Embedded Semantic Segmentation Network Optimized for Matrix Multiplication Accelerator

Jaeyoung Lee

***Abstract*—**Autonomous driving systems require high reliability to provide people with a safe and comfortable driving experience. However, despite the development of a number of vehicle sensors, it is difficult to always provide high perceived performance in driving environments that vary from time to season. The image segmentation method using deep learning, which has recently evolved rapidly, provides high recognition performance in various road environments stably. However, since the system controls a vehicle in real time, a highly complex deep learning network cannot be used due to time and memory constraints. Moreover, efficient networks are optimized for GPU environments, which degrade performance in embedded processor environments equipped simple hardware accelerators. In this paper, a semantic segmentation network, matrix multiplication accelerator network (MMANet), optimized for matrix multiplication accelerator (MMA) on Texas instrument digital signal processors (TI DSP) is proposed to improve the recognition performance of autonomous driving system. The proposed method is designed to maximize the number of layers that can be performed in a limited time to provide reliable driving environment information in real time. First, the number of channels in the activation map is fixed to fit the structure of MMA. By increasing the number of parallel branches, the lack of information caused by fixing the number of channels is resolved. Second, an efficient convolution is selected depending on the size of the activation. Since MMA is a fixed, it may be more efficient for normal convolution than depthwise separable convolution depending on memory access overhead. Thus, a convolution type is decided according to output stride to increase network depth. In addition, memory access time is minimized by processing operations only in L3 cache. Lastly, reliable contexts are extracted using the extended atrous spatial pyramid pooling (ASPP). The suggested method gets stable features from an extended path by increasing the kernel size and accessing consecutive data. In addition, it consists of two ASPPs to obtain high quality contexts using the restored shape without global average pooling paths since the layer uses MMA as a simple adder. To verify the proposed method, an experiment is conducted using perfsim, a timing simulator, and the Cityscapes validation sets. The proposed network can process an image with 640 x 480 resolution for 6.67 ms, so six cameras can be used to identify the surroundings of the vehicle as 20 frame per second (FPS). In addition, it achieves 73.1% mean intersection over union (mIoU) which is the highest recognition rate among embedded networks on the Cityscapes validation set.

***Keywords*—**Edge network, embedded network, MMA, matrix multiplication accelerator and semantic segmentation network

I. Introduction

GENERAL autonomous driving systems use sensors to recognize driving conditions, determine vehicle direction and speed, and control in real time. Self-driving cars require high reliability because malfunctions can threaten human life. However, despite the development of a number of vehicle sensors, such as RADAR, camera, ultrasonic sensors, and LiDAR, it is difficult to always provide high perceived performance in a variety of driving environments that vary from time to season. Thus, most mass-production vehicles remain in the driving assistance phase, and original equipment manufacturers have achieved level 3 only in limited driving conditions.

Improving driving environment recognition performance is necessary to overcome the limits of the reliability of the autonomous driving system. Of the vehicle sensors, cameras are essential for autonomous driving systems because they are most similar to the way people perceive them and can provide information such as lanes, signs and traffic lights. In particular, deep learning algorithm developed recently can provide higher perceived performance than the limits of classical recognition methods, instead of using high computations. Since the autonomous driving system is a vehicle embedded system for real-time control, it should meet the power consumption, semiconductor reliability, latency, throughput and price conditions. However, high-end deep learning networks cannot satisfy these conditions. Moreover, since the typical embedded network is optimized for GPU or ARM environments, the performance is degraded in processors with simple accelerators.

In this paper, we propose MMANet optimized for MMA module, TI's embedded hardware accelerator, to improve the perceived performance of the autonomous driving system. It uses three methods to optimize the network structure for the MMA computation of TDA4V-MID processors. First, we configure the branch to minimize double data rate (DDR) memory access by fixing the number of activation channels and use only L3 cache for internal operation. Secondly, depending on the size of the activation map, depthwise separable convolutions (DSC) or normal convolutions are selected to increase the network's expressive power. Finally, the correct contexts are extracted using extended ASPP. 5x5 and 7x7 convolution branches are added for stable operation. In addition, the second-stage ASPP is used to decode shape using previous contexts. Since this network can process images 640 x 480 for 6.67 ms, it can use six cameras to recognize 360 degrees around the vehicle in real time (20 FPS). It also achieves 73.1 mIoU, the highest accuracy rate in embedded networks, on the Cityscapes validation set.

