Conv2Former: A Simple Transformer-Style ConvNet for Visual Recognition

Qibin Hou, *Member, IEEE*, Cheng-Ze Lu, Ming-Ming Cheng, *Senior Member, IEEE*, and Jiashi Feng

Abstract—Vision Transformers have been the most popular network architecture in visual recognition recently due to the strong ability of encode global information. However, its high computational cost when processing high-resolution images limits the applications in downstream tasks. In this paper, we take a deep look at the internal structure of self-attention and present a simple Transformer style convolutional neural network (ConvNet) for visual recognition. By comparing the design principles of the recent ConvNets and Vision Transformers, we propose to simplify the self-attention by leveraging a convolutional modulation operation. We show that such a simple approach can better take advantage of the large kernels ($\geq 7 \times 7$) nested in convolutional layers and we observe a consistent performance improvement when gradually increasing the kernel size from 5×5 to 21×21. We build a family of hierarchical ConvNets using the proposed convolutional modulation, termed Conv2Former. Our network is simple and easy to follow. Experiments show that our Conv2Former outperforms existent popular ConvNets and vision Transformers, like Swin Transformer and ConvNeXt in all ImageNet classification, COCO object detection and ADE20k semantic segmentation. Our code is available at https://github.com/HVision-NKU/Conv2Former.

Index Terms—Convolutional neural networks, vision transformer, convolutional modulation, large-kernel convolution

1 Introduction

The prodigious progress in visual recognition in the 2010s was mostly dedicated to convolutional neural networks (ConvNets), typified by VGGNet [1], Inception series [2], [3], [4], and ResNet series [5], [6], [7], [8], etc. These recognition models mostly aggregate responses with large receptive fields by stacking multiple building blocks and adopting the pyramid network architecture but neglect the importance of explicitly modeling the global contextual information. SENet series [9], [10], [11] break through the traditional design of CNNs and introduce attention-based mechanisms into CNNs to capture long-range dependencies, attaining surprisingly good performance.

Since 2020, Vision Transformers (ViTs) [12], [13], [14], [15], [16] further promoted the development of visual recognition models and show better results on the ImageNet classification and downstream tasks than the state-of-the-art ConvNets [17], [18]. This is because compared to convolutions that provide local connectivity, the self-attention mechanism in Transformers is able to model global pairwise dependencies, providing a more efficient way to encode spatial information as demonstrated in [19]. Nevertheless, the computational cost caused by the self-attention when processing high-resolution images is considerable.

Recently, an interesting work, named ConvNeXt [20], reveals that by simply modernizing the standard ResNet and using the similar design and training recipe as Transformers, ConvNets can behave even better than some popular ViTs [14], [15]. RepLKNet [21] also shows the potential of leveraging large-kernel convolutions for visual recognition. These explorations encourage many researchers to rethink the design of ConvNets by leveraging either large-kernel convolutions [22], [23], or high-order spatial interactions [24], or sparse convolutional kernels [25], etc. Till now, how to more efficiently take advantage of convolutions to construct powerful ConvNet architectures is still a hot research topic in computer vision.

In this paper, we are also interested in investigating new ways to make better use of spatial convolutions. Different

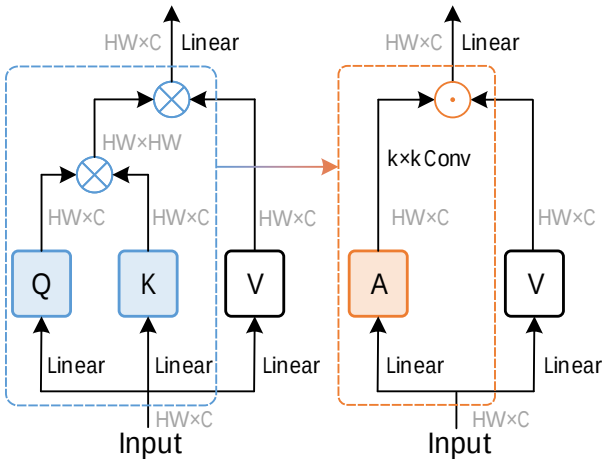

Fig. 1. Comparison between the self-attention mechanism and the proposed convolutional modulation operation. As can be seen, instead of generating attention matrices via a matrix multiplication between the query and key, we directly produce weights using a $k \times k$ depthwise convolution to reweigh the value via the Hadamard product ($\odot$: Hadamard product; $\otimes$: matrix multiplication).

- *Q. Hou, C.Z. Lu, and M.M. Cheng are with School of Computer Science, Nankai University, Tianjin, China. (andrewhoux@gmail.com, cmm@nankai.edu.cn) Ming-Ming Cheng is the corresponding author.*
- *J. Feng is with ByteDance, Singapore.*

- *This research was supported by NSFC (NO. 62225604, No. 62276145), the Fundamental Research Funds for the Central Universities (Nankai University, 070-63223049), CAST through Young Elite Scientist Sponsorship Program (No. YESS20210377). Computations were supported by the Supercomputing Center of Nankai University (NKSC).*

from the ConvNeXt work [20] that aims to adjust the training recipe or the position of spatial convolutions in building blocks, we compare the different ways ViTs and ConvNets use to encode spatial information. As shown in the left part of Fig. 1, self-attention computes the output of each pixel by a weighted summation of all other positions. This process can also be mimicked by computing the Hadamard product between the output of a large-kernel convolution and the value representations, which we call convolutional modulation as depicted in the right part of Fig. 1. The difference is that the convolutional kernels are static while the attention matrix generated by self-attention can adapt to the input content. Our experiments show that using convolutions to generate weight matrix yields great results as well.

Simply replacing the self-attention in ViTs with the proposed convolutional modulation operation yields the proposed network, termed Conv2Former. The meaning behind it is that we aim to use convolutions to construct a Transformer-style ConvNet, in which the convolutional features are used as weights to modulate the value representations. In contrast to the classic ViTs with self-attention, our method, like many classic ConvNets, is fully convolutional and hence its computations increase linearly rather than quadratically as in Transformers with the image resolution being higher. This makes our method more friendly to downstream tasks, like object detection and high resolution semantic segmentation.

Another main contribution of this paper is that we show Conv2Former can benefit more from convolutions with larger kernels, like 11×11 and 21×21. This is different from the conclusions made in previous ConvNets [20], [26], which demonstrate using standard depthwise convolutions with kernel sizes larger than 9×9 brings nearly no performance gain but computational burden. Our experiments show that a consistent performance improvement can be obtained when gradually increasing the convolutional kernel size from 5×5 to 21×21. We also show that our method using 11×11 depthwise convolutions performs even better than the recent works using super large kernel convolutions [21], [25] (e.g., 31×31), reflecting the effectiveness of our proposed spatial encoding method.

We evaluate Conv2Former on popular vision tasks, including ImageNet classification [27], COCO object detection/instance segmentation [28], and ADE20k semantic segmentation [29]. To validate the capability of Conv2Former on larger datasets, we also pretrain our model on the ImageNet-22k dataset and evaluate the performance on downstream tasks. Experiments show that Conv2Former performs better than popular ConvNets, like ConvNeXt [20] and EfficientNetV2 [18]. We hope our work could provide informative design choices for future visual recognition models.

