Exploring Federated Learning for Semantic Segmentation in Autonomous Driving Scenarios

Giacomo Zuliani
Politecnico di Torino
s315052@studenti.polito.it

Davide Buoso
Politecnico di Torino
s317660@studenti.polito.it

Marco Castiglia
Politecnico di Torino
s317381@studenti.polito.it

Abstract

This project explores the application of Federated Learning (FL) to the task of Semantic Segmentation (SS), with a focus on preserving client privacy while utilizing their data for model training. The proposed approach involves a centralized server pre-training phase on a labeled dataset, incorporating a style-transfer technique for domain adaptation. In the federated decentralized setting, our approach tackles the challenge of absent labels on client data. By leveraging pseudo-labels and self-training, the approach enables the utilization of unlabeled client images, effectively addressing the issue of limited ground truth availability. The report also provides additional insights into extending the applicability of this approach to domains beyond self-driving cars, such as satellite imagery. Additionally, an intriguing possibility explored in this project is the integration of a transformer model into the existing framework, presenting a promising alternative to the commonly employed CNN architectures. Code available at: https://github.com/lambdavi/mldl23

1. Introduction

Autonomous driving has emerged as a promising technology that has the potential to revolutionize transportation systems and improve road safety. One of the key requirements for enabling autonomous vehicles is their ability to understand and interpret the environment in real-time. Semantic segmentation (SS), a computer vision task, plays a crucial role in achieving this goal by assigning semantic labels to individual pixels in an image, enabling vehicles to perceive and comprehend their surroundings.

However, in order to perform semantic segmentation effectively, large amounts of labeled data are required for training robust models. Collecting data from the clients could be a potential solution to the lack of data, but ensuring the privacy of user data, collected from various autonomous vehicles, is a critical concern. Federated Learning (FL) [1] offers a potential solution to these challenges, with a local training taking place on the client's device, transmitting only the model weights instead of the actual data. This approach addresses privacy and security concerns by avoiding the need to share sensitive information while still allowing collaborative learning to improve the overall model's performance.

In this context, another important problem is given by the absence of labels on the images that come from these users: the project focuses on the task, defined in [2] as Federated Source-Free Domain Adaptation (FFreeDA), which addresses the limitations of existing FL approaches that assume labeled data in remote clients. The FFreeDA scenario assumes that the clients' data is unlabeled, and the server only has access to a labeled source dataset for pre-training.

This report presents the progress and findings of the project, which involves implementing and evaluating different stages and techniques within the FL and SS framework, with additional insights and potential avenues for future research, encouraging innovative exploration in the field.

2. Related work

2.1. Semantic Segmentation

Semantic segmentation plays a crucial role in autonomous driving applications as it aims to predict the class of every pixel in an image. Deep learning-based approaches have achieved remarkable success in this area. [3] provides a comprehensive view on semantic segmentation with deep learning, and, in general, images classification tasks, highlighting the advancements and challenges in the field.

2.2. Federated Semantic Segmentation for Autonomous Driving

In our report, we draw inspiration from two notable papers in the field, namely *FedDrive* [4] and *LADD* [2], which are closely related to our work. The FedDrive paper introduces a novel approach for federated learning in the context of autonomous driving, aiming to enhance the performance of self-driving systems across multiple distributed

data sources. This work addresses the challenge of domain shift and data heterogeneity, offering insights into effective adaptation strategies for federated learning scenarios. Similarly, the LADD paper presents an algorithm specifically designed for Source-Free Domain Adaptation in Federated Learning for Semantic Segmentation. It tackles the task of adapting a pre-trained model to unlabeled client data within the federated learning framework, highlighting the importance of domain adaptation techniques in improving the generalization capabilities of federated models.

Additionally, our report also acknowledges the contribution of the *FedAvg* paper [5]. The latter is explained in greater detail within the next sections of the paper. The integration of ideas and inspirations from these papers into our work enriches our report and allows us to build upon existing knowledge in the field.

2.3. Domain Adaptation for Semantic Segmentation

Domain adaptation (DA) techniques aim to address the domain shift problem in semantic segmentation. The main approach we utilized is *FDA* [6], based on transferring the style from the target dataset to the source dataset. In our final experiments, this technique is used during the pre-training of the server model.

2.4. Pseudo-labels and Federated Learning

Pseudo-labels have been widely used as an unsupervised or semi-supervised self-training technique to improve performance in various domains. In semantic segmentation, pseudo-labels can be generated by making a forward pass through a teacher model and using confidence probabilities to assign labels to pixels [7]. The use of pseudo-labels in federated learning for semantic segmentation has not been extensively explored but holds potential for improving model performance and generalization.