This paper is composed as follows. In Section II, efficient embedded network implementation methods are introduced and explained the limitations when applied to the MMA environment. MMANet is proposed in Section III for MMA in

Jaeyoung Lee is with the Hyundai Mobis Co., Ltd, 16891, Yongin-si, Republic of Korea (phone: 82-10-9946-7831; fax: 303-5720-2058; e-mail: ljy@mobis.co.kr).

TDA4V-MID DSP. In addition, experiments are described using Cityscapes dataset in Section IV. Finally, we conclude in Section V.

TABLE I
THE NUMBER OF MULTIPLICATIONS AND LAYERS ACCORDING TO CONVOLUTION METHODS

Layer	k_x	k_y	Group	Processing Time (us)		#Multiplication (M)		# Layers (per 8ms)
Convolution	3	3	1	60.75	60.75	168.75	168.75	132
DSC	3	3	64	32.47	49.84	2.64	21.39	161
	1	1	1	17.37		18.75		
CP-Decomposition	3	1	64	15.12	94.22	0.88	39.26	85
	1	1	1	17.52		18.75		
	1	3	64	44.06		0.88		
	1	1	1	17.52		18.75		

The processing time is simulated by using TI perfsim by setting the size of activation maps to 80 x 60 and the number of input and output channels to 64. k_x and k_y means kernel width and height respectively.

II. RELATED WORK

In this section, we introduce how to lower the number of computations and improve the recognition performance of networks. To optimize networks, convolution decomposition, context extraction and efficient layers are explored. Lastly, the development status of semantic segmentation networks for embedded systems is summarized.

A. Convolution Decomposition

A typical 3 x 3 convolution layer uses all input channels to calculate one output channel, resulting in high computations and parameter usage [1]. The DSC is decomposed into depthwise convolution that performs operations on each channel and pointwise convolution for exchange of information between the channels to reduce the number of computations and parameters by the number of input channels [2]. CP decomposition reduces complexity by dissolving 3 x 3 convolution into 3 x 1 and 1 x 3 by reducing 2-dimensional kernel to one dimension [3]. However, the lower complexity layer also reduces the amount of information expressed, and the complexity and processing time are not proportional depending on HW accelerators as shown in Table I. Therefore, in the TDA4V-MID environment, decomposition can reduce the processing time of a single layer, but it is less computationally efficient.

B. Efficient Structure

Convolutional neural networks (CNN) process information by reducing the size of the feature step by step to extract the image context. ResNet introduced the skip connection firstly, facilitating the flow of gradient, enabling deep network implementation [4]. UNet fuses low level features and context by transporting the each stage outputs of encoder to decoder [5]. The size of receptive field has an effect on extracting contexts, pyramid spatial pooling (PSP) and ASPP uses multiple sizes or dilated convolutions respectively [6], [7]. In general, CNN preserves the amount of information by increasing the number of channels as the feature size decreases [8]. However, since the overall processing time is fixed, as the number of channels increases, the time to process a single layer increases, which reduces the network depth as given in Fig. 1.

C. Effective Layer

Normal CNN consists of convolution, batch normalization and ReLU. MobileNet has increased nonlinearity by applying the ReLU6 and H-Swish functions as an activation function [9], [10]. CGNet employs PReLU to learn about negative features and channel attentions by using sigmoid [11]. The deformable convolution shows robust performance in the geometric transformation of an object by changing the reference position of the convolution kernel depending on the image position [12]. However, in an embedded environment, these layers consume a lot of time or memory because they cannot use hardware accelerators. Moreover, for most hardware accelerators, the ReLU function is provided in combination with convolution, so no additional processing time is required.

