Federated Learning and Semantic Segmentation for Autonomous Driving

Riccardo Zanchetta
Politecnico di Torino
Turin, Italy
s313344@studenti.polito.it

Valerio Mastrianni
Politecnico di Torino
Turin, Italy
s308781@studenti.polito.it

Lal Akin
Politecnico di Torino
Turin, Italy
s300192@studenti.polito.it

Abstract

In this project we present an analysis of various scenarios of Semantic Segmentation and Federated Learning implementations for autonomous driving. A centralized baseline with the IDDA Dataset is created to serve as a stepping stone for the following steps, such as the Federated Learning scenario with decentralized devices. Moreover, we explore the application of a novel approach, Federated source-Free Domain Adaptation (FFreeDA), testing it with different parameters to gain deeper insights. Our evaluations include challenges such as handling unlabeled client data, working with previously unseen domains, and data heterogeneity. Finally, an Ensemble Learning method is created as our implementation. By delving deeper into these challenges, we aim to improve our understanding of their effectiveness in autonomous driving scenarios while keeping the privacy risks in mind. The source code is made available at https://github.com/zari19/MLDL_Project

1. Introduction

For state-of-the-art applications of autonomous, self-driving cars, Semantic Segmentation (SS) [5] is the foundation for allowing the cars to grasp their surrounding environment. This algorithm uses client data as input, that is in most of the cases private in nature. The dependence on private client data in this algorithm poses significant privacy concerns when sharing real-life images with a global model. Moreover, integrating data captured on the edge by remote devices, such as cars, into a global model creates challenges. To address these issues, the implementation of Federated Learning [7] was proposed.

This approach enables training models without sharing client data, therefore preserving privacy. By adopting a decentralized learning framework, Federated Learning allows remote devices to collaboratively train a global model while keeping data local.

The proposed combination of Semantic Segmentation and Federated Learning not only alleviates privacy risks but also facilitates the integration of edge devices' data. Thus, the utilization of Federated Learning in Semantic Segmentation is promising in the field of autonomous, self-driving cars. Having said this, Federated Learning and Semantic Segmentation have some challenges to tackle such as domain shifts, statistical heterogeneity and unlabeled client data.

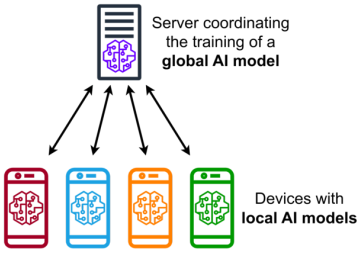

Figure 1. Federated Learning setting where edge devices share their individual parameters with the server

The steps that will be explained in this paper are:

1. Centralized Baseline and Data Augmentation
2. Supervised Federated Learning experiments with same and different domain data-sets
3. Domain Adaptation task, pre-training phase
4. Federated Self-training using Pseudo-Labels
5. YOLOv8 Ensemble Learning

2. Related Works

In this section, we review and discuss research studies. We present an overview of the methodologies, architectures, data-sets, and evaluation metrics that will be used in this project.

Federated Learning approach [7], was proposed by Google to handle the private nature of specific data such

as faces of individuals or private information in scenarios in which the data is needed as input for inference.

FL works on a decentralized setting where there are many edge devices, also called clients, e.g. telephones or cars, who collect data with different parameters. There is a global model that handles the orchestration, with whom the clients only share the individual weights and model updates. Since the client data is not shared with the global model the privacy risks are mitigated.

The parameters of the client data can be the amount of images/client or the size of the images taken, which intuitively introduces statistical heterogeneity. In this setting, the global model's distribution is not representative of the client's distribution which causes discrepancies.

Semantic Segmentation works by giving a class label to each of the image pixels therefore presenting a high-level understanding of the image. SS [5] aims to segment an image into meaningful and visually coherent regions.

Figure 2. Example of an IDDA Dataset image vs. Semantically Segmented classes after inference of the same image, different classes are visible in different colors

The general working principle of SS is as follows; usually Convolutional Deep Neural Networks are chosen to perform the segmentation task. Different backbone architectures such as ResNet [6], which is a Deep Learning model that leverages "Residual Functions" to overcome the vanishing gradient problem of the general CNN structure, are used. Also MobileNetv2 backbone is used in scenarios in which the speed of the model is a key factor, such as real-time segmentation. The chosen model is then trained on a data-set of labeled images, where each pixel is annotated with the corresponding class. The training process involves adjusting the model's parameters (weights and biases) to minimize the error between predicted labels and ground truth labels.