3. Methodology and experimental setup

In this section, we present the methodology and experimental setup employed in our project. We discuss the various components that contribute to the development of our approach, including the datasets used, the network architecture of our model, and the preprocessing techniques applied. Furthermore, we delve into the hyperparameters selection and tuning.

By providing a detailed account of our methodology and experimental setup, we aim to ensure transparency and reproducibility while offering insights into the decisions made throughout the project. This section serves as a foundation for understanding the subsequent challenges, results and discussions.

3.1. Datasets

In this subsection, we provide an overview of the datasets used in our project: the ItalDesign DAtaset (IDDA) [8] and the GTA5 dataset [9]. The IDDA dataset is a large-scale synthetic dataset for semantic segmentation in autonomous driving. For our project, we picked from IDDA a limited number of images: 600 for the training set, 120 for the same-domain test set, and 120 for the different-domain test set. The GTA5 dataset is a synthetic dataset generated from the video game Grand Theft Auto V (GTA5). It is a labeled dataset that offers a large-scale and diverse set of urban driving scenes. The dataset comprises virtual driving scenarios in a realistic urban setting, featuring various traffic conditions, pedestrian behaviors, and environmental factors. From GTA, we used a total of 500 pictures.
These datasets are utilized in various ways and configurations to address the different tasks and challenges presented in this project. The specific approaches and techniques employed using these datasets will be discussed in subsequent sections.

3.2. Network architecture

In our project, we utilize the DeeplabV3 [10] network with MobilenetV2 [11] as a backbone, pre-trained on ImageNet. To handle the problem of segmenting objects at multiple scales, DeeplabV3 architecture employ atrous convolution making it suitable for the diverse tasks and challenges in autonomous driving. MobilenetV2 is specifically selected as the backbone due to its lightweight and efficient nature. This choice aligns with the project's objective of developing a system capable of running near real-time applications. Additionally, the utilization of MobilenetV2 becomes particularly relevant in later stages of the project, where mini-trainings are performed on the client side.

3.3. The two configurations

The initial stages of the project involved experiments conducted in both centralized and decentralized settings. Exploring these two settings allowed for a comprehensive understanding and comparison of their respective advantages and limitations. For evaluation, we used both IDDA same-domain and different-domain test datasets.

3.3.1 Centralized Configuration

In the centralized configuration, we trained a server model using the images from the IDDA train dataset, with a naive assumption that all client images are available for server training. This setup allows us to analyze the performance and behavior of the model when trained on a centralized dataset. Our focus in this configuration is on optimizing hyperparameters and preprocessing techniques to improve the model's performance.

3.3.2 Client-Server Configuration

Building upon the insights gained in the centralized configuration, we expanded our experimentation to the client-server setup, reflecting a more realistic scenario where clients only have access to their own car observations. To simulate this setting, the training images were allocated to the clients following the clients' training set partition, and a central model was collectively produced using the Federated Averaging algorithm. During this phase, we explored distributed-specific settings to identify an optimal configuration.

3.4. FederatedAveraging

Federated learning enables the training of models across multiple clients without centralized data storage. However, a method is required to obtain a central model from the federated computations performed by the clients. This becomes challenging due to the statistical heterogeneity among clients' data. To address this, FedAvg, a widely used federated optimization approach, employs a weighted average of client updates to learn a global model. This algorithm is particularly advantageous for scenarios with a large number of clients, as it is a communication-efficient algorithm, transmitting only the model parameters.

3.5. Data Augmentation and transformations

In our experiments, we explore the combination of various data augmentation techniques for semantic segmentation tasks. By leveraging these techniques, we aim to enhance the performance and generalization ability of our models to handle various challenges.

Specifically, we employ these well-known transformations from Pytorch library [12]: RandomResizedCrop, RandomCrop, RandomHorizontalFlip and ColorJitter. Furthermore, we implement a custom transformation which we will refer to as CustomWeatherTransform in the report. It is a custom augmentation technique designed to simulate the effect of rain on images. With a certain probability, this transformation randomly applies a rain effect to the input images, adding realistic weather conditions. By incorporating such variations, the model becomes more robust to environmental changes, thus improving its performance in real-world scenarios.

4. Challenges and proposed approach

When trying to apply Federated Learning to real-life scenarios, we consider on one side the clients with their private images taken from their cars, and on the other side a centralized server, which we will pre-train with labelled images. In these settings, two major obstacles arise: the absence of labels on the client data and the domain shift between the clients' and the server's dataset. We refer to this challenging setting as FFreeDA.