2 Related Work

2.1 Convolutional Neural Networks

The success of early visual recognition models is mostly dedicated to the development of ConvNets, typified by VGGNet [1] and GoogLeNet [2]. These models, suffering from the gradient vanishing problem, mostly contain less than 20-layer convolutions. Later, the emerging of ResNets [5] advances the conventional ConvNets by introducing shortcut connections, which make training very deep models possible. Inceptions [3], [4] and ResNeXt [6] further enrich the design principles of ConvNets and propose to use building blocks with multiple parallel paths of specialized-filter convolutions. Instead of tuning network architectures, SENet [9] and its follow-ups [10], [11] aim to improve ConvNets with lightweight attention modules that can explicitly model the inter-dependencies among channels. EfficientNets [17], [18] and MobileNetV3 [30] take advantage of neural architecture search [31] to search for efficient network architectures. Very recently, some works aim to show the advantages of introducing large-kernel convolutions [20], [21], [22], [24], [25]. A typical example should be VAN [22] that utilizes a standard depthwise convolution and a dilated one to decompose large-kernel convolutions. HorNet [24] further advances VAN by explicitly building high-order spatial interactions based on recursive gated convolutions. Our Conv2Former is different from VAN and HorNet in that we do not aim to decompose large-kernel convolutions but show self-attention can be reduced to the convolutional modulation operation, which results in good recognition performance as well. Our work is also related to DWNet [32] that also attempts to connect local self-attention and depth-wise convolutions. Different from DWNet, our Conv2Former aims to produce attention weights with depthwise convolution to reweigh the value via the Hadamard product but DWNet replaces the whole local self-attention with depthwise convolution. In addition, there are also some works leveraging different training or optimization methods or finetuning techniques [33], [34], [35] to advance EfficientNet.

2.2 Vision Transformers

Transformers, originally designed for natural language processing tasks [36], have been widely used in visual recognition. The most typical work should be Vision Transformer (ViT) [12] which shows the great potential of Transformers for processing large-scale data in image classification. DeiT [13] improves the original ViT by using strong data augmentation methods and knowledge distillation and gets rid of the dependence of ViTs on large-scale data. Motivated by the success of pyramid architecture in ConvNets, some works [14], [15], [37], [38] design pyramid structures using Transformers to take advantage of multi-scale features. Some works [39], [40], [41], [42], [43], [44] propose to introduce local dependencies into ViTs, showing great performance in visual recognition. Besides, there are also some works [16], [45], [46], [47], [48] exploring the scaling capability of ViTs in visual recognition. Specially, Yuan et al. [16] show that a two-stage ViT outperforms the state-of-the-art CNNs on ImageNet for the first time.

2.3 Other Models

Some recent works show that mixing both Transformers and convolutions [19], [43], [49] is a promising way to develop stronger visual recognition models especially for those aiming at efficient network design. A typical example should be MobileViT [50], which provides an efficient way to fuse

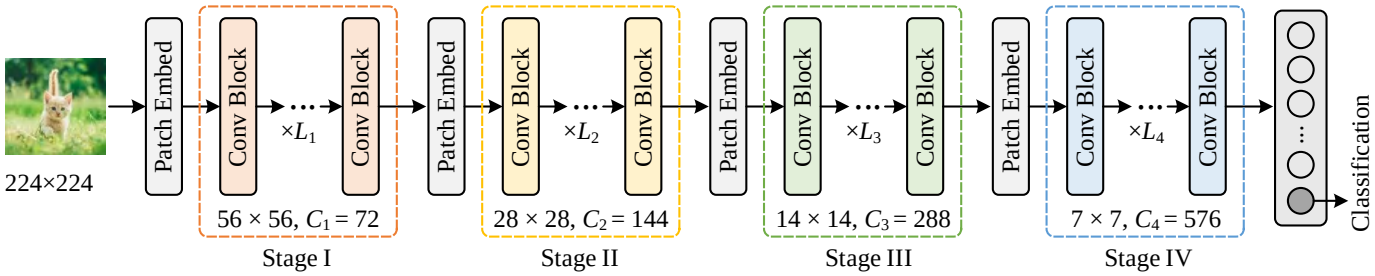

Fig. 2. Overall architecture of Conv2Former. Like most previous ConvNets and Swin Transformer, we adopt a pyramid structure with four stages. In each stage, different numbers of convolutional blocks are used. This figure shows the setting of the proposed Conv2Former-T, where $\{L_1, L_2, L_3, L_4\} = \{3, 3, 12, 3\}$.

both convolutions and Transformers. EfficientViT [51], EdgeNeXt [52], and MobileFormer [53] bring back convolutions to Transformers and show great performance in both image classification and downstream tasks. Moreover, there are also hybrid networks that introduce different attention mechanisms into ConvNet for global context encoding [54], [55], [56], [57], [58]. In addition, designing MLP-like architectures is also a popular research topic for visual recognition [59], [60], [61].

Our method is also related some recent works [62], [63], [64], [65] that aim to improve the spatial encoding ability or efficiency of CNNs or ViTs. RIFormer [62] introduces the re-parameterizing idea to reduce the token mixing operations in ViTs to improve inference efficiency. LITv2 [64] encodes part of the spatial information with self-attention at a lower resolution to increase running speed. Hu et al. [63] experimentally studied some typical spatial token mixers and analyzed their performance on multiple vision tasks. SMT [65] is a concurrent work to our method that demonstrates mixing convolutions with different kernel sizes helps visual recognition.

3 Model Design

In this section, we describe the architecture of our proposed Conv2Former and provide some useful suggestions in model design and layer adjustment.

3.1 Architecture

Overall architecture. The overall architecture has been shown in Fig. 2. Similarly to the ConvNeXt [20] and Swin Transformer network [14], our Conv2Former also adopts a pyramid architecture. There are four stages in total, each of which has a different feature map resolution. Between two consecutive stages, a patch embedding block is used to reduce the resolution, which is often a 2×2 convolution with stride 2. Different stages have different numbers of convolutional blocks. We build five Conv2Former variants, namely Conv2Former-N, Conv2Former-T, Conv2Former-S, Conv2Former-B, Conv2Former-L. Details are summarized in Tab. 1.

Stage configuration. When the number of learnable parameters is fixed, how to arrange the width and depth of the network has an impact on the model performance [17], [35]. The original ResNet-50 sets the number of blocks in each stage to $(3, 4, 6, 3)$. ConvNeXt-T changes the block numbers to $(3, 3, 9, 3)$ following the principle used in Swin-T and uses the stage compute ratio of $1 : 1 : 9 : 1$ for larger models. Differently, we slightly adjust the ratios as shown in Tab. 1. We observe that for a tiny-sized model (with less than 30M parameters) deeper networks perform better. A brief comparison among four different tiny-sized models can be found in Tab. 2.

TABLE 1
Brief configurations of the proposed Conv2Former. We implement 5 variants with numbers of parameters 15M, 27M, 50M, 90M, and 199M, respectively.

Model	$\{C_1, C_2, C_3, C_4\}$	$\{L_1, L_2, L_3, L_4\}$
★ Conv2Former-N	$\{64, 128, 256, 512\}$	$\{2, 2, 8, 2\}$
★ Conv2Former-T	$\{72, 144, 288, 576\}$	$\{3, 3, 12, 3\}$
★ Conv2Former-S	$\{72, 144, 288, 576\}$	$\{4, 4, 32, 4\}$
★ Conv2Former-B	$\{96, 192, 384, 768\}$	$\{4, 4, 34, 4\}$
★ Conv2Former-L	$\{128, 256, 512, 1024\}$	$\{4, 4, 48, 4\}$

TABLE 2
Stage comparison with three popular models. Slightly adjusting the number of convolutional blocks as shown in the last row improves the performance.

