Dual-inferences mechanism 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*—Autonomous cars have potential developments based on technology evolution. In the street scenes, the car needs to deal with a wide range of object sizes. Existing methods generally concentrate on deploying a single inference for semantic segmentation. However, one single scale is not suitable to capture the whole information of diverse sizes. It can effectively capture the context of thin objects, but it will get problems to cover the whole information of large objects, and reversely. In this paper, we propose an approach based on multi-scale inference to tackle the above difficulty. The multi-scale mechanism proposal employs two inference scales. Each scale is processed by a specific rate set of atrous spatial pyramid pooling. The segmentation maps are added together to take advantage of all scales. We validate our networks with a series of experiments on different open datasets. The approaches achieve high accuracy while reaching the speed for real-time semantic segmentation. The results are 75.5 % mIoU at 51 FPS on Cityscapes and 42.0 % mIoU on Mapillary Vistas.**

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

I. Introduction

Computer vision is a field of study solving the problem of scene understanding. The goal is to interpret information from digital sources such as photographs and videos. Semantic segmentation is one type of computer vision, which labels each pixel of the images into corresponding classes. This method treats the same way for both stuffs and things.

Besides the development of technology infrastructure, modern cameras and fast computers have mainly contributed to the growth of self-driving cars. Initially, lightweight semantic segmentation structures are presented for autonomous driving by utilizing 3D LiDAR point cloud scans with depth-wise separable convolution layers [1], designing a novel layer to treat a single deep model as a cascade of several sub-models [2], employing a hierarchical dilation and feature refining with different receptive fields to capture the object sizes [3], using multi-scale depth-wise residual blocks to fuse the local information and contextual information [4], proposing Turbo Unified network to customize decoder module [5], and using factorized dilated depth-wise separable convolutions to extract features from image inputs. The lightweight structures have critically reduced the computation and have a fast speed, but the segmentation accuracy also is degraded. In order to have less computation and improve accuracy, the effective convolution network is proposed to minimize computation resources. In [6], [7], they optimize the complex architectures of previous method by using residual connections and factorized convolutions. This approach can run on a high-performance graphics processing unit (GPU) and on embedded system-on-module as well. Alvaro *et al* [8] apply fisheye cameras to enlarge the receptive field. Efficient residual factorized CNN are employed to extract information from eyefish inputs. In [9], the atrous spatial pyramid pooling module combines with various residual connections and depthwise convolutions to spend a short time for network computation. The approach has a speed at 161 fames for second and 67.81 % mIoU on the Cityscapes dataset. MiniNet depicted in [10], the method uses multi-dilation depthwise separable convolutions to decrease the computation requirements. By modifying and improving the convolutions, the networks can effectively extract features from inputs with less computation cost.

In this study, we propose dual-inferences mechanism for real-time semantic segmentation. The image sizes are changed before feeding into a shared-weights network. A high scale remains the size of input images, and a low scale is downsampled by a factor of 2. In order to enhance advantages of each scale, we apply different sets of the dilation rates to particular scale. The results of Cityscapes are 75.5 % mIoU at 51 FPS and Mapillary Vistas are 42.0 % mIoU.

II. Related Works

Unlike other tasks of computer vision such as classification and object detection, they mainly demand coarse information from the network. Semantic segmentation solves the task to distinguish objects at boundaries. It requires not only coarse information but also high-resolution output. The deep network has the main goal to contribute semantic features, and the high resolution intends to deal with the object's boundary information. By these requirements, many approaches concentrate on the multiple-branches network. In [11], the approach illustrates the multi-scale combination of VGG-16 backbone. Three top layers of the network are processed by different branches and then added together to predict a final segmentation. The proposal includes two main branches generated from the four-deepest layers depicted in [12]. The outputs are concatenated together and passed through a convolution to predict the results. Other approaches also intend to effectively obtain both semantic information and rich spatial information such

as multi-scale contextual intertwining [13] or dynamic multi-scale filter [14].

The deeplab-series architecture is widely known as multiple fields of views. The proposed atrous convolution generates diverse receptive fields by changing the rates. Generally, it involves a set of different rates which can suitably capture objects of various sizes. The deeplab series are the atrous convolution combining with fully connected CRFs [15], the atrous spatial pyramid pooling (ASPP) [16], and the encoder-decoder with atrous separable convolution [17]. Due to the computation cost, the atrous convolution is generally applied to the deepest layer of the backbone. The lower layers are calculated by a normal convolution, so it leads to the aforementioned problems of a single receptive field. In order to overcome these drawbacks, some novel methods propose multiple scales of input images. The approach will change the size of an input image before passing through the backbone network [18], [19].

