

CASSOD-Net: Cascaded and Separable Structures of Dilated Convolution for Embedded Vision Systems and Applications

Tse-Wei Chen*, Deyu Wang†, Wei Tao†, Dongchao Wen†, Lingxiao Yin†,
 Tadayuki Ito*, Kinya Osa*, and Masami Kato*

twchen@ieee.org

*Device Technology Development Headquarters, Canon Inc., Japan

†Canon Innovative Solution (Beijing) Co., Ltd., China

Abstract

The field of view (FOV) of convolutional neural networks is highly related to the accuracy of inference. Dilated convolutions are known as an effective solution to the problems which require large FOVs. However, for general-purpose hardware or dedicated hardware, it usually takes extra time to handle dilated convolutions compared with standard convolutions. In this paper, we propose a network module, Cascaded and Separable Structure of Dilated (CASSOD) Convolution, and a special hardware system to handle the CASSOD networks efficiently. A CASSOD-Net includes multiple cascaded 2×2 dilated filters, which can be used to replace the traditional 3×3 dilated filters without decreasing the accuracy of inference. Two example applications, face detection and image segmentation, are tested with dilated convolutions and the proposed CASSOD modules. The new network for face detection achieves higher accuracy than the previous work with only 47% of filter weights in the dilated convolution layers of the context module. Moreover, the proposed hardware system can accelerate the computations of dilated convolutions, and it is 2.78 times faster than traditional hardware systems when the filter size is 3×3 .

1. Introduction

Dilated convolutions in Convolutional Neural Networks (CNNs) can be applied to different kinds of applications, including audio processing [6], crowd counting [9], semantic image segmentation [3, 14, 18, 20], image classification [7], image super-resolution [11], road extraction [22], image denoising [17], face detection [8], object detection [10], and so on. For many vision applications, the receptive field, or the field of view (FOV) [12] of convolutional neural networks is highly related to the accuracy of inference. Large FOV can be obtained by increasing the filter size or by increasing the number of convolution layers. However, the

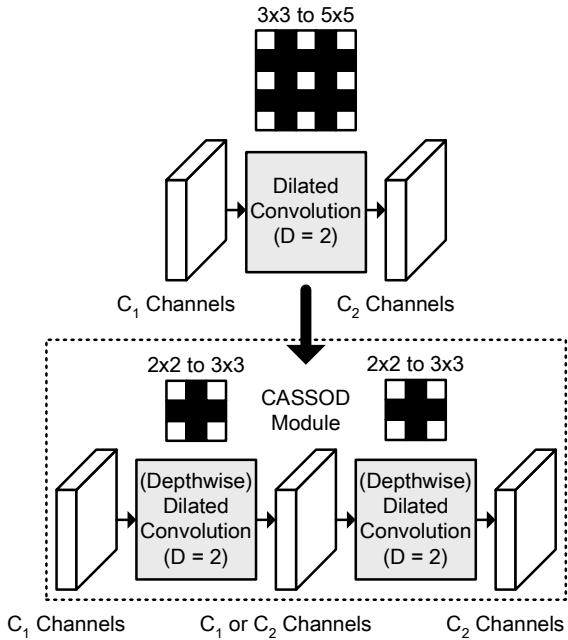


Figure 1: Concept of the proposed CASSOD module.

computational time might also increase because the computational cost is proportional to the filter size and the number of layers. Dilated convolutions are known as an effective method to enlarge the FOV of a network without increasing the computational costs. By adjusting the dilation rate D , a 3×3 filter can be enlarged to a $(2D + 1) \times (2D + 1)$ filter with 3×3 Multiply-Accumulate (MAC) operations.

Deng et al. propose a context module, which is inspired by the SSH face detector [16], to increase the FOV [2]. Wu et al. propose a Joint Pyramid Up-sampling (JPU) module and formulate the task of extracting high-resolution feature maps into a joint up-sampling problem [20]. In the JPU structure, the generated feature maps are up-sampled and

Table 1: Comparison of Number of Filter Weights

Network Structure	No. of Filter Weights
Dilated Convolution	$3^2 \times C_1 \times C_2$
CASSOD-A (1st layer: DW Conv.*)	$2^2 \times C_1 \times (1 + C_2)$
CASSOD-C (No DW Conv.*)	$2^2 \times (C_1 + C_2) \times C_1$ or $2^2 \times (C_1 + C_2) \times C_2$
Dilated / Depthwise Convolution	$3^2 \times C_1$
CASSOD-D (1st and 2nd layer: DW Conv.*)	$2^2 \times (C_1 \times 2)$

*DW Conv. denotes the depthwise convolutions.

concatenated, and 4 separable convolutions with different dilation rates ($D = 1, 2, 4, 8$) are included in the same convolution layer. Hamaguchi et al. propose a segmentation model, which includes a front-end module, a local feature extraction (LFE) module, and a head module [3]. The dilation rate in the front-end module is gradually increased to extract the features from a large range, and the dilation rate in the subsequent LFE module is gradually decreased to aggregate local features generated by the front-end module. The above mentioned network models are designed for applications related to face detection and image segmentation. There are still many kinds of network architectures containing dilated convolution layers, which are used to improve the accuracy of other applications.