The most popular metrics used in Semantic Segmentation to calculate the error between prediction labels and the ground truth labels are Accuracy, F1-Score and Mean Intersection-over-Union (MIoU).

Pixel Accuracy , since SS works on a pixel-level, is the proportion of pixels correctly classified. In some scenarios this is not the best metric because class imbalance is a considerable issue with image classification, and in that scenario the accuracy may not reflect the ability of segmentation.

F1-Score is calculated as two times the area of overlap of pixels classified divided by the total pixels combined.

Mean Intersection-over-Union (mIoU) is the most commonly used because of its suitability and simplicity. It works by calculating how well a prediction label or bounding box fits the ground truth label.

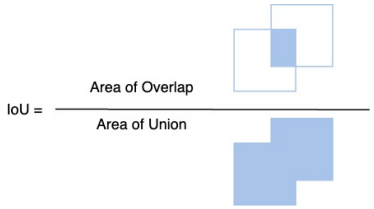

Figure 3. Intersection-over-Union

The combined implementation of FL and SS has promising potential in addressing the challenges faced by autonomous driving systems. By utilizing Federated Learning, it is possible to train SS models using data from decentralized sources, such as different vehicles.

Researchers have been exploring various FL algorithms, but there is few specifically tailored for SS tasks, and a novel algorithm has been proposed.

FedDrive [4] is an approach designed specifically with the aim of enhancing the accuracy and robustness of SS for autonomous vehicles. FedDrive compares two datasets,one of synthetic images and one natural, IDDA [1] and Cityscapes [3] and three different settings. Critical challenges for computer vision are tackled in this research, such as domain shift and statistical heterogeneity.

When a model only sees images of one domain it will learn to classify that one, but will have issues when facing a different domain. This domain adaptation problem comes up again when the clients' data have very different parameters, such as one client having images of a sunny day and the other a rainy one. Also in a real life scenario client data is generally not labeled. Hand-labeling said data is a very costly process, therefore unlabeled data has to be used as input.

Many researchers [4] [16] [10] [14] [15] delved into solving Domain Adaptation challenge, proposing numerous ideas as to how a Semantic Segmentation model can be made more robust when faced with new domains. Even further, in a Federated Learning framework domain adaptation has been found [16] to hold much significance because the non-iid nature of client data fuels the situation more.

For this purpose **Deep Unsupervised Domain Adaptation** [16] was proposed.

Unsupervised Domain Adaptation (UDA) is seen as the foundation of tackling the issue of SS models requiring pixel-level annotated data. The problem associated with UDA in a Federated Learning Scenario is that the source datasets are mostly private in nature, hence they cannot be shared. **Source-Free Domain Adaptation Framework** was proposed [10] in which a dual attention distillation method is created to capture and transfer semantic knowledge.

Another method proposed for overcoming unlabeled client data challenge is **FFreeDA** [12] in which a server-side labeled dataset is used for the pre-training step of the model. The local training in the client-side are made by only their own data.

Lastly, in **Bidirectional Learning** [8] for Domain Adaptation, two modules such as image-to-image translation model and the segmentation adaptation model were used. Since both of these models work towards improving each other, the domain gap gradually reduces. On the forward direction a Self-Supervised Learning (SSL) approach was implemented that uses **Pseudo-labels**. [8] [12]

There are two models; the teacher model and the student model. Intuitively, the dataset will be put through a forward pass on the teacher model and the results will "teach" the student model.

Pseudo-labels can be obtained from the prediction probability of the specific target label. After obtaining them, the corresponding pixels can be mapped onto the segmentation labels and this can also provide the model with segmentation loss.

The working principle is making one forward pass through the teacher model to get the pseudo-labels. The teacher model outputs an array of confidence probabilities for each pixel.Then by normalizing the predictions, the model outputs a labelled image for each unlabelled image input. The pseudo-labels are used as the ground truth labels for the student model. The highest class probability of the pixel predictions is only accepted if it is above a certain threshold. If none of the predictions are above the threshold they are not taken into account.

3. Methodology

In this section, the methodology used in the project to address the objectives and obtain relevant results is discussed.

3.1. Datasets

All of the experiments are implemented not exclusively, but mainly on IDDA Dataset, which is a large-scale multi-domain dataset with 1080 x 1920 pixel frames taken from the virtual world simulator CARLA. In this dataset, there are multiple domains such as different town settings, weather conditions, viewpoints. There are 24 semantic classes and in this project 16 of them are used.

