Federated Semantic Segmentation for self driving cars

Elia Fontana
S290188@studenti.polito.it

Lorenzo Bellino
S309413@studenti.polito.it

Martina Caputo
S299458@studenti.polito.it

Abstract

This paper proposes a federated semantic segmentation framework for self-driving cars that leverages the power of federated learning to train a deep neural network using segmented datasets obtained from multiple vehicles while preserving data privacy. Traditional methods of semantic segmentation rely on centralized computing, which is impractical in real-world scenarios. The proposed framework includes a central server that coordinates the training process and multiple participating vehicles that provide their segmented data. In addition to the proposed framework, the paper applied domain generalization techniques such as Fourier Domain Adaptation (FDA) to improve the model's generalization and robustness, as well as implemented a pseudo labelling technique to overcome the challenge of unlabelled data from the participating vehicles in a real-world applications. The combination of these techniques with federated learning resulted in a robust and efficient semantic segmentation framework for self-driving cars.

1. Introduction

The development of self-driving cars has been rapidly advancing over the years, and one of the critical components of an autonomous vehicle is its ability to perceive its surroundings accurately. Semantic Segmentation (SS) is a crucial task that enables the vehicle to understand its environment and make appropriate decisions based on the input received. However traditional methods of semantic segmentation rely heavily on centralized computing which may not be practical in real-world scenarios.

Federated Learning (FL) has emerged as a promising solution to address the challenges associated with centralized computing. In this paper, we propose a federated semantic segmentation framework for self-driving cars. The proposed framework leverages the power of federated learning to train a deep neural network that can perform semantic segmentation on the segmented datasets obtained from multiple vehicles. Through the use of distributed training across multiple remote clients, each with their own distinct segmented dataset, the proposed framework enables a shared model to be learned through the aggregation of updates on a central server. The tested aggregator implemented FedAVG calculating the average of the updates from each client in order to update the parameters.

The segmentation task is performed on each client's dataset which was chosen based on two different approaches, the first is represented by a uniform distribution of images in respect to the represented city and the second one is a more real to life approach, represented by an heterogeneous distribution where a client will perform the segmentation task only on images from a particular city.

But the success of current segmentation techniques depends on large-scale densely-labeled datasets that are prohibitively expensive to be collected in reality. An intuitive method to address this issue is to train the server model on a synthetic dataset where it is easier to obtain ground truth labels and then transfer the learned model to a real world domain. However simply training the model on a different dataset is not enough. For this reason in addition to the proposed federated semantic segmentation framework, we also applied domain generalization techniques such as Fourier Domain Adaptation (FDA). This tecnique allows the model to be better generalized and robust by extracting styles from the real world target dataset represented by the Cityscapes dataset and applying them to the training one composed of sythetic images found in the GTA5 dataset.

Furthermore, to overcome the challenge of unlabelled data from the participating vehicles, we implemented a pseudo labelling technique. The pseudo labelling approach involves using the trained model to predict the labels of the unlabelled data from the participating vehicles and then using these predicted labels to train the model further. This technique enabled us to leverage the unlabelled data from the participating vehicles, leading to good segmentation accuracy and better utilization of available data resources even in condition of unlabelled data.

The combination of these techniques with federated learning resulted in a robust and efficient semantic segmentation framework for self-driving cars.

