Attention and Feature Fusion Network for Real-time Semantic Segmentation

Yu Cheng

***Abstract*—The real-time semantic segmentation technology has great application potential in the current mobile robot navigation and the autonomous driving fields. Pixel-wise classification accuracy and inference speed are the key evaluation metrics of real-time semantic segmentation. However, modern approaches usually sacrifice the inference speed for the high classification accuracy by building a very deep network. A mini convolutional network is proposed in this project to solve this paradox, which includes a set of symmetric Attention and Feature Fusion (AAFF) blocks to provide a large receptive field and achieve features fusion. The depth-wise convolution is also applied to reduce the number of parameters, this network can achieve faster inference speed than previous networks without the loss of pixel-wise classification accuracy. Further research about attention mechanism and feature fusion has great potential in the field of real-time semantic segmentation.**

***Keywords*—semantic segmentation, pixel-wise classification, inference speed, convolutional network, Attention and Feature Fusion, receptive field.**

I. INTRODUCTION

The real-time semantic segmentation technology is an important research task in computer vision [1], which is mainly used to represent the surrounding environment by building semantic maps in real-time. In semantic maps, different categories are usually represented by different colors, and each color represents a semantic label. It has significant implications in some industrial areas, such as the autonomous driving [2] and the robot semantic navigation [3].

The research on the inference speed and accuracy on semantic segmentation are the key components to solve the current difficulties in autonomous driving and other related fields. Inference speed refers to the speed with which semantic maps are built, and pixel-wise classification accuracy refers to the ratio of correctly classified pixels to the total number of pixels.

In the past several years, the efficiency of semantic segmentation has been widely studied. However, a large number of researches are based on only the pixel-wise classification accuracy consideration, use a very deep convolutional network to improve the accuracy, which led to the very low inference speed. On the contrary, some works prune the channels of their original network or reduce the layers directly without further optimization [4]. But a very basic network is usually difficult to have high pixel-wise classification accuracy.

The Attention and Feature Fusion (AAFF) block has been proposed in this paper to solve this problem. This block contains an attention mechanism and a feature fusion mechanism, which is an improvement for the previous method [5]. The attention mechanism enables the deep layers to pay more attention to some useful information, and the feature fusion mechanism allows the fusion of features from different layers to enhance the generalization ability of this model. The depth-wise convolution [6] is also applied to speed up the inference procedure. As a result, the inference speed is about 5 frames per second (fps) faster than the previous network [2] without accuracy sacrifice. The detailed description is shown in part Ⅲ and part Ⅳ.

The research of this paper provides a solution to the contradiction between the semantic segmentation accuracy and the inference speed. Further research based on the evidence and approaches proposed in this paper has great potential to achieve the industrial-level application.

II. EXISTING TECHNIQUES FOR SEMANTIC SEGMENTATION

The new era of semantic segmentation begins in 2015, because the Fully Convolutional Networks (FCN) [7] greatly improves the accuracy of semantic segmentation. Using convolutional layer to replace the traditional fully-connected layer can greatly reduce the number of parameters and retains the spatial information of images, makes the pixel-wise classification possible. However, the total number of parameters of FCN is still too large to achieve real-time segmentation.

Based on FCN [7], SegNet [4] and UNet [8] increased the symmetry of the network and introduced the semantic feature fusion mechanism respectively, which is an improvement of FCN [7]. SegNet [4] performs feature extraction through the encoder, and then recover the feature image size to the input size by using the decoder. The prune operation has been applied to boost the inference speed lead to the broken of the spatial information. UNet [8] introduced the feature fusion mechanism based on SegNet [4], which improved the generalization ability and pixel-wise classification accuracy to a certain extent but still failed to solve the low inference speed problem.

ICNet [9] attempts to accelerate the inference by resizing the input image to reduce the training parameters. The smallest input image is cut to 1/4 of the original one, this operation has caused the loss of spatial details and resulting in the reduction of pixel-wise classification accuracy. ENet [10] abandoned the downsampling operation in the final stage of the feature extraction, which made the receptive field of the model not big enough to cover the global feature of the input image.

BiseNet [5] is a new approach to solve the problems above, the Context path and the Spatial path are used to provide the global receptive field and enough spatial information respectively, which achieved good results in some public datasets. The research of this project is based on this two-path method, and a new AAFF block is proposed to improve the overall performance of the previous network.

