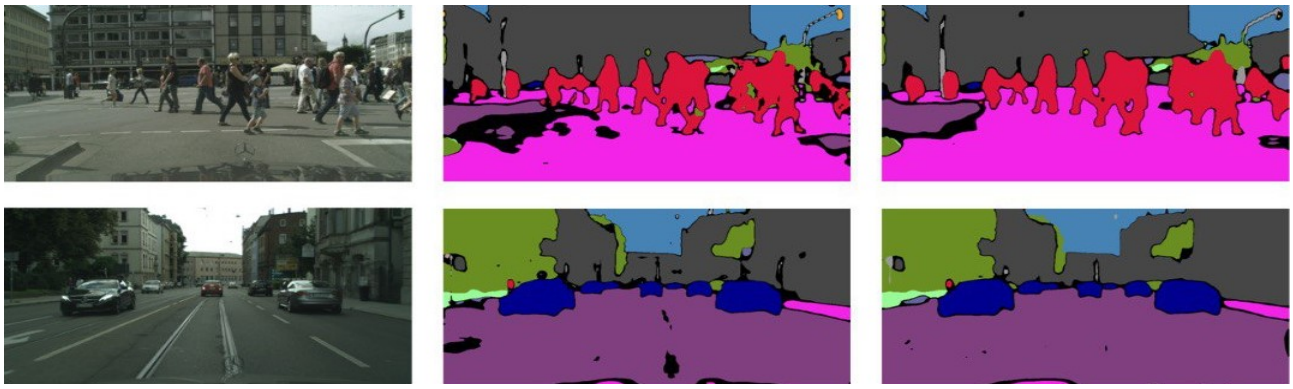
Meta Pseudo Labels for Real-Time Semantic Segmentation

Pietro Montresori
Politecnico di Torino
Student ID: s296440
s296440@studenti.polito.it

Micol Rosini
Politecnico di Torino
Student ID: s302935
s302935@studenti.polito.it

Chiara Vanderputten
Politecnico di Torino
Student ID: s306273
s306273@studenti.polito.it



Preview figure. Comparison between the real image, pseudo labels, and *meta* pseudo labels.

***Abstract*—Due to their ability to learn hierarchical representations of the image data, Convolutional Neural Networks (CNNs) have become a popular method for solving the task of pixel-wise semantic segmentation. Since the state-of-the-art models rely on a large amount of annotated samples and synthetic data are significantly cheaper to obtain than real data, it is not surprising that Unsupervised Domain Adaptation reached a broad success within the semantic segmentation field. In this paper, we describe our work on real-time domain adaptation for semantic segmentation. We have combined adversarial training, showing also a light version of the discriminator, with a self-supervised technique that allows us to create pseudo labels. To further improve the predictions of the adapted model we also implemented an algorithm to create Meta Pseudo Labels (MPL), that are constantly updated with a teacher-student learning technique. The project repository is available at the following link: https://github.com/micolrosini/Real-Time-Domain-Adaptation-in-Image-Segmentation**

I. Introduction

Semantic segmentation is a crucial task in computer vision, enabling the cluster of each pixel of the image together which belongs to the same object class. This task has a wide array of applications ranging from scene understanding and inferring support relationships among objects to autonomous driving. Nevertheless, in applications that require low-latency operations, the computational cost of these methods is still quite limiting. Autonomous driving, for example, needs to make decisions in precise intervals. Therefore, it is necessary to improve the design of segmentation models towards achieving efficient architectures that can perform in real-time with the appropriate precision [1].

Furthermore, the algorithms that allow this kind of results require a huge amount of labeled data, which is not available in practical scenarios. To avoid this issue some works proposed using a modified version of the video games software to produce a large number of high-quality road scenarios [2].

Unfortunately, the use of labeled images taken from video games has one major drawback: the *domain shift* between the synthetic data and the real-world images. To close the gap between these two datasets, we propose an unsupervised domain adaptation strategy based on adversarial learning, which goal is to match the overall feature-level distributions of the two different domains [3]. In order to reduce training time and improve the predictions, we also implemented a light version of the discriminator using depth-wise separable convolutions [4]. We have adopted BiSeNet [5] as the backbone, a Bilateral Segmentation Network which is light and computationally fast.

