Real time domain adaptation in semantic segmentation

Jacopo Bernaudo
Politecnico di Torino
s276228@studenti.polito.it

Lorenzo Scarciglia
Politecnico di Torino
s290202@studenti.polito.it

Marco Tasca
Politecnico di Torino
s285174@studenti.polito.it

***Abstract*—Deep learning techniques have been widely used in autonomous driving systems for the semantic understanding of urban scenes. However, they need a huge amount of labeled data for training, which is difficult and expensive to acquire. A recently proposed workaround is to train deep networks using synthetic data, but the domain shift between real world and synthetic representations limits the performance. In this work we propose an unsupervised domain adaptation strategy to adapt a synthetic supervised training to real world data. The proposed learning strategy exploits two components: a standard supervised learning on real world data and an adversarial learning module that exploits both labeled synthetic data and unlabeled real data. Furthermore, we describe a simple method based on Fourier Transforms for unsupervised domain adaptation aimed to obtain better performances, whereby the discrepancy between the source and target distributions is reduced by swapping the low-frequency spectrum of one with the other.**

I. Problem overview

In semantic segmentation tasks, networks try to assign a category label to each pixel of an image, in other words, it aims to partition an image into mutually exclusive subsets, in which each one represents a meaningful region of the original image (e.g. see Fig. 1). Semantic segmentation exploits thoroughly deep learning since, with sufficient training data, the supervised learning strategy would be able to greatly extend the capacity of a segmentation model, achieving great results. Moreover, real time semantic segmentation aims to lighten the computational cost in order to achieve the same accuracies but in a shorter amount of time. In our work, we will focus in particular on a specific Residual Network, ResNet.
In an ideal world we would have enough labels to train networks in a fully supervised way but labelling images in a pixel-wise way is not an easy task and takes too much time and resources. This is where the domain adaptation concept enters the scene, given the fact that we can train the network on a synthetic dataset (IDDA) in which we have ground truth labels for each image and then adapt this domain knowledge on another dataset, sharing similar semantics. In order to adapt the IDDA (synthetic) domain to CamVid (real), we will follow the approach proposed in the AdaptSegNet paper [1], implementing an adversarial learning and a discriminator based on convolutional layers. Next, we tried to achieve better results by applying Fourier Domain Adaptation (FDA) with single scale on the previous model.

A. Datasets

In this section, we describe the datasets used for our project.

- *CAMVID* [2]: the Cambridge-driving Labeled Video Database (CamVid, 2008) is the first collection of videos with object class semantic labels, complete with metadata (Fig. 1 shows an example). The database provides ground truth labels that associate each pixel with one of 32 semantic classes, which were grouped into 11 larger classes for this project to better reflect the statistically significant classes. Only 2.68% of pixels overall were labeled as Void, i.e., not assigned to a class. This low rate is an indicator of consistent labeling and also shows that the choice of the 32 class labels was adequate for this problem. It contains 701 images in total, in which we are using 367 for training, 101 for validation and 233 for testing. The images have a resolution of 960x720.
  The database addresses the need for experimental data to quantitatively evaluate emerging algorithms. While most videos are filmed with fixed-position CCTV-style cameras, these data were captured from the perspective of a driving automobile. The driving scenario increases the number and heterogeneity of the observed object classes. Over ten minutes of high quality 30Hz footage is being provided, with corresponding semantically labeled images at 1Hz and in part, 15Hz.
  The CamVid Database offers four contributions that are relevant to object analysis researchers. First, the per-pixel semantic segmentation of 701 images was specified manually, and was then inspected and confirmed by a second person for accuracy. Second, the high-quality and large resolution color video images in the database represent valuable extended duration digitized footage to those interested in driving scenarios or ego-motion. Third, they filmed calibration sequences for the camera color response and intrinsics, and computed a 3D camera pose for each frame in the sequences. Finally, in support of expanding this or other databases, they offer custom-made labeling software for assisting users who wish to paint precise class-labels for other images and videos.
  They evaluated the relevance of the database by measuring the performance of an algorithm from each of three distinct domains: multi-class object recognition, pedestrian detection, and label propagation. In summary,

the key aspects of the ground truth annotations are: the pixel resolution labeling, the few unlabeled pixels, the high image resolution, the semantic descriptions, and the extended duration.

