Towards More Efficient Semi-Supervised Semantic Segmentation

Mentor: Hao Chen
Xinyue Lu (xinyuelu@andrew.cmu.edu)
Ricky Lu (qiruil@andrew.cmu.edu)
Bradley Feng (wangchuf@andrew.cmu.edu)
Zeda Xu (zedax@andrew.cmu.edu)

Abstract

In this project, we studied the semi-supervised semantic segmentation problem based on both labeled and unlabeled data. The baseline model is a consistency regularization approach with cross pseudo supervision (CPS), which imposes the consistency on 2 segmentation networks perturbed with different initialization for the same input. It is the state-of-the-art semi-supervised segmentation approach on Cityscapes and PASCAL VOC 2012 datasets. In order to improve the model, we first encouraged only confident predictions by applying threshold to filter the probability. We modified the current model to include the fixed threshold and the progressive class-based training threshold, but they did not work well. We then made use of knowledge distillation on ensemble model and reinforced the learning process with 2 stages of training. Experiment results show that this approach leads to a better performance compared to the baseline model.

1 Introduction

Semi-supervised learning has become a popular topic in the area of machine learning in recent years. In traditional machine learning problems, a label is require for all the data in order to perform the learning process. However, in some real-world problems, it is hard to find all the data labeled. In fact the more common situation is that there are far more unlabeled data than the labeled data. This problem is particularly prominent in computer-vision-related areas[2], for example, autonomous driving. For an autonomous driving vehicle, huge amount of data of the road can be easily accessed with cameras, radars, etc., but there's no efficient way to easily label all the data yet. In situations like this, semi-supervised learning, where the training process doesn't require every data to have a label, can be especially meaningful, as it can exploit a group of data even when a large amount of them are unlabeled.

Consistency regularization is a key step in semi-supervised segmentation[4]. In consistency regularization, the same unsupervised data with various perturbations are predicted using the same network, and the consistency of the predicted results is required. The perturbation methods include flipping, cropping, adversarial attacks, etc. On top of that, feature perturbation is a scheme where an unsupervised data is predicted by two networks with the same structure but initialized differently, and by enforcing the consistency of the prediction results, the network features can be further perturbed. The consistency regularization can be performed in an iterative process, where in each iteration the pseudo labels for a certain amount of unsupervised data are predicted and these unsupervised data are treated as new supervised data, which is an expansion of currently existing supervised data, in all following iterations, and in each iteration the original model is also retrained with all original and newly-generated supervised data.

Preprint. Under review.

The PASCAL VOC 2012 dataset, which consists of over 13000 images that can be divided into 20 object classes and 1 background class, is used in this project to test and evaluate the model performance. A knowledge distillation scheme on ensemble model with cross pseudo supervision is implemented on the basis of our baseline model, and it successfully improves the model performance by around 0.7 % in each category, which verifies the feasibility of our method.

2 Related Work

2.1 Semi-Supervised Semantic Segmentation

Semantic segmentation is one of the basic tasks in computer vision. In semantic segmentation, the training data is labeled at pixel level, which means specific regions in an image are labeled according to what is being showed in those regions. Compared to other traditional computer vision tasks like image classification and object detection, semantic segmentation is more computationally expensive. As mentioned in the introduction section, most real-world semantic segmentation tasks are semi-supervised semantic segmentation, since the labels for most data are always hard to obtain. Different approaches of consistency regularization have been widely used in semi-supervised segmentation tasks, including feature perturbation[6], GAN-based approach[5], self-training[8], etc.

Feature perturbation predicts the same data using two networks with the same structure but different initializations[2], and the network is augmented accordingly to force consistency between the predicted results. In the GAN-based approach, the features of the segmentation maps from the ground-truth for labeled data and the predicted results for unlabeled data are extracted using a discriminator network and compared in order to force the consistency of both results. Self-training, which was initially invented for self-supervised classification tasks, is a process of iteratively training the segmentation model, and in each iteration pseudo labels are generated for a certain amount of unsupervised data and then incorporated into the existing supervised data for the training process in following iterations.

2.2 Semi-Supervised Classification

Semi-supervised classification is also a key topic in semi-supervised learning[1]. For a semi-supervised dataset, the supervised data can be easily classified using their corresponding labels, while unsupervised data are hard to classify since there's are no labels attached to them. So the main objective of semi-supervised classification is to roughly classify the unsupervised portion of the data based on the classification results of the supervised portion of the data.

