SDGFormer: An Efficient Convolution Network Structurally Similar to Transformer

Chaohao Wen[1], Xun Gong[1,2,3,4]*
[1]School of Computing and Artificial Intelligence, SWJTU, Chengdu, China
[2]Engineering Research Center of Sustainable Urban Intelligent Transportation, Ministry of Education, China
[3]National Engineering Laboratory of Integrated Transportation Big Data Application Technology, SWJTU, Chengdu, China
[4]Manufacturing Industry Chains Collaboration and Information Support Technology Key Laboratory of Sichuan Province, SWJTU, Chengdu, China
xgong@switu.edu.cn

***Abstract*—Deep Neural Networks (DNN) have achieved extraordinary success in many visual recognition tasks. Visual Transformer (ViT), which is derived from Natural Language Processing (NLP), has achieved state-of-the-art (SOTA) results on many tasks due to its capability of capturing long-range dependencies in visual data. However, Existing ViT models are challenging to deploy on devices due to their massive computational consumption, huge memory overhead, and reliance on large datasets. In this work, we address these issues by replacing some computationally expensive and memory-intensive modules in ViT with standard Convolutional Neural Network (CNN) modules. Firstly, we propose an efficient Self-Attention module called SDG-Attention (SDGA) with linear space and time complexity, and an economical FeedForward Network (FFN) composed of group convolution and shuffle channel (SFFN). Then, we develop a lightweight CNN model with SDGA and SFFN, SDGFormer, which embraces several priors of ViT and is LayerNorm-Free. We evaluate SDGFormer on ImageNet-1K and Mini-ImageNet, and the SDGFormer-S achieves a comparable top-1 accuracy of 77.6% on ImageNet-1K with 9.1M parameters and 1.6 GFlops regimes. Moreover, our SDGFormer-T achieves SOTA performance on Mini-ImageNet with 83.3% accuracy, demonstrating good generalization on small datasets without extra data.**

***Index Terms*—ViT, CNN, Attention, FFN, Lightweight**

I. Introduction

Deep learning has experienced a period of stagnation due to its excessive computational overhead. However, the last decade has seen a resurgence in deep learning, thanks to hardware improvements and the introduction of many effective and innovative network architectures. It is now widely used in various computer vision tasks, including image classification [1], object detection [2], [3], and semantic segmentation [4], and has a profound impact on these tasks. Now, as researchers strive to achieve higher evaluation metrics, the computational overhead and memory footprint of models have become increasingly large. Thus, model lightweighting become an increasingly important research direction for expanding the application of deep learning.

In 2012, the introduction of the CNN-based AlexNet [1] marked a significant breakthrough in image classification, surpassing traditional feature-based machine learning methods on the ImageNet dataset. Later, ResNet [5] introduced residual structures which allows for deeper neural network designs. As a general rule, deeper networks have stronger feature extraction and representation capabilities. In recent years, the development of CNN has encountered a bottleneck. Meanwhile, the model structure design of NLP undergone a significant transformation with the rise of Transformer [6], replacing recurrent neural networks as the mainstream backbone framework. Transformer is also introduced into vision tasks [7], yielding state-of-the-art performance. It revolutionizes the design of network models in computer vision. However, some researchers noted that the Vision Transformer lacks inductive bias [8], which makes it reliant on large datasets. Furthermore, the computational complexity of Transformers is quadratic in terms of the number of patches, which can be computationally expensive when processing high-resolution images with a large number of patches.