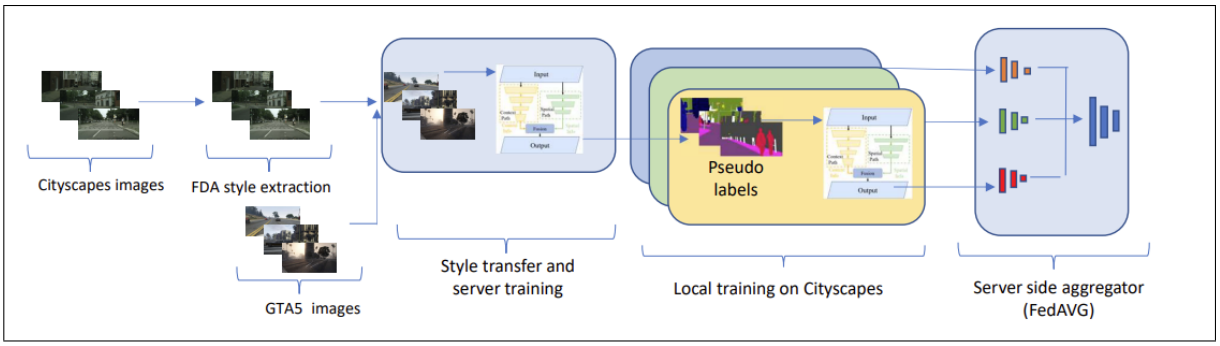


Figure 1. Basic illustration of the complete model, where the server is trained from source images with styles from GTA and clients are trained using pseudo labels generated by the server. In the end all the parameters are aggregated with FedAVG and returned to the server.

2. Related Work

2.1. Semantic Segementation

In computer vision exists a large number of segmentation task [1], in particular Semantic Segmentation is a computer vision task that involves the classification pixel by pixel of an image into a set of predefined mutually exclusives categories or classes [2]. Unlike object detection or classification, semantic segmentation aims at the production of a dense pixel-wise labelling of an image, which then can be used to perform a variety of tasks; in particular in this paper the focus is on the task of autonomous driving. In order to reach this goal typically the neural network is trained on a large dataset of labelled images such as the one provided in the Cityscapes dataset [3] and the GTA5 dataset [4].

2.2. BiseNet V2

BiseNet V2 is a lightweight neural network architecture used for semantic segmentation, which is the task of assigning a label to each pixel in an image. It was proposed for Real-time Semantic Segmentation by Changqian Yu et al [1]. The network is an improvement over the original BiSeNet architecture.
The key idea behind BiseNet V2 is to use a bilateral network that combines spatial and channel-wise information for better feature representation. The network has two pathways: a context path that captures global context information using dilated convolutions, and a spatial path that captures spatial details using spatial pyramid pooling. The outputs of both pathways are then fused using a guided aggregation module that learns to combine features in a context-aware manner.
Bisent V2 is designed for real-time semantic segmentation on resource-constrained devices, such as mobile phones or embedded systems. The network achieves state-of-the-art performance on several benchmark datasets, while being efficient in terms of computation and memory usage

2.3. MobileNet V2

MobileNet [5] is a lightweight efficient network for mobile and embedded visual application were is often used as a reference model. Due to its relative small complexity still maintaining high level of accuracy it can be used as the main network for semantic segmentation tasks resulting in faster training and inference time, especially for real time application on smartphones and embedded devices.

2.4. Federated Learning

Federated Learning is a type of machine learning where the model is trained across multiple decentralized devices , they could be phones, servers , IoT devices or, like the case study of this paper, vehicles. The training is performed without the need to exchange the raw data used but only the updated parameters are transmitted between client and server. The aim of this approach is to address the challenges of data privacy and security. This approach consist of an initial phase where the model is trained on a subset of the dataset and then it is sent to each client in order to perform further training step. In this second phase each client receives the model and performs the training of locally available images and updates the model's parameter locally. In the third and final phase the learned parameters on each client are sent back to the server where they are aggregated to form a new updated model at which point the steps are repeated until the model reaches a satisfactory level of accuracy.
The advantage of this approach are many, even without considering the privacy concern, for example the reduction in the data transmitted and the possibility of asynchronous transmission and the inherited fault tolerance [6].

2.5. FedAvg

FedAvg [7] is the alghoritm behind the federated learning process previously described, it allows multiple clients to train a shared model locally and then share the updates with the server, which perform aggregation on the learned parameters from the clients averaging the updates. The key innovation of FedAvg is that it uses a simple yet effective averaging technique to combine the update from different devices, which reduces the amount of communication needed compared to other federated learning methods by up to two order of magnitudes while still maintaining good performance on a large number of benchmark datasets.

2.6. FedDrive

One of the most notable implementation of a federated semantic segmentation task in autonomous driving is FedDrive [8] a benchmark consisting of three settings and two datasets, incorporating the real-world challenges of statistical heterogeneity and domain generalization.