Most current solutions for semi-supervised classification problems use various assumptions to provide extra information for the unlabeled data, such as smoothness, consistency, low-density, and clustered. Generally these solutions are based on the intuition that data within neighborhood is highly possible to be similar with each other, and data with different labels can be easily separated by clear decision boundaries that lie in low-density regions[2].

2.3 Cross Pseudo Supervision

Cross pseudo supervision is an approach proposed for the learning process of a segmentation network using both labeled data and unlabeled data[2]. In cross pseudo supervision, the same dataset is predicted using two different but parallel segmentation networks, and the results are compared and the networks are augmented accordingly. Figure 1 shows the basic structure of cross pseudo supervision. In figure 1, X represents the input, $f(\theta_1)$ and $f(\theta_2)$ are two segmentation networks with weights θ_1 and θ_2 respectively, P_1 and P_2 are the segmentation confidence map that represents the probability distribution of the segmentation results output by the network, and Y_1 and Y_2 are the pseudo segmentation maps, which is the predicted one-hot label map.

The total loss of the training process $\mathcal{L}$ is made up by two parts: supervision loss $\mathcal{L}_s$ and cross pseudo supervision loss $\mathcal{L}_{CPS}$, and it is formulated as:

$$\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_{CPS} \tag{1}$$

where λ is the trade-off weight.

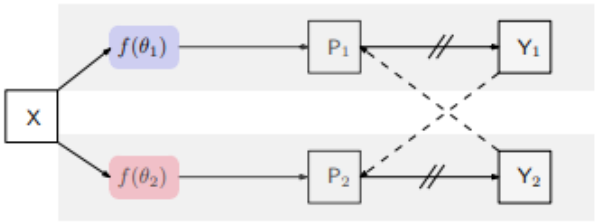

Figure 1: Cross-Pseudo Supervision[2]

The supervision loss $\mathcal{L}_s$ measures the the pixel-wise cross-entropy loss over the labeled images for both networks:

$$\mathcal{L}_s = \frac{1}{|\mathcal{D}^l|} \sum_{X \in \mathcal{D}^l} \frac{1}{W \times H} \sum_{i=0}^{W \times H} \left(l_{ce} \left(\mathbf{p}_{1i},\ \mathrm{y}^*_{1i} \right) + l_{ce} \left(\mathbf{p}_{2i},\ \mathrm{y}^*_{2i} \right) \right) \tag{2}$$

where $\mathcal{D}^l$ is the set of all labeled images, and W and H represent the width and height of the input image, respectively.

The cross pseudo supervision loss $\mathcal{L}_{CPS}$ is simply the sum of the losses on both the labeled and unlabeled data:

$$\mathcal{L}_{CPS} = \mathcal{L}^l_{CPS} + \mathcal{L}^u_{CPS} \tag{3}$$

The CPS loss on the unlabeled data $\mathcal{L}^u_{CPS}$ is defined as:

$$\mathcal{L}^u_{CPS} = \frac{1}{|\mathcal{D}^u|} \sum_{X \in \mathcal{D}^u} \frac{1}{W \times H} \sum_{i=0}^{W \times H} \left(l_{ce} \left(\mathbf{p}_{1i},\ \mathrm{y}_{2i} \right) + l_{ce} \left(\mathbf{p}_{2i},\ \mathrm{y}_{1i} \right) \right) \tag{4}$$

where $\mathcal{D}^u$ is the set of all unlabeled images.

The CPS loss on the labeled data $\mathcal{L}^l_{CPS}$ is formulated in the same way as on the unlabeled data.

2.4 Noisy Student

Noisy student training is a widely-applied approach in semi-supervised learning[8]. It has three main steps: (1) use all labeled images to train a teacher model; (2) use the teacher model to generate pseudo labels for all unlabeled images; (3) use the labeled images (with true labels) and unlabeled images (with pseudo labels) to train a student model. In this process, the student model is required to be larger than (or at least equal to) the teacher model, and thus the student model can have abetter performance in learning from a large dataset. On top of that, some noise may also be added to the student network, so the student network is required to learn harder from the pseudo labels. The noise includes input noise (e.g. data augmentation) and model noise (e.g. dropout and stochastic depth). Figure 2 shows the basic structure of the noisy student training process.