Firstly, the domain shift between the clients' and the server's dataset adds another layer of complexity. This problem is even more determining when the source dataset is syntethic, and our model tries to generalize for the real observations of our clients.

Secondly, the absence of labels on the client data poses a significant hurdle because it puts a limit on the possibility of federated training on the clients' images.

In our settings, we employ the GTA5 dataset for server pre-training, which represents the available labelled data. On the other hand, the IDDA dataset, without labels, simulates the clients' data.

By incorporating the following techniques, we aim to tackle the absence of labels on the client data and mitigate the domain shift between the server and client datasets.

Figure 1. FDA: style transfer with two different window sizes

4.1. Fourier Domain Adaptation

The Fourier Domain Adaptation [6] (FDA) technique applies FFT to source and target images and then replaces the low frequency part of the source amplitude with the one from the target. Finally, inverse FFT is applied to the modified source spectrum. We used this method to extract the average style from each client's subset of the IDDA training set, forming a bank of target styles. The natural division of the dataset into clients accurately captures the distribution differences between the domains, mimicking a real-life scenario. This enables us to explore domain adaptation techniques and address the domain shift challenge. During training, we apply to the source images (from GTA5 dataset) a random style from the bank of target styles. This operation acts as a new transformation, simulating the variations and styles present in the IDDA dataset. By incorporating FDA into our training pipeline, we can train our

models to adapt to different styles and appearance characteristics encountered in real-life scenarios. When applying FDA, we choose a target dataset from which we'll extract the styles, and we expect our model to perform better when evaluating in this dataset, while worsening its performance on the source dataset.

To ensure the effectiveness of the FDA technique, we carefully tune the hyperparameters, particularly the window size and the number of images. The first one controls the size of the low frequency window to be replaced, while the second is the number of images per client that contribute to the extraction of the style. An example of this method can be seen in Figure 1. Through systematic experimentation, we explore different window sizes to find the optimal configuration that promotes successful style transfer during training.

4.2. Federated source-Free Domain Adaptation

In order to facilitate the learning process of our model from unlabeled data provided by clients, we adopt the Teacher-Student paradigm. This framework involves a Teacher model, residing on the server, which transmits its knowledge to the clients in each round. Additionally, a Student model is employed, serving as the target model that we aim to enhance using pseudolabels. Each client, then, possesses its own local student model, the parameters of which diverge from the others by the end of the round. To initiate the process, the server initializes both the Teacher and the Student models using the parameters of a pretrained model. Then, at the beginning of each round, a subset of clients is selected, and the Teacher model makes predictions on the client side, generating pseudolabels that act as ground truth for training the local student models. To ensure model stability, only predictions with high confidence are retained, thus preventing model divergence.

Subsequently, the parameters of all the client's student models are transmitted back to the server, where they are aggregated using the FedAvg method. This aggregation step updates the server's student model to incorporate the collective knowledge from the client models. This iterative process continues for a specified number of rounds, denoted as NR.

In the first variant of the FFreeDA task, the teacher model parameters remain fixed throughout the entire training process. Additionally, we explore the possibility of periodically updating the teacher model, specifically after every TS teacher steps (rounds).

5. Results and Analysis

All the experiments of the following section have been run using a single NVIDIA TESLA P100 provided by Kaggle platform. Throughout our work, we consistently use the mean Intersection over Union (mIoU) as evaluation metric for the performance of the segmentation tasks. We decided to run our experiments for only 30 epochs due to our computational limitations, noticing that the mIoU scores, which assess the segmentation quality, continue to improve gradually on both the training and test sets beyond this point, albeit at a slower rate.

The tables presented in this report will include mIoU values obtained from evaluations on various test sets. To provide a concise legend, we will refer to the mIoU value from the source dataset as SD, from the target (train partition) dataset as T1, from the same domain test set as T2, and from the different domain test set as T3.

LR	BS	WD	T1	T2	T3
$1 \cdot 10^{-4}$	2	$1 \cdot 10^{-4}$	0.616	0.547	0.366
$5 \cdot 10^{-4}$	2	$1 \cdot 10^{-4}$	0.541	0.429	0.233
$1 \cdot 10^{-5}$	2	$1 \cdot 10^{-4}$	0.458	0.413	0.287
$5 \cdot 10^{-5}$	2	$1 \cdot 10^{-4}$	**0.596**	**0.549**	**0.400**
$5 \cdot 10^{-5}$	4	$1 \cdot 10^{-4}$	0.577	0.530	0.380
$5 \cdot 10^{-5}$	2	0	0.59	0.544	0.393

Table 1. Centralized setting: hyperparameters tuning with RandomCrop transformation and 30 epochs for each experiment.