2.7. Domain Adaptation

Domain Adaptation is a technique used in machine learning to improve the performance of a model when it is applied to a new, unseen domain that may be different from the one it was trained on. In traditional machine learning it is assumed that the training and testing data come from the same distribution. However this assumption may not be true in real world application where it is common that data distribution in the testing domain is different from the one in the training domain. To address this issue domain adaptation techniques attempt to learn a model that can be well generalized to the target domain. The main idea behind this technique is to transfer most of the relevant information from the training domain to the target domain by leveraging various approaches [9]. Domain adaptation is widely used in various applications, especially where the availability of labelled data in the target domain is limited and it is proven that can improve the robustness and generalization of the model and ultimately enhance its performance in real world scenarios.

2.8. FDA

In many real-world scenarios, it is common for the distribution of the data used for training a semantic segmentation model to differ from the distribution of the data that the model will be applied to. This is known as the domain shift problem. When there is a significant domain shift, a model trained on one dataset may perform poorly on another dataset, even if the two datasets contain similar classes. FDA [10] aims to address this problem by adapting a model trained on a source dataset to a target dataset. It does this by leveraging the Fourier domain, which is a mathematical representation of the frequency components of an image. Specifically, FDA learns a Fourier-based transformation that maps the source and target domains to a common feature space. This transformation is designed to align the frequency components of the two domains, which helps to reduce the domain shift. Once the transformation is learned, it is applied to the feature maps generated by the segmentation model, effectively adapting the model to the target domain. This allows the adapted model to generate accurate segmentations on the target dataset, even if it was not specifically trained on that dataset. Overall, FDA is a promising approach for addressing the domain shift problem in semantic segmentation, and it has shown promising results in several real-world scenarios.

2.9. Pseudo-labels

Pseudo-labeling [11] is a semi-supervised learning technique in machine learning where a model is trained using both labeled and unlabeled data. In pseudo-labeling, the model first trains on the labeled data as in a supervised learning setting. It then uses this trained model to predict labels for the unlabeled data points. These predicted labels are referred to as ”pseudo-labels.”
The next step is to combine the labeled and pseudo-labeled data and use them to train a new model. The model is then iteratively retrained on the combined dataset until convergence through a process known as ”self-trainig”. Pseudo-labeling can be useful when there is a limited amount of labeled data, but a larger amount of unlabeled data is available [12].

2.10. Federated source-Free Domain Adaptation

Federated Source-Free Domain Adaptation (FFreeDA) is a technique used in machine learning to improve the performance of a model when it is applied to a new dataset from a different domain without requiring access to the original source data. In traditional Domain Adaptation, the model is adapted to the new domain using labeled data from the original source domain. However, in some situations, it may not be possible to access the original source data due to privacy or regulatory concerns. FFreeDA addresses this problem by using a federated learning approach. In this approach, multiple clients hold their own private dataset. These clients then collaborate in a decentralized manner to adapt the model to the new domain. The key idea behind Federated Source-Free Domain Adaptation is to leverage the similarities across the different client datasets to learn domain-invariant features that can improve the model’s

ability to generalize to the new domain. This is achieved by combining the local updates from each client in a privacy-preserving manner.

2.11. LADD

LADD [13] stands for Layer-wise Attention-based Domain-adaptive Dropout,it's a technique used to improve the performance of the network when applied to new datasets that are different from the ones used during training. It uses domain adaptation that helps the model learn how to make accurate predictions on new data from a different domain than the one used during training. The LADD algorithm applies a technique called dropout, which randomly drops out certain nodes in the neural network during training to prevent overfitting. It then applies a layer-wise attention mechanism to help the model focus on the most important features in each layer of the network. This helps the model learn more robust and domain-invariant features that are relevant to the new data.