In this paper our main contributions are:

- The creation of pseudo-labels with an unsupervised learning technique: from unlabeled images the first model (*the teacher*) generates pseudo labels, which are then combined with labeled images to retrain the second model (*the student*) [3] [6].
- An improvement of the previous method, the creation of meta pseudo-labels: we design a systematic mechanism for the teacher to correct the bias in its pseudo labels by observing the performance of the student model on the pseudo-labeled dataset [7], and using it to improve the teacher's creation of pseudo labels.

II. Related work

Semantic Segmentation: This task is usually performed by deep convolutional neural networks [8]. The work introduced by *Long et al.* [9] has shown that is possible to build fully convolutional networks that take an image and return an equal size output. However, the trained model may not generalize well to unseen image domains. To train these advanced networks there is an urgent need for thousands of pixel-level annotations in order to match the model capacity of deep CNNs. To overcome this problem, weak and semi-supervised approaches have been proposed in past years. Different projects have tried to overcome the slowness of manual labeling with the implementation of synthetic datasets based on rendering, such as SYNTHIA [1] or GTA5 [2]. The integration of synthetic datasets with real images is performed in order to enhance the learning capability of the model. However, if there is a discrepancy in the distribution between the synthetic and real images, it may result in an issue known as *domain-shift*.

Domain Adaptation: To solve the domain-shift problem, different domain adaptation models have been implemented. One of the most common techniques is *adversarial training*, which works by means of two agents: the generator and the discriminator. The first one will try to align the distributions of the source (synthetic data) and the target (real-time data) in order to "fool" the second one, whose task is to recognize if the image is from the source or the target. Many works have been implemented following this trend, like *Tsai et al.* [10], that have incorporated adversarial learning at different feature levels of the segmentation model, or *Li et al.* [3], that use bidirectional knowledge to learn alternatively the image translation model and the segmentation model.

Pseudo-labeling: Pseudo labeling is an unsupervised learning technique used in image segmentation, where the model generates with domain adaptation technique its own labels for the unlabeled dataset and then trains the model also with these pseudo-labels. This approach leve rages the model's confidence in its predictions to improve its performance on unseen data. The aim is to increase the amount of labeled data available for training while still maintaining high-quality annotations. Pseudo labeling was first proposed by *Lee* in 2013 [6], it starts by training a model on a batch of labeled data, then it uses the trained model to predict labels on a batch of unlabeled data. As a next step, *Pham et al.* presents Meta Pseudo Labels [7], a semi-supervised learning method that achieves a new state-of-the-art top-1 accuracy of 90.2% on ImageNet, which has a 1.6% improvement in performance. Meta Pseudo Labels has a teacher network to generate pseudo labels on unlabeled data to teach a student network, however, the teacher is constantly adapted by the feedback of the student's performance on the pseudo-labeled dataset. In their implementation, they augment the teacher's training with a supervised learning objective and a semi-supervised learning objective, additionally training the teacher on unlabeled data using the UDA objective.

III. Method

BiSeNet

The model used in this paper is BiSeNet [5] which is composed of two parts: **Spatial Path** (SP) and **Context Path** (CP). The first component is used to preserve the spatial information and generate high-resolution features, it contains three layers to obtain the 1/8 feature map. Each layer includes a convolution with stride = 2 and padding = 1, followed by batch normalization and ReLU. The Context Path utilizes a lightweight model [11], which can be ResNet-101 or ResNet-18, and can downsample the feature map fast to encode high-level semantic context information, then is added a global average pooling to provide a large receptive field with global context information. The up-sampled output feature of global pooling is then combined with the features of the lightweight model.

After that, since the information captured by the two paths is different in terms of feature representation, a Feature Fusion Module (FFM) fuses these features. First, the output features of the SP and the CP are concatenated. Then, batch normalization is used to balance the scales of the features. Next, the concatenated features are pooled into a feature vector, and a weight vector is computed. This weight vector can re-weight the features, which amounts to feature selection and combination.

Unsupervised Adversarial Network:

In order to perform domain adaptation between the source data (*GTA5*) and the target data (*Cityscapes* [12]), a discriminator D is used together with the segmentation network (the generator G) [10]. For the source domain, a segmentation loss is computed as the cross-entropy loss between the ground truth annotations Y_s for source images I_s and the segmentation output P_s:

$$\mathcal{L}_{seg}\left(I_s\right) = -\sum_{h,w}\sum_{c\in C} Y_s^{(h,w,c)} \log(P_s^{(h,w,c)}) \tag{1}$$

Where C is the set of all classes and h, w refers to a specific pixel of the image with dimensions $H \times W$.

