Indoor Semantic Segmentation for Robot Navigating on Mobile

Wonsuk Kim
School of Electrical Engineering
Korea University
Seoul, South Korea
won425@korea.ac.kr

Junhee Seok
School of Electrical Engineering
Korea University
Seoul, South Korea
jseok14@korea.ac.kr

***Abstract*—In recent years, there have been many successes of using Deep Convolutional Neural Networks (DCNNs) in the task of pixel-level classification (also called "semantic image segmentation"). The advances in DCNN have led to the development of autonomous vehicles that can drive with no driver controls by using sensors like camera, LiDAR, etc. In this paper, we propose a practical method to implement autonomous indoor navigation based on semantic image segmentation using state-of-the-art performance model on mobile devices, especially Android devices. We apply a system called 'Mobile DeepLabv3', which uses atrous convolution when applying semantic image segmentation by using MobileNetV2 as a network backbone. The ADE20K dataset is used to train our models specific to indoor environments. Since this model is for robot navigating, we re-label 150 classes into 20 classes in order to easily classify obstacles and road. We evaluate the trade-offs between accuracy and computational complexity, as well as actual latency and the number of parameters of the trained models.**

Keywords—Semantic Segmentation; Convolutional Neural Networks; Atrous Convolution; Indoor Navigation

I. INTRODUCTION

Nowadays, semantic image segmentation is more and more being of interest for computer vision and machine learning researchers. Many applications, such as autonomous driving [1][2][3], medical application[4], augmented reality system[5], robotics[6], and indoor navigation[7], require accurate and efficient semantic image segmentation algorithms. Advances in Deep Convolutional Neural Networks (DCNNs) have brought tremendous progress in semantic image segmentation. In order to achieve higher accuracy, the neural networks used in modern state of the art networks are getting deeper and require high computational resource[8]. However, when it comes to common mobile platforms, such as drones, robots, and smartphones, the recognition tasks need to be carried out in a resource constrained environment.

There have been several approaches to make neural networks more efficient with respect to size and speed [9]. For example, SqueezeNet[10] achieves AlexNet[11]-level accuracy with 50x fewer parameters by compression techniques. MobileNet[12] uses depthwise seperable convolutions to build light weight DCNNs. NASNet[13], the program automates the design of machine learning models, is capable of producing small DCNNs which performed as well as those designed by humans while having very low computational costs. ShuffleNet [14] reduces computation complexity of 1x1 convolutions through pointwise group convolutions. MobileNetV2[15] shows state-of-the-art performance for mobile applications by using inverted residuals and linear bottlenecks based on MobileNetV1.

In this paper, we present an application of semantic image segmentation specific to indoor navigation for a mobile robot. We trained mobile semantic segmentation model through a reduced form of Deeplabv3[16], a memory-efficient and high-performance architecture on the PASCAL VOC 2012 dataset, by using MobileNetV2 as a network backbone. The ADE20K dataset [17] is used to train models and Tensorflow android library is used to build mobile Android application. We examine the relationship between the inference time and the number of parameters and computational cost of the trained models. We also evaluate the trade-offs between accuracy and computational complexity. We make our application publicly available at HTTPS://GITHUB.COM/WONDERIT/INDOOR-SEGMENTATION-ANDROID

II. 'MOBILE DEEPLABV3' FOR SEMANTIC SEGMENTATION

A. System Structure

There are many network implementations based on encoder-decoder architectures for semantic segmentation tasks. FCNs[18], SegNet[19], UNet[20], RefineNet[21] are some of the most popular ones. Different from these encoder-decoder designs, DeepLab[22] offers a different approach to semantic segmentation. Atrous convolution, shown in Fig. 1-(a), allows us to effectively enlarge the field of view of filters without increasing the number of parameters. In Deeplabv2[23], multiscale processing is achieved by passing multiple rescaled versions of original images to parallel CNN branches (Image pyramid) and using multiple parallel atrous convolutional layers with different sampling rates (ASPP). The proposed module in DeepLabv3 consists of atrous convolution in cascade or in parallel with various rates and batch normalization layers, shown in Fig. 1-(b). This module has attained comparable performance (85.7% on the PASCAL VOC 2012 test set) with other state-of-art models.

