Computational Visual Media
https://doi.org/10.1007/s41095-023-0364-2 Vol. 9, No. 4, December 2023, 733–752

Research Article

Visual attention network

Meng-Hao Guo[1], Cheng-Ze Lu[2], Zheng-Ning Liu[3], Ming-Ming Cheng[2], and Shi-Min Hu[1] (✉)

Abstract While originally designed for natural language processing tasks, the self-attention mechanism has recently taken various computer vision areas by storm. However, the 2D nature of images brings three challenges for applying self-attention in computer vision: (1) treating images as 1D sequences neglects their 2D structures; (2) the quadratic complexity is too expensive for high-resolution images; (3) it only captures spatial adaptability but ignores channel adaptability. In this paper, we propose a novel linear attention named large kernel attention (LKA) to enable self-adaptive and long-range correlations in self-attention while avoiding its shortcomings. Furthermore, we present a neural network based on LKA, namely Visual Attention Network (VAN). While extremely simple, VAN achieves comparable results with similar size convolutional neural networks (CNNs) and vision transformers (ViTs) in various tasks, including image classification, object detection, semantic segmentation, panoptic segmentation, pose estimation, etc. For example, VAN-B6 achieves 87.8% accuracy on ImageNet benchmark, and sets new state-of-the-art performance (58.2% PQ) for panoptic segmentation. Besides, VAN-B2 surpasses Swin-T 4% mIoU (50.1% vs. 46.1%) for semantic segmentation on ADE20K benchmark, 2.6% AP (48.8% vs. 46.2%) for object detection on COCO dataset. It provides a novel method and a simple yet strong baseline for the community. The code is available at `https://github.com/Visual-Attention-Network`.

Keywords vision backbone; deep learning; ConvNets; attention

1 Department of Computer Science, Tsinghua University, Beijing, China. E-mail: M.-H. Guo, gmh20@mails.tsinghua.edu.cn; S.-M. Hu, shimin@tsinghua.edu.cn (✉).

2 Nankai University, Tianjin, China. E-mail: C.-Z. Lu, czlu919@outlook.com; M.-M. Cheng, cmm@nankai.edu.cn.

3 Fitten Tech, Beijing, China. E-mail: lzhengning@gmail.com.

Manuscript received: 2023-04-03; accepted: 2023-06-28

1 Introduction

As the basic feature extractor, vision backbone is a fundamental research topic in the computer vision field. Due to remarkable feature extraction performance, convolutional neural networks (CNNs) [1–3] are indispensable topic in the last decade. After the AlexNet [3] reopened the deep learning decade, a number of breakthroughs have been made to get more powerful vision backbones, by using deeper network [4, 5], more efficient architecture [6–8], stronger multi-scale ability [9–11], and attention mechanisms [12, 13]. Due to translation invariance property and shared sliding-window strategy [14], CNNs are inherently efficient for various vision tasks with arbitrary sized input. More advanced vision backbone networks often result in significant performance gain in various tasks, including image classification [5, 13, 15], object detection [16], semantic segmentation [17], and pose estimation [18].

Based on observed reaction time and estimated signal transmission time along biological pathways [23], cognitive psychology [24] and neuroscience [25] researchers believe that human vision system processes only parts of possible stimuli in detail, while leaving the rest nearly unprocessed. Selective attention is an important mechanism for dealing with the combinatorial aspects of complex search in vision [26]. Attention mechanism can be regarded as an adaptive selecting process based on the input feature. Since the fully attention network [27] has been proposed, self-attention models (i.e., transformer) quickly become the dominated architecture [28, 29] in natural language processing (NLP). Recently, Dosovitskiy et al. [13] proposed the vision transformer (ViT), which introduces transformer backbone into computer vision and outperforms well-known CNNs on image classification tasks. Benefited from its

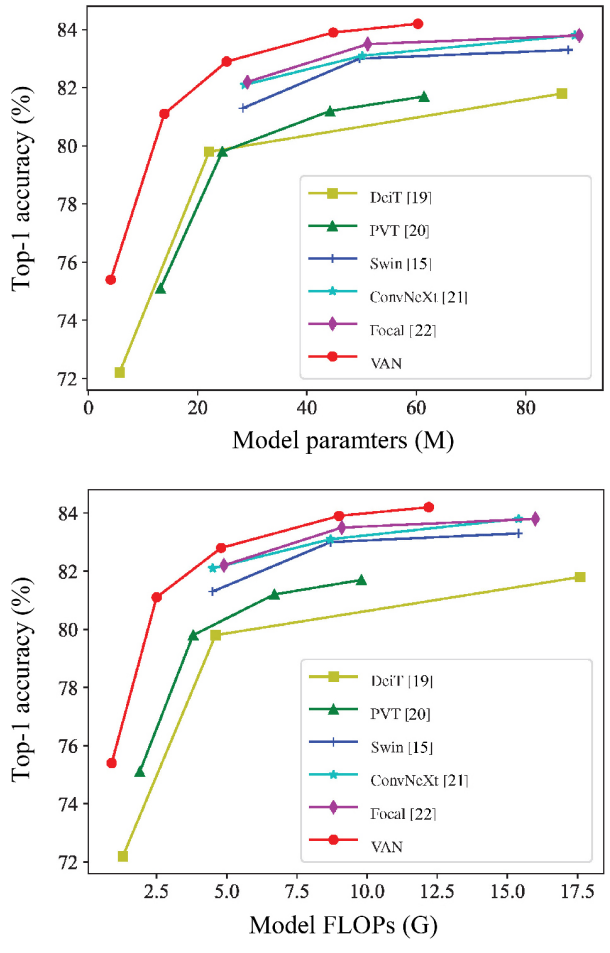

Fig. 1 Results of different models on ImageNet-1K validation set, comparing the performance of recent models DeiT [19], PVT [20], Swin Transformer [15], ConvNeXt [21], Focal Transformer [22], and our VAN. Above: accuracy–parameters trade-off diagram. Under: accuracy–FLOPs trade-off diagram.

powerful modeling capabilities, transformer-based vision backbones quickly occupy the leaderboards of various tasks, including object detection [15], semantic segmentation [17], etc.

Even with remarkable success, convolution operation and self-attention still have their shortcomings. Convolution operation adopts static weight and lacks adaptability, which has been proven critical [12, 16]. As originally designed for 1D NLP tasks, self-attention [13] regards 2D images as 1D sequences, which destroys the crucial 2D structure of the image. It is also difficult to process high-resolution images due to its quadratic computational and memory overhead. Besides, self-attention is a special attention that only considers the adaptability in spatial dimension but ignores the adaptability in channel dimension, which is also important for visual tasks [12, 30–32].

In this paper, we propose a novel linear attention mechanism dubbed large kernel attention (LKA), which is tailored for visual tasks. LKA absorbs the advantages of convolution and self-attention, including local structure information, long-range dependence, and adaptability. Meanwhile, it avoids their disadvantages such as ignoring adaptability in channel dimension. Based on the LKA, we present a novel vision backbone called Visual Attention Network (VAN) that significantly surpasses well-known CNN-based and transformer-based backbones. The contributions of this paper are summarized as follows:

- We design a novel linear attention mechanism named LKA for computer vision, which considers the pros of both convolution and self-attention, while avoiding their cons. It presents a large kernel structure, which is different from previous common architectures. Based on LKA, we further introduce a simple vision backbone called VAN.
- We show that VANs achieve comparable results with similar level CNNs and ViTs in extensive experiments on various tasks, including image classification, object detection, semantic segmentation, instance segmentation, pose estimation, etc.

2 Related work

2.1 Convolutional neural networks

How to effectively compute powerful feature representations is the most fundamental problem in computer vision. Convolutional neural networks (CNNs) [1, 2], utilize local contextual information and translation invariance properties to greatly improve the effectiveness of neural networks. CNNs quickly become the mainstream framework in computer vision since AlexNet [3]. To further improve the usability, researchers put lots of effort in making the CNNs deeper [4, 5, 9, 10, 33, 34] and lighter [6, 8, 35]. Our work has similarity with MobileNet [6], which decouples a standard convolution into two parts, a depthwise convolution and a pointwise convolution (i.e., 1×1 Conv [36]). Our method decomposes a convolution into three parts: depthwise convolution, depthwise and dilated convolution [37, 38], and pointwise convolution. Benefiting from this decomposition, our method is more suitable for efficiently decomposing large kernel convolutions. We

also introduce attention mechanism into our method to obtain adaptive property.

Parallel work. We notice some parallel works, which also adopt large kernel convolution such as ConvNeXt [21], and RepLKNet [39]. Different from them, our work not only focuses on large kernel convolution, but also makes effort on introducing attention mechanism and decomposing a large convolution kernel.

2.2 Visual attention methods

Attention mechanism can be regarded as an adaptive selection process according to the input feature, which is introduced into computer vision in RAM [40]. It has provided benefits in many visual tasks, such as image classification [12, 30], object detection [16, 41], and semantic segmentation [42, 43]. Attention in computer vision can be divided into four basic categories [44], including channel attention, spatial attention, temporal attention, and branch attention, and their combinations such as channel & spatial attention. Each kind of attention has a different effect in visual tasks.

Originating from NLP [27, 28], self-attention is a special kind of attention mechanism. Due to its effectiveness of capturing the long-range dependence and adaptability, it is playing an increasingly important role in computer vision [45–53]. Various deep self-attention networks (i.e., vision transformers) [13, 15, 20, 54–70] have achieved significantly better performance than the mainstream CNNs on different visual tasks, showing the huge potential of attention-based models. However, self-attention is originally designed for NLP. It has three shortcomings when dealing with computer vision tasks: (1) it treats images as 1D sequences which neglects the 2D structure of images; (2) the quadratic complexity is too expensive for high-resolution images; (3) it only achieves spatial adaptability but ignores the adaptability in channel dimension. For vision tasks, different channels often represent different objects [44, 71]. Channel adaptability is also proven important for visual tasks [12, 30, 31, 71, 72]. To solve these problems, we propose a novel visual attention method, namely, LKA. It involves the pros of self-attention such as adaptability and long-range dependence. Besides, it benefits from the advantages of convolution such as making use of local contextual information.