D. Embedded Semantic Segmentation Model

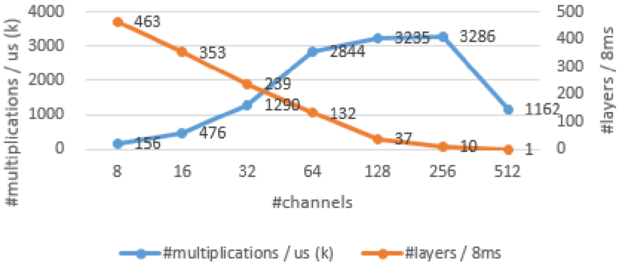


Fig. 1 The number of multiplications and layers according to 3 x 3 convolution channels when the activation map is 80 x 60

The general semantic segmentation network consists of encoder for extracting context and decoder for restoring local shape, so it has twice the complexity of classification network. SegNet and UNet have symmetrical auto encoder structures, but DeepLab v3+ simplified decoder structures to increase efficiency. To make complex semantic segmentation networks operate on edge devices, ENet uses pointwise convolution to minimize the number of channels and CP-decomposition to reduce the computations [13]. ESPNet used variant ASPP structures as basic blocks to efficiently extract contexts even with small depth [14]. SQNet and CGNet use parallel convolution structures to fuse features to each layer, while FPENet reduces complexity by splitting the channel [15], [16]. FastSCNN offers a cost effective solution by increasing the number of channels and improving learning methods instead of reducing the repetition of blocks [17].

TABLE II
MMANET REQUIREMENTS

Group	Specification	Value
Requirement	Image size	640, 480
	Camera	6EA
	FPS	20
	Weight	7,936kB
Target	Processing time (85%)	7.08ms
	Weight size (60%)	4761kB
	#Layers	> 150

III. METHOD

This paper proposes MMANet, a semantic segmentation network that provides information about 360° environment with six camera sensors to improve the reliability of the autonomous driving system. To control the vehicle in real time using the recognition results, the elapsed time must be less than 50 ms, and the size of the parameters must be less than the L3 cache capacity. Detailed specifications for network design are shown in Table II. 15% processing margin is allocated for camera fault detection network. In addition, 40% of L3 cache is used for data temporal storage to minimize DDR memory access.

Because the high-end network is located in the recognition rate saturation area, the perceived performance changes as the computation increases are small. However, for embedded networks with 1 TFLOPS or less, recognition performance is also improved as the computations increase. Thus, special layers that take a long time can be replaced by a number of simple layers. In particular, if the optimal layers and structures for hardware accelerators are used, a network with the same elapsed time can be implemented with the maximum number of computations. This section briefly introduces the structure of MMA, the hardware accelerator of TDA4V-MID. Also, we suggest a way to design CNNs optimized for MMA.

A. MMA

C7x DSP in TDA4V-MID performs vector multiplication in MMA by reading weights and activation maps stored in the L3 cache. MMA performs vector multiplication by reading 64 data and weights per one cycle, as in (1). Therefore, 4096 multiplication and addition are performed in one cycle, so it provides 8TOPS computational capability when the DSP operates at 1 GHz. In addition, since offset operations and RELU are implemented as hardware, one can handle convolution, batch normalization and rectified linear unit (ReLU) at one time, as shown in Fig. 2.

$$[c_1 \quad c_2 \quad \cdots \quad c_{64}] = [a_1 \quad a_2 \quad \cdots \quad a_{64}] \times \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,64} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,64} \\ \vdots & \vdots & \ddots & \vdots \\ b_{64,1} & b_{64,2} & \cdots & b_{64,64} \end{bmatrix} \qquad (1)$$

where a means weights, b means activation maps and c is layer outputs.

MMA cannot predict the elapsed time with simple MAC or FLOPS. If MMA is used for simple multiplication and addition operations, such as ALU, only 64 mac operations can be performed per a cycle. Moreover, memory access time cannot be ignored since consecutive operations are used to implement one operation if the size of the activation map is greater than 64. In this paper, a TI simulator called perfsim is used to predict the exact processing time in the TDA4V-MID environment by setting the L3 cache to 7,936 kB and the data type to 8 bits.