- *IDDA* [3]: the ItalDesign DAtaset (IDDA, 2020) is a large scale, synthetic dataset consisting of 1,006,800 frames associated with 24 semantic classes taken from the virtual world simulator CARLA [4] (an open-source project developed to support prototyping, training, and validation of autonomous driving systems). In terms of quantity of frames, IDDA is 2 orders of magnitude larger than GTAV [5] and SYNTHIA [6] and 5 order of magnitude larger than semantically annotated images in KITTI [7]. Most importantly, IDDA features many scenarios spanning different cities, weather conditions and viewpoints, so as to support the development and evaluation of single or multi-source Domain Adaptation techniques applied to Semantic Segmentation. The 105 scenarios composing IDDA (examples in Fig. 2) are obtained by varying three aspects of the simulation:
  - *Towns*: the frames of the dataset are collected across seven different towns. In particular Town7 stands out from the rest because it depicts a bucolic countryside with narrow roads, fewer traffic lights and lots of non-signalized crossings. All seven cities are populated by vehicles and pedestrians.
  - *Weather Conditions*: three weather settings are considered that differ significantly from each other: Clear Noon (CN), characterized by bright daylight, Clear Sunset (CS), with the sun low above the horizon and pink/orange hues, and Hard Rain Noon (HRN), with a cloudy sky, intense rain and puddles that cause reflections on the floor.
  - *Viewpoints*: the third parameter that is varied to create the scenarios is the player vehicle. For each vehicle the sensor system is positioned approximately at the height of the rear-view mirror. Five player vehicles that differ significantly in their height and shape were used. This choice guarantees not only that the resulting images have distinct perspectives, but also that the hood of the player vehicle, if visible, is dissimilar in both shape and color.

B. Metrics

When evaluating a standard machine learning model, we usually classify our predictions into four categories: true positives, false positives, true negatives, and false negatives. However, for the dense prediction task of image segmentation, it's not immediately clear what counts as a "true positive" and, more generally, how we can evaluate our predictions. Recall that the task of semantic segmentation is simply to predict the class of each pixel in an image. Our prediction output shape matches the input's spatial resolution (width and height) with a channel depth equivalent to the number of possible classes to be predicted. Each channel consists of a binary mask which labels areas where a specific class is present.

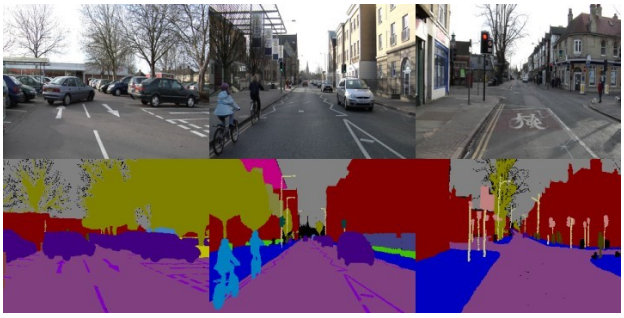

Fig. 1. CamVid: example of captured frames and their corresponding labeled frames

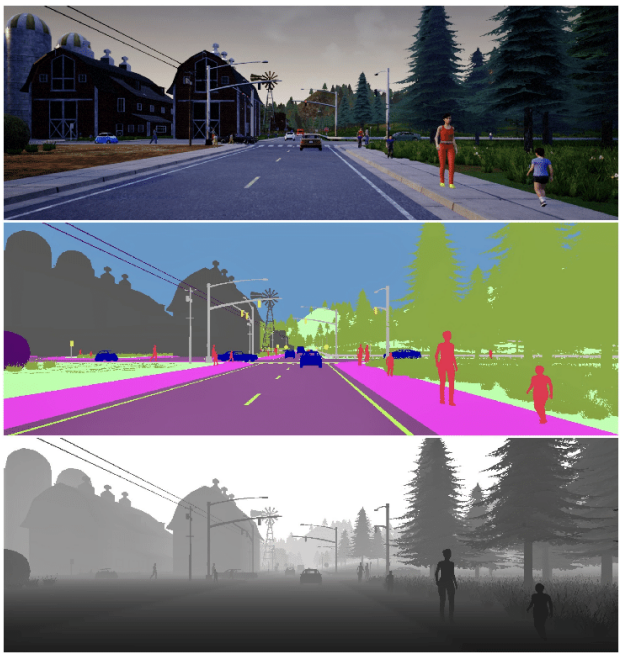

Fig. 2. The IDDA dataset. An example with an RGB image and its corresponding semantic and depth maps