2.3 Vision MLPs

Multilayer Perceptrons (MLPs) [73, 74] were a popular tool for computer vision before CNNs appearing. However, due to high computational requirements and low efficiency, the capability of MLPs was been limited in a long time. Some recent research successfully decouple standard MLP into spatial MLP and channel MLP [75–78]. Such decomposition allows significant computational cost and parameter reduction, which releases the amazing performance of MLP. Readers are referred to recent surveys [79, 80] for a more comprehensive review of MLPs. The most related MLP to our method is the gMLP [78], which not only decomposes the standard MLP but also involves the attention mechanism. However, gMLP has two drawbacks. On the one hand, gMLP is sensitive to input size and can only process fixed-size images. On the other hand, gMLP only considers the global information of the image and ignores their local structure. Our method can make full use of its advantages and avoid its shortcomings.

3 Method

3.1 Large kernel attention (LKA)

Attention mechanism can be regarded as an adaptive selection process, which can select the discriminative features and automatically ignore noisy responses according to the input features. The key step of attention mechanism is producing attention map which indicates the importance of different parts. To do so, we should learn the relationship between different features.

There are two well-known methods to build relationship between different parts. The first one is adopting self-attention mechanism [13, 46, 50, 51] to capture long-range dependence. There are three obvious shortcomings for self-attention applied in computer vision which have been listed in Section 2.2. The second one is to use large kernel convolution [30, 81–83] to build relevance and produce attention map. There are still obvious cons in this way. Large-kernel convolution brings a huge amount of computational overhead and parameters.

To overcome above listed cons and make use of the pros of self-attention and large kernel convolution, we propose to decompose a large kernel convolution operation to capture long-range relationship. As

shown in Fig. 2, a large kernel convolution can be divided into three components: a spatial local convolution (depth-wise convolution), a spatial long-range convolution (depth-wise dilation convolution), and a channel convolution (1×1 convolution). Specifically, we can decompose a $K \times K$ convolution into a $\lceil \frac{K}{d} \rceil \times \lceil \frac{K}{d} \rceil$ depth-wise dilation convolution with dilation d, a $(2d-1) \times (2d-1)$ depth-wise convolution, and a 1×1 convolution. Through the above decomposition, we can capture long-range relationship with slight computational cost and parameters. After obtaining long-range relationship, we can estimate the importance of a point and generate attention map. As demonstrated in Fig. 3(a), the LKA module can be written as

$$Attention = \text{Conv}_{1\times1}(\text{DW-D-Conv}(\text{DW-Conv}(F))) \quad (1)$$

$$Output = Attention \otimes F \quad (2)$$

Here, $F \in \mathbb{R}^{C\times H\times W}$ is the input feature. $Attention \in \mathbb{R}^{C\times H\times W}$ denotes the attention map. The value in attention map indicates the importance of each feature. $\otimes$ means element-wise product. Different from common attention methods, LKA dose not require an additional normalization function like sigmoid and softmax, which is demonstrated in Table 3. We also believe the key characteristics of attention methods is adaptively adjusting output based on input feature, but not the normalized attention map. As shown in Table 1, our proposed LKA combines the advantages of convolution and self-attention. It takes the local contextual information, large receptive field, linear complexity, and dynamic process into consideration. Furthermore, LKA not only achieves the adaptability in the spatial dimension but also the adaptability in the channel dimension. It worth noting that different channels often represent different objects in deep neural networks [44, 71] and adaptability in the channel dimension is also important for visual tasks.

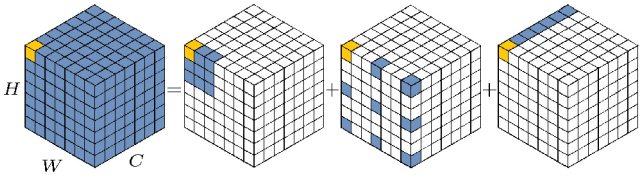

Fig. 2 Decomposition diagram of large-kernel convolution. A standard convolution can be decomposed into three parts: a depth-wise convolution (DW-Conv), a depth-wise dilation convolution (DW-D-Conv), and a pointwise convolution (1×1 Conv). The colored grids represent the location of convolution kernel and the yellow grid means the center point. The diagram shows that a 13×13 convolution is decomposed into a 5×5 depth-wise convolution, a 5×5 depth-wise dilation convolution with dilation rate 3, and a pointwise convolution. Note: zero paddings are omitted in the figure.

3.2 Visual attention network (VAN)

Our VAN has a simple hierarchical structure, i.e., a sequence of four stages with decreasing output spatial resolution, i.e., $\frac{H}{4} \times \frac{W}{4}$, $\frac{H}{8} \times \frac{W}{8}$, $\frac{H}{16} \times \frac{W}{16}$, and $\frac{H}{32} \times \frac{W}{32}$. Here, H and W denote the height and width of the

Table 1 Desirable properties belonging to convolution, self-attention, and LKA

Property	Conv	Self-attention	LKA
Local receptive field	✓	✗	✓
Long-range dependence	✗	✓	✓
Spatial adaptability	✗	✓	✓
Channel adaptability	✗	✗	✓
Computational complexity	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$

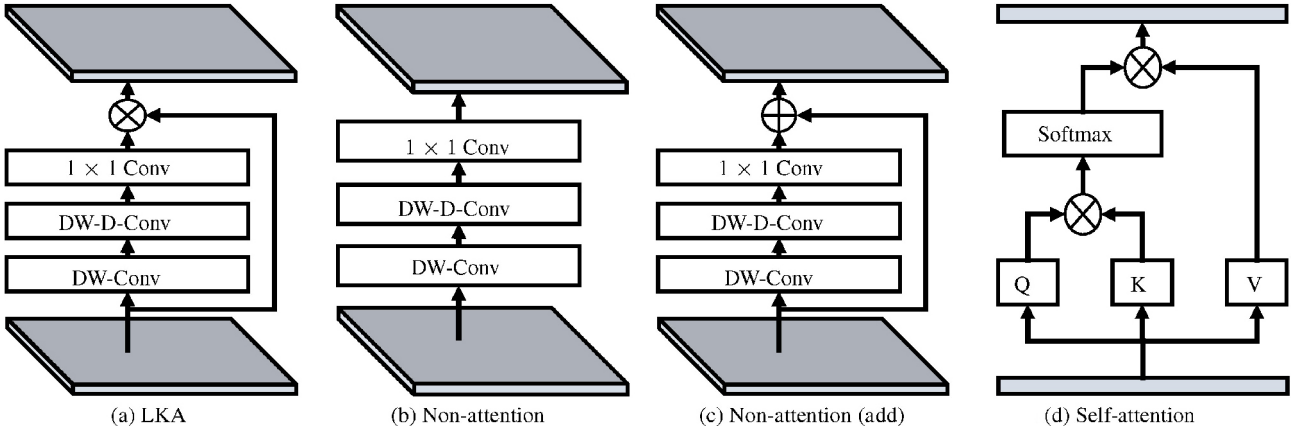

Fig. 3 Structure of different modules: (a) the proposed large kernel attention (LKA); (b) non-attention module; (c) replace multiplication in LKA with addition; (d) self-attention. It is worth noting that (d) is designed for 1D sequences.

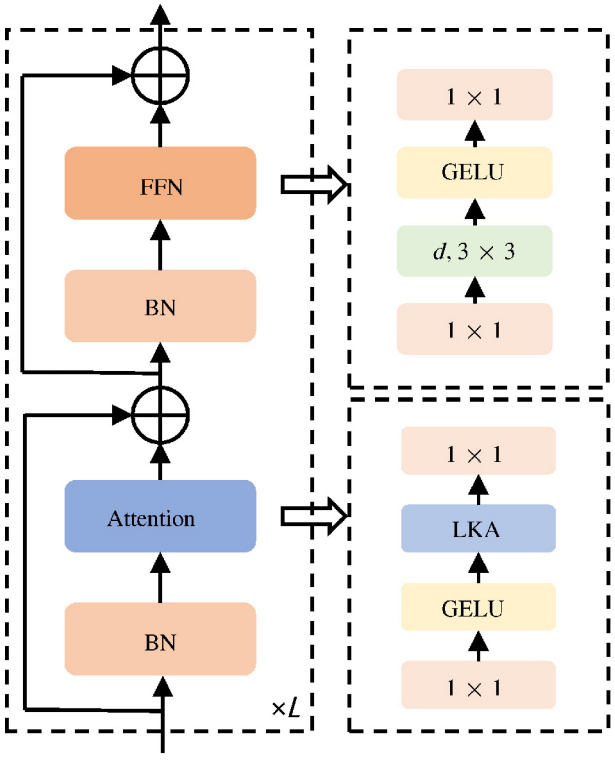

Fig. 4 A stage of VAN. d means depth-wise convolution. $k \times k$ denotes $k \times k$ convolution.

Table 2 Number of parameters for different forms of a 21×21 convolution. For instance, when the number of channels $C = 32$, standard convolution and MobileNet decomposition use 133× and 4.5× more parameters than our decomposition respectively

Channel number	Standard convolution	Decomposition type	
		MobileNet [6]	Ours
$C = 32$	451,584	15,136	3,392
$C = 64$	1,806,336	32,320	8,832
$C = 128$	7,225,344	72,832	25,856
$C = 256$	28,901,376	178,432	84,480
$C = 512$	115,605,504	487,936	300,032

Table 3 Ablation study of different modules in LKA. Top-1 accuracy (Acc) on ImageNet validation set suggests that each part is critical. w/o Attention means we adopt Fig. 3(b)

VAN-B0	Params. (M)	FLOPs (G)	Acc (%)
w/o DW-Conv	4.1	0.9	74.9
w/o DW-D-Conv	4.0	0.9	74.1
w/o Attention	4.1	0.9	74.3
w/o Attention (add)	4.1	0.9	74.6
w/o 1×1 Conv	3.8	0.8	74.6
w/ Sigmoid	4.1	0.9	75.2
w/ (21×21 Conv)	124.7	23.2	76.0
VAN-B0	4.1	0.9	75.4

input image, respectively. With the decreasing of resolution, the number of output channels is increasing. The change of output channel C_i is presented in Table 5.

