SPLConv: Channel-Separated Convolution Using Large Convolution Kernels

Yiming Yao [1, a, *], Luyang Jie [1, b], Yilong Guo [1, c] and Yongkang Liu [1, d]

[1] School of Microelectronics, Shanghai University, Shanghai, 201899, China

[a, *] yaoym@shu.edu.cn, [b] jly@she.edu.cn, [c] wdz_gyl@shu.edu.cn, [d] yk_liu@shu.edu.cn

Abstract. In modern society, chips play a crucial role in various aspects of everyday life. Nevertheless, the diverse chip manufacturing processes can introduce defects, making the detection of faulty chips imperative. Convolutional neural networks (CNNs) are extensively employed in computer vision due to their ability to deliver high-performance results. However, this comes at the expense of significant computational resources. Current research focuses on employing pruning or quantization techniques to minimize the size of model parameters and investigate the development of lightweight models. This paper proposes a large depth-wise convolutional module called split-based large convolutional (SPLConv), which is devised and optimized from the perspective of split-based convolutional operations (SPConv). SPLConv not only disregards redundant features in the convolutional layer but also enhances spatial information capture, thereby mitigating excessive computational overhead. Experiment demonstrate that SPLConv effectively reduces complexity and computational cost while delivering commendable performance.

Keywords: CNN; Defect segmentation; Model compression.

1. Introduction

The CNN model has demonstrated exceptional performance in classification and segmentation tasks; however, this comes at the expense of increased complexity. Existing models often comprise parameter quantities in the range of tens of millions to billions, requiring billions of FLOPs [1]. However, deploying such models becomes challenging when hardware resources are limited, as they demand substantial computing and storage resources. In recent years, numerous methodologies and techniques have been proposed to address this issue, including network pruning, weight quantization, low rank decomposition, lightweight models, and knowledge distillation.

Network pruning is a technique that involves removing unimportant neural connections, eliminating redundant parameters, and reducing complexity within a network [2,3]. Weight quantization refers to the process of reducing the number of numerical digits used to represent weights in a network model, which helps conserve computational resources [4,5]. Low rank decomposition utilizes matrix decomposition techniques to estimate information parameters [6]. Knowledge distillation involves leveraging large models to guide the development of smaller networks [7,8]. Lightweight models aim to minimize the inherent redundancy of model parameters, as well as reduce both the parameter and computational complexity of the entire network, thereby achieving cost-effective computational efficiency [9,10].

This paper proposes a module that designs a lightweight model aimed at minimizing the number of parameters, thereby reducing computation time and enabling faster inference. In comparison to SPConv [11], SPLConv reduces channel redundancy and captures more contextual information without incurring additional computational cost. By deploying this module to the U-Net model for semiconductor defect segmentation, as demonstrated in Figure 1, significant reduction of parameters and computational complexity, without any significant decrease in performance. The contributions of this study can be summarized as follows:

·We employ deep separable large convolutions to replace channel convolutions in SPConv, effectively reducing the computational redundancy within the channel. Additionally, large kernel convolutions have the ability to capture information from distant regions and can be implemented as separable convolutions with minimal computational cost. To further enhance the performance of the

U-Net model, we incorporate batch normalization and convolution in SPLConv, followed by the addition of activation functions.

·The number of SPLConv parameters has been reduced by 5 times, Flops have been reduced by 5 times, inference speed has been accelerated, and performance in semiconductor defect segmentation is not inferior to U-Net.

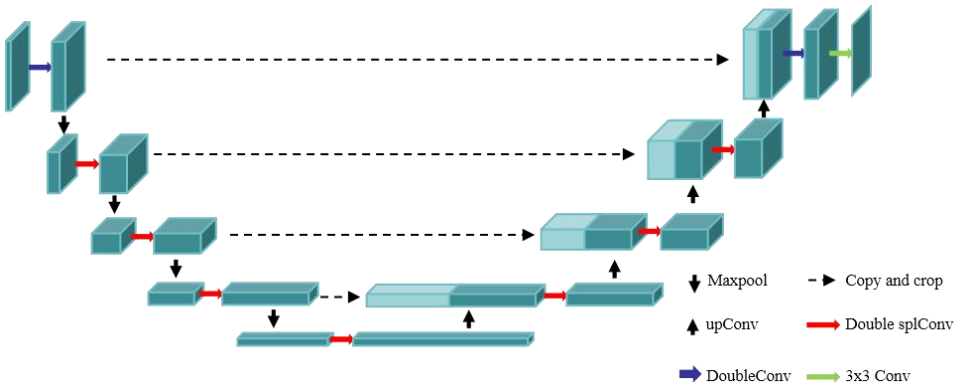

Fig. 1 Deploy SPLConv on UNet

2. Methodology

Deployed the designed module on the UNet network, in addition to the first layer of the encoder and the last layer of the decoder, the standard 3×3 convolutions replace with SPLConv as shown in Figure 2.

