SCGNet: Shifting and Cascaded Group Network

Hao Zhang, Shenqi Lai, Yaxiong Wang, Zongyang Da, Yujie Dun, and Xueming Qian

***Abstract*— Many lightweight networks have been proposed for resource-limited applications, however, they cannot be efficiently applied to neural-network processing units (NPUs) due to the limited operations supported by the NPUs, and few works focus on efficient network design on the NPUs. The basic blocks of networks such as MobileNetV2 and RegNet use smaller convolution kernels with relatively small receptive fields, which are not conducive to capturing large-scale spatial information. To address this weakness, we propose Shifting and Cascaded Group (SCG) block, where we cascade group convolutions with larger kernels to exploit multi-scale information and propose shifting group convolution to communicate channel information between different groups. Besides, we carefully devise our architecture guided by some principles and finally build a very efficient network called Shifting and Cascaded Group Network (SCGNet) on NPUs. To verify the superiority of our method, we conduct extensive experiments on various tasks including image classification, object detection, human pose estimation, person re-identification, and semantic segmentation to comprehensively evaluate the performance. Results on widely used datasets such as ImageNet, PASCAL VOC, COCO, MPII, Market-1501, DukeMTMC-ReID, CUHK03, and Cityscapes demonstrate that the proposed network is a more effective network on the corresponding vision tasks.**

***Index Terms*— Lightweight networks, NPU, image classification, object detection, human pose estimation, person re-identification, semantic segmentation.**

I. Introduction

With the development of computer vision, lightweight networks have attached much attention due to the urgent requirement for mobile applications. As a result, so many excellent lightweight networks like MobileNet series [3], [4], [5], ShuffleNet series [6], [7], *etc.*, have been widely used due to their fewer parameters and satisfactory FLOPs. For hardware deployment, the Neural Network Processing Unit (NPU) is a popular choice because it is specifically designed for mobile applications, and is widely used to replace CPU and GPU in robotics and edge computing with extremely low power consumption.

To achieve a faster inference, a straightforward thought is to adapt the classical lightweight models to NPU. However, the performance and efficiency of existing models on NPU are unsatisfactory. Some typical lightweight networks, such as [1], [3], [4], [8], [9], and [10] cannot reach an ideal inference speed on NPU. Reference [6], [7], [11], and [12] even cannot be deployed because of the unsupported operation. For example, the ShuffleNet series could not be embedded in NPU due to the unsupported "channel shuffle" operation. Few works focus on the inference speed of NPU, this motivates us to devise an efficient and effective network on NPU.

An important truth about NPUs is that the lower computational complexity does not always mean a faster inference. In Table I, we report the FLOPs of some well-known lightweight networks and the inference speed on NPUs. As shown in this table, MobileNet series have much smaller FLOPs and parameters compared with ResNet-18 [13], but the inference speed on NPU is slower. The reason behind this observation is a fact that the widely used depthwise convolution significantly increases the memory access cost (MAC), which is an important factor for efficient model design on NPUs. In contrast, RegNetX [14] achieves more efficient inference with fewer FLOPs, which is an intuitive result. Compared with ResNet and MobileNet, RegNetX introduces group convolution to build the network. Given this, we potentially think the group convolution can well reduce the computation cost and not bring much MAC burden.

The above observations inspire us to use the group convolution operation as the basic operation for better efficiency. However, the group convolution achieves a faster inference at the cost of performance dropping, just the RegNetX-200MF shown in Table I. RegNet and MobileNet series only use two 1×1 convolutions and one 3×3 convolution to form the basic unit of the networks, but this form has a relatively small receptive field and lacks enough spatial information. Therefore, we propose to cascade two 3×3 group convolutions to get a bigger reception field. However, two cascaded group convolutions result in information exchange within only the corresponding groups, with barriers between different groups. To solve this problem, we replace the first group convolution with our newly proposed shifting group convolution, which repeats circular shifting procedures to adjust the grouping situation.

Manuscript received 22 August 2022; revised 9 November 2022 and 3 January 2023; accepted 3 February 2023. Date of publication 22 February 2023; date of current version 6 September 2023. This work was supported in part by the NSFC under Grant 62272380; in part by the Science and Technology Program of Xi'an, China, under Grant 21RGZN0017; in part by the Natural Science Foundation in Shaanxi Province of China under Project 2021JQ-289; and in part by the China Postdoctoral Science Foundation under Project 2021M700533. This article was recommended by Associate Editor J.-H. Xue. *(Hao Zhang and Shenqi Lai contributed equally to this work.) (Corresponding authors: Yujie Dun; Yaxiong Wang; Xueming Qian.)*

Hao Zhang and Zongyang Da are with the Smiles Laboratory, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China (e-mail: zhanghao520@stu.xjtu.edu.cn; dzy1134483011@stu.xjtu.edu.cn).

Shenqi Lai is with Fabu, Hangzhou, Zhejiang 310030, China (e-mail: laishenqi@fabu.ai).

Yaxiong Wang is with the Zhejiang Laboratory, Hangzhou, Zhejiang 311100, China (e-mail: wangyx15@stu.xjtu.edu.cn).

Yujie Dun is with the School of Information and Communication, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: dunyj@mail.xjtu.edu.cn).

Xueming Qian is with the Ministry of Education Key Laboratory for Intelligent Networks and Network Security and the Smiles Laboratory, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China (e-mail: qianxm@mail.xjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2023.3246999.

Digital Object Identifier 10.1109/TCSVT.2023.3246999

TABLE I
A COMPARISON OF DIFFERENT MODELS ON THE IMAGENET. FLOPS, PARAMS, TOP-1 ACCURACY AND FPS ON NPU ARE REPORTED. FPS REPRESENTS INFERENCE SPEED IN FRAMES PER SECOND

Models	FLOPs	Params	Top-1 Acc	FPS(NPU)
ResNet-18 [13]	1.8G	11.68M	71.2%	200.02
MobileNetV2 1.4× [4]	587M	6.08M	73.3%	101.51
MobileNext 1.0× [15]	300M	3.40M	74.0%	107.64
RegNetX-200MF [14]	205M	2.70M	68.9%	**294.64**
RegNetX-400MF [14]	410M	4.41M	72.7%	**191.13**
RegNetX-600MF [14]	613M	6.21M	74.1%	**159.87**

It exchanges information between different groups, and then realizes channel information communication between groups.

In this paper, we design a novel Shifting and Cascaded Group (SCG) block and build a more efficient lightweight network named Shifting and Cascaded Group Network (SCGNet) based on the SCG blocks. To realize faster inference speed on NPU, reduce memory access cost and efficiently use group convolution, we cascade our newly proposed shifting group convolution and group convolution with larger kernels. Although the exchange of information between groups using one shifting group convolution is limited, stacking a series of SCG blocks makes the channel information exchange between different groups feasible and effective. Besides, we carefully devise our architecture guided by some principles and finally build a very efficient network on NPUs.

We expect this work could serve as a solid baseline for future NPU-focused lightweight models. Our contributions can be summarized as follows:

- **A Shifting and Cascaded Group (SCG) block**. We design an effective and efficient block named SCG block containing our newly proposed shifting group convolution and one group convolution with larger kernels, which can effectively collect information across spatial and group dimensions.
- **A NPU-friendly lightweight network (SCGNet).** We build a very efficient lightweight network on NPU using our SCG blocks. Our model achieves an accuracy improvement of 2.0 points and a 107% improvement on MobileNetV2 with similar FLOPs. SCGNet also achieves an accuracy improvement of 3.8 points compared to MobileNetV3, which is even about **2 times** faster.
- **Excellent Performance on mainstream vision tasks.** Extensive experiments on a wide range of tasks including image classification, object detection, human pose estimation, person re-identification, and semantic segmentation show that our network SCGNet is an excellent backbone.

The rest of this paper is organized as follows: In Section II, related works are briefly reviewed. We describe how to construct SCGNet as our method in Section III. Extensive experiments are shown in Section IV. Conclusions are drawn in Section V.

II. RELATED WORKS

In recent years, deep networks have improved network performance by increasing the size of the model, *e.g.*, ViT [16], ConvNeXts [17], *etc.* But these networks cannot even be deployed on the NPU of RK3399PRO due to their large model size. As a result, lightweight networks have played an important role in mobile chips with limited computing power and memory, especially NPUs for neural network operations and applications. Therefore, a lot of work need to be done to explore the trade-off between accuracy and speed when designing deep neural network architectures.