Model	Params.	FLOPs	Stage Conf.	Top-1 Acc.
ResNet-50 [5]	26M	4.0G	3-4-6-3	78.5%
Swin-T [14]	28M	4.5G	2-2-6-2	81.5%
ConvNeXt-T [20]	29M	4.5G	3-3-9-3	82.1%
★ Conv2Former-N	15M	2.2G	2-2-8-2	81.5%
★ Conv2Former-T	28M	4.4G	3-3-8-3	82.8%
★ Conv2Former-T	27M	4.4G	3-3-12-3	83.2%

3.2 Convolutional Modulation Block

Our convolutional block used in each stage shares a similar structure to Transformers, which mainly contains a self-attention layer for spatial encoding and an MLP for channel mixing. Differently, we replace the self-attention layer with a simple convolutional modulation layer.

Self-attention. For an input token sequence $\mathbf{X}$ of length N, self-attention first generates the key $\mathbf{K}$, query $\mathbf{Q}$, and value $\mathbf{V}$ using linear layers, where $\mathbf{X}, \mathbf{K}, \mathbf{Q}, \mathbf{V} \in \mathbb{R}^{N \times C}$, $N = H \times W$, C is the channel number, H and W are the spatial size of the input. The output is the weighted average of the

value based on a similarity score $\mathbf{A}$,

$$\mathrm{Attention}(\mathbf{X}) = \mathbf{AV}, \tag{1}$$

where $\mathbf{A}$ measures the relationships between each pair of input tokens, which can be written as

$$\mathbf{A} = \mathrm{Softmax}(\mathbf{QK}^\top). \tag{2}$$

Note that we omit the scaling factor for simplicity. In spite of the high efficiency in encoding spatial information, the similarity score matrix $\mathbf{A}$ has a shape of $\mathbb{R}^{N\times N}$, making the computational complexity of self-attention grows quadratically as the sequence length N increases.

Convolutional modulation. In our convolutional modulation layer, instead of calculating the similarity score matrix $\mathbf{A}$ via Eqn. 2, we simplify self-attention by modulating the value $\mathbf{V}$ with convolutional features. Specifically, given the input tokens $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, we use a simple depthwise convolution with kernel size $k \times k$ and the Hadamard product to calculate the output $\mathbf{Z}$ as follows:

$$\mathbf{Z} = \mathbf{A} \odot \mathbf{V}, \tag{3}$$

$$\mathbf{A} = \mathrm{DConv}_{k\times k}(\mathbf{W}_1\mathbf{X}), \tag{4}$$

$$\mathbf{V} = \mathbf{W}_2\mathbf{X}, \tag{5}$$

where $\odot$ is the Hadamard product, $\mathbf{W}_1$ and $\mathbf{W}_2$ are weight matrices of two linear layers, and $\mathrm{DConv}_{k\times k}$ denotes a depthwise convolution with kernel size $k \times k$. The above convolutional modulation operation enables each spatial location (h, w) to be correlated with all the pixels within the $k \times k$ square region centered at (h, w). The information interaction among channels can be achieved by the linear layers. The output for each spatial location is the weighted sum of all the pixels within the square region.

Advantages. Compared to self-attention, our method utilizes convolutions to build relationships, which are more memory-efficient than self-attention especially when processing high-resolution images. Compared to the classic residual blocks [5], [20], our method can also adapt to the input content due to the modulation operation.

3.3 Micro Design

Larger kernel than 7×7. How to make use of spatial convolutions is important for ConvNet design. Since VGGNet [1] and ResNets [5], [6], 3×3 convolutions have been a standard choice for building ConvNets. Later, the emerging of depthwise separable convolution [66] changes this situation. ConvNeXt shows that enlarging the kernel size of ConvNets from 3 to 7 can improve the classification performance. However, further increasing the kernel size nearly brings no performance gain but computational burden without reparameterization [21], [67].

We argue that the reason making ConvNeXt benefit little from larger kernel sizes than 7×7 is the way to use spatial convolutions. For Conv2Former, we observe a consistent performance gain as the kernel size increases from 5×5 to 21×21. This phenomenon not only happens for Conv2Former-T ($82.8 \rightarrow 83.4$) but also holds for Conv2Former-B with 80M+ parameters ($84.1 \rightarrow 84.5$). Considering the model efficiency, we set the kernel size to 11×11 by default.

TABLE 3
Top-1 accuracy result comparison on ImageNet [27]. Compared to previous popular Transformers and ConvNets, our Conv2Former achieves a surprisingly good results for network variants with different model sizes.

Model	#Params	FLOPs	Image Size	Top-1 Acc.
VAN-B1 [22]	14M	2.5G	224×224	81.1%
★ Conv2Former-N	15M	2.2G	224×224	81.5%
ResNet50-d [5], [68]	26M	4.3G	224×224	79.5%
SwinT-T [14]	28M	4.5G	224×224	81.5%
DWNet-T [32]	26M	4.4G	224×224	81.8%
ConvNeXt-T [20]	29M	4.5G	224×224	82.1%
VAN-B2 [22]	27M	5.0G	224×224	82.8%
★ Conv2Former-T	27M	4.4G	224×224	83.2%
SwinT-S [14]	50M	8.7G	224×224	83.0%
ConvNeXt-S [20]	50M	8.7G	224×224	83.1%
VAN-B3 [22]	45M	9.0G	224×224	83.9%
NFNet-F0 [35]	72M	12.4G	256×256	83.6%
★ Conv2Former-S	50M	8.7G	224×224	84.1%
DeiT-B [13]	86M	17.5G	224×224	81.8%
DWNet-B [32]	80M	14.3G	224×224	83.4%
RepLKNet-31B [21]	79M	15.3G	224×224	83.5%
SwinT-B [14]	88M	15.4G	224×224	83.5%
ConvNeXt-B [20]	89M	15.4G	224×224	83.8%
FocalNet-B [69]	89M	15.4G	224×224	83.9%
SLak-B [25]	95M	17.1G	224×224	84.0%
MOAT-2 [70]	73M	17.2G	224×224	84.2%
EffNet-B7 [51]	66M	37.0G	600×600	84.3%
★ Conv2Former-B	90M	15.9G	224×224	84.4%

Weighting strategy. As shown in Fig. 1, we consider the outputs of depthwise convolutions as weights to modulate the features after the linear projection. It is worth noting that we use neither activation nor normalization layers (e.g., Sigmoid or L_p normalization) before the Hadamard product. This is an essential factor to attain good performance. For example, adding a Sigmoid function as done in SENet [9] decreases the performance by more than 0.5%.

We want to stress that FocalNet [69] adopts a similar weighting strategy as ours but its motivation is different. FocalNet aims to extract multi-level features via 3×3 depthwise convolutions and global average pooling for hierarchical context aggregation. Differently, we attempt to simplify the self-attention operation by leveraging simple large kernel convolutions and investigate an efficient way to make use of large kernel convolutions for ConvNets. Our method is much simpler than FocalNet and experiments demonstrate the advantages of Conv2Former over FocalNet.

