DTFSeg: A Dynamic Threshold Filtering Method for Semi-Supervised Semantic Segmentation

1st Lingyan Ran
Ningbo Institute of Northwestern Polytechnical University
Xi'an, China
lran@nwpu.edu.cn

2nd Weiqi Zhan
School of Computer Science
Northwestern Polytechnical University
Xi'an, China
zhanweiwq@mail.nwpu.edu.cn

3rd Yali Li
School of Computer Science
Northwestern Polytechnical University
Xi'an, China
yarili@mail.nwpu.edu.cn

4th Xiaoqiang Zhang
School of Information Engineering
Southwest University of Science and Technology
Mianyang, China
xqzhang@swust.edu.cn

5th Shizhou Zhang
School of Computer Science
Northwestern Polytechnical University
Xi'an, China
szzhang@nwpu.edu.cn

***Abstract*—Recently, the research on semi-supervised semantic segmentation has made rapid progress, where a large number of unlabeled images with pseudo labels are adopted for boosting performance. Despite their achievement, how to get high-quality pseudo labels still remain challenging. Most methods would use complexly designed threshold strategies for pseudo tag generation. In this article, we propose a semi-supervised semantic segmentation method based on simple threshold filtering and self-training. In the process of generating pseudo-labels, the method deals with the thresholds of different categories of image pixels separately. It filters the labels of each category of pixels by dynamically changing thresholds to guide the model to train. This method is a general strategy and can be combined with the existing semi-supervised semantic segmentation methods based on generating pseudo-labels. We fully demonstrate its effectiveness on the Cityscapes dataset and UAVid dataset.**

***Index Terms*—semi-supervised learning, semantic segmentation, dynamic thresholding**

I. Introduction

Semantic segmentation is a vital computer vision topic, which can be applied in different fields, such as autonomous driving [1], [2], robotic grasping prediction [3], [4], dynamic SLAM [5], [6], etc. The severe shortage of labelled data and the huge labour assumption for acquiring pixel-wise labels make the task challenging. Semi-supervised learning (SSL) methods aim to make good predictions using a limited amount of labelling data and a large amount of unlabeled data, which makes it suitable for training semantic segmentation models.

There are various ways to use affluent unlabeled data with SSL. For example, consistent regularization and self-training methods are commonly used for semi-supervised semantic segmentation, where strong/weak augmentation is added to the training procedure. French et al. [7] verified that mask-based image-level strong augmentations, like CutOut [8] and CutMix [9], can be helpful to the semantic segmentation task. CCT [10] introduces feature-level perturbations and enforces consistency between the predictions of different decoders. GCT [11] performs network perturbations by using two differently initialized segmentation models and encourages consistency in their predictions. Others pay their attention to loss functions, DMT [12] re-weights the loss on different regions based on the disagreement of two different initialized models. Moreover, pseudo-labels with self-training methods are more efficient. CPS [13] applies two models with the same architecture but different initialization to create pseudo labels for each other to conduct cross-pseudo. However, the generation of pseudo labels is not easy.

Usually, the pseudo labels are of low quality and how to select confident labels becomes challenging. PseudoSeg [14] adopts grad-CAM based on image-level labels to enhance the quality of pseudo labels. ST++ [15] further proposes to separate high-confidence and low-confidence pseudo-labels for phased retraining, giving priority to high-confidence samples to generate better pseudo labels.

The more efficient way would be to set a threshold for high confident scores. FixMatch [16] is an image classification method based on pseudo-labels, in the process of pseudo-label generation, when the image classification confidence is greater than a fixed threshold, the loss calculation of this image is performed. In semantic segmentation, some methods use simple threshold filtering in the process of generating pseudo-labels. CAC [17] uses a fixed threshold to generate the pseudo-label. U2PL [18] uses a fixed entropy value as a filtering criterion for each pixel's prediction result. DST-CBC [19] linearly increases the proportion of pixels in the pseudo-label during training and determines the threshold by the proportion and the overall confidence distribution of a particular class of pixels. Similarly, DGCL [20] adopts the entropy of each pixel's prediction result as the standard, and linearly increases the proportion of pixels in the pseudo-label with the training, too. However, the above threshold filtering approach does not take into account the fact that model learning is a tendency to go faster and then slower. Dash