A. Compression and Acceleration of Models

Many deep learning networks achieve high-precision performance by designing complex structures or deepening the depth of the network. Admittedly, these methods are very useful in some areas. However, complex networks consume a lot of time in inference, and they are difficult to meet real-time requirements. However, the compression and acceleration of models are very important in many real-time applications, which are generally divided into pruning, quantization, distillation, and structural design.

Pruning can be divided into synaptic pruning [18], neuron pruning [19], weight matrix pruning [20], and other methods, whose general idea is to set the unimportant parameters in the weight matrix to 0 and combine the sparse matrix to carry out storage and calculation. Quantization [21] is a very common method of model compression. It greatly reduces the size of the model by reducing the 32-bit floating point to 8-bit or even 1-bit. To some extent, quantization can achieve model compression but has little effect on model speedup. Knowledge distillation [22] is widely used in the training of efficient neural networks. The essence of knowledge distillation is the learning of the student network from the teacher network. The structure of the student network is simple, and it is necessary to learn useful information from the teacher network and then obtain comparable results with the teacher network. In this paper, we mainly focus on the structural design.

B. Lightweight Networks Structural Design

As for the structural design, there are many very classic works, which greatly reduce the size of the model and the number of operations, and the loss of accuracy is small. SqueezeNet [23] only uses 1×1 convolution in the squeeze layer and reduces the input channel of 3×3 convolution, which significantly reduces the number of parameters. Later, MobileNet [3] uses depthwise separable convolutions to build lower latency and smaller networks. ShuffleNet [6] generalizes group convolution and depthwise separable convolution in a novel form and utilizes "channel shuffle" to greatly reduce computation cost while maintaining accuracy, which is the first work investigate the usage of "channel shuffle" in small model design. MobileNetV2 [4] proposes a new lay module named the inverted residual with linear bottleneck, which reduces the need for main memory access. ShuffleNetV2 [7] introduces a simple operator called channel split and reduces element-wise operations to speed up. IGCV3 [24] uses the Interleaved Low-Rank Group Convolutions, which consists of a channel-wise spatial convolution, and a low-rank group point-wise convolution, and gets the better performance. C-GhostNet [8] proposes a Ghost module, which can be taken as a plug-and-play component to upgrade existing convolutional neural networks. Similarly, G-GhostNet [8] uses cheap

operations to exploit stage-wise redundancy and achieves a good trade-off between accuracy and latency for GPU. Besides, [5], [14], [15], *etc.*, also pay attention to structure design with the aim of compression and acceleration of models.

With these previous research and work, we rethink the design of the basic units of the lightweight network, and also draw some design lessons from related work, especially [7], [14].

C. Low Latency on Mobile Chips

Artificial intelligence (AI) mobile chips are mainly divided into two categories, one can complete the training and reasoning of neural networks, and the other can complete the acceleration of reasoning. Currently, in terminal applications, users pay more attention to inference speed. That is to say, real-time performance is a requirement for mobile tasks. Developers can train models on CPUs or GPUs and deploy them directly on AI mobile chips. Among them, NPU is an inference acceleration chip, which has been widely used in mobile phones, mobile robots, unmanned driving, and other fields. Compared with traditional chips, NPU has a more specialized hardware design and lower power consumption due to relatively fixed algorithms and application scenarios.

Recently, some researchers began to search for low-latency network structures influenced by the network structure search algorithm [25]. MobileNetV3 [5] applies squeeze and excite as effective tools in mobile models and applies the hardware-aware network architecture search. MobileNext [15] verifies the effectiveness of the sandglass block by adding it into the search space of neural architecture search method DARTS [26], which is also a successful case of neural network structure search. Some typical lightweight networks, such as [1], [3], [4], and [8] cannot assign the ideal inference speed on NPU. Even though MobileNetV3 [5] and RegNetY [14] prove that using Squeeze-and-Excite bottleneck proposed in [27] can significantly improve the lightweight network accuracy, its speed tested on NPU is very low as shown in Fig. 1. [6], [7] even cannot be deployed on some NPUs because the "channel shuffle" operation is not well supported. Few works have focused on model inference speed on NPU. Therefore, this paper focuses on lightweight network design with low latency on NPU.

III. Proposed Approach

In this section, we will first introduce our motivation in section III-A. Then we propose the shifting group convolution in section III-B. We build Shifting and Cascaded Group block in section III-C. In section III-D, we construct three networks with different parameter configurations to meet various applications. Hereinafter, the details of each part would be elaborated.

A. Motivation

In deep convolution neural network, a larger spatial receptive field often means a better performance [28] due to

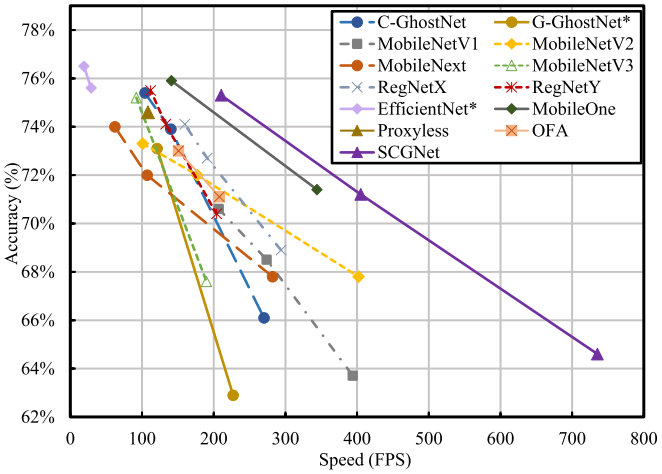


Fig. 1. The trade-off between FPS and top-1 accuracy for several SOTA models on ImageNet. Except that the input image size of EfficientNet [1] and OFA [2] are consistent with [1] and [2] respectively, the input size of other networks is 224 × 224. All speeds are tested on the NPU of RK3399Pro. * represents the result is our implementation, which is consistent with the training settings and data enhancement strategy of SCGNet. SCGNet is our proposed network.

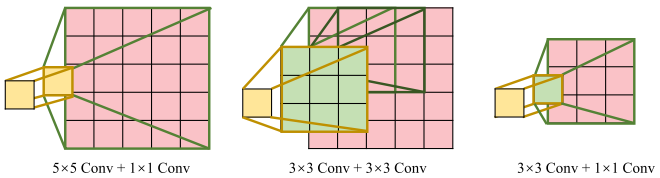


Fig. 2. The receptive fields obtained by different convolution combinations. (a) One 5×5 convolution and one 1×1 convolution. (b) Two 3×3 convolutions. (c) One 3 × 3 convolution and one 1 × 1 convolution.

the broader view and richer information. Intuitively, simply employing a convolution with a large kernel is a naïve strategy to harvest a wider receptive field. However, this solution would introduce more parameters, and the mobile device NPU may not be able to withstand the additional resource consumption that comes with it. To trade-off the effectiveness and computation complexity in a resource-limited environment, many methods only employ 3×3 convolution and 1×1 convolution interchangeably [3], [4], [14]. Nevertheless, the spatial receptive field of the combined mode of such convolution kernels is usually too narrow to perceive wider contexts, thus, existing models on NPU are still unsatisfactory. Given the above, the problem is how to expand the receptive field without putting too much burden on memory and computation.