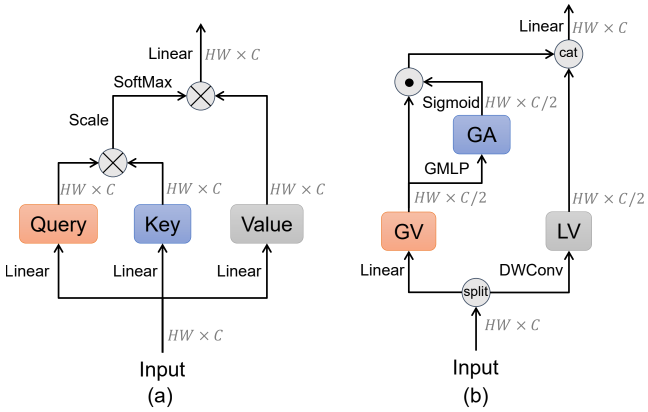

Fig. 1. Comparison between the vanilla Self-Attention (left) and the proposed SDGA (right). Instead of all features for global information interaction, we use half for global and the other half for local.

Recently, several ViT structures revisited the advantages of CNN structures to address the aforementioned issues. These structures, such as Swin Transformer [9], PVT [10], and CMT [11], leverage hierarchical structures of CNNs to improve themselves. To introduce inductive bias, Swin uses the local window attention, while CMT uses a hybrid structure of convolutioin and Transformer. Meanwhile, ConvNeXt [12],

*Corresponding author

based on ResNet, incorporates prior knowledge from Swin Transformer to design an optimal CNN structure. In this study, we propose a lightweight CNN architecture, named SDGFormer, which structurally resembles the Transformer model. To achieve this, we replace the time-consuming modules in Swin Transformer with convolution modules. Unlike ConvNeXt, SDGFormer retains the overall structure of Swin Transformer and includes SDGA and Shuffle-FFN (SFFN), corresponding to the Multi-Head Self-Attention (MHSA) and FFN in Swin Transformer, respectively. In summary, our main contributions are as follows:

- We propose the SDGA to capture long-range dependencies with linear time complexity, as shown in Figure 1(b). In comparison to Self-Attention shown in Figure 1(a), SDGA eliminates the Query, Key and the subsequent *softmax* function. The SDGA comprises two branches: one for capturing global information (GV: Global Value; GA: Global Attention) and the other for capturing local information (LV: Local Value).
- We propose a innovative FFN called SFFN, which utilizes Group Convolution and Shuffle Channel to significantly reduce the number of parameters and computations required.
- SDGFormer employs a novel combination of self-attention (SDGA) and FFN (SFFN), enabling the replacement of LayerNorm in ViT with BatchNorm without significant loss in accuracy, thereby facilitating model acceleration.

We conduct a comprehensive evaluation of SDGFormer on Mini-ImageNet and ImageNet-1K datasets. Without the need for additional data, SDGFormer outperforms the current state-of-the-art models on Mini-ImageNet in the same environment. This compensates for the weaker performance of Transformer models on smaller datasets. Moreover, SDGFormer-S achieves comparable performance with mainstream lightweight methods on ImageNet-1K, achieving a top-1 accuracy of 77.6% with 9.1M parameters and 1.6 GFlops of computation.

II. Related Work

As models become deeper and wider, they naturally require more computational resources and memory occupation. Current SOTA models on ImageNet-1K have computational and parametric volumes of several gigabytes. However, the computing power of existing hardware devices has not kept pace with the growth of model demand. Therefore, how to design an efficient network structure and reduce the amount of network parameters become one of the important research directions in the field of deep neural networks. There are currently two main directions of compression and acceleration of deep learning models: 1) model compression, such as quantization and pruning; 2) designing lightweight models directly. Lightweight model design is now one of the mainstream compression and acceleration methods. There are now two approaches: manual design based on the researcher's prior knowledge and automatic design search using some techniques (such as neural network architecture search [13]). In this paper, we present a new artificially designed lightweight model. We leverage the prior knowledge from ViT to design a CNN-based lightweight model with a structure that closely resembles that of the Swin Transformer.