2.1 Channel Separation

For the input feature $X \in R^{C\times H\times W}$, The output is $Y \in R^{M\times H\times W}$, A feature map representing the input C channels and the output M channels with a height of H and a width of W. We divide the channel into two parts for calculation, one part of the channel αC uses depth-wise large convolutional kernels for calculation, while the remaining channels (1-α) use a less computational 1×1 convolution to supplement small details. For the separation ratio α, its value is the proportion of the number of channels in that part to the number of input channels, $0 \le \alpha \le 1$.

2.2 Channel Extract

For the first part of the channel, we use depth-wise large convolution kernels[12] to extract features. depth-wise large convolution kernels utilize the characteristics of deep separable convolutions and can be divided into three parts: depth-wise convolution(DConv), depth-wise dilation convolution(DDConv), and point convolution($Conv_{1\times1}$). Depth-wise convolution and deep dilation convolution apply a single filter to each input channel, while point convolution combines the outputs obtained through depth-wise convolution and depth-wise dilation convolution. Through this decomposition, the computational cost is relatively low and the parameters can capture remote relationships. The depth-wise large convolutional kernel can be written as:

$$D(X_\alpha) = Conv_{1\times1}\left(DDConv\left(DConv(X_\alpha)\right)\right) \#(1)$$

$$Output_\alpha = ReLU\left(bn\left(D(X_\alpha)\right)\right) \#(2)$$

Here, input $X_\alpha \in R^{\alpha C\times H\times W}$, bn, and ReLU represent the batchnorm and Relu activation function, respectively, $Output_\alpha \in R^{M\times H\times W}$.

The second half of the channel uses a very low computational cost of 1×1. Convolution is used for calculation to obtain small details of shallow features. As a supplement to the first half of channel convolution, it can be expressed as:

$$Output_{1-\alpha} = Conv_{1\times1}(X_{1-\alpha}) \#(3)$$

Here, input $X_{1-\alpha} \in R^{(1-\alpha)C\times H\times W}$, $Output_{1-\alpha} \in R^{M\times H\times W}$.

2.3 Channel Fusion

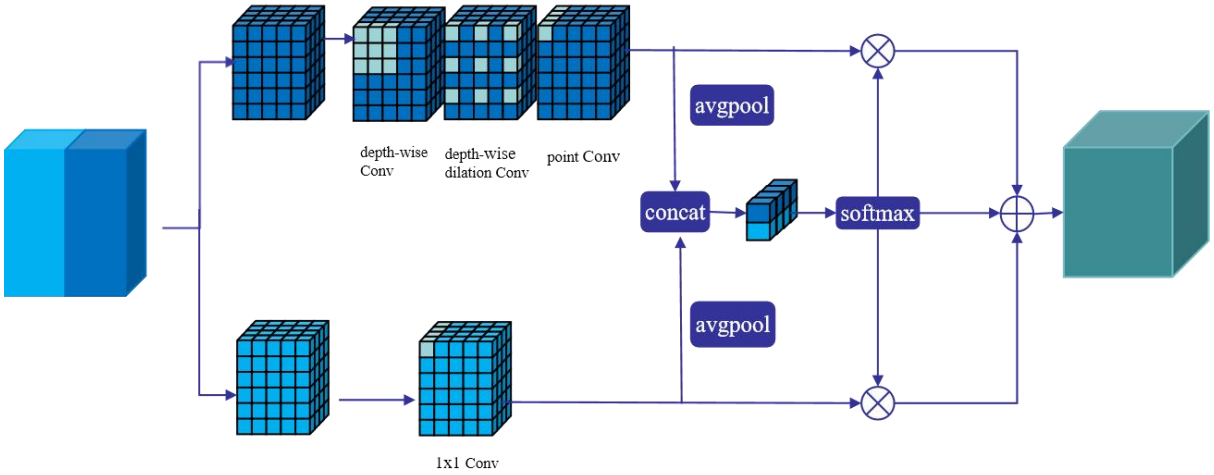

Fig. 2 Module of SPLConv

Obtained from the first half of the channel $Output_{\alpha}$ and Obtained from the second half of the channel $Output_{1-\alpha}$, extracting global spatial information $L_i \in R^{M\times1\times1}$ from two output channels through global average pooling, the calculation formula is as follows:

$$L_i = \frac{1}{H \times W} \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} Output_i(h,w), i = \alpha, 1-\alpha \#(4)$$