Classes of IDDA that are included in our experiments: Building, Fence, Pedestrian, Pole, Road, Sidewalk, Vegetation, Vehicle, Wall, Traffic Sign, Traffic Light, Bicycle, Motorcycle, Rider, Terrain, Sky

Two test sets were used, "Same Domain" and "Different Domain", to have an understanding of it's effects on MIoU.

Alongside IDDA, **GTA5 Dataset**is also used for the unlabeled client data task. GTA5 is a synthetic dataset that contains pixel-level annotations, and it was rendered from the game GTA5. It contains many of the same semantic classes that are of IDDA, and for the purpose of this project only 16 matching classes are considered.

For Step 5, **Cityscapes Dataset** is used with FDA application to also include a natural dataset in our research since the original task is autonomous driving, and the real-life applications will be on natural images. Cityscapes is a urban street view dataset with 30 classes with different scenarios such as different seasons.

3.2. Backbone Architecture

As the architecture for the centralized and federated settings, a DeepLabV3 is implemented. DeepLabV3 is a state-of-the-art architecture that achieves high accuracy in pixel-level segmentation tasks.

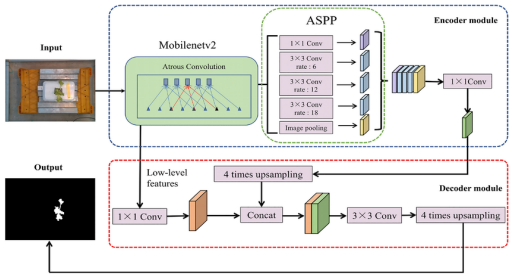

Figure 4. DeepLabV3 with MobileNetV2 Backbone [15]

As a backbone for this architecture, MobileNetV2 is used. It is a lightweight CNN model designed for efficient classification applications. It was developed to provide a good trade-off between accuracy, model size and the computational complexity. This backbone model is suitable for real-time applications such as autonomous driving, since it is more lightweight compared to other CNN models. The architecture of MobileNetV2 consists of inverted residual blocks that help to diminish the effects of vanishing gradient, and helps leverage residual functions.

3.3. Algorithms

Various predefined algorithms were implemented in this project that will be explained in this section.

Stochastic Gradient Descent (SGD) works by calculating the gradient for one data-point picked randomly at each iteration, instead of calculating the gradient for the entire dataset. As the algorithm descends through the loss value, an optima is reached.

Adam (Adaptive Moment Estimation) is a one-step-further version of SGD. It uses estimations of the first and second moments of the gradient to adapt the learning rate for each weight of the neural network.

FedAvg is an optimization method used in FL to tackle the challenges of training ML models on decentralized data. In this approach, the training process is done on multiple rounds between the global model on a server and edge devices, whilst iteratively updating the model weights by an averaging method.

- w_t model weights on communication round *t*
- w_t^k model weights on round *t* on client *k*
- η learning rate
- P_k set of data points on client *k*
- n_k number of data points on client *k*
- $f(w)$ is Loss $l(\mathrm{x}_i, \mathrm{y}_i : w)$, loss on example $\mathrm{x}_i, \mathrm{y}_i$ with model parameters w

$$\mathrm{F}_k(w) = \frac{1}{\mathrm{n}_k} \sum_{i \in \mathrm{P}_k} \mathrm{f}_i(w) \tag{1}$$

$$\mathrm{g}_k = \nabla \mathrm{F}_k(\mathrm{w}_t) \tag{2}$$

$$\forall k, \mathrm{w}_{t+1}^k \longleftarrow \mathrm{w}_t - \eta \mathrm{g}_k \tag{3}$$

$$\mathrm{w}_{t+1} \longleftarrow \sum_{k=1}^{K} \frac{\mathrm{n}_k}{n} \mathrm{w}_{t+1}^k \tag{4}$$

In Equation 1 loss of each step is averaged by the number of data points of the client. Equation 2 shows the gradient of this function of weights being calculated. In Equation 3, by subtracting the multiplication of learning rate and the gradient of the client from the current weight, the new iteration weight is found. By Equation 4 a weighted average of the client model weight is taken.

3.3.1 FDA

Fourier Domain Adaptation (FDA) [14] is a method that uses the Fourier Transform, specifically Fast Fourier Transform (FFT) to align the statistical properties of source and target datasets, making possible the transfer of knowledge from a labeled source domain to an unlabeled target domain.