3 Baseline Model

The baseline model that we built upon in this project is a model proposed by Xiaokang Chen et al[2] in 2021. It applies cross pseudo supervision to construct two segmentation networks, and makes use of pseudo segmentation maps on the unlabeled images to achieve consistency regularization. It is the state-of-the-art model in the semi-supervised semantic segmentation task on the Cityscapes dataset (12.5% labeled, 25% labeled, and 50% labeled), and also one of the top-performed model in the semi-supervised semantic segmentation task on the PASCAL VOC 2012 dataset (both 12.5% labeled and 25% labeled).

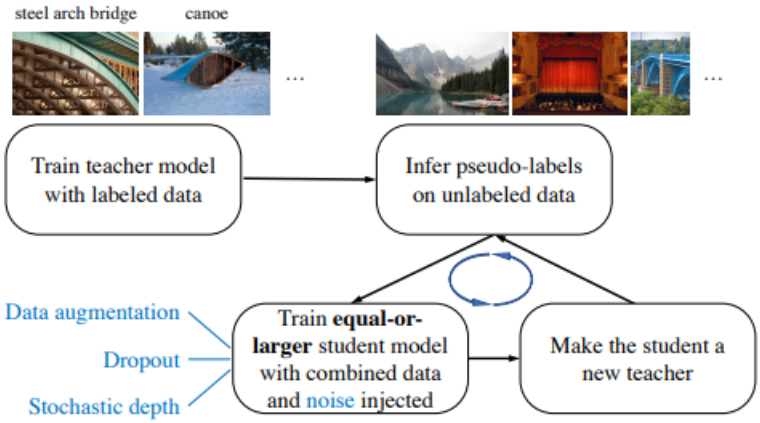

Figure 2: Noisy Student Training

4 Methodology

4.1 Fixed Threshold

To improve the performance, we borrowed some ideas from the FixMatch model[7]. We are proposing a threshold method that only encourages confident predictions. Only when the model assigns a probability above a certain threshold, the prediction will be converted to a one-hot pseudo-label. We tested with different fixed threshold values in our experiments. This in theory could help improving the accuracy by clearing out those fussy predictions.

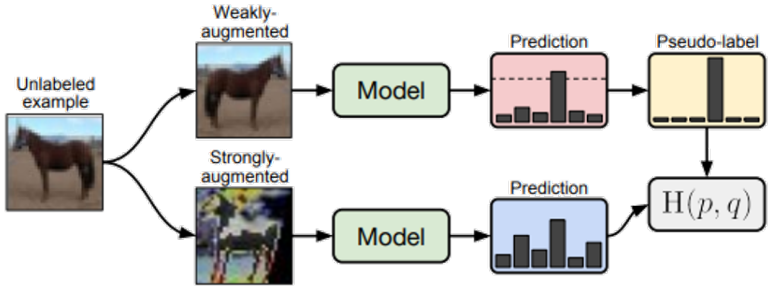

Figure 3: Basic idea of FixMatch(2020)

4.2 Progressive Class-Based Training Threshold

During each iteration, the Exponential Moving Average(EMA) of mean confidence is applied as the threshold for each class, as this provides more flexibility comparing to fixed thresholds. The EMA of class-wise threshold can be calculated recursively:

$$v_t^i = \beta v_{t-1}^i + (1-\beta)\theta_t^i \tag{5}$$

where θ_t^i here can be interpreted as the mean confidence for ith class during the tth iteration, v_t^i is the exponential moving average which will be used as the threshold for ith class during the first tth iteration. The coefficient β represents the weight decrease degree, a higher β discounts the previous confidence faster. We pick $\beta = 0.96$ here.