Next, overlay the global spatial information of the two parts on a new dimension, and through the softmax layer, obtain the importance vectors of the two parts γ_{α} and $\gamma_{1-\alpha}$, The sum of the two is 1. Multiply the two importance vectors separately $Output_{\alpha}$ and $Output_{1-\alpha}$ Merge the obtained results to obtain the final output feature map:

$$Y = \gamma_{\alpha} Output_{\alpha} + \gamma_{1-\alpha} Output_{1-\alpha} \#(5)$$

2.4 Complexity analysis

For the above input feature map X, the number of channels is C, and the output feature map Y, the number of channels is M, The parameter quantity of a standard $k\times k$ convolution is:

$$P_s = k \times k \times C \times M = k^2 CM \#(6)$$

The parameter quantities of the SPLConv module are as follows：

$$P_{splConv} = k \times k \times \alpha C + \left\lceil \frac{k}{d} \right\rceil \times \left\lceil \frac{k}{d} \right\rceil \times \alpha C + \alpha CM + (1-\alpha)CM = \left(\alpha k^2 + \alpha \left\lceil \frac{k}{d} \right\rceil^2 + M \right) C \#(7)$$

Among them, k represents the size of the convolutional kernel, and d represents the expansion rate, α represents the separation ratio, with C and M representing the input and output characteristic channels. In the experiment, set parameter k=3, d=2, α=0.5, $P_{SPLConv}/P_s \approx 1/9$, The number of parameters has been reduced by approximately 9 times, but the model performance is better than standard convolution.

3. Experiment

3.1 Dataset and Implementation

Experiments were conducted on a semiconductor defect dataset comprising 1590 training sets and 682 test sets. The images have dimensions of 2048 in width and height, and the Mask represents a black-and-white binary image artificially labeled with defects. The model was trained

on the training set and subsequently validated on the test set, and the segmentation effect was evaluated using metrics such as intersection and union ratio and dice coefficient.

The images were processed and resized to 512 × 512 × 3 before being fed into the network model. The network is randomly initialized without any pre-training at the beginning of the training process. The network was trained for 200 epochs using the SGD optimizer with a learning rate of 0.005 and a momentum of 0.9. The experiment was performed on an NVIDIA RTX A5000 GPU.

3.2 Evaluation Metrics

Evaluate the performance of the model based on the indicators IOU and Dice, The larger the value of the indicator, the better the performance of the model. The parameter quantity represents the number of parameters involved in the calculation in the model, Flops is the total number of floating-point operations, and MAdd is the total number of multiplication and addition operations, both of which are indicators used to calculate the complexity of the model. FPS is the number of frames transmitted per second, that is, the number of images that can be processed per second or the time required to process one image, to evaluate the detection speed.

3.3 Results

Table 1. Results on Semiconductor Defect Dataset

Model	Param	Flops	MAdd	FPS	Mean IOU	Dice
UNet	7.76M	27.31G	57.05G	77	0.6012	0.7139
Depth-Wise-UNet	3.60M	19.82G	42.06G	74	0.6019	0.7138
SPConv-UNet	3.27M	11.08G	24.5G	36	0.6266	0.7383
SPLConv-UNet(encorder)	3.33M	19.27G	40.97G	49	0.6345	0.7448
SPLConv-UNet(r=0.5)	1.53M	5.04G	12.4G	39	0.6285	0.7423
SPLConv-UNet(r=0.75)	1.54M	5.2G	12.7G	39	0.6513	0.7576

Table 1 presents the experimental results. When comparing SPLConv-UNet and UNet, it is observed that SPLConv-UNet exhibits significantly reduced model complexity while achieving superior performance compared to UNet. SPLConv-UNet only requires 1/5 of the parameters and calculations needed by UNet. It achieves a 2.73% increase in IOU and a 2.84% increase in Dice compared to UNet, while also FPS has also been reduced by about twice. In comparison to other compression modules that directly replace standard convolution, such as depthwise, SPLConv-UNet reduces the number of parameters and calculations by half and achieves a 2.7% increase in IOU. When the performance is similar to that of SPConv-UNet, SPLConv-UNet achieves approximately 50% reduction in the number of parameters and calculations. Additionally, by replacing only the standard convolution in the encoder layer of UNet with SPLConv, a significant reduction in the number of parameters and calculations is achieved, surpassing the performance obtained by replacing the standard convolution throughout the entire network. This result can be attributed to the decoder's standard convolution extracting extra features to complement those extracted by the encoder layer, thereby enhancing performance. To investigate the influence of various separation ratios α, we adjusted α to 0.75 and conducted subsequent experiments. With an increase in α, there was an improvement in IOU, suggesting that a higher α allows the model to capture more abundant feature information, consequently enhancing overall performance.