This research was supported by a grant from Korea Evaluation Institute of Industrial Technology (10073166). The correspondence should be addressed to jseok14@korea.ac.kr

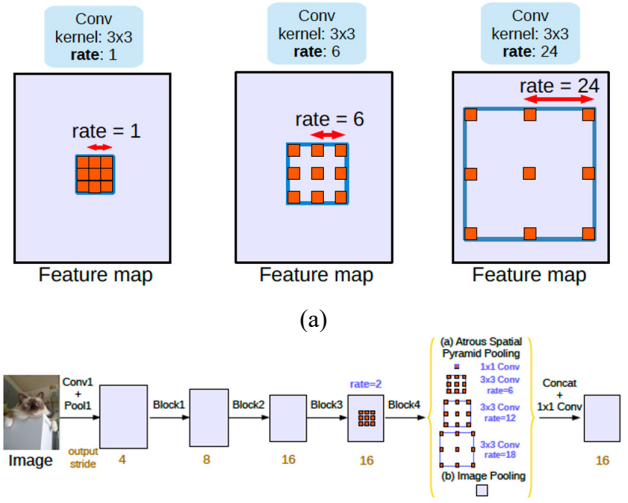

Fig. 1. (a) Atrous convolution, also called dilated convolution, with kernel size 3x3 and different rates. The first convolution (rate=1) is actually the ordinary convolution. (b) Parallel modules with Atrous Spatial Pyramid Polling(ASPP) in DeepLabv3 system.

B. Network Backbone

MobileNetV1 is based on a streamlined architecture that uses depthwise seperable convolutions to build light weight DCNNs as shown in Fig.2-(a). MobileNetV2 is a significant improvement over MobileNetV1 and pushes the state of the art for mobile visual recognition including classification, object detection and semantic segmentation. MobileNetV2 introduces two new features to the architecture: 1) linear bottlenecks between the layers and 2) shortcut connections between the bottlenecks shown in Fig.2-(b).

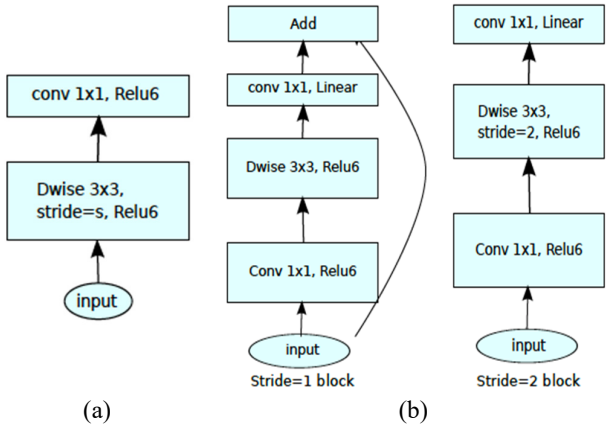

Fig. 2. The basic structure of MobileNetV1(a) and MobileNetV2(b).

C. Dataset

We used ADE20K dataset instead of other datasets, such as cityscapes [24], PASCAL VOC2012 and COCO, because ADE20K is more suitable for indoor environment than any other datasets. We also choose this dataset because DeepLabv3 system using MobileNetV2 was not benchmarked. There are 20,210 images in the training set, 2,000 images in the validation set, and 150 classes are annotated. We trained with 20,210 images of training set and evaluated our model with 2,000 images of validation set.

III. Experiment Results

We trained over 0.1M, 1M iterations by using Tensorflow. With NVIDIA GTX 1080Ti GPU, the training time took 0.28 sec/step with crop size of [513, 513] and 0.1 sec/step with crop size of [257, 257]. Additionally, we trained models by using Xception[25] as a network backbone for performance comparison. Training time took 0.88 sec/step with crop size of [513, 513] and 0.35 sec/step with crop size of [257, 257].