Normalization and activations. For normalization layers, we follow the original ViT and ConvNeXt and adopt the Layer Normalization [71] instead of the widely-used batch normalization [72]. For activation layers, we use GELU [73]. We found that the combination of Layer Normalization and GELU brings 0.1%-0.2% performance gain.

TABLE 4
Top-1 accuracy results on ImageNet [27] with pretraining on the ImageNet-22k dataset. We can observe consistent improvement compared to ConvNeXt. Our Conv2Former-L also performs better than EfficientNetV2-XL and CoAtNet-3.

Model	#Params	FLOPs	Image Size	Top-1 Acc.
ConvNeXt-S [20]	50M	8.7G	224×224	84.6%
★ Conv2Former-S	50M	8.7G	224×224	84.9%
SwinT-B [14]	88M	15.4G	224×224	85.2%
ConvNeXt-B [20]	89M	15.4G	224×224	85.8%
★ Conv2Former-B	90M	15.9G	224×224	86.2%
SwinT-B [14]	88M	47.0G	384×384	86.4%
ConvNeXt-B [20]	89M	45.1G	384×384	86.8%
★ Conv2Former-B	90M	46.7G	384×384	87.0%
EffNet-V2-XL [18]	208M	94.0G	480×480	87.3%
SwinT-L [14]	197M	34.5G	224×224	86.3%
ConvNeXt-L [20]	198M	34.4G	224×224	86.6%
★ Conv2Former-L	199M	36.0G	224×224	87.0%
SwinT-L [14]	197M	104G	384×384	87.3%
ConvNeXt-L [20]	198M	101G	384×384	87.5%
CoAtNet-3 [43]	168M	107G	384×384	87.6%
★ Conv2Former-L	199M	105.9G	384×384	87.7%

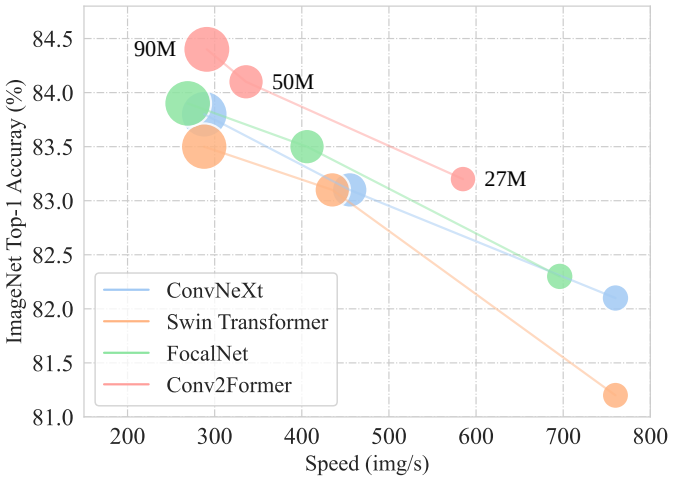

Fig. 3. ImageNet classification accuracy v.s. inference speed. We can see that our proposed Conv2Former achieves the best trade-off. The inference speed is tested on an NVIDIA V100 GPU.

4 Experiments

4.1 Experiment Setup

Datasets. We evaluate the classification performance of the proposed Conv2Former on the widely-used ImageNet-1k dataset [27], which contains around 1.2M training images and 1,000 different categories. We report the results on the validation set that has in total 50k images. Like some other popular models [14], [20], we also test the scaling ability of the proposed Conv2Former using the large-scale ImageNet-22k dataset for pretraining, which has around 14M images and 21,841 classes. After pretraining, we use the ImageNet-1k dataset for finetuning and report results on the ImageNet-1k validation set as well.

Training settings. We implement our model based on `PyTorch` [74]. During training, we use the AdamW optimizer [75] with a linear learning rate scaling strategy $lr = \text{LR}_{\text{base}} \times \text{batch\_size}/1024$. The initial learning rate LR_{base} is set to 0.001 and weight decay rate is set to 5×10^{-2} as suggested in previous work [20]. Throughout the experiments on ImageNet, we randomly crop the image size to 224×224 and adopt some common data augmentation methods, such as MixUp [76] and CutMix [77]. Stochastic Depth [78], Random Erasing [79], Label Smoothing [3], RandAug [80], and Layer Scale [46] of initial value 1e-6 are used as well. We train all the models for 300 epochs. For experiments on the ImageNet-22k, we first pretrain our model on this dataset for 90 epochs and then finetuning on ImageNet-1k for 30 epochs following ConvNeXt [20].

4.2 Comparison with Other Methods

We compare our Conv2Former with some popular network architectures, including Swin Transformer [14], ResNet [5], ConvNeXt [20], NFNet [35], DeiT [13], DWNet [32], FocalNet [69], VAN [22], SLak [25], EfficientNets [17], [18], CoAtNet [43], RepLKNet [21], and MOAT [70]. Note that some of them are hybrid models of CNNs and Transformers.

ImageNet-1k. We first train our Conv2Former on the ImageNet-1k dataset and show the results in Tab. 3. For tiny-sized models ($<$ 30M), our Conv2Former has 1.1% and 1.7% performance gains compared to ConvNeXt-T and SwinT-T, respectively. Even our Conv2Former-N with 15M parameters and 2.2G FLOPs performs the same as SwinT-T with 28M parameters and 4.5G FLOPs. For the base models, the performance gain decreases but there are still 0.6% and 0.9% improvement over ConvNeXt-B and SwinT-B. Compared to other popular models, our Conv2Former also perform better than those with similar model sizes. In particular, our Conv2Former receives better results than DWNet [32] which also attempts to connect the local self-attention and depthwise convolution, for both tiny- and base-sized models. Notably, our Conv2Former-B even behaves better than EfficientNet-B7 (84.4% v.s. 84.3%), whose computations are two times larger than ours (37G v.s. 15G).

ImageNet-22k. We pretrain our Conv2Former on the large ImageNet-22k dataset and then finetune on the ImageNet-1k dataset. This experiment can reflect the data scaling capability of our Conv2Former. For all experiments, we follow the settings used in [20] to train and finetune the models. The results have been listed in Tab. 4. Compared to the different variants of ConvNeXt, our Conv2Formers all perform better when the model sizes are similar. In addition, we can see that when finetuning on a larger resolution 384×384 our Conv2Former-L attains better result than hybrid models, like CoAtNet. Our Conv2Former-L achieves the best result 87.7%.

Timing. Here, we compare our Conv2Former with three popular models, including ConvNeXt, Swin Transformer, and FocalNet, in terms of inference speed. We show the accuracy-speed curves of all four methods in Fig. 3. We can see that our Conv2Former achieves the best trade-off between classification accuracy and inference speed on ImageNet.