III. Three Branch Network

The overall layout of the network proposed in this project is shown in Fig.1. This network has three main branches and two main functional modules, the left and right branches can obtain a large receptive field and fuse the features from different layers to improve the generalization ability and pixel-wise classification accuracy. The vertical branch in the middle part applies the depth-wise convolution [6], which can greatly reduce the training parameters and greatly improve the inference speed of this network.

A. Depth wise convolution for inference speed

The traditional convolution is usually applied to extract features. After a traditional convolution, each pixel in the feature image carries the region features of the size of a convolution kernel in the original image (receptive field). For example, a convolution operation with a 3x3 kernel size could obtain a 3x3 receptive field. In general, the larger the receptive field, the higher the final pixel-wise classification accuracy. However, the traditional convolution kernel has a large number of training parameters, which will increase the computational complexity greatly. Therefore, the use of traditional convolution will have a negative impact on inference speed. Equation 1 [11] shows the training parameters needed for a traditional convolution.

$$\begin{aligned} input\_size &= I_w \times I_h \times I_c \\ Kernel\_size &= n \times n \times I_c \\ training\_parameters &= n \times n \times I_c \times O_c \end{aligned} \tag{1}$$

Where O_c represents the output channels, I_c represents the input channels. As a summary, the total training parameters needed for a n x n convolution is $n^2O_cI_c$. For example, a 500x500x3 RGB image input wants to output a 256-channels feature map after the traditional convolution with a kernel size of 5x5, the number of training parameters is 5x5x3x256 = 19,200.

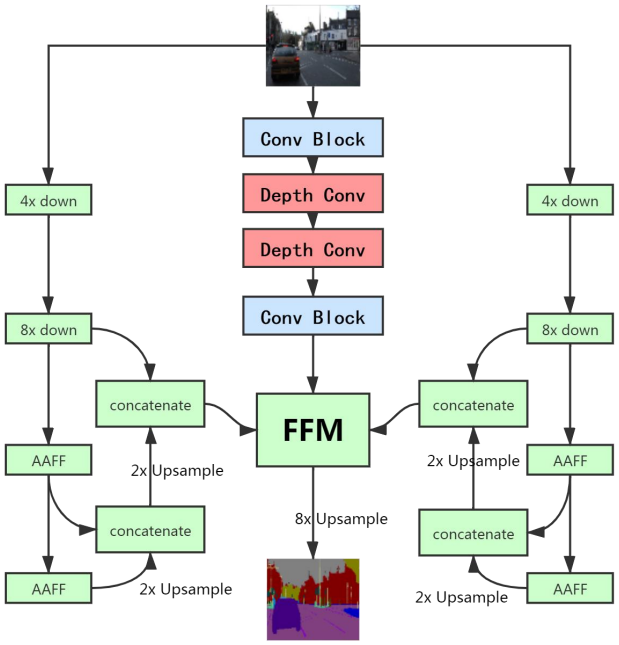

Fig.1. The overall architecture

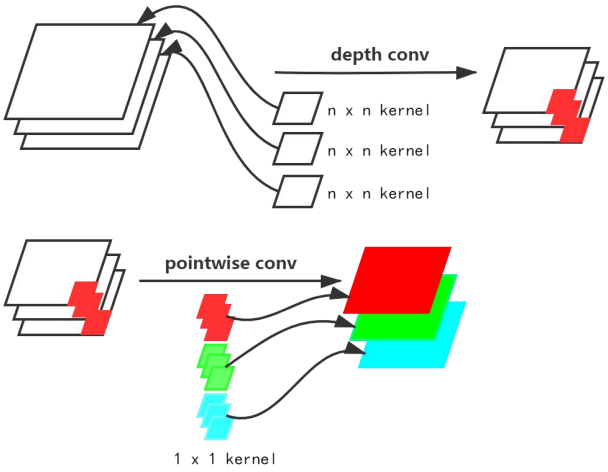

Fig.2. The depth-wise convolution [6]

In order to reduce the training parameters, the depth-wise convolution [6] has been applied in this project. Fig.2. shows the concepts of the depth-wise convolution. Two continuous convolutions are applied to change the input size and the channels respectively. The use of 1x1 convolution [6] in stage 2 can greatly reduce the number of parameters.