2023 China Automation Congress (CAC) | 979-8-3503-0375-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/CAC59555.2023.10451534

[21] is a semi-supervised image classification method based on simple FixMatch [16] framework, which considers the pattern of model's learning process and achieves very good results in combination with a unique threshold filtering process in which the threshold is decreased based on an exponential function. However, the threshold filtering method in Dash is set for image classification tasks, uses the same threshold to filter all images, and the category imbalance in semantic segmentation is more obvious, so the confidence of each pixel varies greatly.

In this work, we propose a dynamic threshold filtering semantic segmentation method (DTFSeg) that takes into account the process by which the predictive power of a model changes over time. We adjust the threshold filtering method in Dash to adapt it to the semantic segmentation task, set a separate threshold for each category. As shown in Fig. 1, our key method is based on a fast and then slow threshold drop process based on an exponential function. After a warm-up phase of training using only labeled data, the initial value of the threshold is set based on the average initial confidence of the model's prediction results for each category. As the training progresses, because the model's predictive ability increases, the probability of correct pseudo-labels increases, so the filtering threshold is reduced. The initial threshold represents the reliability of each sample to a certain extent and also reflects the learning difficulty of different samples, after threshold filtering, we assume that the current threshold-filtered label is trusted. Therefore, the weighted unsupervised loss based on the initial threshold is added. Our main contributions are summarized as follows:

- The DTFSeg method is proposed, with an exponentially decreasing threshold based on the confidence level of the labeled data categories in the previous epoch before the unlabeled data is added.
- On the basis of the initial threshold, we set and add loss weights on the unsupervised loss, and focus more on the difficult-to-distinguish categories.
- Experiments on two public datasets, Cityscapes [22] and UAVid [23] dataset, demonstrate the effectiveness of the proposed method.

II. The Proposed Method

A. Problem Definition

Given a combination set of M pixel-wise labeled images $D^l = \{(x_l, y_l)\}_{l=1}^{M}$ and N unlabeled images $D^u = \{x_u\}_{u=1}^{N}$, with $N >> M$, the key to semi-supervised semantic segmentation is how to utilize a large number of unlabeled images to get a sufficient performance boost compared to models obtained by training with only a small number of labeled images.

B. Self-training Based Semantic Segmentation Framework

In semi-supervised segmentation, the framework based on self-training has been widely used, which usually consists of one teacher model and one student model. The teacher model is responsible for generating pseudo labels, while the student model learns from both ground-truth labels and pseudo labels. Pseudo-labels need to be generated in each iteration. Considering the j_{th} pixel on the i_{th} unlabeled image, x_{ij}^u, we define the model's prediction probability as p_{ij}^u, and its corresponding confidence as c_{ij}^u, the inference can be described as:

$$p_{ij}^u = f\left(x_{ij}^u\right), \tag{1}$$

$$c_{ij}^u = max\{p_{ij}^u\}, \tag{2}$$

with $f(\cdot)$ being the trained model.

Assuming there are K categories in the dataset, the class label of pixel x_{ij}^u can be predicted as:

$$\hat{y}_{ij}^u = \begin{cases} \underset{k}{argmax} p_{ij}^u, & c_{ij}^u > \tau, k \in K \\ ignore, & other \end{cases} \tag{3}$$

where τ is the confidence threshold, and the corresponding pixel is valid only if the probability of c_{ij} is greater than it, otherwise, it is ignored. The larger the c_{ij}^u, the larger the probability that the pixel will be predicted as the k-th category. After generating pseudo-labels, the training for the student model contains supervised loss L_l and unsupervised loss L_u. The total loss L is

$$\begin{aligned} L &= L_l + \mu L_u \\ &= -\frac{1}{MP_l}\sum_{i=1}^{M}\sum_{j=1}^{P_l}\sum_{k=1}^{K} y_{ij}^k log(p_{ij}^k) \\ &- \frac{\mu}{NP_u}\sum_{i=1}^{N}\sum_{j=1}^{P_u}\sum_{k=1}^{K} \hat{y}_{ij}^k log(p_{ij}^k), \end{aligned} \tag{4}$$

in which the P_l and P_u represent the number of pixel of each labeled image and unlabeled image, the L_l and L_u are the cross-entropy loss on labeled data and unlabeled data with pseudo labels, and μ is a loss weight to balance supervised and unsupervised loss.