A. Lightweight of CNN

AlexNet [1] and VGGNet [14] demonstrate that a deep convolutional network composed of convolutional layers, Batch-Normalization and activation layers can achieve remarkable results on many vision recognition tasks. GoogleNet [15] introduced a multi-branch neural network model that enhances the feature extraction capability of the model through the use of multiple basic convolution modules. This approach also led to the development of group convolution, which has been adopted by many subsequent lightweight solutions. ResNets [5] improves traditional CNNs by introducing residual connections that reduce the risk of gradient explosion and disappearance, which makes it possible to train models with deeper layers. ShuffleNet [16] uses group convolution to reduce the huge computational complexity of 1x1 pointwise convolution in the case of high feature dimensions. Meanwhile, it uses channel shuffling technology to reduce the impact of inter-channel information blocking caused by group convolution. ShuffleNetV2 [18] and GhostNet [19] use channel splitting and feature multiplexing technology to reduce the number of memory accesses and calculations. In comparison, this paper utilizes channel splitting to simultaneously extract local features and global information interaction. MobileNet [17] presents the depth-separable convolution, a special group convolution paradigm. In recent years, there have been many excellent lightweight networks searched by AutoML or artificial. EfficientNet [20] uses AutoML to search out a baseline network, and then adjusts the width, depth and input resolution of the network based on it. Regnet [21] is designed by manually observing and narrowing the model design space.

B. Lightweight of Transformer

It is well known that ViT heavily relies on large datasets and requires significant computational and memory resources. DeiT [22] created a new training scheme that could achieve better results directly on ImageNet-1K. PVT [10] introduced a pyramid structure in ViT, reducing computation and memory usage. Swin Transformer [9] also adopts the stage-wise architecture design similar to CNNs, and proposed a windowed attention mechanism. CMT [11] introduces hierarchy and depth-separable convolution instead of FFN and reduces the spatial size of Key and Value to achieve a lightweight MHSA. MetaFormer [23] presents $SpatialFC$ to calculate the attention by using the Pooling Layer in the first few stages of the model, which can significantly reduce the computation overhead. MobileFormer [24] is attempting to mitigate the computation by linking Transformer and MobileNet, mainly by using double branching to fuse the two results and using a similar technique to CMT by reducing the number of Key and Value.

ConvNeXt [12] improves ResNet by leveraging Swin Transformer's prior knowledge, modern training methods and using larger convolution kernels (As shown in Figure 3(a)). Similarly, NLNet [25] also incorporates the self-attention mechanism into CNN, which is analogous to ViT, but its computational overhead is immense. However, there is a lack of a lightweight CNN model with a structure similar to ViT. Thus, this paper replaces some time-consuming components in ViT with efficient CNN modules to obtain an efficient, lightweight CNN model. We propose a module that captures global information in linear complexity by adopting the idea of spatial MLP.

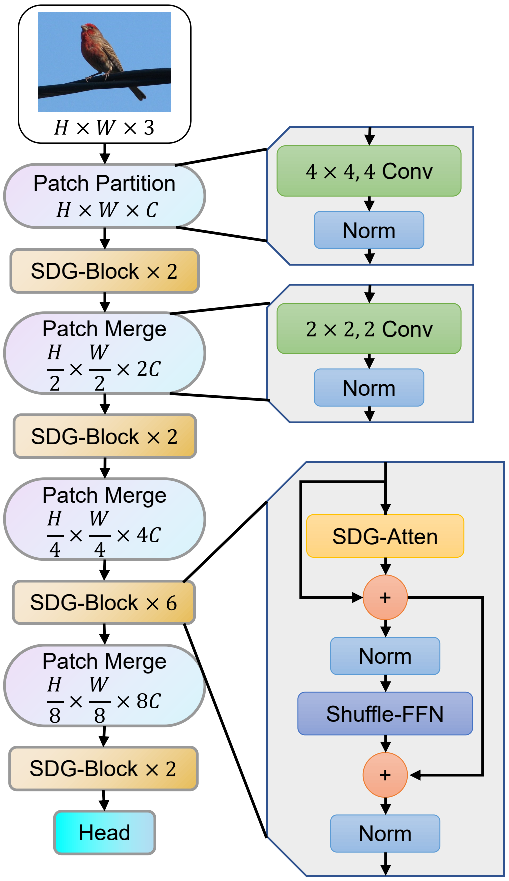

Fig. 2. Overall architecture of SDGFormer-T. Similar to the Swin Transformer, We adop a pyramid architecture described in Sec.III-A with our proposed SDGA and SFFN. Some ordinary layers are omitted for simplicity.