- *Mean Intersection over Union*: the Intersection over Union (*IoU*) metric, also referred to as the Jaccard index, is essentially a method to quantify the percent overlap between the target mask and our prediction output. This metric is closely related to the Dice coefficient which is often used as a loss function during training. Quite simply, the IoU metric measures the number of pixels common between the target and prediction masks divided by the total number of pixels present across both masks.

$$IoU = \frac{target \cap prediction}{target \cup prediction} \tag{1}$$

  The IoU score is calculated for each class separately and then averaged over all classes to provide a global mean IoU score of our semantic segmentation prediction.
- *Pixel Accuracy*: an alternative metric to evaluate a semantic segmentation is to simply report the percent of pixels in the image which were correctly classified. The pixel

accuracy is commonly reported for each class separately as well as globally across all classes. When considering the per-class pixel accuracy we are essentially evaluating a binary mask; a true positive (TP) represents a pixel that is correctly predicted to belong to the given class (according to the target mask) whereas a true negative (TN) represents a pixel that is correctly identified as not belonging to the given class.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

This metric can sometimes provide misleading results when the class representation is small within the image, as the measure will be biased in mainly reporting how well you identify negative cases (ie. where the class is not present).

- *Training Time*: speed is a vital factor of an algorithm especially when we apply it in practice. So relative to metrics we also considered the training time to compare our models, in particular the average time needed to complete one epoch. However, this is not to be taken too much into consideration since we trained all our models using the Google Colab online platform, which does not always provide the same amount of resources (GPU and CPU), thus affecting the time needed by the model to perform its training.

C. Baseline model: BiSeNet

Semantic segmentation requires both rich spatial information and a sizeable receptive field. However, modern approaches usually compromise spatial resolution to achieve real-time inference speed, which leads to poor performance. In the paper [8], they address this dilemma with a novel Bilateral Segmentation Network (BiSeNet). They first design a Spatial Path (SP) with a small stride to preserve the spatial information and generate high-resolution features, encoding rich spatial information due to the large spatial size of feature maps. Meanwhile, a Context Path (CP) with a fast downsampling strategy is employed to obtain sufficient receptive field, an element of great significance for performance. On top of the two paths, they introduce a new Feature Fusion Module to combine features efficiently. BiSeNet is proposed to improve the speed and accuracy of real-time semantic segmentation simultaneously. In this work, the lightweight model, like Xception [9], can downsample the feature map fast to obtain a large receptive field, which encodes high-level semantic context information. In the context path, they proposed a specific Attention Refinement Module to refine the features of each stage. This module integrates the global context information without any up-sampling operation.
To summarise they used a pre-trained Xception model as the model of CP and 3 convolution layers with stride equal to 2 as the SP. Then they fused the output features of these two paths to make the final prediction. This can achieve real-time performance and high accuracy at the same time.

II. Related work

A. Semantic segmentation

Semantic segmentation (SS) assigns a category label to each pixel of an image, which is a fundamental but challenging task in computer vision research. The pixel-level semantic information helps intelligent systems to grasp spatial positions or make important judgments. With sufficient training data, the supervised learning strategy is able to greatly extend the capacity of a segmentation model applied in natural scene understanding. With the advent of deep learning in SS, new researches were done, e.g., the Fully Convolutional Network which dramatically increased the segmentation accuracy. The industrial community is also making great efforts in developing advanced systems based on semantic segmentation techniques. SS has benefited from the continuous evolution of DNN architectures.
To train these advanced networks, a substantial amount of dense pixel annotations must be collected in order to match the model capacity of deep CNNs. As a result, weakly and semi-supervised approaches are proposed in recent years to reduce the heavy labeling cost of collecting segmentation ground truths. However, in most real-world applications, it is difficult to obtain weak annotations and the trained model may not generalize well to unseen image domains. Another approach to tackle the annotation problem is to construct synthetic datasets based on rendering, e.g., GTA5 [5], SYNTHIA [6] and IDDA [3]. While the data collection is less costly since the pixel-level annotation can be done with a partially automated process, these datasets are usually used in conjunction with real-world datasets for joint learning to improve performance. However, when training solely on the synthetic dataset, the model does not generalize well to real-world data, mainly due to the large domain shift between synthetic images and real-world images, i.e., appearance differences are still significant with current rendering techniques. Although synthesizing more realistic images can decrease the domain shift, it is necessary to use domain adaptation to narrow the performance gap.