5.1. Centralized Baseline

The initial experiment serves to establish a centralized baseline for training our model on the IDDA dataset. To streamline our initial parameter tuning process, we focused on comparing two popular optimization algorithms: Stochastic Gradient Descent (SGD) with and without a scheduler, and Adam [13]. After careful evaluation, we selected the Adam optimizer due to its ability to dynamically adapt the learning rate. This adaptivity is particularly beneficial in deep learning scenarios and can facilitate faster convergence and potentially better overall performance. With this configuration, described in Section 3.3.1, we tested the learning rate (LR), batch size (BS), and weight decay (WD) as the primary optimization parameters.

We conducted an incremental search on those hyperparameters applying the RandomResizedCrop transformation as only data augmentation technique, described in Section 3.5. The results presented in Table 1 were obtained by identifying the optimal learning rate and subsequently exploring other combinations. As depicted in the table, the highest performance was achieved with a LR of $5 \cdot 10^{-5}$, BS of 2, and WD of $1 \cdot 10^{-4}$. Notably, the results obtained on the test set from the different domain are of particular interest as they indicate the model's capability for generalization. Due to this aspect, we prefer a learning rate of $5 \cdot 10^{-5}$ instead of $1 \cdot 10^{-4}$, which achieved the highest mIoU value during the evaluation on the training partition.

A research for the best combination of transformations

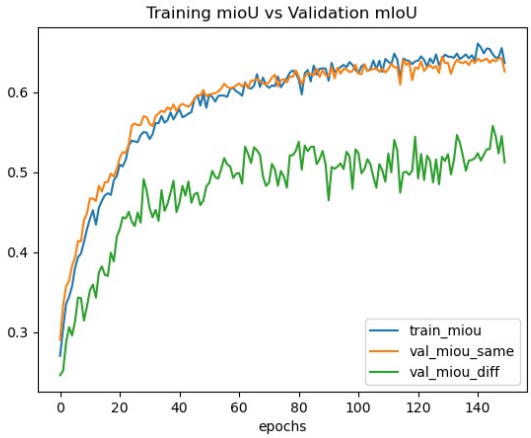

Figure 2. Final run with the best configuration of the centralized baseline using ADAM optimizer.

Test set	mIoU $\pm$ std
Eval (train partition) (T1)	0.643 ± 0.004
Test Same Domain (T2)	0.636 ± 0.002
Test Diff Domain (T3)	0.519 ± 0.012

Table 2. Final results IDDA centralized

(among the one mentioned Section 3.5) lead to the decision of using RandomCrop and CustomWeatherTransform.

To validate this configuration, we conducted an experiment spanning 150 epochs, employing the finalized set of hyperparameters and transformations in Figure 2. We can see, with respect to the initial run with SGD, that the hyperparameters tuning improved the model performance, especially on its ability to generalize. The results obtained from multiple runs with this setting, as shown in Table 2, serve as a baseline for the subsequent stages of the task.

5.2. Supervised Federated Experiments

This phase focuses on the distributed scenario, wherein a server entity and multiple clients are involved. These clients operate independently and do not share any information amongst themselves. The server acts as an intermediary, facilitating communication between the clients and updating the models using the FedAvg principle, as explained in earlier sections. In this context, we train the server model using an iterative approach consisting of N rounds.

During each round, a specific number of clients are randomly sampled, and subsequently trained for a designated number of local epochs using their respective subsets of data.

In the Supervised Federated setting, we applied the best settings from the previous point, and focused the search on the typical parameters of a distributed scenario, previously described. From the Table 3 we can notice two main general trends:
1) Increasing the number of local epochs increases the performance. We can notice this behaviour for all settings, and we tested this up to 20 epochs for just one configuration. This experiment indicated that, even with 20 epochs and the small number of images, the overfitting is not significant. This suggests the potential for further runs with a higher number of epochs but probably with some limitations due to the number of images;
2) Increasing the number of clients generally leads to better results. This can be attributed to the utilization of a larger portion of the available data. In our opinion, choosing a larger number of clients, the FedAvg algorithm should introduce less approximation error. However, in scenarios involving more local epochs and rounds, where the same data is reused multiple times, this relationship may not hold. Given additional resources, it would be valuable to conduct experiments with a higher number of epochs to investigate potential differences.