For the target domain, in order to make the target prediction P_T of the model closer to the one of the source, a binary cross-entropy loss called adversarial loss $\mathcal{L}_{adv}$ is computed: it trains the segmentation network to fool the discriminator by maximizing the probability of the target prediction being considered as the source prediction:

$$\mathcal{L}_{adv}\left(I_t\right) = -\sum_{h,w} \log\left(\mathbf{D}\left(P_t\right)^{(h,w,1)}\right) \tag{2}$$

This loss is then multiplied by a coefficient λ_{adv}, and stored as the loss of the target. Both the segmentation loss of the source image and the loss of the target one are then back-propagated through the model.

The discriminator will get two predictions and will try to infer if the input is from the target or the source domain, it is trained with a binary cross-entropy loss $\mathcal{L}_d$ computed with the

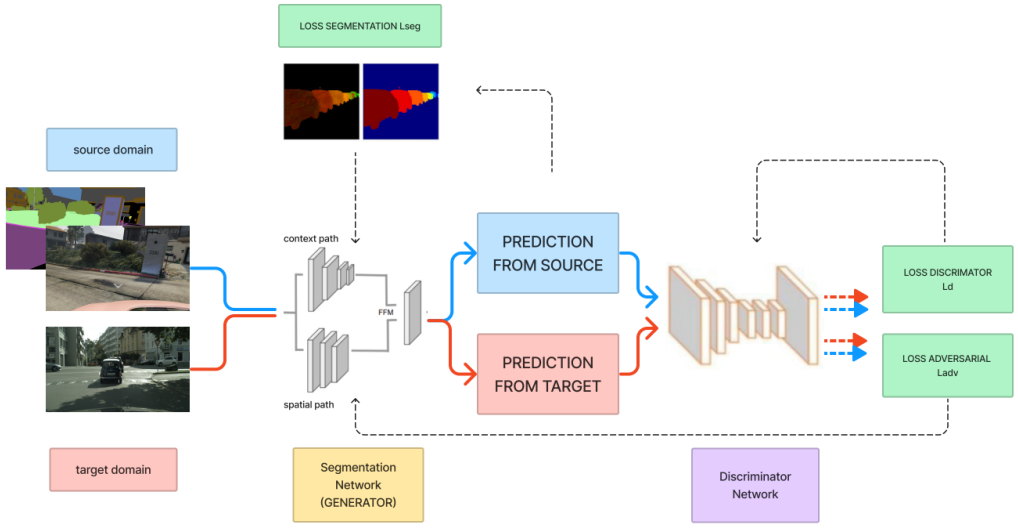


Fig. 1. Algorithmic overview for unsupervised adversarial domain adaptation. The source and target images are passed to the segmentation network to obtain predictions. Thanks to the prediction of the source, the generator calculates the segmentation loss that will be used by the generator itself to improve the predictions. The predictions are also passed through the discriminator which calculates the adversarial loss and the loss of the discriminator.

discriminator forecast $D(P)$ on the two classes (i.e., source and target):

$$\mathcal{L}_d(P) = -\sum_{h,w}(1-z)\log\left(\mathbf{D}(P)^{(h,w,0)}\right) + z\log\left(\mathbf{D}(P)^{(h,w,1)}\right) \tag{3}$$

If the sample belongs to the source $z = 1$ while if it belongs to the target $z = 0$. The joint loss for the adaptation task of the generator will be defined as:

$$\mathcal{L}\left(I_s, I_t, Y_s\right) = \mathcal{L}_{seg}\left(I_s, Y_s\right) + \lambda_{adv}\mathcal{L}_{adv}\left(I_t\right) \tag{4}$$

Lightweight Depthwise Separable Convolutions

Another discriminator architecture was proposed in which each convolution operation is replaced by a Depthwise Separable Convolution (DSC). The DSC consists of a spatial convolution performed independently over each channel of a tensor, and a Pointwise Convolution, a 1x1 convolution that combines information from all channels. This alternative architecture is reported in Fig. 2 [4]. With this simple modification, the author obtained a fast and lightweight discriminator that requires fewer parameters while still providing significant performance in terms of the number of Floating-Point Operations per Second (FLOPS).