3. Experiments

3.1. Datasets and metrics

Cityscapes

We evaluate our framework using Cityscapes as dataset. It contains a large collection of high-resolution images of urban scenes, with a total of 5,000 images captured across 50 different cities. Each image has a resolution of 2048x1024 pixels. There are 30 different object classes that are labeled in the dataset, including common objects found in urban environments such as roads, sidewalks, buildings, trees, and vehicles. For our purpose we consider reduced version of the dataset, with a subset of 750 images belonging to 21 different cities and 19 classes instead of the previous 30, setting the remaining one as don't care. We initially train the model using 2 different train/test split partitions:

- Partition A: Test 2 random images from each city, Train the remaining images, 708 train 42 test
- Partition B: 500 train 250 test

Above this 2 partitions for the Federated task we create two clients typology:

- uniform: each client contains images belonging to different cities
- heterogeneous: to each client, images of a singular city are associated each client contains at most 20 images

GTA5

GTA5 is a dataset of images that is commonly used for training and testing computer vision algorithms. The dataset contains images captured from the popular video game "Grand Theft Auto V," which is set in a virtual world resembling a city. The images in the GTA5 dataset are labeled with different categories, such as road, building, sky, and vehicle. This means that each object in the image is assigned a label that describes what it is. GTA5 contains a total of 25,000 images with a resolution of 1,914×1,052, which is smaller compared to other datasets such as Cityscapes. As well as for Cityscapes we use a subset of 500 images.

SS metric

The metric we used to evaluate the Semantic Segmentation model is the MIoU. Mean Intersection-Over-Union (MIoU) measures the average overlap between the predicted segmentation mask and the ground truth mask for all classes in the dataset. To calculate MIoU, the Intersection-Over-Union (IoU) is first calculated for each class. IoU (also known as Jaccard Index) measures the overlap between the predicted segmentation mask and the ground truth mask, it is calculated by dividing the area of overlap between the two masks by the area of union between the two masks. Once the IoU is calculated for each class, the MIoU is computed as the average IoU across all classes in the dataset. This provides a single number to evaluate the overall performance of the algorithm across all classes. MIoU is a useful metric because it takes into account the performance of the algorithm for all classes in the dataset, rather than just a single class. It provides a more complete picture of the algorithm's performance.

3.2. Experiment Implementation

Centralized baseline

The network adopted for the segmentation task is the pretrained BiSeNetV2. We used as loss function the Cross Entropy Loss and Stochastic Gradient Descent (SGD) as optimizer. The learning rate remain constant throw the entire training procedure. In order to make images more manageable to handle we resize them to 1024×512. To augment data we apply different tranforms, in particular *Horizontal Flip* with with a 5% probability, *Random Rotation* of $5\,\mathrm{deg}$, *Central Crop* and *Color Jitter* which modifie hue, brightness, contrast and saturation.

We tried different hyperparameters configuration but the main focus of our testing was done on different values of learning rate. In the end for the hyperparameters the following were chosen as the baseline for all the subsequent steps:

- *Momentum*: 0.9
- *Weight decay*: $5 \cdot 10^{-4}$
- *Learning rate*: 0.05

- *Epochs*: 10
- *Batch size*: 8

Figure 2. Training loss at different learning rate for Partition A

Figure 3. Training loss at different learning rate for Partition B

With this values and transforms we obtained the highest value of MioU in both test partitions, reaching 35.79% for partition **A** and 31.32% for partition **B**.

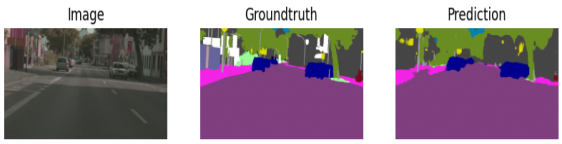


Figure 4. Prediction label after the training for partition A

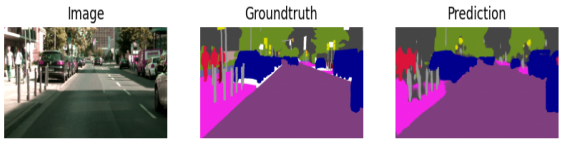


Figure 5. Prediction label after the training for partition B

Federated learning

In this step the focus our attention is on a Federated baseline. We take into account both for partition A and B two different type of data distribution between the clients. A first distibution represent an ideal situation where each clients has access to a uniform distribution of samples between the city, while the second distribution is more similar to a real world scenario where each client has access to samples from a single city.
The clients were trained on all four of this division on a mini batch of maximum 20 images. For this experiment every round 5 clients were selected randomly from a pool of available clients and where trained for 2 epochs, this was repeated for 50 rounds. This parameter were chosen because they represent well enough the real world were we could not expect a client to train for a large number of epochs. The number of client to train on each round was chosen because we observed that over a certain minimum threshold the number of client picked does not influence performance. This are the chosen parameters:

- *Rounds*: 50
- *Client per round*: 5
- *Epochs per client*: 2

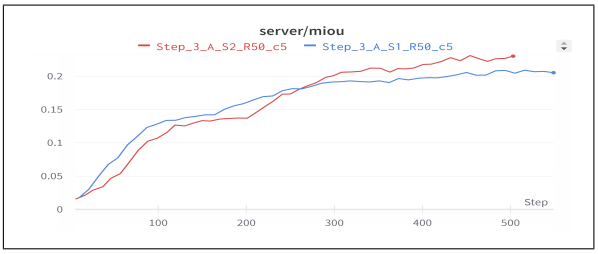


Figure 6. Val miou on server model for Partition A Uniform (S1 in blue) and Heterogeneuos (S2 in red) clients

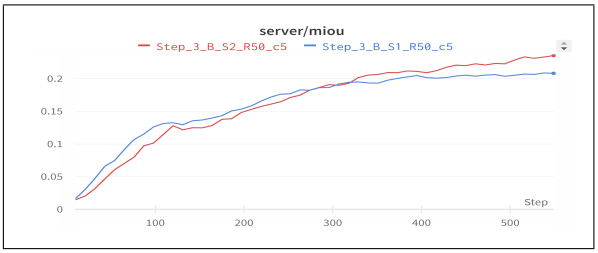


Figure 7. Val miou on server model for Partition B Uniform (S1 in blue) and Heterogeneuos (S2 in red) clients

As shown in the graphs for both partition A and B we obtain sightly better performance for the heterogeneous split. Given best model for each partition the final mIoU results are:

	Uniform	Heterogeneuos
partition A	22.67%	22.06%
partition B	20.96%	21.22%

Table 1. MioU on different Partitions and Splits

Domain Adaptation

Turning again our attention on a centralized baseline we now consider a important issue in Semantic Segmentation tasks. In fact generating semantic labels for real images is a time consuming operation. For this reason we can use as training dataset a synthetic one, on witch the label construction is an easy task that can be completed using a limited amount of resources. As first step we train the model using raw images of GTA testing its effectiveness on Cityscapes dataset. We applied the same transforms for both datasets and the values of the parameter chosen for all the experiments in this step are:

- *Momentum*: 0.9
- *Weight decay*: $5 \cdot 10^{-4}$
- *Learning rate*: 0.05
- *Epochs*: 10
- *Batch size*: 8

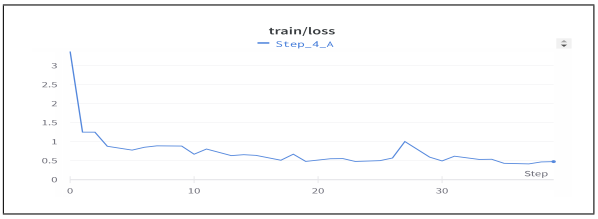


Figure 8. Train loss on partition B

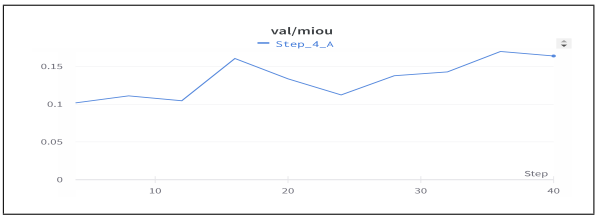


Figure 9. Val miou on partition A

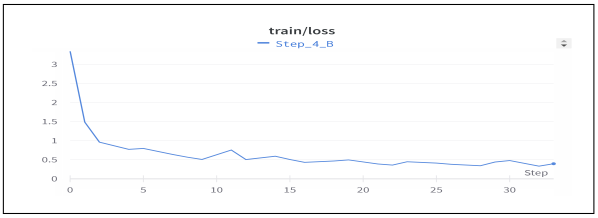


Figure 10. Train loss on partition B

As expected we have a considerable performance drops because of the domain shift, with a partition **A** 18.47%, and partition **B** 16.04%.