B. Domain adaptation

Large-scale labeled training datasets have enabled deep neural networks to excel across a wide range of benchmark vision tasks. However, in many applications, it is prohibitively expensive and time-consuming to obtain large quantities of labeled data. Domain adaptation methods for image classification have been developed to address the domain-shift problem between the source and target domains. Direct transfer across domains often performs poorly due to the presence of domain shift or dataset bias. Domain adaptation is a machine learning paradigm that aims to learn a model from a source domain that can perform well on a different (but related) target domain.
Numerous methods are developed based on CNN classifiers due to performance gain. The main insight behind these approaches is to tackle the problem by aligning the feature distribution between source and target images. Although feature space adaptation has been successfully applied to image

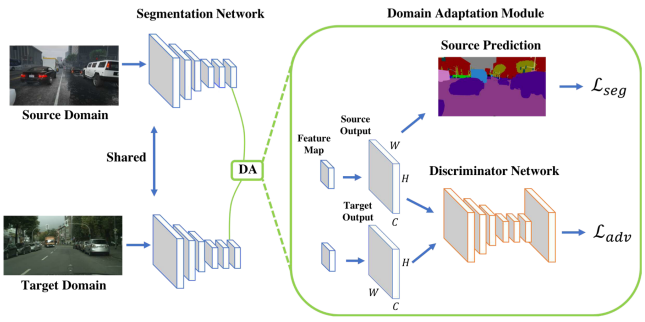

Fig. 3. Algorithm overview of domain adaptation. Original image, borrowed by [1]

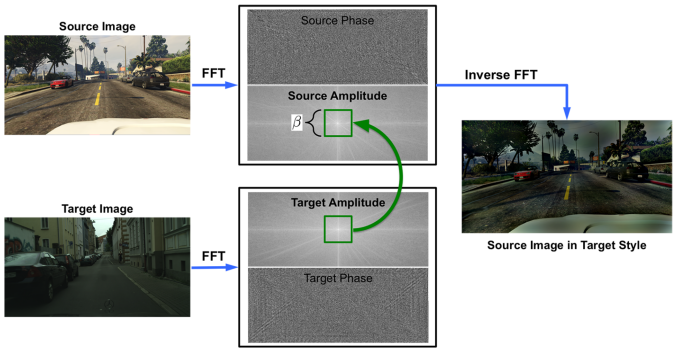

Fig. 4. Spectral transfer of FDA, image borrowed by [11]

classification, pixel-level tasks such as semantic segmentation remain challenging based on feature adaptation-based approaches. In the paper [1], they use the property that pixel-level predictions are structured outputs that contain information spatially and locally, to propose an efficient domain adaptation algorithm through adversarial learning in the output space.

C. Unsupervised domain adaptation: AdaptSegNet

In the paper [1] an adversarial learning method for domain adaptation in the context of semantic segmentation has been proposed. Considering semantic segmentations as structured outputs that contain spatial similarities between the source and target domains, we adopt adversarial learning in the output space. The crux of CNN-based approaches is to annotate a large number of images that cover possible scene variations. However, this trained model may not generalize well to unseen images, especially when there is a domain gap between the training (source) and test (target) images. For instance, the distribution of appearance for objects and scenes may vary in different cities, and even weather and lighting conditions can change significantly in the same city. To address this issue, knowledge transfer or domain adaptation techniques have been proposed to close the gap between source and target domains, where annotations are not available in the target domain.
Different from the image classification task, feature adaptation for semantic segmentation may suffer from the complexity of high-dimensional features that needs to encode diverse visual cues, including appearance, shape and context. This motivates the development of an effective method for adapting pixel-level prediction tasks rather than using feature adaptation. For instance, even if images from two domains are very different in appearance, their segmentation outputs share a significant amount of similarities, e.g., spatial layout and local context. We address the pixel-level domain adaptation problem in the output (segmentation) space. Based on the generative adversarial network (GAN) [10], the proposed model consists of two parts:

1) A segmentation model to predict output results.
2) A discriminator to distinguish whether the input is from the source or target segmentation output. With an adversarial loss, the proposed segmentation model aims to fool the discriminator, with the goal of generating similar distributions in the output space for either source or target images.