Source and Target Domains: The goal of FDA is to transfer the knowledge of the source domain to the target domain.

Fourier Transform: The knowledge extracted in the previous step is transformed into the Fourier Domain that represents the features as a merge of frequencies and orientations.

Statistical Alignment: The aim is to align the statistical properties of the domains distributions in the Fourier Domain.

Figure 5. FDA Source and Target images, and the result of the algorithm [14]

3.3.2 LADD

LADD [12] is an approach specifically developed to solve FFreeDA, which considers that some clients in a federated setting may have similar distributions which will then be clustered based on how similar their styles are.

Server Pre-Training: FDA is implemented to extract the client's style, then through the server, this style is applied to the target in some given intervals.

Style-Based Clustering: Local training is done with a cluster specific teacher model that allows knowledge distillation. This process then outputs the pseudo-labels.

Server-Side Aggregation: The server carries out the distinguishing of the global and cluster specific parameters, and aggregation is done.

4. Experiments

All of the following experiments are run on Google Colab, using a commercial subscription. To extract as much knowledge as possible, the number and size of experiments are at the border of availability of resources.

4.1. Centralized Baseline

To have a benchmark to the FL-setting-based steps of the project, a centralized baseline was created. Different data augmentation methods are tested with their combinations and according to the results, the best performing ones are implemented.

In Figure 6 it is seen that some objects are much more easily segmented by the model, which is usually objects such as the sky or road that are large in proportion, and occur to our knowledge most of the images from IDDA.

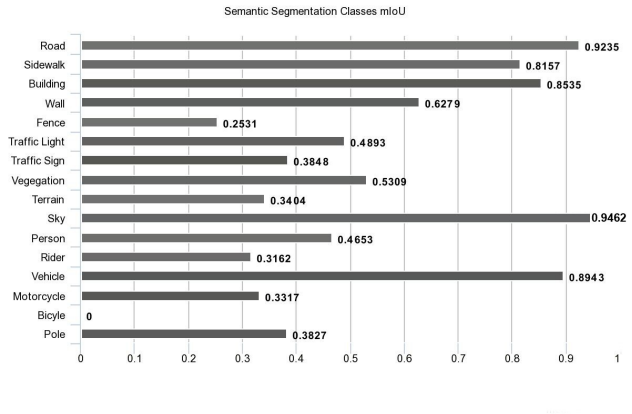

Figure 6. Classes of IDDA and the corresponding mIoU values from the Centralized Baseline

For the simplicity of demonstration, the subset of IDDA used for training is named "TR", the test set containing images of the same domain as the training set is named "TS1" and the test set containing images of different domain is named "TS2".

Learning Rate *(lr)* is a hyper-parameter that controls how fast the algorithm learns a certain parameter. Different values for this hyperparameter were implemented in Table 1 to inspect how it affects the performance. It can be seen that the best values were found with the highest value of *(lr)*.

In coherence with our findings, it was observed in [13], that most of the transformations added more noise than the model can handle and had lower mIoU score. When used together, the data augmentation methods as seen in Table 1, had lower mIoU values compared to the singular data augmentation values. With respect to this, *RandomCrop()* is used for the following steps.

4.2. Supervised Federated Learning

After concluding which is the best performing setting in the previous step, a Federated Learning model is created and tested.

In each different value of clients per round, it is seen from Table 5 that the best mIoU values are obtained with the highest number of rounds. Aside from the mIoU metric, the results of the runs with highest number of rounds are the ones who are the most consistent and robust when faced with a different domain test set. It is also evident that when there are a lower number of clients, the model performs slightly better, around 0.02 to 0.01 increase in mIoU. This signals a trade-off between higher number of clients and the performance of the model.

To demonstrate how the *learning rate scheduler* effects the performance, the tests resulting in Table 3 were implemented. LRS1 (learning rate scheduler 1) works by setting a lambda value, and working iteratively through choosing the learning rate. However, the LRS2 (learning rate scheduler 2) that we implemented works by setting a range for the learning rate, then calculating the decay factor according to the current epoch and the total number of epochs. In this way we customized the learning rate which is an important hyper-parameter as it was seen in previous experiments.