III. Method

In this part, we introduce more detail our proposal. Firstly, we explain characteristics of each input scale in section III-A. Next, the overview of proposed structure is shown in section III-B. Lastly, the optimization function in the model is illustrated in section III-C.

A. Characteristics of inference scales

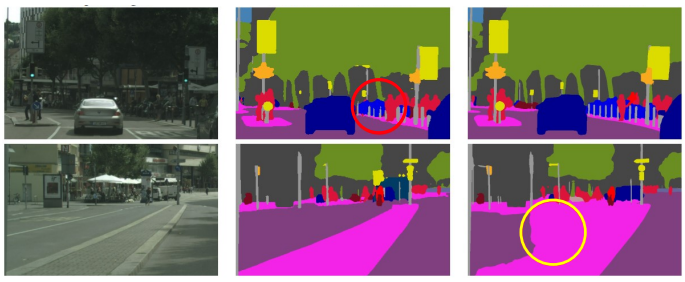


Fig. 1: The characteristics of different scale inferences. Three columns are input images, low-scale inference, and high-scale inference, respectively.

In this section, we introduce the characteristics of each scale inference and then show the advantages of multi-scale inference. Fig. 1 illustrates two input images of street scenes and their predictions at different scales. The first row consists of a lot of small and thin objects, especially near screens. We can concentrate on the red circle which has thin poles. The high scale achieves much better accuracy than the low scale. In the second row of Fig. 1, it comprises large objects and is far away from screens. We draw the yellow circle to compare the prediction of different scales. The high scale has a small receptive field which can effectively capture the thin and narrow objects. While the road is large objects, this receptive field can not cover the whole features. Hence, it leads to the problem like a yellow-draw circle.

After analyzing the above example of different inference scales, we can recognize that the receptive field is reversely proportional to inference scales. The low scale has a bigger receptive field which can effectively capture large objects such as roads, buildings. Oppositely, the high scale has a small receptive field that can cover the whole information of thin objects. Conclusively, each inference scale has some advantages and dis-advantages, so we propose a combination of diverse scales to take advantages. In this study, we design an approach to combine the dual inferences. Our approaches surpass existing methods to have higher accuracy while still achieving real-time segmentation on Cityscapes and Mapillary Vistas datasets.

B. Proposed structure

The proposed structure is shown in Fig. 2. It includes three main components such as input images, a shared-weights network, and multi-scale fusion module.

Based on the characteristics of inference scales, we implement the network with multi-scale inputs to enhance the semantic performance. Each image will be resized into two different dimensions. The high scale is assigned as a larger size when it has the same size of input images, and the low scale equals half sizes of the high scale. Next, they are straightly fed into the shared-weights network.

We use DeeplabV3+ with backbone ResNet50 to extract features from input images. For Deeplabv3+, it processes not only the final layer but also one low-level layer of Resnet50. The low-level features contain more spatial information, so it is simply fed into 1x1 convolution to calculate features. The final features include rich contextual information, then it is passed through atrous spatial pyramid pooling to effectively capture diverse sizes of objects. The representation of class channels is generated by feeding concatenating features through the 3x3 convolution layer. This above process is applied to each scale. Based on the characteristics of each inference scale, a low scale can effectively capture large objects and a high scale is suitable for narrow objects as mention in previous section. We apply different sets of dilation rates to contextual branch. A low scale deploys output stride = ”16” in which dilation rates have larger fields of views. A high scale will has a set of dilation rates with smaller receptive fields. By utilizing this method, we can improve characters of each input scale. Table I illustrates different rate sets of atrous spatial pyramid pooling.

TABLE I: Atrous spatial pyramid pooling with different sets of dilation rates.