In order to implement dilated convolutions on mobile devices and embedded system platforms, it is necessary to reduce the memory size and the computational costs without decreasing the accuracy. For some hardware devices, including GPUs, convolution operations are only optimized for standard convolutions. Even though the 3×3 dilated filters require only 3×3 MAC operations, the overhead to skip the pixels which do not engage in the process of convolutions is not always zero. For some dedicated hardware systems based on systolic arrays, when the dilation rate is high, the memory footprint might increase since the addresses of the pixels to be processed are not consecutive.

In this paper, we propose a new network module, Cascaded and Separable Structure of Dilated (CASSOD) Convolutions, to approximate the 3×3 dilated convolutions with low memory cost for filter weights and low computational costs for convolutions. The concept of the proposed method is shown in Fig. 1.

2. Proposed Network Module

In this section, the 2×2 dilated convolutions and the proposed CASSOD module are introduced in the following

subsections. The cascaded 2×2 dilated convolution, which can be a separable version of the traditional 3×3 dilated convolution, is an important component of the CASSOD module. In this paper, the 2×2 (or 3×3) dilated convolutions refer to the convolutions where the filter size is 2×2 (or 3×3) before the filters are dilated ($D = 1$).

2.1. Dilated Convolutions with Fewer Filter Weights

In previous works [3, 6, 7, 9, 11, 14, 17, 20, 22], the dilated filters are generated based on 3×3 filters. A total of 9 filter weights are used to compute the convolution results for one input feature map. The FOV is expanded when the dilation rate D is larger than 1. The larger the dilation rate D , the larger the FOV.

The dilated filters which are generated based on 2×2 filters, are proposed in this work. The output result of the proposed dilated filters is calculated based on the following equation.

$$O_{(i,j)} = \sum_{c=0}^{C-1} \sum_{x=0}^1 \sum_{y=0}^1 I_{(c,i-\frac{D}{2}+x \cdot D, j-\frac{D}{2}+y \cdot D)} \times W_{(c,x,y)}, \quad (1)$$

where $O_{(i,j)}$ is the output feature map, $I_{(c,i,j)}$ is the c -th input feature map, and $W_{(c,x,y)}$ is the filter weight for the c -th input feature map. The parameters (i, j) represent the index of the pixels in the feature maps, and the ranges of both i and j depend on the size of feature maps. The parameter C denotes the number of channels. There are only 4 filter weights for 1 input feature map and the ranges of both x and y are $[0, 1]$.

Two examples of the 2×2 filters and dilated convolutions are shown in Fig. 2. Fig. 2(a) shows an example where the dilation rate D is 2. Two cascaded 2×2 dilated convolutions where the dilation rate D is 2 can be an approximation to a 3×3 dilated convolution with the same dilation rate since the positions of zero weights are exactly the same. Similarly, as shown in Fig. 2(b), two cascaded 2×2 dilated convolutions with the dilation rate $D = 4$ can be an approximation to a 3×3 dilated convolution with the same dilation rate. The dilation rate D is always set to a multiple of 2.

2.2. Cascaded and Separable Structure of Dilated (CASSOD) Convolution

The concept of the proposed CASSOD module is shown in Fig. 1. The upper part of the figure shows an example of dilated convolutions [19] where the dilation rate D is 2. There are C_1 channels in the input feature maps, and C_2 channels in the output feature maps. The number of filter weights of the dilated convolutions is $3 \times 3 \times C_1 \times C_2$.

The lower part of the figure shows an example of the proposed CASSOD module where the dilation rate D is

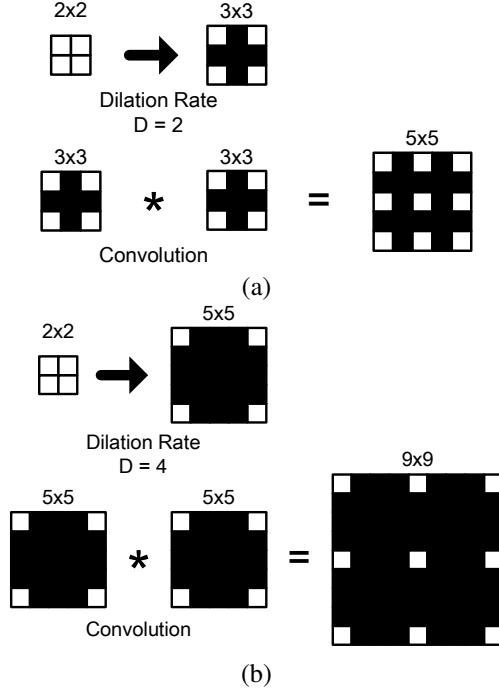


Figure 2: Example of the 2×2 dilated convolutions where (a) the dilation rate $D = 2$ and (b) the dilation rate $D = 4$.

