Federated Semantic Segmentation for Self-driving Cars

Gabriele Cassetta
s296284

Selen Akkaya
s289332

Mengdi Yang
s288434

***Abstract*—The process of semantic segmentation is crucial in making self-driving vehicles autonomous by allowing them to identify different objects within their surroundings by assigning pixels to specific categories. However, this process uses sensitive data collected from cars, making the protection of user privacy a top priority. To address this concern, an approach called Federated Learning has been proposed to learn a global model while keeping data private and utilizing data from multiple remote devices. In this study, a Federated Learning method was applied to semantic segmentation in self-driving cars using the BiSeNet [1] model. Experiments were conducted using the Cityscapes dataset, and a method to consider data privacy for individual users in real-world scenarios was proposed by using various techniques such as Federated Averaging, Fourier Domain Adjustment, and the creation of pseudo-labels.**
GitHub Repository

I. Introduction

In this paper we propose an application of Federated Learning for semantic segmentation in self-driving cars by using the BiSeNetV2 [1] and MobileNetV2 models and the Cityscapes and GTA5 datasets, by employing different methods such as Federated Averaging, Fourier Domain Adaptation, and pseudo-labels generation.

II. Related work

A. Semantic Segmentation

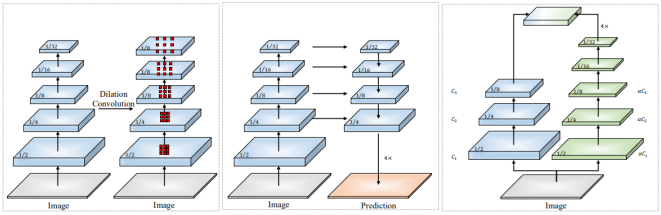
Semantic segmentation consists in subdividing an image into multiple segments which correspond to the same category, without distinguishing different objects, as other types of segmentation do [2]. The main difference between semantic and other types of image segmentation is that semantic segmentation assigns a semantic label, such as ”car” or ”road,” to each pixel in the image. In other words semantic segmentation aims to parti- tion an image into mutually exclusive subsets. Finally, it’s worth noting that semantic segmentation is a very active research area and new techniques and architectures are being proposed frequently. [4].

B. BiSeNet V2

The Bilateral Segmentation Network (BiSeNet V2) is a two-pathway architecture that is designed for real-time semantic segmentation [1]. One pathway, called the detail branch, is designed to capture spatial details with wide channels and shallow layers. The other pathway, called the semantic branch, is intended to extract categorical semantics with narrow channels and deep layers. The semantic branch only needs a large receptive field to capture semantic context, while the detail branch provides the additional detail information. This allows the semantic branch to be made lightweight with fewer channels and a fast-downsampling strategy. The two types of feature representation are combined to create a stronger and more comprehensive feature representation. This design results in an efficient and effective architecture for real-time semantic segmentation.

(a) Image (b) Semantic segmentation

Fig. 1. An example of semantic segmentation. Source image in the left and corresponding mask in the right [4]



(a) Dilation Backbone (b) Encoder-Decoder Backbone (c) Bilateral Segmentation Backbone

Fig. 2. Illustrations of different backbone architectures are shown,
(a) is a dilation-based network, which removes downsampling operations and upsamples the corresponding convolution filters. However, this architecture has high computational complexity and memory usage.
(b) is an encoder-decoder-based network, which adds extra top-down and lateral connections to recover high-resolution feature maps. However, these connections increase the memory access cost. To achieve both high accuracy and efficiency simultaneously, a new architecture called
(c) Bilateral Segmentation network is designed. This architecture has two pathways, a Detail Branch for capturing spatial details and a Semantic Branch for extracting categorical semantics. The Detail Branch has wide channels and shallow layers, while the Semantic Branch has narrow channels and deep layers, which can be made lightweight by a factor.

C. Federated Learning

Federated Learning is a method of machine learning that enables the training of a global model while protecting the privacy of the data used for training. Instead of collecting data

from all devices and centralizing it, Federated Learning allows multiple devices to train a shared model locally and share updates with a central server, which aggregates the updates to improve the global model.

In the context of self-driving cars, Federated Learning can be used to train semantic segmentation models without compromising the privacy of the data collected from individual cars [3].