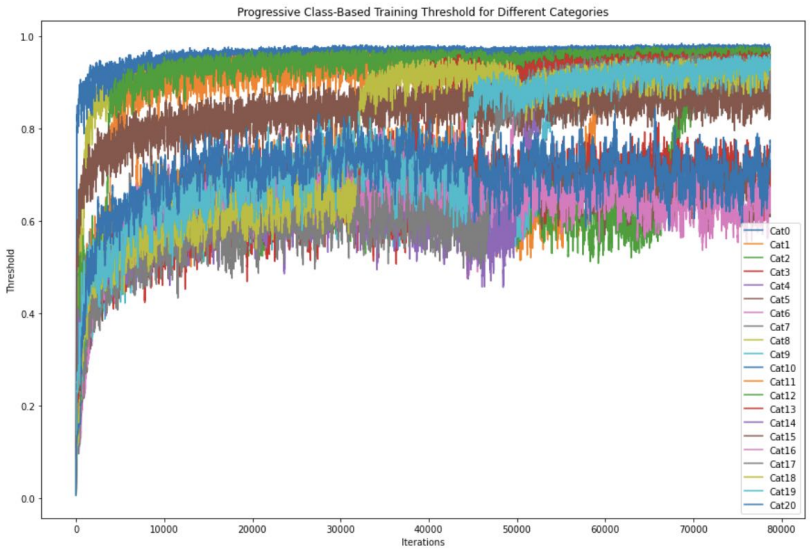

Figure 4: Training threshold is updated each iteration for each class.

4.3 Knowledge Distillation on Ensemble Model

Using the Noisy Student Training method as a reference, a knowledge distillation approach is proposed to modify our model. As shown in the figure below, the model applied in this project consists of two major stages. In the first stage, the labels of all supervised data and unsupervised data are predicted using the same baseline network but following different rules. For supervised data, the network is first trained using the corresponding true labels of the data and then generate the predicted labels for the data. For unsupervised data, the labels are predicted using the consistency regularization algorithm, where the label for the same data is predicted by two identically-structured network with different initialization. When the two networks are generating different labels, at least one of them is definitely wrong, then the two networks are augmented accordingly. And when they are generating the same labels, this label is treated as the true label. The second stage is another training stage, where the same network with no initialization is trained with all supervised and unsupervised data with their corresponding labels. For supervised data, the input are the true labels, while for unsupervised data, the input are the predicted labels obtained from the first stage.

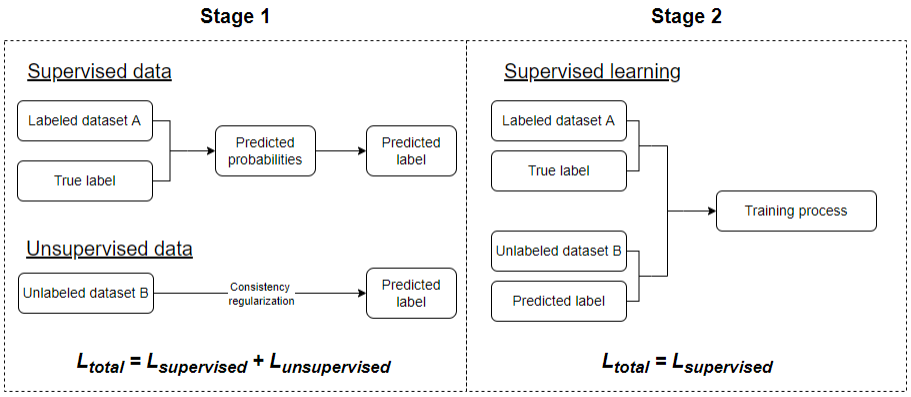

Figure 5: Knowledge Distillation on Ensemble Model

5 Experiments

5.1 Setup

Datasets In this study, the PASCAL VOC 2012[3] has been used as the main dataset to test and evaluate the model performance. This dataset consists of over 13000 images that can be divided into 20 object classes and 1 background classes. On top of that, there are 1464 images for standard training set, 1449 for validation set, and 1456 for test set. We used the augmented set (10,582 images) as our full training set. Every input will have a size of $512 \times 512 \times 3$. 1/8 of the whole dataset is sub-sampled randomly as the labeled partition, and the rest are unlabeled partition. Different portions in this process can also be used to further validate the results.

Evaluation We are evaluating the performance of the model by three metrics: mean Intersection over Union (mIoU), mean Intersection over Union with no background class, and mean pixel label accuracy. These are also the performance metrics used in the baseline model paper. Combined, they could give us a relatively good idea of the overall performance of each method. The evaluation results are obtained from only one network.

Experimental Setup The setup is running two Nvidia GeForce RTX 2080 Ti GPUs for all the training and testing work. The Nvidia driver version is 510.47.03 (UNIX). PyTorch version 1.8.0 with CUDA version 11.1.

5.2 Results

The performance of each model is shown in Table 1 below. Please notice that, the baseline performance shown here is our recreated results, using the exact model setup and training procedure indicated by the original paper and author's GitHub repository. This is different from the claimed 73.20% in the paper, probably because of different experimental environment setup and randomness in the training process.