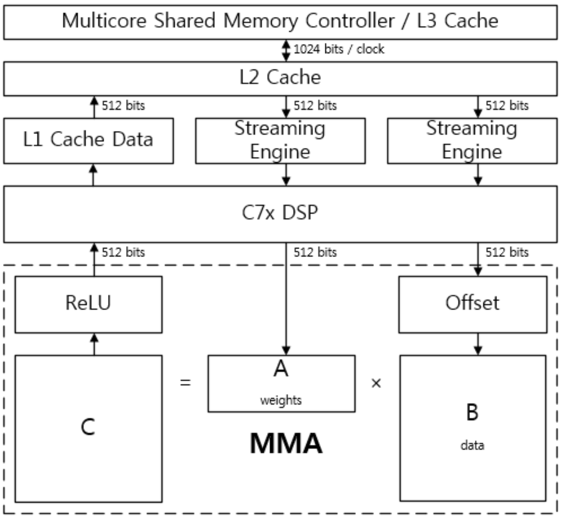


Fig. 2 The MMA of TDA4V-MID

B. Entry Block

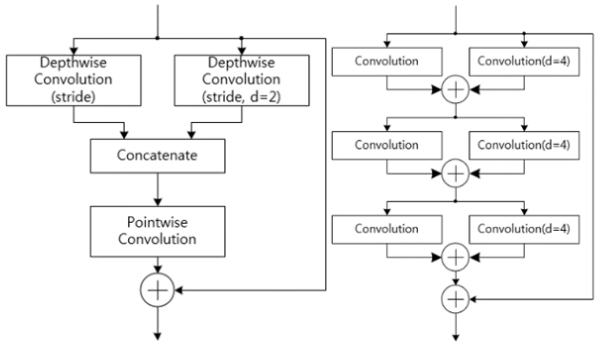
The first convolution layer of the embedded networks sets stride to two to reduce the size of the activation map and increase the number of channels to maintain the amount of information. In general, the network decreases image size in a step-by-step to extract contexts, but increases the number of channels to keep the amount of information. In addition, it provides input of various sizes in parallel to supplement information lost during the abstraction process However, since the input image is stored in DDR memory, the more layers that access the image, data input and output time increase exponentially.

In MMANet, the activation map of the first convolution output is used instead of the input image to minimize DDR memory access and produce multiple input size effects. Fig. 1 shows that it is difficult to preserve the amount of information by increasing the number of channels since feasible depth is inversely proportional to channels. Therefore, parallel paths with output stride 2, 4, 8, and 16 are implemented to replenish information volumes. In addition, 1 x 1 convolution is used minimize storage capacity in L3 cache. If the input size is larger, normal convolution is more efficient than DSC as shown in Table III since memory access overhead is increased. Therefore, the first and second layers are implemented as a normal convolution.

TABLE III
COMPARISON BETWEEN NORMAL CONVOLUTIONS AND DSC ACCORDING TO INPUT SIZE

Input Size	Parameters	Convolution		DSC		Decision
		#Multiplication [M]	Time [us]	#Multiplication [M]	Time [us]	
640 x 480	3, 16, s=2	32	88	5 (17%)	120 (136%)	Convolution
320 x 240	16, 29, s=2	76	73	11 (15%)	64 (88%)	Convolution
160 x 120	32, 32	169	115	24 (14%)	64 (56%)	DSC
80 x 60	64, 64	169	61	21 (13%)	50 (82%)	DSC
40 x 30	64, 64	42	21	5 (13%)	22 (109%)	Convolution

The parameters mean input and output channels in sequence.
s represents the stride value.



(a) Level 3 module (b) Level 4 module

Fig. 3 The body modules of MMANet (d represents the dilation factor)

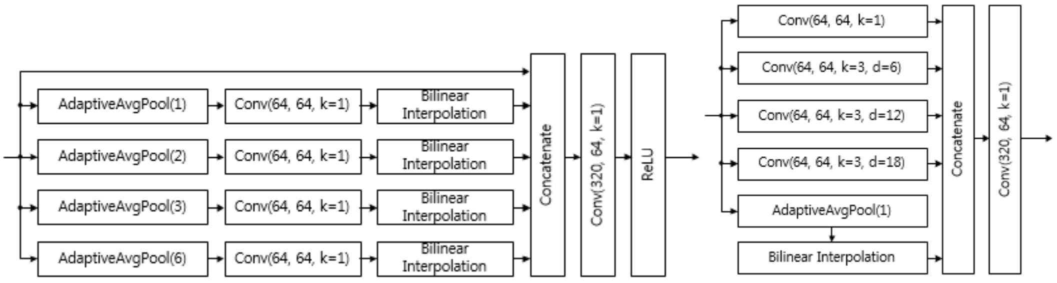
C. Body Block

If the size of the activation map is greater than one-quarter (level 2) of the input size, it is difficult to increase the network's depth because more than 64us is used to process one layer. MMANet uses the parallel layers which have a different field of view (FOV) to help extracting contexts. Thus, it can obtain highly abstracted information. In addition, it changes processing modules depending on the size of activation maps to increase the number of layers. The Level 3 (1/8th of the input size) module used the surrounding context extraction part of CGNet is shown in Fig. 3 (a). In level 4, because of the large overhead associated with pointwise convolution, features are extracted by placing the parallel layer that uses the different dilation value as given in Fig. 3 (b).