In each iteration, the parameters of the student model can be updated with

$$\theta_{student}^t = \theta_{student}^{t-1} - lr \cdot \frac{\partial L}{\partial \theta_{student}^{t-1}}, \tag{5}$$

where lr is the learning rate.

After that, we update the teacher model with the parameters of the student model using the EMA (Exponential Moving Average) method [24], as

$$\theta_{teacher}^t = \xi\theta_{teacher}^{t-1} + (1-\xi)\,\theta_{student}^t, \tag{6}$$

with "ξ" being the weight that controls model updates.

C. Dynamic Threshold Filtering

Today, most of the existing threshold filtering methods do not consider the objective change trend of the model learning ability nor treat each category of pixel in the image separately. In our work, we use the pseudo-labels [16] method to complement our approach of dynamic threshold filtering as Fig. 1. Concretely, we adjusted the threshold filtering method in Dash [21] to adapt it to the semantic segmentation task, set a separate threshold for each category. In our approach, the threshold is calculated as follows,

$$\tau_k = C \cdot \gamma^{\left(1-\frac{cur\_iter}{total\_iter}\cdot T\right)} \cdot \tau_k^{init} \tag{7}$$

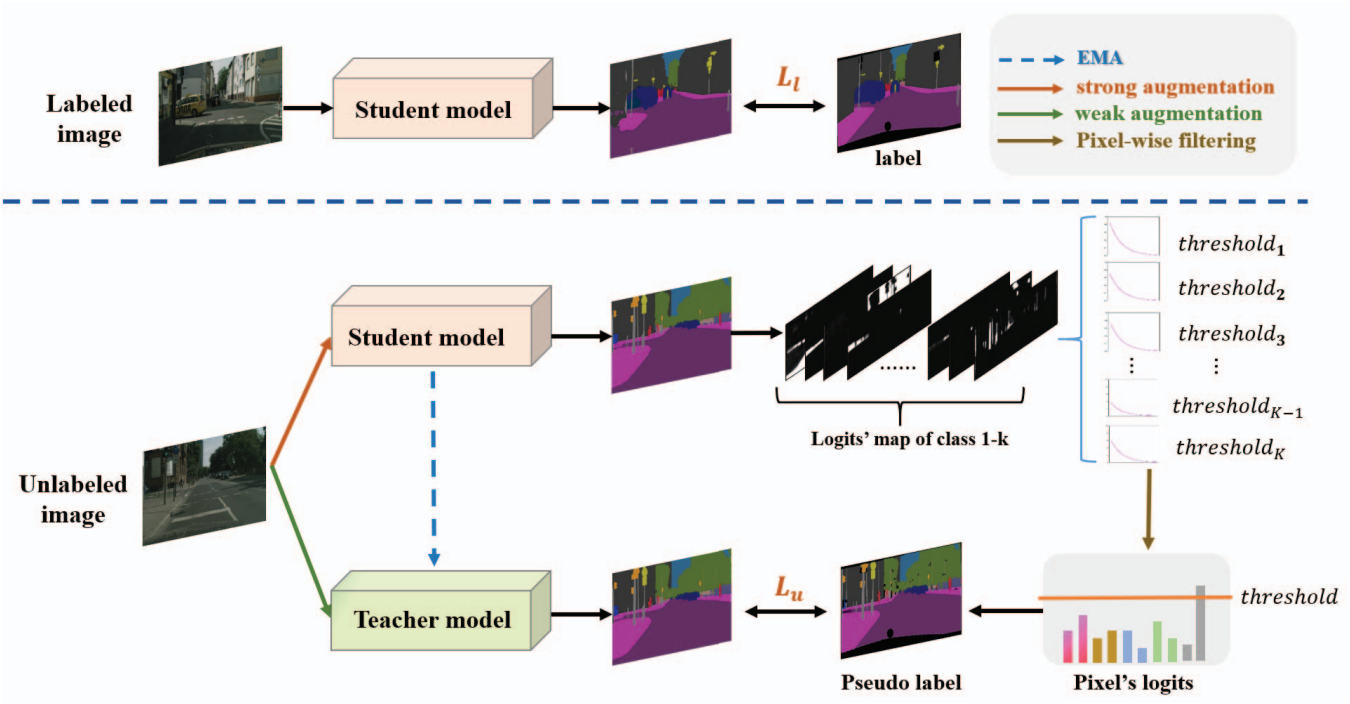

Fig. 1. The proposed DTFSeg framework. The upper part of the dotted line represents the preheating training using only labeled images, when some pixels in the image are predicted to be a certain category, we record the average confidence of these pixels in this category as the initial threshold of the current category. The bottom part represents the process of generating pseudo-labels after warm-up and filtering pseudo-labels based on the confidence threshold. During the training process, the threshold of each class decreases in the form of an exponential function of the training time.

the τ_k^{init} represents the initial threshold, and C is a constant used to deflate the size of the initial confidence level, the γ is the base used to control how fast or slow the exponential function declines, which usually takes a value between 1.0001 and 1.01, the "cur_iter" and "$total\_iter$" represent the current number of iterations and the total number of iterations that have been performed in training, respectively. Because the value of "$\frac{cur\_iter}{total\_iter}$ "is always less than 1, we multiply it by a hyper-parameter T to let the overall "$\left(1-\frac{cur\_iter}{total\_iter}\cdot T\right)$" value can be less than 0, compared to not adding T, the overall results have a more substantial decrease.