Table 4 Throughput of Swin Transformer and VAN on RTX 3090

Method	FLOPs (G)	Throughput (images/s)	Acc (%)
Swin-T	4.5	821	81.3
Swin-S	8.7	500	83.0
Swin-B	15.4	376	83.5
VAN-B0	0.9	2140	75.4
VAN-B1	2.5	1420	81.1
VAN-B2	5.0	762	82.8
VAN-B3	9.0	452	83.9
VAN-B4	12.2	341	84.2

For each stage as shown in Fig. 4, we firstly downsample the input and use the stride number to control the downsample rate. After the downsample, all other layers in a stage stay the same output size, i.e., spatial resolution and the number of channels. Then, L groups of batch normalization [84], 1×1 Conv, GELU activation [85], large kernel attention, and feed-forward network (FFN) [86] are stacked in sequence to extract features. We design seven architectures VAN-B0, VAN-B1, VAN-B2, VAN-B3, VAN-B4, VAN-B5, VAN-B6 according to the parameters and computational cost. The details of the whole network are shown in Table 5.

Complexity analysis. We present the parameters and floating point operations (FLOPs) of our decomposition. Bias is omitted in the computation process for simplifying format. We assume that the input and output features have the same size $H \times W \times C$. The number of parameters $P(K, d)$ and FLOPs $F(K, d)$ can be denoted as

$$P(K, d) = C\left[\left\lceil\frac{K}{d}\right\rceil^2 + (2d-1)^2 + C\right] \tag{3}$$

$$F(K, d) = P(K, d) \times H \times W \tag{4}$$

Here, d means the dilation rate and K is the kernel size. According to the formula of FLOPs and parameters, the ratio of budget saving is the same for FLOPs and parameters.

Hyperparameter selection. It is worth noting that Eq. (3) has a minimum value independent of C when K is fixed. We adopt $K = 21$ by default. For $K = 21$, Eq. (3) takes the minimum value when $d = 3$, which corresponds to 5×5 depth-wise convolution and 7×7 depth-wise convolution with dilation 3. For different number of channels, we show the specific parameters in Table 2. It shows that our decomposition owns significant advantages

Table 5 Detailed setting for different versions of the VAN. e.r. represents expansion ratio in the feed-forward network

Stage	Output size	e.r.	VAN-						
			B0	B1	B2	B3	B4	B5	B6
1	$\frac{H}{4} \times \frac{W}{4} \times C$	8	$C = 32$ $L = 3$	$C = 64$ $L = 2$	$C = 64$ $L = 3$	$C = 64$ $L = 3$	$C = 64$ $L = 3$	$C = 96$ $L = 3$	$C = 96$ $L = 6$
2	$\frac{H}{8} \times \frac{W}{8} \times C$	8	$C = 64$ $L = 3$	$C = 128$ $L = 2$	$C = 128$ $L = 3$	$C = 128$ $L = 5$	$C = 128$ $L = 6$	$C = 192$ $L = 3$	$C = 192$ $L = 6$
3	$\frac{H}{16} \times \frac{W}{16} \times C$	4	$C = 160$ $L = 5$	$C = 320$ $L = 4$	$C = 320$ $L = 12$	$C = 320$ $L = 27$	$C = 320$ $L = 40$	$C = 480$ $L = 24$	$C = 384$ $L = 90$
4	$\frac{H}{32} \times \frac{W}{32} \times C$	4	$C = 256$ $L = 2$	$C = 512$ $L = 2$	$C = 512$ $L = 3$	$C = 512$ $L = 3$	$C = 512$ $L = 3$	$C = 768$ $L = 3$	$C = 768$ $L = 6$
Parameters (M)			4.1	13.9	26.6	44.8	60.3	90.0	200
FLOPs (G)			0.9	2.5	5.0	9.0	12.2	17.2	38.4

in decomposing large kernel convolution in terms of parameters and FLOPs.

Coverage. Here, we adopt a formal expression to show how our decomposition covers the whole area. We take 21×21 convolution as example. For a standard 21×21 convolution, we can build connection between $p(i, j)$ and $p(i + x, j + y)$ directly, where $p(i, j)$ is the center point and $x, y \in \{-10, -9, \cdots 0, \cdots, 9, 10\}$ is offset coordinate. As for our decomposition, we also can build this connection via information passing. As shown in Fig. 2, the first two steps transfer information in spatial dimension, and the third step passes information in channel dimension. The third step is easy to understand because the interactions in the channel dimension are dense. Here, we present the information passing in spatial dimension between $p(i, j)$ and $p(i + x, j + y)$. The first step is the interaction between $p(i+x, j+y)$ and $p(i + 3\lfloor \frac{x}{3} \rfloor, j + 3\lfloor \frac{y}{3} \rfloor)$. The second step is the interaction between $p(i, j)$ and $p(i + 3\lfloor \frac{x}{3} \rfloor, j + 3\lfloor \frac{y}{3} \rfloor)$. The above two steps correspond to DW-Conv and DW-D-Conv to finish interaction in spatial dimension. After completing the spatial interaction, a 1×1 convolution follows them and completes the channel interaction.

4 Experiments

In this section, quantitative and qualitative experiments are exhibited to demonstrate the effectiveness and efficiency of the proposed VAN. We conduct quantitative experiments on ImageNet-1K [88] and ImageNet-22K image classification dataset, COCO [89] benchmark for object detection, instance segmentation, panoptic segmentation, and pose estimation, and ADE20K [90] semantic segmentation dataset. Furthermore, we visualize the experimental results and class activation mapping (CAM) [91] by using Grad-CAM [87] on ImageNet validation set. Experiments are based on PyTorch [92] and Jittor [93].

4.1 Image classification

4.1.1 ImageNet-1K experiments

Settings. We conduct image classification on ImageNet-1K [88] dataset. It contains 1.28M training images and 50k validation images from 1000 different categories. The whole training scheme mostly follows Ref. [19]. We adopt random clipping, random horizontal flipping, label-smoothing [94], mixup [95], cutmix [96], and random erasing [97] to augment the training data. In the training process, we train our VAN for 300 epochs by using AdamW [98, 99] optimizer with momentum=0.9, weight decay=5×10^{-2}, and batch size = 1024. Cosine schedule [100] and warm-up strategy are employed to adjust the learning rate (LR). The initial LR is set to 5×10^{-4}. We adopt a variant of LayerScale [101] in attention layer which replaces $x_{\text{out}} = x + \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{\text{d}})f(x)$ with $x_{\text{out}} = x + \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{\text{d}})(f(x) + x)$ with initial value 0.01. Exponential moving average (EMA) [102] is also applied to improve training process. During the eval stage, we report the Top-1 accuracy on ImageNet validation set under single crop setting.

Ablation study. We conduct an ablation study to prove that each component of LKA is critical. In order to obtain experimental results quickly, we choose VAN-B0 as our baseline model. The experimental results in Table 3 indicate that all components in LKA are indispensable to improve performance.

- **DW-Conv.** DW-Conv can make use of the local contextual information of images. Without it, the classification performance will drop by 0.5% (74.9% vs. 75.4%), showing the importance of local structural information in image processing.
- **DW-D-Conv.** DW-D-Conv denotes depth-wise dilation convolution which plays a role in capturing long-range dependence in LKA. Without it, the classification performance will drop by 1.3% (74.1% vs. 75.4%) which confirms our viewpoint that long-range dependence is critical for visual tasks.
- **Attention mechanism.** The introduction of the attention mechanism can be regarded as making network achieve adaptive property. Benefited from it, the VAN-B0 achieves about 1.1% (74.3% vs. 75.4%) improvement. Besides, replacing attention with adding operation is also not achieving a lower accuracy.
- **1×1 Conv.** Here, 1×1 Conv captures relationship in channel dimension. Combining with attention mechanism, it introduces adaptability in channel dimension. It brings about 0.8% (74.6% vs. 75.4%) improvement which proves the necessity of the adaptability in channel dimension.
- **Sigmoid function.** Sigmoid function is a common normalization function to normalize attention map from 0 to 1. However, we find it is not necessary for LKA module in our experiment. Without sigmoid, our VAN-B0 achieves 0.2% (75.4% vs. 75.2%) improvement and less computation.

Through the above analysis, we can find that our proposed LKA can utilize local information, capture long-distance dependencies, and have adaptability in both channel and spatial dimension. Furthermore, experimental results prove that all properties are positive for recognition tasks. Although standard convolution can make full use of the local contextual information, it ignores long-range dependencies and adaptability. As for self-attention, although it can capture long-range dependencies and has adaptability in spatial dimensions, it neglects the local information and the adaptability in the channel dimension. Meanwhile, We also summarize above discussion in Table 1.

Besides, we also conduct ablation study to decompose different size convolution kernels in Table 6. We can find that decomposing a 21×21 convolution works better than decomposing a 7×7 convolution which demonstrates that large kernel is critical for visual tasks. Decomposing a larger 28×28 convolution, we find that the gain is not obvious comparing with decomposing a 21×21 convolution. Thus, we choose to decompose a 21×21 convolution by default.

Comparison with existing methods. Table 7 presents the comparison of VAN with other MLPs, CNNs and ViTs. VAN outperforms common CNNs (ResNet [5], ConvNeXt [21], etc.), and ViTs (PVT [20] and Swin Transformer [15], etc.) with similar parameters and computational cost. We visually show the comparison of our method with similar level classical methods on different tasks in Fig. 7, which clearly reveals the improvement of our method. In the following discussion, we will choose a representative network in each category.