We notice that two cascaded 3×3 convolutions provide the same field as 5×5 convolution, but with fewer parameters, as shown in Fig. 2. And vanilla convolutions are not efficient on a mobile platform, *e.g.*, NPUs. The group convolution achieves a good trade-off between accuracy, FLOPs, and inference speed on NPUs as mentioned in Section I. These observations motivate us to introduce cascaded 3×3 group convolutions for a larger receptive field. However, simply stacked group convolution cannot perform the information exchange between different groups. To address this problem, we propose shifting group convolution, which could enable the information exchange between different groups of group

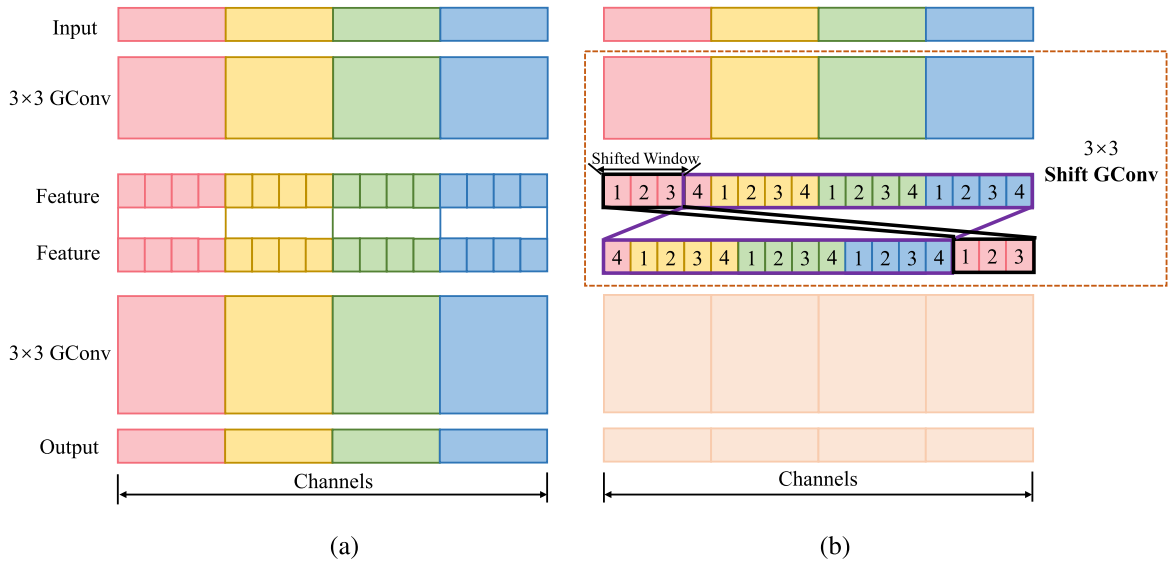


Fig. 3. Comparison of the effects of group convolution and shifting group convolution. (a) Two cascaded group convolutions. (b) One shifting group convolution and one group convolution. Shift GConv means shifting group convolution.

convolution without additional parameters or computation. To pursue better efficiency, we carefully devise our network following some principles of mobile-friendly architecture design. Finally, an efficient and effective network (SCGNet) on NPU is built. In the following, the details of each module will be presented.

B. Shifting Group Convolution

Given the motivation in the previous subsection, we decide to use a combination of two 3×3 group convolutions to obtain a larger receptive field and devise the shifting group convolution to realize channel information communication between groups.

Specifically, shifting group convolution contains a group convolution and a shifting operation. The input feature first passes through the group convolution, and then the shift operation is performed to adjust the channel groups. As shown in Fig. 3 (b), the shift operation is a repeated circular shifting procedure, where the first channel is moved to the tail in each shifting. And the repeated times are determined by the output channels and the group size: $t = \lfloor c_{out}/(4 * G) \rfloor$, where t is the number of repeated shifting, c_{out} and G are the number of channels of output features and group number, respectively, $\lfloor \cdot \rfloor$ is the round down function. Let $c_{out} = G \times N$. For a more intuitive and simple display, we set $G = 4$ and $N = 4$ in Fig. 3 (b) to illustrate the shifting group convolution with 3 repeated shifting. The output feature map $O = [O_{11}, O_{12}, \ldots, O_{44}]$ is modified to $[O_{14}, O_{21}, \ldots, O_{44}, O_{11}, O_{12}, O_{13}]$ by the shifting operation.

Even though shifting group convolution only achieves information exchange between adjacent groups, when shifting group convolutions are stacked, more information between different groups will be exchanged, thereby capturing more channel information relationships between groups. Moreover, compared with group convolution, shifting group convolution does not increase any computation and parameters.

Discussion: Wu et al. [29] propose end-to-end trainable shift-based modules based on "shift" operations, which are used as alternatives to spatial convolutions. However, our shift operation in shifting group convolution is different from theirs. Details are as follows:

1) Different Motivations: Wu et al. [29] propose FLOP-free "shift" to replace the space convolution with a large computational complexity, and achieve stronger performance with fewer parameters. But we are to avoid "information cocoon rooms" between two group convolutions, and promote information exchange between channels.

2) Different Implementations: Wu et al. [29] perform shift operations in spatial dimensions, but we perform shift operations in the channel dimension. Reference [29] is implemented by designing the underlying operators, which cannot be ideally supported on the NPU.

3) Different Information Utilization Rate: Some information on spatial dimensions is abandoned after the shift operation of [29]. The shift operation in [29] is equivalent to depth-wise convolution, which sets the parameters of the convolution kernel to 0 to achieve shift and inevitably results in information loss. However, we use a repeated circular shifting procedure in the channel dimension, and the information of all channels will be retained.

C. Shifting and Cascaded Group Block

In the last subsection, we present our basic operation, *i.e.*, shifting group convolution. In this subsection, we will describe how to use cascaded group convolutions and shifting group convolutions to build a more efficient neural network basic unit, *i.e.*, Shifting and Cascaded Group Block, including SCG convolution block and SCG downsampling block.

1) SCG Convolution Block: As shown in Fig.4 (a), assume that the number of channels is N in the input feature map and the output feature map has M channels. The steps of our basic SCG block are as follows:

1) The number of channels N of the input feature map is changed to $\frac{M}{2}$ by 1×1 convolution.

2) The 3×3 shifting group convolution is further applied without changing the channels of feature maps channels. And

TABLE II
NETWORK CONFIGURATIONS FOR SCGNET-S, SCGNET-M AND SCGNET-L. THEY ARE BASED ON SCG BLOCKS AND TARGET AT DIFFERENT COMPUTING RESOURCES AND MEMORY SPACES

Layer	Output Size	Kernel size	Stride	SCGNet-S		SCGNet-M		SCGNet-L	
				Repeat	Output channels	Repeat	Output channels	Repeat	Output channels
Image	224×224				3		3		3
Conv1	112×112	3×3	2	1	8	1	12	1	16
Conv2	112×112	3×3	1	1	8	1	16	1	16
Conv3	56×56	3×3	2	1	16	1	24	1	32
Conv4	56×56	3×3	1	1	16	1	24	1	32
Stage5	28×28		2	1	64	1	120	1	192
			1	1		2		3	
Stage6	14×14		2	1	128	1	240	1	384
			1	2		4		6	
Stage7	7×7		2	1	256	1	480	1	768
			1	1		1		2	
Conv8	7×7	1×1	1	1	1024	1	1024	1	1024
GlobalPool	1×1	7×7		1		1		1	
FC1				1	1024	1	1024	1	1024
FC2				1	1000	1	1000	1	1000
FLOPs				82M		240M		613M	
Params				2.68M		3.65M		6.22M	

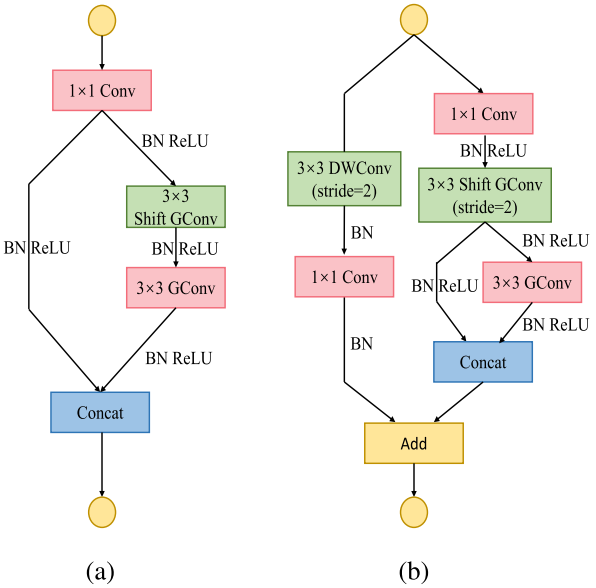


Fig. 4. Shifting and Cascaded Group Blocks. (a) SCG convolution block. (b) SCG downsampling block. Batch Normalization (BN) is one operation for data, ReLU is the activation function of neurons. Shift GConv means shifting group convolution.

the stride of the 3×3 shifting group convolution is set to 1. The use of shifting group convolution brings more information communication between groups.

3) One 3×3 group convolution is followed to further fuse the features. The two group convolutions used consecutively also enlarge the receptive field.