$$\begin{aligned} input\_size &= I_w \times I_h \times I_c \\ depth\_Kernel\_size &= n \times n \times I_c \\ point\_Kernel\_size &= 1 \times 1 \times I_c \times O_c \\ training\_parameters &= n \times n \times I_c + 1 \times 1 \times I_c \times O_c \\ &= \left(n^2 + O_c\right) I_c \end{aligned} \tag{2}$$

For the same situation of the example above, the number of training parameters is (25+256) x 3 = 843. About 23 times less than the traditional one. In this project, the kernel size used in the vertical branch was 3x3, which can accelerate the inference process in the final stage.

B. Attention and Feature Fusion block

Attention mechanism could be applied in semantic segmentation to increase the learning ability of the network, for example, the application of attention modules in [12] [13] have achieved good classification accuracy in some public dataset. Based on some previous observations, this paper proposed the attention and feature fusion module, which can automatically learn to perceive the importance of different channels. Because it will endow different weights according to the degree of importance, these different weights will be multiplied by the feature maps of the previous layer, which can make the neural network pay more attention to these channels with big weights while ignoring those with small weights, the learning ability of the network thus will be improved a lot. The symmetric structure has some AAFF blocks, which can average out most of the loss, and increase the generalization ability of the network, the structure shows in Fig.3.

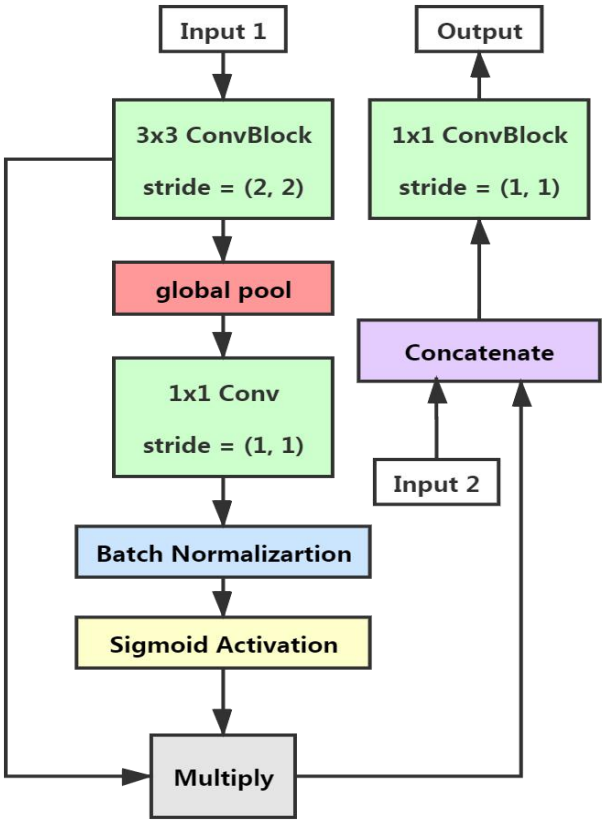

Fig.3. The Attention and Feature Fusion block (AAFF)

There are only three convolution operations in the AAFF block, and two convolutions with the kernel size 1x1 are applied to modify the channels of the feature map. Most of the training parameters come from the first 3x3 convolution, but there is only one 3x3 convolution operation in this block, therefore, the total number of parameters in a single block will be very small, from this perspective, AAFF blocks can accelerate the training and inference speed to some extent.

The loop on the left provides a global receptive field and attention. Firstly, the input images or feature maps are down-sampled to half of the original size by a convolution of 3x3 with strides of 2. The following global pooling compresses the input into a weight vector, for each channel of the input, global pooling operation can result in a weight to represent the importance of that channel, for some useful channels, such as these channel with a lot of objects information of the original image, the output weight for that channel is a big value, and the reverse is true.

Therefore, pooling operation can give the network a mechanism of attention, allowing it to focus on finding and learning those distinctive channels while ignoring those that don't have much useful information. After these subsequent standardized operations like 1x1 convolution, batch normalization and nonlinear activation, the weights vector will be multiplied by the previous feature layers, and because they have the same number of channels, each of these feature layers will be multiplied by only one of the weights, so the feature layers with high weights will be retained for the next round of training, and the feature layers with low weights will be eliminated.