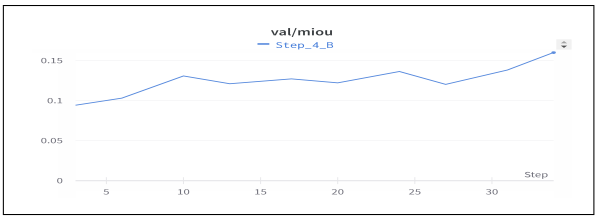


Figure 11. Val miou on partition B

To address the domain shift problem in the following step we use a simple but effective domain adaptation technique, called FDA, to transfer the Cityscapes' style on GTA images. We firstly create a style bank based on Citiscapes' images. After that, during the training phase, we extract a style from the bank and tranfer the style on the GTA training images. For FDA we tried different window size (mainly 1x1, 3x3, 5x5) and noticed small differences between them as displayed in the figure below.

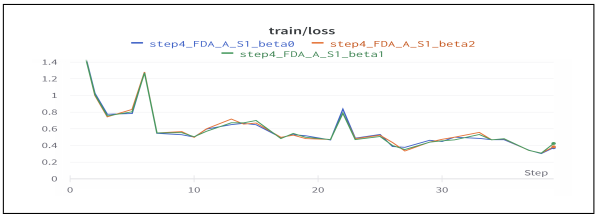


Figure 12. Val miou for different window size (beta), partition B

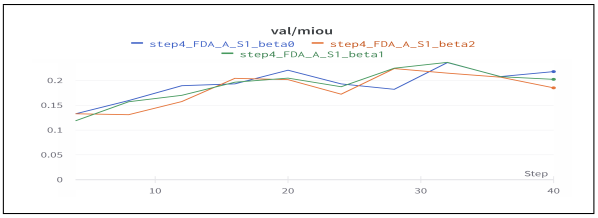


Figure 13. Val miou for different window size (beta), partition B

Our final choice for the beta value is 1 that corresponds to a 3x3 window size. In the images below is possible to see an example of a style transfer from the Cityscapes dataset to a GTA5 sample.

(a) Source (b) Target (c) Target with transferred style

Figure 14. Example of Style transfer

The following graphs represent different value of loss

and validation mIoU calculated on both test partition and for an heterogeneuos client distribution as well as a uniform distribution.

Figure 15. Train loss for Partition A Uniform (S1 in blue) and Heterogeneuos (S2 in red)

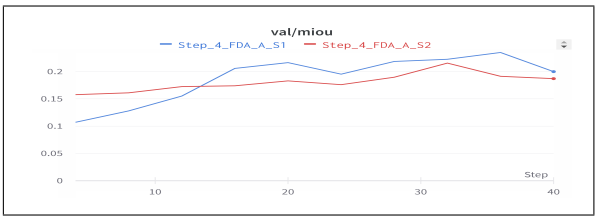


Figure 16. Val miou for Partition A Uniform (S1 in blue) and Heterogeneuos (S2 in red)

Figure 17. Train loss for Partition B Uniform (S1 in blue) and Heterogeneuos (S2 in red)

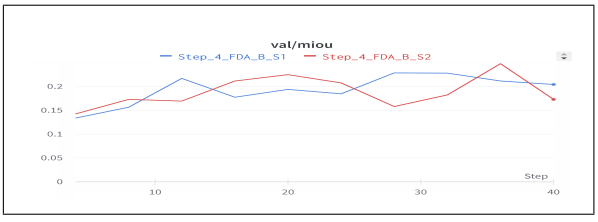


Figure 18. al miou for Partition B Uniform (S1 in blue) and Heterogeneuos (S2 in red)

From the results displayed in Table 2 we notice a small difference of mIoU (few percentage points) between heterogeneous and uniform clients.

Federated Self-Training Using Pseudo-labeling

In the following step we try to create a model closer to a real application. In fact in real world the images captured by client cars are obviously unlabelled. The Self-Training using Pseudo-labeling is a technique that try to address this issue. We use the pre-trained model from the previous step to generate the pseudo-label employed as groundtruth. Every T rounds it is possible to update the teacher model. Modifying T in order to select the best hyperparameter value.