4) The output feature maps from step 1) and 3) are concatenated together to get a feature map with M channels. In this way, more information can be preserved.

In step 4), we do not use additional residual connections, which further reduces the memory access cost, consequently, the inference can be accelerated. Experiments in subsection IV-A well validate this claim.

2) SCG Downsampling Block: To facilitate feature compression and build a complete architecture, we also design an SCG block with downsampling operations. As shown in Fig.4 (b), it has two branches. As for the first branch, assume that the input feature map has N channels and the output feature map has M channels. The detailed design steps are as follows:

1) The number of channels N of input feature map is changed to $\frac{M}{2}$ by 1×1 convolution.

2) Then, a 3×3 shifting group convolution with stride 2 is followed to achieve the downsampling, while the output channels remain unchanged, *i.e.*, $\frac{M}{2}$.

3) Next, one 3×3 group convolution is also utilized to further fuse the feature and the output channels are still $\frac{M}{2}$.

4) We fuse the output features of 2) and 3) to produce a feature map with M channels.

The second branch is used to make up for the information loss of downsampling. Additional detailed steps for SCG block with spatial down-sampling are as follows:

1) The input feature map size is reduced to half by one 3×3 depthwise convolution with stride 2.

2) An 1×1 convolution is followed to perform the channel fusion.

3) The element-wise addition is used to enable information communication between two branches.

D. Towards More Efficient Networks

With our SCG blocks, we realize channel information communication and capture a wider context. And then we finally develop three networks: SCGNet-S, SCGNet-M, and SCGNet-L, which are based on SCG blocks and target different computing resources and memory spaces. SCGNet-S is our most lightweight model for better efficiency, while the SCGNet-L composes of more blocks, pursuing better performance.

Inference efficiency is an important concern for mobile applications, therefore, we attempt to further adjust the architectures for better efficiency. Guided by some principles proposed by [7] and [30], we also make the following improvements when building SCGNet: 1) Reduce the depth of the network as much as possible. 2) Reduce the channels of vanilla convolutions to abate activations. 3) Adjust the input and output channels of convolutions to be the same if possible. Following these principles, we finally build three efficient variants of SCGNet, Table II gives the detailed

architecture configurations. As shown in Table II, there are four 3×3 convolutions at the beginning of SCGNet. Then there are three stages of structure, which consist of stacking of SCG blocks. In each stage, the stride of the first block is set to 2, and the other blocks are set to 1. Then a 1×1 convolution is used to expand the channels followed by a global average pooling layer and two fully-connected layers to predict the final score for each category.

The structural differences of the three networks are as follows: First, the number of output channels of the first four 3×3 convolutions is different. Second, we set the group number of group convolution to 4 in SCGNet-S, 6 in SCGNet-M, and 8 in SCGNet-L. Besides, the number of SCG blocks included in each stage is also different, SCGNet-L has more SCG blocks while SCGNet-S is configured with the fewest SCG blocks.

IV. Experiments

In this section, we evaluate the performance of the proposed network SCGNet for image classification and test the inference speed of SCGNet on the NPU of RK3399PRO. Besides, we use SCGNet-L to replace the backbone of wide vision tasks methods,[1] such as SSD [31] and SSD-Lite [4] for object detection task, SimpleBaseline [32] for human pose estimation, MGN [33] for person re-identification and Deeplabv3plus [34] for semantic segmentation to compare with other state-of-the-art networks. All of these experiments are verifying SCGNet is an effective and efficient lightweight network for many vision tasks.

A. Ablation Study

In this subsection, we first study the following cases to validate the proposed Shifting Group Convolution. **A.** Directly cascading two 3×3 group convolutions. **B.** Our shifting group convolutions. **C.** We also discuss the case of introducing additional residual learning to show the efficiency of our architecture.

1) Dataset: The ImageNet dataset is first introduced in [35]. It is a well-known large-scale image classification dataset containing over 1.2 million training images and 50,000 validation images belonging to 1000 categories. We train the model on the training set and evaluate our method on the validation set separately.

2) Experimental Setup: For a fairer comparison, all experimental settings and data augmentation strategies are kept consistent with ShuffleNetV2 [7]. Not using "Cascaded Group" in the experiments means that we replace the second 3×3 group convolution of the main branch of the SCG block with a 1×1 vanilla convolution. Not using "Shifting Group" in the experiments means that we replace the 3×3 shifting group convolution of the main branch of the SCG block with 3×3 group convolution.

3) Results Analysis: As shown in Table III, when the 3×3 shifting group convolution is replaced by a 3×3 group convolution, the performance of all our variants drops. For example, the Top-1 accuracy of SCGNet-L deteriorates from 75.3% to 74.9%, which indicates that the shifting group convolution promotes the information exchange between channels and realizes the channel information exchange between groups. When the cascaded group convolution is further canceled, the performance of SCGNet-L drops by 0.3 points, and SCGNet-M and SCGNet-S get worse as well, which indicates that the cascaded 3×3 group convolution obtains the performance improvement brought by the large receptive field. We also find that the FPS sacrifice brought by the shifting group convolution is 3.81,1.97,1.42, respectively, which is particularly small. So these tiny sacrifices are not worth mentioning compared to the performance improvement it brings.

TABLE III

Performance of SCGNet on the ImageNet Dataset Under Different Ablation Settings. "C-Group" Represents Cascaded 3×3 Group Convolution. "S-Group" Represents Shifting Group Convolution

Models	Ablation Settings		FLOPs	Params	Top-1 Acc	FPS(NPU)
	C-Group	S-Group				
SCGNet-S	✘	✘	79M	2.62M	64.0%	747.40
	✔	✘	82M	2.68M	64.3%	739.10
	✔	✔	82M	2.68M	**64.6%**	735.29
SCGNet-M	✘	✘	239M	3.57M	70.7%	424.09
	✔	✘	240M	3.65M	70.9%	407.17
	✔	✔	240M	3.65M	**71.2%**	405.20
SCGNet-L	✘	✘	601M	6.12M	74.6%	225.07
	✔	✘	613M	6.22M	74.9%	211.95
	✔	✔	613M	6.22M	**75.3%**	210.53

TABLE IV

Comparison of Additional Residual Learning Effects. We Report Top-1 Accuracy and FPS on NPU

models	Residual Learning	Top-1 Acc	FPS(NPU)
SCGNet-M	✘	71.2%	**405.20**
	✔	**72.1%**	276.85
SCGNet-L	✘	75.3%	**210.53**
	✔	**76.1%**	124.52

We also compare the performance of SCGNet with additional residual learning, that is, introducing an identity branch in Fig. 4 (a): adding the input map to the output of the original structure. As shown in Table IV, we can see that the accuracy of SCGNet with residual learning achieves a certain improvement, but the inference speed is reduced by at least 30%. In particular, SCGNet-M increases accuracy by 0.9 points, and FPS is reduced by 128.35 after adding additional residual learning. But SCGNet-L is 4.1 points ($4\times$ *vs.* 0.9) higher than SCGNet-M, and FPS decreases by 194.67 ($1.5\times$ *vs.* 128.35), which means that increasing the size of the SCGNet achieves a better trade-off between accuracy and inference speed. Actually, we can equip the residual learning for higher precision, however, too much computational burden will be introduced. Specially, adding additional residual learning increases the number of branches of the network. The information of each branch has to be stored to compute the next tensor in the graph, which increases the memory access cost (MAC) [38] and brings computational burden. Besides, additional residual learning introduces more element-wise operations, which is non-negligible [7] and not conducive to inference speed. Therefore, we finally abandon

[1] SCGNet-L is employed for better performance, in our practice, SCGNet-L is a better trade-off between performance and speed.

the residual learning branch in our SCG block, to achieve a good trade-off between effectiveness and efficiency.

B. Image Classification

In this subsection, we compare the performance and inference speed of SCGNet with other state-of-the-art networks on image classification. For the inference speed, since we focus on the NPU-based mobile applications, so the frame per second (FPS) on RK3399Pro is employed as the speed metric, and we use the top-1 accuracy for performance comparison. The choice of dataset and experimental setup is consistent with the previous subsection.