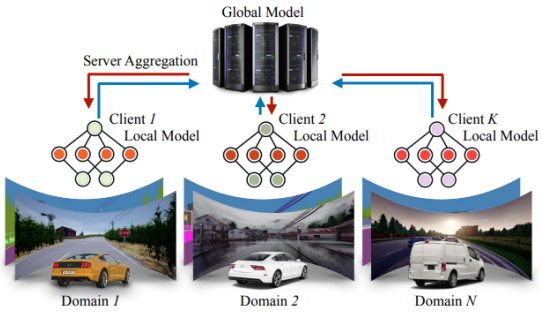


Fig. 3. An illustration of multiple vehicles (clients) driving in different environments, communicating with a central server to learn a global model while protecting their individual data. In each communication cycle, a group of clients trains on their local data and shares only the update with the server, which then sends back the updated global model [5].

1) FedAvg: FedAvg is a algorithm allows multiple devices to train a shared model locally and share updates with a central server. The central server, then, aggregates the updates to improve the global model [6]. The central server selects a subset of devices to participate in the current round of training. Each selected device performs training on its local data using the current global model and sends the updated model parameters back to the server. The server averages the updated model parameters from all participating devices to obtain the new global model. Then, the new global model is sent back to all devices, including those that did not participate in the current round of training. This process is repeated multiple times with different subsets of devices participating in each round of training. However, the algorithm has some limitations, since it assumes that all clients have similar data distribution, which might not be the case in practice.

D. Domain Adaptation

The purpose of domain adaptation is to take a machine learning model that was previously trained on a source domain with a large amount of labeled data and adjust it to work on a target domain where there may be limited or no labeled data available. Fine-tuning is a crucial method used in this process to adapt the pre-trained model to the new domain. The pre-trained model from the source domain is used as the starting point, and its parameters are refined on the target domain data through fine-tuning. This fine-tuning allows the model to better fit the target domain's data distribution, enabling it to perform effectively in the target domain despite the limited labeled data [9].

1) FDA: Fourier Domain Adaptation for Semantic Segmentation: Fourier Domain Adaptation (FDA) is a technique used in semantic segmentation that utilizes the Fourier transform [10] to adapt a model from one domain to another. In the case of FDA for semantic segmentation, the method transforms images from the spatial domain to the frequency domain using the Fourier transform. The images are then decomposed into different frequency components, which allows for better capture of low-level image features that are invariant across domains. The frequency domain information is then utilized by FDA to align the source and target domains, enabling a model trained on the source domain to be adapted to the target domain. This approach has been demonstrated to enhance the performance of semantic segmentation models in target domains where labeled data is limited. FDA uses window size method that serves the purpose of ensuring that the Fourier transform, which is used to convert the images from the spatial domain to the frequency domain, operates on a defined region of the image. This helps to accurately capture the local features of the image, which are crucial for adapting the source and target domains for semantic segmentation. The window size is a crucial factor in the performance of the FDA method. A smaller window size may lead to a more precise decomposition of the local features, while a larger window size may incorporate more global information. However, larger windows result in a more computationally intensive Fourier transform. Thus, the choice of window size is a balancing act between accuracy and computational efficiency.

Hence, FDA is a method for semantic segmentation that uses the Fourier transform to perform domain adaptation, thus improving the performance of semantic segmentation models in target domains with limited labeled data [11].

E. Pseudo-Labels

Pseudo-labels are fabricated or simulated annotations created for the unlabeled data in a semi-supervised learning approach. The algorithm produces these labels by making predictions on the unlabeled data and then using those predictions as pseudo-labels for training.

Teacher-student models involve using a pre-trained "teacher" model to produce labels for a "student" model. The student model is trained with these labels with the aim of learning from the teacher's knowledge and eventually surpassing it. The teacher model can be adjusted or kept fixed during this process [12].

1) Federated self-training using pseudo-labels: Federated self-training using pseudo-labels [12] is a method of training machine learning models in a federated setting using a combination of self-supervised learning and pseudo-labeling. In this approach, each participant in the federated network trains a local model on their own data and generates pseudo-labels for their unlabeled data. These pseudo-labels are then shared among the participants to create a consolidated dataset, and the local models are fine-tuned on the combined dataset. The goal of this method is to leverage the collective information

from the entire federated network to improve the performance of the local models, while also preserving data privacy.

F. MobileNetV2

MobileNetV2 [13] is a deep learning model that has been developed for use on mobile devices and other systems with limited resources. In the context of semantic segmentation, it is a popular choice due to its ability to provide both accuracy and efficiency.