Model	**Mean IoU**	**Mean IoU w/ no Background**	**Mean Pixel Accuracy**
Baseline (BL)	71.192 %	70.124 %	93.332%
Threshold 0.1	70.976 %	69.901 %	93.308%
Threshold 0.3	70.735 %	69.656 %	93.171%
Threshold 0.7	70.245 %	69.138 %	93.174%
Progressive Class-Based Training Threshold	63.012 %	61.695 %	89.884%
Knowledge Distillation on Ensemble Model	71.910% (+0.718 % o/ BL)	70.854% (+0.730 % o/ BL)	93.652% (+0.320 % o/ BL)

Table 1. Performance of Each Method

As we can see from the chart, our Knowledge Distillation on Ensemble Model has better performance than the baseline model in all three metrics. However, the Fixed Threshold method and the Progressive Class-Based Training Threshold model performed noticeably worse than the baseline. In fact, the Progressive Class-Based Training Threshold model is performing the worst comparing to all other models, despite it in theory could yield better results than the Fixed Threshold method.

Here are some sample of the segmented image results, obtained with different models. There are four different classification classes, including airplane, person, bicycle and bus.

An unexpected result here is that the baseline model has the sharpest and clearest borderlines. Especially for the airplane and bus image, the baseline model has the most accurate segmentation borders, and is the only one that retains the shape of the object properly. All other models tested

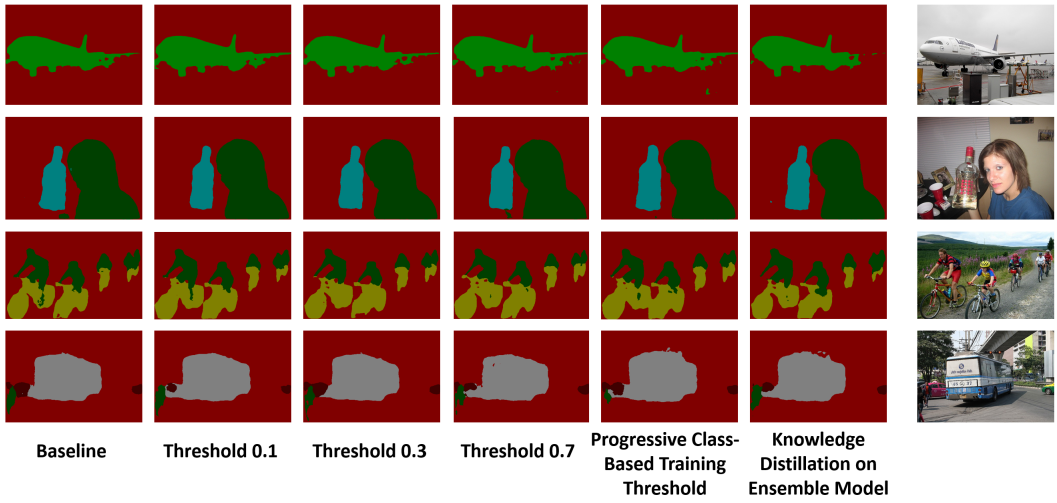

Figure 6: Sample Segmented Image Results

performed worse and losses some of the details. The problems including the missing forearm on the person, the incomplete wheel of the bicycle, and the weird tail on the left of the bus. The Knowledge Distillation on Ensemble Model, despite being the best in average IoU and accuracy, completely lost the shape of the airplane wings.

6 Discussion

6.1 Error Analysis

We had high hopes for the fixed threshold method as it theoretically would help improve the accuracy of the basic Cross Pseudo Supervision technique, but it performed much worse than expected for all tested threshold values. This is potentially because it filtered out many details in the process. We are guessing the same reason why a threshold value of 0.1 is outperforming values of 0.3 or 0.7. In this sense, the values of the threshold could be finer-tuned further, to reduce the negative masking effects while maintaining the boost in prediction confidence. The disappointing performance of the Progressive Class-Based Training Threshold method might be a result of its deficiency in predicting a few certain classes. This leads to inaccurate thresholds of those classes and potentially hinders overall performance. However, it is still surprising to see it performed the worst as it should serve as an advanced version of the simpler Fixed Threshold method. As for the best performing knowledge distillation model, we think its high accuracy is more likely benefited from its nature of being an ensemble model, rather than anything special.