4.3. FFreeDA Pre-Training Phase

A more realistic scenario in computer vision is having unlabeled data, that is usually too time consuming to label by hand. Also in an autonomous driving scenario, this is not a feasible solution. By using two different datasets in this step with different domains, we start delving into Domain Adaptation and transferring knowledge between the two domains by FFreeDA [12].

To be able to match the classes of GTA5 to IDDA, the semantic classes are mapped with the corresponding ones from each dataset.

After training the model on GTA5 and evaluating it with the training set of IDDA, FDA [14] technique is applied with varying window sizes (β) to extract the styles from each client and to make a bank of styles. The source images of GTA5 are swapped with a random image from the style bank. As it can be seen from Table 4, when β = 0.001, the best performance is achieved. This information is implemented to the following step, as FDA window size.

4.4. Federated Self-Training using Pseudo-Labels

In this step, Pseudo-Labels [8] [12] are used in order to predict labels for the unlabeled data. There are the two models; the teacher model and the student model.

The key hyper-parameter in this step is the interval of the teacher model update, T. It can assume different values; never, or in the beginning of each T rounds for different values of T given that T is greater than 1. We selected the most confident predictions of our model setting a threshold of 0.9. Our results are presented in Table 5. The highest result is obtained when the update interval is set to 2 and there are 8 clients per round.

5. YOLOv8 Ensemble Learning

As the final step of this project, we decided to tackle one of the most challenging aspect of machine learning and federated learning: domain shift as previously discussed.
We decided to train our model on IDDA dataset in a classical FL scenario as in Step 2, and create an Ensemble Learning using YOLO [11], testing the result on a portion of Cityscapes Dataset [2] modified with FDA, B = 0.001, since it was the best performing windows size in our step 3 experiment.

YOLO is a real-time object detection model that is different from other models because it only sees an image

	Learning Rate			*Transformations*			
	0.01	0.05	0.1	Baseline (lr=0.01)	Random Crop (lr=0.01)	Random H.Flip	RandomCrop+ Rand.H.Flip
mIoU TR	0.621	0.627	0.643	0.663	0.621	0.605	0.587
mIoU TS1	0.559	0.572	0.558	0.583	0.559	0.528	0.497
mIoU TS2	0.438	0.474	0.440	0.476	0.438	0.412	0.475

Table 1. Results obtained from the Centralized Baseline with Random Crop transformation to test learning rate, each after 50 epochs, with m = 0.9, wd=0.0005.

Clients /rounds	rounds	local epochs	mIoU TR	mIoU TS1	mIoU TS2
2	48	1	0.360	0.357	0.283
	16	3	0.346	0.331	0.264
	8	6	0.314	0.288	0.214
4	24	1	0.342	0.334	0.264
	8	3	0.344	0.270	0.238
	4	6	0.307	0.278	0.280
8	15	8	**0.622**	**0.532**	**0415**
	12	1	0.323	0.325	0.324
	4	3	0.301	0.284	0.280
12	8	1	0.310	0.309	0.309

Table 2. Federated setting results obtained with lr = 0.01, m = 0.9, wd = 0.0005

	LRS1	LRS2
mIoU	**0.425**	0.353
mean acc.	0.468	0.406

Table 3. Federated setting with number of rounds = 20, clients = 8 and local epochs = 2; lr = [0.0001,0.1], m= 0.9, wd = 0.0005, tested with TS1

β	*mIoU TR*	*mIoU GTA5*	*mIoU TS1*	*mIoU TS2*
0.0005	0.262	0.247	0.267	0.160
0.005	0.286	0.266	0.288	0.179
0.001	0.311	0.284	0.308	0.203

Table 4. Results of Pre-training phase on GTA5, lr = 0.01, wd = 0.0005, m = 0.9. β parameter is the size of the FDA window.

once. YOLO has been trained with COCO Dataset [9], so we were able to use some common classes with IDDA and test it on CityScapes to improve the quality of the prediction in certain common classes such as: pedestrians, vehicles, motorcycle and bicycle which are critical in the autonomous driving task due to their importance in terms of security.

Clients /rounds	*local epochs*	*T*	*mIoU TR*	*mIoU TS1*	*mIoU TS2*
2	1	∞	0.278	0.270	0.264
8	1		0.291	0.287	0.273
2	1	1	0.288	0.286	0.272
8	1		0.299	0.294	0.284
2	1	2	0.289	0.281	0.285
8	1		0.309	0.302	0.290

Table 5. Results from 30 rounds of training, with a pretrained model on GTA5 from the previous step, lr = 0.01, m = 0.9, wd = 0.0005, T is the interval for the teacher model update.