2. There are two convolution layers in this module and 3 sets of feature maps. Either of the first convolution layer or the second convolution layer can be a depthwise convolution [5] layer. The variations (CASSOD-A,C,D) are shown in Table 1.

In the CASSOD-A module, there are C_1 , C_1 , and C_2 channels in the first, the second, and the third set of feature maps, respectively. The numbers of channels in the first and the second set of feature maps are the same because the first convolution layer includes depthwise separable operations. The first convolution layer includes a series of 2×2 depthwise and dilated filters, which are different from the traditional 3×3 dilated filters. The second convolution layers includes a series of 2×2 dilated filters. The number of filter weights of the dilated convolutions is $2^2 \times C_1 \times (1 + C_2)$. Table 1 also shows the comparison of the number of filter weights. It can be observed that, when C_2 is large, the number of filter weights in the CASSOD-A module is close to 44.4% (4/9) of the number of filter weights in the traditional dilated convolution.

In the CASSOD-C module, there are C_1 , C_1 (or C_2), and C_2 channels in the first, the second, and the third set of feature maps, respectively. Both convolution layers include a series of 2×2 dilated filters. The number of filter weights of the dilated convolutions is $2^2 \times (C_1 + C_2) \times C_1$ or $2^2 \times (C_1 + C_2) \times C_2$. It can be observed that, when C_1 is equal to

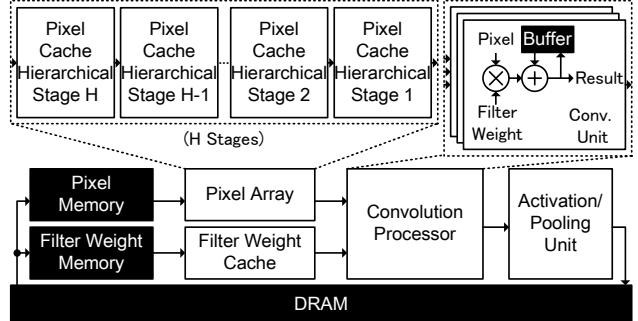


Figure 3: An example of the architecture of the proposed hardware system.

C_2 , the number of filter weights of the CASSOD-C module is close to 88.9% (8/9) of the number of filter weights in the traditional dilated convolution.

The CASSOD-A,C modules have lower computational costs than the traditional 3×3 dilated convolutions. The number of filter weights is also proportional to the computational cost, which is equal to the number of MAC operations. Compared with the traditional 3×3 dilated convolutions, 1 additional convolution layer is required to implement the CASSOD modules. The batch normalization operations and the activation functions (e.g. ReLU) can be included in the additional convolution layer if necessary. The CASSOD-A,C modules are alternatives to the dilated convolutions, and the CASSOD-D module is an alternative to the depthwise and dilated convolutions. The number of filter weights of the CASSOD-D module is close to 88.9% (8/9) of the depthwise and dilated convolutions.

3. Proposed Hardware Architecture

The analysis of computational time of dilated convolutions and the proposed hardware architecture are shown in the following subsections.

3.1. Computational Time of Dilated Convolutions

Table 2 shows an example of the computational time of dilated convolutions when the size of input image is 320×320 pixels. A network with 3 layers is used to measure the computational time of CPU, and a network with 10 layers is used to measure the computational time of GPU. There are 64 input channels and 64 output channels in each layer of the networks, and the dilation rates D of the 3×3 convolution layers are set to the same value.

The results show that the processing time does not change much when the dilation rate D is larger than 1 for either CPU or GPU. The computational time of standard convolutions, where the dilation rate is set to 1, is the shortest, and the reason can be that the framework (e.g. CUDA li-

Table 2: Processing Time of Dilated Convolutions

Dilation Rate	Processing Time (ms)			
	Convolutions		Depthwise Conv.	
	CPU*	GPU**	CPU*	GPU**
$D = 1$	39.53	6.53	21.57	11.81
$D = 2$	862.56	19.12	861.99	11.83
$D = 3$	868.83	19.17	865.09	11.79
$D = 4$	854.23	19.21	862.95	11.78
$D = 5$	852.42	19.23	871.42	11.78

*The CPU is Intel Xeon E5-2640 v4 (2.40 GHz) and the memory size is 256 GB.

**The GPU is TITAN Xp (12 GB), and the version of cuDNN is 7.6.5.

brary) includes some optimized operations to accelerate the computations of standard convolutions. The speed of dilated convolutions may depend on the performance of system platforms, but the dilated convolutions with a dilation rate larger than 1 cannot necessarily achieve the same level of speed as standard convolutions with a dilation rate of 1. The goal of this work is to design a hardware system which can handle both dilated convolutions and standard convolutions efficiently.