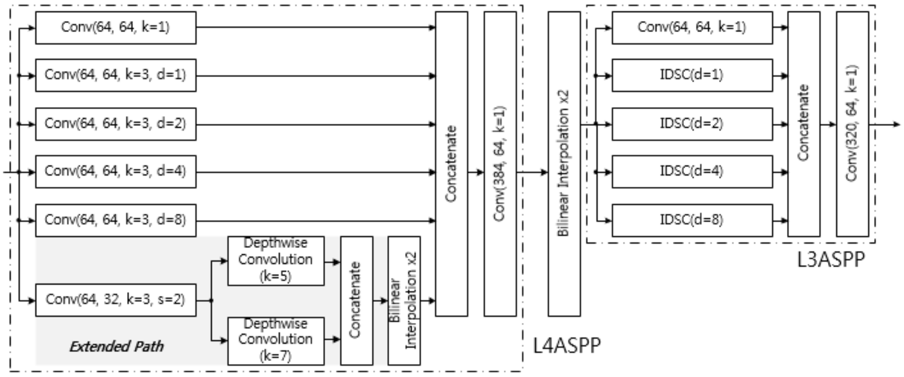
D. Extended ASPP

For encoder, which extracts the image's characteristics only with a convolution layer, context is difficult to identify because the surrounding values that can be referenced in a single layer are limited. To obtain more accurate context values from the correlation with nearby values, the pyramid specific pooling (PSP) and atrous specific pyramid pooling (ASPP) are proposed. Since the average value represents neighboring values, PSP obtains the representative value by dividing the activation map into 1, 2, 3 and 6 sections as shown in Fig. 4 (a). However, since this pooling is simple addition, MMA is used as ALU, which makes it less efficient. ASPP extracts features by changing the dilation value of the convolution to 1, 6, 12 and 18, and weights to use pointwise convolution, as shown in Fig. 4 (b). However, ASPP also has a global average pooling branch. In addition, it is difficult to extract stable contexts like PSP since the kernel does not refer to continuously located data.

In this paper, an extended ASPP, consisting solely of convolution, is proposed to maximize the use of MMA. The proposed method is constructed by adding an extended path to ASPP to get stable features by increasing the kernel size and accessing consecutive data as shown in Fig. 4 (c). In addition, ASPP is implemented consecutively in level 3 (one-eighth the size of the input image) to obtain high quality contexts using the restored shape. ASPP used in level 3 is configured with inverted depthwise separable convolution (IDSC) that employs pointwise convolution first to reorganize level 4 ASPP information and extract context again.



(a) PSP (b) ASPP



(c) Extended ASPP

Fig. 4 Decoder context extraction methods (k, s and d means the size of the kernel, the stride and dilation value respectively. Conv means convolution. IDSC is an inverted DSC that two convolution order is changed.)