D. Fourier Domain Adaptation

Fourier Domain Adaptation (FDA) is a simple method for unsupervised domain adaptation, whereby the discrepancy between the source and target distributions is reduced by swapping the low-frequency spectrum of one with the other. This method does not require any training to perform the domain alignment, just a simple Fourier Transform and its inverse. Despite its simplicity, it achieves state-of-the-art performance, when integrated into a relatively standard semantic segmentation model.

III. Proposed Approach

A. Testing real-time semantic segmentation

After reading the papers related to the topics mentioned above, we proceeded with the training of BiSeNet, starting from the baseline that was provided. During this step, we assumed that the validation set is the same as the test set. The dataset used is CamVid with 11 semantic classes. As backbone, we tried to use two different networks: ResNet18 and ResNet101, both pre-trained on ImageNet [12]. The difference between them is the number of layers. We trained both networks over 50 and 100 epochs. ResNet101 with 100 epochs achieved the best results in pixel accuracy and mIoU. These results (see Table I) were used as the upper bound for the domain adaptation phase we will see next.
We modified the training file implementing checkpoints for parameters like optimiser, number of epochs and the related learning rate. Checkpoints are an essential element since these models need a lot of resources (especially GPU) to be trained and the Google Colab platform where we performed our experiments is not capable of doing the entire process in one instance. Regarding the losses, we initially used the dice loss and then we tried with cross-entropy since in the next steps we will use the latter. Furthermore, we tried to apply data augmentation on the input data with the aim of obtaining better training results, by showing different variations of the

same labeled image to the model, in particular performing Horizontal Flip and Gaussian Blur on the image. But since the results coming from this variant were worse than the simple base model, we decided to proceed to the next steps without it.

B. Implementing unsupervised adversarial domain adaptation

Starting from the model of the previous step we further developed to perform adversarial training with labeled synthetic data (source) and unlabeled real-word data (target), using IDDA (considering only the Red channel of RGB) and CamVid datasets. To use these two datasets together in the training part we had to perform some preprocessing. In particular, we had to select and remap the corresponding 11 IDDA labels to match the 11 ones we previously choose from CamVid, plus a 12th label that includes everything else (named background or void). Moreover, we cropped the IDDA images that originally had a size of 1920x1080 to match the ones from Camvid (960x720), since we need to have no differences in the images we input to the discriminator. In fact, we are trying to fool the discriminator into thinking that some of the images belonging to the target dataset come instead from the source dataset.

We took this paper [1] as a reference (see Fig. 3 for the general structure of the network), where they propose a domain adaptation method for pixel-level semantic segmentation via adversarial learning and demonstrate that adaptation in the output (segmentation) space can effectively align scene layout and local context between source and target images. We first forward the source image I_s (with annotations) to the segmentation network for optimizing it. Then we predict the segmentation softmax output for the target image I_t (without annotations). Since our goal is to make segmentation predictions of source and target images close to each other, we use these two predictions as the input to the discriminator (trained using an Adam optimizer) to distinguish whether the input is from the source or target domain. With an adversarial loss on the target prediction, the network propagates gradients from the discriminator to the model which would encourage it to generate similar segmentation distributions in the target domain to the source prediction.

The total loss is the following:

$$\mathcal{L}(I_s, I_t) = \mathcal{L}_{seg}(I_s) + \lambda_{adv}\mathcal{L}_{adv}(I_t) \tag{3}$$

where $\mathcal{L}_{seg}$ is the cross-entropy loss using ground truth annotations in the source domain, $\mathcal{L}_{adv}$ is the adversarial loss that adapts predicted segmentations of target images to the distribution of source predictions, and λ_{adv} is the weight used to balance the two losses.

The intuition is that no matter if images are from the source or target domain, their segmentations should share strong similarities, spatially and locally. Thus, we utilize this property to adapt low-dimensional softmax outputs of segmentation predictions via an adversarial learning scheme. We performed one-stage end-to-end training for the segmentation model and discriminators jointly, which is very effective, without using any prior knowledge of the data in the target domain. In the testing phase, we can simply discard discriminators and use the adapted segmentation model on target images, with no extra computational requirements. In this step, we exploited the fact that segmentations are structured outputs and share many similarities between source and target domains, tackling the domain adaptation problem for semantic segmentation via adversarial learning in the output space.