Pseudo Labels

In order to improve the performance of the segmentation network, a self-supervised learning technique based on pseudo labels [3] has been developed following these steps:

1) **Initial training of the segmentation model**: the segmentation model is trained for 50 epochs until it is capable of producing confident annotations for the target domain.
2) **Generation of pseudo labels**: confident annotations for the target domain are produced by filtering pixels with high prediction confidence in the image using a *max probability threshold* (MPT). For each class, a different MPT is defined as the median of all the probabilities of pixels predicted to belong to that class. Pixels with a probability lower than the threshold are classified as background, while others are classified with their correct category. If the median is higher than 0.9, the threshold for the specified class is set at 0.9.
3) **Computation of the loss segmentation**: the loss segmentation is computed for the target domain using the generated pseudo labels, $\mathcal{L}_{seg}\left(I_t, \widehat{Y}_t\right)$, which is then incorporated into the target loss that will be back-propagated:

$$\mathcal{L}_{target} = \lambda_{adv}\mathcal{L}_{adv}\left(I_t\right) + \mathcal{L}_{seg}\left(I_t, \widehat{Y}_t\right) \tag{5}$$

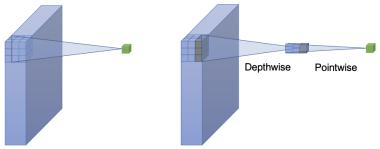


Fig. 2. Difference between standard and depthwise separable convolution

Meta Pseudo Labels

Pseudo labels lead to a bias since they strictly depend on the accuracy of the network. The systematic mechanism of

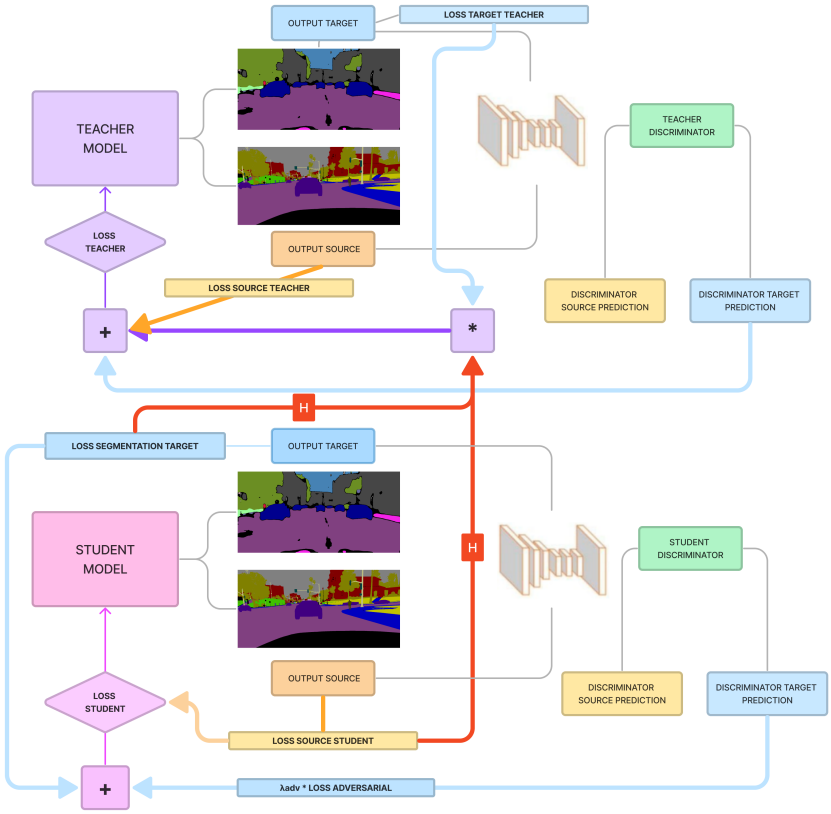


Fig. 3. Meta pseudo label technique

Meta pseudo labels (MPL) [7] can overcome this problem. It involves two actors:

- The *teacher* T, a pretrained network aimed at creating the pseudo labels.
- The *student* S, the segmentation model, which provides annotations for input images.

The MPL method leverages the performance of the student on the pseudo-labeled datasets to improve the quality of the pseudo-labels generated by the teacher. The teacher and student are trained in parallel:

- The student learns both from pseudo-labeled data from *Cityscapes* annotated by the teacher, and labeled images from *GTA5*.
- The teacher learns from the reward signal that indicates how well the student performs on the pseudo-labeled dataset.