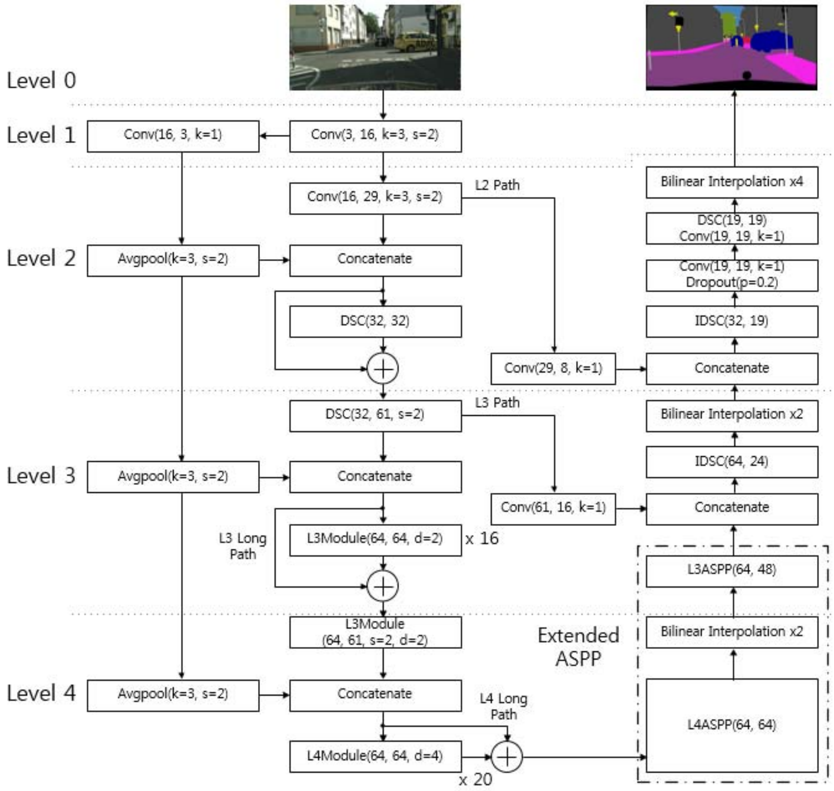


Fig. 5 The proposed MMANet architecture

E. MMANet

One basic layer of MMANet is composed of convolution, batch normalization, and ReLU. Since the elapsed time of MMA is not proportional to the computational amount as shown in Table III, a layer between normal convolution and DSC are selected depending on the activation map size to achieve high computational efficiency. Repeated modules are employed for feature extraction at image size less than 1/8 and FOVs configured in parallel to effectively extract contexts.

To process layers as much as possible in a limited time, the maximum number of channels are limited as 64. The insufficient information is supplemented by increasing the parallel path as shown in Fig. 5. In addition, 40% of the L3 cache is allocated to store the activation map of the parallel path, and the extended ASPP is used in the decoder to increase the recognition performance.

IV. Experiment

In this section, we evaluate the proposed MMANet on the Cityscapes validation dataset. First, the data set is introduced with training protocols. Then, our contributions are validated through ablation studies. Finally, it is compared and analyzed with other embedded networks.

A. Experimental Setting

Cityscapes Dataset: The Cityscape data set is 5,000 road images stored in 50 different cities. The color images of 2048 x 1024 are spitted into 2,975 training set, 500 validation set and 1,525 test set. High accuracy semantic segmentation and instance segmentation ground truths are provided, and for semantic segmentation, recognition performance is usually measured using mIOU.

B. Training Protocol

All experiments are conducted on 2 x Titan XP using PyTorch, and single-scale images are used to calculate the mIOUs on Cityscapes validation set. The polynomial learning rate policy with power 0.9 is employed. The initial ringing rate sets to 0.045 without pre-trained weights. In addition, SGD optimizer with weight decay 4e-5 is used, and training is performed during 1000 epochs with batch size 20. For data augmentation, random crop, random horizontal flip and random scale with 0.5, 0.75, 1, 1.5, 1.75 and 2.0 are used. Moreover, color jitter and lighting noise are added. To weight hard pixels, OHEM is used with a threshold 0.7 and 100,000 pixels.

TABLE IV
MULTIPLE PATH EFFECTIVENESS

Conditions	mIoU (%)
MMANet	73.1
MMANet w/o L2 path	72.9 (0.2↓)
MMANet w/o L3 path	63.0 (10.1↓)
MMANet w/o L3 long path	70.8 (2.3↓)
MMANet w/o L4 long path	67.7 (5.4↓)

TABLE V
EXTENDED ASPP EFFECTIVENESS

Conditions	mIoU (%)
MMANet	73.1
MMANet w/o L3ASPP	70.7 (2.4↓)
MMANet w/o L4ASPP extended path	69.8 (3.3↓)
MMANet w/o L4ASPP	67.0 (6.1↓)
MMANet w/o Extended ASPP	62.4 (10.7↓)

C. Ablation Study

Ablation Study for Multiple Paths: Since MMANet sets the maximum number of channels to 64 to increase the number of layers; the amount of information per step is limited. Multiple paths can help to preserve recognition performance with the small number of channels by sharing processed information by step by step. Since there are many layers in level 3 and 4, the recognition performance is dropped by 10.1% when the level 3 path between encoder and decoder is removed as shown in Table IV. In addition, the level 2 path increases the recognition accuracy by 0.3% by providing the shape information.