Different from the initial threshold setting approach in Dash [21], we set thresholds individually for each category and use the average value of the maximum confidence of pixels predicted by the model as the k-th category in the last epoch of label-only training as the initial confidence threshold for that category. As $\tau_k^{init}=\frac{\sum t_{ij}^l}{num(t_{ij}^l)}$, where $t_{ij}^l=maxp_{ij}^l$ and $argmaxp_{ij}^l=k$. Because of the difference in how the initial threshold is calculated, model training is divided into two stages. In the first stage, only labeled data is used, which not only warms up the model to make the pseudo-labels generated more accurate but also calculates the initial confidence of each category. The second stage uses both labeled and unlabeled data and uses our threshold filtering method in unlabeled data to obtain more accurate pseudo-labels.

D. Weighted Loss

In section II-C, the τ_k^{init} represents the initial confidence of the k category and, to some extent, the different confidence levels between each category, so τ_k^{init} can be used as a reference for setting loss weights. Given τ_k^{init}, the final w_k for category k of unsupervised loss can be described as

$$w_k=\left(\frac{\left(\sum_{k=1}^{K}\tau_k\right)/K}{\tau_k}\right)^{0.5} \tag{8}$$

Finally, the loss at the stage where only labeled data is used to warm up the model and calculate the initialization confidence is

$$L=L_l \tag{9}$$

After the stage where only labeled data is used to warm up the model and calculate the initial confidence, unlabeled data is added to the training. At this time, the overall weighted loss is

$$\begin{aligned} L&=L_l+\mu L_u \\ &=-\frac{1}{MP_l}\sum_{i=1}^{M}\sum_{j=1}^{P_l}\sum_{k=1}^{K}y_{ij}^k log(p_{ij}^k) \\ &-\frac{\mu}{NP_u}\sum_{i=1}^{N}\sum_{j=1}^{P_u}\sum_{k=1}^{K}w_k\hat{y}_{ij}^k log(p_{ij}^k) \end{aligned} \tag{10}$$

The joint usage of threshold filtering and weighted loss ensures that our DTFSeg model has good performance with limited labeled images.