In Fig. 3, we visualize segmentation maps predicted by UNet and our method. Because the data involves confidentiality, here we only intercept the defective parts of the image for display. As can be seen from the Figure 3, the segmentation results of our model are more superior than UNet, but there are also areas with obvious segmentation errors, which can be further explored in future work.

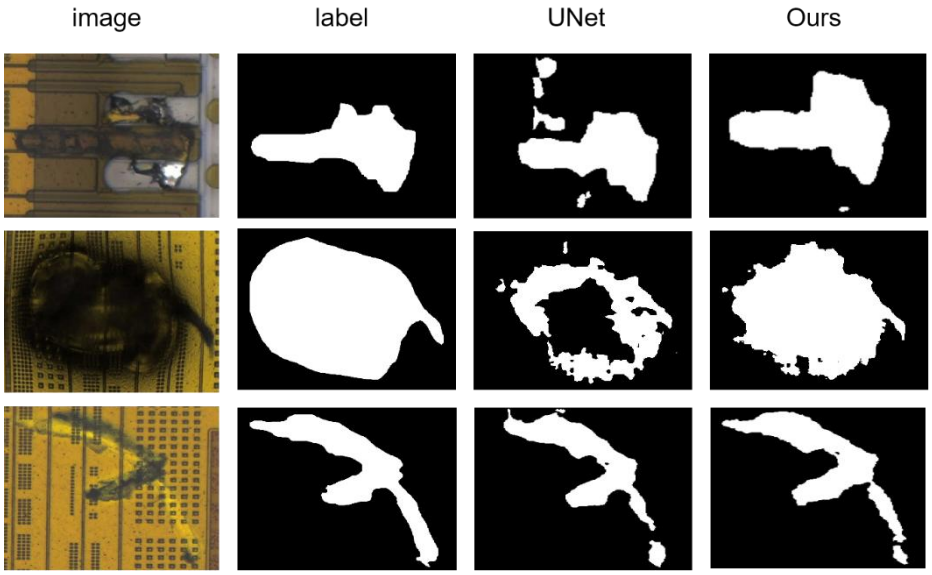

Fig. 3 Segmentation Results of The Proposed Model and UNet Based on Semiconductor Dataset

We performed ablation experiments on depth-wise large convolution kernels to evaluate the impact of convolution within depth-wise large convolution kernels on the model. The results are shown in Table 2. If the depth-wise convolution or the depth-wise dilation convolution is removed, the performance of the model will decrease. The model performance will decrease the most when there is no depth-wise dilation convolution. Therefore, the ablation experiment proves the importance of each part in the deep large convolution kernel.

Table 2. Ablation Experiment

Numble	Mean IOU	Dice
Without depth-wise convolution	0.6015	0.7295
Without depth-wise dilation convolution	0.5800	0.6993
SPLConv-UNet	0.6285	0.7423

4. Summary

A lightweight module was designed and implemented on the UNet model to perform experiments and evaluate its performance on the semiconductor defect dataset. The experimental results demonstrated that the module has fewer parameters and calculations compared to existing methods, which indicates superior performance. Additionally, this module can complement other model compression methods and be integrated into a more lightweight network architecture.

References

[1] Kai Chen, Jiaqi Wang, et al. Mmdetection: Open mmlab detection toolbox and benchmark. CoRR, abs/1906.07155, 2019.

[2] Mengzhou Xia, Zexuan Zhong, and Danqi Chen. Structured pruning learns compact and accurate models. arXiv preprint arXiv:2204.00408, 2022.

[3] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficientconvolutional networks through network slimming. In Proceedings of the IEEE international conference on computer vision, pages 2736 – 2744, 2017.

[4] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning,trained quantization and huffman coding. arXiv preprint arXiv:1510.00149, 2015.

[5] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In ECCV, pages 525 – 542. Springer, 2016.

[6] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. Advances in neural information processing systems, 27, 2014.

[7] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition, pages 11953–11962, 2022.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

[9] Andrew G Howard, Menglong Zhu,et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[10] Daquan Zhou, Qibin Hou, Yunpeng Chen, Jiashi Feng, andShuicheng Yan. Rethinking bottleneck structure for efficient mobile network design. In European Conference on Computer Vision, pages 680–697. Springer, 2020.

[11] Qiulin Zhang, Zhuqing Jiang, Qishuo Lu, Jia’nan Han,Zhengxin Zeng, Shang-Hua Gao, and Aidong Men. Split to be slim: An overlooked redundancy in vanilla convolution. arXiv preprint arXiv:2006.12085, 2020.

[12] Guo M H, Lu C Z, Liu Z N, et al. Visual attention network[J]. Computational Visual Media, 2023, 9(4): 733-752.