III. Approach

In our specific implementation, we employ **Split** along the channel, **Depth-separable** convolution to extract local information, and **Grid-based** Spatial MLP to capture long-range dependency, thus naming our method as SDGFormer. In this section, we provide a detailed description of the structure and components of the proposed SDGFormer.

A. Overall Architecture

Our intention is to construct a lightweight CNN model that takes advantage of the Transformer, which is analogous to the architecture of the Transformer but captures long-range dependency in linear time complexity. Similarly to ResNet and Swin Transformer, SDGFormer also adopts a staged architecture. The proposed SDGFormer is composed of Patch Partition, SDG-Block and Patch Merge. Initially, the input image is divided into non-overlapping patches through Patch Partition, which is equivalent to downsampling the image by 1/4, and then embedding the patch. Subsequently, there are four stages, and each Stage consists of $N\times$SDG-Block and a Patch Merge, but the last stage omits Patch Merge for the global pooling layer of head. When the feature map passes through the Patch Merge, the feature resolution is reduced to 1/2, and the feature dimension is expanded to 2 times. Different stages have different numbers of SDG-blocks and different num-groups. The proposed tiny version of SDGFormer (SDGFormer-Tiny, SDGFormer-T) is illustrated in Figure 2.

B. SDG-Block

In Transformer, each Block is composed of Layer Norm, MHSA and FFN, which can be divided into Pre-Norm, Post-Norm and Res-Post-Norm in SwinV2 according to the position of Norm. Pre-Norm places the Norm before MHSA and FFN, Post-Norm is placed after the residual connection, and Res-Post-Norm puts the Norm after MHSA and FFN. Pre-Norm is easier to train since it does not destroy the residual connection, but it may reduce the expressiveness of the model for degrading the network into a wide network. In contrast, Post-Norm training is relatively difficult due to gradient problems. To ensure that the model output is subject to a normal distribution, we improved it by placing a Norm before the FFN and connecting a Norm after the last residual connection. This method is shown in the right half of Figure 2. In this work, our proposed SDGA and SFFN correspond to the MHSA and FFN, respectively. With the two components above, the SDG-block can be formulated as:

$$\mathbf{Y}_i = \mathbf{X}_i + SDGA(\mathbf{X}_i), \tag{1}$$

$$\mathbf{X}_{i+1} = Norm(\mathbf{Y}_i + SFFN(Norm(\mathbf{Y}_i))), \tag{2}$$

where $\mathbf{X}_i$ is the input of the current Block, $\mathbf{X}_{i+1}$ represents the output of the current Block, and Norm represent Layer Normalization or Batch Normalization.

SDGA: For an input $\mathbf{X} \in R^{hw\times c}$, the original MHSA in ViT first generates the Query ($\mathbf{Q}$), Key ($\mathbf{K}$) and Value ($\mathbf{V}$) through the linear layer, where c is the channel dimensions, h and w are the resolution of the input. It is easy to prove that the linear layer is equivalent to the 1x1 convolution. Subsequently, MHSA computes the similarity between tokens through $\mathbf{Q}$ and $\mathbf{K}$, and the output of MHSA is the weighted average of Attention and V, as shown in Figure 3(b). This process can be expressed as:

$$MHSA(\mathbf{X}) = Softmax(\mathbf{QK^T})\mathbf{V}. \tag{3}$$

To obtain both local and global information interactions, Swin and MaxViT require the serial connection of two blocks, increasing the computation. In contrast, we propose a parallel

