

# Real-time Domain Adaptation in Semantic Segmentation

TA: Antonio Tavera ([antonio.tavera@polito.it](mailto:antonio.tavera@polito.it))

Link to this file: [shorturl.at/prCKP](https://shorturl.at/prCKP)

## OVERVIEW

The main objective of this project is to become familiar with the task of Domain Adaptation applied to the Real-time Semantic Segmentation networks. The student should understand the general approaches to perform Domain Adaptation in Semantic Segmentation and the main reason to apply them to real-time networks. Before starting, the student should read [1] [2] [3] to get familiar with the tasks. The student should be able to replicate the network proposed in [2]. As the next step, the student should implement and modify an Adversarial Domain Adaptation algorithm, like in [6]. For the last part of the project, the student should implement a variation for the project, selecting from a set of possible ideas.

## GOALS

1. Read [1][2][3][4][5] and get familiar with “Semantic Segmentation”, Real Time networks”, “Domain Adaptation” and the datasets used;
2. Replicate the experiments detailed in the following;
3. Implement the Domain Adaptation branch and perform the experiment detailed in the following;
4. Implement and test a variation of the project.

## 1<sup>st</sup> STEP) RELATED WORKS

### Reading paper to get familiar with the task

Before starting it is mandatory to take time to familiarize yourself with the tasks of Semantic Segmentation, Domain Adaptation and Real-time Semantic Segmentation. It is compulsory to understand what are the main problems and the main solutions to tackle them in literature. More in detail, read:

- [1][2] to understand Semantic Segmentation and Real-time solution;
- [3] to get familiar with the several solutions to perform unsupervised domain adaptation in Semantic Segmentation, focusing principally on adversarial methods;
- [4] [5] to get familiar with the datasets that will be used in this project;
- [6] to get familiar with adversarial training techniques.

## 2<sup>nd</sup> STEP) IMPLEMENTING AND TESTING REAL-TIME SEMANTIC SEGMENTATION NETWORK

### Defining the baseline/upper bound for the domain adaptation phase

For this step you can assume for simplicity that your validation set is the same as the test set. Therefore:

- Model: BiSeNet [2] ([link](#))
- Dataset: a subset of Cityscapes [4] ([download here](#))
- Training Set: Train folder
- Validation Set = Test Set: Val folder
- Training epochs: 50 epochs
- Backbone: ResNet-18 (pre-trained on ImageNet)
- Semantic Classes: 19
- Metrics: Pixel Accuracy and Mean Intersection over Union (mIoU) [[read this to understand the metrics](#)]

Complete the table below using the same hyperparameters of the paper:

Table 1) Experiment	Accuracy (%)	mIoU (%)
BiSeNet (50 Epochs + ResNet-18 as backbone)	0.788	0.454

The results above will be the student upper bound (the maximum accuracy/mIoU) that the student wants/tries to reach for the Domain Adaptation phase.

## 3<sup>rd</sup> STEP) IMPLEMENTING UNSUPERVISED ADVERSARIAL DOMAIN ADAPTATION. MAKE THE FRAMEWORK LIGHTWEIGHT

Perform adversarial training with labelled synthetic data (source) and unlabelled real-world data (target). Substitute discriminator convolution with its lightweight counterpart to make the whole network real-time

You can assume:

- Source Synthetic Labelled Dataset: GTA5 [5]
  - A subset of this dataset is provided [here](#). (The folder is the same as above, which contains Cityscapes and GTA5).
  - implement loader class for the GTA5 synthetic dataset
  - Pay attention to the semantic classes. You have to select just the 19 in common with Cityscapes. The json file provided to you within

the dataset indicates the correct mapping between GTA5 and Cityscapes.

- Target Real-World Unlabelled Dataset: Cityscapes [4]
  - The same as for step 2, notice that during training semantic labels are not used
  - Test Set: val folder
  - Semantic classes: 19
- Implement discriminator function, like in [6].
- Re-write the training file to perform adversarial domain adaptation between the source and the target domain.
- Take the same parameters of step 2 and perform training. What is the maximum accuracy/mIoU reached when testing on Cityscapes test data? Measure and report the total number of parameters and Floating Point Operations (FLOPS) (use torchscan library, follow [this guide](#)). Report result here on the table:

<b>Table 2) Experiment</b>	<b>Accuracy (%)</b>	<b>mIoU (%)</b>	<b>Total Parameters</b>	<b>FLOPS</b>
Adversarial Domain Adaptation	0.593	0.216	12.6 M + 2.78 M	51.4G + 30.9G

