Multi-scale synergy approach for real-time semantic segmentation

Quyen Van Toan
School of Electronic and Electrical Engineering
Kyungpook National University
Daegu, Korea
yersin@knu.ac.kr

Min Young Kim
School of Electronic and Electrical Engineering
Kyungpook National University
Daegu, Korea
minykim@knu.ac.kr

***Abstract*—In deep convolution neural network based models for semantic segmentation, diverse receptive fields improve the performance by capturing disparate context information. Multi-scale inference is good for both thin and large objects. However, the final result is not optimal through averaging or Max pooling combination. In this paper, we propose an approach to take advantage of multi-scale predictions. Our uncertain-pixels part discovers the worse prediction of a low scale and chooses the complement from a high scale. The final output is effectively merged from two scales. We validate our proposed model with a series of experiments on different datasets. The results achieve the accuracy and speed for real-time semantic segmentation. On Cityscapes dataset, our network achieves 76.3 % mIoU at 50 FPS, and on Mapillary, 42.6 % mIoU.**

***Index Terms*—Multi-scale, semantic segmentation, real time.**

I. Introduction

In the age of artificial intelligence (AI) advances and the high qualities of modern computers and cameras, several applications in the fields of robot vision and self-driving cars have remarkable developments. Semantic segmentation plays an important role in the input information of the autonomous system. It is utilized to recognize and understand what is in the image at pixel levels. This method assigns labels to specific regions of an image. The performance has been directly affected by accuracy segmentation and inference speed.

In the semantic segmentation revolution, we briefly review the proposed approaches after the deep learning emergence. The Fully Convolution Networks (FCNs) [1] pioneer the way to build the deeper structures of models. The method extracts image features through hierarchical convolution layers after utilizing the last one to predict the segmentation. At that time, the output results have reached an excellent benchmark and significantly impacted the field. Characteristics of the final layer are deep features and low resolutions. To overcome low resolution drawback, DeeplabV3 [2] enrich spatial contextual information by deploying dilated convolutions with flexible rates. Another problem of FCNs, the gradient values are gradually inclined to zeros, so the process suffers the loss of importation details. Skip connections are introduced and provide an alternative path for the gradient by adding information of lower layers to higher layers [3]. The combination of multiple skip connections and multi-dilated convolutions enhanced accuracy [4].

Throughout the improvement of semantic segmentation, novel methods concentrate on enhancing the accuracy and speed inference. Early, they address the problem of segmentation in class categories. They propose region-based object detectors with scanning-windows part models and global appearances to obtain quality object segmentation [5], a combination of regions and convolution neural network (CNN) to boost the performance [6], room-out the regions to obtain a higher resolution in [7], and exploit a multi-region to rich representation [8]. The accuracy is gradually improved over time. Later, innovative approaches focus on real-time semantic segmentation with high accuracy. To attain real-time, we need to deal with spatial information for high accuracy as well. The rich spatial information is captured by a new design with a small stride [9], an effective fast attention with cosine modification [10], and incorporation between high-resolution global edge and low resolution [11]. In these ways, we can achieve 74.4 % at 72 FPS on Cityscapes dataset or 68.4 % with 105 FPS on COCO datasets. When features have high resolution, multi-scale inputs are proposed to capture diverse context information [12], [13].

In this paper, we propose multi-scale synergy approach for real-time semantic segmentation. We use two different-size inputs. One is remaining as a high scale, and the other is downsampling by 2 as a low scale. In order to leverage advances from both scales, we deploy the uncertain-pixel determination that detects the bad prediction of the low scale and effectively fuses the predictions from two scales. We achieve high results on Cityscapes 76.3 mIoU and 50.1 FPS, and on Mapillary 42.6 mIoU.

II. Related Works

Semantic segmentation demands spatial information to resolve fine detail. When the resolution is high, it causes the receptive field to be shrunken. In this case, the receptive field is small, it is hard to cover the whole context of large objects. It leads the bad prediction for large objects. Many approaches are proposed a contextual combination from multiple scales. Multi-scale context can be obtained from various levels of pyramid pooling [14] or different rates of dilation [15]. In [16], author builds the architecture with two different-size inputs. The small scale is used to get contextual information and the

others are used for generating spatial detail. An image cascade network with multi-resolution branches is proposed in [17]. The score maps of multiple scales are combined by averaging or max-pooling methods to generate the final output. With the average method, each pixel value is the result of an average combination between a pair scale or all scales. It leads to the problem of combining the best predictions with poorer ones. Instead of selecting only one of N scales like Max-pooling, We propose a novel method to combine two scale inputs via the uncertain-pixel determination part. The uncertain-pixel part keeps the best prediction of the low scale and improves the uncertain parts through the high scale complement.