For hardware implementation, the pixel values in the feature maps are usually stored in a shift register array. One solution to handle dilated convolutions using a shift register array is to add zeros to the filter weights and compute the results of standard convolutions. To handle a dilated 3×3 filter where the dilation rate D is 2, it is necessary to compute the products of 25 filter weights and 25 input pixels. A total of 16 filter weights equal to zero can be skipped. The computational time is proportional to the size of dilated filters, and the efficiency of filter processing is relatively low compared with CPU and GPU.

To solve this problem, a new hardware architecture is proposed to speed up dilated convolutions. An example of the architecture of the proposed hardware system is shown in Fig. 3, which includes 6 modules and DRAM. The filter weights stored in the “Filter Weight Memory” are sent to the “Filter Weight Cache,” and the feature maps of the current convolution layer stored in the “Pixel Memory” are sent to the “Pixel Array.” The filter weights and the pixels of feature maps are sent to the “Convolution Processor” to compute the convolution results, and the convolution results are sent to the “Activation and Pooling Unit” to generate the feature maps for the next convolution layer. The “Pixel Array” includes multiple hierarchical stages of pixel caches, which can generate the input pixels for dilated convolutions with a different dilation rate D .

3.2. Pixel Array for Dilated Convolutions

The “Pixel Array,” including multiple hierarchical stages, can handle dilated convolutions with different dilation rates efficiently. The proposed hardware architecture can be implemented in a Field-Programmable Gate Array (FPGA) or an Application-Specific Integrated Circuit (ASIC). Fig. 4 shows an example of the architecture and the interconnections of the proposed shift register array, where there are 3 hierarchical stages. In this example, there are 6×6 pixel buffers (or selectors) in each stage, and each pixel buffer (or selector) is connected to 4 other pixel buffers (or selectors) in the up, right, down, and left directions. It means that the pixel stored in the pixel buffer can be transferred to one of the 4 connected pixel buffers (or selectors). In the first hierarchical stage, the input of a pixel buffer is connected to the input of its neighboring pixel buffer and the pixel selector with the same position in the second stage. The difference of the shifted index and the original index of the pixel, X_0 , is 0 or 1 (2^0). Similarly, in the second hierarchical stage, the difference of the shifted index and the original index of the pixel, X_1 , is 0 or 2 (2^1), and in the third hierarchical stage, the difference of the shifted index and the original index of the pixel, X_2 , is 0 or 4 (2^2).

The supported dilation rate, which is equal to the total difference of the shifted index and the original index of the pixel in all hierarchical stages ($X_0, X_1, X_2, \dots, X_{H-1}$) is shown in the following equation.

$$D = \sum_{h=0}^{H-1} X_h \cdot 2^h, \quad (2)$$

where H is the total number of the hierarchical stages. In this example, H is equal to 3, but H can be set to any number to support $2^H - 1$ dilation rates ($D = [1, 2^H - 1]$) according to the network architectures. When the gate count of each stage is the same, the total hardware cost is proportional to the number of hierarchical stage, H .

4. Experimental Results

The experimental results contain 3 parts. The first part is the comparison of accuracy of face detection. The second part is the comparison of accuracy of image segmentation. The third part is the analysis of the performance of the proposed hardware architecture.

4.1. Accuracy of Face Detection

The proposed network architecture, CASSOD-Net, is evaluated based on the RetinaFace [2], which is designed for face detection tasks. The context module, inspired by the SSH face detector [16], is used to increase the FOV and enhance the rigid context modeling power. To compare the accuracy, the convolution layers in the context modules is