We created an Android application based on Tensorflow sample application, which overlay semi-transparent images of the real-time semantic image segmentation in the smartphone's camera. In order to use this app for indoor navigation, we have re-labeled the 150 predicted classes. Table 1 shows how we re-labeled the classes. Besides those 6 classes listed in Table 1, we used 14 more classes (sky, ceiling, grass, pavement, people, door, mountain, water, picture, lounge, house, mirror, carpeting, and others).

TABLE I. Re-labeled classes for indoor navigation

Class	Re-labeled classes
Wall	9(window), 33(fence), 43(pillar), 44(sign board), 145(bulletin board)
Floor	7(route), 14(ground), 30(field), 53(path), 55(runway)
Tree	18(plant), 67(flower), 73(tree), 126(flowerpot), 136(vase)
Furniture	8(bed), 11(cabinet), 16(table), 19(pall), 20(chair), 25(shelf), 34(desk)
Staircase	54(steps), 122(stair), 97(stairway)
Box	51(icebox), 90(box), 139(bin)

The example result images from the application are shown in Fig. 3. We found that the model of large crop size of [513, 513] is more accurate than the model of small crop size of [257, 257]. This is because larger crop size is required for atrous convolution with large rates to be effective; otherwise, the filter weights with large atrous rate are mostly applied to the padded zero region. But the difference is too small to choose a model that requires high computational costs.

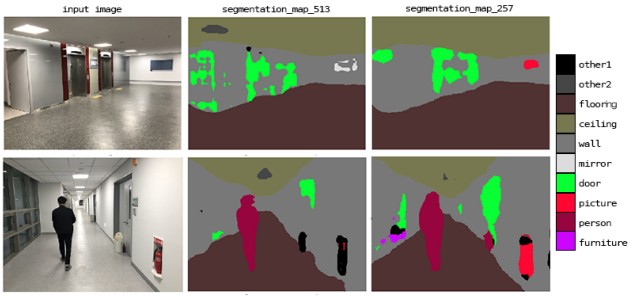

Fig. 3. Test results in a practical real-time environment.

Fig. 4 shows the performance comparison of crop size and the number of iterations through per-class mIOU. We found that the model trained with crop size [257, 257] with 1 million iterations has better performance than the model trained with crop size [513, 513] with 0.1 million iterations. Since the classes which mIOU is greater than 50% is more than 10 classes and the performance difference between the model of small crop size and large crop size is very small, we thought that we can use the model with small crop size with many iterations.

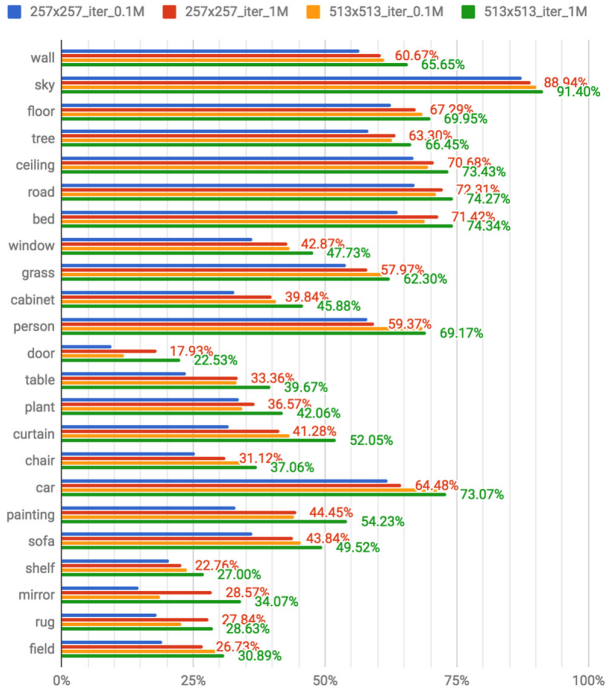

Fig. 4. Performance (mIOU) comparison per some classes and trained models.

Table 2 shows the processing time in milliseconds and computational complexity for each model with different crop size and network. We optimized the model to include only the actual nodes needed to run the prediction process. The inference time is based on the results of running on Samsung Galaxy S8. We could achieve 4x faster inference time by reducing 4x crop size of the model.