MobileNetV2 can be used as the main network for a semantic segmentation model. Its lightweight design, with depthwise separable convolutions, reduces the number of parameters and computational complexity compared to traditional convolutions, resulting in faster training and inference times. This makes it a suitable choice for real-time applications on mobile devices and similar resource-constrained systems. Therefore, MobileNetV2 is a reasonable choice for semantic segmentation since it offers good accuracy while also being computationally efficient.

III. Data

In this study subsets of Cityscapes dataset and GTA5 dataset are used. Data is composed of two different subfolers corresponding to raw images and corresponding labels.

A. Cityscapes dataset

Cityscapes is a widely used dataset for semantic segmentation, it contains real-world photos taken in the streets of 50 different cities under good weather conditions. The dataset is split into 2975 images for training and 500 for testing, all of which are labeled with annotations [7].

B. GTA5 dataset

The GTA5 dataset is a collection of 24966 synthetic images that have been labeled with pixel-level semantic annotations. These images were generated using the open-world video game Grand Theft Auto 5, and all feature a car's perspective of American-style virtual cities. [9].

IV. Implementation

A. Partition CityScapes Into Train AND Test

We started the project with implementing 2 different data-loaders for training and test set.

1) Partition A: Partition A is composed of Test and Train partitions: Test partition contains 2 random images from each city and Train partition contains the remaining images.

In total, we obtain 708 images in Train partition and 42 images in Test partition.

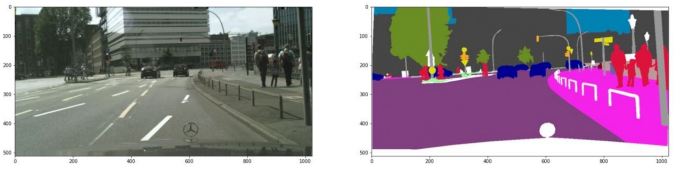


Fig. 4. Visualized form of a random image from partition A - train split and it's corresponding label image [7].

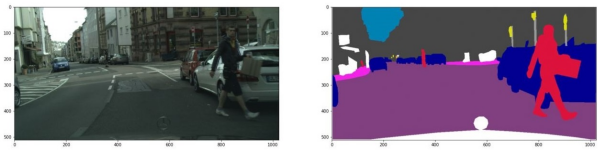


Fig. 5. Random image from partition B - train split and it's corresponding label [7].

2) Partition B: Partition B is composed of Test and Train partitions: Test partition contains the images specified in the val.txt file, and Train partition consists the list of images in train.txt file.

In total, we obtain 500 images in Train partition and 250 images in Test partition.

Partition A's validation set contains the same number of images from each city (2 per city), and thus it should provide a better estimate of the model's accuracy, since partition B's validation set is more imbalanced domain-wise, with some cities being under-represented. Additionally, partition A has a larger train partition, and is thus expected to be able to generalize more and yield better results.

B. Centralized Baseline

In this section, we implemented a centralized baseline by using the BiSeNet V2 [1] model for the partition A and B that we built.

We used Cross Entropy loss as the loss function, SGD as the optimizer and we tried different combinations of hyper-parameters in order to find best possible model by computing the mean intersection over union (mIoU) as a validation metric.

The list of hyper-parameters that was set during the model implementation are listed as :

- Batch size : 8
- Learning Rate : 0.05
- Weight Decay : 5e-5
- Number of epochs : between 20-100 (although validation mIoU stopped increasing way before the 100th epoch)
- Resized height=512, width=1024

We found out that these hyper-parameters worked quite well for both partitions and decided to keep them.

The results that we obtained are visualized below with a random image from the validation split of Partition A, along

with the model's prediction. Follows, respectively: visualization of a validation image, its target label (ground truth), the model's prediction at different epochs.

Fig. 6. EPOCH 1, mIoU=4.4%

Fig. 7. EPOCH 26, mIoU=28%

Fig. 8. EPOCH 40, mIoU=41%

At epoch 42 mIoU was equal to 42%.

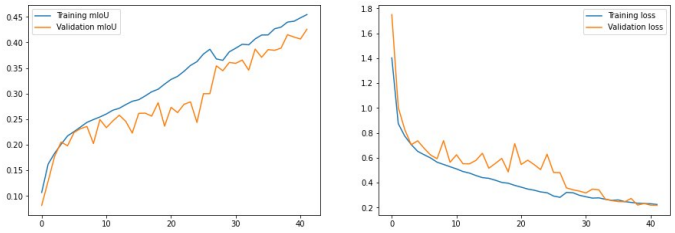


Fig. 9. Epochs on x axis, mIoU / loss on y axis

Let's now consider partition B. Follows, respectively: visualization of a validation image, its target label, the model's prediction at different epochs.