CR	NLE	T1	T2	T3
2	1	0.254	0.283	0.238
2	3	0.330	0.347	0.298
2	6	0.365	0.391	0.331
4	1	0.274	0.292	0.236
4	3	0.316	0.336	0.272
4	6	0.365	0.368	0.279
8	1	0.262	0.282	0.246
8	3	0.316	0.319	0.267
8	6	0.374	0.389	0.327
2	20	0.475	0.491	0.419

Table 3. Distributed Configuration experiment, performed with 30 rounds for each run. CR - clients per round, NLE - Number of local epochs per client.

5.3. Pre-training Phase on GTA5

In this section we explored the configurations for an optimal centralized pre-training on the GTA5 dataset. We started from the best settings found in Section 5.1 and retuned our parameters for the different dataset. We obtained the best settings with a similar configuration, with just the batch size changing from 2 to 4. Once again, the training was performed with 30 epochs.

As shown in Table 4, we then tuned the hyperparameters for the FDA technique, finding the best configuration with window size of 0.01 and extracting the style taking into account all the images for each client.

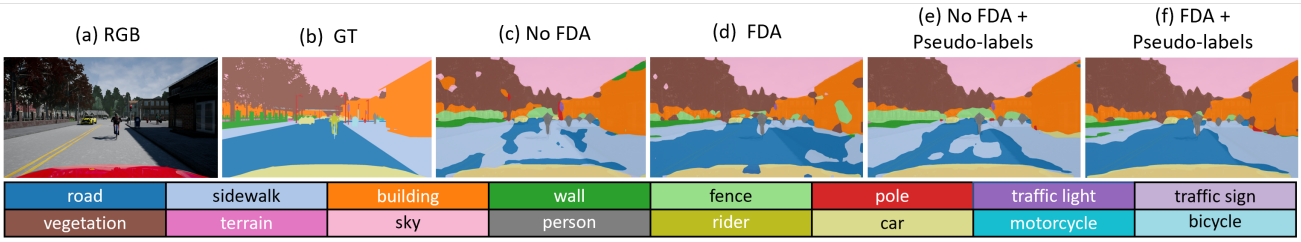

Figure 3. GTA5: qualitative results

The application of FDA results in inferior performance during evaluation on the source dataset. However, we observe an improvement (0.04 in miou) in the target (partition of the IDDA dataset), which is the dataset from which the styles are actually extracted. This trend was predictable and can be explained with the application of the new style which adds another layer of difficulty to the training and makes it more challenging. Therefore, when evaluating on the source dataset, we cannot expect to see a big improvement using the same number of epochs.

We then conducted separate runs using the best results from both the searches, with and without FDA, each spanning 100 epochs. These runs allowed us to provide two checkpoints that we will use in the next section of experiments. The results of these runs are presented as baseline in Table 5. With this increased number of epochs, we still see a slight improvement in the evaluation of the target domain, but the FDA does not show much better results as it did with 30 epochs.

WS	BS	Nimg	TS	T1	T2	T3
No FDA	4	-	0.455	0.228	0.249	0.21
0.001	4	all	0.42	0.27	0.29	0.225
0.01	4	all	0.413	**0.28**	**0.285**	**0.241**
0.1	4	all	0.39	0.219	0.238	0.166
0,01	4	15	0.41	0.27	0.282	0.228
0,01	4	5	0.414	0.275	0.289	0.241
0.01	2	all	0.384	**0.273**	**0.290**	**0.252**

Table 4. Centralized GTA5 dataset applying FDA: these runs were conducted with previous settings (Section 5.3)

5.4. Self-Training framework (FFreeDA)

This section is devoted to the Self-Training framework with pseudo-labels, described in Section 4.2. We start from the two checkpoints we saved at the end of the centralized pre-training phase, keeping the best models found. We explored different scenarios, trying to variate clients per round, local epochs and the step which updates the teacher model. This research let us analyze and draw some interesting point shown in Table 5. It is evident that both the FDA and non-FDA settings demonstrated improvements in their scores compared to the starting baseline. Both the configurations, with and without FDA, showed the best results in this task with the following settings: 2 clients per round, conducting 5 local epochs for each client, and utilizing an infinite teacher step (where the teacher model was never updated). Comparing the two, we notice that, as expected, running this experiment with FDA leads to better results in the target dataset. We also observe that we get a higher validation mIoU in the different domain test. Overall, while these observations are subject to variations, FDA generally leads to the best results.