ConvNeXt [21] is a special CNN which absorbs the some advantages of ViTs such as large receptive field (7×7 convolution) and advanced training strategy (300 epochs, data augmentation, etc). Comparing VAN with ConvNeXt [21], VAN-B2 surpasses ConvNeXt-T by 0.7% (82.8% vs. 82.1%) since VAN has larger receptive field and adaptive ability. Swin Transformer is a well-known ViT variant that adopts local attention and shifted window manner. Due to that VAN is friendly for 2D structural information, has larger receptive field, and achieves adaptability in channel dimension, VAN-B2 surpasses Swin-T by 1.5% (82.8% vs. 81.3%). Considering the some methods do not adopt LayerScale, we also conduct experiments without LayerScale for fair comparison. As shown in Table 7, we find that LayerScale has a weak change on VAN's performance. It may be because LayerScale is designed for original self-attention and not suitable for LKA.

Throughput. We test the throughput of the Swin Transformer [15] and VAN on some hardware environment with the RTX 3090. Results are shown in Table 4. Besides, we also plot the accuracy–throughput diagram in Fig. 5, which clearly demonstrates that VAN achieves a better accuracy–throughput trade-off than Swin Transformer [15].

4.1.2 Visualization

Class activation mapping (CAM) is a popular tool to visualize the discriminative regions (attention maps). We adopt Grad-CAM [87] to visualize the

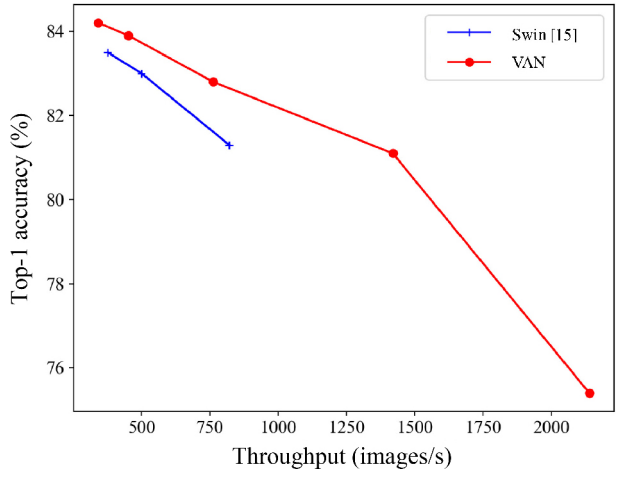

Fig. 5 Accuracy–throughput diagram. It claerly shows that VAN achieves a better trade-off than Swin Transformer [15].

Table 6 Ablation study of different kernel size K in LKA. Acc means Top-1 accuracy on ImageNet validation set

Method	K	Dilation	Params. (M)	FLOPs (G)	Acc (%)
VAN-B0	7	2	4.03	0.85	74.8
VAN-B0	14	3	4.07	0.87	75.3
VAN-B0	21	3	4.11	0.88	75.4
VAN-B0	28	4	4.14	0.90	75.4

attentions on the ImageNet validation set produced by VAN-B2 model. Results in Fig. 6 show that VAN-B2 can clearly focus on the target objects. Thus, the visualizations intuitively demonstrate the effectiveness of our method. Furthermore, we also compare different CAM produced by Swin-T [15], ConvNeXt-T [21], and VAN-B2. We can find that

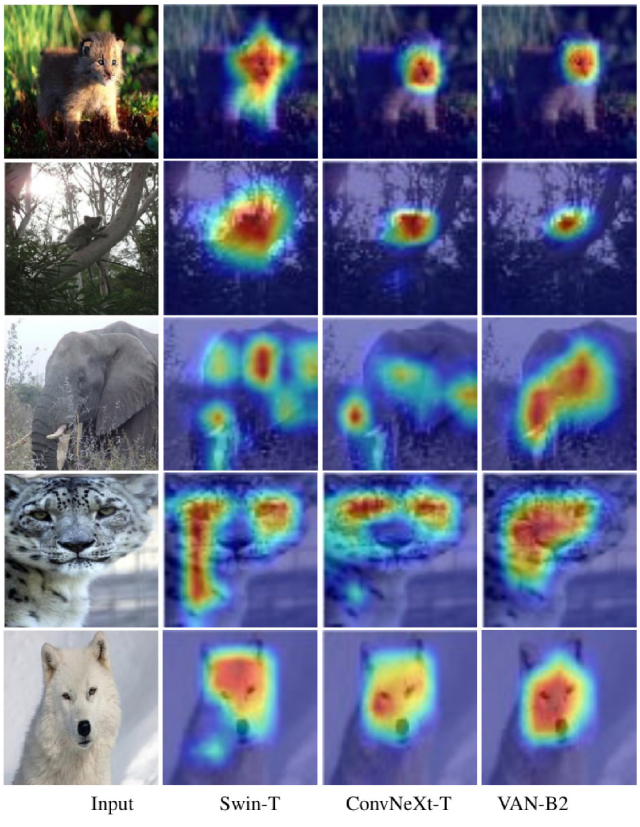

Fig. 6 Visualization results. All images come from different categories in ImageNet validation set. CAM is produced by using Grad-CAM [87]. We compare different CAMs produced by Swin-T [15], ConvNeXt-T [21], and VAN-B2.

the activation area of VAN-B2 is more accurate. In particular, our method shows obvious advantages when the object is dominant in an image (last 3 lines

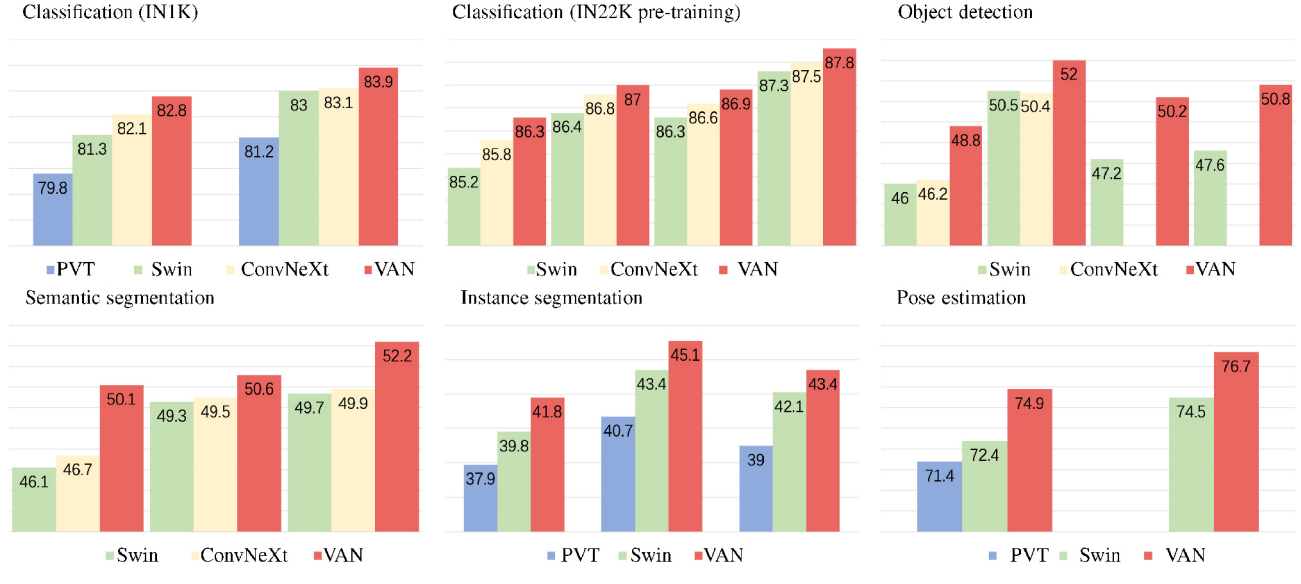

Fig. 7 Comparing with similar level PVT [20], Swin Transformer [15], and ConvNeXt [21] on various tasks, including image classification, object detection, semantic segmentation, instance segmentation, and pose estimation.

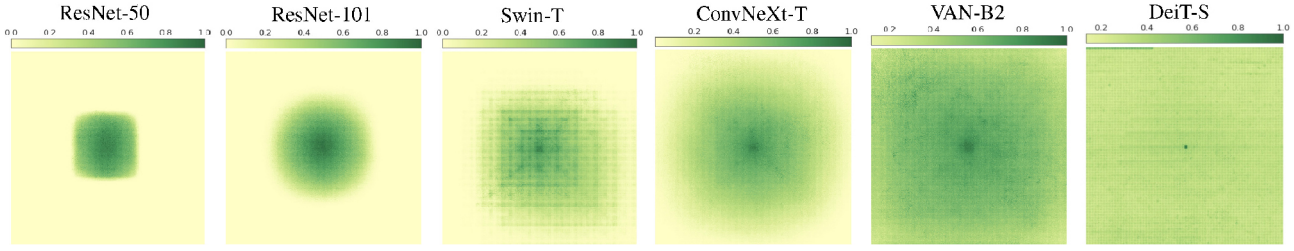

Fig. 8 Visualization results of effective receptive field (ERF), which is visualized by using Ref. [106]. We randomly select 100 images and visualize their averaged ERF in an image. We compare different ERF produced by different methods.