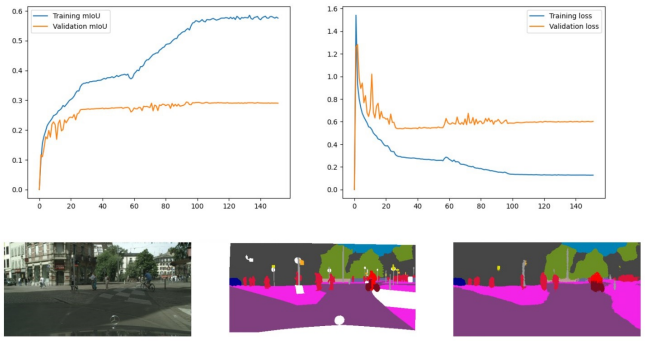


Fig. 10. Epochs on x axis, mIoU / loss on y axis

The highest mIoU, 27%, was obtained by using partition A. Therefore, we justified our assumption that partition A gives better result.

C. Federated Semantic Segmentation Experiments

The next experiment involved decentralizing the dataset on several clients in order compute a global model on the server, by averaging the updates sent by the clients. Alongside partition A and B, we tried two further partitions concerning domain imbalance. In the uniform partition, every client was assigned a random subset of the Cityscapes dataset, thus containing pictures from multiple cities. Given that real cars don't experience such an abrupt domain shift, we created a heterogeneous partition wherein every client is assigned images from one city only. Since federated averaging assumes a homogeneous data distribution, we can expect performance to drop considerably with heterogeneous partitions.

The chosen hyperparameters are:

- Batch size : 4
- Learning Rate : 0.05
- Weight Decay : 5e-5
- Number of epochs : 8
- Rounds: 8
- Clients selected: 10/tot
- Height = 512, Width = 1024