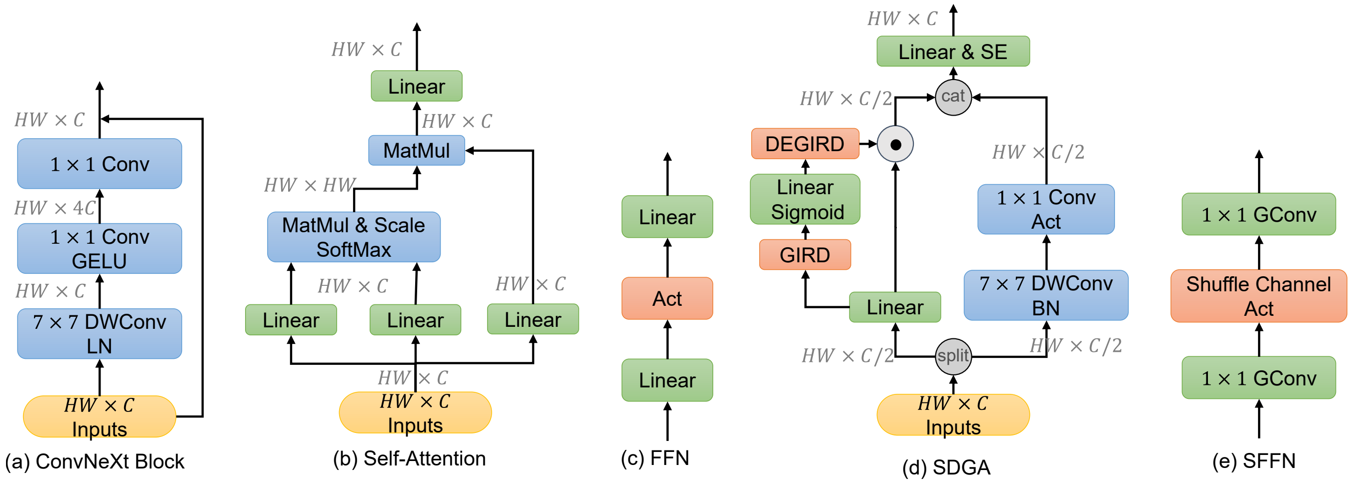

Fig. 3. The comparison of ConvNeXt-Block, SA, FFN, SDGA and SFFN. Our method employs the GRID Spatial Linear and Depthwise Convolution to encode spatial information in (d).

dual-branch module to achieve similar effects. To reduce the feature dimension, we divide the input feature map into two groups along the channel, which is inspired by the GhostNet. The first group uses Depth-Wise Separable Convolution (DWConv) to extract local features, which is composed of DWConv+BN+PWConv+Act, as shown in the blue part of Figure 3(d). The second group is used to capture long-range dependency. Motivated by MLP Mixer [26], we argue that the relationship between tokens can still be calculated when num-heads in MHSA is equal to the feature dimension. Thus, we flatten the spatial dimension, and the interaction between tokens is gained via linear layer. It is very easy to prove that this linear layer is equivalent to a 1x1 convolution with transpose the spatial dimension of the feature map to the channel dimension. Inspired by the MaxVit [27], we first *GRID* the feature map, and then form a new smaller feature map according to the position correspondence to mitigate the computation overhead for its large resolution, as shown in Figure 4. Aligned with Swin Transformer and MaxViT, the grid size is (7,7). After getting the interaction, we *DEGRID* it to the original feature map. At the end of SDGA, we concatenate the local and global features, then project it and to fuse the two branches' information.

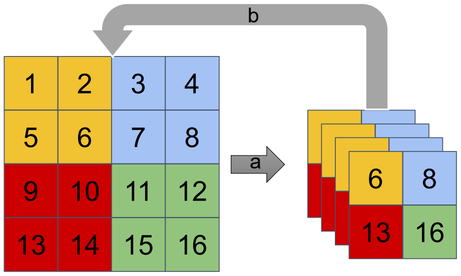

Fig. 4. The Grid Schematic Diagram (Arrow a: GRID, arrow b: DEGRID).