III. Method

In this section, we introduce three main parts. The uncertain-pixels determination for low scale segmentation is described in section III-A. An overview of the proposed architecture is shown in section III-B. Lastly, we will represent the optimization function III-C

A. Uncertain-pixels determination for low scale segmentation

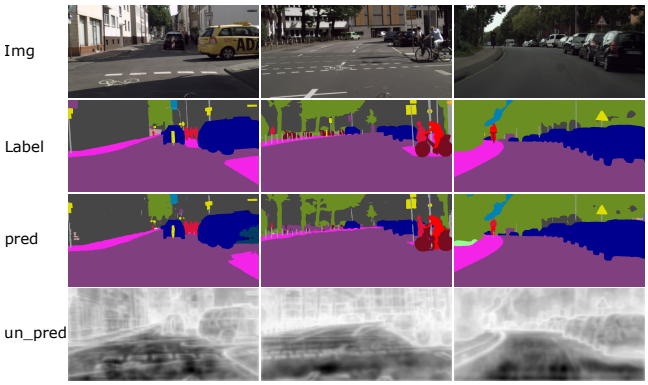


Fig. 1: Illustrate uncertain masks of the prediction. The rows are images, labels, prediction masks and uncertain-pixels masks of the prediction, top to bottom respectively. The uncertain pixels are pixels at prediction which have high probabilities of wrong labels. The white color indicate the uncertain areas.

In general, the semantic segmentation of large objects in the image such as buildings, sky, or trucks always have good predictions, contrast narrow objects like poles, fences, traffic lights, etc. will have low precision. Moreover, object's boundaries are the most uncertain prediction. To achieve high accuracy, we intend to deal with the precision of narrow objects and boundaries.

In the networking, the trunk generates separately the final map for each scale, including N_classes channels of the datasets. We subtract the chosen final segmentation by one to produce "1-segmentation layers" or uncertain layers. The pixel's values of the uncertain layers are depended on those of the final segmentation layers. Pixels at the object's body have one probability much higher than the others, but pixels at the boundaries will have at least two high probabilities which are nearly equal. For clear understanding, we make a prediction step for an explanation. Fig. 1 includes images, labels, prediction masks, and uncertain-pixels masks. The final layers are predicted to generate prediction masks. The uncertain masks are utilized to determine which parts have of the prediction have high probabilities of wrong labels.

In our proposed method, the uncertain masks are generated from the lower scale. The low scale predicts good results with large objects, especially near the screen. The fourth row in Fig. 1 shows that the uncertain areas have a white color and locate at boundaries and far away from screens. The main function of this mask tweaks the high scale to more focus on the white color.

B. Proposed structure

Fig. 2 illustrates the proposed architecture, including three main components. Firstly, there are two scales as inputs. The low scale is 0.5x and the other is 1.0x. Second, we utilized the model Deeplabv3+ with ResNet-50 as trunk. The shared weighted model employs for generating score maps for all scales. Lastly, the combination of two scales is demonstrated in the dash-line box which is executed in pixel-wise levels.

We visualize the combination part for obviously understanding, depicted in the red box. N_classes channels, generated by trunk, are passed through the Max function to predict the segmentation label for each location. As mentioned above, the segmentation of a low scale is focused on the near screen. It achieves good precision with large objects such as roads, cars, but it gets worse for small objects, shown in the red box of Fig. 2. Inversely, the segmentation label of a high scale has a better result at a far screen. The primary hinder of the high scale is not covering the whole necessary context of large objects, especially large objects near the screen. Our target is to take advantage of both scales. Instead of utilizing max pooling or average pooling methods for the combination, we deploy an uncertain-pixels determination for the network. The uncertain mask is produced from a low scale. This mask detects precarious labels of the low scale, assigned in white color in the mask. The white color appears at the far screen, small objects, and particularly at the object's boundaries. The white color indicates attentive locations and the others are non-attentive. The uncertain mask will select which parts of the high scale mainly contribute to the final output. The contributions by multiplying between the uncertain mask and the high scale are object's boundaries, small objects, and far screen objects. conclusively, the advance of the uncertain mask leverages all best predictions and almost neglects the worst prediction of the high scale.