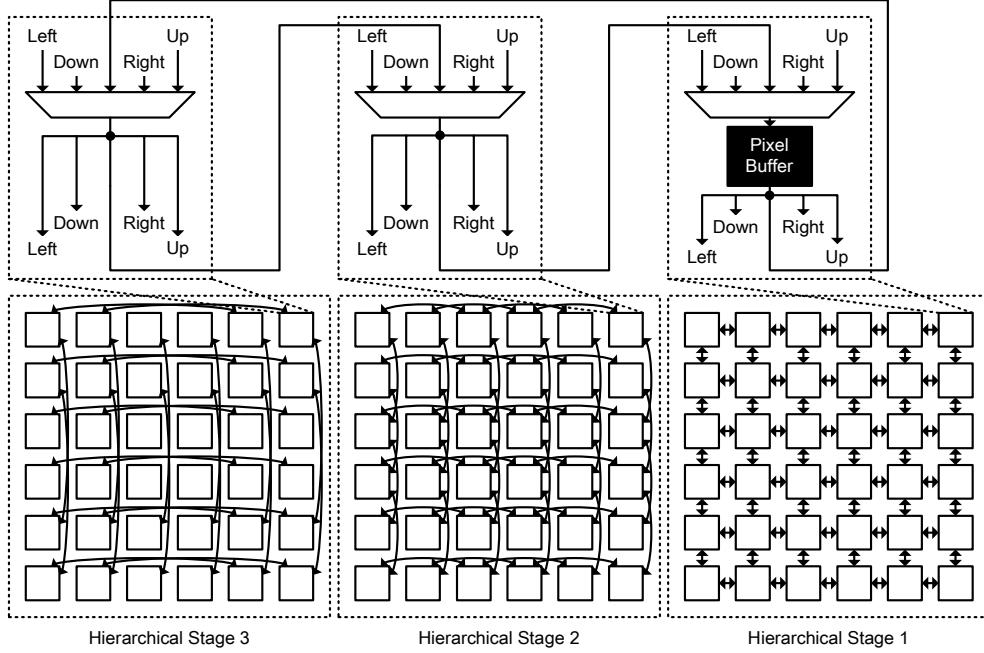


Figure 4: An example of the hardware architecture and the interconnections of the “Pixel Array” with 6×6 pixels.

replaced by the Feature Enhance Module (FEM), which includes dilated convolution structures, in the Dual Shot Face Detector (DSFD) [8]. Then, the dilated convolutions in the DSFD [20] are replaced by the proposed CASSOD modules. The network architecture of the modified context module is shown in Fig. 5. To keep the computational costs at the same level, the number of input channels of the FEM is changed from 256 to 64, and the number of input channels in the remaining layers is reduced by the same ratio. Fig. 5(a) shows the original context module in the RetinaFace, and Fig. 5(b) shows the modified context module with FEM, which includes dilated convolution layers with a dilation rate (D) of 2.

MobileNet-0.25 [5] is used as the backbone network. The accuracy of face detection is shown in Table 3. After replacing the dilated convolutions with the CASSOD-C and CASSOD-A modules, the accuracy does not decrease. Besides, the proposed CASSOD module can achieve higher accuracy than the previous architecture with batch normalization and activation. The CASSOD-C module has better performance than the CASSOD-A module on the 3 categories (easy, medium, and hard), and the reason can be that the CASSOD-C module has more parameters than the CASSOD-A module. It is shown that the CASSOD module is a good alternative to the original dilated convolutions. The accuracy of the CASSOD-A module shows that the new network achieves higher accuracy than the FEM-based network with only 47% of filter weights in the dilated convolution layers of the context module.

4.2. Accuracy of Image Segmentation

The proposed network architecture, CASSOD-Net, is evaluated based on the FastFCN [20], which is designed for semantic image segmentation tasks. The JPU module, which is included in the FastFCN, combines the up-sampled feature maps by using 4 groups of dilated convolutions with different dilation rates. The JPU modules also contain depthwise convolution layers. It can extract multiple-scale context information and increase the accuracy of image segmentation.

To compare the accuracy, the dilated convolutions in the JPU [20] are replaced by the proposed CASSOD-D modules. The network architecture of the modified JPU is shown in Fig. 6. ResNet-50 [4] is used as a backbone network and trained with Pascal Context [15] and ADE20K datasets [21]. The accuracy of image segmentation is shown in Table 4. The results [20] and our re-implementation results are very close. After replacing the dilated convolutions with the CASSOD modules, the accuracy does not decrease. Besides, the proposed CASSOD module can achieve higher accuracy than the previous architecture with batch normalization and activation. It is shown that the CASSOD module is a good alternative to the original dilated convolutions for image segmentation. Also, the proposed CASSOD modules can be applied to networks with depthwise and dilated convolutions. The computational time of GPU increases after