SFFN: In VIT, FFN first applies a linear layer followed by an activation function to expand the dimension, then reduces the dimension to original level through a linear layer, as shown in Figure 3(c). The dimension expansion ratio is set to 4. Although Point-Wise Convolution (PWConv) is used here, the computational overhead for the lightweight model is also considerable for the huge number of dimensions. The specific computational overhead will be analyzed in the next section. To reduce the computational overhead, we use the technology in ShuffleNet, replacing the original PWConv with group convolution and adding Shuffle Channel to eliminate the influence of group convolution blocking information between channels, as shown in Figure 3(e).

C. Complexity Analysis and Scaling Strategy

In this subsection, we analyze the time complexity of Windows-MHSA (WMSA) in Swin, SDGA, FFN and SFFN. For the sake of simplicity, we assume that all module inputs and outputs have the same shape.

Given $\mathbf{X} \in R^{hw \times c}$, the computational (FLOPs) complexity of WMSA and FFN can be formulated as:

$$\mathcal{O}(WMSA) = 4hwc^2 + 2M^2hwc, \tag{4}$$

$$\mathcal{O}(FFN) = 2hwc^2R, \tag{5}$$

where R is the expansion ratio of FFN, and M is the window size. Under the above setting, the FLOPs of SDGA and SFFN are as follows:

$$\mathcal{O}(SDGA) = \frac{3hwc^2}{2} + \frac{7^2 + M^2}{2}hwc, \tag{6}$$

$$\mathcal{O}(SFFN) = \frac{2hwc^2R}{G}, \tag{7}$$

where G is the number of groups in SFFN. Compared to the modules in Swin, the SDGA and SFFN have lower computational overhead, making them more suitable for terminal devices.

SDGFormer-T is the basic version of SDGFormer, and the depth of the four stages is (2,2,6,2), which is the same as Swin-Tiny. In SDGFormer, the G in the SFFN of each Stage

TABLE I
DETAILED SETTINGS OF SDGFORMER SERIES

	SDGFormer-Tiny	SDGFormer-Small
depths	(2, 2, 6, 2)	(2, 4, 8, 2)
width	(80, 160, 320, 640)	(80, 160, 320, 640)
groups	(4, 4, 4, 4)	(2, 2, 4, 4)
ratio	2	4

is the same. Assuming that the same G is used in two adjacent Stages, the time complexity of the SFFN of the next stage can be calculated as follows:

$$\mathcal{O}(SFFN) = \frac{2 \times hwc^2R}{4 \times G}. \tag{8}$$

For SDGFormer, there are four hyperparameters, including depth, width, G and R, which determine the model's size. The parameter quantity of the $Stage_i$ is quadruple that of the $Stage_{i-1}$. Typically, R is equal to 4, so in terms of the computation, the $Stage_{i-1}$ stage is twice as much as the $Stage_i$. Therefore, we can control the model's flops and parameter's size by adjusting the depth of the front and back stages. We build our model SDGFormer-T to have a similar model size and computation complexity with RegNetX-800MF. We have also introduced SDGFormer-Small (SDGFormer-S), similar to RegNetX-1.6GF, according to the proposed scaling strategy. The detailed hyper-parameters of the two models are shown in Table I.

IV. EXPERIMENTS

In this section, we evaluate the effectiveness of the proposed SDGFormer by conducting experiments on image classification tasks on ImageNet-1K and Mini-ImageNet. We first detail the experiment settings and compare the proposed SDGFormer with previous SOTA lightweight models on the aforementioned tasks, and then ablate the important design choices in SDGFormer on Mini-ImageNeet classification.