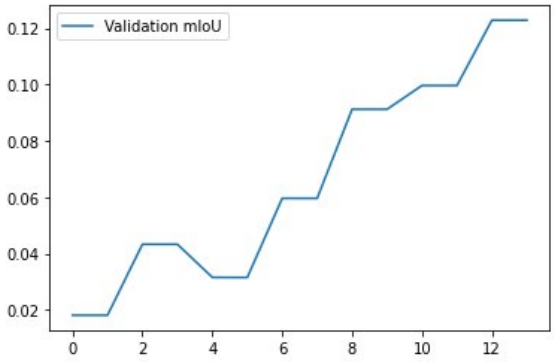


Fig. 11. Partition A uniform

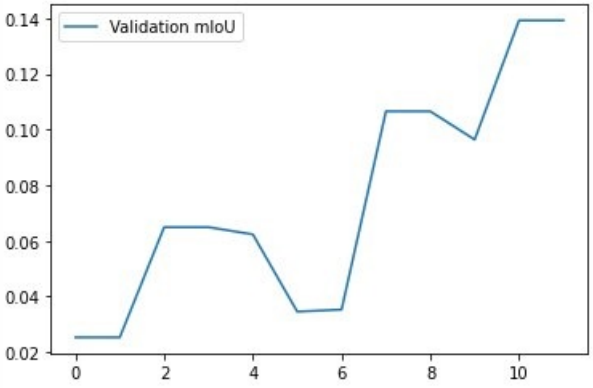


Fig. 12. Partition A heterogeneous

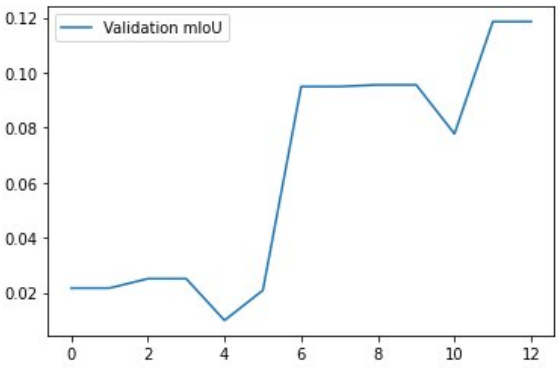


Fig. 13. Partition B uniform

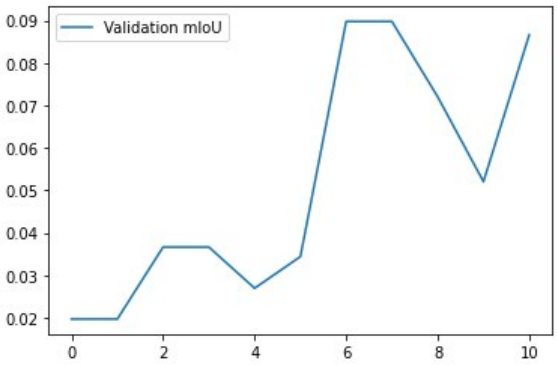


Fig. 14. Partition B heterogeneous

D. Moving towards source free domain adaptation

In this stage, a more practical scenario was taken into account. In reality, self-driving cars do not have access to accurate label information and manual labeling can be expensive. It is unrealistic to expect clients to have access to the labels for the images they gather. However, it is reasonable to assume that the model has already been trained on a publicly available data-set, such as GTA5. The following images show the predictions of the model trained on GTA5 and validated, respectively, on GTA5 and Cityscapes. Only the 19 labels in common with Cityscapes were selected. The performance with Cityscapes is, as predicted, considerably worse because of the domain shift between the two dataset.

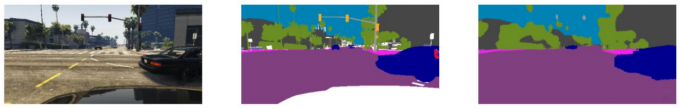


Fig. 15. model trained on GTA5 - partition A and validated on GTA5 after 45 EPOCHS val mIoU = 0.13

As it is show in the previous predictions, the model that is trained on GTA5 images achieves a good training mIoU, while being incapable of generalizing to the unseen domain

Fig. 16. model trained on GTA5 - partition A and validated on Cityscapes 45 EPOCHS

of Cityscapes, thys yielding a much lower validation mIoU. This was expected because of the domain imbalance between the two datasets.

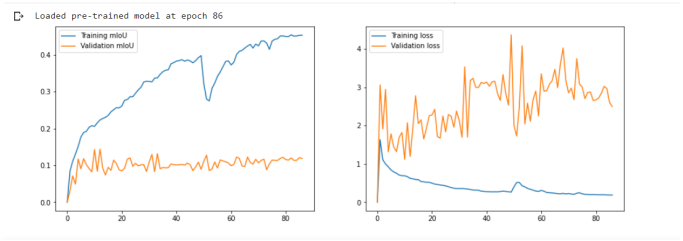


Fig. 17. Epochs on x axis, mIoU / loss on y axis, model trained on GTA5 - partition A and validated on Cityscapes

We observed that increasing number of epochs does not affect on mIoU score on validation set.

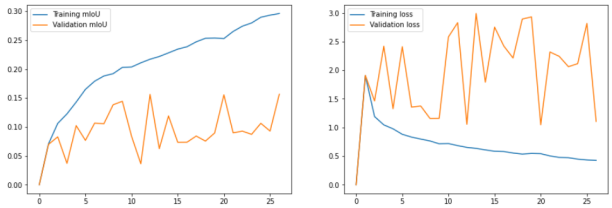


Fig. 18. Epochs on x axis, mIoU / loss on y axis, model trained on GTA5 - partition B and validated on Cityscapes

In order to reduce the effect of domain imbalance, FDA was used to transfer to style of Cityscapes images to GTA5. A bunch of target styles was created out of each city in Cityscapes, and the model was subsequently trained on GTA5 images transformed with a style chosen randomly from the list. We chose a window size L=0.01 because it manages to transfer the style without distorting the source image too much. In this image we can appreciate the transformation that FDA provides to a GTA5 image:

Fig. 19. Window size L=0.01

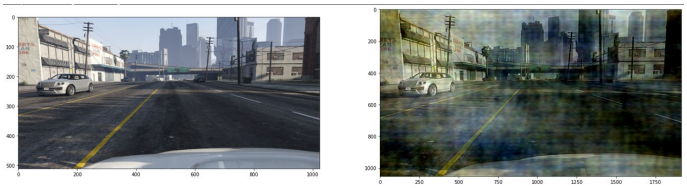


Fig. 20. Window size L=0.1

The window size in FDA refers to the number of data points used to calculate the Fourier coefficients to describe the frequency components of a signal.

A larger window size provides a better frequency resolution. Therefore finer details of the signal can be captured but it is computationally expensive.