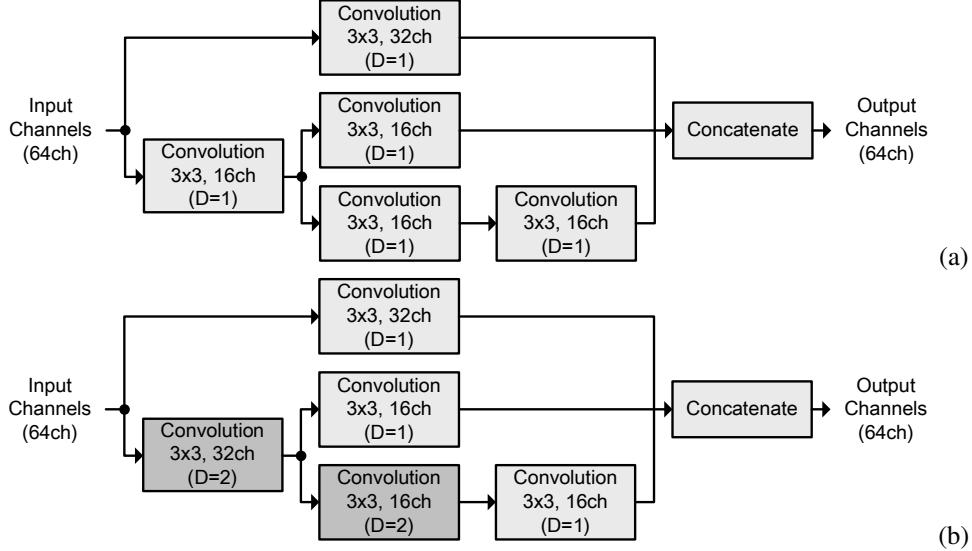


Figure 5: (a) The context module of RetinaFace [2] with the SSH structure [16] and (b) the modified context module with the FEM structure [8], in which the dilated convolutions can be replaced by the proposed CASSOD modules.

Table 3: Comparison of Accuracy of Face Detection [2] between Dilated Convolutions and CASSOD Modules

Context Module of RetinaFace [2]	Easy (%)	Medium (%)	Hard (%)	Parameter Size of Dilated Conv. Layers
SSH [†]	88.72	86.97	79.19	(11,520)
FEM ^{††}	88.87	86.74	80.26	23,040
FEM ^{††} -CASSOD-C	89.05	87.46	81.09	15,360
FEM ^{††} -CASSOD-C with BN*	89.21	87.55	81.28	15,552
FEM ^{††} -CASSOD-C with BN* and ReLU**	89.12	87.62	81.16	15,552
FEM ^{††} -CASSOD-A with BN* and ReLU**	88.88	87.40	80.74	10,912

[†]SSH represents single stage headless face detector [16]. The parameter size of convolution layers to be replaced by dilated convolution layers and CASSOD modules is 11,520.

^{††}FEM represents feature enhance module [8].

*BN represents batch normalization.

**ReLU represents rectified linear unit.

applying the CASSOD modules because an extra layer is added. This problem can be solved by using the proposed hardware architecture.

4.3. Analysis of Hardware Systems

To show the advantages of the proposed hardware system, the hardware architecture and the CASSOD modules are compared with previous works. The previous works refer to the hardware systems [1, 13] which do not accelerate the algorithms with the 3×3 dilated convolutions shown in Sec. 2. Since the hardware systems of the related works are not available, we re-implement a hardware system on our platforms without using the “Pixel Array” shown in Sec. 3.2

for comparison. The results are shown in Fig. 7.

Fig. 7(a) shows the relation between the computational time and the dilation rate (D) with 3×3 filters. The number of cycles is equivalent to the computational time. In previous works, since there are no special hardware architectures to handle dilated convolutions, it is necessary to add zero-values to the filter weights after a filter is dilated. The computational time increases as the dilation rate D increases and is roughly proportional to a square of the dilation rate (D). In the proposed hardware system, the input pixels of dilated convolutions can be adjusted according to the dilation rate (D) and dumped consecutively, and the computational time does not vary with the dilation rate D .

Table 4: Comparison of Accuracy of Image Segmentation [20] between Dilated Convolutions and CASSOD Modules

Datasets	Networks	pixAcc (%)	mIoU (%)	Speed (fps [†])
Pascal Context [15]	EncNet + JPU (Table 2, 3 in [20])	–	51.20	37.56
	EncNet + JPU (Re-implementation)	77.88	49.44	35.30
	EncNet + JPU-CASSOD-D (Proposed Work)	79.52	52.51	34.73
	EncNet + JPU-CASSOD-D with BN and ReLU (Proposed Work)	79.67	52.72	34.57
	EncNet + JPU-CASSOD-D with BN (Proposed Work)	79.75	52.76	34.12
ADE20K [21]	EncNet + JPU (Table 2, 4 in [20])	80.39	42.75	37.56
	EncNet + JPU (Re-implementation)	80.04	42.09	35.30
	EncNet + JPU-CASSOD-D (Proposed Work)	80.35	42.72	34.73
	EncNet + JPU-CASSOD-D with BN and ReLU (Proposed Work)	80.48	42.86	34.57
	EncNet + JPU-CASSOD-D with BN (Proposed Work)	80.42	42.78	34.12

[†]Fps represents frame per second, which is the unit of processing speed of the GPU, TITAN Xp (12 GB).

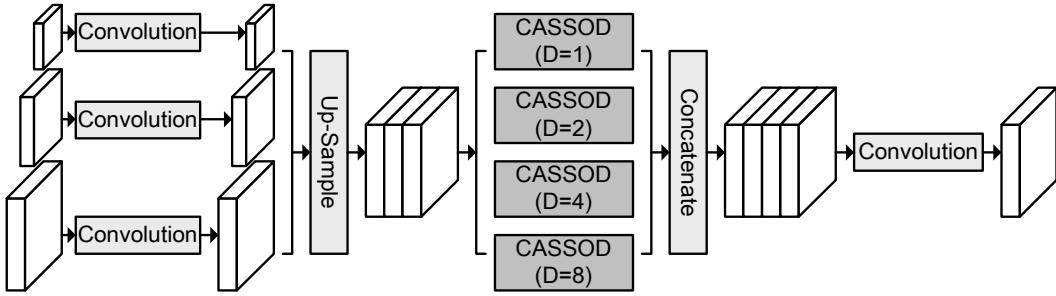


Figure 6: The modified JPU [20] in which the dilated convolutions are replaced by the proposed CASSOD modules.