	Uniform	Heterogeneuos
partition A	22.98%	21.41%
partition B	20.84%	19.29%

Table 2. Results 4.4

- *Momentum*: 0.9
- *Weight decay*: $5 \cdot 10^{-4}$
- *Learning rate*: 0.05
- *Epochs*: 10
- *Batch size*: 8
- *Rounds*: 50
- *Clients per round*: 5
- *Epochs per client*: 2

Update the teacher model at each round or generally in a small time window T, leads to worse results, On the contrary a large time window could be a good choice. In the graphs below it can be seen for each of the four partition how the window behave given the fixed values $T = 0$ (teacher never update), $T = 1$ (teacher update every round), $T = 10$ (teacher updates every 10 rounds).

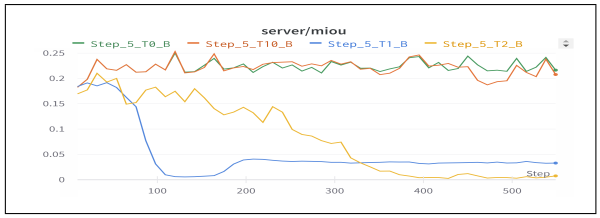


Figure 19. Val miou for different T, partition B

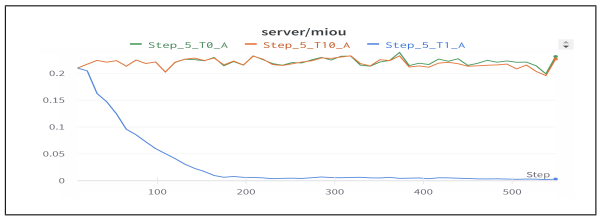


Figure 20. Val miou for different T, partition A

Due to resource constrains the results are calculated on

50 rounds of training and it can be seen that for value of $T >= 10$ the performance of the model remain stable and in some cases improve by a small percentage, on the contrary for small value of T the performance degrade in relative small number of rounds. This performance degradation is due to the fact that the clients are using inaccurate labels for training and continue to learn inaccurate pattern throughout the rounds.
In the below graphs is shown the mIoU for each partition.