The loop on the left can not only introduce the attention mechanism but also provide a large receptive field by global pooling operation. And the larger the receptive field is, the more global information of the input image can be obtained, the more global information it has, the higher the pixel-wise classification accuracy. Therefore, the global pooling on the left side can also improve the pixel-wise classification accuracy to some extent.

But the application of this attention mechanism brings the prejudice issue, which may cause the network to pay too much attention to the so-called useful channels and completely ignore some of the less obvious channels. In order to solve this kind of prejudice or bias, the feature fusion mechanism has been applied in the right-side. Input 2 contains very comprehensive information, which includes some background information that is not very distinctive. The features after attention mechanism will concatenate with the input 2 and output some very good features.

C. The overall working process

The input images from the public dataset will be processed synchronously by these three branches after some pre-processing. As mentioned before, the middle branch applied the depth-wise convolution [6] instead of traditional convolution, which greatly reduced the training parameters and speeds up the inference process of the network. A 3x3 conv Block will output the feature maps with the size of 1/2 of the input images, then these two depth-wise down-sampling operations will reduce the input size for 4 times, finally, a convolution Block is applied to limit the number of output channels, the output feature maps is 1/8 the size of the input images, which retains enough spatial information because the original input image has not been cropped substantially.

The symmetric branches on the left and right are mainly used to obtain a large receptive field and introduce the attention to this network, which can greatly improve the accuracy of pixel-wise classification in principle. The backbone of this network is xception [6], because xception [6] has good performance in image classification, it can be used as the backbone to extract the features of the input image. The 4x down and 8x down illustrated in Fig.1. are the primary features of the input image extracted after passing through the xception [6] backbone network. Two successive AAFF blocks are then applied on the deeper network layer, each with two input feature maps, coming from different convolutional layers. After the attention and feature fusion operations, features with the global receptive field will be output, and these features will be integrated through the up-sampling operation, as shown in Fig.1., the concatenate block is only applied to make a fusion of different features. Finally, features from three different branches are merged through a Feature Fusion Module (FFM) [5], and the pixel-wise classified images are output after a 8 times up-sampling operation. Therefore, the output image has the same size as the input image.

The proposed network uses the CamVid [14] dataset as the input, which is the same dataset used in BiseNet [5], and experimented with different activation functions and feature fusion methods in the AAFF block. The results showed that different networks have different pixel-wise classification accuracy and inference speed, the specific results will be shown in part Ⅳ.

IV. Results Visualization

The training and test results are mainly embodied in these three aspects: 1. Train process visualization 2. Inference results visualization 3. Test and validation results visualization. But different aspects are visualized in different ways, the line chart was applied to visualize the training process, with the increase of the epoch, the training accuracy gradually increased, as shown in Fig.4. Subsequently, semantic maps is applied to visualize the inference results, as shown in Fig.5. Table 1. is applied to present the detailed segmentation results. Appendix E [15] provides additional results and data.

A. Training process visualization

This is the result of training for four different networks, the original BiseNet is the implementation result of the BiseNet network proposed in [5]. However, for some realistic reasons, the training epoch in this project is not big enough, obviously, the accuracy of these networks is still improving, so the network is not 100% trained. The size of the input images in this project is 800x640, not the 960x720 in [5]. These two factors may result in a lower final accuracy than the original paper. But this result can still be used to make a localization comparison, because the four networks are trained in the same training configuration, so they can be compared with each other based on the same hyper-parameters.