A. Experiment Settings

Datasets. ImageNet-1K contains about 1.2M training images with 1000 categories. We report the accuracy as the performance on an official standard validation set, which contains 50 images for each category. Meanwhile, to explore the effectiveness of the SDGFormer on small datasets, we also conduct image classification experiments on Mini-ImageNet, which contains 60,000 images and 100 different categories. We divide Mini-ImageNet into two parts with stratified sampling: training verification and testing, which contain 48000 and 12000 pictures, respectively. In order to save time and explore more, the ablation experiments are conducted on Mini-ImageNet, then the proposed SDGFormer is evaluated on ImageNet.

Training settings. We implement our model based on *Pytorch 1.12*, and the training code uses the reference training given in *torchvision*[1], which has successfully reproduced several official models open-sourced in torchvision, including this paper comparing RegNet and ConvNeXt. In training, we use SGD as the model optimizer with a warmup and cosine annealing schedule. Due to hardware and training time constraints, we used a mixed precision training on ImageNet-1K. The initial learning rate is 0.1-0.8, and the weight decay is 5e-5. The data preprocessing is the same as RegNet, which only contains image flipping, *RandomResizedCrop*, and normalization.

TABLE II
TOP-1 ACCURACY COMPARISON ON IMAGENET-1K

	flops(B)	params	accuracy
	(B)	(M)	ours+std [orig]
SDGFormer-T	0.8	6.3	76.2±0.10
REGNETY-800MF	0.8	6.4	76.3±0.09
EFFICIENTNET-B2	1.0	9.2	76.4±0.06 [79.8]
SDGFormer-S	1.6	9.1	77.6 ±0.12
REGNETY-1.6GF	1.6	11.2	78.0±0.13
EFFICIENTNET-B3	1.8	12.2	77.5±0.08 [81.1]
CMT-XS	1.5	15.2	76.2±0.09 [81.8]
PoolFormer-S12	1.8	11.9	75.3±0.13 [77.2]
ResNet50	4.1	25.6	77.8±0.13 [76.1]
ConvNeXt-T	4.5	28.6	77.9±0.13 [82.1]
Swin-T	4.5	28.3	77.5±0.11 [81.3]
MaxViT-T	5.6	30.9	76.7±0.12 [83.6]

B. Comparison with Other Approach

We compared the SDGFormer with the currently some outstanding lightweight models, including the pure CNN, pure ViT structure, and hybrid structure. We fairly compared them in groups according to similar computation and parametric size. We utilize the AdamW optimizer for Transformer-based methods, which often require longer training and regularization techniques. Additionally, we increase the number of training epochs from 100 to 120 and incorporate the TrivialAugmentWide data augmentation strategy and label smoothing to further improve performance. As shown in the Table II, the SDGFormer achieves a comparable result on Image-Net-1k. Moreover, the SDGFormer performs SOTA on the Mini-ImageNet, as shown in Table III, which proves that the SDGFormer is very effective for small datasets.

TABLE III
TOP-1 ACCURACY COMPARISON ON MINI-IMAGENET

	flops(B)	params	accuracy
	(B)	(M)	ours+std
SDGFormer-S	1.6	8.6	83.3±0.14
REGNETY-1.6GF	1.6	10.4	83.0±0.10
EFFICIENTNET-B3	1.8	10.9	82.6±0.08
ResNet50	4.1	23.7	82.7±0.10
ConvNeXt-T	4.5	27.9	76.5±0.13
Swin-T	4.5	27.6	73.5±0.16

The experimental results demonstrate that Transformer type models require long-term training, especially when the amount of data is limited. We will investigate whether the split scheme can alleviate or even solve this problem in our future work.

[1] https://github.com/pytorch/vision/tree/main/references

TABLE IV
ABLATION STUDY OF SDGFORMER

Methon	accuracy
LayerNorn+GELU	83.4±0.10
FFN	82.2±0.10
Local	81.4±0.11
Global	80.1±0.13
SDGFormer	83.3±0.14

C. Ablation Study

In this subsection, we verify the impact of BatchNorm, SFFN, and the combination of global and local features on the accuracy in SDGA, as shown in Table IV. The SDGFormer's accuracy reduction is negligible relative to the inference speed when the LayerNorn+GELU in ViT is replaced with the BatchNorm+GELU. Compared with FFN, our proposed SFFN has fewer parameters and lower computational overhead. Therefore, SFFN has a higher accuracy rate under the same computational overhead for it's deeper and wider networks.