1) Results Analysis: To show the improvement of SCGNet in terms of accuracy and speed more intuitively, we draw the accuracy-speed map of it and SOTA lightweight networks in Fig. 1. As we can see, SCGNet gets the fastest speed on NPU with the same accuracy as the other models compared with MobileNet series [3], [4], [5], [15], C-GhostNet [8], G-GhostNet [8], RegNetX [14], RegNetY [14], EfficientNet [1], OFA [2], Proxyless [37], MobileOne [38], and RegNetY [14].[2] Particularly, SCGNet has a greatly obvious advantage in speed compared to EfficientNet. Different networks perform differently on different hardware platforms, which is related to the underlying design of the hardware and the resource capacity of the hardware. Then some existing networks designed for hardware have their own more suitable hardware, as follows: MobileNetV3 [5] for mobile phone CPUs, FBNet [36] for Samsung S8, MobileOne [38] for iPhone 12, OFA [2] for Samsung Note10, *etc.*Therefore, it is normal for different networks to show an accuracy-speed trade-off inconsistent with the original paper on the NPU.

More specifically, as shown in Table V, SCGNet-L gets 1.4 points gain in accuracy and is 51% faster than C-Ghost. SCGNet-L gets 3.1 points gain in accuracy compared with IGCV3, however, the latter cannot be deployed on NPU. It gets 2.0 points gain in accuracy and 107% faster than MobileNetV2 1.4× with similar FLOPs. And SCGNet-L is 32% faster than RegNetX-600MF with 1.2 points gain in accuracy. Particularly, SCGNet-L also performs better compared to some hardware-ware NAS backbones, *e.g.*, OFA [2], FBNet-B [36] Proxyless [37], Single-One-Shot [12]. Our SCGNet variants also show great superiority of efficiency on NPU. SCGNet-M gets 3.8 points gain in accuracy and is even about 2 times faster than MobileNetV3 Small, which is even based on NAS. SCGNet-S gets 4.3 points improvement in accuracy compared with ShuffleNetV2 0.5×, while ShuffleNetV2 cannot be applied on the NPU of RK3399PRO because of the "channel shuffle" operation. Besides, SCGNet-S achieves 0.9 points gain in accuracy with 86% faster than MobileNetV1 0.5×. And SCGNet-S achieves 0.2 points gain in accuracy compared with CondenseNetV2-A, while CondenseNetV2-A also cannot be applied on the NPU of RK3399PRO because of the "channel shuffle" operation. Above all, benefiting from the ability of our network to capture multi-scale information and large-range information communications between channels of different groups, we get better performance and inference efficiency simultaneously on image classification task.

[2]ShuffleNet series, CondenseNetV2 [11], IGCV3 [24], Single-One-Shot [12] cannot be deployed on NPU of EK3399PRO because of the "channel shuffle" operation, so we cannot show them in Fig. 1.

TABLE V

COMPARISON WITH THE STATE-OF-THE-ART METHODS ON IMAGENET. THE FRAME PER SECOND (FPS) ON NPU AND THE TOP-1 ACCURACY ARE REPORTED. * REPRESENTS THE RESULT IS OUR IMPLEMENTATION, WHICH IS CONSISTENT WITH THE TRAINING SETTINGS AND DATA ENHANCEMENT STRATEGY OF SCGNET

Models	FLOPs	Params	FPS(NPU)	Top-1 Acc
ShuffleNetV2 0.5× [7]	41M	1.4M	-	60.3%
ShiftNet-B [29]	371M	1.1M	-	61.2%
MobileNetV1 0.5× [3]	149M	1.3M	394.48	63.7%
CondenseNetV2-A [11]	46M	2.0M	-	64.4%
SCGNet-S	82M	2.7M	**735.29**	**64.6%**
MobileNetV3 Small [5]	65M	2.5M	190	67.4%
RegNetX-200MF [14]	205M	2.7M	294.64	68.9%
ShuffleNetV2 1.0× [7]	146M	2.3M	-	69.4%
ShiftNet-A [29]	1400M	4.1M	-	70.1%
MobileNetV1 1.0× [3]	584M	4.2M	207.43	70.6%
SCGNet-M	240M	3.7M	**405.20**	**71.2%**
ShuffleNet 1.5× [6]	300M	3.3M	-	71.5%
IGCV3 [24]	347M	3.5M	-	72.2%
ShuffleNetV2 1.5× [7]	307M	3.5M	-	72.6%
OFA [2]	103M	5.0M	151.38	73.0%
G-GhostNet* [8]	1020M	11.2M	121.43	73.1%
MobileNetV2 1.4× [4]	587M	6.1M	101.51	73.3%
C-GhostNet [8]	141M	5.2M	139.59	73.9%
MobileNext 1.0× [15]	300M	3.4M	107.64	74.0%
FBNet-B [36]	295M	4.5M	136.25	74.1%
RegNetX-600MF [14]	613M	6.2M	159.87	74.1%
Proxyless [37]	320M	4.0M	108.20	74.6%
Single-One-Shot [12]	329M	3.4M	-	74.7%
SCGNet-L	613M	6.2M	**210.53**	**75.3%**

C. Object Detection

In this subsection, we use SCGNet-L to replace the backbone of SSD network [3] and the backbone of SSD-Lite [4] to further evaluate the performance and speed of SCGNet on object detection, which is the same as [53]. We compare the mean precision (mAP) and inference speed on the NPU of RK3399PRO with other classical and state-of-the-art networks.

1) Datasets: The classic datasets PASCAL VOC and MS COCO are taken to perform the object detection experiments. PASCAL VOC contains 20 object categories, each image has pixel-level segmentation annotations, bounding box annotations, and object class annotations. The union of VOC2007 and VOC2012 trainval dataset is used for training, which has 16,551 pictures. The VOC2007 testset, in which there are 4,952 images, is used for testing. The MS COCO dataset is a large-scale object detection, segmentation, key-point detection, and captioning dataset. Based on community feedback, the training/validation split is 118K/5K. And we train the model on the training part and test on the validation part.

2) Experimental Setup: As for PASCAL VOC, we choose SSD as the basic network. The input size is resized to 300 × 300 with data augmentation strategy such as flip, brightness adjustment, contrast adjustment, and saturation adjustment, *etc*. And we adopt the cosine learning rate decay method and use SGD optimizer. The momentum is set to 0.9 and the weight decay is set to 0.0005. We set the learning rate starting from 0.001 to 0 within 120 epochs. As for MS COCO, we choose SSD-Lite as the basic network. The input image size is resized

TABLE VI
RESULTS ON PASCAL VOC DATASET. THE FPS ON NPU AND MAP ARE REPORTED

Models	Backbones	Params	FLOPs	Input Size	mAP (%)	FPS (NPU)
MobileNet-SSD	MobileNet	6.9M	1520M	300*300	67.55%	105.23
MobileNetV2-SSD	MobileNetV2	5.0M	1100M	300*300	70.01%	89.28
ShuffleNetV2-SSD	ShuffleNetV2 1.5×	5.1M	953M	300*300	69.70%	-
SCGNet-SSD	SCGNet-L	7.9M	1648M	300*300	**71.69%**	**111.74**

TABLE VII
RESULTS ON MS COCO DATASET. THE FPS ON NPU AND MAP ARE REPORTED

Models	Backbones	Params	FLOPs	Input Size	mAP (%)	FPS (NPU)
MobileNet-SSDLite	MobileNet	5.1M	1300M	320*320	22.20%	100.56
MobileNetV2-SSDLite	MobileNetV2	4.3M	805M	320*320	22.10%	90.48
IGCV3-SSDLite	IGCV3	4.0M	830M	320*320	22.20%	-
ShuffleNetV2-SSDLite	ShuffleNetV2 1.5×	4.3M	802M	320*320	22.20%	-
SCGNet-SSDLite	SCGNet-L	6.0M	1492M	320*320	**23.05%**	**112.51**

TABLE VIII
COMPARISONS OF LIGHTWEIGHT NETWORK ON MPII DATASET. THE FPS ON NPU, PCKH@0.5 AND PCKH@0.1 ARE REPORTED

Backbones	Params	FLOPs	Hea	Sho	Elb	Wri	Hip	Kne	Ank	PCKh@0.5	PCKh@0.1	FPS(NPU)
MobileNext [15]	9.0M	7089M	95.9	93.5	84.4	77.0	84.5	77.2	73.0	84.4	26.7	67.68
MobileNetV2 [4]	5.3M	6138M	96.1	93.9	85.3	78.2	86.4	79.2	73.4	85.4	28.2	88.78
RegNetX-600MF [14]	9.9M	6736M	**96.2**	94.2	85.7	78.8	86.1	79.2	74.7	85.7	28.4	90.06
SCGNet-L	8.6M	6941M	95.9	**94.3**	**86.4**	**79.4**	**86.8**	**80.0**	**75.4**	**86.1**	**29.0**	**109.21**

to 320 × 320. The data augmentation strategy and training strategy are consistent with [24]. We terminate the training at 240 epochs. The learning rate starting is set as 0.004 and discounted by 0.1 every 60 epochs. Besides, in this subsection, all backbones are pretrained on ImageNet.