Table 7 Comparision with the state-of-the-art methods on ImageNet validation set. Params means parameter. FLOPs denotes floating point operations. Top-1 Acc represents Top-1 accuracy. w/o LS means without LayerScale

Method	Params. (M)	FLOPs (G)	Top-1 Acc (%)
Tansformer-based methods			
DeiT-Tiny/16 [19]	5.7	1.3	72.2
DeiT-Small/16 [19]	22.1	4.6	79.8
PVT-Tiny [20]	13.2	1.9	75.1
PVT-Small [20]	24.5	3.8	79.8
PVT-Medium [20]	44.2	6.7	81.2
PVT-Large [20]	61.4	9.8	81.7
Swin-T [15]	28.3	4.5	81.3
Swin-S [15]	49.6	8.7	83.0
Focal-T [22]	29.1	4.9	82.2
Focal-S [22]	51.1	9.1	83.5
CNN-based methods			
ResNet18 [5]	11.7	1.8	69.8
ResNet50 [5]	25.6	4.1	76.5
ResNet101 [5]	44.7	7.9	77.4
ResNet152 [5]	60.2	11.6	78.3
ConvNeXt-T [21]	28.6	4.5	82.1
ConvNeXt-S [21]	50.1	8.7	83.1
ConvNeXt-B [21]	89.0	15.4	83.8
VAN-B0	4.1	0.9	75.4
VAN-B0 w/o LS	4.1	0.9	75.2
VAN-B1	13.9	2.5	81.1
VAN-B1 w/o LS	13.9	2.5	81.0
VAN-B2	26.6	5.0	82.8
VAN-B2 w/o LS	26.6	5.0	82.9
VAN-B3	44.8	9.0	83.9
VAN-B3 w/o LS	44.8	9.0	83.8
VAN-B4	60.3	12.2	84.2
VAN-B4 w/o LS	60.3	12.2	84.2

Table 8 Comparision with the state-of-the-art methods on ImageNet validation set. Params means parameter. FLOPs denotes floating point operations. Top-1 Acc represents Top-1 accuracy. All models are pretrained on ImageNet-22K dataset

Method	Params. (M)	Input size	FLOPs (G)	Top-1 Acc (%)
Tansformer-based methods				
ViT-B/16 [13]	87	384^2	55.5	85.4
Swin-S [15]	50	224^2	8.7	83.2
Swin-B [15]	88	224^2	15.4	85.2
Swin-B [15]	88	384^2	47.0	86.4
Swin-L [15]	197	224^2	34.5	86.3
Swin-L [15]	197	384^2	103.9	87.3
CoAtNet-3 [103]	168	384^2	107.4	87.6
CNN-based methods				
EffNetV2-L [104]	120	480^2	53.0	86.8
EffNetV2-XL [104]	208	480^2	94.0	87.3
ConvNeXt-S [21]	50	224^2	8.7	84.6
ConvNeXt-S [21]	50	384^2	25.5	85.8
ConvNeXt-B [21]	89	384^2	45.1	86.8
ConvNeXt-B [21]	89	224^2	15.4	85.8
ConvNeXt-L [21]	198	224^2	34.4	86.6
ConvNeXt-L [21]	198	384^2	101.0	87.5
ConvNeXt-XL [21]	350	384^2	179.0	87.8
FocalNet-L [105]	197.1	224^2	34.2	86.5
FocalNet-L [105]	197.1	384^2	100.6	87.3
RepLKNet-B [39]	79	224^2	15.3	85.2
RepLKNet-B [39]	79	384^2	45.1	86.0
RepLKNet-L [39]	172	384^2	96.0	86.6
VAN-B4	60	224^2	12.2	85.7
VAN-B4	60	384^2	35.9	86.6
VAN-B5	90	224^2	17.2	86.3
VAN-B5	90	384^2	50.6	87.0
VAN-B6	200	224^2	38.9	86.9
VAN-B6	200	384^2	114.3	87.8

in Fig. 6), which demonstrates its ability to capture long-range dependence.

Effective receptive field (ERF) is proposed by Ref. [106]. To demonstrate the capability of our method to capture long-range dependencies, we visualize the ERF by adopting Ref. [106]. Here, we randomly select 100 images in ImageNet val dataset and resize them to 1120×1120. Then, we visualize their ERF and average them in a single image. As shown in Fig. 8, we compare the ERF of different methods, including ResNet [5], Swin Transformer [15], ConvNeXt [21], DeiT [19], and our VAN. It clearly shows VAN-B2 has a larger ERF than Swin-T [15] and ConvNeXt [21]. Deit-S has a global ERF and

VAN-B2 shows a similar ability to capture long-range dependencies for 1120 × 1120 images.

Attention map. Besides CAM and ERF, we also visualize the attention map in LKA directly. The visualization method follows FocalNet [105], which visualizes the absolute value of attention map. We choose images with single or multi objects in ImageNet val set. As shown in Fig. 9, it demonstrates that the attention of LKA focuses on the main objects, which is a meaningful result.

4.1.3 Pretraining on ImageNet-22K

Settings. ImageNet-22K is a large-scale image classification dataset, which contains about 14M images and 21,841 categories. Following Swin Transformer [15] and ConvNeXt [21], we use it to pretrain our VAN for 90 epochs without EMA. The batch size is set as 8196. Other training details are the same with ImageNet-1K settings. After pretrained on ImageNet-22K, we fine-tune our model on ImageNet-1K for 30 epochs. We pretrain our model with 224×224 input and fine-tune our model with 224×224 and 384 × 384 respectively.

Results. We compare current state-of-the-art CNNs (e.g., ConvNeXt [21], EFFNetV2 [104]) and ViTs (e.g., Swin Transformer [15], ViT [13], and CoAtNet [103]). As shown in Table 8, VAN achieves 87.8% Top-1 accuracy with 200M parameters and surpasses the same level ViT [13], Swin Transformer [15], EFFNetV2 [104], and ConvNeXt [21] on different resolution, which proves the strong capability to adapt large-scale pretraining.

4.2 Object detection

Settings. We conduct object detection and instance segmentation experiments on COCO 2017 benchmark [89], which contains 118k images in training set and 5k images in validation set. MMDetection [117] is used as the codebase to implement detection models. For fair comparison, we adopt the same training/validating strategies with Swin Transformer [15] and PoolFormer [108]. Many kinds of detection models (e.g., Mask R-CNN [110], RetinaNet [107], Cascade Mask R-CNN [112], Sparse R-CNN [118], etc.) are included to demonstrate the effectiveness of our method. All backbone models are pre-trained on ImageNet training set. For object detection task, AdamW [98] optimizer with initial learning rate 0.0001 and weight decay 0.05 is adopted to train related models. The batch size is set as 16. There are two training schedule 1× (12 epochs) and 3× (36 epochs), which we illustrate in specific experiments. For fair comparsion, we also adopt multi-scale training like Swin Transformer [15] and ConvNeXt [21] for 3× schedule.

Results. According to Table 9 and Table 10, we find that VAN surpasses CNN-based method ResNet [5] and transformer-based method PVT [20] with a large margin under RetinaNet [107] 1× and Mask R-CNN [110] 1× settings. Besides, we also compare

Table 9 Object detection on COCO 2017 dataset. #P means parameter. RetinaNet 1× denotes models based on RetinaNet [107] and we train them for 12 epochs. PF represents PoolFormer

Backbone	RetinaNet 1×						
	#P (M)	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_S (%)	AP_M (%)	AP_L (%)
VAN-B0	13.4	**38.8**	**58.8**	**41.3**	**23.4**	**42.8**	**50.9**
ResNet18 [5]	21.3	31.8	49.6	33.6	16.3	34.3	43.2
PF-S12 [20]	21.7	36.2	56.2	38.2	20.8	39.1	48.0
PVT-T [20]	23.0	36.7	56.9	38.9	22.6	38.8	50.0
VAN-B1	23.6	**42.3**	**63.1**	**45.1**	**26.1**	**46.2**	**54.1**
ResNet50 [5]	37.7	36.3	55.3	38.6	19.3	40.0	48.8
PVT-S [20]	34.2	40.4	61.3	43.0	25.0	42.9	55.7
PF-S24 [108]	31.1	38.9	59.7	41.3	23.3	42.1	51.8
PF-S36 [108]	40.6	39.5	60.5	41.8	22.5	42.9	52.4
CMT-S [109]	44.3	44.3	65.5	47.5	27.1	48.3	**59.1**
VAN-B2	36.3	**44.9**	**65.7**	**48.4**	**27.4**	**49.2**	58.7
ResNet101 [5]	56.7	38.5	57.8	41.2	21.4	42.6	51.1
PVT-M [20]	53.9	41.9	63.1	44.3	25.0	44.9	57.6
VAN-B3	54.5	**47.5**	**68.4**	**51.2**	**30.9**	**52.1**	**62.4**

Table 10 Object detection and instance segmentation on COCO 2017 dataset. #P means parameter. Mask R-CNN 1× denotes models based on Mask R-CNN [110] and we train them for 12 epochs. AP^b and AP^m refer to bounding box AP and mask AP respectively. PF means PoolFormer

Backbone	Mask R-CNN 1×						
	#P (M)	AP^b (%)	AP^b_{50} (%)	AP^b_{75} (%)	AP^m (%)	AP^m_{50} (%)	AP^m_{75} (%)
VAN-B0	23.9	**40.2**	**62.6**	**44.4**	**37.6**	**59.6**	**40.4**
ResNet18 [5]	31.2	34.0	54.0	36.7	31.2	51.0	32.7
PF-S12 [108]	31.6	37.3	59.0	40.1	34.6	55.8	36.9
PVT-T [20]	32.9	36.7	59.2	39.3	35.1	56.7	37.3
VAN-B1	33.5	**42.6**	**64.2**	**46.7**	**38.9**	**61.2**	**41.7**
ResNet50 [5]	44.2	38.0	58.6	41.4	34.4	55.1	36.7
PVT-S [20]	44.1	40.4	62.9	43.8	37.8	60.1	40.3
PF-S24 [108]	41.0	40.1	62.2	43.4	37.0	59.1	39.6
PF-S36 [108]	50.5	41.0	63.1	44.8	37.7	60.1	40.0
CMT-S [109]	44.5	44.6	66.8	48.9	40.7	63.9	43.4
VAN-B2	46.2	46.4	67.8	51.0	41.8	65.2	44.9
ResNet101 [5]	63.2	40.4	61.1	44.2	36.4	57.7	38.8
PVT-M [20]	63.9	42.0	64.4	45.6	39.0	61.6	42.1
VAN-B3	64.4	**48.3**	69.6	**53.3**	**43.4**	**67.0**	**46.8**

