Real-time Domain Adaptation in Semantic Segmentation

Xiangbo Gong
Politecnico di Torino
s295754
s295754@studenti.polito.it

Shaoyong Guo
Politecnico di Torino
s296966
s296966@studenti.polito.it

Yuting Wang
Politecnico di Torino
s296226
s296226@studenti.polito.it

***Abstract*—Real-time semantic segmentation techniques are applied for various image classification tasks due to their excellent performance in pixel-level semantic classification. However, implementing a semantic segmentation model is time consuming and requires a very large number of manually labeled semantic annotation. In this paper, we apply 3 steps separately to enhance the efficiency of the semantic segmentation model by improving the architecture of the CNN network and applying adversarial domain adaptation and image-to-image transformation algorithms. First of all, a bilateral segmentation network is used to implement semantic classification of the cityscape dataset. Subsequently, integrating adversarial domain adaptation and deepthwise separable convolution yields a lightweight network, which reduce the reliance on computational resources and manually semantic annotations. Finally, an image-to-image transformation algorithm converts the transformed LAB images from the source domain to the target domain style for improving the efficiency of the model.**

***Index Terms*—Bilateral Segmentation Network, Adversarial Domain Adaptation, Depthwise Separable Convolution,Image-to-image Transformation**

I. Introduction

The project aims to learn the domain adaptation theory in semantic segmentation and try to implement real-time semantic segmentation networks on the benchmarks of Cityscapes and GTA5 datasets. The whole implementation for real-time semantic segmentation is divided into 3 steps. In step 1, we implemented the Bilateral Segmentation Network (BiSeNet), using ResNet-101 and ResNet-18 as backbone networks, respectively. We then evaluated the proposed BiSeNet on Cityscapes in terms of pixel accuracy and mIoU metrics. In step 2, we implemented and modifed an Adversarial Domain Adaptation algorithm. In this step, the GTA5 dataset is used as the source domain and the Cityscapes dataset is used as the target dataset to train a new BiSeNet model. In the last step, we applied a image translation algorithm to train new BiSeNet model. The Image-to-image translation approach can be utilized to improve the performance of domain adaptation.

II. Related work

A. Semantic Segmentation

Semantic segmentation is a classification technique at the pixel level of an image, rather than the usual CNN classification technique. It could provide rich category information at the pixel level, which can enable deep convolutional networks to grasp spatial positions [1] . During execution, semantic segmentation will assign a predefined category label to each pixel [2].

B. Multi-CNN nets

- ResNet: When it raises the number of layers of convolutional network in the experiment, there is an obvious phenomenon, vanishing gradient or exploding gradient. A ResNet network with residual learning framework can gain accuracy from considerably increased depth while it solves vanishing gradient and exploding gradient [3].
- BiseNet: Semantic segmentation needs both rich spatial information and sizeable receptive field. Bilateral Segmentation Network (BiSeNet) could balance spatial resolution while keeping the speed of computing [4]. The network includes two paths, i.e., Spatial Path and Context Path, which can obtain the rich spatial information and sufficient receptive field.

C. Unsupervised adversarial domain adaptation

- Domain adaptation: Deep convolutional networks require Large-scale labeled datasets to train an effect model for different kinds of vision tasks. However, in many applications, large quantities of raw datasets are unlabeled. For real-time applications, manually labeled datasets are inefficient and time-consuming. Domain adaptation (DA) is an unsupervised transfer learning strategy, which aims to learn a model from a source domain that can perform well on another target domain [5].
- Generative adversarial nets: Generative adversarial nets(GAN) are effective methods to address vision tasks. It has developed many variations e.g., CycleGAN, StarGAN. The GAN network has two main models, a generator G and a discriminator D. This framework corresponds

to a minimax two-player game [6]. The generator G captures the data distribution to generate new dataset. And then the discriminator D estimates the probability that a sample is from the training data rather than G.

- Adversarial domain adaptation: Although domain adaptation has been widely applied to many image classification tasks, there are still many challenges in feature adaptation-based approaches to semantic segmentation [7]. In the semantic segmentation task, adversarial domain adaptation algorithm introduces the task of domain adaptation on semantic segmentation by applying adversarial learning. Using only a small amount of computational resources and training time, it can achieve significant performance in pixel-level tasks [8].
- Image-to-image transformation: Image-to-image translation (I2I) is the process of acquiring images from the source domain and converting them to have the style or features of images in the target domain. More precisely, its purpose is to preserve the inner source content of the images and replace the outer target style. [9] For example, Grayscale images can be converted to color, and daytime images can be converted to night and so on. I2I plays a vital role in many computer vision, computer graphics, and image processing problems [10].

III. Proposed Approach

A. Step2–Implement and testing real-time semantic segmentation network

- **Setting**

  Real-time semantic segmentation requires both rich spatial information and sizeable receptive field. Bilateral segmentation network (BiSeNet) can obtain segmented spatial information while maintaining efficient computational speed. The proposed architecture has two components: Spatial Path (SP) and Context Path (CP). In the SP component, it consists of 3 convolutional layers, each with a convolution model, a batch normalization model and a ReLU model. In the CP component, it uses the global average pooling and applies ResNet-101 and ResNet-18 as the backbone of the network, which can quickly downsample the input feature maps and obtain large receptive field.
- **Training**

  The Spatial Path encodes rich spatial information, while the Context Path use a lightweight model to provides large receptive field. The Spatial Path has 3 convolutional layers, each layer uses a 3X3 convolution filter to downsample the input feature. This path extracts an output feature map with only 1/8th of the original image. The Context Path model uses a ResNet-101 and ResNet-18 model as the backbone of network for fast feature downsampling, respectively. The Context Path also employs a specific Attention Refinement Module (ARM) to refine the feature vector at each stage. ARM can capture global context of the output feature and compute an attention vector to guide the feature learning, which can integrate

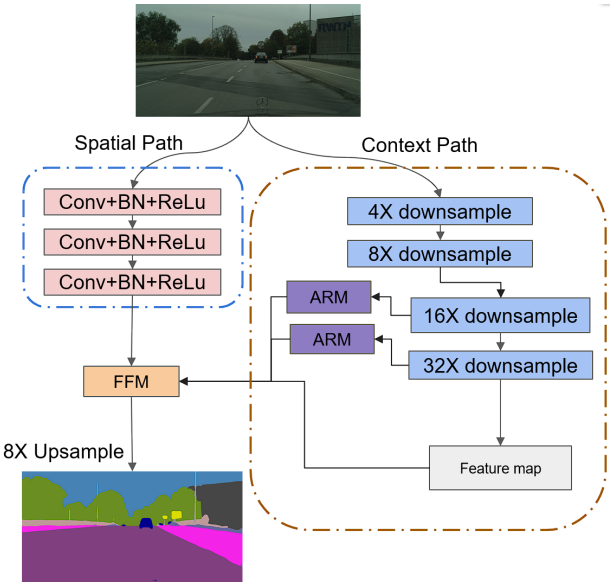

Fig. 1. An overview of the Bilateral Segmentation Network [4].

the global context information easily without any upsampling operation.

The output feature vectors of spatial paths are low-level, while the output features of contextual paths are high-level. Therefore, a specific feature fusion module (FFM) is proposed to fuse these features. Finally, a batch normalization method is used to obtain an output feature vector and a weight vector. The structure of BiSeNet is shown in Fig. 1.

In the BiSeNet, it utilizes stochastic gradient descent (SGD) with a batch size of 16 as the optimizer and adds an auxiliary loss function for optimization. It uses the parameter α to adjust the weights of the primary and auxiliary losses.As "(1)"and "(2)" shows:

$$L(X;W) = l_p(X;W) + \alpha \sum_{i=2}^{k} l_i(X_i;W) \tag{1}$$

$$loss = \frac{1}{N} \sum_{i} -log(\frac{e^{p_i}}{\sum_j e^{p_j}}) \tag{2}$$

where L is the joint loss function SGD loss and auxiliary loss. The l_p represents the SGD loss of the output feature, l_i represents the auxiliary loss of the output feature. X_i is the output feature of the i-th stage of the ResNet model. In this proposed model, K is fixed to 3.

B. Step3–Implementing unsupervised adversarial domain adaptation and making the framework lightweight

- **Setting**

  In the adversarial domain adaptation algorithm, it introduces the GAN architecture on the domain adaptation task of semantic segmentation. The architecture of adaptive domain adaptation retains the Discriminator module

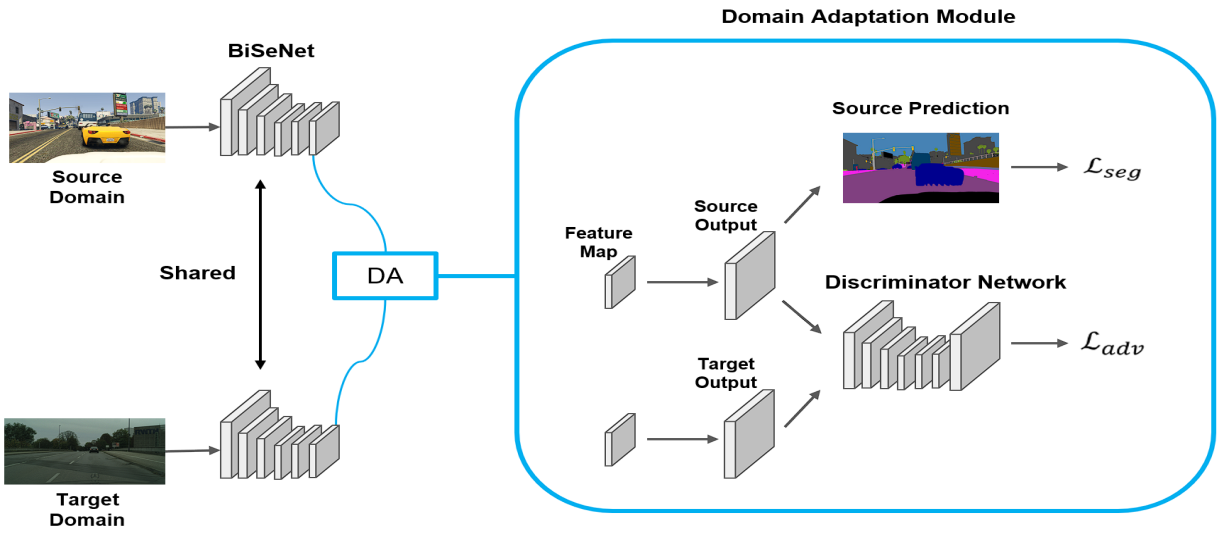

Fig. 2. An overview of Adversarial Domain Adaptation Network [8].

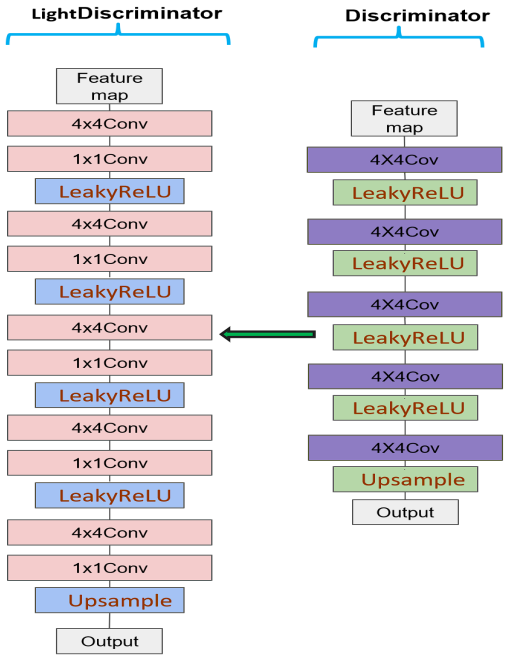

Fig. 3. Illustration of different discriminators.

of the GAN network and replaces the Generator module by the source and target domains. The whole network is divided into two modules: the segmentation network G and the discriminator D.

In the project, we adopt BiSeNet network as segmentation network G. The images in the target domain lack annotation information compared to the images in the source domain. However, they have a similar distribution in semantic segmentation.

The source image is first introduced into the segmentation network to optimize the G and predicts the segmentation result P_s by softmax founction. Then the optimized G predicts the segmentation result P_t for the target images. With P_s and P_t as input features, a discriminator D is trained to distinguish whether the input is from the source or target domain. The structure of our network is shown in Fig 2.

- **Training**

With an adversarial loss on the target prediction, the network propagates gradients from D to G, which would encourage G to generate similar segmentation distributions in the target domain to the source prediction.

The Discriminator can be trained by cross-entropy loss functions "(3)". The segmentation softmax output P = G(I) $\in R^{H*W*C}$ is forwarded to the discriminator D, where C is the number of categories. In the cross-entropy loss L_d, z=0 indicates that the sample is from the target domain, while z=1 indicates that the sample is from the source domain.

The Segmentation Network be trained by loss functions "(4)" and "(5)". For images from the source domain, cross-entropy loss is defined by functions "(4)". Y_s represents the ground truth annotations for source images and P_s = G(I_s) represents the segmentation output. It yields the P_t = G(I_t) when images are from target domain.Finally, discriminators D employ the adversarial loss L_{adv} "(5)" to make the distribution of P_t closer to P_s.

$$L_d(P) = -\sum_{h,w}((1-Z)\log(D(P)^{h,w,0}) + Z\log(D(P)^{h,w,0})) \tag{3}$$

$$L_{seg}(I_s) = -\sum_{h,w}\sum_{c\in C} Y_s^{(h,w,c)}\log(P_s^{(h.w.c)}) \tag{4}$$

$$L_{adv}(I_t) = -\sum_{h,w} \log(D(P_t)^{(h.w.1)}) \tag{5}$$

- **Lightweight adversarial domain adaptation**
  The common architecture of Fully Convolutional Discriminator D has 5 convolutional layers with kernel size of 4×4 with channel numbers $\{64, 128, 256, 512, 1\}$, stride of 2, and padding of 1. Each convolution layer is followed by a Leaky-ReLU parameterized by 0.2 except the last layer. Finally, we add an up-sampling layer to the last layer to rescale the output to the size of the input map, in order to match the size of local alignment score map. For our Lightweight Discriminator LD, it replaces all the full convolutions with Depthwise Separable Convolutions, which can make the network more lightweight. LD is obtained by replacing each typical full convolution with a depthwise separable convolution, and each convolution layer contains a 4x4 depthwise Conv, a 1x1 pointwise Conv. The structures of D and LD are shown in Fig. 3. This lightweight discriminator significantly improves the efficiency by reducing the computational parameters for real-time semantic segmentation.

C. Step4–Image-to-image translation to improve domain adaptation: LAB

- **Setting**
  In this step, we intend to further improve the performance of the overall domain adaptation in step3. We are aware that changes in the visual appearance of images can degrade the performance of a visual recognition system. We observe that the difference in appearance between our source and target images is mainly a difference in colour style by studying at [11] and [12], we applied a straightforward yet effective color style transfer that can reduce this difference to some extent. As the LAB color space has a greater color range than the RGB color space, the image style converted on the LAB color space is more similar to the target domain than if the image style was altered directly on the RGB color space. So, the image translation in this paper is based on the LAB color space. The flow of the color style transfer algorithm for the source domain images is shown in Fig. 4, with details shown below.
  An RGB image X_S^{RGB} is first read from the source domain and then converted to the LAB color space, resulting in a LAB image X_S^{LAB}. Next, we calculate the mean μ_S and standard deviation σ_S for each channel of the resulting LAB image X_S^{LAB}. At the same time, a random RGB image X_T^{RGB} is read from the target domain and processed in the same way as the source image. We obtain the mean μ_T and standard deviation value σ_T of the converted target image. Finally, by shifting the distribution of pixel values to the image in the target domain, we transform the converted LAB image from the source domain to the target style, *ie*.,

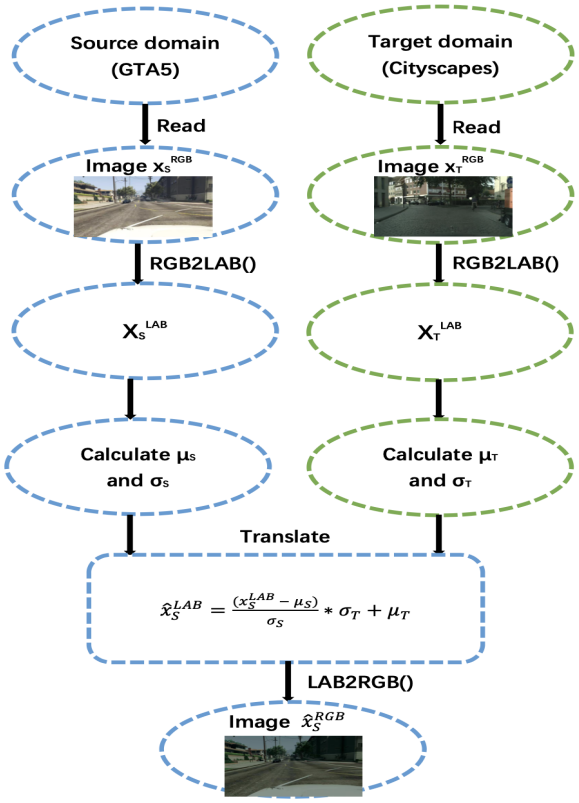

Fig. 4. The flow of the color style transfer algorithm.

$$\hat{X}_S^{LAB} = \frac{(X_S^{LAB} - \mu_S)}{\sigma_S} * \sigma_T + \mu_T \tag{6}$$

After aligning the distribution, we then convert the translated LAB image $\hat{X}_S^{LAB}$ back to the RGB color space $\hat{X}_S^{RGB}$.

- **Training**
  In the training process we use the same architecture as in step 3, the only difference is that the images in the source domain are transformed during the loading process.

IV. Experiment Results

A. Datasets

In the experiment, the Cityscapes dataset and the GTA5 dataset were used to train the BiSeNet model, the domain adaptation model, the adversarial domain adaptation model, and the LAB-based adversarial domain adaptation model, respectively. Cityscapes is a dataset of urban scenes from the streets of 50 different cities, which consists of a large and diverse set of stereoscopic video sequence recordings [8]. The original Cityscapes have 5000 images with high quality pixel-level annotations and 20000 additional images with coarse annotations. However, in our experiment, we just leverage a subset of Cityscapes dataset that has 19 semantic categories and 750 images (500 for the training set and 250 for the validation set).

TABLE I
COMPARISON OF MIOU FOR EACH SEMANTIC CATEGORY ON THE ADVERSARIAL DOMAIN ADAPTATION MODEL AND THE LAB-BASED ADVERSARIAL DOMAIN ADAPTATION MODEL.

Experiment	Road	Sidewalk	Building	Wall	Fence	Pole	TLight	TSign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle
ADA	58.44	27.70	58.24	14.43	16.28	31.22	24.64	10.76	81.09	19.81	62.89	51.59	5.66	66.65	8.43	7.24	0.01	13.14	0.0
lightweight ADA	75.08	30.13	73.11	7.47	14.31	31.48	20.04	8.96	81.93	24.10	66.91	52.63	8.53	53.94	17.17	11.78	0.0	10.14	0.0
lightweight LAB-ADA	87.68	35.40	77.57	19.71	14.61	33.95	21.12	7.77	81.11	27.97	71.91	57.30	3.76	79.57	17.21	14.68	0.0	13.80	0.0

*ADA is Adversarial Domain Adaptation.

TABLE II
COMPARISON OF BISENET MODEL, ADVERSARIAL DOMAIN ADAPTATION MODEL AND LAB-BASED ADVERSARIAL DOMAIN ADAPTATION MODEL IN TERMS OF SEMANTIC CATEGORY EFFICIENCY.

Experiment	Accuracy (%)	mIoU (%)	Total Parameters	FLOPS
BiseNet-18	79.60	47.30	–	–
BiseNet-101	81.60	59.30	–	–
ADA	43.02	29.38	2.78M	15.47 GMac
lightweight ADA	43.99	30.93	0.19M	1.11 GMac
lightweight LAB-ADA	46.90	35.00	0.19M	1.11 GMac

*ADA is Adversarial Domain Adaptation.

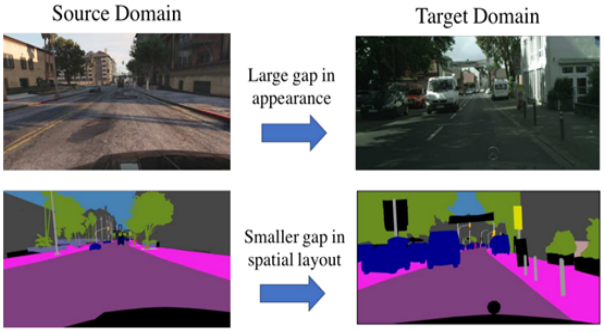

Fig. 5. Differences in appearance and semantic segmentation output between GTA5 and Cityscapes [8].

In image classification, the cost of creating large datasets with pixel-level labels is very high, since millions of labeled images are required to train deep models. Researchers like Hunt, G. have found that extracting pixel-accurate object labels from off-the-shelf games can be achieved by a technique known as detouring [13]. GTA5 datasets is extracted from the game Grand Theft Auto V, which has 25,000 pixel-level semantic segmentation images [9]. In the experiment, the GTA5 dataset can be used as the source domain and the Cityscapes dataset can be used as the target domain to train the domain adaptation model. There is a small difference between the GTA5 dataset and the Cityscapes dataset in terms of segmentation output, and a large difference in terms of appearance [10], as show in Fig. 6.

B. Implementation details

The segmentation model is trained with batch size 4 and SGD with an initial learning rate of 5×10^{-4}, which is then changed at each iteration with a "poly" learning rate decay with power 0.9, momentum 0.9, and weight decay 0.0005. Adam is used to train all of the discriminators, with momentum $(0.9, 0, 99)$, learning rate 1×10^{-5}, and the same segmentation model scheduler. The value of λ_{adv} is set to 0.01. We random horizontal flip and random scale on the training images to augment the dataset. The scales contains $\{0.5, 0.75, 1.0, 1.5\}$. Finally, we crop the training images to $(1024, 512)$, whereas the evaluation is done on the original image size which is $(2048, 1024)$. In Step 2 we set the number of epochs to 50. In step 3 and step 4 we set maximum iterations to 250k but early stop at 25k iterations. Per-pixel classification accuracy as well as the metric mean intersection-over-union (mIoU) are used to evaluate the performance of our proposed adaptation method in all the steps.

C. Results

In the step 2, we implemented and tested a real-time semantic segmentation network based on BiSeNet-18 and BiSeNet-101 framework, respectively. In this step, 750 images of the Cityscapes dataset were adopted and 19 semantic categories were evaluated for semantic category accuracy and mIoU. In the step 3 and step 4, we train images of target domain (Cityscapes dataset) with shared parameters from source domain (GTA5 datasets). In the Table II, it is obvious that sematic categories accuracy and mIoU value are better than model on the domain adaptation. The performance trained on domain adaptation is about half of what we would have obtained if we had trained only on the target domain. When using the same BiSeNet network on the Cityscapes dataset, the semantic category accuracy and mIoU values with ResNet-101

(a) Source Image (b) Trans. on LAB (c) Target Image

Fig. 6. The comparison of image translation on different color space.

as the framework were better than those with ResNet-18 as the framework, which had 81.6 % accuracy and 59.3 % mIoU.

Due to the drawbacks of manually annotated images, we substituted the method of training the model in the target domain with a domain adaptation algorithm. Table II shows that the performance gradually improves when the network is trained on adversarial domain adaptation, lightweight adversarial domain adaptation(Lightweight-ADA), and LAB-based adversarial domain adaptation, respectively. In the step 3, replacing typical fully convolutional discriminator with lightweight depthwise-separable convolutions to achieve a lightweight adversarial domain adaptation(lightweight- ADA) network. It can significantly reduce the computational sources, from 2.78M to 0.19M in the total parameters and from 15.47G to 1.11G in FLOPS, while improving the sematic category accuracy by 0.97% and mIoU value by 1.55%. In each semantic category, Lightweight-ADA has slightly higher mIoU than ADA. Especially, both Lightweight-ADA and ADA models have high mIoU results of over 60% on the items of road, building, vegetation, sky, people and cars sematic categories, while the results of mIoU are almost 0 for the items of train and bicycles, as shown in Table I. The possible reason is that several semantic categories in the target domain and source domain are not sufficient to train a plausible model to identify the semantic categories of trains and bicycles.

In the step 4, based on lightweight adversarial domain adaptation, we apply image translation based on LAB color space to further improve the performance of the network. This is because that the gamut of LAB color space is larger than RGB color space, and the image style converted on the LAB color space can achieve the goal of reducing domain differences. The transformed LAB image from the source domain has more similarity to the target image in the gamut color, as shown in Figure 6. After image translation on the lightweight adversarial domain adaptation, we obtained better network performance. With the same total parameters and Flops, the scheme improves semantic category accuracy by 2.91% and mIoU by 4.07% over lightweight ADA. The code and model are available at https://github.com/GongXiangbo/Real-time-Domain-Adaptation-in-Semantic-Segmentation.

V. Conclusion

The project set out to learn the semantic segmentation for images and domain adaptation in semantic segmentation. In step 2 of our experiment, the final performance is not as good as in the original paper, possibly because the number of images available for training is too small and not enough epochs are run. In step 3 of our experiment, we applied random horizontal flip and random scale on the training images. It may be because the random scale provides the network with detailed information of many pictures, which improves the robustness of the network and makes the experiment achieve good performance when the number of pictures in the training set is small. In step 4, based on step 3 we applied a simple image translation on different color space to reduce the gap between domains, it makes the experiment achieved better performance compare to step 3. Our experiment result shows that the image translation between domains is powerful, future research can focus on applying better image translation to get better performance. A problem in the experiment is that the training time is too long in both step 3 and step 4, future research can also focus on making the network lighter without compromising performance.

References

[1] Hao, S., Zhou, Y., and Guo, Y. (2020). A brief survey on semantic segmentation with deep learning. Neurocomputing, 406, 302-321.

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

[3] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

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

[5] Zhao, S., Yue, X., Zhang, S., Li, B., Zhao, H., Wu, B., ... and Keutzer, K. (2020). A review of single-source deep unsupervised visual domain adaptation. IEEE Transactions on Neural Networks and Learning Systems.

[6] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... and Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

[7] Tavera, A., Masone, C., and Caputo, B. (2021). Reimagine BiSeNet for Real-Time Domain Adaptation in Semantic Segmentation. arXiv preprint arXiv:2110.11662.

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

[9] Isola, P., Zhu, J. Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).

[10] Pang, Y., Lin, J., Qin, T., and Chen, Z. (2021). Image-to-image translation: Methods and applications. IEEE Transactions on Multimedia.

[11] He, J., Jia, X., Chen, S., and Liu, J. (2021). Multi-source domain adaptation with collaborative learning for semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11008-11017).

[12] Reinhard, E., Adhikhmin, M., Gooch, B., and Shirley, P. (2001). Color transfer between images. IEEE Computer graphics and applications, 21(5), 34-41.

[13] Hunt, G., Brubacher, D.: Detours: Binary interception of Win32 functions. In: 3rd USENIX Windows NT Symposium (1999)