In our proposal, the weight output is excellently merged by two scales. The objects on the far screen are mainly contributed from both scales. The final segmentation map is calculated as equation 1.

$$\mathcal{S} = \mathcal{U}(\mathcal{S}_{lo}) + \mathcal{S}_{hi} * \mathcal{U}(1 - \mathcal{S}_{lo}) \quad (1)$$

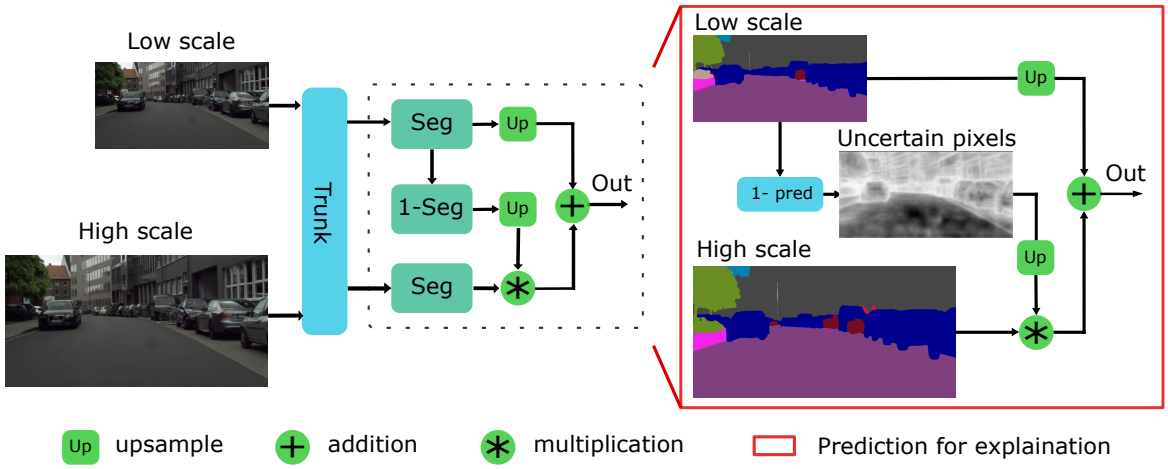


Fig. 2: Structural diagram of the proposed multi-scale synergy approach. The red box is prediction step for explanation. In the uncertain-pixel mask, the white color indicates a high value, and the dark color is small value.

where $\mathcal{S}$ and $\mathcal{U}$ are the semantic segmentation and the bilinear umsampling operation, respectively. Two input scales, "lo" denote as low and "hi" is high scale.

C. Optimization

We use Stochastic Gradient Descent (SGD) for our optimizer. SGD performs frequent updates with a high variance that cause the objective function to fluctuate heavily. SGD in contrast performs a parameter update for each training example x_i and label y_i:

$$\theta = \theta - \eta \Delta_\theta \mathcal{J}(\theta; x_i; y_i) \tag{2}$$

where θ is stochastic gradient decent, Δ_θ and $\mathcal{J}(\theta)$ are gradient of the objective function and an objective function

For loss computation, we utilize a Cross-Entropy to calculate the losses. The Cross-Entropy is measured how accurate the model is in predicting the data shown in equation 3.

$$L = \frac{1}{N} \sum_{i=1}^{N} (-y_i \; log(p_i)) \tag{3}$$

where L and N are the loss and the size of dataset, respectively. p_i denotes the segmentation predicted probability, and y_i is true labels.

The last formula, intersection over union (IoU) is an evaluation metric used to measure the accuracy of an object detector on a particular dataset shown in equation below.

$$IoU = \frac{|Target \cap Prediction|}{|Target \cup Prediction|} \tag{4}$$

IV. Experiments

In experiments, we use the following standard measures: mini-batch stochastic gradient descent (SGD) for the optimizer, cross-entropy loss function, intersection of union (IoU), and fame per second (FPS). We trained the model on an Nvidia Titan X with 12GB of GDDR5X memory for the Cityscapes dataset, and a GeForce RTX 3090 with 24GB of G6X memory for Mapillary. We train for 150 epochs with a batch size of 2 per GPU, a momentum of 0.9, weight decay of $5e{-}4$, and a batch size of 2 per GPU. With an initial learning rate of 0.01, we use a polynomial learning rate.