Ablation Study for Extended ASPP: We use extended ASPP to extract contexts from the correlation between features and surrounding values. L4ASPP employs an extended path to extract high-level features stably by using the continuous information like PSP. Table V shows that it has an effect on increasing 3.3% accuracy. When removing L3ASPP composed of IDSCs, the recognition performance is reduced by 2.4%. On the other hand, without L4ASPP, mIoU is reduced by 6.1%. Therefore, it is a key component of decoder to refine contexts. In addition, when the entire extended ASPP is removed, the performance drops from 73.1% to 62.4%.

TABLE VI
PARALLEL DILATED CONVOLUTION EFFECTIVENESS

Conditions	mIoU (%)
MMANet	73.1
Level 3 serial configuration	72.9 (0.2↓)
Level 4 serial configuration	63.0 (10.1↓)

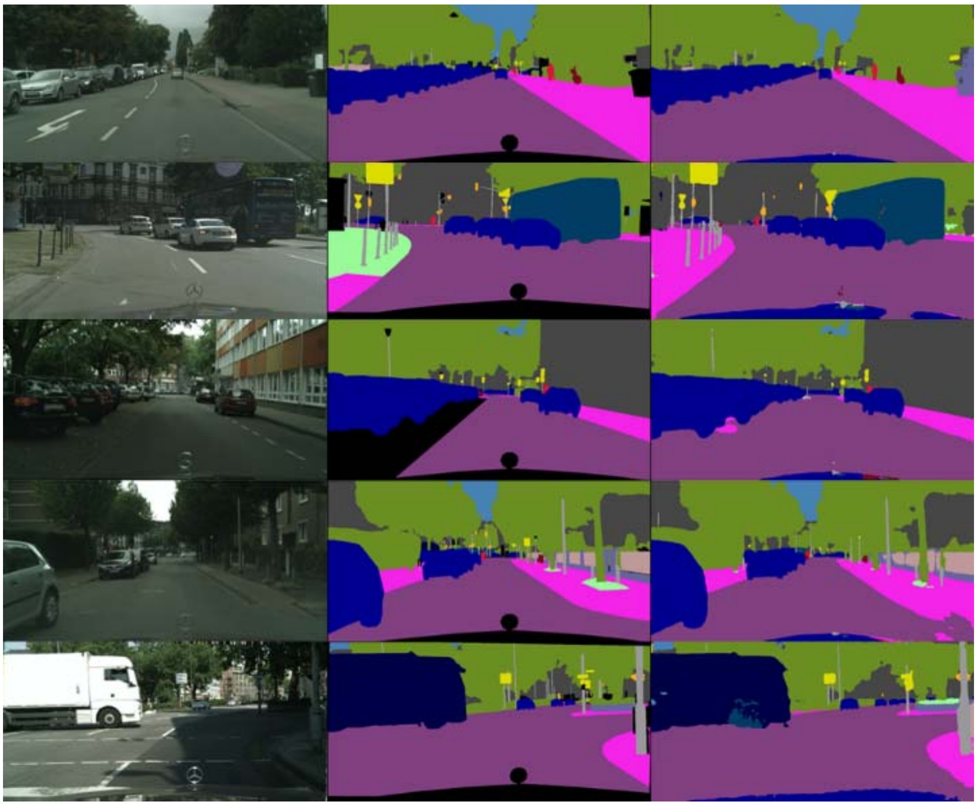
When the configuration is changed from parallel to serial, the dilation factor sets to 1.

Ablation Study for Parallel Dilated Convolution: A parallel path using a dilated convolution is applied for level 3 and 4 to obtain information about the surroundings. If the configuration is changed from parallel to serial, the network depth can be increased by using the same number of layers. In this case, however, the quality of the information processed at each stage is reduced, resulting in lower recognition performance as employed in Table VI. When the configuration of the level 3 module is changed to series, mIoU is lowered 3.4%. When changing the level 4 module, the recognition performance is dropped by 7.2%.