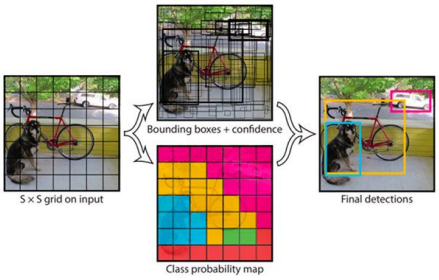

Figure 7. Working Principle of YOLO

5.1. Implementation

Main difference in implementation is that instead of having one model, DeepLabv3, we have two models that work side by side, both in the Federated scenario, separate from each other. YOLOv8 makes its predictions for the individual pixel class predictions, and the confidence values associated with the predicted class labels before test time of DeepLabv3. This also enables us to ensure safety, since in case of a model error or crash, YOLOv8 model is still up and running.

As the output of YOLOv8 model we have two matrices, the predicted class values from 1 to 15, and the corresponding probabilities as floating points from 0 to 1.

In order to achieve that we trained our model with DeepLabv3 on the MobileNetv2 architecture, as in Step 2, and we let YOLO make it's prediction on the images. As the result we had 2 different images with 2 different probability

Table 6. Difference in prediction between DeepLabv3 and DeepLabv3 + YOLO tested on CityScapes Dataset

	OverallAcc	MeanAcc	MeanPrecision	mIoU	person IoU	vehicle IoU	motorcycle IoU	bicycle IoU
DeepLabv3	0.740	0.329	0.489	0.250	0.116	0.326	0.013	0.0
DeepLabv3 + YOLO	0.773	0.411	0.576	0.333	0.475	0.451	0.285	0.388

masks associated with the model's predictions as in Figure 8. For the sake of simplicity we will address these matrices as: $P_{deepLab}$, P_{YOLO}, $C_{deepLab}$, C_{YOLO}, F, respectively: Probability of prediction of DeepLab and YOLO and their confidences and the resultant matrix.
In order to get a final prediction we used an algorithm as described below:

Algorithm 1 Pseudo code ensemble prediction

for each pixel in P_{deepLab} and P_{YOLO} **do**
  if $C_{\text{deepLab}} > \tau$ **and** $C_{\text{YOLO}} > \tau$ **then**
    if $|C_{\text{deepLab}} - C_{\text{YOLO}}| < \Delta$ **then**
      if $P_{\text{deepLab}} = P_{\text{YOLO}}$ **then**
        $F = P_{\text{deepLab}}$
      else
        apply Nearest Neighbour with radius r^*
      end if
    else
      if $P_{\text{deepLab}} < P_{\text{YOLO}}$ **then**
        $F = P_{\text{YOLO}}$
      else
        apply Nearest Neighbour with radius r^*
      end if
    end if
  else if $P_{\text{deepLab}} < P_{\text{YOLO}}$ **then**
    $F = P_{\text{YOLO}}$
  else
    $F = P_{\text{deepLab}}$
  end if
end for
return F

* The Nearest Neighbour is implemented to find the most probable value around the given pixel using the inverse euclidean distance in order to weight the probabilities in the $C_{deepLab}$ and C_{YOLO} matrices.

5.2. Results

The results of the proposed YOLOv8-based Ensemble learning approach were exceptionally promising. Integrating YOLO alongside Semantic Segmentation through DeepLabV3, we obtained a substantial improvement in the overall performance. When evaluated on CityScapes dataset our Ensemble Learning model demonstrated enhanced predictive capabilities, particularly in handling classes such as pedestrians, vehicles, motorcycles and bicycles, that are vital elements for the safety and precision of autonomous driving systems. As shown in Table 6, there is a +0.36 gain in Person IoU, +0.28 in motorcycle IoU and +0.388 in bicycle IoU. These results underline the potential of YOLOv8 in the field of autonomous driving.

6. Conclusion

In this paper, a comprehensive research on Federated Learning for different Semantic Segmentation tasks were explored with an Ensemble Learning model proposed to create a one-step-forward approach. First a centralized baseline was done in order to have a comparison for when the clients are decentralized and in a Federated setting.