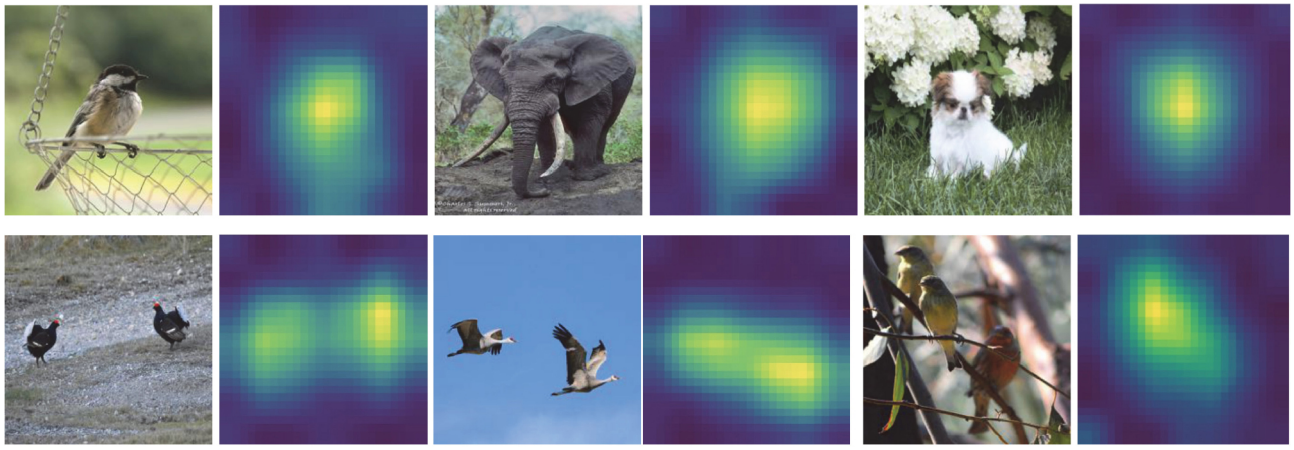

Fig. 9 Visualization results of attention map. We select images in ImageNet val set and visualize their attention maps directly. The visualization method follows FocalNet [105], which visualizes the absolute value of attention map.

the state-of-the-art methods Swin Transformer [15] and ConvNeXt [21] in Table 11. Results show that VAN achieves the state-of-the-art performance with different detection methods such as Mask R-CNN [110] and Cascade Mask R-CNN [112].

4.3 Semantic segmentation

Settings. We conduct experiments on ADE20K [90], which contains 150 semantic categories for semantic segmentation. It consists of 20,000, 2000, and 3000 respectively for training, validation, and testing. MMSEG [119] is used as the base framework and two famous segmentation heads, Semantic FPN [115] and UperNet [116], are employed for evaluating our VAN backbones. For a fair comparison, we adopt two training/validating schemes following Refs. [108] and [15] and quantitative results on the validation set are shown in the upper and lower part in Table 12, respectively. All backbone models are pre-trained on ImageNet-1K or ImageNet-22K training set. For segmentation experiments, we adopt some common data augmentations, including random horizontal flipping, random scaling, and random cropping. We choose AdamW with initial learning 0.00006 and weight decay 0.01 as optimizer. The batch size is set as 16. We adopt poly-learning rate decay policy. We train our model 40k or 160k iterations respectively for fair comparison.

Table 11 Comparison with the state-of-the-art vision backbones on COCO 2017 benchmark. All models are trained for 36 epochs. We calculate FLOPs with input size of 1280 × 800. #F means FLOPs. #P denotes parameters

Backbone	Method	AP^b (%)	AP^b_{50} (%)	AP^b_{75} (%)	#P (M)	#F (G)
Swin-T [5]	Mask R-CNN [110]	46.0	68.1	50.3	48	264
ConvNeXt-T [15]		46.2	67.9	50.8	48	262
MPViT-T [111]		48.4	70.5	52.6	43	268
VAN-B2		48.8	70.0	53.6	46	273
ResNet50 [5]	Cascade Mask R-CNN [112]	46.3	64.3	50.5	82	739
Swin-T [15]		50.5	69.3	54.9	86	745
ConvNeXt-T [21]		50.4	69.1	54.8	86	741
VAN-B2		**52.0**	**70.9**	**56.4**	84	752
ResNet50 [5]	ATSS [113]	43.5	61.9	47.0	32	205
Swin-T [15]		47.2	66.5	51.3	36	215
VAN-B2		**50.2**	**69.3**	**55.1**	34	221
ResNet50 [5]	GFL [114]	44.5	63.0	48.3	32	208
Swin-T [15]		47.6	66.8	51.7	36	215
VAN-B2		**50.8**	**69.8**	**55.7**	34	224

Results. From the upper part in Table 12, compared with different backbones using FPN [115], VAN-based methods are superior to CNN-based (ResNet [5], ResNeXt [7]) or transformer-based (PVT [20], PoolFormer [108], PVTv2 [86]) methods. For instance, we surpass four PVTv2 [86] variants by +1.3% (B0), +0.4% (B1), +1.5% (B2), +0.8% (B3) mIoU under comparable parameters and FLOPs. In the lower part in Table 12, when compared with previous state-of-the-art CNN-based methods and Swin-Transformer-based methods, four VAN variants also show excellent performance with comparable parameters and FLOPs. For instance, based on UperNet [116], VAN-B2 is +5.2% and +4.0% mIoU higher than ResNet-101 and Swin-T, respectively. For ImageNet-22K pretrained models, VAN also performs better than Swin Transformer [15] and ConvNeXt [21] with less computational overhead, which is shown in Table 13.

Table 12 Results of semantic segmentation on ADE20K [90] validation set. The upper and lower parts are obtained under two different training/validation schemes following Refs. [108] and [15]. We calculate FLOPs with input size 512 × 512 for Semantic FPN [115] and 2048× 512 for UperNet [116]. #P means parameters. #F denotes FLOPs

Method	Backbone	#P (M)	#F (G)	mIoU (%)
Semantic FPN [115]	PVTv2-B0 [86]	8	25	37.2
	VAN-B0	8	26	**38.5**
	ResNet18 [5]	16	32	32.9
	PVT-Tiny [20]	17	33	35.7
	PoolFormer-S12 [108]	16	31	37.2
	PVTv2-B1 [86]	18	34	42.5
	VAN-B1	18	35	**42.9**
	ResNet50 [5]	29	46	36.7
	PVT-Small [20]	28	45	39.8
	PoolFormer-S24 [108]	23	39	40.3
	PVTv2-B2 [86]	29	46	45.2
	VAN-B2	30	48	**46.7**
	ResNet101 [5]	48	65	38.8
	PVT-Medium [20]	48	61	43.5
	PoolFormer-S36 [108]	35	48	42.0
	PVTv2-B3 [86]	49	62	47.3
	VAN-B3	49	68	**48.1**
UperNet [116]	ResNet-101 [5]	86	1029	44.9
OCRNet [42]		56	923	45.3
HamNet [43]		69	1111	46.8
UperNet [116]	Swin-T [15]	60	945	46.1
	ConvNeXt-T [21]	60	939	46.7
	MPViT-S [111]	52	943	48.3
	VAN-B2	57	948	50.1
	Swin-S [15]	81	1038	49.3
	ConvNeXt-S [21]	82	1027	49.5
	VAN-B3	75	1030	50.6
	Swin-B [15]	121	1188	49.7
	ConvNeXt-B [21]	122	1170	49.9
	RepLKNet-B [39]	112	1170	50.6
	VAN-B4	90	1098	**52.2**

Table 13 Comparision with the state-of-the-art methods on ADE20K validation set. Params means parameter. FLOPs denotes floating point operations. All models are pretrained on ImageNet-22K dataset. We calculate FLOPs with input size 2560 × 640 for 640 input image and 2048 × 512 for 512 input image

Method	Params. (M)	Input size	FLOPs (G)	mIoU (%)
Swin-B [15]	121	640^2	1841	51.7
ConvNeXt-B [21]	122	640^2	1828	53.1
VAN-B5	117	512^2	1208	**53.9**
Swin-L [15]	234	640^2	2468	53.5
ConvNeXt-L [21]	235	640^2	2458	53.7
VAN-B6	231	512^2	1658	**54.7**

4.4 Panoptic segmentation

Settings. We conduct our panoptic segmentation on COCO panoptic segmentation dataset [89] and choose Mask2Former [120] as our segmentation head. For fair comparison, we adopt the default settings in MMDetection [117] and the same training/validating scheme as Mask2Former [120]. All backbone models are pre-trained on ImageNet-1K or ImageNet-22K set. For fair comparsion, we follow the training settings of Mask2Former [120]. We choose AdamW with initial learning rate 0.0001 and weight decay. We adopt step learning rate schedule to adjust learning rate. The total epochs and batch size are set as 50 and 16 respectively. Besides, we adopt the same data augmentations with Mask2Former [120], including random scale, large-scale jittering, etc.

Results. As shown in Table 14, we observe that VAN outperforms Swin Transformer for both large and small models. Here, VAN-B2 exceeds Swin-T +1.7% PQ. Besides, it is worth noting that VAN-B6 achieves 58.2% PQ, which sets new state-of-the-art performance for panoptic segmentation task.

Table 14 Experimental results on COCO panoptic segmentation. * means that model is pretrained on ImageNet-22K dataset. All methods are based on Mask2Former [120]. PQ means panoptic quality

Backbone	Query type	Epochs	PQ (%)	PQ^{Th} (%)	PQ^{St} (%)
Swin-T	100 queries	50	53.2	59.3	44.0
VAN-B2	100 queries	50	54.9	61.2	45.3
Swin-L*	200 queries	50	57.8	64.2	48.1
VAN-B6*	200 queries	50	**58.2**	**64.8**	**48.2**

4.5 Pose estimation

Settings. We conduct pose estimation experiments on COCO human pose estimation dataset, which contains 200k images with 17 keypoints. Models are trained on COCO train 2017 set and tested on COCO val 2017 set. We adopt SimpleBaseline [121] as our decoder part, which is the same with Swin Transformer [15] and PVT [20]. All experiments are based on MMPose [122]. For fair comparison, we follow the training strategy of Swin Transformer. We adopt Adam as optimizer with initial learning rate 5×10^{-4}. We adopt step learning rate schedule to adjust learning rate. The total epochs and batch size are set as 210 and 64 respectively.