3) Results Analysis: The results on the PASCAL VOC dataset are shown in Table VI. Compared with ShuffleNetV2-SSD, SCGNet-SSD improves mAP by 1.99 points, while ShuffleNetV2-SSD cannot be applied on NPU. Particularly, SCGNet-SSD gets 4.14 points gain in terms of mAP compared with MobileNet-SSD and achieves a comparable speed on NPU. The results on MS COCO dataset are shown in Table VII. SCGNet-SSDLite gets 0.85 points gain in terms of mAP compared with MobileNet-SSDLite with 11.95 points higher FPS on NPU. Besides, SCGNet-SSDLite gets 0.85 points gain in terms of mAP compared with IGCV3-SSDLite. Consequently, SCGNet also gets better performance and speed on object detection than other methods.

D. Human Pose Estimation

In this subsection, we use some lightweight architectures to replace the backbone of human pose estimation network proposed in [32] to evaluate the performance and speed of SCGNet. We use PCKh (head-normalized probability of correct keypoint) score as the performance metric. If the key-joint is located within αl pixels of the real position, the human key-joint prediction is thought as correct. α is a constant. l corresponds to 60% of the diagonal length of the real head bounding box. The PCKh@0.5 ($\alpha = 0.5$) score and PCKh@0.1 ($\alpha = 0.1$) score are reported. We also test the inference speed of NPU on RK3399PRO.

1) Dataset: The MPII dataset is extracted from online videos, which has about 25K images. Each image has uncertain number of people with over 40K people annotated. We train the model on a subset of MPII training set [54] and evaluate on 2975 images as validation set [54].

2) Experimental Setup: All backbones are pretrained on ImageNet. And The input size is cropped to 256 × 256. The data augmentation is the same as [54], which includes random scale, random rotation, and flipping. We also use Adam optimizer. The total epoch is set to 150 and the base learning rate is set as 1e-3, and is dropped to 1e-4 and 1e-5 at the 90th and 125th epochs. Besides, we use the provided person bounding boxes rather than detecting them. When testing, we compute the average of the heatmaps of the original and flipped images as the final heatmap.

3) Results Analysis: From Table VIII, as a backbone of human body pose estimation, SCGNet-L once again surpasses the state-of-the-art networks on both accuracy and speed. SCGNet-L gets 1.7 points gain in terms of PCKh@0.5, 2.3 points gain on PCKh@0.1 and 61% faster speed compared with MobileNext when they replace the backbone of [32]. SCGNet-L even outperforms RegNetX-600MF by 0.4 points and 0.6 points respectively in terms of PCKh@0.5 and PCKh@0.1 with 21% faster speed. Specifically, we also show some visualization of results on MPII dataset in Fig. 5. As we can see, compared with results of RegNetX-600MF, SCGNet-L can get better pose estimation performance in some challenging cases such as occlusion, fuzzy, confusing actions, and so on.

E. Person Re-Identification

In this subsection, we employ SCGNet as the backbone of MGN [33] with some tricks from BagOfTricks [41] for the vision task of person Re-identification [55]. We com-

TABLE IX

COMPARISON WITH THE STATE-OF-THE-ART METHODS ON THE MARKET-1501 AND THE DUKEMTMC-REID. WE COMPARE THE MAPAND RANK-1 ACCURACY WITH OTHER STATE-OF-THE-ART METHODS

Method	Backbone	Market-1501		DukeMTMC-ReID	
		Rank-1	mAP	Rank-1	mAP
VCFL+ [39]	ResNet-50	91.9	77.0	82.7	65.7
MGN [33]	ResNet-50	95.7	86.9	88.7	78.4
OBM [40]	ResNet-50	93.2	80.4	85.3	71.7
BagOfTricks [41]	ResNet-50	94.5	85.9	86.4	76.4
HPM [42]	ResNet-50	94.2	82.7	86.6	74.3
IANet [43]	ResNet-50	94.4	83.1	87.1	73.4
DGNet [44]	ResNet-50	94.8	86.0	86.6	74.8
DSAReID [45]	ResNet-50	95.7	87.6	86.2	74.3
OSNet [46]	ResNet-50	94.8	84.9	86.6	74.8
CRAN [47]	ResNet-50	94.9	84.9	87.6	74.7
MHN-6 [48]	ResNet-50	95.1	85.0	89.1	77.1
PISNet [49]	ResNet-50	95.6	87.1	88.8	78.7
CBDB-Net [50]	ResNet-50	94.4	85.0	87.7	74.3
DPN[51]	GogglesNet-S.	95.6	87.4	89.2	78.3
MGN	MobileNetV2[4]	95.3	87.0	88.5	77.7
MGN	ShuffleNetV2 1.5[7]×	94.9	87.6	89.3	78.6
MGN	SCGNet-L	**95.9**	**88.1**	**89.4**	**79.4**

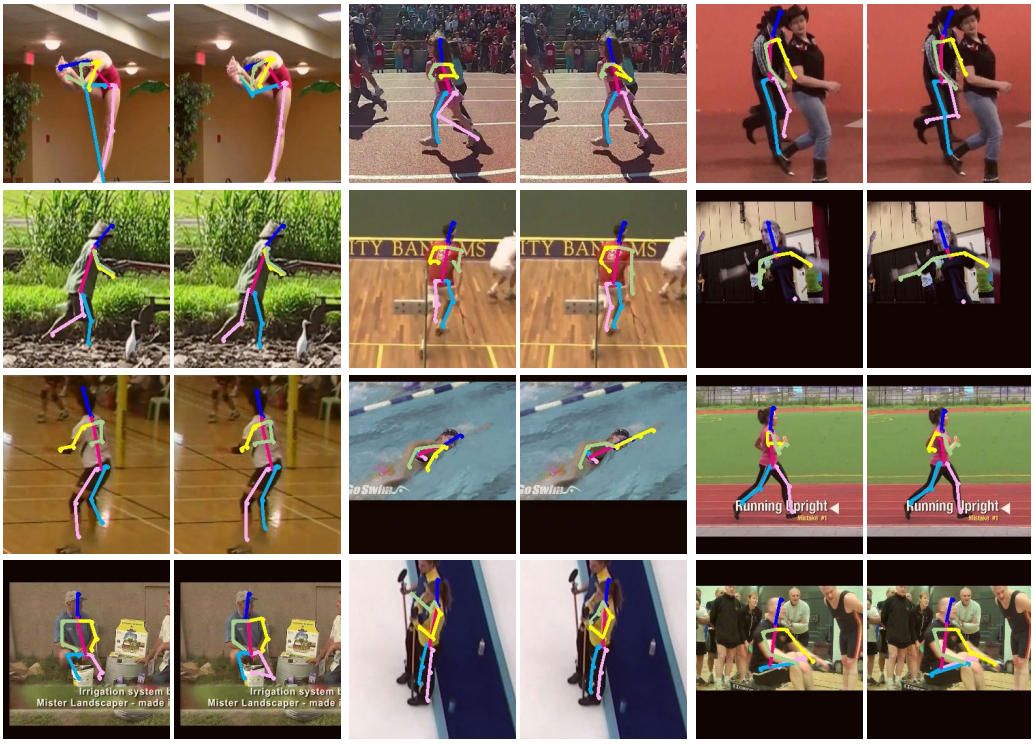


Fig. 5. Comparisons of the visualization of results on MPII. RegNetX-600MF [14] (left side) and our SCGNet-L (right side) are as the backbone of SimpleBaseline [32].