On the other hand, a smaller window size reduces the computational cost, but it may lead to lower frequency resolution and more noise in the analysis. The choice of window size depends on the problem, such as the frequency resolution that is desired, the amount of available computational resources, and the level of noise in the data.

In our case, different values of window sizes was tried on images and window size 0.01 was not distorting the image. therefore we took 0.01 as a reference.

After transforming every image in GTA5 with the bunch of target styles, the performance improves on both partition A and B:

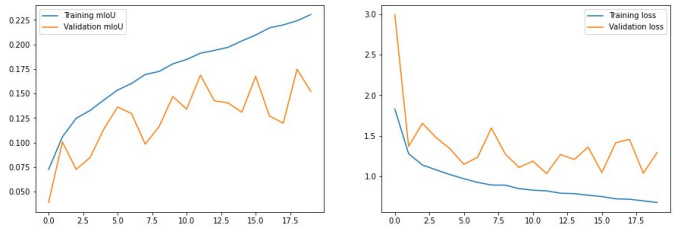


Fig. 21. Epochs on x axis, mIoU / loss on y axis, model trained on GTA5 with FDA-transformed images, partition A

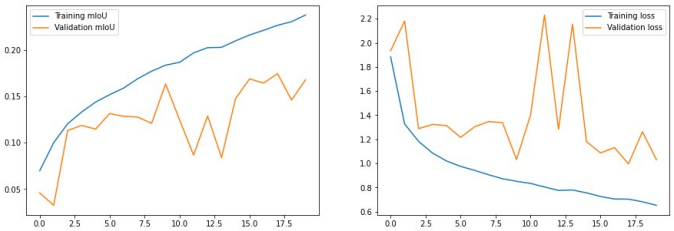


Fig. 22. Epochs on x axis, mIoU / loss on y axis, model trained on GTA5 with FDA-transformed images, partition B

E. Federated Self-Training Using Pseudo-Labels

In this section, we employed Pseudo-labels to explore how we can handle the challenge of obtaining labeled images for semantic segmentation in real-world scenarios. Pseudo-labels act as a self-training technique in an unsupervised or semi-supervised manner to enhance performance.

Fig. 23. Example of prediction

The chosen teacher models are trained on the GTA5 dataset, at first with no domain adaptation technique, and secondly with FDA.

We experimented with different update frequencies of the teacher model, and found out that, with a small number of rounds, not updating the teacher model yields the best results. Due to limitations in gear and GPU usage we couldn't experiment with a higher number of rounds.

Plots with the mIoU change over the rounds follow:

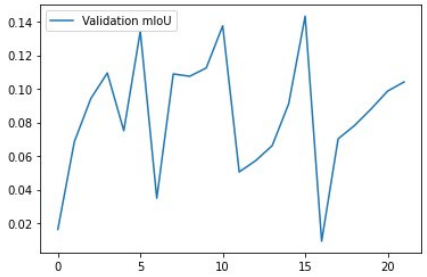


Fig. 24. no FDA partition A - heterogeneous

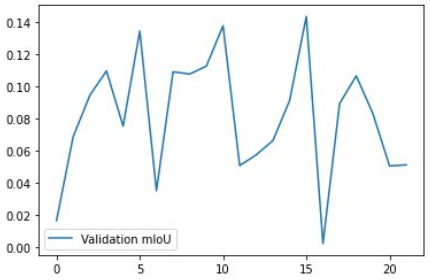


Fig. 25. no FDA partition B - heterogeneous

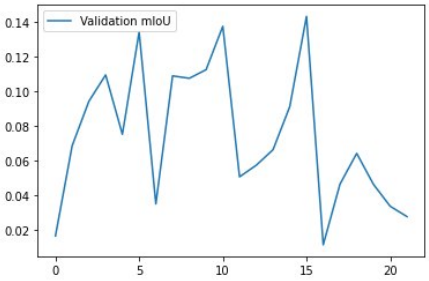


Fig. 26. no FDA partition A - uniform

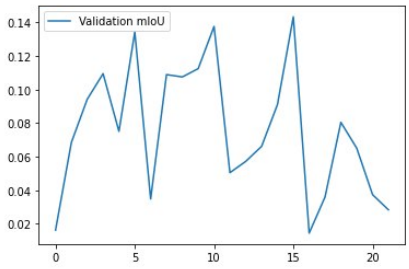


Fig. 27. no FDA partition B - uniform

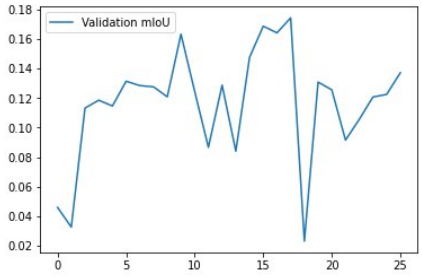


Fig. 28. FDA partition A - heterogeneous

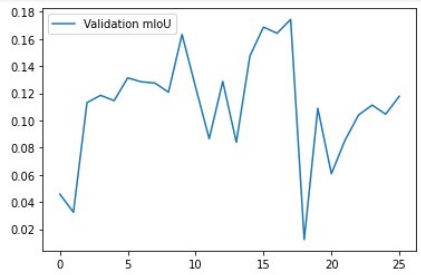


Fig. 29. FDA partition B - heterogeneous

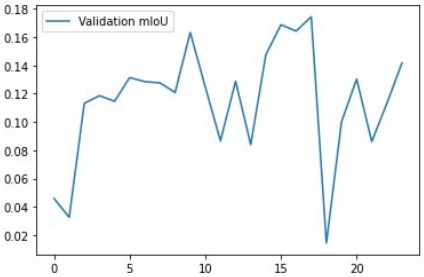


Fig. 30. FDA partition A - uniform

We observed that using the federated paradigm with Fourier domain adaptation leads to improved results compared to models without FDA applied when using Pseudo-Labels. The best result we achieved without FDA is around 14%. On the other hand the best result that we achieved by FDA applied is around 17%.

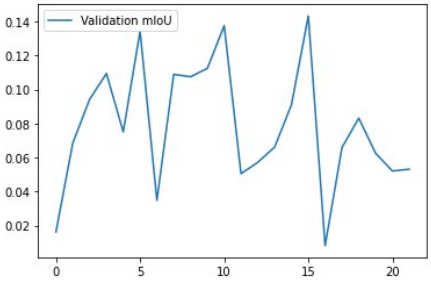


Fig. 31. FDA partition B - uniform

Fig. 32. Prediction example, model pre-trained with FDA, mIoU=17%

F. Consideration of MobileNetV2 as an Optional Step

In the field of self-driving cars, semantic segmentation is a crucial task with severe speed constraints. Given the real-time nature of the problem, it makes sense to consider some light-weight models such as MobileNetv2 [13], which can achieve high speed without sacrificing accuracy. In this sense, we re-tested both a centralized and a federated approaches by using a pretrained version of this network with a DeeplabV3 [14] head that is a deep learning model for semantic image segmentation, meaning it can identify and classify objects in an image and assign each pixel in the image to a particular class, and we achieved comparable results to BiSeNet in only a few epochs of further training.

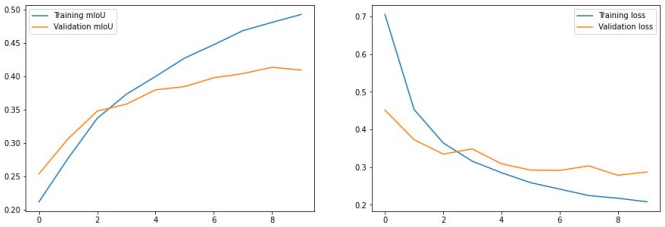


Fig. 33. mobilenetV2 scores implemented in step 2: Centralized Baseline

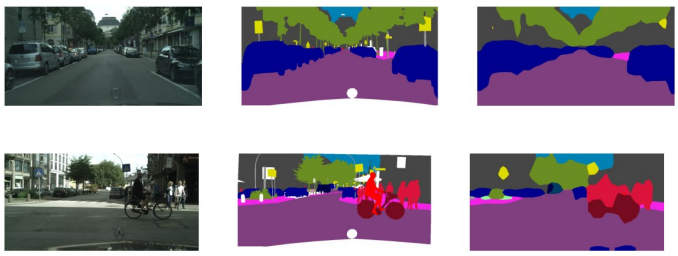


Fig. 34. best predictions val-mIoU = %41 after 10 EPOCHS obtained by mobilenetV2, step 2: Centralized Baseline