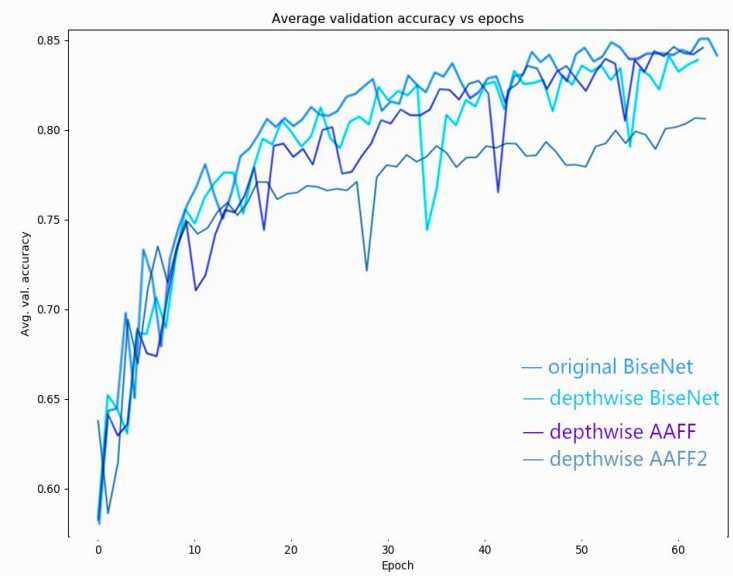

Fig.4. Training results for different networks

The original BiseNet achieved about 85% pixel-wise classification accuracy on CamVid [14] dataset for epoch 60, this is the highest one among them, and if the spatial path in [5] is changed and the original convolution is replaced by the depth-wise convolution, a very similar accuracy result will be obtained. If AAFF blocks are applied on the basis of depth-wise convolution, another similar accuracy will be achieved, about 84%. Finally, a modification of the AAFF block is applied but result in a poor classification accuracy, as shown in the depthwise AAFF2 in Fig.4.

B. Inference results visualization

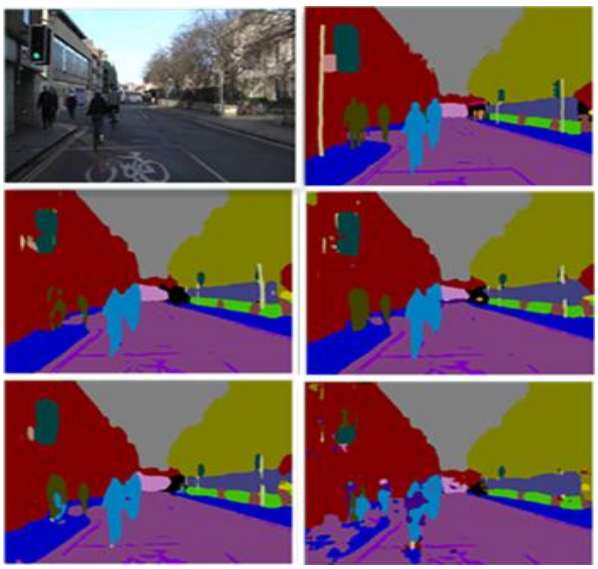

Fig.5. Inference results for different networks

The inference results for a sample image is shown in Fig.5., the input image is in row1+column1, and the ground truth image is in the right side of the input image. The inference results for original BiseNet, depth-wise BiseNet, depth-wise AAFF and depth-wise AAFF2 were displayed at row2+column1, row2+column2, row3+column1 and row3+column2 respectively.

C. Test and validation results visualization

Table 1. Accuracy for different networks

Method / Acc. for	Original BiseNet	Depthwise BiseNet	Depthwise AAFF	Depthwie AAFF2
Building	0.92	0.84	0.83	0.85
Tree	0.63	0.67	0.65	0.67
Sky	0.96	0.97	0.96	0.96
Car	0.76	0.73	**0.79**	0.79
Road	0.94	0.94	**0.97**	0.94
Pedestrian	0.19	0.26	**0.34**	0.15
Fence	0.19	0.28	0.19	0.10
Traffic Light	0.27	0.26	**0.26**	0.22
Sidewalk	0.67	0.62	0.56	0.53
Average	0.85	0.84	0.84	0.80
Test	0.71	0.62	0.71	0.80
fps	25	35	31	27

Table 1. shows the segmentation results of nine different categories, and their pixel-wise classification accuracy is presented, they all show their talent in big objects recognition. The average classification accuracy for different networks and the test results in the test dataset were also provided.

V. Analysis of the Results

The training and test results can be compared across four different networks, as mentioned in part Ⅳ, the training process does not converge completely but is close to converge. But the training configuration used for these different networks are identical, so they can be compared locally. The experimental results will be analyzed from the following four perspectives in this chapter.