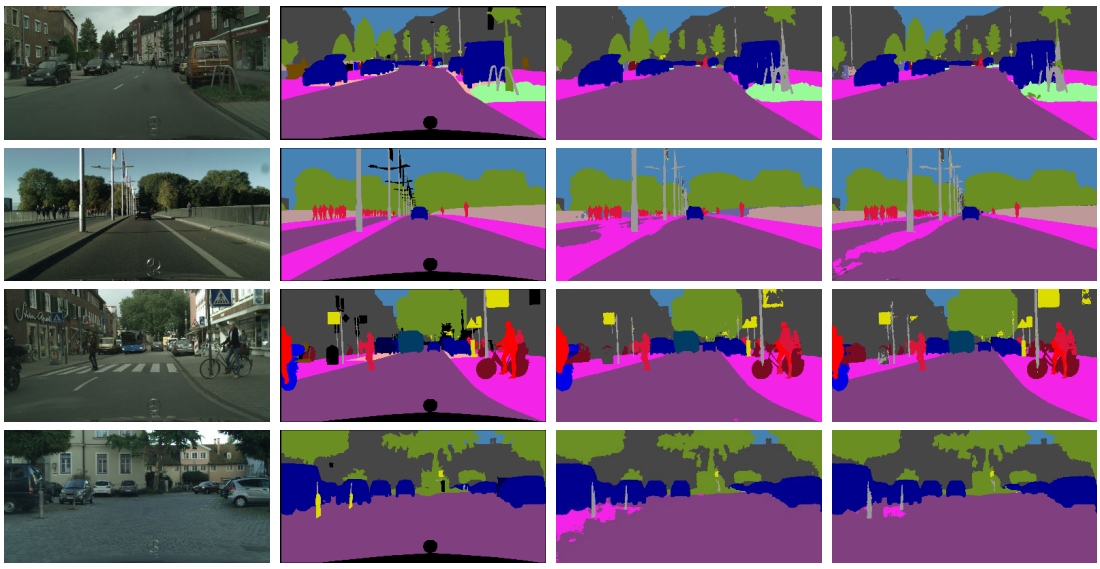
pare the mAP and Rank-1 accuracy with other state-of-the-art methods to show the accuracy and effectiveness of SCGNet.

1) Datasets: The Market-1501 [56] is a large-scale public dataset, which contains 1501 identities captured by six different cameras. The dataset is split into two parts: 751 identities with 12,936 images are utilized for training and the remaining 750 identities with 19,732 images are used for testing, in which there are 3,368 query images.

The DukeMTMC-ReID [57] dataset is created from high-resolution videos from 8 different cameras, which is also spilt into two parts: 702 identities with 16,522 images for training and the other 702 identities with 17,661 images for testing, in which there are 2,228 query images.

The CUHK03 [58] dataset consists of 1,467 different identities deployed by 6 campus cameras, which contains 13,164 images of 1,467 identities. We evaluate this dataset both using hand-labeled and DPM-detected bounding boxes. We adopt the training/testing protocol in [59].

2) Experimental Setup: We first resize the input images to 384×128. Then we use zero-pad operation with 10 pixels and randomly crop them in the size of 384×128. Besides, the data augmentation includes random erasing, horizontal flipping, and normalization are applied. All networks are trained with SGD algorithm with 4 GPUs. We set the total number of the epoch as 90 and set the batch size as 64. We set the weight decay as 0.0001 and momentum as 0.9. We adopt the linear warm-up strategy for the first 5 epochs with the learning rate



(a) Image (b) Ground Truth (c) MobileNetV2 (d) SCGNet

Fig. 6. Comparisons of the visualization of results on Cityscapes. MobileNetV2 and SCGNet replace the backbone of Deeplabv3plus [34] separately.

TABLE X

COMPARISON WITH THE STATE-OF-THE-ART METHODS ON THE CUHK03-NP. WE COMPARE THE MAP AND RANK-1 ACCURACY WITH OTHER STATE-OF-THE-ART METHODS

Method	Backbone	Labeled		Detected	
		Rank-1	mAP	Rank-1	mAP
VCFL+ [39]	ResNet-50	-	-	61.29	54.26
MGN [33]	ResNet-50	68.20	67.40	66.80	66.00
OBM [40]	ResNet-50	-	-	65.00	62.19
HPM [42]	ResNet-50	-	-	63.90	57.50
PPS [52]	ResNet-50	75.64	72.66	73.73	70.56
OSNet [46]	ResNet-50	-	-	72.30	67.80
CRAN [47]	ResNet-50	72.70	68.20	69.90	64.90
MHN-6 [48]	ResNet-50	77.20	72.40	71.70	65.40
MGN	MobileNetV2[4]	75.57	72.02	74.21	69.09
MGN	ShuffleNetV2 1.5[7]×	74.57	70.86	72.93	68.69
MGN	SCGNet-L	**77.57**	**74.27**	**75.29**	**71.58**

ranging from 0.001 to 0.1. Then we train the network with the cosine learning rate decay.

3) Results Analysis: As shown in Table IX and Table X, our method gets better results than the state-of-the-art methods. Compared to the MGN with MobileNetV2, MGN with SCGNet-L achieves 0.6 points gain in terms of Rank-1 accuracy, 1.1 points gain in terms of mAP on Market-1501 dataset, and 0.9 points gain in terms of Rank-1 accuracy, 1.7 points gain in terms of mAP on DukeMTMC-ReID dataset compared with MobileNetV2. SCGNet also gets better results on the CUHK03-NP dataset. Not only does our model outperform light-weight backbones, such as MobileNetV2 and ShuffleNetV2, but it also outperforms some methods that use ResNet-50 as the backbone, which is nearly 7 times bigger than SCGNet.

F. Semantic Segmentation

In this subsection, we use SCGNet and MobileNetV2 as drop-in replacements for the backbone of the Deeplabv3plus [34] to evaluate the performance of SCGNet on semantic segmentation tasks. We compare the performance with metric mean IoU and pixel accuracy.

TABLE XI

THE RESULTS OF SEMANTIC SEGMENTATION ON CITYSCAPES TESTED ON THE VALIDATION SUBSET

Backbone	Mean IoU	Pixel acc
MobileNetV2[4]	70.30	94.88
RegNetY-600MF[14]	70.01	95.06
EfficientNet-B1 [1]	70.22	95.00
SCGNet-L	**71.50**	**95.13**

1) Dataset: The Cityscapes [60] is a large-scale dataset that focuses on semantic understanding of urban street scenes, which contains around 5000 fine annotated images labeled in 19 semantic classes and are divided respectively into 2,975 images for training, 500 images for testing.

2) Experimental Setup: The data augmentation strategy and training strategy are consistent with [34]. In addition, the output stride of the training and testing procedure is set as 16.

3) Results Analysis: As shown in Table XI, SCGNet-L attains a performance of 71.5% (1.2 points improvement) mIOU and 95.1% (0.25 point improvement) compared with MobileNetV2. And some visualization of results on Cityscapes is shown in Fig. 6, which shows that SCGNet gets more accurate segmentation.

V. Conclusion

In this work, we present a much more effective and efficient lightweight network for mobile applications. Our proposed shifting group convolution makes the channel information exchange between groups feasible and effective. And we cascade shifting group convolution and group convolution with larger kernels in the SCG block to capture multi-scale spacial information and improve the network performance. Finally, our SCGNet could achieve a better trade-off between accuracy and efficiency on NPU chips at the same time. To validate the superiority of our method, we perform extensive experiments on a wide variety of vision tasks including image classification, object detection, human pose estimation, person re-identification, and semantic segmentation. The results reveal that our proposed SCGNet is not only an effective and efficient network for NPU-focused mobile application but a better backbone for many vision tasks. We hope this work could serve as a solid baseline for future lightweight and NPU-focused network designs.

References

[1] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. ICML*, 2019, pp. 6105–6114.

[2] H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han, "Once-for-all: Train one network and specialize it for efficient deployment," in *Proc. ICLR*, 2020, pp. 1–15.

[3] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[4] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[5] A. Howard et al., "Searching for MobileNetV3," in *ICCV*, Oct. 2019, pp. 1314–1324.

[6] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[7] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *ECCV*, 2018, pp. 116–131.

[8] K. Han et al., "GhostNets on heterogeneous devices via cheap operations," *Int. J. Comput. Vis.*, vol. 130, no. 4, pp. 1050–1069, 2022.