Ouput stride	Rate no.1	Rate no.2	Rate no.3
8	6	12	18
16	12	24	32

Each scale prediction contain class channels of corresponding datasets. In order to leverage useful learnable weights from both scales, score maps are fused together at pixel-wise levels calculated by equation 1. Fig. 3 illustrates the combined process. For the first step, a low scale prediction is upsampled

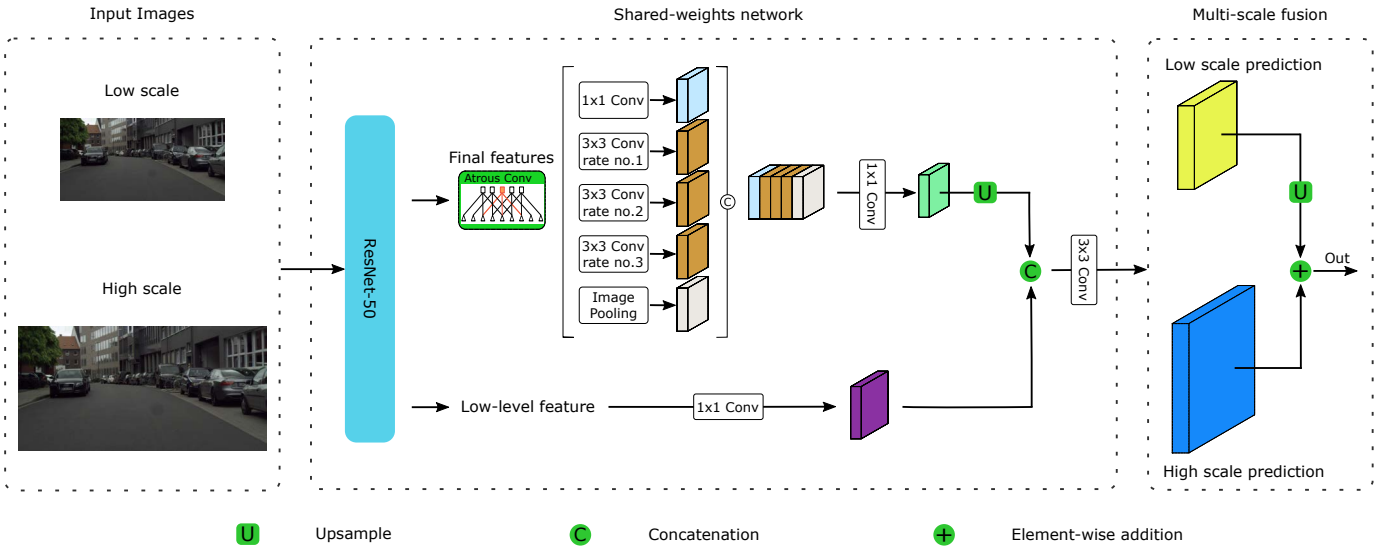


Fig. 2: Structural diagram of the proposed dual-inferences mechanism for real-time semantic segmentation.

by a factor of 2 to have the same size of a high scale prediction. A single pixel of a low scale is combined with a corresponding pixel from a high scale to generate final values.

$$\mathcal{S} = \sum_{i=0}^{C} \sum_{j=0}^{H*W} (Hi_{ij} + Lo_{ij}) \tag{1}$$

where $\mathcal{S}$, Hi and Lo are the final segmentation, a high scale features, and a low scale features, respectively. H, W, and C are dimensions of height, width, and class numbers. i and j are instant location of pixels.

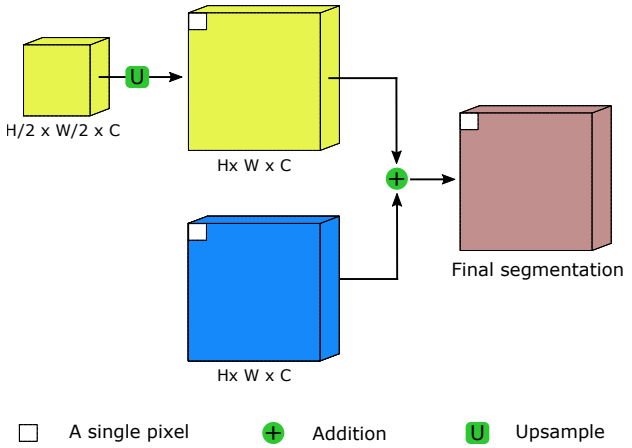


Fig. 3: Feature fusion of dual inferences at pixel-wise levels. The yellow box represents a low scale and the blue box is a high scale.