III. Experiments

Firstly, the experimental settings used to evaluate our proposed approach are introduced. Then, we demonstrate our method on the Cityscapes and UAVid datasets at different labeled data scales and compare it to a supervised baseline and other SOTA methods. Next, we compare the proposed DFTSeg with other common threshold filtering methods under different proportions of division on the Cityscapes dataset.

A. Experimental Setup

Dataset. Our experiments and ablation studies are tested on two public datasets, Cityscapes [22] and UAVid [23].

Cityscapes contains 5,000 fine annotated images with 19 semantic classes of urban scenes, and those images are divided into training, validation, and testing sets which contain 2,975, 500, and 1,525 images, respectively. We train only with the training set and evaluate with the validation set. The original UAVid dataset contains footage taken from the perspective of a drone, divided into eight categories, dividing cars into moving cars and parked cars. For the purpose of image semantic segmentation, we combine the two categories of cars into one category. Besides, based on the original large resolution image, the length and width are cut to one-third of the length of the original image, resulting in a smaller resolution in order to facilitate model training. The processed UAVid dataset includes 3480 training images and 340 validation images.

For both datasets, we divide the whole training set into two groups via randomly subsampling 1/4, 1/8, and 1/30 of the whole training set as the labeled set and regard the remaining images as the unlabeled set, and evaluate with the validation set.

Evaluation. Our performance evaluation is based on single-scale testing and the mean of intersection over union (mIoU). We report the results of the Cityscapes [22] val set and the UAVid [23] val. We compare our results with recent reports in a fair manner. We use ResNet-101 [25] as our backbone networks. We load the ResNet-101 weights pre-trained on ImageNet [26]. In addition, we use DeepLabv3+ [27] as a segmentation head. We use mini-batch SGD with momentum to train our model with Sync-BN. For the Cityscapes dataset, we set the crop size as 800 × 800 and for the UAVid dataset, we set the crop size as 600 × 600. For both datasets, we adopt a learning policy with an initial learning rate of 0.01 which is then multiplied by $\left(1-\frac{iter}{max\_iter}\right)^{0.9}$, and set weight decay as 0.0005, and batch size as 16. We use random horizontal flip, random scale, and crop as our default data augmentation, and OHEM [28] loss is used on Cityscapes. We set the parameter γ in Eq. (7) to value 1.001, C to value 1.001 and T to value 1000.

B. Comparison with Current Methods

The comparison results with other state-of-the-art methods Cityscapes and UAVid datasets are presented in Table I and II, respectively.

Cityscapes. As shown in Table I, across a wide range of the number of labeled images, all of our methods obtain great results under a fair comparison with previous methods. The threshold decline curves of each analogy obtained during training are shown in Fig. 2. When a quarter of the data is labeled, our experimental results exceed the supervised learning baseline by 6.97%, which slightly exceeds the performance of ST++ [15] by 0.18%, AEL [33] by 0.38%, etc, and reached the current optimum. When the amount of labeled data accounts for 1/30, our performance exceeds the baseline of supervised learning by 13.09%, and it can also be slightly better than methods such as ST++ [15] by 0.62% and CPS [13] by 2.41%, second only to AEL [33]. When the amount of labeled data accounts for 1/8, our performance has declined somewhat, only exceeding the baseline of supervised learning by 3.39%, but it is also better than PseudoSeg [14] by 0.2%, ClassMix [32] by 8.6% and other worse methods. The Cityscapes dataset contains complex street view images, and the experimental results on the Cityscapes dataset indicate that our DTFSeg achieves competitive performance for multi-label images with complicated scenes. The experimental results demonstrate that effective threshold filtering can eliminate imprecise false labels in semi-supervised semantic segmentation to some extent.

UAVid. Compared to the Cityscapes dataset, the UAVid dataset images come from a top-down view, the distribution of categories is more unbalanced, and the generation of correct pseudo-labels is more difficult, in which case our method

TABLE I
Comparison with SOTA methods on the Cityscapes dataset.