A. Cityscapes dataset

Cityscapes dataset consists of 5,000 images with 19 semantic classes captured from 50 different countries. The resolution is 2048 x 1024 pixels. The whole dataset is partitioned into three sets training, validation, and test. There are 2,979 images in the training set, 500 images in the validation set, and 1,525 images in the test set.

In Table I , we demonstrate the class-accuracy comparison. The performance displays a reasonable balance between thin objects and large objects compared to Previous SOTA methods. Our proposed model achieves 76.3 % mIoU. In Table II, the results show that our model outperforms existing approaches for segmentation accuracy while still achieving real-time implementing efficiency. Although FANet [10] has a faster speed than ours, our proposal gets 1.3 mIoU greater than them in an accuracy improvement. conclusively, Our approach performs effectively in terms of accuracy and speed on Cityscapes dataset. Qualitative results are visualized in Fig. 3a

B. Mapillary Vista

Mapillary Vista is a large dataset collected from city streets around the world. It consists of 25,000 images with 66 object categories, and the images have various resolutions. Due to large number of classes and high resolution, images are cropped to 2177x1632 pixels to reduce the computation and memory requirement.

In this section, we evaluate and compare the segmentation accuracy with other methods such as AGLNet [25], DABNet [21], RGPNet [27]. With 66 classes, our approach still catches 42.6 mIoU for segmentation accuracy. Table III show that our method surpasses other approaches, particularly DABNet. Finally, Qualitative results are visualized in Fig. 3b

Method	Road	swalk	build	wall	fence	pole	tlight	tsign	veg.	terr	sky	pers	rider	car	truck	bus	train	mcle	bicle	mIoU
CGNet [18]	97.7	81.0	89.8	42.5	48.0	56.2	59.8	65.3	91.4	68.2	94.2	76.8	57.1	92.8	50.8	60.1	51.8	47.3	61.7	68.0
EDANet [19]	97.8	80.6	89.5	42.0	46.0	52.3	59.8	65.0	91.4	68.7	93.6	75.7	54.3	92.4	40.9	58.7	56.0	50.4	64.0	67.3
ESPNet [20]	97.3	78.6	88.8	43.5	42.1	49.3	52.6	60.0	90.5	66.8	93.3	72.9	53.1	91.8	53.0	65.9	53.2	44.2	59.9	66.2
DABNet [21]	97.8	80.7	90.2	47.9	48.1	56.4	61.8	67.0	92.0	69.5	94.3	80.3	59.2	93.7	46.0	57.1	35.0	50.4	66.8	68.1
CFPNet [22]	97.8	81.4	90.5	46.4	50.6	56.4	61.5	67.7	92.1	68.9	94.3	80.4	60.7	93.9	51.4	68.0	50.8	51.2	67.7	70.1
RELAXNet [23]	98.94	84.9	92.2	57.2	54.8	64.3	70.6	74.0	93.0	71.8	94.8	83.7	64.4	95.1	58.6	72.7	58.2	59.9	71.8	74.8
DSANet [24]	96.8	78.5	91.2	50.5	50.8	59.4	64.0	71.7	92.6	70.0	94.5	81.8	61.9	92.9	56.1	75.6	50.6	50.9	66.8	71.4
FANet [10]	97.9	83.3	91.6	55.5	55.1	60.3	66.2	74.9	91.7	61.8	94.7	78.5	58.1	94.1	76.8	85.1	74.5	50.7	73.9	75.0
AGLNe [25]	97.8	81.0	91.0	51.3	50.6	58.3	63.0	68.5	92.3	71.3	94.2	80.1	59.6	93.8	48.4	68.1	42.1	52.4	67.8	70.1
Ours	97.9	83.9	92.0	60.1	60.2	59.4	63.6	74.6	91.8	61.9	94.3	78.6	59.6	94.2	80.1	87.1	75.3	62.56	73.3	76.3

TABLE I: Class-accuracy comparison on the Cityscapes dataset. List of classes from left to right: road, side walk, building, wall, fence, pole, traffic light, traffic sign, vegetation, terrain, sky, person, rider, car, truck, bus, train, motorcycle, and bicycle.