C. Optimization

We utilize a Cross-entropy as the default loss function. The function calculate the average difference between the ground truth and predicted probability for all classes shown in equation 2.

$$L = -\frac{1}{N} \sum_{n=1}^{N} y_n \, log(\hat{y}_n) \tag{2}$$

where L and N are the loss value and the size of dataset, respectively. y_n denotes the actual segmentation probability, and $\hat{y}_i$ denotes the predicted probability.

The intersection of union (IoU) metric validate the number of pixels between the target and prediction masks divided by the total number of pixels present across both masks.

$$IoU = \frac{TP}{TP + FP + FN} \tag{3}$$

where TP, FP, and FN denote truth positive, false positive, and false negative respectively.

Stochastic Gradient Descent (SGD) is utilized as our optimizer. SGD algorithms minimizes the loss function of a predicted model on a training process and update for each training example x_i and ground truth y_n. The SGD is calculated as equation 4.

$$\theta = \theta - \eta \nabla_\theta \mathcal{J}(\theta; x_i; y_i) \tag{4}$$

where ∇_θ denotes gradient of the objective function and $\mathcal{J}(\theta)$ is an objective function.

IV. Experiments

In our experiments, we set up 150 epochs for training and the number of batch sizes is 2 per GPU. The others parameters are a momentum of 0.9, the weight decay of $5e{-}4$. The polynomial learning rate is utilized with an initial learning rate of 0.01. We use the standard measures of semantic segmentation to improve the training model such as stochastic gradient descent (SGD) and cross-entropy loss function. Other standards are deployed to evaluate the performance including intersection of union (IoU) and fame per second (FPS). On

TABLE II: Comparison of our approach and existing methods on the Cityscapes with respect to class accuracy.

Method	Road	swalk	build	wall	fence	pole	tlight	tsign	veg.	terr	sky	pers	rider	car	truck	bus	train	mcle	bicle	mIoU
AGLNet [20]	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
CGNet [7]	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 [21]	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 [22]	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
CFPNet [23]	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
FSCNN [24]	97.4	77.8	87.4	39.7	41.8	35.0	39.4	50.5	88.5	63.3	92.7	65.7	46.4	91.0	57.0	70.3	56.5	40.9	52.6	62.8
DABNet [25]	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 [23]	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 [26]	98.9	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 [27]	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
Ours	97.1	83.9	91.8	55.3	58.7	61.3	64.4	75.6	91.8	64.3	93.0	78.9	58.9	94.3	81.2	86.5	70.0	62.2	74.2	75.5

TABLE III: Performance and speed comparison of our approach and existing methods on Cityscapes.

Method	Resolution	mIoU	FPS
AGLNet [20]	512×1024	70.1	52
TCNet [28]	1024x2048	74.6	16
ICNet [29]	1024x2048	67.7	38
BiseNet [30]	768×1536	74.8	47
SwiftNet [31]	1024×2048	75.4	40
Ours	1024x2048	75.5	51

Cityscapes dataset, we train the model with an Nvidia Titan X. The configurations of Titan X have 12 GB of GDDR5X memory. For Mappilary vistas, the large dataset requires more memory to calculate, so we train on a GeForce RTX 3090 with 24GB GPU and G6x memory.

A. Cityscapes dataset

Cityscapes is a traffic scene dataset which is collected from 50 different countries. The whole dataset includes 5,000 images with resolution 1024x2048. It is divided into three subsets such as training with 2975 images, validation with 500 images, and test with 1525 images. The data contains 19 object categories.

In Table II, we analyze the detail of class accuracy. All approaches achieve high accuracy for stuff classes involving road, building, vegetation, and sky. Only cars is thing class and have good results because objective size is suitably captured by the receptive fields. For large objects of such trucks, buses, or trains, our method is much higher accuracy than the others by deploying a low scale branch with large receptive fields. The results also illustrate the improvement for thin structure class. By utilizing dual inferences with different fields of views, our method has good results for thin and large classes.

In Table III, we demonstrate the trade-off between performance and inference speed. Our study accomplishes the results with 75.9 mIoU and 51 FPS. For the accuracy aspect, we surpass all approaches. Although the SwiftNet [31] achieve the highest segmentation accuracy among existing methods, our study still exceed them, and especially the speed only reaches 40 FPS. The AGLNet [20] with 52 FPS has less time consumption than our approach, but they only achieve 70.1 mIoU. The results show that our proposal not only has high performance but also achive real-time application. The qualitative results are demonstrated in Fig. 4a.