It can be observed that the proposed hardware system can handle both dilated convolutions and standard convolutions ($D = 1$) efficiently. When the dilation rate (D) of a 3×3 filter is 2, the proposed hardware system is 2.78 times faster than the previous work.

Fig. 7(b) shows the relation between the computational time and the dilation rate (D). By replacing the traditional dilated convolutions with the CASSOD module, the computational cost and the parameter size can be reduced even though 1 additional convolution layer is required. In the proposed hardware system, the time to set the parameters for 1 additional layer is relatively small. The results show that the computational time of the proposed hardware system can be further reduced by using the CASSOD module, which achieves similar accuracy as the traditional dilated convolutions.

Fig. 7(c) shows the relation between the gate count of the "Pixel Array" and the maximum dilation rate (D). It can be observed that the overhead of hardware cost increases as

the dilation rate D increases, but the difference of hardware cost between the previous work and the proposed work is not proportional to the maximum dilation rate, D . The proposed hardware system scales well because it can support $2^3 - 1$ dilation rates ($D = [1, 7]$) with less than 3 times of hardware costs.

The supported dilation rates also depend on the filter size and the interface between modules. Table 5 shows the specifications of the proposed hardware architecture. Different from previous works, the proposed hardware system can handle 2×2 and 3×3 dilated convolutions efficiently. The maximum supported dilation rate for 2×2 filters is 6, and the maximum supported dilation rate for 3×3 filters is 3. By replacing the traditional dilated convolutions with the CASSOD module, an approximated 3×3 dilated filters with a dilation rate of 6 can be implemented with cascaded 2×2 filters with a dilation rate of 6. The computational costs and memory footprints can also be reduced. Since the "Pixel Array" with 3 hierarchical stages only occupies

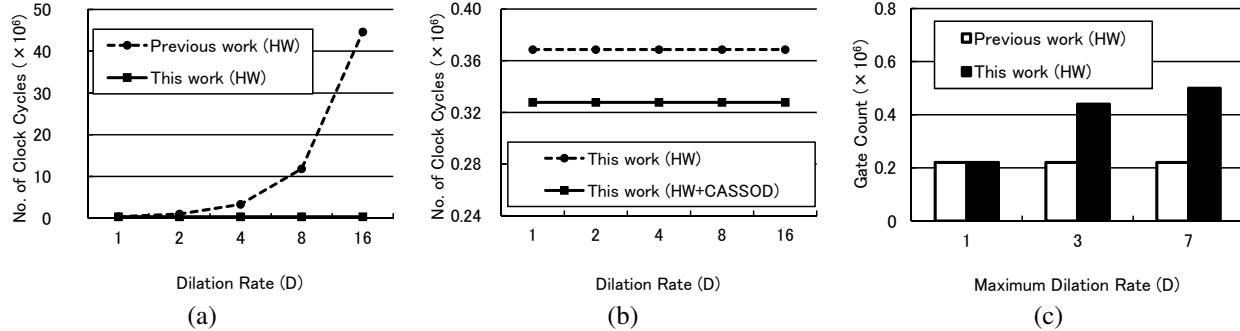


Figure 7: Analysis of (a) the computational time of 3×3 filters and the dilation rate (D), (b) the computational time of the proposed CASSOD module and the dilation rate (D), and (c) the gate count of the “Pixel Array” and the maximum dilation rate (D). The computational time of 1 convolution layer is measured. The size of input feature maps is 128×128 pixels, and the numbers of input channels and output channels are both set to 64. “HW” in the figure caption represents hardware.

Table 5: Specifications of Proposed Hardware Architecture

Gate Count (NAND-Gates)	Total: 2.4 M (Pixel Array: 0.5 M)
Process	28-nm CMOS technology
Clock Frequency	400 MHz
Supported Filter Size	Maximum 7×7
Supported Dilation Rate	2×2 filter: $D = 2, 4, 6$ 3×3 filter: $D = 1, 2, 3$
No. of Hierarchical Stages	$H = 3$
Memory Size	128 KB
Performance	409.6 GOPS*

*GOPs represents giga operations per second. There are 2 operations in 1 MAC operation.

only 21% of the total area, the overhead to support the dilated convolutions is relatively small. A comparison of the processing speed of the proposed system is shown in Table 6. The resolution of the input image is 512×512 pixels, and the network is RetinaFace [2] with FEM [8], which is shown in Table 3. By using the proposed hardware architecture with the shift register array, 23% of processing time can be reduced. By replacing the dilated 3×3 filters with the proposed CASSOD modules, 9% of processing time can be further reduced. The result clearly shows the advantages in terms of computational speed.