Method	Resolution	mIoU	FPS
AGLNet [25]	512 × 1024	70.1	52
FANet [10]	1024×2048	75.0	72
ICNet [17]	1024x2048	67.7	38
BiseNet [9]	768×1536	74.8	47
SwiftNet [26]	1024×2048	75.4	40
Ours	1024x2048	76.3	50

TABLE II: Accuracy and speed comparison of proposed method against other SOTA methods on Cityscapes.

Method	Resolution	mIoU
AGLNet [25]	1024x2048	30.7
DABNet [21]	1024x2048	29.6
RGPNet [27]	1024×2048	41.7
Ours	2177x1632	42.6

TABLE III: Accuracy comparison of proposed method against other SOTA methods on Mapillary Vista.

V. Conclusion

In this work, the multiple scales help capture the contextual information at different receptive fields. We propose an uncertain-pixels determination which brings an effective way to combine multiple scales at element-wise levels. Our approach shows the improvement in segmentation accuracy while still achieving real-time implementing efficiency. Due to the hardware limitation, we just use a lightweight network for our method. In the future, we will implement on a heavy network to enhance the accuracy.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1A6A1A03043144)

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*, 2015, pp. 3431–3440.

[2] L.-C. 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*, vol. 40, no. 4, pp. 834–848, 2017.

[3] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.

[4] T. Yamashita, H. Furukawa, and H. Fujiyoshi, "Multiple skip connections of dilated convolution network for semantic segmentation," in *2018 25th IEEE international conference on image processing (ICIP)*. IEEE, 2018, pp. 1593–1597.

[5] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik, "Semantic segmentation using regions and parts," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3378–3385.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[7] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3376–3385.

[8] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware cnn model," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1134–1142.

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

[10] P. Hu, F. Perazzi, F. C. Heilbron, O. Wang, Z. Lin, K. Saenko, and S. Sclaroff, "Real-time semantic segmentation with fast attention," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 263–270, 2020.

[11] H. Lyu, H. Fu, X. Hu, and L. Liu, "Esnet: Edge-based segmentation network for real-time semantic segmentation in traffic scenes," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1855–1859.

[12] W. Wang, S. Wang, Y. Li, and Y. Jin, "Adaptive multi-scale dual attention network for semantic segmentation," *Neurocomputing*, vol. 460, pp. 39–49, 2021.

[13] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," *arXiv preprint arXiv:2005.10821*, 2020.