C. Improvements: Fourier Domain Adaptation

In order to try to improve the results of the network developed in the previous step, we applied some changes to the domain adaptation part using an image to image translation. Following the approach of [11] we applied the Fourier Domain Adaptation (FDA) technique which is a simple method for unsupervised domain adaptation. It works by reducing the discrepancy between the source and the target distribution by swapping the low-frequency spectrum of one with the other. This approach allows aligning domains without any learning and it can be used to transform unsupervised domain adaptation into semi-supervised learning. Since it works well in aligning domains, it should fool better the discriminator. We only applied the FDA with single scale in which we have β as a fixed parameter, describing the portion of the image that we transfer from target Fourier domain to source Fourier domain.

The motivation of the FDA approach stems from the observation that the amplitude can vary significantly without affecting the perception of high-level semantics. However, low-level sources of variability have a significant impact on the spectrum, forcing the model to learn them with other nuisance variabilities. But we have to note that if these variabilities are not represented in the training set, the model fails to generalise. The Fourier Domain Adaptation is used to reduce the domain gap between the two datasets. They compute the Fast Fourier transform of both target image and source image, then they substitute a piece of target amplitude into the source amplitude (in the same position). After doing this, we come back to an image using the Inverse Fast Fourier Transform. In this way we map a source image to a target style without altering semantic content (see Fig. 4).

To be precise, in order to run the fft and ifft on Colab with the same signature of [11] we had to downgrade these components to a previous version: Pytorch to 1.6, Torchvision to 0.7 and Torchtext to 0.7.

IV. Results

The experiments performed in the previous sections, produced the results summarised in Table I. First we tried to train different backbone networks (ResNet18 and ResNet101) varying the number of training epochs. We found that the best combination was ResNet101 model with 100 epochs, so we developed the following experiments considering this as base model. This step stated that the values to consider as upper bound are: 0.878 and 0.666 respectively for accuracy and mIoU. As we can see from Table I, when we tried to apply data

TABLE I
RESULTS

Experiment	Accuracy (%)	mIoU (%)	Training Time (avg per-epoch) (minutes)
BiSeNet(50 Epochs + ResNet-18)	0,859	0,602	04:58
BiSeNet(50 Epochs + ResNet-101)	0,872	0,650	04:34
BiSeNet(100 Epochs + ResNet-18)	0,867	0,620	04:57
BiSeNet(100 Epochs + ResNet-101)	**0,878**	**0,666**	04:56
BiSeNet(100 Epochs + ResNet-101) + Data augmentation	0,878	0,664	04:42
BiSeNet(100 Epochs + ResNet-101) + Adversarial learning	**0,673**	**0,308**	05:05
BiSeNet(100 Epochs + ResNet-101) + Adversarial learning + FDA	0,643	0,292	06:24