TABLE IV: Accuracy comparison of our approach and existing SOTA methods on Mapillary Vistas.

Method	Resolution	mIoU
AGLNet [20]	1024x2048	30.7
DABNet [25]	1024x2048	29.6
RGPNet [32]	1024×2048	41.7
Ours	2177x1632	42.0

B. Mapillary Vistas

Mapillary Vistas is a large semantic dataset with 65 objective classes. This data is collected from street scenes around the world. The whole set has 20,000 images and is splitted into the training set with 18,000 images and the validation set with 2,000 images. Mapillary dataset contain a wide range of resolutions.

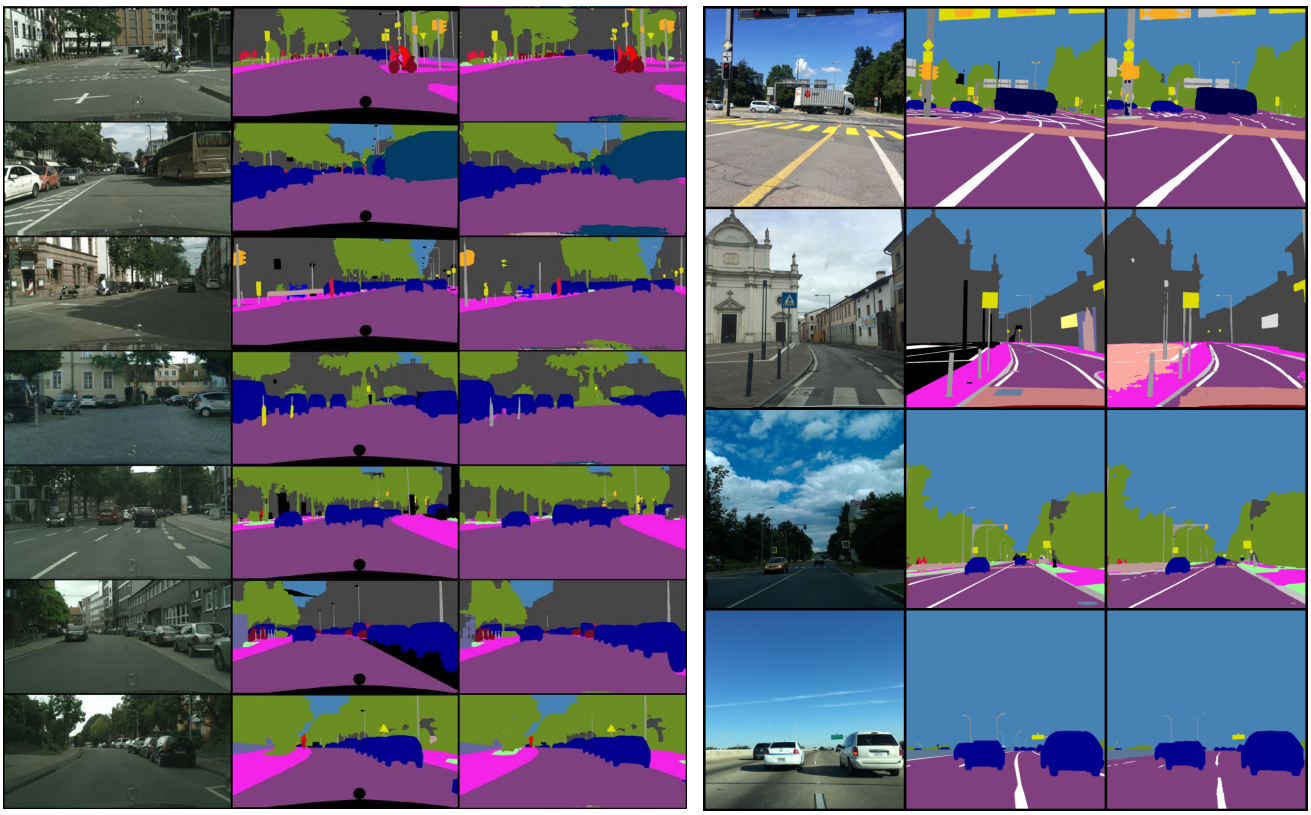
Mapillary is a complex dataset, so all studies approach to improve the performance. Our method outperforms other existing methods with 42.0 mIoU shown in Table IV. We exceed 10% mIoU better than DABNet [25] and AGLNet methods [20]. The qualitative results are illustrated in Fig. 4b. Despite of low accuracy, the visualization shows good results for necessary classes such as cars, traffic signs or people.