Table 3 shows the performance for each model with different crop size and iterations. The result shows that, despite of small crop size, similar performance can be achieved with many iterations. So we decided to choose the model with bold text for our application.

TABLE II. COMPUTATIONAL COMPLEXITY COMPARISON

Model	CropSize	FLOPs	Params	Inference Time
			Normal/Optimized	*Normal/Optimized*
Mobile NetV2	257x257	4.63B	2.18M / 2.15M	1480ms / 1185ms
	513x513	17.96B	2.18M / 2.15M	5452ms / 4175ms
Xception	257x257	85.86B	39.27M / 39.13M	N/A
	513x513	333.20B	39.27M / 39.13M	N/A

TABLE III. PERFORMANCE ON ADE20K

Model	CropSize	Iteration	Accuracy	mIOU
MobileNetV2	257x257	100K	64.92%	15.64%
	257x257	**1M**	**68.94%**	**22.93%**
	513x513	100K	69.40%	21.16%
	513x513	1M	72.08%	27.00%

IV. CONCLUSION

In this paper, we proposed a practical indoor navigation method for mobile device with no need for any additional computing device (e.g. laptop, cloud server). Our experimental results show that using MobileNetV2 is 3 times faster than using Xception in training time and 20 times faster at inference time. By reducing the crop size four times, we can also reduce the inference time by four times and training time by 2.7 times. We will further study how to reduce training and inference time of the model and how to improve the performance of the model.

REFERENCES

[1] Chen, Chenyi, et al. "Deepdriving: Learning affordance for direct perception in autonomous driving." Computer Vision (ICCV), 2015 IEEE International Conference on. IEEE, 2015.

[2] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3234–3243.

[3] M. Siam, S. Elkerdawy, M. Jagersand, and S. Yogamani, "Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges," arXiv preprint arXiv:1707.02432, 2017

[4] Kayalibay, Baris, Grady Jensen, and Patrick van der Smagt. "CNN-based segmentation of medical imaging data." arXiv preprint arXiv:1701.03056 (2017).

[5] O. Miksik, V. Vineet, M. Lidegaard, R. Prasaath, M. Nießner, S. Golodetz, S. L. Hicks, P. P´erez, S. Izadi, and P. H. Torr, "The semantic paintbrush: Interactive 3d mapping and recognition in large outdoor spaces," in Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. ACM, 2015, pp. 3317–3326.

[6] A. Valada, G. L. Oliveira, T. Brox, and W. Burgard, "Deep multispectral semantic scene understanding of forested environments using multimodal fusion," in The 2016 International Symposium on Experimental Robotics (ISER 2016), 2016.

[7] Y. Li and S. Birchfield, "Image-based segmentation of indoor corridor floors for a mobile robot," IEEE/RSJ Int. Conf. Intelligent Robot. Syst., Taipei, Taiwan, 2010, pp. 837–843

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In CVPR, 2015.

[9] He, Kaiming and Jian Sun. "Convolutional neural networks at constrained time cost." 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015): 5353-5360.

[10] Iandola, Forrest N., et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size." arXiv preprint arXiv:1602.07360 (2016).

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[12] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017

[13] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. CoRR , abs/1707.07012, 2017.

[14] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. CoRR , abs/1707.01083, 2017

[15] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: Inverted Residuals and Linear Bottlenecks. arXiv preprint. arXiv:1801.04381, 2018.

[16] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. CoRR , abs/1706.05587, 2017

[17] Zhou, Bolei, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso and Antonio Torralba. "Scene Parsing through ADE20K Dataset." 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017): 5122-5130.

[18] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015

[19] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv:1511.00561, 2015

[20] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In MICCAI, 2015

[21] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multipath refinement networks with identity mappings for highresolution semantic segmentation. arXiv:1611.06612, 2016

[22] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," in ICLR, 2015

[23] 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. arXiv:1606.00915, 2016

[24] 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 CVPR, 2016

[25] Chollet, François. "Xception: Deep learning with depthwise separable convolutions." arXiv preprint (2016)