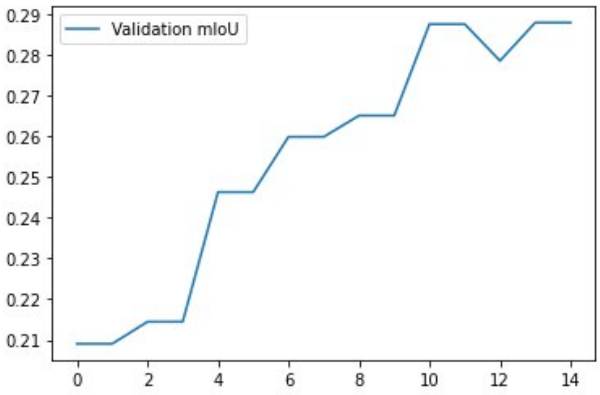


Fig. 35. mobilenetV2 scores implemented in step 3:Federated Semantic Segmentation Experiments

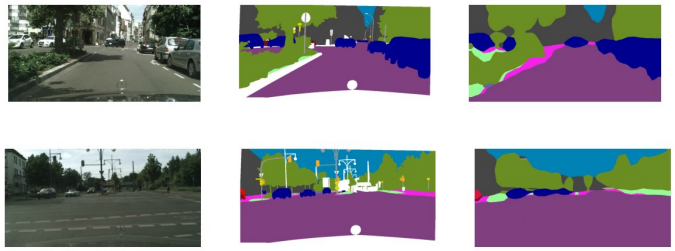


Fig. 36. best predictions val-mIoU = %28 after 14 ROUNDS obtained by mobilenetV2,step 3:Federated Semantic Segmentation Experiments

V. Conclusion

Throughout the project, various methods were utilized for semantic segmentation in self-driving cars by utilizing both the Cityscapes and GTA5 datasets. The BiSeNetV2 semantic segmentation model was implemented and its performance was evaluated using the pixel-wise metric, mIoU. Furthermore, federated and domain adaptation approaches were also utilized. Additionally, a light-weight network (Mobilenetv2) was also taken into consideration to observe any changes in the scores.

The experimentations in this paper started in ideal conditions, with a centralized approach on a manually annotated dataset, and then moved to more real scenarios, such as pseudo-label generation in a federated setting. We conclude by reporting the mIoU values obtained in the previous steps with the validation set.

- **Step2 (Centralized baseline):** mIoU=42% on partition A, mIoU=30% on partition B
- **Step3 (Federated setting):** mIoU=14% on partition A heterogeneous split
- **Step4 (Fourier domain adaptation):** mIoU=17% with FDA on partition A
- **Step5 (Federated setting with pseudo-labels):** mIoU=17% with FDA-trained teacher on partition A uniform split
- **Step6 (MobileNetv2):** mIoU=41% centralized, mIoU=28% federated

REFERENCES

[1] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "Bisenet V2: bilateral network with guided aggregation for real-time semantic segmentation," CoRR, vol. abs/2004.02147, 2020.

[2] Shijie Hao, Yuan Zhou and Yanrong Guo, "A Brief Survey on Semantic Segmentation with Deep Learning", Neurocomputing, vol. 406, pp. 302-321, September 2020.

[3] H. Zhang, J. Bosch, and H. H. Olsson, "End-to-end federated learning for autonomous driving vehicles," in IJCNN, 2021.

[4] Hao, Shijie, Yuan Zhou, and Yanrong Guo. "A brief survey on semantic segmentation with deep learning." Neurocomputing 406 (2020): 302-321.

[5] Fantauzzo, Lidia, et al. "FedDrive: generalizing federated learning to semantic segmentation in autonomous driving." 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2022.

[6] McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." Artificial intelligence and statistics. PMLR, 2017.

[7] Cordts, Marius, et al. "The cityscapes dataset for semantic urban scene understanding." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[8] Richter, Stephan R., et al. "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.

[9] Zhao, Sicheng, et al. "A review of single-source deep unsupervised visual domain adaptation." IEEE Transactions on Neural Networks and Learning Systems 33.2 (2020): 473-493.

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

[11] Shenaj, Donald, et al. "Learning across domains and devices: Style-driven source-free domain adaptation in clustered federated learning." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2023.

[12] Li, Yunsheng, Lu Yuan, and Nuno Vasconcelos. "Bidirectional learning for domain adaptation of semantic segmentation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[13] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[14] Yurtkulu, Salih Can, Yusuf Hüseyin Şahin, and Gozde Unal. "Semantic segmentation with extended DeepLabv3 architecture." 2019 27th Signal Processing and Communications Applications Conference (SIU). IEEE, 2019.