Method	**1/30(100)**	**1/8(372)**	**1/4(744)**
SupOnly	56.84	66.41	70.70
AdvSeg [29]	-	58.80	62.30
S4GAN [30]	-	59.30	61.90
ECS [31]	-	67.40	70.70
CutMix [7]	55.70	65.80	68.30
ClassMix [32]	54.10	61.40	63.60
PseudoSeg [14]	61.00	69.80	72.04
DCC [17]	-	69.70	72.70
CPS [13]	61.52	73.82	74.02
AEL [33]	**64.36**	73.95	75.72
ST [15]	62.49	73.56	75.61
ST++ [15]	63.31	**74.16**	75.92
DTFSeg(Ours)	63.93	70.00	**76.10**

TABLE II
Compared with other threshold filtering methods on UAVid dataset

Method	1/30(116)	1/8(435)	1/4(870)
SupOnly	63.41	67.19	68.48
ST [15]	66.55	69.84	71.24
ST++ [15]	68.02	70.66	**72.04**
SSL_ELN [34]	65.55	68.63	68.63
DTFSeg(ours)	**68.10**	**71.00**	71.80

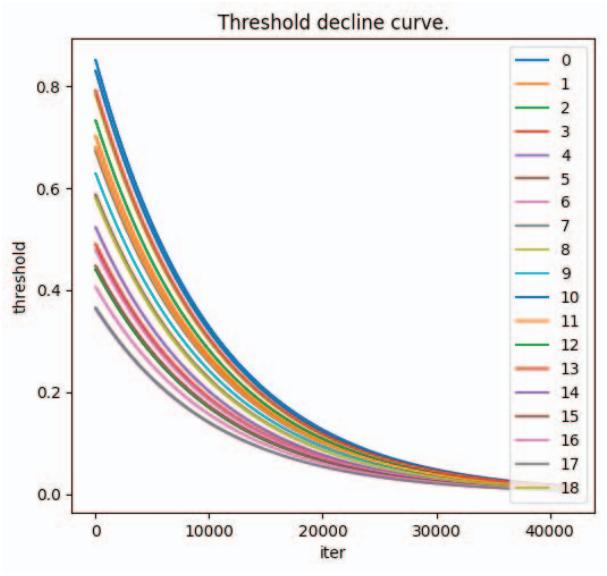

Fig. 2. Class-wise dynamic threshold decline curve on the Cityscapes dataset (1/4 labeled). The training procedure starts with a high threshold (limited but confident labels) and ends with a threshold near zero (almost the whole dataset).

TABLE III
COMPARED WITH OTHER THRESHOLD FILTERING METHODS ON THE CITYSCAPES DATASET

Method	1/30(100)	1/8(372)	1/4(744)
SupOnly	56.84	66.41	70.70
wo-th	62.89	69.37	74.01
fixed-th	62.90	69.45	75.50
en-th	49.46	62.30	69.09
max-th	58.55	69.34	75.48
DTFSeg(ours)	**63.93**	**70.00**	**76.10**

also has some advantages over other methods. As we can see in Table II, our method exceeds the supervisory method by 4.69%, 3.81%, and 6.32% when the proportions of labeled data are 1/30, 1/8, and 1/4 respectively, most of them exceeding the methods such as ST [15], SSL_ELN [34], etc. When the amount of labeled data accounts for 1/30, our performance exceeds the methods such as ST++ [15] by 0.08% and SSL_ELN [34] by 2.55%, when the amount of labeled data accounts for 1/8, our performance exceeds ST++ [15] by 0.34% and SSL_ELN [34] by 2.37%, when the amount of labeled data accounts for 1/4, our performance exceeds SSL_ELN [34] by 3.17%, and it only does not exceed the ST++ method, but it's very close as well. These results illustrate that our method of filtering pseudo-labels based on dynamic threshold still achieves good results in the presence of an extreme imbalance of categories in the image.