Training process - Student

The student model is trained with pseudo-labels generated by the teacher using the same self-supervised learning technique described in the previous sections. The student model will have a loss function for both the source and target datasets. The source loss, $\mathcal{L}^S_{seg\_source}$, is calculated as the cross-entropy between the output predictions of the student model on the source images and their actual labels. The target loss, $\mathcal{L}^S_{target}$, is defined in Eq. 5. These two losses are then backpropagated to the student model. The discriminator is also trained as defined in Eq. 3. With each epoch, the student model trains with different pseudo-labels, which become increasingly precise.

Training process - Teacher

The teacher model is a pretrained network. Its loss is based on the performance of the student model on the teacher's pseudo labels. Specifically, the teacher's losses computed are: the segmentation losses with respect to the source label and the pseudo label using cross-entropy, respectively $\mathcal{L}^T_{seg\_source}$ and $\mathcal{L}^T_{seg\_target}$, and the adversarial loss $\mathcal{L}^T_{adv}$ calculated using binary cross entropy such as in Eq. 2. However, a unique coefficient, H, is also computed as part of the teacher's loss calculation:

$$H = \lambda^S \cdot \mathcal{L}^S_{seg\_target} \cdot \mathcal{L}^S_{seg\_source}$$

where λ^S is the student learning rate and $L^S_{seg\_target}$ and $L^S_{seg\_source}$ are respectively the segmentation loss of the student on the target domain and on the source domain. To improve the alignment of the two different domains a UDA loss is used additionally to train the teacher. This loss is computed by adding Gaussian noise to a portion of the target images. Finally, the complete teacher loss that will be backpropagated through the teacher model is:

$$L_{teacher} = \mathcal{L}^T_{seg\_source} + \mathcal{L}^T_{adv} + H \cdot \mathcal{L}^T_{seg\_target} + \mathcal{L}^T_{UDA}$$

The relationship between the student and teacher models is established through the use of the coefficient H. If the student model experiences large losses, the teacher model will make significant updates to its parameters. These losses will occur only if the teacher's pseudo-labels are not sufficiently accurate. Through this mutual relationship, both the teacher and student models will continually improve their performance.

IV. Experiments

Dataset

The proposed model has been trained and evaluated using two different subsets of the *Cytyscapes* and *GTA5* datasets.
From *GTA5* 500 synthetic images with pixel level semantic annotation are used to train the generator in the adversarial learning framework.
From *Cytyscapes* 500 unlabeled images to train it and 250 labeled images to evaluate it have been used. For both datasets only the 19 classes in common have been taken into account. Also, the *Cytyscapes* images have been resized to 1024×512, while the *GTA5* images have been reduced to 1280×720.

Evaluation metrics

The metrics that were used to evaluate the model are two:

- *precision per pixel*: the percentage of pixels in the image that is classified correctly.
- *mIoU*: the average over all the classes IoU score, that calculates the number of pixels common between the target and prediction mask divided by the total number of pixels across the two masks.

Implementation details

A. Hypertuning for Supervised Learning

First, both ResNet-18 and ResNet-101, pretrained on ImageNet, have been tested for the implementation of the BiseNet network. Table I shows that Resnet-101 achieved better results. This is due to the fact that this network uses more convolutions allowing the model to learn more complex features from the image and consequently to perform better for more complex classification tasks. In this Supervised Learning model, three different types of optimizers have been tested: SGD, Adam, and RMSprop. From Fig.4, the optimizer for the model that achieved the best results is the SGD with a batch size of 4 and a crop size of 1024×512 for the images, momentum 0.9, and weight decay 10^{-4}, the mIoU is = 0.53 and the precision achieved with this optimizer is 0.804. The initial learning rate for this model is $2.5 \cdot 10^{-2}$ [5].