[9] X. Li, S. Lai, and X. Qian, "DBCFace: Towards pure convolutional neural network face detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 1792–1804, Apr. 2022.

[10] T. Wen, Z. Ding, Y. Yao, Y. Wang, and X. Qian, "PicassoNet: Searching adaptive architecture for efficient facial landmark localization," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 28, 2022, doi: 10.1109/TNNLS.2022.3167743.

[11] L. Yang et al., "CondenseNet V2: Sparse feature reactivation for deep networks," in *Proc. CVPR*, Jun. 2021, pp. 3569–3578.

[12] Z. Guo et al., "Single path one-shot neural architecture search with uniform sampling," in *Proc. ECCV*, vol. 12361, 2020, pp. 544–560.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[14] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10428–10436.

[15] D. Zhou, Q. Hou, Y. Chen, J. Feng, and S. Yan, "Rethinking bottleneck structure for efficient mobile network design," in *Proc. ECCV*. Cham, Switzerland: Springer, 2020, pp. 680–697.

[16] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. ICLR*, 2021, pp. 1–22.

[17] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11966–11976.

[18] H. Tanaka, D. Kunin, D. L. K. Yamins, and S. Ganguli, "Pruning neural networks without any data by iteratively conserving synaptic flow," 2020, *arXiv:2006.05467*.

[19] T. Fujii, S. Sato, H. Nakahara, and M. Motomura, "An FPGA realization of a deep convolutional neural network using a threshold neuron pruning," in *Proc. Int. Symp. Appl. Reconfigurable Comput.* Cham, Switzerland: Springer, 2017, pp. 268–280.

[20] X. Ruan et al., "EDP: An efficient decomposition and pruning scheme for convolutional neural network compression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4499–4513, Oct. 2021.

[21] P. Wang, Q. Chen, X. He, and J. Cheng, "Towards accurate post-training network quantization via bit-split and stitching," in *Proc. ICML*, 2020, pp. 9847–9856.

[22] T. Wen, S. Lai, and X. Qian, "Preparing lessons: Improve knowledge distillation with better supervision," *Neurocomputing*, vol. 454, pp. 25–33, Sep. 2021.

[23] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016, *arXiv:1602.07360*.

[24] K. Sun, M. Li, D. Liu, and J. Wang, "IGCV3: Interleaved low-rank group convolutions for efficient deep neural networks," 2018, *arXiv:1806.00178*.

[25] L. Li and A. Talwalkar, "Random search and reproducibility for neural architecture search," in *Proc. 35th Uncertainty Artif. Intell. Conf.*, 2020, pp. 367–377.

[26] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," 2018, *arXiv:1806.09055*.

[27] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[28] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.

[29] B. Wu et al., "Shift: A zero flop, zero parameter alternative to spatial convolutions," in *Proc. CVPR*, Jun. 2018, pp. 9127–9135.

[30] P. Dollar, M. Singh, and R. Girshick, "Fast and accurate model scaling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 924–932.

[31] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[32] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proc. ECCV*, 2018, pp. 466–481.

[33] G. Wang, Y. Yuan, X. Chen, J. Li, and X. Zhou, "Learning discriminative features with multiple granularities for person re-identification," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 274–282.

[34] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. ECCV*, 2018, pp. 801–818.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[36] B. Wu et al., "FBNet: Hardware-aware efficient convnet design via differentiable neural architecture search," in *CVPR*, Jun. 2019, pp. 10734–10742.

[37] H. Cai, L. Zhu, and S. Han, "ProxylessNAS: Direct neural architecture search on target task and hardware," in *Proc. ICLR*, 2019, pp. 1–13.

[38] P. Kumar Anasosalu Vasu, J. Gabriel, J. Zhu, O. Tuzel, and A. Ranjan, "An improved one millisecond mobile backbone," 2022, *arXiv:2206.04040*.

[39] L. Zhang, F. Liu, and D. Zhang, "Adversarial view confusion feature learning for person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 4, pp. 1490–1502, Apr. 2021.

[40] Y. Chen, C. Zhao, and T. Sun, "Single image based metric learning via overlapping blocks model for person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 647–656.

[41] H. Luo, Y. Gu, X. Liao, S. Lai, and W. Jiang, "Bag of tricks and a strong baseline for deep person re-identification," in *Proc. CVPR Workshops*, Jun. 2019, pp. 1487–1495.

[42] Y. Fu et al., "Horizontal pyramid matching for person re-identification," in *Proc. AAAI*, 2019, vol. 33, no. 1, pp. 8295–8302.

[43] R. Hou, B. Ma, H. Chang, X. Gu, S. Shan, and X. Chen, "Interaction-and-aggregation network for person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9317–9326.

[44] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, "Joint discriminative and generative learning for person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2138–2147.
[45] Z. Zhang, C. Lan, W. Zeng, and Z. Chen, "Densely semantically aligned person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 667–676.
[46] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3702–3712.
[47] C. Han, R. Zheng, C. Gao, and N. Sang, "Complementation-reinforced attention network for person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3433–3445, Oct. 2020.
[48] B. Chen, W. Deng, and J. Hu, "Mixed high-order attention network for person re-identification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 371–381.
[49] S. Zhao et al., "Do not disturb me: Person re-identification under the interference of other pedestrians," in *Proc. ECCV*. Cham, Switzerland: Springer, 2020, pp. 647–663.
[50] H. Tan, X. Liu, Y. Bian, H. Wang, and B. Yin, "Incomplete descriptor mining with elastic loss for person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 1, pp. 160–171, Jan. 2021.
[51] H. Jin, S. Lai, Q. Tang, T. Zhu, and X. Qian, "MPPM: A mobile-efficient part model for object re-ID," *IEEE Trans. Multimedia*, early access, Sep. 23, 2022, doi: 10.1109/TMM.2022.3207895.
[52] Y. Shen et al., "A part power set model for scale-free person retrieval," in *Proc. IJCAI*, 2019, pp. 3397–3403.
[53] J. Yang, S. Lai, X. Wang, Y. Wang, and X. Qian, "Diversity-learning block: Conquer feature homogenization of multibranch," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 2, 2022, doi: 10.1109/TNNLS.2022.3214993.
[54] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5693–5703.
[55] H. Jin, S. Lai, and X. Qian, "Occlusion-sensitive person re-identification via attribute-based shift attention," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2170–2185, Apr. 2022.
[56] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1116–1124.
[57] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. ECCV*. Amsterdam, The Netherlands: Springer, 2016, pp. 17–35.
[58] W. Li, R. Zhao, T. Xiao, and X. Wang, "DeepReID: Deep filter pairing neural network for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 152–159.
[59] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1318–1327.
[60] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proc. CVPR*, Jun. 2016, pp. 3213–3223.

Hao Zhang received the B.S. degree in information engineering from Xi'an Jiaotong University in 2021, where he is currently pursuing the M.S. degree in information and communication engineering. His research interests include neural network architecture design, face alignment, and human pose estimation.

Shenqi Lai received the B.S. degree in computer science from Xi'an Jiaotong University, Xi'an, China, in 2015, and the M.S. degree from the School of Software Engineering, Xi'an Jiaotong University, in 2018. His research interests include image retrieval and neural network acceleration.

Yaxiong Wang received the B.S. degree from Lanzhou University, Lanzhou, China, in 2015, and the Ph.D. degree from the School of Software Engineering, Xi'an Jiaotong University, in 2021. His current research interests include few-shot learning, cross-media retrieval, generative learning, and superpixel segmentation.

Zongyang Da received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 2020, where he is currently pursuing the M.S. degree with the Smiles Laboratory. His current research interests include single-image super-resolution, real-world super-resolution, generative adversarial networks, and object detection.

Yujie Dun received the Ph.D. degree in information and communication engineering from Xi'an Jiaotong University (XJTU), Xi'an, China, in 2016. After her Ph.D. degree, she visited as a Visiting Scholar at Washington University in St. Louis, St. Louis, MO, USA, from 2017 to 2018, where she was a Post-Doctoral Researcher from 2018 to 2019. She is currently working as an Associate Professor with the School of Information and Communication, XJTU.

Xueming Qian received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree in electronics and information engineering from Xi'an Jiaotong University, Xi'an, in 2008. He was a Visiting Scholar with Microsoft Research Asia, Beijing, China, from 2010 to 2011. He is currently a Full Professor.