached high values of mIOU, which significantly dropped during testing

Augmentation	ColorJitter	Gaussian Blur	Mixture
mIoU	22.8	16.6	24.2
road	72.9	69.3	78.2
sidewalk	4.8	0.1	5.6
building	71.2	51.2	74.0
wall	4.6	9.3	13.6
fence	14.7	4.3	15.1
pole	4.5	2.1	3.3
light	0.9	0.6	3.8
sign	1.5	0.6	0.6
vegetation	69.7	64.6	75.6
terrain	16.0	7.0	17.2
sky	78.9	55.3	79.3
person	26.9	12.6	18.7
rider	0.4	0.0	6.1e-5
car	45.1	35.5	57.1
truck	20.0	2.0	13.6
bus	0.5	0.2	1.6
train	0.0	0.0	0.1
motocycle	0.0	0.9	2.0
bicycle	0.0	0.0	0.0

Table 4. BiSeNet Data Augmentations

Adversarial approach	BiSeNet
mIoU	23.9
road	79.3
sidewalk	19.4
building	70.7
wall	17.4
fence	8.7
pole	2.7
light	0.3
sign	1.2
vegetation	69.8
terrain	20.2
sky	70.7
person	10.1
rider	0.0
car	60.0
truck	16.6
bus	2.1
train	5.4
motocycle	0.0
bicycle	0.0

Table 5. Adversarial BiSeNet

on Cityscapes. The implementation of data augmentations, that introduced noise into the source domain, helped reducing the gap, still quite large.

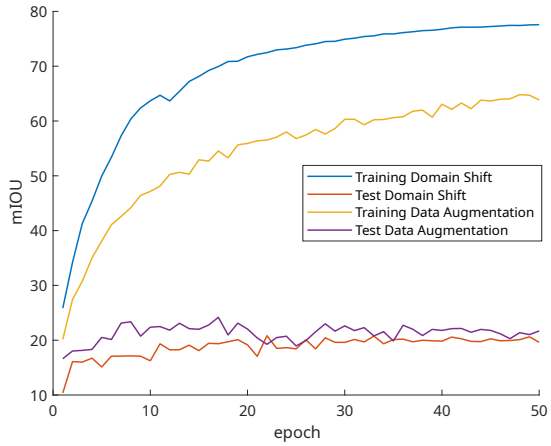

Figure 5. Comparison in terms of mIOU between evaluating domain shift and the best data augmentation

4.4. Adversarial Approach

We considered 1e-4 as initial learning rate for the discriminator. For training, we used GTA V as source domain and Cityscapes as target domain, for testing we used the validation part of Cityscapes.

In Table 5 we observe about the same mIoU of the best data augmentation. In terms of specific IOU for classes we notice that some classes have better performances and some worse. Variations are significantly but they maintain each class in same segment.

5. Conclusion

In this paper we have shown how achieving inference speed does not necessarily compromise performances: from our experimental results, BiSeNet reaches a better mIoU despite being faster and more computationally efficient. We then pointed out how domain shift can significantly decrease performances: both data augmentation and the Adversarial approach can partially resolve the problem. Data augmentation, by introducing noise into the training phase, shrinks the gap between training and test mIoU and improves classification on larger object. On the other side, the Adversarial approach achieves similar performances using a discriminator to improve model classification.

References

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1

[2] Shijie Hao, Yuan Zhou, and Yanrong Guo. A brief survey on semantic segmentation with deep learning. *Neurocomputing*, 406:302–321, 2020. 1

[3] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-

domain mixed sampling. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1379–1389, 2021. 2

[4] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018. 1, 2, 3

[5] Jiacong Xu, Zixiang Xiong, and Shankar P Bhattacharyya. Pidnet: A real-time semantic segmentation network inspired by pid controllers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19529–19539, 2023. 1

[6] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4085–4095, 2020. 2

[7] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018. 1

[8] Sicheng Zhao, Xiangyu Yue, Shanghang Zhang, Bo Li, Han Zhao, Bichen Wu, Ravi Krishna, Joseph E Gonzalez, Alberto L Sangiovanni-Vincentelli, Sanjit A Seshia, et al. A review of single-source deep unsupervised visual domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):473–493, 2020. 1