C. Comparison with Other Threshold Filtering Methods

In order to prove that it is threshold filtering method that we proposed works in our network, and that our threshold filtering method outperforms other filtering methods, we design three other threshold filtering methods, which are based on the fixed threshold, entropy-based threshold, and Logits-based threshold in pixel classification results, and apply the thresholdless filtering method for comparison. As shown in Table III, specifically, the Settings for each threshold method are as follows:

- wo-th: Without threshold. Instead of threshold filtering, the prediction of the teacher model is used as a pseudo-label directly.
- fixed-th: Fixed threshold. Different from other methods that use fixed thresholds [17], [18], we design a more reasonable method to set thresholds for each category separately. We use the average confidence of each category when it is selected as a prediction category in the last round of supervised training before adding unlabeled data, multiplied by 0.75 as the threshold.
- en-th: Entropy threshold. Inspired by [20], we use the teacher model to predict the unlabeled images and use the entropy of each pixel in the results as the basis for threshold filtering. According to the entropy of each pixel in each image, we delete a certain proportion of pixels with high entropy. We reduce linearly the proportion of pixels that are filtered as the training time increases and set each unlabeled image to filter 25% to 0 from the semi-supervised beginning.
- max-th: Maximum probability threshold. Similar to the threshold filtering method in "en-th", according to the maximum logit value of each pixel in each image, we delete a certain proportion of pixels with low maximum logit value, and we reduce linearly the proportion of pixels that are filtered as the training time increases and set each unlabeled image to filter 25% to 0 from the semi-supervised beginning.

In Table III, we can observe that our threshold setting method is more effective than other threshold setting methods in Cityscapes datasets with different proportions of labeled data. When the proportion of labeled data is 1/30, our method is 1.03% higher than the fixed threshold, which is the best-performing method except our method. When the labeled data accounted for 1/8 and 1/4 of the data, the best methods except our method were fixed threshold filters, and our dynamic threshold setting method exceeded them by 0.45% and 0.6%, respectively. Among all the different proportion of labeled data, the threshold filtering method using entropy gets the worst performance. Our method yielded the best results.

IV. CONCLUSION

This paper proposes a semi-supervised semantic segmentation method based on the threshold setting of the exponential function, together with loss weighting, which allows our method to achieve more advanced results. Unlike other threshold filtering methods, ours sets separate thresholds for each category and uses the supervised average confidence level as the initial threshold, which more accurately distinguishes the degree of confidence among the categories, and

the exponential form of the threshold decreases to match the rising state of the model's predictive ability. Experiments on Cityscapes and UAVid dataset demonstrate the effectiveness of our method.

ACKNOWLEDGMENT

This work is supported in part by the Natural Science Foundation of NingBo (2023J262), the Natural Science Foundation of China (62201479), the Natural Science Foundation of Sichuan Province (2023NSFSC1388), the Open Fund of Key Laboratory of Civil Aircraft Airworthiness Technology (SH2020112706).

REFERENCES

[1] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, 2020.

[2] K. Muhammad, T. Hussain, H. Ullah, J. Del Ser, M. Rezaei, N. Kumar, M. Hijji, P. Bellavista, and V. H. C. de Albuquerque, "Vision-based semantic segmentation in scene understanding for autonomous driving: Recent achievements, challenges, and outlooks," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[3] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *The International journal of robotics research*, vol. 39, no. 2-3, pp. 183–201, 2020.

[4] Z. Zhou, X. Zhang, L. Ran, Y. Han, and H. Chu, "Dsc-graspnet: A lightweight convolutional neural network for robotic grasp detection," in *2023 9th International Conference on Virtual Reality (ICVR)*. IEEE, 2023, pp. 226–232.

[5] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "Ds-slam: A semantic visual slam towards dynamic environments," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1168–1174.

[6] L. Zhong, X. Zhang, L. Ran, Y. Han, and H. Chu, "Visual slam for dynamic environments based on static key-points detection," in *2023 9th International Conference on Virtual Reality (ICVR)*. IEEE, 2023, pp. 93–99.

[7] G. French, S. Laine, T. Aila, M. Mackiewicz, and G. Finlayson, "Semi-supervised semantic segmentation needs strong, varied perturbations," in *British Machine Vision Conference*, no. 31, 2020.