For each best configuration, we performed three runs to assess the robustness of these results, which we included in the Table 6.

Our hypothesis was that a configuration without teacher updates would yield the best performance and this was confirmed by the results. This is because the server Teacher model is pre-trained on a source dataset and demonstrates good performance. It is important to note that the teacher model's knowledge of the world is limited and it is not a flawless labeling function. Consequently, when the student model learns from the images that the teacher model

-	Params	-	-	no FDA	-	-	FDA	-
CR	NLE	SD	T1	T2	T3	T1	T2	T3
2	1	∞	0.297	0.332	0.282	0.32	0.346	0.292
8	1	∞	0.292	0.317	0.25	0.325	0.347	0.283
2	1	1	0.24	0.24	0.231	0.285	0.317	0.275
8	1	1	0.19	0.186	0.194	0.274	0.296	0.247
2	1	2	0.265	0.246	0.25	0.283	0.294	0.26
8	1	2	0.194	0.213	0.219	0.301	0.32	0.271
2	1	5	0.257	0.275	0.264	0.306	0.328	0.279
8	1	5	0.277	0.301	0.263	0.323	0.339	0.28
2	**5**	∞	**0.305**	**0.356**	**0.3**	**0.335**	**0.343**	**0.317**
8	5	∞	0.28	0.299	0.279	0.316	0.335	0.295
-	Base	-	0.302	0.341	0.262	0.310	0.331	0.258

Table 5. Experiments in the distributed scenario. (Centralized) Base is obtained with the evaluation of the checkpoints. CR - clients per round, NLE - Number of local epochs per client.

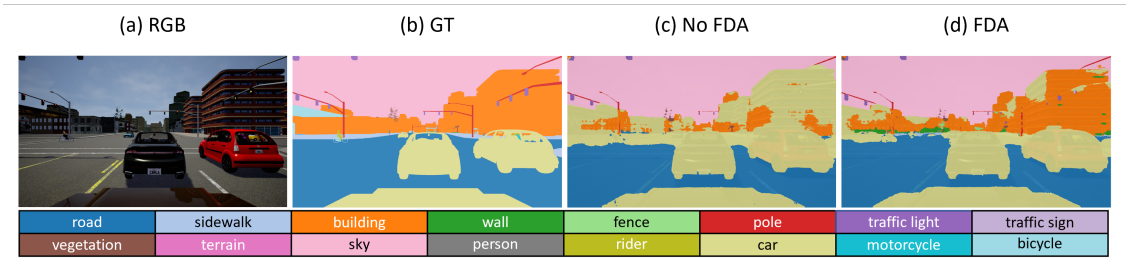

Figure 4. GTA5: qualitative results with transformers model

excels at classifying, it may predominantly focus on specific classes where the teacher is most confident. In our view, achieving a superior teacher model compared to the pretrained model would require numerous rounds of training and updates (with more images), with updates only occurring when we have sufficient evidence that the student model has indeed surpassed the teacher's performance. Otherwise, there is a risk of deviating from the optimal model and potentially encountering divergence issues.

Config	T1	T2	T3
NO FDA	0.302 ± 0.001	0.345 ± 0.009	0.292 ± 0.007
FDA (WS=0.01)	0.332 ± 0.003	0.342 ± 0.001	0.316 ± 0.001

Table 6. Final evaluation for the FFreeDA setting (mIoU $\pm$ sd)

6. Further research possibilities

6.1. Transformers model

Transformers have emerged as a promising architecture in the field of deep learning, offering advancements in various computer vision tasks, including semantic segmentation for autonomous driving. Specifically, in the context of semantic segmentation, the SegFormer model has demonstrated exceptional performance and efficiency. SegFormer [14] combines the strengths of Transformers with a lightweight multilayer perceptron (MLP) decoder, resulting in powerful representations for accurate and robust semantic segmentation. By leveraging the self-attention mechanism of Transformers, SegFormer effectively captures global and local contextual information within images, enabling precise object recognition and segmentation. The adoption of SegFormer in the code implementation has proven to be a valuable choice, leading to improved semantic segmentation results. A qualitative result of the improvements of Segformer using FDA is presented in Figure 4.

6.1.1 Methods and Findings

In our report we tested the application of the Segformer model instead of the DeepLabV3 + MobileNetv2, to see if it would lead to better results overall. Additionally, since we were not completely satisfied with the behaviour of FDA on the previous experiments, we further explored this model and repeated the steps in Section 5.3. For these experiments, we will omit the conducted tuning to the reader and we will only show the two best results, without and with FDA, conducted with 30 epochs each (Table 7). The best configuration found during our tuning, with respect to the centralized setting, was with LR = $1 \cdot 10^{-5}$, WD = $1 \cdot 10^{-4}$ and BS = 2. The best FDA setting found for this configuration is with window size = 0.01 and using all images for each client.