Results. Experimental results are shown in Table 15. For 256 × 192 input, VAN-B2 outperforms Swin-T and PVT-S [20] 2.5% AP (74.9% vs. 72.4%) and 3.5% AP (74.9% vs. 71.4%) and with simliar computing and parameters. Furthermore, VAN-B2 exceeds Swin-B 2% AP (74.9% vs. 72.9%) and 1.8% AP (76.7%

Table 15 Comparison with the state-of-the-art vision backbones on COCO benchmark for pose estimation. Models are based on SimpleBaseline [121]

Backbone	Input size	AP (%)	AP^{50} (%)	AP^{75} (%)	AR (%)	#P (M)	FLOPs (G)
HRNet-W32 [18]	256 × 192	74.4	90.5	81.9	78.9	28.5	7.1
PVT-S [20]	256 × 192	71.4	89.6	79.4	77.3	28.2	4.1
Swin-T [15]	256 × 192	72.4	90.1	80.6	78.2	32.8	6.1
Swin-B [15]	256 × 192	72.9	89.9	80.8	78.6	93.2	18.6
VAN-B2	256 × 192	**74.9**	**90.8**	**82.5**	**80.3**	30.3	6.1
HRNet-W32 [18]	384 × 288	75.8	90.6	82.7	81.0	28.5	16.0
Swin-B [15]	384 × 288	74.9	90.5	81.8	80.3	93.2	39.2
VAN-B2 [15]	384 × 288	**76.7**	**91.0**	**83.1**	**81.7**	30.3	13.6

vs. 74.9%) for 256 × 192 and 384 × 288 respectively with less computation and parameters. In addition to transformer-based models, VAN-B2 also surpasses popular CNN-based model HRNet-W32 [18].

4.6 Fine-grain classification

We conduct fine-grain classification on CUB-200 dataset [123], which is a common fine-grain classification benchmark and contains 11,788 images of 200 subcategories belonging to birds. We do not design specific algorithm for this task and only replace the last linear layer for 200 categories. We implement our model based on mmclassification [124]. For fine-grain classification, we choose AdamW as optimizer with initial learning rate 5×10^{-5} and weight decay 5×10^{-4}. We adopt cosine learning rate schedule to adjust learning rate. The batch size and total epochs are set as 16 and 100 respectively. Results in Table 16 show that VAN-B4 achieves 91.3% Top-1 accuracy without any specially designed algorithms, which exceeds DeiT [19] and ViT-B [13].

4.7 Saliency detection

We conduct saliency detection based on EDN [125]. We replace the backbone with VAN and hold experiments on common saliency detection benchmarks, including DUTS [126], DUT-O [127], and PASCAL-S [128]. All input images are resized to 384 × 384.

Table 16 Experimental results on CUB-200 fine-grain classification dataset. * means that model is pretrained on ImageNet-22K dataset

Method	Backbone	Top-1 Acc (%)
ResNet-50 [5]	ResNet-101	84.5
ViT [13]	ViT-B_16*	90.3
DeiT [19]	DeiT-B*	90.0
VAN	VAN-B4*	91.3

We train model by using Adam optimizer with initial learning rate of 5×10^{-5}. Step learning rate is adopted for adjusting learning rate. The batch size and epoch are set as 24 and 30 respectively. Results in Table 17 show that VAN clearly surpasses other backbones ResNet [5] and PVT [20] on all datasets.

Table 17 Comparing with different backbones on saliency detection task

Backbone	DUTS-TE		DUT-O		PASCAL-S	
	F_{max}	MAE	F_{max}	MAE	F_{max}	MAE
ResNet18 [5]	0.853	0.044	0.769	0.056	0.854	0.071
PVT-T [20]	0.876	0.039	0.813	0.052	0.868	0.067
VAN-B1	**0.912**	**0.030**	**0.835**	**0.046**	**0.893**	**0.055**
ResNet50 [5]	0.873	0.038	0.786	0.051	0.864	0.065
PVT-S [20]	0.900	0.032	0.832	0.050	0.883	0.060
VAN-B2	**0.919**	**0.028**	**0.844**	**0.045**	**0.897**	**0.053**

5 Discussion

Recently, transformer-based models quickly conquer various vision leaderboards. As we know that self-attention is just a special attention mechanism. However, people gradually adopt self-attention by default and ignore underlying attention methods. This paper proposes a novel attention module LKA and CNN-based network VAN, which surpasses state-of-the-art transformer-based methods for vision tasks. We hope this paper can promote people to rethink whether self-attention is irreplaceable and which kind of attention is more suitable for visual tasks.

6 Future work

In the future, we will continue perfecting VAN in followings directions:

- **Continuous improvement of the structure itself.** In this paper, we only demonstrate an intuitive structure. There are a lot of potential improvements such as adopting different kernel size, introducing multi-scale structure [11], and using multi-branch structure [10].
- **Large-scale self-supervised learning and transfer learning.** VAN naturally combines the advantages of CNNs and ViTs. On the one hand, VAN can make use of the 2D structure information of images. On the other hand, VAN can dynamically adjust the output according to the input image which is suit for self-supervised

learning and transfer learning [63, 68]. Combining the above two points, we believe that VAN can achieve better performance in image self-supervised learning and transfer learning field.

- **More application areas.** Due to the limited resource, we only show excellent performance in visual tasks. Whether VANs can perform well in other areas like TCN [129] in NLP is still worth exploring. Besides, for more complex 3D or video data, this decomposition idea of VAN may also be suitable. We look forward to seeing VANs becoming a general model for multiple modalities.

7 Conclusions

In this paper, we present a novel visual attention LKA which combines the advantages of convolution and self-attention. Based on LKA, we build a vision backbone VAN that achieves the state-of-the-art performance in some visual tasks, including image classification, object detection, semantic segmentation, etc. In the future, we will continue to improve this framework from the directions mentioned in Section 6.

Acknowledgements

This paper was supported by National Key R&D Program of China (Project No. 2021ZD0112902), the National Natural Science Foundation of China (Project No. 62220106003), and Tsinghua–Tencent Joint Laboratory for Internet Innovation Technology.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article. The author Ming-Ming Cheng is the Area Executive Editor of this journal, and the author Shi-Min Hu is the Editor-in-Chief of this journal.

References

[1] LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* Vol. 86, No. 11, 2278–2324, 1998.

[2] LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation* Vol. 1, No. 4, 541–551, 1989.

[3] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM* Vol. 60, No. 6, 84–90, 2017.

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

[5] He, K. M.; Zhang, X. Y.; Ren, S. Q.; Sun, J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778, 2016.

[6] Howard, A. G.; Zhu, M. L.; Chen, B.; Kalenichenko, D.; Wang, W. J.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint* arXiv:1704.04861, 2017.

[7] Xie, S. N.; Girshick, R.; Dollár, P.; Tu, Z. W.; He, K. M. Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5987–5995, 2017.

[8] Zhang, X. Y.; Zhou, X. Y.; Lin, M. X.; Sun, J. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6848–6856, 2018.

[9] Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K. Q. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2261–2269, 2017.

[10] Szegedy, C.; Liu, W.; Jia, Y. Q.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1–9, 2015.

[11] Gao, S. H.; Cheng, M. M.; Zhao, K.; Zhang, X. Y.; Yang, M. H.; Torr, P. Res2Net: A new multi-scale backbone architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 43, No. 2, 652–662, 2021.

[12] Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7132–7141, 2018.

[13] Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X. H.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. In: Proceedings of the

International Conference on Learning Representations, 2020.

[14] Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. OverFeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint* arXiv:1312.6229, 2013.

[15] Liu, Z.; Lin, Y. T.; Cao, Y.; Hu, H.; Wei, Y. X.; Zhang, Z.; Lin, S.; Guo, B. N. Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 9992–10002, 2021.

[16] Dai, J. F.; Qi, H. Z.; Xiong, Y. W.; Li, Y.; Zhang, G. D.; Hu, H.; Wei, Y. C. Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, 764–773, 2017.

[17] Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; Luo, P. Segformer: Simple and efficient design for semantic segmentation with transformers. In: Proceedings of the 35th Conference on Neural Information Processing Systems, 2021.

[18] Wang, J. D.; Sun, K.; Cheng, T. H.; Jiang, B. R.; Deng, C. R.; Zhao, Y.; Liu, D.; Mu, Y. D.; Tan, M. K.; Wang, X. G.; et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 43, No. 10, 3349–3364, 2021.

[19] Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. In: Proceedings of the 38th International Conference on Machine Learning, 10347–10357, 2021.

[20] Wang, W. H.; Xie, E. Z.; Li, X.; Fan, D. P.; Song, K. T.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 548–558, 2021.

[21] Liu, Z.; Mao, H. Z.; Wu, C. Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. N. A ConvNet for the 2020s. *arXiv preprint* arXiv:2201.03545, 2022.

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

[23] Gottlieb, J. P.; Kusunoki, M.; Goldberg, M. E. The representation of visual salience in monkey parietal cortex. *Nature* Vol. 391, No. 6666, 481–484, 1998.

[24] Treisman, A. M.; Gelade, G. A feature-integration theory of attention. *Cognitive Psychology* Vol. 12, No. 1, 97–136, 1980.

[25] Wolfe, J. M.; Horowitz, T. S. What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience* Vol. 5, No. 6, 495–501, 2004.

[26] Tsotsos, J. K.; Culhane, S. M.; Kei Wai, W. Y.; Lai, Y. Z.; Davis, N.; Nuflo, F. Modeling visual attention via selective tuning. *Artificial Intelligence* Vol. 78, Nos. 1–2, 507–545, 1995.

[27] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł; Polosukhin, I. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 6000–6010, 2017.

[28] Devlin, J.; Chang, M. W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint* arXiv:1810.04805, 2018.