V. Conclusion

In this paper, multiple inferences can capture a wide range of objective sizes in street scenes. The low scale has good results for large classes while high scale effectively segments narrow objects. Additionally, different sets of dilation rates help enhance the advantages of specific inference. the results show that the segmentation accuracy is outperform and the processing time is short. Due to the hardware limitation, only dual inferences were implemented for our proposed model. In the future, we intend to increase numbers of input scales in order to improve the performance.

Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1A2C2008133) and by Basic Science Research Program through the National Research Institute



(a) cityscapes (b) Mapillary Vistas

Fig. 4: Qualitative results of our proposal on Cityscapes dataset and Mapillary Vistas dataset.

Foundation of Korea (NRF) funded by the Ministry of Education(2021R1A6A1A03043144) .

REFERENCES

[1] W. Zhang, C. Zhou, J. Yang, and K. Huang, "Liseg: Lightweight road-object semantic segmentation in 3d lidar scans for autonomous driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1021–1026.

[2] X. Li, Z. Liu, P. Luo, C. Change Loy, and X. Tang, "Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3193–3202.

[3] Q. Ning, J. Zhu, and C. Chen, "Very fast semantic image segmentation using hierarchical dilation and feature refining," *Cognitive Computation*, vol. 10, no. 1, pp. 62–72, 2018.

[4] Y. Dai, J. Wang, J. Li, and J. Li, "Mdrnet: a lightweight network for real-time semantic segmentation in street scenes," *Assembly Automation*, 2021.

[5] W. Xiang, H. Mao, and V. Athitsos, "Thundernet: A turbo unified network for real-time semantic segmentation," in *2019 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2019, pp. 1789–1796.

[6] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Efficient convnet for real-time semantic segmentation," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1789–1794.

[7] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2018.

[8] A. Sáez, L. M. Bergasa, E. Romeral, E. López, R. Barea, and R. Sanz, "Cnn-based fisheye image real-time semantic segmentation," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1039–1044.

[9] T. Emara, H. E. Abd El Munim, and H. M. Abbas, "Liteseg: A novel lightweight convnet for semantic segmentation," in *2019 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2019, pp. 1–7.

[10] I. Alonso, L. Riazuelo, and A. C. Murillo, "Mininet: An efficient semantic segmentation convnet for real-time robotic applications," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1340–1347, 2020.

[11] Q. Zhou, W. Yang, G. Gao, W. Ou, H. Lu, J. Chen, and L. J. Latecki, "Multi-scale deep context convolutional neural networks for semantic segmentation," *World Wide Web*, vol. 22, no. 2, pp. 555–570, 2019.

[12] J. Zhang, S. Lin, L. Ding, and L. Bruzzone, "Multi-scale context aggregation for semantic segmentation of remote sensing images," *Remote Sensing*, vol. 12, no. 4, p. 701, 2020.

[13] D. Lin, Y. Ji, D. Lischinski, D. Cohen-Or, and H. Huang, "Multi-scale context intertwining for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 603–619.

[14] J. He, Z. Deng, and Y. Qiao, "Dynamic multi-scale filters for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3562–3572.

[15] 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.

[16] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[17] 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.

[18] Q. Van Toan and M. Y. Kim, "Multi-scale synergy approach for real-time

semantic segmentation," in *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2022, pp. 216–220.

[19] G. Gao, G. Xu, Y. Yu, J. Xie, J. Yang, and D. Yue, "Mscfnet: a lightweight network with multi-scale context fusion for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[20] 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.

[21] 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.

[22] 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.

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

[24] R. P. Poudel, S. Liwicki, and R. Cipolla, "Fast-scnn: Fast semantic segmentation network," *arXiv preprint arXiv:1902.04502*, 2019.

[25] 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.

[26] 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.

[27] 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.

[28] Z. Wu, C. Shen, and A. v. d. Hengel, "Real-time semantic image segmentation via spatial sparsity," *arXiv preprint arXiv:1712.00213*, 2017.

[29] 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.

[30] 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.

[31] 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.

[32] 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.