Config	SD	T1	T2	T3
NO FDA	0.502	0.3	0.243	0.17
FDA (WS=0.01)	0.5	0.347	0.304	0.204

Table 7. SegFormer results for centralized run with and w/o FDA.

6.2. Federated learning for satellite images

As an additional aspect of our project, we aimed to evaluate the versatility of our architecture across different scenarios. We considered the application of federated learning and semantic segmentation techniques to analyze earth observation datasets obtained from satellite imagery. This exploration would serve as a foundation for future research. The decision to incorporate these datasets was motivated by the intriguing connection between our project and the potential use of the federated learning approach with satellite data.

Federated learning in satellites offers several advantages. Firstly, it ensures data privacy and security by keeping sensitive information on the satellites instead of transmitting it to a central location. This is crucial for handling classified or sensitive data. Secondly, federated learning allows satellites to learn collectively from a diverse set of data sources, benefiting from the distributed knowledge across the satellite network. Lastly, it reduces communication bandwidth requirements by exchanging only model updates instead of raw data, resulting in more efficient data transfer between satellites and the central server [15].

The collaboration facilitated by federated learning can

CR	NLE	TS	T1	T2	T3
2	**1**	∞	**0.545**	**0.606**	**0.455**
8	1	∞	0.529	0.555	0.458
2	1	1	0.164	0.138	0.155
8	1	1	0.348	0.366	0.397
2	1	2	0.345	0.348	0.281
8	1	2	0.309	0.33	0.252
2	1	5	0.511	0.493	0.406
8	1	5	0.428	0.576	0.454
2	5	∞	0.535	0.588	0.457
8	**5**	∞	**0.538**	**0.574**	**0.461**
-	Base	-	0.529	0.626	0.456

Table 8. Experiments in the distributed scenario using pseudolabels with the best configuration from the centralized experiments on LoveDa dataset. CR - clients per round, NLE - Number of local epochs per client, TS - teacher update step.

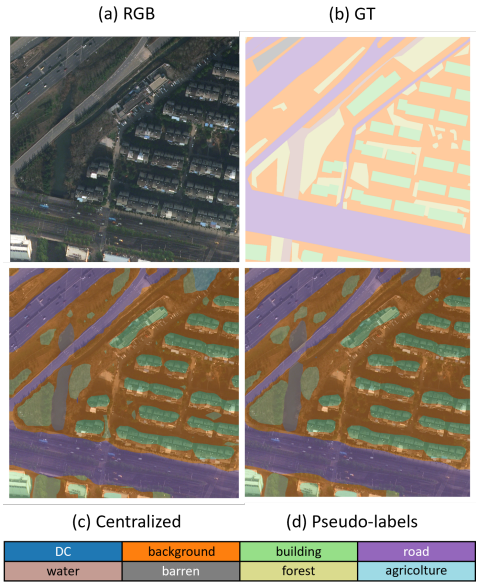

Figure 5. LoveDA: qualitative results

lead to improved predictions and decision-making in various aspects such as natural disasters predictions, weather forecasting, and space exploration. In this case, we'd like to explore its capabilities of enhancing the capabilities of self-driving cars. This approach, in fact, has the potential to provide more accurate and faster predictive information for route planning, surpassing the limitations of current maps that primarily rely on user observations and historical data, which often require human verification.

However, it is crucial to acknowledge the challenges and limitations associated with this idea. One of these is the computational time required for processing large-scale satellite imagery, especially if we wish to use this approach in near real-time application for autonomous cars.

6.2.1 Methods and Findings

To set a starting point to this research possibility, we tried to apply our previously described setting to a different dataset: LoveDA [16]. We chose this dataset because it encompasses two domains (urban and rural), which bring considerable challenges due to the: 1) multi-scale objects; 2) complex background samples; 3) inconsistent class distributions. These challenges and complications could resemble real-life scenarios for which we want to experiment. We organized our datasets as follows, in order to experiment with a FFreeDA-like task using loveDA: 600 urban images for the Source dataset (TS), 300 urban and 300 rural images for the Target dataset (T1), 100 urban images for the same domain Test dataset (T2) and 100 rural images for the different domain dataset (T3). Once again, we conducted a centralized search finding the following configuration: Learning rate = $1 \cdot 10^{-4}$, Batch size = 4, Weight decay = $1 \cdot 10^{-4}$ and setting a baseline. We then proceeded to test some configurations of the Self-Training framework with pseudo-labels, using 20 rounds and considering all images at each one, obtaining the results showed in Table 8. Even with this dataset, our method still improves the performance when evaluated on the target. Also after this experiment qualitative results are presented in Figure 5.