As for the unexpected result on the sample images, we think the problem lies on that all our proposed model are trying to improve the accuracy by the increasing the confidence of unsupervised learning in different ways. This causes our models can be a little too conservative sometimes, that them would trade some of the details for a better overall accuracy.

6.2 Future steps

For future works, The team will also test more methods, including Momentum Moving-Average Growing Thresholds which gradually increasing the threshold, and using larger networks for Knowledge Distillation model.

For Momentum Moving-Average Growing Thresholds, The approach is to implement momentum "growing" thresholds for each output class. The current threshold methods do not achieve competitive performance, either the fix thresholds or the EMA thresholds. For a particular class, the formula for threshold update is

$$v_t = v_{t-1} + \alpha\theta_t + \beta(v_{t-1} - v_{t-2}) \tag{6}$$

where θ_t here can be interpreted as the mean confidence during the t-th iteration, α and β are the coefficients. v_t is the threshold during the t-th iteration.

For larger Knowledge Distillation model, the current approach uses ResNet50 for the final supervised learning. It could be changed with other different networks, especially larger networks to improve the performance.

Moreover, the team is planning to re-run our tests on all proposed models with a ResNet 101 backbone instead of the current ResNet 50 variant. This could in theory improve the performance, and we are excited to see how our models are compared to the current SOTA on PASCAL VOC 2012 that also based on the ResNet 101 model.

Moreover, different structures for the two semi-supervised networks would be tested. For example, instead of supervising each other in parallel, we could set one network to be the "student", and the other network to be the "teacher". Thus, the teacher will supervise the student, but the student will not supervise the teacher.

The team will test all methods on other datasets to validate the results, including Cityscapes which the baseline also evaluated its performance on. Different labeled/unlabeled proportions in the PASCAL VOC 2012 training dataset could be experimented, say 1/2, 1/4, 1/8, etc. This would very possibly yield different results. Comparing results to those obtained with the baseline model could give us more insight of the strengths and weakness of our proposed models. We are also excited to see how our models stack up with the SOTA models in different labeled/unlabeled proportion settings.

7 Conclusion

In this project, we studied the semi-supervised semantic segmentation tasks in the field of computer vision. We referred to the baseline model using cross pseudo supervision proposed by Xiaokang Chen et al in 2021. Then we successfully implemented several new approaches to enhance consistency network based on the baseline model, including fixed threshold, progressive class-based training threshold, knowledge distillation on ensemble model, etc. Although the first two threshold-related approaches don't work as expected, the knowledge distillation approach, where the predicted pseudo labels for unlabeled dataset are generated using consistency regularization and further applied for network training, yields a very good performance, and improves the accuracy by 0.7 % compared to the baseline model.

This result shows that knowledge distillation is an effective way to provide extra information for a semi-supervised dataset, where only a certain portion of the data are labeled, and this addition of dataset information can lead to significant improvement in the model performance when a semi-supervised learning task is involved, as it in fact is an expansion on the size of the labeled dataset. In this process, both consistency regularization and cross pseudo supervision play a vital role in terms of providing reasonable prediction results for unlabeled dataset.

Generally, in this project, we are able to implement a more conservative model with better accuracy in certain semi-supervised semantic segmentation tasks compared to the baseline model. Hopefully our work could contribute to the research in computer vision, especially in semi-supervised semantic segmentation tasks.

References

[1] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 2006.

[2] Xiaogang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. *Semi-Supervised Semantic Segmentation with Cross Pseudo Supervision. Computer Vision and Pattern Recognition*, 2021.

[3] Mark Everingham, SM Ali Aslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. *The Pascal visual object classes challenge: A retrospective*. IJCV, 2015.

[4] Jongmok Kim, Joo young Jang, and Hyunwoo Park. *Structured Consistency Loss for Semi-Supervised Semantic Segmentation*. CoRR, 2020.

[5] Sughanshu Mittal, Maxim Tatarchenko, and Thomas Brox. *Semi-Supervised Semantic Segmentation with High- and Low-Level Consistency*. CoRR, 2019.

[6] Yassine Ouali, Celine Hudelot, and Myriam Tami. *Semi-Supervised Semantic Segmentation with Cross-Consistency Training*. CVPR, 2020.

[7] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Niholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. *FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence*. 2020.

[8] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. *Self-Training with Noisy Student Improves Imagenet Classification*. CVPR, 2020.