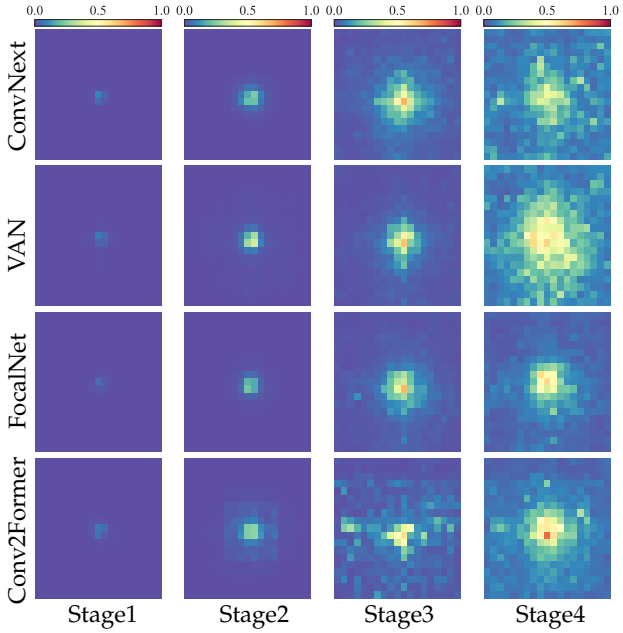

Fig. 4. Effective receptive field visualization of 4 stages for four different CNN-based methods.

TABLE 5
Statistical analysis of the effective receptive fields for four CNN-based methods. 'Mean' indicates the average results under different thresholds from 0.5 to 0.9 with step 0.05. For both 'Side Length' and 'Area Ratio,' our Conv2Former achieves the best results.

Model	Side Length			Area Ratio (%)		
	0.5	0.75	Mean	0.5	0.75	Mean
ConvNeXt-T	70.0	102.5	96.6	14.8	30.4	27.9
VAN-B1	62.0	115.5	104.8	10.9	31.5	28.5
FocalNet-T	59.0	100.0	94.2	10.0	27.0	25.2
Conv2Former-T	75.0	113.5	**106.7**	15.4	33.4	**30.5**

Effective receptive field analysis. To demonstrate why the proposed Conv2Former works better than recent state-of-the-art CNN-based methods, we analyze the effective receptive fields (ERFs) produced by different method as shown in Tab. 5. We use two thresholds 0.5 and 0.75 to compute the side length and area ratio, respectively. In addition, we also calculate the mean results under different thresholds from 0.5 to 0.9 with a step of 0.05. We can see that our Conv2Former performs better than FocalNet, VAN, and ConvNeXt in terms of side length. For area ratio, our Conv2Former yields the best results.

As shown in Fig. 4, we also attempt to visualize the ERFs of different stages for four models. It can be clearly seen that for Stages 2 and 3, our Conv2Former can also have larger ERFs than other three methods. We argue that this is mainly because the proposed convolutional modulation block can encode the spatial information in a more appropriate way. It enables even the middle layers of our Conv2Former to be able to capture large ERFs. Therefore, our Conv2Former with a simple network architecture can produce state-of-the-art results.

Advantages over other large-kernel methods. Employing large-kernel convolutions is a straightforward way to assist CNNs in building long-range relationships. However, directly using large-kernel convolutions ($> 7 \times 7$) in existing CNN-based architectures makes the recognition models difficult to optimize [20], [26]. Recently, there are a few works aiming to develop new techniques to evoke the utilization of large-kernel convolutions in CNNs. In Tab. 6, we show the results by the recent state-of-the-art ConvNets with different kernel sizes. We can see that without any other training techniques, like re-parameterization or using sparse weights, our Conv2Former with kernel size 7×7 already performs better than other methods under the base model setting. Using a larger kernel size 11×11 yields a better performance gain. These results reflect the advantage of our convolutional modulation block.

TABLE 6
Comparison with the recent state-of-the-art ConvNets with different kernel sizes. We can see that without any other training techniques, like re-parameterization or using sparse weights, our Conv2Former with kernel size 11×11 achieves the best result. These experiments indicate that our convolutional modulation operation can more efficiently encode the spatial information.

Model	Kernel size	#Params	FLOPs	Acc.
RepLKNet-31B [21]	31×31	79M	15.3G	83.5%
ConvNeXt-B [20]	7×7	89M	15.4G	83.8%
SLaK-B [25]	51×51	95M	17.1G	84.0%
★ Conv2Former-B	7×7	89M	15.6G	84.2%
★ Conv2Former-B	11×11	90M	15.9G	84.4%

TABLE 7
Performance comparison when different fusion strategies are used in our convolutional modulation block. All results are based on Conv2Former-T. We can see that using the simple Hadamard product yields the best result.

Weighting Strategy	Top-1 Acc.
Element-wise sum	82.7%
Adding a Sigmoid function after $\mathbf{A}$	82.3%
Adding an L_1 normalization after $\mathbf{A}$	82.8%
Linearly normalizing the values of $\mathbf{A}$ to $(0, 1]$	82.2%
★ Hadamard product	83.2%

4.3 Method Analysis

In this subsection, we provide a series of method analysis on the proposed convolution modulation operation.

Kernel size. The ConvNeXt work [20] shows that there is no performance gain when the kernel size of depthwise convolutions is more than 7×7. Here, we investigate how would the model performance change when larger kernel sizes are used. We select 6 different kernels for the depthwise convolutions, *i.e.*, $\{5 \times 5, 7 \times 7, 9 \times 9, 11 \times 11, 15 \times 15, 21 \times 21\}$ and show the results based on two model variants, Conv2Former-T and Conv2Former-B. The results can be found in Fig. 5(a). The performance gain seems to saturates until the kernel size is increased to 21×21. This result is quite different from that made by ConvNeXt who concludes that using larger than 7×7 kernels brings no clear performance gain. This indicates that using the convolutional features as weights as formulated in Eqn. 3

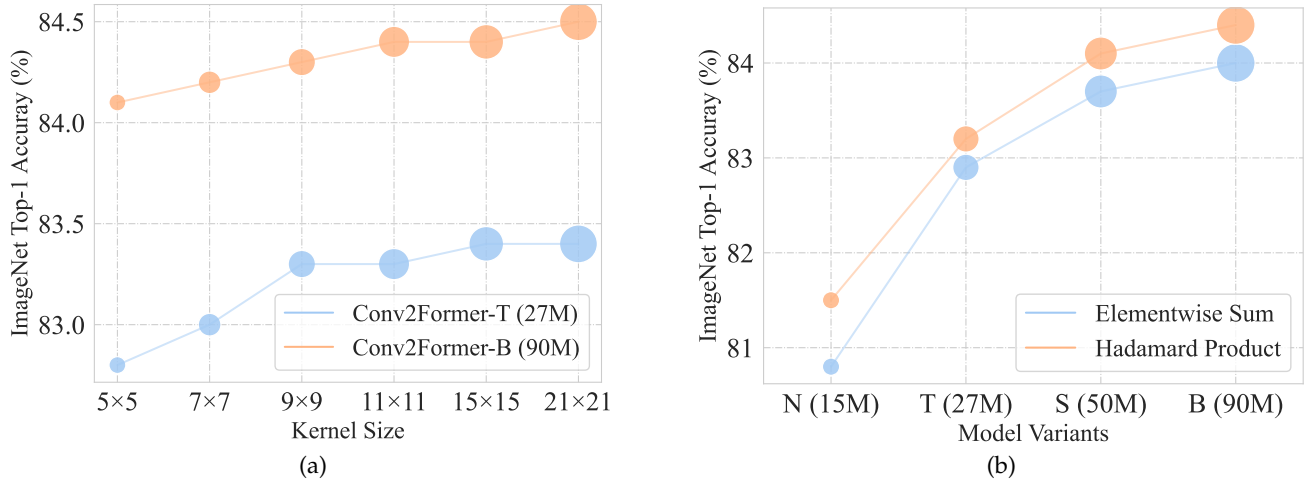

Fig. 5. Ablative experiments. For both Conv2Former-T and Conv2Former-B, we can observe a consistent performance improvement when increasing the kernel size from 5×5 to 21×21. When replacing the Hadamard product with the element-wise summation operation, the performance drops for all four variants of our Conv2Former.

can more efficiently take advantage of large kernels than traditional ways [5], [20].