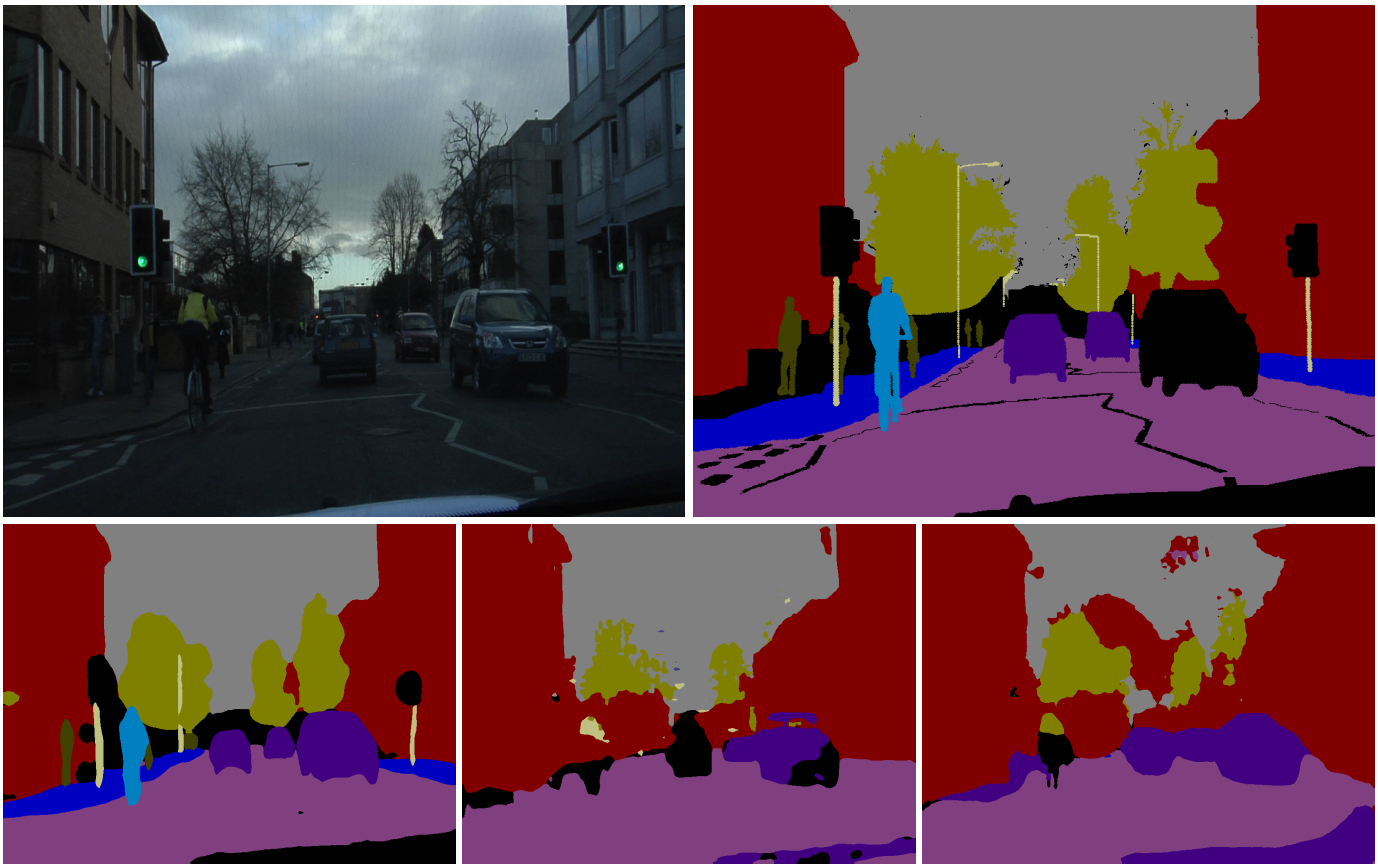

Fig. 5. Starting from the top left we have: source image, ground truth, output prediction of SS, output prediction of SS + domain adaptation, output prediction of SS + domain adaptation + FDA

augmentation, we got slightly worst results. Next, we trained the network applying adversarial learning to perform domain adaptation. Results went down, as expected, but considering the complexity of the task, they still seem good enough. Lastly, we applied FDA. From a theoretical point of view, here the results should have become better but we found that they were not as good as the previous model without FDA. Moreover, analysing more accurately the evaluation results during training (every five epochs) in the above steps, it seems that after a while, the network tends to overfit. More precisely, considering the overfitting process we found that:

- Adversarial learning top results were obtained after 90 epochs: Accuracy 0.678 and mIoU 0.317
- Adversarial learning with FDA top results were obtained after 70 epochs: Accuracy 0.673 and mIoU 0.319

V. DISCUSSION

In this project we developed a network based on BiSeNet with the aim of solving real time domain adaptation. First we trained the network in a supervised manner on real world dataset (CamVid), obtaining a good upper bound for our next experiments. Then, following the approach of AdaptSegNet we implemented a possible solution to perform domain adaptation,

introducing another dataset (IDDA), but this time a synthetic one. As expected, results got worse but this still seems a promising starting point for future analysis. Finally, when we applied FDA we expected to improve our previous results but this did not happen, even if they are not so far.

REFERENCES

[1] Y. Tsai, W. Hung, S. Schulter, K. Sohn, M. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," *CoRR*, vol. abs/1802.10349, 2018.

[2] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[3] E. Alberti, A. Tavera, C. Masone, and B. Caputo, "IDDA: a large-scale multi-domain dataset for autonomous driving," *CoRR*, vol. abs/2004.08298, 2020.

[4] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, and V. Koltun, "CARLA: an open urban driving simulator," *CoRR*, vol. abs/1711.03938, 2017.

[5] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," *CoRR*, vol. abs/1608.02192, 2016.

[6] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," pp. 3234–3243, 2016.

[7] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," pp. 3354–3361, 2012.

[8] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Computer Vision – ECCV 2018* (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), pp. 334–349, Springer International Publishing, 2018.

[9] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *CoRR*, vol. abs/1610.02357, 2016.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014.

[11] Y. Yang and S. Soatto, "FDA: fourier domain adaptation for semantic segmentation," *CoRR*, vol. abs/2004.05498, 2020.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.