- Modify each convolution of the adversarial discriminator with lightweight [depthwise-separable convolutions](#) and perform training again. Measure number of parameters and FLOPS. Are they changed? Reports the result here on the table:

<b>Table 3) Experiment</b>	<b>Accuracy (%)</b>	<b>mIoU (%)</b>	<b>Total Parameters</b>	<b>FLOPS</b>
Lightweight Adversarial Domain Adaptation	0.649	0.227	12.6 M + 191k	51.4G + 2.17G

#### 4<sup>th</sup> STEP) IMPROVEMENTS

Select one variation for the project among the ones proposed:

**a) Image-to-image translation to improve domain adaptation: FDA [7]**

You have to implement FDA, which is a fast and parameterless image-to-image translation algorithm, to improve the overall domain adaptation performance. Test it and compare to step 3 results.

**b) Image-to-image translation to improve domain adaptation: LAB [8]**

You have to implement LAB (section 3.1), which is a fast and parameterless image-to-image translation algorithm, to improve the overall domain adaptation performances. Test it and compare to step 3 results.

**c) Pseudo Labelling of target domain as in BDL [9]**

Generate pseudo-labels for target domain implementing the “Max Probability Threshold (MPT) defined in the BDL method. Test it and compare to step 3 results.

**d) Improving Pseudo Labelling of target domain with Class Activation Maps (CAMs): GRAD-CAM [10]**

Generate pseudo-labels for the target domain implementing the “Max Probability Threshold (MPT) defined in the BDL[9] method. Compute CAMs and combine them intelligently with pre-computed pseudo-labels. Test it and compare to step 3 results.

**e) Change the real-time semantic segmentation network with a different state-of-the-art model and compare results. How do you further improve them? Address the class imbalance problem in semantic segmentation modifying the segmentation loss.**

**f) Propose your extension.**

#### AT THE END

- Deliver PyTorch scripts for all the required steps.
- Deliver this file with the tables compiled.
- Write a complete PDF report (with paper-style). The report should contain a brief introduction, a related work section, a methodological section for describing the algorithm that you're going to use, an experimental section with all the results and discussions, and a final brief conclusion. Follow this [link](#) to open and create the template for the report.

## EXAMPLE OF QUESTIONS YOU SHOULD BE ABLE TO ANSWER AT THE END OF THE PROJECT

- What is Semantic Segmentation?
- What is a Domain Shift?
- What is Domain Adaptation?
- What are the most common solutions to perform domain adaptation in Semantic Segmentation?
- What are the main reasons to use real-time Semantic Segmentation?
- How does adversarial learning technique work for domain adaptation?
- What are the main limitations of domain adaptation?
- What is a depthwise-separable convolution?

## REFERENCES

- [1] "A Brief Survey on Semantic Segmentation with Deep Learning", Shijie Hao, Yuan Zhou, Yanrong Guo, [PDF](#)
- [2] "BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation" Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, Nong Sang, [PDF](#)
- [3] "A Review of Single-Source Deep Unsupervised Visual Domain Adaptation", Sicheng Zhao, Xiangyu Yue, Shanghang Zhang, Bo Li, Han Zhao, Bichen Wu, Ravi Krishna, Joseph E. Gonzalez, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia, Kurt Keutzer, [PDF](#)
- [4] "The Cityscapes Dataset for Semantic Urban Scene Understanding", M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, [PDF](#)
- [5] "Playing for Data: Ground Truth from Computer Games", Stephan Richter, Vibhav Vineet, Stefan Roth, Vladlen Koltun, [PDF](#)
- [6] "Learning to Adapt Structured Output Space for Semantic Segmentation", Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, Manmohan Chandraker, [PDF](#)
- [7] "FDA: Fourier Domain Adaptation for Semantic Segmentation", Yanchao Yang, Stefano Soatto, [PDF](#)
- [8] "Multi-Source Domain Adaptation with Collaborative Learning for Semantic Segmentation", Jianzhong He, Xu Jia, Shuaijun Chen, Jianzhuang Liu, [PDF](#)
- [9] "Bidirectional Learning for Domain Adaptation of Semantic Segmentation", Yunsheng Li, Lu Yuan, Lu Yuan, [PDF](#)
- [10] "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, [PDF](#)
- [11] You can find code for a lot of adversarial domain adaptation methods [here](#).