Hadamard product is better than summation. As shown in Fig. 1, we use the convolutional features extracted by the depthwise convolutions to modulate the weights of the right linear branch via the Hadamard product operation. In our experiments, we have also attempt to leverage the element-wise summation to fuse the two branches. Fig. 5(b) shows the comparison results on our Conv2Former at different model sizes. The Hadamard product performs better than element-wise summation, indicating convolutional modulation is more efficient than summation in encoding spatial information. We can also observe that small models benefit more from Hadamard product.

Weighting strategy. Other than the aforementioned two fusion strategies, we also attempt to use other ways to fuse the feature maps, including adding a Sigmoid function after $\mathbf{A}$, applying L_1 normalization to $\mathbf{A}$, and linearly normalizing the values of $\mathbf{A}$ to $(0, 1]$. The results are summarized in Tab. 7. We can see that the Hadamard product leads to better results than all other operations. More interestingly, when adjusting the values of $\mathbf{A}$ to positive values using either the Sigmoid function or linear normalization to $(0, 1]$, the performance drops more. This is different from the traditional attention mechanisms, like SE [9] and CA [11] that leverage the Sigmoid function before reweighing. We leave this for future research.

4.4 Visual Analysis

Feature visualizations. To further demonstrate the effectiveness of the proposed method over the recent state-of-the-art models, like FocalNet and ConvNeXt, we use Grad-CAM [81] to visualize the feature maps produced by different models. The visual results are shown in Fig. 6. We can see that compared to other three models, our Conv2Former can more accurately locate the target objects. In particular, for objects with elongated shapes (See the top two rows), our Conv2Former can also precisely capture them. This enables our Conv2Former to recognize better than other models.

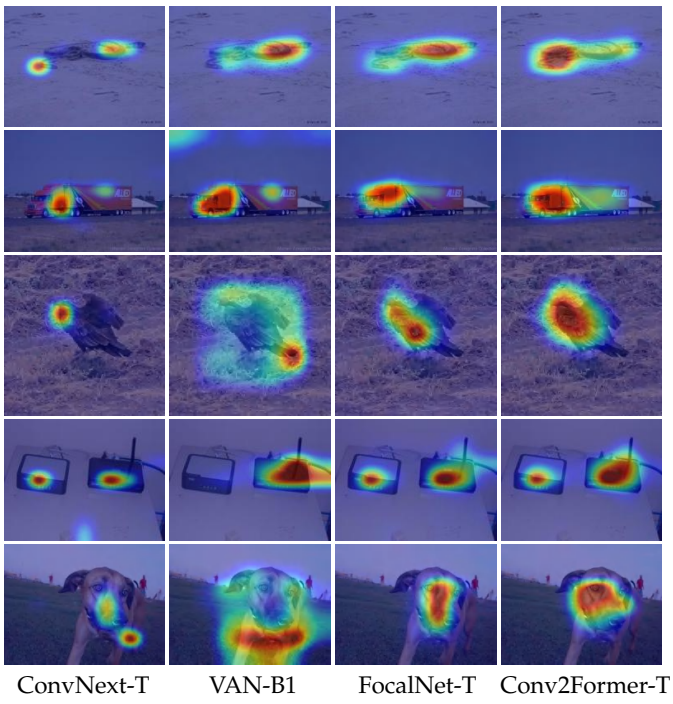

Fig. 6. Feature visualizations of four methods. We adopt the widely-used Grad-CAM [81] as our visualization tool. We can see that our method can more accurately locate the target objects than other three CNN-based methods.

4.5 Results on Isotropic Models to ViTs

Different from the classic CNNs that adopt hierarchical architectures, the vanilla ViT [12], [13] due to the heavy self-attention layer utilizes a plain architecture that contains a patch embedding layer and a stack of Transformers with the same sequence length. This plain architecture has been widely used in recent works on Transformers. Here, we follow ConvNeXt [20] and also attempt to investigate the performance of Conv2Former under the ViT-style architecture settings. Similar to ConvNeXt, we set the number of blocks to 18 for both Conv2Former-IS and Conv2Former-IB and adjust the channel numbers to match the model size. We

TABLE 8
Comparisons among our isotropic Conv2Former, ConvNeXt, and ViT. '3 Convs' means that we use three convolutional layers for patch embedding at the beginning of the network as done in [43], [45], [48]. For both the small-sized and base-sized models, our Conv2Former achieves better results with comparable parameters and computations to other methods.

Model	Patch Embed	#Params	FLOPs	Top-1 Acc.
DeiT-S	1 Conv	22M	4.6G	79.8%
ConvNeXt-IS	1 Conv	22M	4.3G	79.7%
★ Conv2Former-IS	1 Conv	23M	4.3G	81.2%
★ Conv2Former-IS	3 Convs	23M	4.5G	82.0%
DeiT-B	1 Conv	87M	17.6G	81.8%
ConvNeXt-IB	1 Conv	87M	16.9G	82.0%
★ Conv2Former-IB	1 Conv	86M	16.5G	82.7%
★ Conv2Former-IB	3 Convs	87M	17.3G	83.0%

use two versions of the patch embedding module: a 16×16 convolution with stride 16 and three convolutions as done in [45].

Tab. 8 shows the results. We take the DeiT-S and DeiT-B model as baselines. For brevity, we add a letter 'I' in the model names, representing that the corresponding models use the isotropic architecture as the original ViT. We can see that for small-sized models with around 22M parameters, our Conv2Former-IS performs much better than DeiT-S and ConvNeXt-IS. The performance gain is around 1.5%. When scaling up the model size to 80M+, our Conv2Former-IB achieves a top-1 accuracy score of 82.7%, which is also 0.7% better than ConvNeXt-IB and 0.9% better than DeiT-B. In addition, using three convolutions for patch embedding can further improve the result.

4.6 Results on Downstream Tasks

In this subsection, we evaluate our method on two downstream tasks, including object detection on COCO [28] and semantic segmentation ADE20k [29].

Results on COCO. Following previous works [14], [20], we conduct experiments using two popular object detectors, Mask R-CNN [82] and Cascade Mask R-CNN [83] and report both the object detection and instance segmentation results. For training, we follow the experiment settings used in ConvNeXt [20], including multi-scale training, AdamW optimizer with a $3\times$ learning schedule, GIoU loss [84], etc. Readers can refer to [20], [85] for more detailed experimental settings. We use the MMDetection toolbox [86] to run all the object detection experiments.

The results can be found in Tab. 9. For tiny-sized models, our Conv2Former-T achieves about 2% AP improvement over SwinT-T and ConvNeXt-T when using the Mask R-CNN framework in object detection. For instance segmentation, the performance gain is also more than 1%. When using the Cascade Mask R-CNN framework, we can observe more than 1% performance gain than SwinT-T and ConvNeXt-T. When scaling up the models, the improvement is also clear.