TABLE VII
EMBEDDED NETWORK COMPLEXITY, ACCURACY AND PROCESSING TIME COMPARISONS ON CITYSCAPES VALIDATION SET

Name	#Parameter (M)	#Multiplication (G)	FLOPS (G)	mIoU (%)	Processing Time (us)
CGNet	0.5	3.8	7.6	63.5	7.6
ContextNet	0.8	6.6	13.1	65.9	2.3
DABNet	0.7	38.8	77.2	69.1	10.0
EDANet	0.7	33.2	66.8	65.1	6.2
ERFNet	2.0	103.9	208.7	71.5	9.8
FastSCNN	1.1	6.5	13.0	68.6	2.1
ESPNet2	0.7	13.2	26.2	66.4	16.1
MMANet (ours)	4.6	6.3	12.5	73.1	6.7

The PReLU of ESPNet2, CGNet and DABNet is replaced by a ReLU to measure the performance time.
The FGlo of CGNet is not used to remove exponential operations while measuring times.



(a) Images (b) Ground truths (c) Inference results

Fig. 6 MMANet cityscapes validation set inference results (The last row shows a failure mode.)

D.Result Comparison

DSC is an essential element in implementing the embedded network recently. If the processing time is proportional to the number of computations, DSC is a very efficient layer since it can reduce computations by the number of channels. However, in an embedded environment, the actual performance time depends on the type of hardware accelerator, and the memory access overhead cannot be ignored. MMANet is a network designed to have optimum performance time in the MMA environment of TDA4V-MID, providing the highest perceived performance of 73.1 mIOU on the cityscapes validation set as shown in Table VII. Thus, the MMANet can provide information on the location of objects, such as vehicles and pedestrians, as shown in Fig. 6, improving the reliability of the autonomous driving system. Moreover, it satisfies constraints with enough margin as given in Table VIII.

TABLE VIII
MMANET IMPLEMENTATION RESULTS

Attribute	Value	Margin (%)
Processing Time	6.741ms	15.7
L3 Cache Size	4.607MB	41.9
#Layers	209	39.3

V.CONCLUSION

We proposed an embedded semantic segmentation network, MMANet, optimized for the hardware accelerator of TDA4V-MID to improve the recognition performance of autonomous driving systems. Since the processing time of embedded networks are directly proportional to accuracies, the proposed method performs operations only using L3 cache and chooses efficient convolution layers according to the size of the activation map. In addition, it extracts high level context by using the extended ASPP. To increase the number of network depths, the maximum number of channels are limited to 64. In addition, it replenishes the amount of information by using multiple paths. This method can process a VGA image during 6.67ms, so it can provides the information around vehicle by using six cameras in real time. Moreover, it achieves 73.1 mIoU on the Cityscapes validation set.

REFERENCES

[1] S. Karen and Z. Andrew, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in International Conference on Learning Representations, 2015.

[2] C. Liang-Chieh et al, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.

[3] W. Yu, Z. Quan and W. Xiaofu, "ESNet: An Efficient Symmetric Network for Real-time Semantic Segmentation," 2019.

[4] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.

[5] R. Olaf, F. Philipp and B. Thomas, "U-Net: Convolutional Networks for Biomedical Image Segmentation, " MICCAI2015 pp. 234-241.

[6] H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia, "Pyramid Scene Parsing

Network," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 6230-6239.

[7] L. Chen et al, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," ECCV 2018, pp. 801- 818.

[8] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481-2495, 1 Dec. 2017.

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 4510-4520.

[10] A. Howard et al, "Searching for mobilenetv3," arXiv:1905.02244, 2019.

[11] T. Wu, S. Tang, R. Zhang, and Y. Zhang, "Cgnet: A light-weight context guided network for semantic segmentation," arXiv:1811.08201, 2018.

[12] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets v2: More Deformable, Better Results," arXiv:1811.11168, 2018.

[13] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," arXiv:1606.02147, 2016.

[14] S. Mehta, M. Rastegari, L. G. Shapiro, and H. Hajishirzi, "Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network," CoRR, abs/1811.11431, 2018.

[15] T. M. Arjona-Medina et al, “Speeding up semantic segmentation for autonomous driving,” NIPS Workshop, 2016

[16] M. Liu and H. Yin, "Feature Pyramid Encoding Network for Real-time Semantic Segmentation," arXiv:1909.08599, 2019

[17] R. P. Poudel, S. Liwicki and R. Cipolla, “Fast-SCNN: Fast Semantic Segmentation Network,” arXiv:1902.04502, 2019.