A. The learning ability analysis

The learning ability of the network is mainly reflected in the accuracy of learning. For example, for a semantic segmentation network, the stronger learning ability, the higher classification accuracy in the validation dataset, because the feature distribution of the validation dataset and the training dataset is usually very similar.

As shown in Fig.4., the first three networks shown good pixel-wise classification accuracy in the validation dataset, which means that they all have good learning ability. If training for more epochs and use the 960x720 input images, the final classification accuracy may converge to about 95%. The learning ability of the final network (depthwise+AAFF2) is significantly lower than the first three. It is obvious that the new network (depthwise+AAFF) proposed in this project is almost the same as the network (original BiseNet) [5] in terms of the learning ability. According to the data presented in table 1., the classification accuracy of this new network for some big objects exceeded that of BiseNet [5], such as road and car. Although the recognition ability of some small objects is not as good as that of BiseNet [5], such as traffic light and side walk, which might be one of the weaknesses of this network, but overall, the learning ability of the new network is not worse than BiseNet [5].

B. The generalization ability analysis

The generalization ability refers to the migration ability of a network. A good network should be able to perform well on dataset with different feature distributions. If high pixel-wise classification accuracy is only achieved on training and validation dataset, but low accuracy is achieved on test dataset, then the network can be thought of as over-fitting, because this network learned both the features and the variations.

As can be seen in table 1., the refined BiseNet with depth-wise convolution [6] has a very poor migration ability for the reason that the pixel-wise classification accuracy on the test dataset is only about 62%, which is 22% worse than the result obtained on the validation dataset. The best performer on the test dataset is the depthwise+AAFF2 network, because this network has a close pixel-wise classification accuracy on both the validation dataset and the test dataset, and it achieved the highest accuracy on the test dataset compared to other networks. Unfortunately, this network has the worst learning ability and the inference speed is slow, which will be discussed later. It is obvious that the new network depthwise+AAFF proposed in this project has the same pixel-wise classification accuracy (71%) on the test dataset as the previous BiseNet [5]. And their learning ability is are similar, therefore, their generalization ability is similar, but the feature distribution of the test dataset is slightly different from that of the validation dataset, so they both have an accuracy drop of about 10%. The results can be better if more epochs can be trained and use bigger input size.

C. The inference speed analysis

The inference speed of the network is another important metric for measuring the semantic segmentation performance of the network, especially in the real-time semantic segmentation. For most of the existing semantic segmentation networks, 30 frames per second (fps) is the threshold for real-time segmentation, but few existing networks can exceed this threshold, even if they use a much better GPU than the one used in this project. The GPU used in this project is an NVIDIA GeForce 840M, which is not a good GPU when compared with other projects. In [5] the higher fps is achieved by using NVIDIA Titan X.

The depth-wise BiseNet achieved about 35 fps in this project, and it is the highest one of these four networks, the reason is that two depth-wise convolutions [6] are applied to replace the traditional convolution in [5], which greatly reduces the number of training parameters and significantly speed up the inference. Besides, another two networks proposed in this project also have better inference performance than [5], at least with an improvement of about 5 fps. The acceleration of the inference speed in this project does not sacrifice too much accuracy, but is realized by applying the depth-wise convolution [6] on the premise of ensuring similar pixel-wise classification accuracy.

D. The comprehensive analysis

According to the previous experimental results and the analysis of the components, the performance of these networks can be evaluated comprehensively. If the traditional convolution is simply replaced by the depth-wise convolution [6], the inference speed will be greatly improved, but the generalization ability of the network will be weakened, which is specifically reflected in the gap between the pixel-wise classification accuracy in the validation dataset and the test dataset. AAFF block can improve the generalization ability of the networks. Therefore, they can be combined with the depth-wise convolution [6] to achieve a faster inference speed without loss of pixel-wise classification accuracy, this is also one of the core work of this project. In addition, a comparison experiment (depthwise+AAFF2) was designed, and the results show that the network with the AAFF block can indeed have good generalization ability, but the number of training parameters of this network is still large, which makes the slow inference speed. Only the third network (depthwise+AAFF) has performed well in both the pixel-wise classification accuracy and the inference speed for semantic segmentation, which demonstrates the feasibility of this design for the combination of depth wise convolution [6] and AAFF blocks.