Results on ADE20k. Following [14], [20], we train the models using the training set and report results on the val set. For tiny-, small-, base-sized models, we randomly crop the image to 512×512, and for the large-sized model, we crop the image to 640×640. We use the UperNet [87] as our decoder.

TABLE 9
COCO [28] object detection and instance segmentation results using Mask R-CNN [82] and Cascade Mask R-CNN [83]. We use ImageNet-1k pretrained backbones.

Model	FLOPs	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Mask R-CNN [82] 3× schedule							
SwinT-T	267G	46.0	68.1	50.3	41.6	65.1	44.9
ConvNeXt-T	262G	46.2	67.9	50.8	41.7	65.0	44.9
★ Conv2Former-T	255G	48.0	69.5	52.7	43.0	66.8	46.1
Cascade Mask R-CNN [83] 3× schedule							
SwinT-T	743G	50.4	69.2	54.7	43.7	66.6	47.3
ConvNeXt-T	741G	50.4	69.1	54.8	43.7	66.5	47.3
SLaK-T	841G	51.3	70.0	55.7	44.3	67.2	48.1
★ Conv2Former-T	734G	51.4	69.8	55.9	44.5	67.4	48.3
SwinT-S	833G	51.9	70.7	56.3	45.0	68.2	48.8
ConvNeXt-S	827G	51.9	70.8	56.5	45.0	68.4	49.1
★ Conv2Former-S	823G	52.8	71.4	57.3	45.7	69.0	49.8
SwinT-B	975G	51.9	70.5	56.4	45.0	68.1	48.9
ConvNeXt-B	964G	52.7	71.3	57.5	45.6	69.0	49.8
★ Conv2Former-B	968G	52.8	71.1	57.2	45.6	68.7	49.3

TABLE 10
Comparisons with Swin-T and ConvNeXt on ADE20k [29]. We use UperNet [87] as the decoder. At all model sizes, our Conv2Former achieves the best results.

Model	Crop Size	#Params	FLOPs	mIoU (%)
ImageNet-1K pre-trained				
SwinT-T	512^2	59M	946G	45.8
ConvNeXt-T	512^2	59M	940G	46.7
★ Conv2Former-T	512^2	55M	931G	48.0
SwinT-S	512^2	80M	1039G	49.5
ConvNeXt-S	512^2	81M	1024G	49.6
★ Conv2Former-S	512^2	78M	1021G	50.3
SwinT-B	512^2	120M	1189G	49.7
ConvNeXt-B	512^2	121M	1166G	49.9
★ Conv2Former-B	512^2	119M	1171G	51.0
ImageNet-22K pre-trained				
SwinT-L	640^2	232M	2479G	53.5
ConvNeXt-L	640^2	233M	2453G	53.7
★ Conv2Former-L	640^2	230M	2483G	54.3

Results are summarized in Tab. 10. For models at different scales, our Conv2Former can outperform both the Swin Transformer and ConvNeXt. Notably, there is a 1.3% mIoU improvement compared to ConvNeXt at the tiny scale and the improvement is 1.1% at the base scale. When we further increase the model size, our Conv2Former-L with UperNet achieves an mIoU score of 54.3%, which is also clearly better than Swin-L and ConvNeXt-L.

5 Conclusions and Discussions

This paper present Conv2Former, a new convolutional network architecture for visual recognition. The core of our Conv2Former is the convolutional modulation operation that simplifies the self-attention mechanism by using only convolutions and Hadamard product. We show that our

convolutional modulation operation is a more efficient way to take advantage of large-kernel convolutions. Our experiments in ImageNet classification, object detection, and semantic segmentation also show that our proposed Conv2Former performs better than previous CNN-based models and most of the Transformer-based models. We believe there is still a large room to improve the performance of ConvNets and we hope our method could provide insights for future research on ConvNets.

REFERENCES

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1–9.

[3] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2818–2826.

[4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *AAAI Conf. on Artif. Intel.*, 2017.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[6] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1492–1500.

[7] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, 2021.

[8] D. Zhou, Q. Hou, Y. Chen, J. Feng, and S. Yan, "Rethinking bottleneck structure for efficient mobile network design," in *Eur. Conf. Comput. Vis.* Springer, 2020, pp. 680–697.

[9] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7132–7141.

[10] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[11] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 13 713–13 722.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Int. Conf. Learn. Represent.*, 2020.

[13] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *arXiv preprint arXiv:2012.12877*, 2020.

[14] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Int. Conf. Comput. Vis.*, 2021.

[15] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Int. Conf. Comput. Vis.*, 2021.

[16] L. Yuan, Q. Hou, Z. Jiang, J. Feng, and S. Yan, "Volo: Vision outlooker for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.

[17] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 6105–6114.

[18] ——, "Efficientnetv2: Smaller models and faster training," in *International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 10 096–10 106.

[19] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 16 519–16 529.

[20] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[21] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31x31: Revisiting large kernel design in cnns," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 11 963–11 975.

[22] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, "Visual attention network," *arXiv preprint arXiv:2202.09741*, 2022.

[23] M.-H. Guo, C.-Z. Lu, Q. Hou, Z. Liu, M.-M. Cheng, and S.-M. Hu, "Segnext: Rethinking convolutional attention design for semantic segmentation," in *Adv. Neural Inform. Process. Syst.*, 2022.

[24] Y. Rao, W. Zhao, Y. Tang, J. Zhou, S.-N. Lim, and J. Lu, "Hornet: Efficient high-order spatial interactions with recursive gated convolutions," *arXiv preprint arXiv:2207.14284*, 2022.

[25] S. Liu, T. Chen, X. Chen, X. Chen, Q. Xiao, B. Wu, M. Pechenizkiy, D. Mocanu, and Z. Wang, "More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity," *arXiv preprint arXiv:2207.03620*, 2022.

[26] M. Tan and Q. V. Le, "Mixconv: Mixed depthwise convolutional kernels," *arXiv preprint arXiv:1907.09595*, 2019.

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

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Eur. Conf. Comput. Vis.* Springer, 2014, pp. 740–755.

[29] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 633–641.

[30] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.

[31] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[32] Q. Han, Z. Fan, Q. Dai, L. Sun, M.-M. Cheng, J. Liu, and J. Wang, "On the connection between local attention and dynamic depth-wise convolution," *arXiv preprint arXiv:2106.04263*, 2021.

[33] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu *et al.*, "Gpipe: Efficient training of giant neural networks using pipeline parallelism," *Adv. Neural Inform. Process. Syst.*, vol. 32, pp. 103–112, 2019.

[34] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 819–828.

[35] A. Brock, S. De, S. L. Smith, and K. Simonyan, "High-performance large-scale image recognition without normalization," *arXiv preprint arXiv:2102.06171*, 2021.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. Neural Inform. Process. Syst.*, 2017, pp. 5998–6008.

[37] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, "Rethinking spatial dimensions of vision transformers," *arXiv preprint arXiv:2103.16302*, 2021.

[38] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao, "Focal self-attention for local-global interactions in vision transformers," *arXiv preprint arXiv:2107.00641*, 2021.

[39] C.-F. Chen, Q. Fan, and R. Panda, "Crossvit: Cross-attention multi-scale vision transformer for image classification," *arXiv preprint arXiv:2103.14899*, 2021.

[40] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," *arXiv preprint arXiv:2103.00112*, 2021.