Some issues that we faced were unlabeled client side data and different domains creating a domain shift, which were then remedied by the FFreeDA algorithm and FDA approach to create a Domain Adaptation model. Additionally, in the Ensemble Learning method that was proposed with YOLOv8 and Deeplabv3, it was seen that some classes of great importance for the autonomous self-driving cars, such as pedestrians and bicycles, were segmented with a much higher performance. This statement can also be applied to a broader sense, and we can say it was seen that the usage of the Ensemble Method has proved beneficial in overall performance of the model.

References

[1] Emanuele Alberti, Antonio Tavera, Carlo Masone, and Barbara Caputo. IDDA: A large-scale multi-domain dataset for autonomous driving. *IEEE Robotics and Automation Letters*, 5(4):5526–5533, oct 2020. 2

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016. 5

[3] Marius Cordts, Mohamed Omran3 Sebastian Ramos, and Timo Rehfeld. The cityscapes dataset for semantic urban scene understanding. pages 3213–3223. 2

[4] Lidia Fantauzzo, Eros Fani', Debora Caldarola, Antonio Tavera, Fabio Cermelli, Marco Ciccone, and Barbara Caputo. Feddrive: Generalizing federated learning to semantic segmentation in autonomous driving, 2022. 2

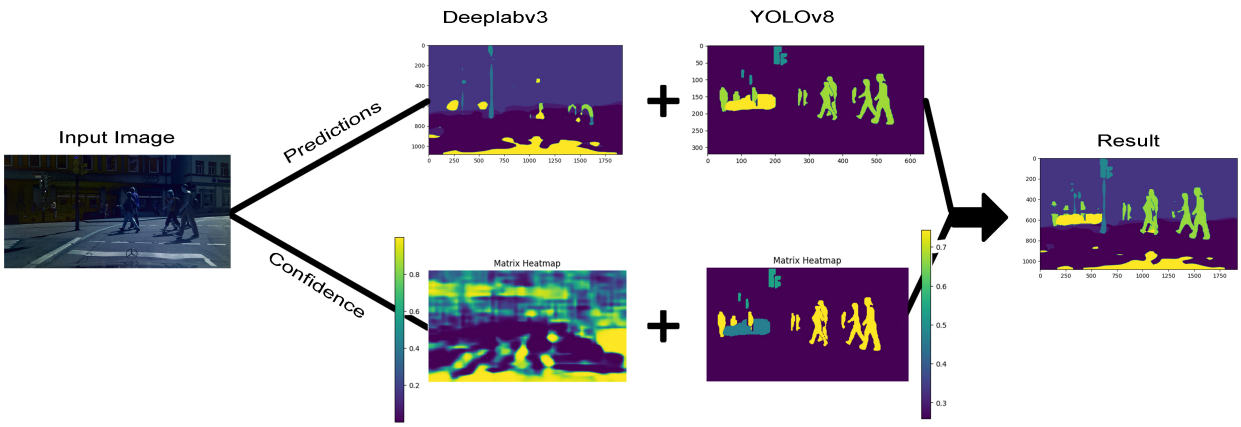

Figure 8. Working pipeline of Step 5. The input CityScapes image on the left, the output of the Ensemble model on the right

[5] Shijie Hao, Yuan Zhou, and Yanrong Guo. A brief survey on semantic segmentation with deep learning. 406:302–321, 2020. 1, 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 2

[7] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, may 2020. 1

[8] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. pages 6936–6945. 3, 5

[9] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 6

[10] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation, 2021. 2, 3

[11] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015. 5

[12] Donald Shenaj, Eros Fanì, Marco Toldo, Debora Caldarola, Antonio Tavera, Umberto Michieli, Marco Ciccone, Pietro Zanuttigh, and Barbara Caputo. Learning across domains and devices: Style-driven source-free domain adaptation in clustered federated learning, 2022. 3, 4, 5

[13] Jerry Tang, Manasi Sharma, and Ruohan Zhang. Explaining the effect of data augmentation on image classification tasks. 5

[14] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation, 2020. 2, 4, 5

[15] Yu Zhang, Mengliu Wu, Jinsong Li, Si Yang, Lihua Zheng, Xinliang Liu, and Minjuan Wang. Automatic non-destructive multiple lettuce traits prediction based on deeplabv3 +. *Journal of Food Measurement and Characterization*, 17, 10 2022. 2, 3

[16] Sicheng Zhao, Xiangyu Yue, Shanghang Zhang, Bo Li, Han Zhao, Bichen Wu, Ravi Krishna, Joseph E. Gonzalez, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia, and Kurt Keutzer. A review of single-source deep unsupervised visual domain adaptation, 2020. 2, 3