SEMI-SUPERVISED SEMANTIC SEGMENTATION USING AUXILIARY NETWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, the convolutional neural networks (CNNs) have shown great success on semantic segmentation task. However, for practical applications such as autonomous driving, the popular supervised learning method faces two challenges: the demand of low computational complexity and the need of huge training dataset accompanied by ground truth. Our focus in this paper is semi-supervised learning. We wish to use both labeled and unlabeled data in the training process. A highly efficient semantic segmentation network is our platform, which achieves high segmentation accuracy at low model size and high inference speed. We propose a semi-supervised learning approach to improve segmentation accuracy by including extra images without labels. While most existing semi-supervised learning methods are designed based on the adversarial learning techniques, we present a new and different approach, which trains an auxiliary CNN network that validates labels (ground-truth) on the unlabeled images. In the supervised training phase, both the segmentation network and the auxiliary network are trained using labeled images. Then, in the unsupervised training phase, the unlabeled images are segmented and a subset of image pixels are picked up by the auxiliary network; and then they are used as ground truth to train the segmentation network. Thus, at the end, all dataset images can be used for retraining the segmentation network to improve the segmentation results. We use Cityscapes and CamVid datasets to verify the effectiveness of our semi-supervised scheme, and our experimental results show that it can improve the mean IoU for about 1.2% to 2.9% on the challenging Cityscapes dataset.

1 INTRODUCTION

Semantic Segmentation, which identifies the category label of each pixel, is an important task in computer vision. A number of convolutional neural network (CNN) based semantic segmentation systems have been developed in recent years such as Chen et al. (2017a;b; 2018); Long et al. (2015); Zhao et al. (2017); Ronneberger et al. (2015). For practical applications such as autonomous driving, there are a high demand for real-time processing and sufficient training data. Hence, different research directions have been explored. For example, some proposed various efficient network structures (Chen et al., 2019; Paszke et al., 2016; Poudel et al., 2018; Lo et al., 2018; Poudel et al., 2019; Yu et al., 2018; Zhao et al., 2018; Romera et al., 2017; Li et al., 2019), and others focus on the weakly- or semi-supervised learning schemes (Hung et al., 2018; Lin et al., 2016; Bearman et al., 2016; Qi et al., 2016; Rajchl et al., 2016; Pathak et al., 2015; Papandreou et al., 2015; Hong et al., 2015; Luc et al., 2016; Dai et al., 2015; Pathak et al., 2014).

In this study, we first design an efficient segmentation network inspired by Chen et al. (2019) as our baseline model, and then propose a semi-supervised learning scheme. There are three training stages in our system. In the first stage, the segmentation network is trained using the labeled images. Then, in the second stage, an auxiliary network is trained using the segmentation network trained in the previous stage and the labeled data to generate the confidence map. Inspired by Hung et al. (2018), we use the concept of confidence map to assign a confidence score to each segmented pixel on the unlabeled images. In their work (Hung et al., 2018), the confidence map is the output of a discriminator network trained by the GAN framework (Goodfellow et al., 2014), where the discriminator network learns to distinguish between the segmentation map and the ground truth map. However, we find that the trusted (high confidence) regions in their confidence map are mostly located on the large target objects, and thus the effectiveness of semi-supervised learning is limited. To obtain more reliable small object labels, we adopt a different approach in generating the confidence map. In our approach, the confidence map is generated by an auxiliary (CNN) network, which is trained using the proposed auxiliary loss function. We carefully design the auxiliary loss function such that it leads to a reliable confidence map, particularly, on the small objects. In the third stage, the unlabeled images are used as inputs to the proposed system to generate the labels. Therefore, we can use both originally labeled and unlabeled images to retrain the segmentation network to achieve a better performance in the end.

In summary, our main contributions of this work are as follows. First, Based on DSNet-fast (Chen et al., 2019), we design a powerful segmentation network, which achieves a very good balance between speed and accuracy. It produces 73.9% mean IoU on the Cityscapes testing set with a speed of 73 FPS on a single GTX 1080Ti card. Second, We propose a semi-supervised learning scheme with the notion of *auxiliary network*, which can be used to annotate the unlabeled images. Third, Our semi-supervised learning method can include unlabeled images in training, which improves the segmentation accuracy.

2 Related work

Recently, CNNs have been widely used in many fields of computer vision. For the semantic segmentation task, FCN (Long et al., 2015) is a pioneer. It replaces the fully-connected layers of the classification network (Krizhevsky et al., 2012) with convolution layers, and thus it can generate dense label prediction of the same size as input image. After that, SegNet (Badrinarayanan et al., 2017) is subsequently proposed, which uses a symmetric encoder-decoder architecture for feature down-sampling and up-sampling. U-Net (Ronneberger et al., 2015) introduces the concatenation operation to up-sample the features with different levels. PSPNet (Zhao et al., 2017) and DeepLab (Chen et al., 2017a;b; 2018) propose the atrous spatial pyramid pooling (ASPP) module to integrate multi-scale features. There are many other studies on improving the segmentation results; however, for practical applications, a high inference speed network is very desirable. For example, ENet (Paszke et al., 2016) is a real-time segmentation model with good segmentation results. ICNet (Zhao et al., 2018) and BiSeNet (Yu et al., 2018) were recently proposed, and they aim at a better balance between speed and accuracy.

Another challenge in training a semantic segmentation network is the need of a large amount of labeled data (ground truth). The labeling cost of pixel-level annotation is extremely expensive. Hence, in recent years, the study of the weakly- and semi-supervised learning approaches become popular to tackle this problem and attracted a lot of attention. For the weakly supervised methods, pixel-level annotations are no longer used to train the segmentation network, but instead, other forms of annotations that are easier to obtain are used for training. There are several different types of weakly-supervision have been studied, including bounding box supervision (Rajchl et al., 2016; Dai et al., 2015), scribbles supervision (Lin et al., 2016), point supervision (Bearman et al., 2016), and image-level supervision (Qi et al., 2016; Pathak et al., 2014). Particularly, the development of image-level supervision is the most popular one. Pathak et al. (2015) convert image-level labels to restrict the distribution of CNN output. Papandreou et al. (2015) combine image-level labels with EM algorithms to train the segmentation model. Wei et al. (2018) propose a generic classification network, which adopts the convolutional blocks with different dilated rates to generate dense object localization maps. It can produce reliable segmentation masks for training the segmentation model. For semi-supervised learning, Hong et al. (2015) propose a method to separately train the classification and segmentation networks, and then pass the information from the classification network to the segmentation network to reduce the search space for effective segmentation. Moreover, Luc et al. (2016) employ an adversarial network to enhance the segmentation quality. Hung et al. (2018) propose a self-taught learning approach based on the adversarial network.

Inspired by Hung et al. (2018), we propose an auxiliary network, which can determine the credibility of each pixel on the segmentation map of unlabeled images and the trusted pixels are used as ground truth in training. In designing the auxiliary network, we propose an auxiliary loss function containing two vital terms to address the small objects of segmentation map; we believe that the supervision signals of small objects are more important than the bigger one.

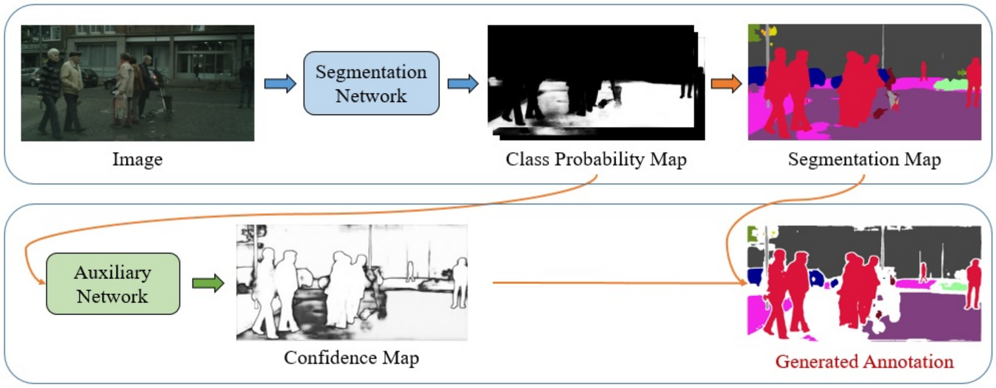

Figure 1: The process of generating annotations for unlabeled image.

3 METHODOLOGY

In this section, we first describe the general framework of our semi-supervised semantic segmentation scheme, and then we describe the architecture as well as design concepts of the proposed segmentation network and auxiliary network.

3.1 SEMI-SUPERVISED LEARNING USING AUXILIARY NETWORK

There are in total three training stages in our semi-supervised learning scheme. In the first stage, we simply train the segmentation network using labeled images. In this paper, we only use the typical cross entropy loss in the supervised training of segmentation network. Then in the second stage, an auxiliary network is trained using the proposed auxiliary loss function. Both the originally labeled images and the previously trained segmentation network are used in this stage. Our auxiliary network takes the class probability maps of size $H \times W \times C$ produced by the segmentation network as input, where H and W are the height and width of the image size and C denotes the number of classes. The auxiliary network outputs a confidence map of size $H \times W \times 1$. Each pixel of the confidence map is a probability value between 0 and 1, which represents the credibility of that pixel on the segmentation map. In the third stage, we feed the unlabeled images into the segmentation network pre-trained in the first stage to obtain the class probability map, and then pass this map through the auxiliary network to obtain the confidence map. Based on the probability value of each pixel of the confidence map, we can determine the segmentation reliability of each pixel on the segmentation map. Then, we select the pixels with high reliability as the annotated data points. And we ignore the pixels with lower reliability in the next-phase training. We set a threshold T_{aux} on the confidence map to separate the high reliability pixels from the low reliability pixels. Hence, we can mask the error-prone pixels on the output segmentation map and produce the annotated maps for the unlabeled images. Finally, since all images now have labels, we retrain the segmentation network using all of them, and produce a more accurate segmentation model. Figure 1 shows the process for generating annotated labels for images without ground truth.

Loss function for auxiliary network To train the auxiliary network, we create another auxiliary ground truth map of dimension $H \times W \times 1$. Assume that we have the original labeled ground truth class map and the estimated (segmented) class map produced by the segmentation network. The pixel value on auxiliary ground truth map is set to 1 if the class at that pixel on the estimated segmentation map matches that on the ground truth class map; and it is set to 0, if not. Different from the typical binary cross entropy loss, the loss function for auxiliary network is defined as:

$$L_{aux} = -\sum_{h,w} W^n_{h,w}(\gamma^n_{h,w} \times y^n_{h,w} \log(a_{h,w}(x^n)) + (1 - y^n_{h,w}) \log(1 - a_{h,w}(x^n))) \quad (1)$$

where $y^n_{h,w}$ equals to 1 if the estimate class on the segmentation map at pixel (h, w) is identical to the class value on the ground truth label map at the same pixel, and $y^n_{h,w}$ equals to 0 if they are different. In equation (1), $a_{h,w}(x^n)$ denotes the confidence map output at pixel (h, w), when the auxiliary network takes the class probability map x^n as input. $W^n_{h,w}$ represents the class weighting value at pixel (h, w) on the confidence map, and it is defined as $1/\ln(c + p_{class})$, inspired by ENet (Paszke et al., 2016), where we set c equal to 1.04. The class weighting at different locations of confidence map is decided by the ground truth label map. For example, when the class at the pixel (h, w) on ground truth label map is the class *i*, the class weight at that pixel on the confidence map is the weighting value associated with class *i*. Class weighting term is used to tackle the data imbalance issue. In other words, some classes having few number of pixels are often unfavored in training. With class weighting, our auxiliary network performs well not only on the big objects but also on small objects. Moreover, for different classes, the issue of imbalance between positive and negative samples are quite different. Thus, we propose a weight term $\gamma^n_{h,w}$, which is defined as $\#negative\,pixel_{class}/\#positive\,pixel_{class}$, and it is calculated using the auxiliary network ground truth. Similarly, this weight term is location variant depending on the ground truth label map. Both $W^n_{h,w}$ and $\gamma^n_{h,w}$ are the crucial terms in the proposed auxiliary loss function.

3.2 NETWORK ARCHITECTURE

3.2.1 SEGMENTATION NETWORK

We adopt the DSNet-fast (Chen et al., 2019) as our backbone model, but we make some modifications. From our observations, once the receptive field of the network is large enough to cover the entire image, make good use of the shallow-layer features is more effective than increasing the network depth for the semantic segmentation task. So, we redesign the initial block and the early block of DSNet-fast, and reduce layers in the deeper block to achieve a better accuracy and inference speed. The overall architecture is shown in Figure 2.

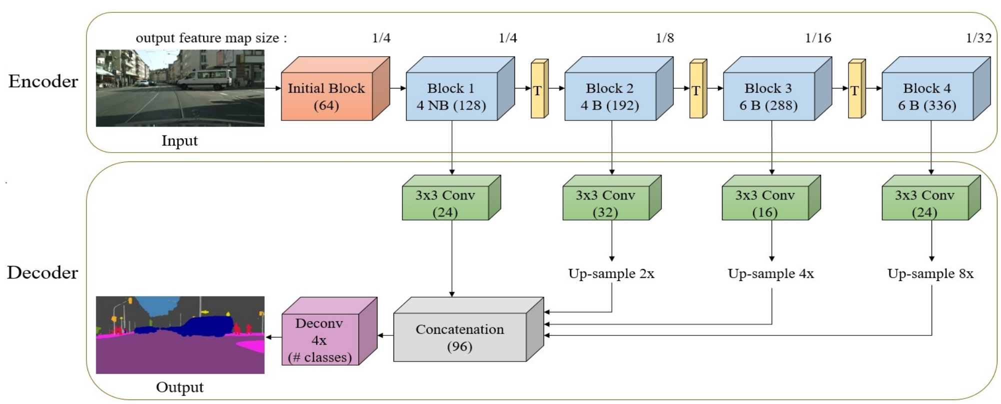

Figure 2: The architecture of the proposed segmentation network. “NB”: non-bottleneck unit. “B”: bottleneck unit. “T”: transition layer. The number of output channels for each block are marked inside parentheses.

In the encoder part, we adopt the same bottleneck unit and transition layer from DSNet-fast, and propose new initial block and non-bottleneck unit. All our core units are shown in Figure 3. In initial block, after applying 3×3 convolution with stride 2 on the input image, we divide it into two branches. The 1×1 convolution is applied in left branch to integrate features and reduce the number of channels, and on the right branch, a stack of two 3×3 convolutions is applied to learn the features. Furthermore, the feature maps of these two branches are concatenated and followed by an average pooling with stride 2 to produce the output. Since 1×1 convolution does not take the spatial information into account, the feature maps concatenated from the two branches have different receptive fields, and thus it helps the network to extract features at different scales. On the other hand, due to dimension reduction made by the 1×1 convolution operation, the left branch helps the right branch to produce more feature channels under the constraint of fixed number of concatenated feature maps. Similarly, we use the same concept to design our non-bottleneck unit. We find that this design strategy is indeed helpful for producing better segmentation results at similar model complexity.

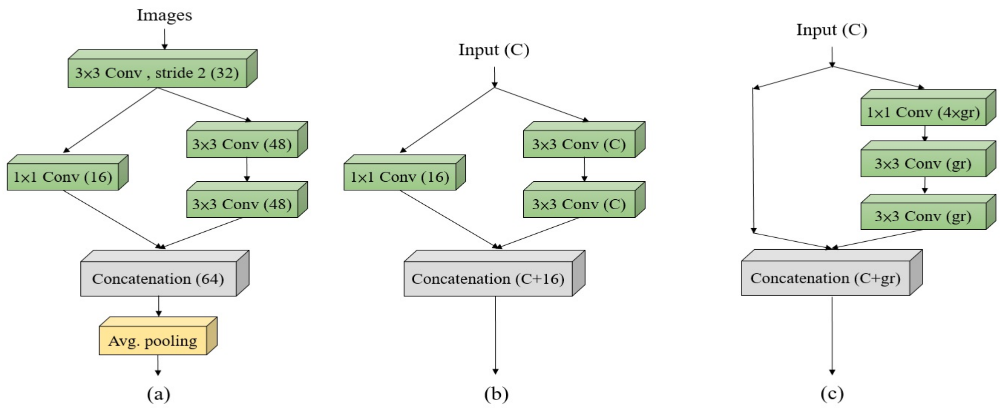

Figure 3: Proposed core units. (a) initial block. (b) non-bottleneck unit. (c) bottleneck unit. “C”: the number of channels from input. “gr”: growth rate, which is set to 32 in all our bottleneck unit. The number of channels for each layer are marked inside parentheses.

In the decoder part, we employ 3×3 convolution with {24, 32, 16, 24} channels after the Block1-4, respectively, to reduce the computational complexity. Furthermore, the up-sampling and concatenation operations are adopted, same as DSNet-fast, to integrate the feature maps at different levels. Finally, the concatenated feature map is up-sampled by a factor of 4 using the deconvolution kernel to produce the estimated segmentation map. All the convolution layers are followed by batch normalization (Ioffe & Szegedy, 2015) and ReLU activations.

3.2.2 AUXILIARY NETWORK

Our auxiliary network mainly consists of five convolution layers, one atrous spatial pyramid pooling (ASPP) module, and one deconvolution layer. The overall architecture is shown in Table 1.

Table 1: Auxiliary network architecture.

Input size	Block	Output size
512×1024×3	3×3 Conv , stride 2	256×512×32
256×512×32	3×3 Conv , stride 1	256×512×64
256×512×64	3×3 Conv , stride 2	128×256×128
128×256×128	3×3 Conv , stride 1	128×256×256
128×256×256	3×3 Conv , stride 1	128×256×256
128×256×256	ASPP	128×256×160
128×256×160	8×8 Deconv , 4x	512×1024×1

Since we find that detailed spatial information is vital in the auxiliary network, our feature map size is only 1/4 of the original image size before the deconvolution layer. We use the 3×3 convolution layers at the beginning, and then it is followed by a single atrous spatial pyramid pooling module similar to DeepLab v3+ (Chen et al., 2018) to extract multi-scale information as well as enlarging the receptive field. For the ASPP (Atrous Spatial Pyramid Pooling) module of our auxiliary network, we use dilated convolution with dilated rate {1, 2, 4, 8, 16} for branch 1-5 respectively, and do not use the image pooling branch. Each branch in our ASPP module has 32 feature channels. Finally, the deconvolution kernel is applied after ASPP module to rescale the output to the input size and generate dense confidence map, moreover, it is followed by sigmoid function to limit output between 0 and 1. Same as segmentation network, All the convolution layers are followed by batch normalization (Ioffe & Szegedy, 2015) and ReLU activations.

4 EXPERIMENTAL RESULTS

In this section, we first introduce two datasets, Cityscapes (Cordts et al., 2016) and CamVid (Brostow et al., 2008), that are used to evaluate proposed method. Both are popular road scene semantic segmentation datasets. Then, we describe our training details for the segmentation and the auxiliary network. At the end, we show the comparisons with the state-of-the-art methods as well as some visual results.

4.1 DATASETS

Cityscapes The Cityscapes dataset is a road scene dataset with the image resolution of 1024 $\times$2048, and it provides 19 object classes for evaluation. There are two forms of annotation data in its database, namely, fine and coarse. Only the fine annotation set is used in our experiments, where it contains 2975, 500, 1525 images for training, validation, and testing respectively. We resize the image to 512$\times$1024 for training and testing due to the hardware limitation; however, for evaluation on the testing set, we restore the resolution of segmentation results to the original size for fair comparison.

CamVid The CamVid dataset is also a road scene dataset, but its image resolution is much smaller than the Cityscapes dataset, which is about 360$\times$480. It provides 11 semantic classes, and contains 367, 101, 233 images for training, validation, and testing.

4.2 IMPLEMENTATION DETAILS

We use PyTorch framework to implement our method, and we measure the network speed on a single GTX 1080Ti GPU. The popular metric, mean IoU, is used to compute the segmentation performance. In addition, the encoder part of our segmentation network has been pre-trained on ImageNet (Deng et al., 2009) to produce better initial parameters.

Segmentation network To train the segmentation network for the Cityscapes dataset, we use the stochastic gradient decent (SGD) optimization with momentum 0.9, weight decay 0.0001, and batch size 4. We adopt the poly learning rate policy as described in Chen et al. (2017a) with power 0.9 in all our experiments and the initial learning rate is set to 0.05 here. We train it for 200 epochs in total. For the CamVid dataset, the SGD optimization is also used but with momentum 0.9, weight decay 0.0005, and batch size 8. In addition, the initial learning rate is set to 0.075, and we train it for 150 epochs in total.

For both Cityscapes and CamVid datasets, we adopt the class weighting scheme same as in Paszke et al. (2016) with an additional hyper-parameter c set to 1.1, and the data augmentation strategies are employed to boost the performance. Our data augmentation strategies include random horizontal flip, random pixel translation, and random scaling. For the Cityscapes dataset, the scaling factors in the random scaling technique are $\{0.75, 1.0, 1.25, 1.5, 1.75\}$, and for the CamVid dataset, they are $\{0.8, 1.0, 1.3, 1.7\}$.

Auxiliary network To train the auxiliary network for Cityscapes dataset, we use the Adam optimization (Kingma & Ba, 2014) with momentum 0.9, weight decay 0.0001, and batch size 4, and the initial learning rate is set to 0.0005. For the CamVid dataset, different from the former, the weight decay for Adam optimization is set to 0.0005, and batch size is set to 8.

Both datasets are training for 50 epochs, and we use the proposed auxiliary loss function in training the auxiliary network. The data augmentation strategies for training auxiliary network are same as described in the above.

4.3 PERFORMANCE EVALUATION

Evaluation on Cityscapes dataset In order to evaluate the performance of the proposed segmentation network, we test it on the Cityscapes testing set, and compare the results with the state-of-the-art networks. It is worth noticing that we do not adopt any testing technology such as multi-scale testing in the evaluation process. From Table 2, we find that our segmentation network can achieve

Table 2: Comparison with the state-of-the-art methods on the Cityscapes testing set. The efficiency-based methods are adopted for comparison. "†": inference speed using Titan X GPU. "‡": inference speed using Titan XP GPU. "††": inference speed using GTX 1080Ti GPU.

Method	Training dataset	Mean IoU (%)	Speed (FPS)	Params
ENet (Paszke et al., 2016)	Fine	58.3	76.9†	0.37 M
EDANet (Lo et al., 2018)	Fine	67.3	108.7††	0.68 M
Fast-SCNN (Poudel et al., 2019)	Fine, Coarse	68.0	123.5‡	1.11 M
BiSeNet (Yu et al., 2018)	Fine	68.4	105.8‡	5.8 M
DSNet-fast (Chen et al., 2019)	Fine	69.1	68.0††	3.0 M
ICNet (Zhao et al., 2018)	Fine	69.5	30.3†	6.68 M
ERFNet (Romera et al., 2017)	Fine	69.7	41.7†	2.1 M
DF1-Seg-d8 (Li et al., 2019)	Fine	71.4	136.9††	-
DF2-Seg2 (Li et al., 2019)	Fine	75.3	56.3††	-
Ours	Fine	73.9	73.2††	2.11 M

Table 3: The measuring results for confidence map on the Cityscapes validation set using half of training data.

T_{aux}	Mean IoU (%)	Average class semi ratio (%)	Selected pixels (%)
0	70.3	100	100
0.7	83.7	75.6	86.1
0.8	85.6	71.2	83.3
0.9	88.2	64.2	78.0

a good trade-off between speed and accuracy. It achieves 73.9% mean IoU with fewer parameters and higher inference speed comparing to the other segmentation nets.

Moreover, to evaluate the proposed semi-supervised learning scheme, we randomly sample half of images from the training set as the labeled data similar to that in Hung et al. (2018), and then take the other half as the unlabeled data. We conduct the experiments on Cityscapes validation set. First of all, we train the segmentation and auxiliary network using the labeled images as described in Section 3.1, and then we measure the confidence map performance. In taking the measure, we feed the validation set images to our entire network, and then mask the error-prone pixels of the segmentation map by the confidence map. Then, compare the masked segmentation map (generated annotations) with its true ground truth to calculate the accuracy. We use three performance metrics in total for evaluation, and the experimental results are shown in Table 3, where T_{aux} represents the threshold set for the confidence map. The mean IoU metric is the mIoU calculated by the generated annotations and the truth ground truth without considering the masked pixels (by the confidence map). The *average class semi ratio* is the average value of the "class semi ratio" of all classes. To compute the class semi ratio for class *i*, we only consider the region of pixels that belong to class *i* on the ground truth label map. The denominator of class semi ratio for class *i* is the number of all pixels; and the numerator is the number of pixels, not masked by the confidence map, on the estimated segmentation map. In calculating the *selected pixels*, we first count the pixels of the estimated segmentation map that are selected by the confidence map and used as ground truth in training. Then, we compute its ratio with respect to the entire image. From Table 3, we can find that with a higher threshold, the pixels selected as GT are more accurate, but the selected pixels are fewer; this is a trade-off. In addition, observing from the average class semi ratio and the individual semi ratio for each class, our confidence map can also produce good results on the small objects with a higher threshold. Thus, the generated annotations used for retraining segmentation network can achieve good quality.

Table 4: The proposed semi-supervised learning segmentation results on the Cityscapes validation set based on different threshold values on the confidence map.

Data amount			
Labeled data	Unlabeled data	T_{aux}	Mean IoU (%)
1/2	1/2	0	69.9
1/2	1/2	0.7	71.3
1/2	1/2	0.8	71.4
1/2	1/2	0.9	71.5
1/2	1/2	1	70.3

Table 5: Comparison with Hung et al. (2018) on the Cityscapes validation set.

	Data amount (labeled data + unlabeled data)			
Method	1/8 + 7/8	1/4 + 3/4	1/2 + 1/2	1 + 0
(Hung et al., 2018) baseline	55.5	59.9	64.1	66.4
(Hung et al., 2018) + adversarial learning	57.1	61.8	64.6	67.7
(Hung et al., 2018) + adversarial learning + semi	58.8	62.3	65.7	-
Ours baseline	56.8	63.0	66.1	69.6
Ours + semi	60.7	65.5	67.7	-
Ours baseline (proposed network)	57.7	65.5	70.3	74.4
Ours + semi (proposed network)	60.2	68.4	71.5	-

After generating the annotated labels for 1/2 unlabeled images, we retrain the segmentation network using all images. The experimental results of selecting pixels based on different threshold values are shown in Table 4. The proposed semi-supervised learning scheme can improve the performance for about 1.0% to 1.2%. If T_{aux} is set to 0, it means that all the pixels of the estimated segmentation map are used as ground truth for unlabeled image, which is obviously unreasonable and it decreases the accuracy. Since T_{aux} at 0.9 gives the best results, we use the same value in the following experiments.

To verify the effectiveness of our approach, we conduct experiments with different amount of data, and shows the comparison with Hung et al. (2018) in Table 5. For the fair comparison, we also adopt the same segmentation network used by Hung et al. (2018) as our backbone model. Under the same settings that using 1/2, 1/4, and 1/8 training images as labeled data and using the rest of images as unlabeled data, our semi-supervised learning method can lead to higher performance improvements. It can improve the mean IoU by 3.9%, 2.5%, and 1.6% for 1/8, 1/4, and 1/2 training datasets, respectively. Particularly, if we focus on the performance differences between the schemes with and without semi-supervised learning mechanism, our semi-supervised learning approach offers quite significant improvement.Then, if our proposed segmentation network is used as the backbone network, it increases the mean IoU by another 1.2% to 2.9%. In the end, we show some sample results in Figure 4.

Evaluation on CamVid dataset We also evaluate our approach on the CamVid dataset. As the results shown in Table 6, our semi-supervised learning method can boost the performance of the segmentation network, and it can achieve 1.3% improvement in mean IoU.

Table 6: Comparisons on the CamVid testing set.

	Data amount		
Method	Labeled data	Unlabeled data	Mean IoU (%)
Ours baseline (proposed network)	1	0	71.8
Ours baseline (proposed network)	1/2	1/2	69.3
Ours + semi (proposed network)	1/2	1/2	70.6

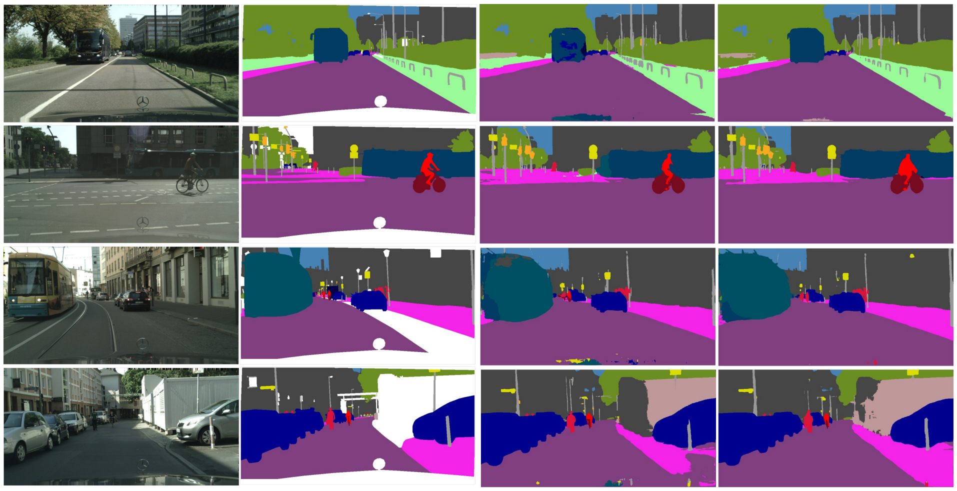

Figure 4: Sample results on the Cityscapes validation set using half of training data. From left to right: (a) image (b) ground truth (c) segmentation results without semi-supervised learning (d) segmentation results with semi-supervised learning

5 CONCLUSIONS

In this paper, we first propose a highly efficient segmentation network as our platform, and then we design a semi-supervised learning scheme using an auxiliary network. The auxiliary network is used to verify the estimated segmentation map and to generate annotations on the unlabeled images. Equipped with the carefully designed auxiliary loss function in training, the auxiliary network performs well not only on the large objects but also on the small objects. It shows that the unlabeled images together with the generated annotations (labels) can be used to retrain the segmentation network for better segmentation quality. Our experimental results on the Cityscapes and CamVid datasets demonstrate the effectiveness of the proposed method.

REFERENCES

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pp. 549–565. Springer, 2016.

Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *European conference on computer vision*, pp. 44–57. Springer, 2008.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4): 834–848, 2017a.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017b.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

Ping-Rong Chen, Hsueh-Ming Hang, Sheng-Wei Chan, and Jing-Jhih Lin. Dsnet: An efficient cnn for road scene segmentation. *arXiv preprint arXiv:1904.05022*, 2019.

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. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1635–1643, 2015.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Seunghoon Hong, Hyeonwoo Noh, and Bohyung Han. Decoupled deep neural network for semi-supervised semantic segmentation. In *Advances in neural information processing systems*, pp. 1495–1503, 2015.

Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. *arXiv preprint arXiv:1802.07934*, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Xin Li, Yiming Zhou, Zheng Pan, and Jiashi Feng. Partial order pruning: for best speed/accuracy trade-off in neural architecture search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9145–9153, 2019.

Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3159–3167, 2016.

Shao-Yuan Lo, Hsueh-Ming Hang, Sheng-Wei Chan, and Jing-Jhih Lin. Efficient dense modules of asymmetric convolution for real-time semantic segmentation. *arXiv preprint arXiv:1809.06323*, 2018.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, 2016.

G Papandreou, L-Ch Chen, K Murphy, and AL Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. *arXiv preprint arXiv:1502.02734*, 2015.

Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

Deepak Pathak, Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional multi-class multiple instance learning. *arXiv preprint arXiv:1412.7144*, 2014.

Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proceedings of the IEEE international conference on computer vision*, pp. 1796–1804, 2015.

Rudra PK Poudel, Ujwal Bonde, Stephan Liwicki, and Christopher Zach. Contextnet: Exploring context and detail for semantic segmentation in real-time. *arXiv preprint arXiv:1805.04554*, 2018.

Rudra PK Poudel, Stephan Liwicki, and Roberto Cipolla. Fast-scnn: fast semantic segmentation network. *arXiv preprint arXiv:1902.04502*, 2019.

Xiaojuan Qi, Zhengzhe Liu, Jianping Shi, Hengshuang Zhao, and Jiaya Jia. Augmented feedback in semantic segmentation under image level supervision. In *European Conference on Computer Vision*, pp. 90–105. Springer, 2016.

Martin Rajchl, Matthew CH Lee, Ozan Oktay, Konstantinos Kamnitsas, Jonathan Passerat-Palmbach, Wenjia Bai, Mellisa Damodaram, Mary A Rutherford, Joseph V Hajnal, Bernhard Kainz, et al. Deepcut: Object segmentation from bounding box annotations using convolutional neural networks. *IEEE transactions on medical imaging*, 36(2):674–683, 2016.

Eduardo Romera, José M Alvarez, Luis M Bergasa, and Roberto Arroyo. Efficient convnet for real-time semantic segmentation. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1789–1794. IEEE, 2017.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S Huang. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7268–7277, 2018.

Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 325–341, 2018.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 405–420, 2018.