5. Conclusions and Future Work

In this paper, we propose an efficient module, which is called Cascaded and Separable Structure of Dilated (CASSOD) Convolutions, and a special hardware system to handle the CASSOD networks efficiently.

To analyze the accuracy of algorithms, two example applications, face detection and image segmentation, are

Table 6: Processing Speed of Proposed Hardware

	RetinaFace [2] +FEM [8]	RetinaFace [2] +FEM [8] +CASSOD-A
without Pixel Array	171 fps	224 fps
with Pixel Array	222 fps	244 fps

tested with dilated convolutions and the proposed alternatives. For face detection, the RetinaFace [2] network architecture can achieve the same level of accuracy after replacing the dilated convolutions in the context module with the proposed CASSOD modules. For image segmentation, the FastFCN [20] can achieve the same level of accuracy after replacing the dilated convolutions in the JPU with the proposed CASSOD modules, which also contain depthwise convolutions. It is shown that the CASSOD module is a good alternative to the traditional dilated convolutions for both applications.

The performance of hardware is analyzed in terms of computational time and hardware costs. The input pixels of dilated convolutions can be adjusted according to the dilation rate (D) and dumped consecutively, the computational time does not vary with the dilation rate D . By using the proposed hardware architecture with the shift register array, 23% of processing time can be reduced for face detection applications.

The experiments clearly show that both the proposed hardware system and the proposed CASSOD modules have advantages over previous works. For future works, we plan to test the proposed hardware system and the CASSOD modules with other applications.

References

- [1] Yu-Hsin Chen, Joel Emer, and Vivienne Sze. Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. In *Proceedings of ACM/IEEE International Symposium on Computer Architecture (ISCA)*, June 2016.
- [2] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. RetinaFace: Single-stage dense face localisation in the wild, 2019. CoRR, abs/1905.00641.
- [3] Ryuhei Hamaguchi, Aito Fujita, Keisuke Nemoto, Tomoyuki Imaizumi, and Shuhei Hikosaka. Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery, 2017. CoRR, abs/1709.00179.
- [4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. CoRR, abs/1512.03385.
- [5] 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, 2017. CoRR, abs/1704.04861.
- [6] Shehzeen Hussain, Mojtaba Javaheripi, Paarth Neekhara, Ryan Kastner, and Farinaz Koushanfar. FastWave: Accelerating autoregressive convolutional neural networks on FPGA. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019.
- [7] Xinyu Lei, Hongguang Pan, and Xiangdong Huang. A dilated CNN model for image classification. *IEEE Access*, 7:124087–124095, July 2019.
- [8] Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Jilin Li, and Feiyue Huang. DSFD: Dual shot face detector, 2018. CoRR, abs/1810.10220.
- [9] Yuhong Li, Xiaofan Zhang, and Deming Chen. CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes, 2018. CoRR, abs/1802.10062.
- [10] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. DetNet: A backbone network for object detection, 2018. CoRR, abs/1804.06215.
- [11] Guimin Lin, Qingxiang Wu, Lida Qiu, and Xixian Huang. Image super-resolution using a dilated convolutional neural network. *Neurocomputing*, 275:1219–1230, 2018.
- [12] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks, 2017. CoRR, abs/1701.04128.
- [13] Yufei Ma, Naveen Suda, Yu Cao, Sarma Vrudhula, and Jaesun Seo. ALAMO: FPGA acceleration of deep learning algorithms with a modularized RTL compiler. *Integration*, 62:14–23, 2018.
- [14] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.
- [15] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [16] Mahyar Najibi, Pouya Samangouei, Rama Chellappa, and Larry Davis. SSH: Single stage headless face detector, 2017. CoRR, abs/1708.03979.
- [17] Chunwei Tian, Yong Xu, Lunke Fei, Junqian Wang, Jie Wen, and Nan Luo. Enhanced CNN for image denoising, 2018. CoRR, abs/1810.11834.
- [18] Zhengyang Wang and Shuiwang Ji. Smoothed dilated convolutions for improved dense prediction, 2018. CoRR, abs/1808.08931.
- [19] Yunchao Wei, Huixin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S. Huang. Revisiting dilated convolution: A simple approach for weakly- and semi-supervised semantic segmentation, 2018. CoRR, abs/1805.04574.
- [20] Huikai Wu, Junge Zhang, Kaiqi Huang, Kongming Liang, and Yizhou Yu. FastFCN: Rethinking dilated convolution in the backbone for semantic segmentation, 2019. CoRR, abs/1903.11816.
- [21] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [22] Lichen Zhou, Chuang Zhang, and Ming Wu. D-LinkNet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.