7. Discussion and conclusions

In this analysis, we conducted an extensive exploration by means of various frameworks, configurations, models, and datasets. The incorporation of pseudo-labels led to modest improvements in our results. However, we believe that this approach could be more effective with the availability of a larger number of clients and a greater number of images. In the representative Figure 3, these qualitative results show improvements in the segmentation task at each level of experiments. Our observations indicate that Transformers show more expressiveness compared to traditional methods. Furthermore, they exhibit better compatibility with the FDA technique, possibly due to synergistic effects with the attention mechanism. After exploring the LoveDA dataset, and have shown that pseudo-labels improve the results, further researches on how to include satellite images in the autonomous driving framework would be interesting. Moving forward, several paths for future research should be explored. One such direction could involve investigating alternatives to the FedAvg approach we employed in this study. The main problem to solve in this scenario remains the statistical heterogenity of the clients data and this should be the key aspect to focus on.

References

[1] Tian Li et al. “Federated Learning: Challenges, Methods, and Future Directions”. In: *CoRR* abs/1908.07873 (2019). arXiv: 1908.07873. URL: http://arxiv.org/abs/1908.07873.

[2] Donald Shenaj et al. “Learning Across Domains and Devices: Style-Driven Source-Free Domain Adaptation in Clustered Federated Learning”. In: *arXiv preprint* (2022). WACV 2023; 11 pages manuscript, 6 pages supplemental material. arXiv: 2210.02326 [cs.CV]. URL: https://arxiv.org/abs/2210.02326.

[3] Shijie Hao, Yuan Zhou, and Yanrong Guo. “A Brief Survey on Semantic Segmentation with Deep Learning”. In: *Neurocomputing* 406 (2020), pp. 302–321. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.11.118. URL: https://www.sciencedirect.com/science/article/pii/S0925231220305476.

[4] Lidia Fantauzzo et al. *FedDrive: Generalizing Federated Learning to Semantic Segmentation in Autonomous Driving*. 2022. arXiv: 2202.13670 [cs.CV].

[5] H. Brendan McMahan et al. *Communication-Efficient Learning of Deep Networks from Decentralized Data*. 2023. arXiv: 1602.05629 [cs.LG].

[6] Yanchao Yang and Stefano Soatto. *FDA: Fourier Domain Adaptation for Semantic Segmentation*. 2020. arXiv: 2004.05498 [cs.CV].

[7] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. *Bidirectional Learning for Domain Adaptation of Semantic Segmentation*. 2019. arXiv: 1904.10620 [cs.CV].

[8] Emanuele Alberti et al. “IDDA: A Large-Scale Multi-Domain Dataset for Autonomous Driving”. In: *IEEE Robotics and Automation Letters* 5.4 (Oct. 2020), pp. 5526–5533. DOI: 10.1109/lra.2020.3009075. URL: https://doi.org/10.1109%5C%2Flra.2020.3009075.

[9] Stephan R. Richter et al. “Playing for Data: Ground Truth from Computer Games”. In: *CoRR* abs/1608.02192 (2016). arXiv: 1608.02192. URL: http://arxiv.org/abs/1608.02192.

[10] Liang-Chieh Chen et al. *Rethinking Atrous Convolution for Semantic Image Segmentation*. 2017. arXiv: 1706.05587 [cs.CV].

[11] Mark Sandler et al. *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. 2019. arXiv: 1801.04381 [cs.CV].

[12] *Transforming and augmenting images*. URL: https://pytorch.org/vision/stable/transforms.html.

[13] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[14] Enze Xie et al. *SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers*. 2021. arXiv: 2105.15203 [cs.CV].

[15] Edward Akito Carlos, Raphael Pinard, and Mitra Hassani. *Over-the-Air Federated Learning in Satellite systems*. 2023. arXiv: 2306.02996 [cs.LG].

[16] Junjue Wang et al. “LoveDA: A Remote Sensing Land-Cover Dataset for Domain Adaptive Semantic Segmentation”. In: *CoRR* abs/2110.08733 (2021). arXiv: 2110.08733. URL: https://arxiv.org/abs/2110.08733.