B. Hypertuning of discriminator for Unsupervised learning

For the unsupervised domain adaptation technique two discriminators were tested: Fully Convolutional Discriminator (FCD) and Depthwise Separable Convolution Discriminator (DSC Discriminator). FCD is characterized by 5 convolutional layer with kernel size = 4x4, and a number of channels equals to $[64, 128, 256, 512, 1]$, stride = 2 and padding = 1. While in the DSC Discriminator, each convolution is replaced with a depthwise separable convolution, composed by a depthwise convolution with kernel size = 4×4 and a Pointwise Convolution with kernel size = 1×1. In this technique, each convolutional layer is followed by a Leaky ReLU with a negative slope of 0.2. The number of channels, the stride, and the padding has remained the same. In this experiment, both have been tested with the Adam optimizer. Fig. 5 shows that DSC performs better achieving an accuracy of 0.706. Moreover, it is evident from Table II that the DSC variant of the Discriminator is much lighter than the FCD, this implies a higher computational speed. Moreover, the three different optimizers were tested also on the DSC Discriminator. Fig. 6 shows that the best optimizer for the discriminator is Adam, with mIoU = 0.307 and precision = 0.706.

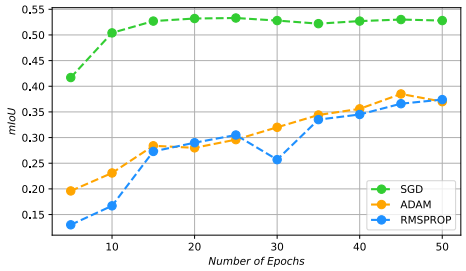


Fig. 4. mIoU for the supervised learning model with optimizer Adam, SGD, and RMSprop optimizers.

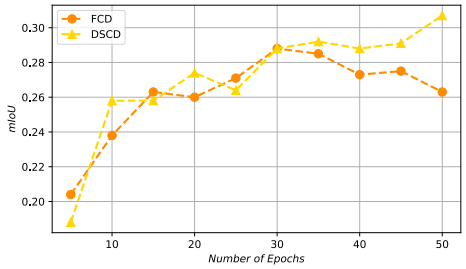


Fig. 5. mIoU for the unsupervised model trained with FCD and DSC Discriminator

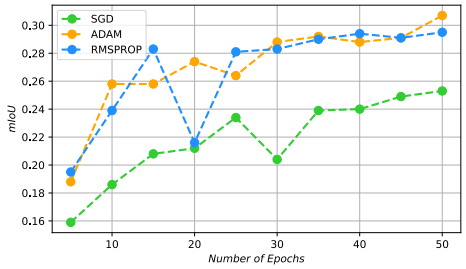


Fig. 6. mIoU for the model trained with unsupervised learning with the following discriminator's optimizer Adam, SGD and RMSprop optimizers

C. Different learning rates for pseudo-labels training technique

To find the best initial learning rate on the Unsupervised learning techniques that involve the use of pseudo labels, the

TABLE I
COMPARISON OF THE **RESULTS** OBTAINED

	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motocycle	bicycle	mIoU%	precision
ResNet-18	95.51	67.3	83.62	19.31	18.64	31.82	28.37	40.59	86.19	46.38	88.82	51.81	16.25	84.37	8.37	24.79	12.37	11.26	50.1	**45.6**	**71.5**
ResNet-101	96.4	74.49	86.19	33.51	27.26	38.16	42.37	54.42	87.81	49.2	89.95	60.7	33.9	88.24	26.83	42.1	22.39	26.55	58.67	**54.7**	**80.5**
FCD	70.32	22.71	68.94	23.15	17.65	8.17	0.048	0	76.96	32.07	64.45	33.16	2.21	73.35	27.19	24.39	0	2.62	0	**28.8**	**66.8**
DSC	80.45	31.64	75.32	19.81	9.38	23	9.08	4.84	79	35.84	68.19	35.96	8.43	70.26	13.90	13.73	0	4.11	0	**30.7**	**70.6**
DSC + PL	86.30	35.68	77.94	22.10	8.13	24.32	14.26	7.68	81.9	36.14	73.56	39.71	2.46	74.62	17.24	4.71	0	5.92	0	**32.2**	**73.4**
DSC + MPL	83.65	36.02	79.24	21.09	9.37	23.48	15.17	8.51	82.27	36.84	71.95	37.88	2.04	75.32	15.86	5.02	0	4.6	0	**32.5**	**74**

TABLE II
NUMBERS OF **FLOPS AND PARAMETERS** OF THE FULLY CONVOLUTIONAL DISCRIMINATOR AND OF THE LIGHTWEIGHT DISCRIMINATOR

Network	Total parameters	FLOPS [GFlops]
FCD	2781000	30.89
Discriminator DSC	189424	2.147

following values have been tested: $2.5 \cdot 10^{-2}$, $1.25 \cdot 10^{-2}$, $5 \cdot 10^{-2}$, $2.5 \cdot 10^{-4}$. Fig. 7 shows that initial learning rates that are large lead to decrease performances. The best initial learning rate is $1.25 \cdot 10^{-2}$ which achieves a mIou = 0.323 and precision = 0.738.