[41] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens, "Scaling local self-attention for parameter efficient visual backbones," *arXiv preprint arXiv:2103.12731*, 2021.

[42] J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, and C. Xu, "Cmt: Convolutional neural networks meet vision transformers," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 12 175–12 185.

[43] Z. Dai, H. Liu, Q. Le, and M. Tan, "Coatnet: Marrying convolution and attention for all data sizes," *Adv. Neural Inform. Process. Syst.*, vol. 34, 2021.

[44] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *Int. Conf. Comput. Vis.*, October 2021, pp. 558–567.

[45] D. Zhou, B. Kang, X. Jin, L. Yang, X. Lian, Z. Jiang, Q. Hou, and J. Feng, "Deepvit: Towards deeper vision transformer," *arXiv preprint arXiv:2103.11886*, 2021.

[46] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," in *Int. Conf. Comput. Vis.*, 2021, pp. 32–42.

[47] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong *et al.*, "Swin transformer v2: Scaling up capacity and resolution," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 12 009–12 019.
[48] Z.-H. Jiang, Q. Hou, L. Yuan, D. Zhou, Y. Shi, X. Jin, A. Wang, and J. Feng, "All tokens matter: Token labeling for training better vision transformers," *Adv. Neural Inform. Process. Syst.*, vol. 34, pp. 18 590–18 602, 2021.
[49] W. Xu, Y. Xu, T. Chang, and Z. Tu, "Co-scale conv-attentional image transformers," in *Int. Conf. Comput. Vis.*, 2021, pp. 9981–9990.
[50] S. Mehta and M. Rastegari, "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer," *arXiv preprint arXiv:2110.02178*, 2021.
[51] H. Cai, C. Gan, and S. Han, "Efficientvit: Enhanced linear attention for high-resolution low-computation visual recognition," *arXiv preprint arXiv:2205.14756*, 2022.
[52] M. Maaz, A. Shaker, H. Cholakkal, S. Khan, S. W. Zamir, R. M. Anwer, and F. S. Khan, "Edgenext: Efficiently amalgamated cnn-transformer architecture for mobile vision applications," *arXiv preprint arXiv:2206.10589*, 2022.
[53] Y. Chen, X. Dai, D. Chen, M. Liu, X. Dong, L. Yuan, and Z. Liu, "Mobile-former: Bridging mobilenet and transformer," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 5270–5279.
[54] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," *arXiv preprint arXiv:1906.05909*, 2019.
[55] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Global context networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.
[56] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7794–7803.
[57] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3588–3597.
[58] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," *Adv. Neural Inform. Process. Syst.*, vol. 31, 2018.
[59] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Adv. Neural Inform. Process. Syst.*, vol. 34, 2021.
[60] H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, G. Izacard, A. Joulin, G. Synnaeve, J. Verbeek *et al.*, "Resmlp: Feedforward networks for image classification with data-efficient training," *arXiv preprint arXiv:2105.03404*, 2021.
[61] Q. Hou, Z. Jiang, L. Yuan, M.-M. Cheng, S. Yan, and J. Feng, "Vision permutator: A permutable mlp-like architecture for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.
[62] J. Wang, S. Zhang, Y. Liu, T. Wu, Y. Yang, X. Liu, K. Chen, P. Luo, and D. Lin, "Riformer: Keep your vision backbone effective but removing token mixer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 443–14 452.
[63] X. Hu, M. Shi, W. Wang, S. Wu, L. Xing, W. Wang, X. Zhu, L. Lu, J. Zhou, X. Wang, Y. Qiao, and J. Dai, "Demystify transformers & convolutions in modern image deep networks," *arXiv preprint arXiv:2211.05781*, 2022.
[64] Z. Pan, J. Cai, and B. Zhuang, "Fast vision transformers with hilo attention," *Advances in Neural Information Processing Systems*, vol. 35, pp. 14 541–14 554, 2022.
[65] W. Lin, Z. Wu, J. Chen, J. Huang, and L. Jin, "Scale-aware modulation meet transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 6015–6026.
[66] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1251–1258.
[67] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "Repvgg: Making vgg-style convnets great again," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 13 733–13 742.
[68] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 558–567.
[69] J. Yang, C. Li, and J. Gao, "Focal modulation networks," *arXiv preprint arXiv:2203.11926*, 2022.
[70] C. Yang, S. Qiao, Q. Yu, X. Yuan, Y. Zhu, A. Yuille, H. Adam, and L.-C. Chen, "Moat: Alternating mobile convolution and attention brings strong vision models," *arXiv preprint arXiv:2210.01820*, 2022.
[71] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.
[72] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*. PMLR, 2015, pp. 448–456.
[73] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.
[74] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inform. Process. Syst.*, 2019, pp. 8026–8037.
[75] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.
[76] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.
[77] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Int. Conf. Comput. Vis.*, 2019, pp. 6023–6032.
[78] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Eur. Conf. Comput. Vis.* Springer, 2016, pp. 646–661.
[79] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *AAAI Conf. on Artif. Intel.*, 2020, pp. 13 001–13 008.
[80] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2020, pp. 702–703.
[81] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Int. Conf. Comput. Vis.*, 2017, pp. 618–626.
[82] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *Int. Conf. Comput. Vis.*, Oct 2017.
[83] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.
[84] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 658–666.
[85] H. Bao, L. Dong, S. Piao, and F. Wei, "BEit: BERT pre-training of image transformers," in *Int. Conf. Learn. Represent.*, 2022.
[86] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.
[87] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Eur. Conf. Comput. Vis.*, 2018, pp. 418–434.

Qibin Hou received his Ph.D. degree from the School of Computer Science, Nankai University. Then, he worked at the National University of Singapore as a research fellow. Now, he is an associate professor at School of Computer Science, Nankai University. He has published more than 20 papers on top conferences/journals, including T-PAMI, CVPR, ICCV, NeurIPS, etc. His research interests include deep learning, image processing, and computer vision.

Cheng-Ze Lu is currently a master student from the College of Computer Science at Nankai University, under the supervision of Prof. Ming-Ming Cheng. Before that, he received his B.E. degree from Xidian University in 2020. His research interests include deep learning and computer vision.

Ming-Ming Cheng received his PhD degree from Tsinghua University in 2012. Then he did 2 years research fellow, with Prof. Philip Torr in Oxford. He is now a professor at Nankai University, leading the Media Computing Lab. His research interests include computer graphics, computer vision, and image processing. He received research awards including ACM China Rising Star Award, IBM Global SUR Award, CCF-Intel Young Faculty Researcher Program.

Jiashi Feng is currently a research scientist at ByteDance. Before joining ByteDance, he was an assistant professor with the Department of Electrical and Computer Engineering at National University of Singapore and a postdoc researcher in the EECS department and ICSI at the University of California, Berkeley. He received his Ph.D. degree from NUS in 2014. His research areas include deep learning and their applications in computer vision. His recent research interest focuses on deep learning models, representation learning, and 3D vision. He received the best technical demo award from ACM MM 2012, best paper award from TASK-CV ICCV 2015, best student paper award from ACM MM 2018. He is also the recipient of Innovators Under 35 Asia, MIT Technology Review 2018. He served as the area chairs for NeurIPS, ICML, CVPR, ICLR, WACV, and program chair for ICMR 2017.