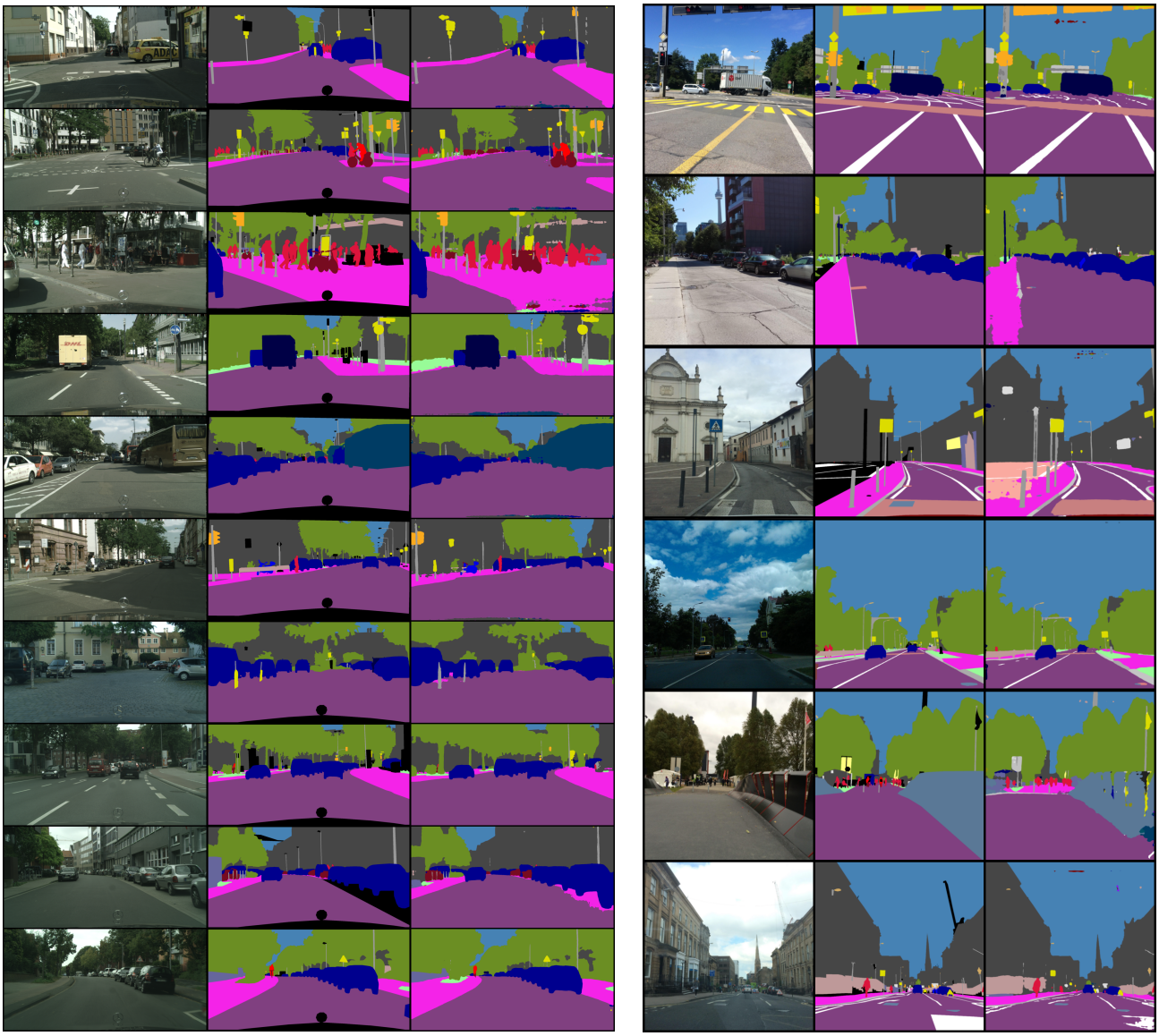
[14] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

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

[16] R. P. Poudel, U. Bonde, S. Liwicki, and C. Zach, "Contextnet: Exploring context and detail for semantic segmentation in real-time," *arXiv preprint arXiv:1805.04554*, 2018.

[17] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 405–420.

[18] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation,"



(a) cityscapes (b) Mapillary

Fig. 3: Qualitative results of proposed approach on Cityscapes val set (left) and Mapillary Vista val set (right).

IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 1, pp. 263–272, 2017.

[19] S.-Y. Lo, H.-M. Hang, S.-W. Chan, and J.-J. Lin, "Efficient dense modules of asymmetric convolution for real-time semantic segmentation," in *Proceedings of the ACM Multimedia Asia*, 2019, pp. 1–6.

[20] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, "Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9190–9200.

[21] G. Li, I. Yun, J. Kim, and J. Kim, "Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation," *arXiv preprint arXiv:1907.11357*, 2019.

[22] A. Lou and M. Loew, "Cfpnet: Channel-wise feature pyramid for real-time semantic segmentation," *arXiv preprint arXiv:2103.12212*, 2021.

[23] J. Liu, X. Xu, Y. Shi, C. Deng, and M. Shi, "Relaxnet: Residual efficient learning and attention expected fusion network for real-time semantic segmentation," *Neurocomputing*, vol. 474, pp. 115–127, 2022.

[24] M. A. Elhassan, C. Huang, C. Yang, and T. L. Munea, "Dsanet: Dilated spatial attention for real-time semantic segmentation in urban street scenes," *Expert Systems with Applications*, vol. 183, p. 115090, 2021.

[25] Q. Zhou, Y. Wang, Y. Fan, X. Wu, S. Zhang, B. Kang, and L. J. Latecki, "Aglnet: Towards real-time semantic segmentation of self-driving images via attention-guided lightweight network," *Applied Soft Computing*, vol. 96, p. 106682, 2020.

[26] M. Orsic, I. Kreso, P. Bevandic, and S. Segvic, "In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 607–12 616.

[27] E. Arani, S. Marzban, A. Pata, and B. Zonooz, "Rgpnet: A real-time general purpose semantic segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3009–3018.