VI. Conclusion

This project proposed an AAFF block based on the attention and feature fusion mechanism, with the intention to improve the pixel-wise classification accuracy and generalization ability of the previous network [5]. However, the implementation results show that the AAFF block has no obvious effect on semantic classification accuracy, but can improve the generalization ability of the network, as shown in depthwise+AAFF2. In order to accelerate the inference speed of the network, this project applied the depth-wise convolution [6] to replace the traditional convolution, which greatly reduced the number of training parameters and ultimately accelerating the inference speed for about 5fps. These two approaches above can be combined to speed up the inference speed of the network and improve the generalization ability without too many costs of the pixel-wise classification accuracy. The depthwise+AAFF network is the best example of this combination.

In the future, this project is likely to focus on maximizing the performance of the network with better hyper-parameter configurations and may do further works on AAFF block to improve pixel-level classification accuracy.

References

[1] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, 'A Review on Deep Learning Techniques Applied to Semantic Segmentation', *arXiv:1704.06857 [cs]*, Apr. 2017, Accessed: Aug. 27, 2020. [Online]. Available: http://arxiv.org/abs/1704.06857.

[2] B. Chen, C. Gong, and J. Yang, 'Importance-Aware Semantic Segmentation for Autonomous Driving System', in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia, Aug. 2017, pp. 1504–1510, doi: 10.24963/ijcai.2017/208.

[3] J. Lin, W.-J. Wang, S.-K. Huang, and H.-C. Chen, 'Learning based semantic segmentation for robot navigation in outdoor environment', in *2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS)*, Otsu, Japan, Jun. 2017, pp. 1–5, doi: 10.1109/IFSA-SCIS. 2017.8023347.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla, 'SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation', *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[5] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, 'BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation', *arXiv:1808.00897 [cs]*, Aug. 2018, Accessed: Aug. 22, 2020. [Online]. Available: http://arxiv.org/abs/1808.00897.

[6] F. Chollet, 'Xception: Deep Learning with Depthwise Separable Convolutions', *arXiv:1610.02357 [cs]*, Apr. 2017, Accessed: Aug. 22, 2020. [Online]. Available: http://arxiv.org/abs/1610.02357.

[7] E. Shelhamer, J. Long, and T. Darrell, 'Fully Convolutional Networks for Semantic Segmentation', *arXiv:1605.06211 [cs]*, May 2016, Accessed: Aug. 22, 2020. [Online]. Available: http://arxiv.org/abs/1605.06211.

[8] O. Ronneberger, P. Fischer, and T. Brox, 'U-Net: Convolutional Networks for Biomedical Image Segmentation', *arXiv:1505.04597 [cs]*, May 2015, Accessed: Aug. 22, 2020. [Online]. Available: http://arxiv.org/abs/1505.04597.

[9] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, 'ICNet for Real-Time Semantic Segmentation on High-Resolution Images', *arXiv:1704.08545 [cs]*, Aug. 2018, Accessed: Aug. 22, 2020. [Online]. Available: http://arxiv.org/abs/1704.08545.

[10] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, 'ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation', *arXiv:1606.02147 [cs]*, Jun. 2016, Accessed: Aug. 22, 2020. [Online]. Available: http://arxiv.org/abs/1606.02147.

[11] Madhivarman, 'How to calculate the number of parameters in the CNN?', *Medium*, May 30, 2018. https://medium.com/@iamvarman/how-to-calculate-the-number-of-parameters-in-the-cnn-5bd55364d7ca (accessed Aug. 27, 2020).

[12] J. Hu, L. Shen, and G. Sun, 'Squeeze-and-Excitation Networks', 2018, pp. 7132–7141, Accessed: Aug. 22, 2020. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html.

[13] H. Li, P. Xiong, J. An, and L. Wang, 'Pyramid Attention Network for Semantic Segmentation', *arXiv:1805.10180 [cs]*, Nov. 2018, Accessed: Aug. 22, 2020. [Online]. Available: http://arxiv.org/abs/1805.10180.

[14] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, 'Segmentation and Recognition Using Structure from Motion Point Clouds', in *Computer Vision – ECCV 2008*, vol. 5302, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 44–57.

[15] Y. Cheng, "Attention and Feature Fusion Network for Real-time Semantic Segmentation", MEng Final Portfolio, Appendix E - Test & Results, August 2020.