[29] Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, Article No. 159, 1877–1901, 2020.

[30] Woo, S.; Park, J.; Lee, J. Y.; Kweon, I. S. CBAM: Convolutional block attention module. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11211*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 3–19, 2018.

[31] Wang, Q. L.; Wu, B. G.; Zhu, P. F.; Li, P. H.; Zuo, W. M.; Hu, Q. H. ECA-net: Efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11531–11539, 2020.

[32] El-Nouby, A.; Touvron, H.; Caron, M.; Bojanowski, P.; Douze, M.; Joulin, A.; Laptev, I.; Neverova, N.; Synnaeve, G.; Verbeek, J.; et al. XCiT: Cross-covariance image transformers. In: Proceedings of the 35th Conference on Neural Information Processing Systems, 2021.

[33] Han, Q.; Fan, Z.; Dai, Q.; Sun, L.; Cheng, M.-M.; Liu, J.; Wang, J. Demystifying local vision transformer: Sparse connectivity, weight sharing, and dynamic weight. *arXiv preprint* arXiv:2106.04263, 2021.

[34] Bello, I.; Fedus, W.; Du, X.; Cubuk, E. D.; Srinivas, A.; Lin, T.-Y.; Shlens, J.; Zoph, B. Revisiting ResNets: Improved training and scaling strategies. In: Proceedings of the 35th Conference on Neural Information Processing Systems, 2021.

[35] Sandler, M.; Howard, A.; Zhu, M. L.; Zhmoginov, A.; Chen, L. C. MobileNetV2: Inverted residuals

and linear bottlenecks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4510–4520, 2018.

[36] Lin, M.; Chen, Q.; Yan, S. Network in network. In: Proceedings of the International Conference on Learning Representations, 2014.

[37] Chen, L. C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A. L. Semantic image segmentation with deep convolutional nets and fully connected CRFs. *arXiv preprint* arXiv:1412.7062, 2014.

[38] Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv preprint* arXiv:1511.07122, 2015.

[39] Ding, X. H.; Zhang, X. Y.; Han, J. G.; Ding, G. G. Scaling up your kernels to 31 × 31: Revisiting large kernel design in CNNs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11953–11965, 2022.

[40] Mnih, V.; Heess, N.; Graves, A.; Kavukcuoglu, K. Recurrent models of visual attention. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, Vol. 2, 2204–2212, 2014.

[41] Hu, H.; Gu, J. Y.; Zhang, Z.; Dai, J. F.; Wei, Y. C. Relation networks for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3588–3597, 2018.

[42] Yuan, Y. H.; Chen, X. L.; Wang, J. D. Object-contextual representations for semantic segmentation. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12351*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 173–190, 2020.

[43] Geng, Z. Y.; Guo, M. H.; Chen, H. X.; Li, X.; Wei, K.; Lin, Z. C. Is attention better than matrix decomposition? In: Proceedings of the International Conference on Learning Representations, 2021.

[44] Guo, M. H.; Xu, T. X.; Liu, J. J.; Liu, Z. N.; Jiang, P. T.; Mu, T. J.; Zhang, S. H.; Martin, R. R.; Cheng, M. M.; Hu, S. M. Attention mechanisms in computer vision: A survey. *Computational Visual Media* Vol. 8, No. 3, 331–368, 2022.

[45] Xu, Y. F.; Wei, H. P.; Lin, M. X.; Deng, Y. Y.; Sheng, K. K.; Zhang, M. D.; Tang, F.; Dong, W. M.; Huang, F. Y.; Xu, C. S. Transformers in computational visual media: A survey. *Computational Visual Media* Vol. 8, No. 1, 33–62, 2022.

[46] Wang, X. L.; Girshick, R.; Gupta, A.; He, K. M. Non-local neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7794–7803, 2018.

[47] Fu, J.; Liu, J.; Tian, H. J.; Li, Y.; Bao, Y. J.; Fang, Z. W.; Lu, H. Q. Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3141–3149, 2019.

[48] Ramachandran, P.; Parmar, N.; Vaswani, A.; Bello, I.; Levskaya, A.; Shlens, J. Stand-alone self-attention in vision models. *arXiv preprint* arXiv:1906.05909, 2019.

[49] Bello, I.; Zoph, B.; Le, Q.; Vaswani, A.; Shlens, J. Attention augmented convolutional networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 3285–3294, 2019.

[50] Yuan, Y. H.; Huang, L.; Guo, J. Y.; Zhang, C.; Chen, X. L.; Wang, J. D. OCNet: Object context network for scene parsing. *arXiv preprint* arXiv:1809.00916, 2018.

[51] Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-attention generative adversarial networks. In: Proceedings of the 36th International Conference on Machine Learning, 7354–7363, 2019.

[52] Xie, S. N.; Liu, S. N.; Chen, Z. Y.; Tu, Z. W. Attentional ShapeContextNet for point cloud recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4606–4615, 2018.

[53] Huang, Z. L.; Wang, X. G.; Huang, L. C.; Huang, C.; Wei, Y. C.; Liu, W. Y. CCNet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 603–612, 2019.

[54] Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12346*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 213–229, 2020.

[55] Guo, M. H.; Cai, J. X.; Liu, Z. N.; Mu, T. J.; Martin, R. R.; Hu, S. M. PCT: Point cloud transformer. *Computational Visual Media* Vol. 7, No. 2, 187–199, 2021.

[56] Srinivas, A.; Lin, T. Y.; Parmar, N.; Shlens, J.; Abbeel, P.; Vaswani, A. Bottleneck transformers for visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 16514–16524, 2021.

[57] Yuan, L.; Chen, Y. P.; Wang, T.; Yu, W. H.; Shi, Y. J.; Jiang, Z. H.; Tay, F. E. H.; Feng, J. S.; Yan, S. C. Tokens-to-token ViT: Training vision transformers from scratch on ImageNet. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 538–547, 2021.

[58] Liu, R.; Deng, H. M.; Huang, Y. Y.; Shi, X. Y.; Lu, L. W.; Sun, W. X.; Wang, X. G.; Dai, J. F.; Li, H. S. Decoupled spatial-temporal transformer for video inpainting. *arXiv preprint* arXiv:2104.06637, 2021.

[59] Bello, I. LambdaNetworks: Modeling long-range interactions without attention. In: Proceedings of the International Conference on Learning Representations, 2021.

[60] Xu, Y.; Zhang, Q.; Zhang, J.; Tao, D. ViTAE: Vision transformer advanced by exploring intrinsic inductive bias. In: Proceedings of the 35th Conference on Neural Information Processing Systems, 2021.

[61] Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; Wang, Y. Transformer in transformer. In: Proceedings of the 35th Conference on Neural Information Processing Systems, 2021.

[62] Liu, R.; Deng, H. M.; Huang, Y. Y.; Shi, X. Y.; Lu, L. W.; Sun, W. X.; Wang, X. G.; Dai, J. F.; Li, H. S. FuseFormer: Fusing fine-grained information in transformers for video inpainting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 14020–14029, 2021.

[63] Bao, H. B.; Dong, L.; Piao, S. H.; Wei, F. R. BEiT: BERT pre-training of image transformers. In: Proceedings of the International Conference on Learning Representations, 2022.

[64] Liu, S. L.; Li, F.; Zhang, H.; Yang, X.; Qi, X. B.; Su, H.; Zhu, J.; Zhang, L. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In: Proceedings of the International Conference on Learning Representations, 2022.

[65] Wu, H. P.; Xiao, B.; Codella, N.; Liu, M. C.; Dai, X. Y.; Yuan, L.; Zhang, L. CvT: Introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 22–31, 2021.

[66] Liu, S. L.; Zhang, L.; Yang, X.; Su, H.; Zhu, J. Query2Label: A simple transformer way to multi-label classification. *arXiv preprint* arXiv:2107.10834, 2021.

[67] Wu, Y. H.; Liu, Y.; Zhan, X.; Cheng, M. M. P2T: Pyramid pooling transformer for scene understanding. *arXiv preprint* arXiv:2106.12011, 2021.

[68] He, K. M.; Chen, X. L.; Xie, S. N.; Li, Y. H.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 15979–15988, 2022.

[69] Xu, W. J.; Xu, Y. F.; Chang, T.; Tu, Z. W. Co-scale conv-attentional image transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 9961–9970, 2021.

[70] Chen, Q.; Wu, Q. M.; Wang, J.; Hu, Q. H.; Hu, T.; Ding, E. R.; Cheng, J.; Wang, J. D. MixFormer: Mixing features across windows and dimensions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5239–5249, 2022.

[71] Chen, L.; Zhang, H. W.; Xiao, J.; Nie, L. Q.; Shao, J.; Liu, W.; Chua, T. S. SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6298–6306, 2017.

[72] Qin, Z. Q.; Zhang, P. Y.; Wu, F.; Li, X. FcaNet: frequency channel attention networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 763–772, 2021.

[73] Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* Vol. 65, No. 6, 386–408, 1958.

[74] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning internal representations by error propagation. Technical Report. Institute for Cognitive Science, University of California, San Diego, 1985.

[75] Tolstikhin, I. O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. MLP-Mixer: An all-MLP architecture for vision. In: Proceedings of the 35th Conference on Neural Information Processing Systems, 2021.

[76] Guo, M. H.; Liu, Z. N.; Mu, T. J.; Hu, S. M. Beyond self-attention: External attention using two linear layers for visual tasks. *arXiv preprint* arXiv:2105.02358, 2021.

[77] Touvron, H.; Bojanowski, P.; Caron, M.; Cord, M.; El-Nouby, A.; Grave, E.; Izacard, G.; Joulin, A.; Synnaeve, G.; Verbeek, J.; et al. ResMLP: Feedforward networks for image classification with data-efficient training. *arXiv preprint* arXiv:2105.03404, 2021.

[78] Liu, H.; Dai, Z.; So, D.; Le, Q. V. Pay attention to MLPs. In: Proceedings of the 35th Conference on Neural Information Processing Systems, 2021.

[79] