Multi-Domain Semantic-Segmentation using Multi-Head Model

Shota Masaki[1],Tsubasa Hirakawa[1], Takayoshi Yamashita[1] and Hironobu Fujiyoshi[1]

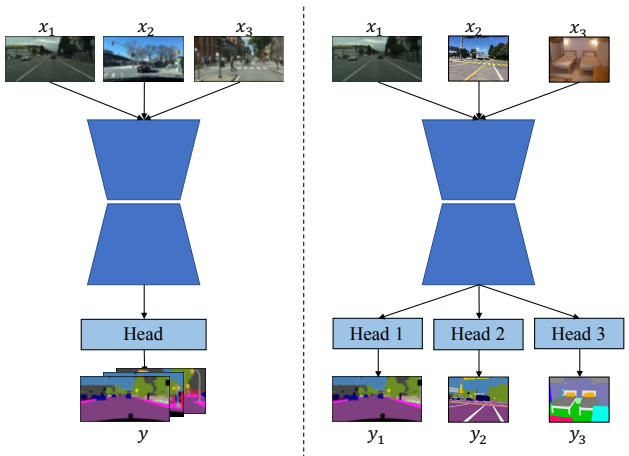
***Abstract*— Semantic segmentation is a pixel-wise class identification problem, which is important for automatic driving support such as recognizing the driving area. However, segmentation accuracy significantly degrades in scenes that differ from the training domain. Therefore, it is necessary to prepare multiple models for each domain, which increases the memory cost. When training multiple datasets with a single-head model, it is also necessary to redefine a different object class for each dataset. We propose a semantic-segmentation method that involves using a multi-head model for supporting multiple domains. The proposed method also involves using a shared network for sharing all domains for training datasets. This makes it possible to train multiple datasets with different object classes in a single network. To train all datasets equally, we also introduce mix loss, which simultaneously back-propagates the loss of each dataset. From experiments evaluating the proposed method, we confirmed that the method achieves higher or equivalent recognition accuracy with fewer parameters than using a single-head model for each dataset when training datasets with the same class, training different datasets at the same time, and training datasets individually.**

I. INTRODUCTION

Semantic segmentation is used to classify objects in images at the pixel level. It can be used to recognize not only the object class but also the position and shape of the object. However, the recognition accuracy of semantic segmentation significantly degrades due to changes in the domain such as the scene or camera position, which differ from the training. When semantic segmentation is used for automatic driving systems that operate in various regions, multiple models trained with data from different regions must be prepared. This increases the memory cost and puts other burdens on the system side.

We propose a semantic-segmentation method that introduces a domain attention (DA) module and uses a multi-head model to train datasets from different domains simultaneously. The proposed method uses a shared network with an encoder-decoder structure, where the encoder and a part of the decoder are shared by all domains. By introducing a DA module into ResNet, we can extract domain-specific features that cannot be obtained when using a single domain model. In addition, each head outputs a dataset-specific class, as shown in Figure 1(b). This makes it possible to simultaneously train datasets with different object classes, such as Cityscapes and Mapillary, which cannot be trained with the single-head model in Figure 1(a). During training, our method introduces mix loss, which simultaneously back-propagates the loss of each dataset to avoid bias towards a single dataset. With this the proposed method, it is possible to train a multi-head model that can handle multiple domains with only a small increase in parameters. We evaluated the effectiveness of the proposed method through experiments on multiple datasets.

[1]Authors are with the College of Engineering, Chubu University, Kasugai, 487-8501, Japan masaki@mprg.cs.chubu.ac.jp



(a) Single Head model (b) Multi Head model

Fig. 1. Overview of (a) single-head model and (b) multi-head model.

The contributions of this paper are as follows.

- We propose a semantic-segmentation method that involves using a multi-head model. By preparing an output head specific to each domain, datasets with different object classes can be trained simultaneously.
- By introducing a DA module, which shares information, and mix loss, which simultaneously back-propagates the loss of each dataset, to the multi-head model, multiple domains can be trained simultaneously.
- We evaluated the effectiveness of the proposed method by measuring its recognition accuracy on multiple datasets, those with the same object class and those with different object classes.

II. RELATED WORK

With the advent of fully convolutional networks (FCNs) [1], semantic-segmentation methods using CNNs has been actively studied and achieved high recognition accuracy [2], [3], [4]. SegNet [5] and U-Net [6], which are FCNs that have an encoder-decoder structure, contribute to memory saving. Dilation convolution [7], [8] captures a wide range of features by using a wide range of filter strides. It has been incorporated into many semantic-segmentation methods. PSPNet [9] and DeepLab [10], [11], [12] use spatial pyramid pooling [13] between the encoder and decoder and can acquire multi-scale contexts by pooling feature

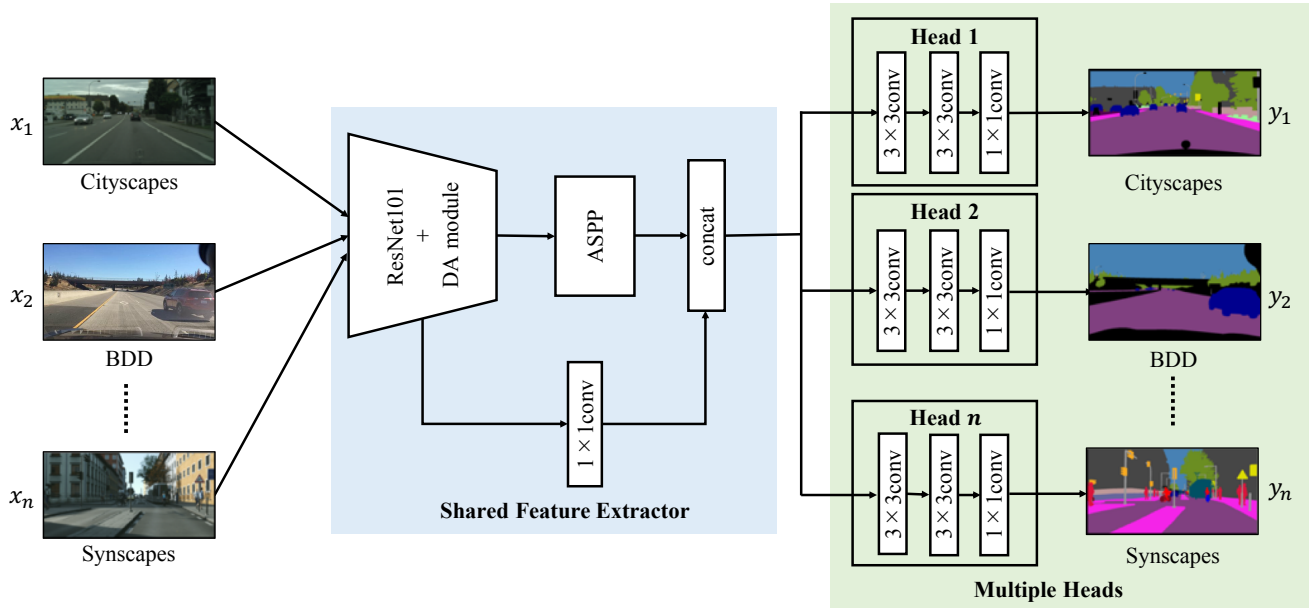


Fig. 2. Network structure used with proposed method.

maps of different sizes. Channel-wise attention, which is used in object-recognition tasks, is also used in semantic segmentation [14], [15], [16], [17]. By assigning importance to the feature maps, the recognition accuracy of each object class can be improved.

To adapt a single model to multiple domains, multi-domain learning has been studied [18], [19], [20]. Many methods are aimed at object recognition [21], [22]. Such methods introduce domain-specific convolutional layers or batch normalization to train each domain separately. An optimal network structure for multi-domain learning was proposed [23] for object detection. In particular, by eliminating the domain-specific parameters and adding a module that shares domain information in the shared network, it is possible to train a single model that acquires multi-domain information. For semantic segmentation, composite datasets that combine multiple datasets into one, such as MSeg [24] and Bevandic et al [25], have been created. A composite dataset is created by converting different datasets into a common label. This requires re-definition of labels, re-annotation of certain classes, deletion, and integration, which is a time-consuming process. Kalluri et al [26]. use a domain-specific model and a domain-sharing model with semi-supervised learning to achieve Multi Domain learning with a single model.

III. PROPOSED METHOD

We propose a semantic-segmentation method that involves using a DA module introduced to a multi-head model to train multiple domains simultaneously. The network structure used with the proposed method is shown in Figure 2. The base network is DeepLab v3+ [12] with ResNet101 [27] as the backbone. DeepLab v3+ is a network that uses Atrous spatial pyramid pooling (ASPP). ASPP can acquire multi-scale features by integrating different convolutional processes of dilations in parallel and executes 1×1 convolution, 3×3 convolution with dilation set to 6, 12, and 18 and global average pooling (GAP) in parallel and concatenates the acquired five feature maps. To improve the segmentation accuracy around the boundary of each object, the feature maps of the lower layers are skipped to the decoder. The feature map to be skip-connected is that of the first stage of ResNet. The skip-connected feature map is 1×1 convoluted and concatenated with the feature map acquired from ASPP. This allows us to obtain the features of object boundaries that are obscured in the high-dimensional layer.

A. Multi-head model

With conventional semantic-segmentation methods, the feature maps acquired by the encoder and decoder are input to the output head. The output head has a single-head structure that outputs probability maps for the number of classes. Since this structure can only output for predefined classes, it cannot train datasets with different numbers of classes simultaneously. Therefore, we adopt a multi-head model to train datasets with different numbers of classes simultaneously. This allows us to prepare an output head for each dataset so that we can deal with dataset-specific classes. In the shared network, the feature map acquired from ASPP and that of the first stage of ResNet are concatenated and input to the dataset-specific output heads. Each output head consists of three convolutional layers of two 3×3 and 1×1. Bilinear up-sampling is used to resize the acquired probability maps to the input size. The proposed method inputs the feature maps obtained from the shared network to the output head corresponding to the dataset. It then outputs a probability map for each class. Here, let N be the number of training datasets. Given input data x, the probability of

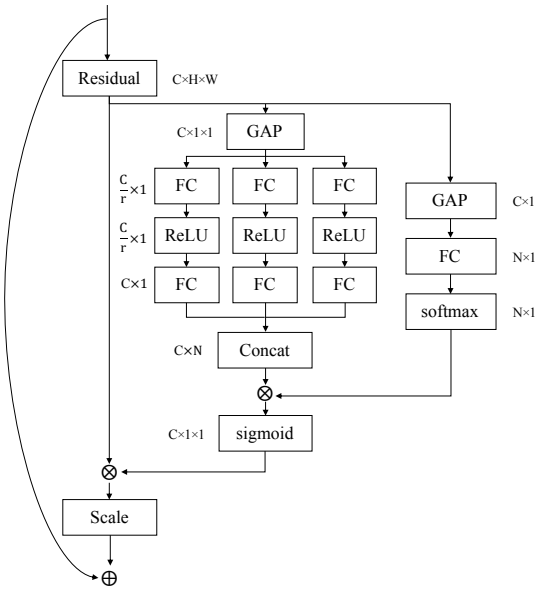


Fig. 3. Structure of domain attention (DA) module

each class is calculated as

$$y_n = \mathrm{F}_{\mathrm{head}_n}\left(\mathrm{F}_{\mathrm{FE}}\left(x\right)\right), \tag{1}$$

where F_{FE} is the shared network and $\mathrm{F}_{\mathrm{head}_n}$, $n \in \{1, \ldots, N\}$ is the dataset-specific output head. The only dataset-specific parameter to be prepared in this structure is the output head. This makes it possible to minimize the increase in the number of parameters due to the increase in datasets since the feature extractor is shared.

B. Domain attention module

The DA module [23] is applied to the residual block of ResNet and can acquire multiple feature representations by learning while sharing the information of each domain. The structure of the DA module is shown in Figure 3. It is composed of an squeeze excitation (SE) adapter and domain assignment. The SE adapter consists of multiple SE modules [28], each one specialized for each domain. By concatenating each output, the expression space of all domains can be formed. The weight vector acquired by each SE module from input x can be obtained by

$$x_{\mathrm{SE}} = \mathrm{F}_{\mathrm{SE}}\left(\mathrm{F}_{\mathrm{avg}}\left(x\right)\right), \tag{2}$$

where $\mathrm{F}_{\mathrm{avg}}$ is GAP and F_{SE} is the FC+ReLU (rectified linear unit)+FC layer. The domain assignment consists of GAP, a fully connected layer, and softmax layer and acquires weights that adapt to the domain. The weights of domain assignment can be calculated as

$$w_{da} = \mathrm{softmax}\left(\mathrm{W}_{\mathrm{DA}}\mathrm{F}_{\mathrm{avg}}\left(x\right)\right), \tag{3}$$

where x is the feature map, and W_{DA} is the weight matrix of the softmax layer. The output from all the combined layers after GAP is equal to the number of SE modules in the SE adapter. The acquired weights are multiplied by the output of the SE adapter and calculated by the sigmoid function. This allows us to obtain a domain-appropriate weight vector from the SE adapter.

C. Loss function

In conventional multi-domain learning, we input different domain data in order, calculate the cross-entropy error at each head, and back-propagate each time. Since the parameters are updated for each dataset, there is a possibility of bias toward a particular dataset depending on the order in which the back propagation is executed. Therefore, we introduce mix loss, which inputs all domain data and sums the losses output by each head before back propagating, into the multi-head model. When training N datasets, the back propagation error L can be calculated as

$$L = \sum_{n=1}^{N} L_n. \tag{4}$$

By calculating the loss of all domains then back-propagating, it is possible to update the parameters of each domain at the same time and prevent recognition-accuracy improvement from improving for only a specific dataset.

D. Training process

Since the number of images contained in each dataset differs, the number of images trained per epoch is not equal. To train all datasets in a balanced manner, it is necessary to avoid an unbalanced number of images during training. Therefore, we adjust the number of images by matching the number of images used per epoch to the dataset with the largest number of training images. The mini-batch for training is composed of only data from the same dataset. This is to avoid mixing multiple domains in a single input since each head has domain-specific parameters. During training, mini-batches of all datasets are input sequentially, and the errors are accumulated and back-propagated simultaneously. However, as the number of datasets to be trained increases, the training time and memory usage become huge. To reduce the training time and memory usage, we use automatic mixed-precision [29] training.

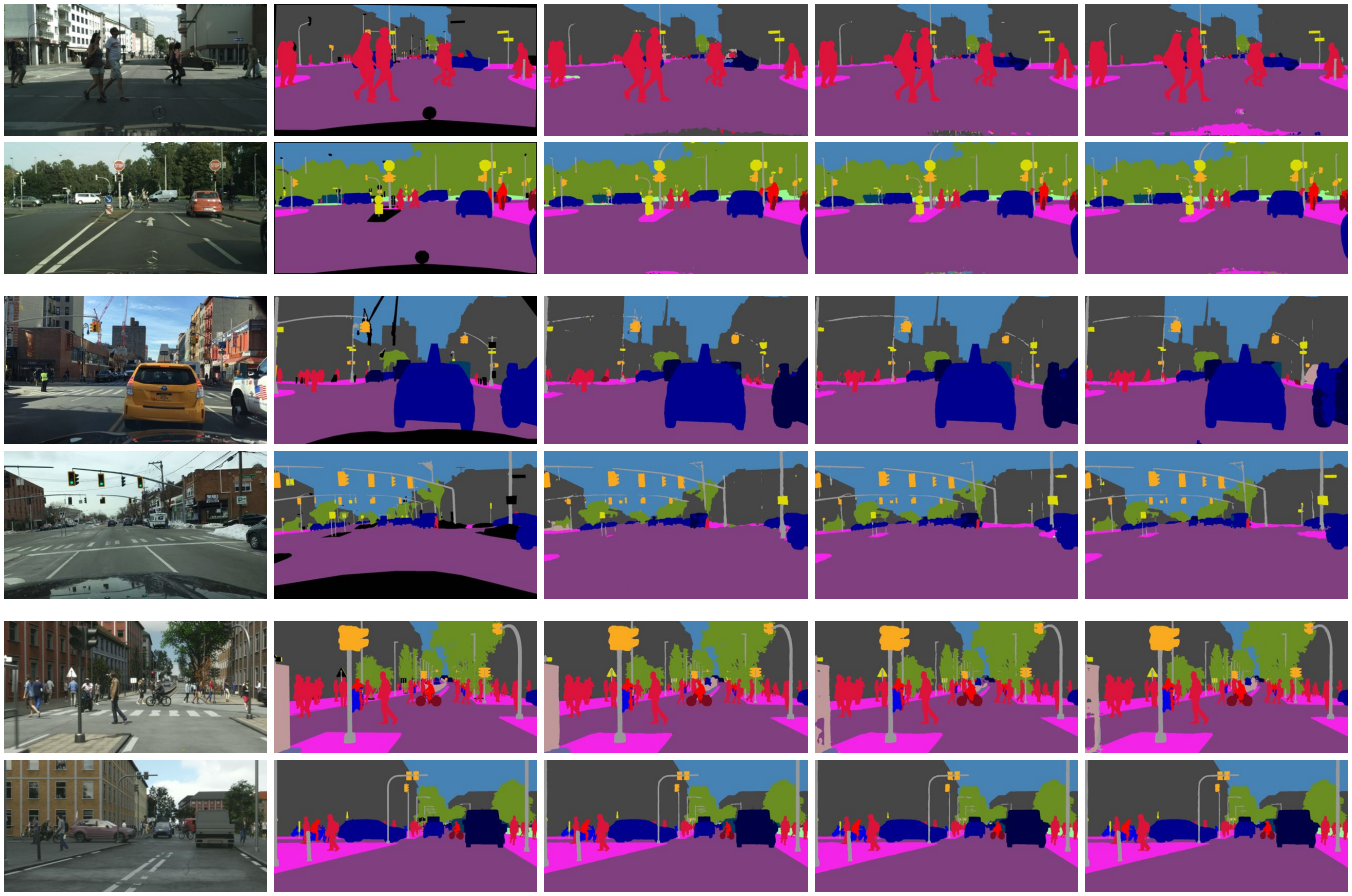
IV. Experiments

We evaluated the effectiveness of the proposed method using multiple datasets. We conducted the experiments on three datasets with the same classes, three datasets with different numbers of classes, and five datasets trained simultaneously. We applied random horizontal flipping, random scaling in the range [0.5, 2.0], and random cropping with 512×512 pixels for data augmentation for each dataset. We used stochastic gradient descent (SGD) with momentum set to 0.9 and weight decay set to 0.0001. We set the initial learning rate to 0.01 and scheduled the learning rate by multiplying by $(1 - \frac{itertotal}{iter})^{0.9}$. We set the number of training cycles to 100 epochs, and used the mIoU as the evaluation metric.

TABLE I

DATASETS USED FOR THIS STUDY

Dataset	Cityscapes	BDD	Synscapes	A2D2	Mapillary	ADE20K
Domain	Driving (Germany)	Driving (USA)	Driving (Simulator)	Driving (Germany)	Driving (Worldwide)	Everyday objects
Class	19	19	19	18	63	150
Training Images	2,975	7,000	23,000	26,955	18,000	20,210
Validation Images	500	1,000	2,000	4,493	2,000	2,000



Input image | Ground truth | Single Domain | Multi Head | Proposed

Fig. 4. Comparison of visualization results. The first and second rows show the results of Cityscapes dataset. The third and fourth rows show the results of BDD dataset. The fifth and sixth rows show the results of Synscapes dataset.

A. *Datasets*

Table I shows the number of classes, number of images, and domain information of the datasets used in the experiments. The experimental datasets can be divided into two categories: automotive image datasets and daily scenes. Both Cityscapes and A2D2 consist of images taken in Europe. However, Cityscapes only contains images taken in cities, while A2D2 includes those from highways and rural roads. Mapillary [30] and ADE20K [31] are datasets that contain images of different sizes. Therefore, we resized the short side of all images to 720 pixels. Cityscapes [32], BDD [33], and Synscapes [34] consist of the same 19 classes. For Mapillary, we used images from 63 classes, excluding the void category that includes vehicles, and used the images of 63 classes, excluding the void category that includes vehicles, etc. We also used the images of A2D2, which was redefined into 18 classes.

TABLE II

COMPARISON OF INFERENCE RESULTS OF CONVENTIONAL METHOD OF TRAINING DATASETS INDIVIDUALLY AND PROPOSED METHOD OF TRAINING MULTIPLE DATASETS SIMULTANEOUSLY [%]

Train \ Test	Cityscapes	BDD	Synscapes
Cityscapes	77.57	39.81	63.06
BDD	59.05	61.55	55.78
Synscapes	39.04	12.66	**91.55**
proposed method	**78.49**	**62.63**	90.18

TABLE III

COMPARISON OF INFERENCE RESULTS ON THE SAME CLASS OF DATASETS [%]

	DA module	Multi Head	Mix Loss	Cityscapes	BDD	Synscapes	Mean
Single Domain	-	-	-	77.57	61.55	91.55	76.89
Multiple Domains	-	-	-	75.55	63.14	86.91	75.34
	-	-	✓	75.86	**63.33**	88.10	75.76
	-	✓	-	77.30	59.47	**90.20**	75.66
	-	✓	✓	77.92	62.51	90.14	76.86
	✓	✓	✓	**78.49**	62.63	90.18	**77.10**

B. Experiments on dataset with the same classes

We used Cityscapes, BDD, and Synscapes as the target datasets. Table II shows the comparison of inference results from a conventional method of training datasets (domains) individually (hereafter, Single Domain) and the proposed method. Table III shows the comparison of recognition accuracies when training multiple datasets using only a single-head model (Single Head), using only mix loss (Mix Loss), using only a multi-head model (Multi Head), and using the proposed method of using introducing mix loss and a DA module to a multi-head model (Proposed). Table IV shows the comparison of number of parameters used with. And, Fig. 4 shows the output results of; those for Cityscapes in the first and second rows, BDD in the third and fourth rows, and Synscapes in the fifth and sixth rows.

Comparison with Single Domain. From Table II, we can see that the recognition accuracy of Single Domain considerably decreased for all datasets when domain information was learned and evaluated differently. This indicates that semantic segmentation cannot deal with untrained domains. However, the proposed method achieved better recognition accuracy for Cityscapes and BDD, and comparable accuracy for Synscapes by training the three datasets simultaneously. These results indicate that the proposed method is effective for multi-domain learning.

Comparison of Single Head, Mix Loss, Multi Head, and Proposed. To confirm the recognition accuracies of Proposed, we first compared it with Single Head. From Table III, we can see that Single Head achieved high accuracy for BDD but showed decreased accuracy for Cityscapes and Synscapes. This indicates that this it is biased towards one dataset. For Mix Loss, high recognition accuracy was achieved only with BDD. For Multi Head, high accuracy was achieved with Cityscapes and Synscapes, but decreased in accuracy for BDD. With Proposed, all datasets were trained in a balanced manner. Therefore, we can say that the introduction of both mix loss and a multi-head model is effective for training datasets with the same label.

Proposed improved in recognition accuracy compared with Multi Head on all datasets. The recognition accuracies for Cityscapes and BDD improved compared to that with Single Domain. This may be due to the fact that different domain information can be shared and used by the DA module. These results indicate that the DA module is effective for multi-domain learning.

Comparison of the number of parameters. As shown in Table IV, the number of parameters with Single Domain for training datasets individually increased with the number of datasets. Proposed reduced the number of parameters by 57. 10% by using a shared network compared with Single Domain. The increase rate of the number of parameters for Single Head was 1.04 times that for Multi Head and 1.28 times that for Proposed. These results indicate that the Proposed can train multiple datasets with only a small increase in the number of parameters, compared with preparing multiple models trained on each dataset (Single Domain).

TABLE IV

COMPARISON OF NUMBER OF PARAMETERS WITH SINGLE DOMAIN, SINGLE HEAD, MULTI HEAD, AND PROPOSED

	Params	Reduction rate [%]
Single Domain	178.02M	-
Single Head	59.34M	66.67
Multi Head	61.94M	65.21
Proposed	76.37M	57.10

TABLE V

COMPARING INFERENCE RESULTS ON DATASETS WITH DIFFERENT CLASSES [%]

	Cityscapes	Mapillary	ADE20K	Mean
Single Domain	77.57	43.71	36.42	52.57
Proposed	76.01	43.31	37.16	52.16

From these results, we confirmed that mix loss, a multi-head model structure, and DA module can achieve the same or higher accuracy than the base accuracy when training datasets with the same class.

C. Experiment with datasets with different classes

Next, we compared the accuracy of training datasets consisting of different numbers of classes simultaneously. We used Cityscapes, Mapillary, and ADE20K to compare Proposed, which showed high accuracy in the above experiments, with Single Domain.

The comparison results are listed in Table V. The accuracy of Proposed was comparable to that of Single Domain even when trained on datasets with different number of classes and domain information. Even if you learn the domain information of drive scenes such as Cityscapes and Mapillary and the domain information of everyday scenes such as ADE20K at the same time, you can see that each domain information can be learned with one model. Therefore, we

TABLE VI
COMPARISON OF INFERENCE RESULTS ON FIVE DATASETS [%]

	Cityscapes	BDD	Synscapes	A2D2	Mapillary	Mean
Single Domain	77.57	61.55	**91.55**	**78.08**	43.71	70.49
Proposed	**79.51**	**65.59**	89.47	76.56	**44.96**	**71.22**

confirmed that Proposed can be trained on datasets with different numbers of classes at the same time.

D. Experiments when training five datasets

The number of datasets used for training simultaneously was set to five. We used Cityscapes, BDD, Synscapes, A2D2, and Mapillary to compare Proposed with Single Domain.

The comparison results are listed in Table VI. When the five datasets were trained simultaneously, the recognition accuracy for Cityscapes, BDD, and Mapillary improved by 1.94, 4.04, and 1.25 points, respectively, over Single Domain. We also confirmed that the segmentation accuracy for Synscapes and A2D2 was the same as that of Single Domain. These results confirm that Proposed is effective even when training five datasets simultaneously.

V. CONCLUSIONS

We proposed a semantic-segmentation method that uses a multi-head model that learns different domains simultaneously. By applying a DA module, which shares domain information and mix loss, which simultaneously back-propagates the loss of each dataset, and using a multi-head model, which prepares an output head for each dataset, it is possible to train a single model for datasets with different classes. In the experiments, the segmentation accuracy was higher than that of using a single-head model when trained on data with the same classes, and was higher than that of a conventional method for training datasets individually. Even when simultaneously training datasets with different numbers of classes, which is difficult to train, the proposed method achieved the same accuracy as the conventional method. In the experiment on five datasets, the segmentation accuracy of the proposed method was higher than that of the conventional method. In the future, we will aim to have the segmentation accuracy of the proposed method surpass that of the conventional method even when training datasets with different numbers of classes and confirm its versatility by applying it to other base networks.

ACKNOWLEDGMENT

This work was supported by Council for Science, Technology and Innovation(CSTI), Cross-ministerial Strategic Innovation Promotion Program (SIP), Automated Driving for Universal Services/ Research on the recognition technology required for automated driving technology (levels 3 and 4).

REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[2] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-scnn: Gated shape cnns for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 5229–5238.

[3] F. Zhang, Y. Chen, Z. Li, Z. Hong, J. Liu, F. Ma, J. Han, and E. Ding, "Acfnet: Attentional class feature network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 6798–6807.

[4] A. Kirillov, Y. Wu, K. He, and R. Girshick, "Pointrend: Image segmentation as rendering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9799–9808.

[5] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., 2015, pp. 234–241.

[7] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *International Conference on Learning Representations (ICLR)*, 2016.

[8] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," in *International Conference on Learning Representations (ICLR)*, 2015.

[9] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6230–6239.

[10] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 834–848, 2018.

[11] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," in *arXiv: 1706.05587*, 2017.

[12] L. 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. 833–851.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[14] H. Zhang, H. Zhang, C. Wang, and J. Xie, "Co-occurrent features in semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 548–557.

[15] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3146–3154.

[16] Z. Zhong, Z. Q. Lin, R. Bidart, X. Hu, I. B. Daya, Z. Li, W.-S. Zheng, J. Li, and A. Wong, "Squeeze-and-attention networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 13 065–13 074.

[17] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7151–7160.

[18] M. Joshi, M. Dredze, W. W. Cohen, and C. Rosé, "Multi-domain learning: When do domains matter?" in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 1302–1312.

[19] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the IEEE Conference*

on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4293–4302.

[20] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70, 2017, pp. 1857–1865.

[21] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 506–516.

[22] ——, "Efficient parametrization of multi-domain deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8119–8127.

[23] X. Wang, Z. Cai, D. Gao, and N. Vasconcelos, "Towards universal object detection by domain attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7289–7298.

[24] J. Lambert, Z. Liu, O. Sener, J. Hays, and V. Koltun, "Mseg: A composite dataset for multi-domain semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2879–2888.

[25] P. Bevandić, M. Oršić, I. Grubišić, J. Šarić, and S. Šegvić, "Multi-domain semantic segmentation on datasets with overlapping classes," in *arXiv: 2009.01636*, 2021.

[26] T. Kalluri, G. Varma, M. Chandraker, and C. Jawahar, "Universal semi-supervised semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2019, pp. 5259–5270.

[27] 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 (CVPR)*, 2016, pp. 770–778.

[28] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[29] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu, "Mixed precision training," in *International Conference on Learning Representations (ICLR)*, 2018.

[30] G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *International Conference on Computer Vision (ICCV)*, 2017, pp. 4990–4999.

[31] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 633–641.

[32] 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 (CVPR)*, 2016, pp. 3213–3223.

[33] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2636–2645.

[34] M. Wrenninge and J. Unger, "Synscapes: A photorealistic synthetic dataset for street scene parsing," in *arXiv: 1810.08705*, 2018.