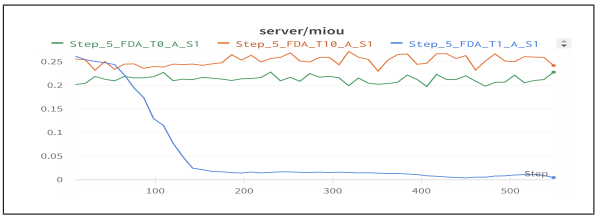


Figure 21. Val miou for different T, partition A Uniform

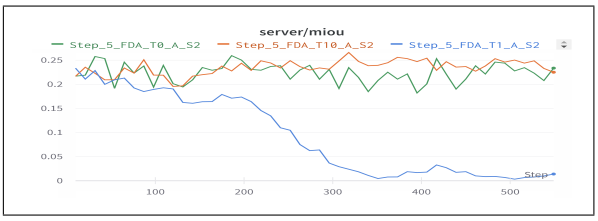


Figure 22. Val miou for different T, partition A Heterogeneuos

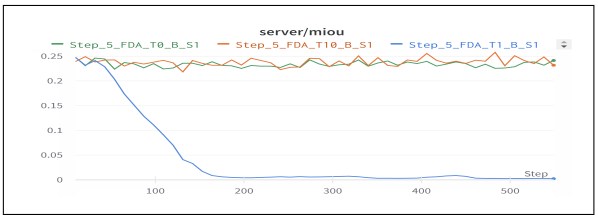


Figure 23. Val miou for different T, partition B Uniform

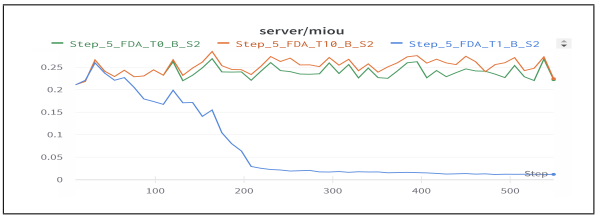


Figure 24. Val miou for different T, partition A Heterogeneuos

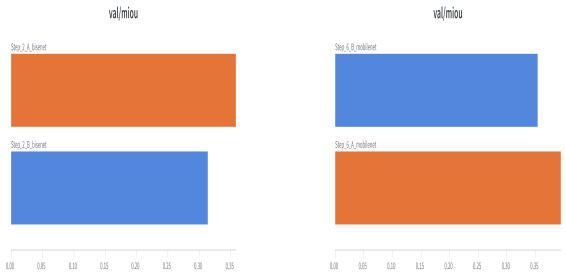
As it can be seen from the table below, we reach similar results for both $T = 0$ and $T = 10$ independently from the partition type. Since on average the model performs slightly better for $T = 10$ we choose this value.

T	0	1	10
partition A Unif	23.53%	0.59%	24.72%
partition A Heter	22.71%	1.09%	22.83%
partition B Unif	21.69%	0.24%	21.63%
partition B Heter	21.77%	1.15%	22.84%

Table 3. Results 5.4

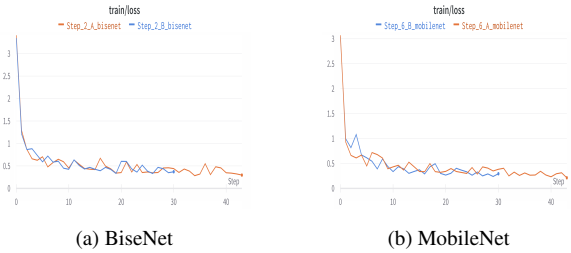
MobileNet V2 as an alternative network

Semantic segmentation for autonomous vehicle is a task that rely heavily on real-time image processing on device with a limited amount of computational power, for this purpose it seems reasonable to explore other lightweight network. One example is Mobilenet V2 with DeepLab V3 head [14]. This is a lightweight efficient network which can achieve elevated level of accuracy at a high speed with minimum computation. For this experiment as a proof of concept we repeted the centralized configuration using MobileNet instead of BiseNet.



(a) BiseNet (b) MobileNet

Figure 25. Comparison for both partition between BiseNet and MobileNet



(a) BiseNet (b) MobileNet

Figure 26. Comparison for both partition between BiseNet and MobileNet

As a result we obtained and improvement of 4% on the

validation MioU for the same amount of epochs on both test partitions.

3.3. Conclusion

At the end of our study, having considered various modeled scenarios it is clear that the best results for semantic segmentation are obtained in a centralized environment but since this model is not applicable in real world application the presented federated learning model are a viable solution for this kind of task in a self driving scenario. In conclusion we demonstrated that a federated learning model for semantic segmentation with decent accuracy result is achievable and self training methods with pseudo labels are a good solution to the problem caused by unlabelled data on the client's datasets while diminishing the amount of data transferred between server and devices. However, given the numerous computational resource limitations and the added complexity of these models, the obtained results are slightly less performing. We are confident that with more data and time, significantly better results will be achievable.

References

[1] Changqian Yu1, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *CoRR*, abs/2004.02147, 2020. 2

[2] Shijie Haoa, Yuan Zhoua, and Yanrong Guoa. A brief survey on semantic segmentation with deep learning. *Neurocomputing*, 406:320–321, 2020. 2

[3] Marius Cordts at al. The cityscapes dataset for semantic urban scene understanding. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[4] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. *Computer Vision-ECCV 2016: 14TH European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14 Springer International Publishing*, 2016. 2

[5] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 406:4510–4520, 2018. 2

[6] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37:50 – 60, 2019. 2

[7] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20 th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 54, 2017. 3

[8] Lidia Fantauzzo at al. Feddrive: Generalizing federated learning to semantic segmentation in autonomous driving. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022. 3

[9] Sicheng Zhao at al. A review of single-source deep unsupervised visual domain adaptation. *IEEE transactions on Neural Networks and Learning Systems*, 2020. 3

[10] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 3

[11] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 3

[12] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. 2021. 3

[13] Donald Shenaj at al. Learning across domains and devices: Style-driven source-free domain adaptation in clustered federated learning. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 4

[14] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 8