V. CONCLUSION

This paper firstly investigates the lightweight convolutional model which is structurally similar to Swin Transformer. The SDGA inspired by GhostNet and MLP is capable of capturing long-range dependency with only linear time complexity. Moreover, inspired by ShuffleNet, we propose a more lightweight and novel FFN. Our ideas can also be used to improve models like Swin and MaxViT that require two consecutive modules to achieve global and local feature extraction. Furthermore, our studies on the placement of Normalization layers can provide a priori knowledge for subsequent researchers. Next, we will study the combination of the proposed modules and Transformers variants in the future.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to the anonymous reviewers for their insightful comments and suggestions that have helped to improve this paper. We would also like to extend my heartfelt appreciation to Zikun Yuan from Southwest Jiaotong University and Dr. Kang Zhao from Huawei for their invaluable guidance and assistance throughout the process. This work is partially supported by the Sichuan Science and Technology Program (2023YFG0267, 2023YFS0202), Natural Science Foundation of Sichuan Province (2023NSFSC1616), Fundamental Research Funds for the Central Universities (2682021ZTPY030, 2682022KJ045).

REFERENCES

[1] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

[2] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.

[3] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.

[4] Chen L C, Papandreou G, Kokkinos I, et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(4): 834-848.

[5] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[6] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[7] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.

[8] Park N, Kim S. How Do Vision Transformers Work?[C]//International Conference on Learning Representations, 2022.

[9] Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 10012-10022.

[10] Wang W, Xie E, Li X, et al. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 568-578.

[11] Guo J, Han K, Wu H, et al. Cmt: Convolutional neural networks meet vision transformers[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 12175-12185.

[12] Liu Z, Mao H, Wu C Y, et al. A convnet for the 2020s[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 11976-11986.

[13] Elsken T, Metzen J H, Hutter F. Neural architecture search: A survey[J]. The Journal of Machine Learning Research, 2019, 20(1): 1997-2017.

[14] Karen S, Andrew Z. Very Deep Convolutional Networks for Large-Scale Image Recognition[C]//In 3rd International Conference on Learning Representations, 2015.

[15] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.

[16] Zhang X, Zhou X, Lin M, et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 6848-6856.

[17] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017.

[18] Ma N, Zhang X, Zheng H T, et al. Shufflenet v2: Practical guidelines for efficient cnn architecture design[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 116-131.

[19] Han K, Wang Y, Tian Q, et al. Ghostnet: More features from cheap operations[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 1580-1589.

[20] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks[C]//International conference on machine learning. PMLR, 2019: 6105-6114.

[21] Radosavovic I, Kosaraju R P, Girshick R, et al. Designing network design spaces[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 10428-10436.

[22] Touvron H, Cord M, Douze M, et al. Training data-efficient image transformers & distillation through attention[C]//International conference on machine learning. PMLR, 2021: 10347-10357.

[23] Yu W, Luo M, Zhou P, et al. Metaformer is actually what you need for vision[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 10819-10829.

[24] Chen Y, Dai X, Chen D, et al. Mobile-former: Bridging mobilenet and transformer[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 5270-5279.

[25] Wang X, Girshick R, Gupta A, et al. Non-local neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7794-7803.

[26] Tolstikhin I O, Houlsby N, Kolesnikov A, et al. Mlp-mixer: An all-mlp architecture for vision[J]. Advances in neural information processing systems, 2021, 34: 24261-24272.

[27] Tu Z, Talebi H, Zhang H, et al. Maxvit: Multi-axis vision transformer[C]//Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV. Cham: Springer Nature Switzerland, 2022: 459-479.