[8] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[9] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6023–6032.

[10] Y. Ouali, C. Hudelot, and M. Tami, "Semi-supervised semantic segmentation with cross-consistency training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 674–12 684.

[11] D. Ruan, D. Wang, Y. Zheng, N. Zheng, and M. Zheng, "Gaussian context transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 129–15 138.

[12] Z. Feng, Q. Zhou, Q. Gu, X. Tan, G. Cheng, X. Lu, J. Shi, and L. Ma, "Dmt: Dynamic mutual training for semi-supervised learning," *Pattern Recognition*, vol. 130, p. 108777, 2022.

[13] X. Chen, Y. Yuan, G. Zeng, and J. Wang, "Semi-supervised semantic segmentation with cross pseudo supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2613–2622.

[14] Y. Zou, Z. Zhang, H. Zhang, C.-L. Li, X. Bian, J.-B. Huang, and T. Pfister, "Pseudoseg: Designing pseudo labels for semantic segmentation," in *International Conference on Learning Representations*, 2020.

[15] L. Yang, W. Zhuo, L. Qi, Y. Shi, and Y. Gao, "St++: Make self-training work better for semi-supervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4268–4277.

[16] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 596–608.

[17] X. Lai, Z. Tian, L. Jiang, S. Liu, H. Zhao, L. Wang, and J. Jia, "Semi-supervised semantic segmentation with directional context-aware consistency," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1205–1214.

[18] Y. Wang, H. Wang, Y. Shen, J. Fei, W. Li, G. Jin, L. Wu, R. Zhao, and X. Le, "Semi-supervised semantic segmentation using unreliable pseudo-labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4248–4257.

[19] Z. Feng, Q. Zhou, G. Cheng, X. Tan, J. Shi, and L. Ma, "Semi-supervised semantic segmentation via dynamic self-training and classbalanced curriculum," 2020.

[20] Z. Cai, X. Yan, Y. Wu, K. Ma, J. Cheng, and F. Yu, "Dgcl: an efficient communication library for distributed gnn training," in *Proceedings of the Sixteenth European Conference on Computer Systems*, 2021, pp. 130–144.

[21] Y. Xu, L. Shang, J. Ye, Q. Qian, Y.-F. Li, B. Sun, H. Li, and R. Jin, "Dash: Semi-supervised learning with dynamic thresholding," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 525–11 536.

[22] 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*, 2016, pp. 3213–3223.

[23] Y. Lyu, G. Vosselman, G.-S. Xia, A. Yilmaz, and M. Y. Yang, "UAVid: A semantic segmentation dataset for UAV imagery," *ISPRS journal of photogrammetry and remote sensing*, vol. 165, pp. 108–119, 2020.

[24] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[27] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[28] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 761–769.

[29] W. C. Hung, Y. H. Tsai, Y. T. Liou, Y. Y. Lin, and M. H. Yang, "Adversarial learning for semi-supervised semantic segmentation," in *29th British Machine Vision Conference, BMVC 2018*, 2019.

[30] S. Mittal, M. Tatarchenko, and T. Brox, "Semi-supervised semantic segmentation with high-and low-level consistency," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 43, no. 4, pp. 1369–1379, 2019.

[31] R. Mendel, L. A. De Souza, D. Rauber, J. P. Papa, and C. Palm, "Semi-supervised segmentation based on error-correcting supervision," in *Proceedings of the European conference on computer vision (ECCV)*. Springer, 2020, pp. 141–157.

[32] V. Olsson, W. Tranheden, J. Pinto, and L. Svensson, "Classmix: Segmentation-based data augmentation for semi-supervised learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1369–1378.

[33] H. Hu, F. Wei, H. Hu, Q. Ye, J. Cui, and L. Wang, "Semi-supervised semantic segmentation via adaptive equalization learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 106–22 118, 2021.

[34] D. Kwon and S. Kwak, "Semi-supervised semantic segmentation with error localization network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9957–9967.