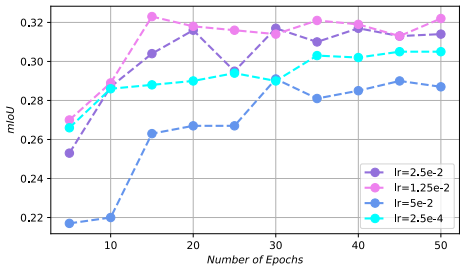


Fig. 7. mIoU for different initial learning rate values for the model trained with pseudo labels

D. Meta-pseudo-labels hyper tuning

The Meta Pseudo Labels training process is repeated for 50 epochs. The model optimizer is *SGD* and the discriminator optimizer is *Adam*. Various experiments have been tested for the meta-pseudo labels technique. Firstly, different initial learning rate values have been tested: $[2.5 \cdot 10^{-2}, 1.25 \cdot 10^{-2}, 2.5 \cdot 10^{-4}]$, without having any further improvement with respect to pseudo-labels unsupervised learning technique. Since the student was not pretrained, in the first epoch it could affect negatively the teacher's performance, because it is not able to predict correctly in the beginning. To overcome this problem a different test has been made: H is multiplied by a parameter $\alpha \in [0, 1]$ that will increase as the epochs go by. In this way, α gives more and more importance to H and it will make grow exponentially, so that the student's prediction will have no effect on the teacher in the initial period and then, as the student learns better and better it will have more impact. The best performance is obtained with this final method and it reaches a precision of 0.74 and a mIoU of 0.325. Fig.8 shows the results of this final MPL training method compared to the classical adversarial learning technique with DSC discriminator and the pseudo labels unsupervised learning technique, in this test both models are trained with a learning rate of $2.5 \cdot 10^{-2}$

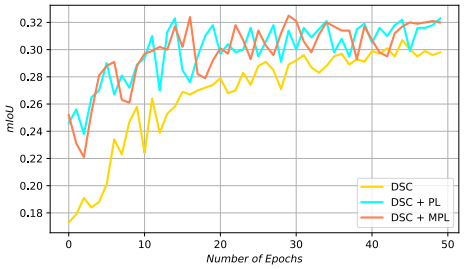


Fig. 8. mIoU for the model with discriminator DSC, the model with DSC + Pseudo labels,the model with DSC + Meta pseudo labels

V. CONCLUSION

In conclusion, pseudo labels have proven to be a promising approach for real time domain adaptation in image segmentation. By leveraging the ability of deep neural networks to generate high-quality predictions, pseudo labels allow us to use unlabeled data to improve the performance of image segmentation models. Our experiments demonstrate that this technique can achieve good results that are better than classical adversarial learning methods. This suggests that pseudo labels can be an effective tool for overcoming the challenge of acquiring large annotated datasets in the field of computer vision. Moreover, the use of a teacher-student technique in unsupervised learning tools proves to be very efficient as it manages to improve pseudo labels and make them more confident and precise, providing meta pseudo labels. However, since the method created for meta pseudo labels expects to use labeled and unlabeled images from the *same dataset* and not from two different ones, the results obtained are not as good as the one reported in the paper [7]. Possible future implementations could consider the idea of testing different type of UDA loss in order to better align the gap between the two different domains. Furthermore, results might improve if the student model for the training is pretreined with classical adversarial learning technique, in order to better help the teacher to create more accurate pseudolabels in a useful and meaningful way, since a 'more intelligent' student will lead to a 'more performing' professor.

REFERENCES

[1] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3234–3243, 2016.

[2] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European conference on computer vision*, pp. 102–118, Springer, 2016.

[3] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6936–6945, 2019.

[4] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

[5] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 325–341, 2018.

[6] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, p. 896, 2013.

[7] H. Pham, Z. Dai, Q. Xie, and Q. V. Le, "Meta pseudo labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11557–11568, 2021.

[8] S. Hao, Y. Zhou, and Y. Guo, "A brief survey on semantic segmentation with deep learning," *Neurocomputing*, vol. 406, pp. 302–321, 2020.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[10] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7472–7481, 2018.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.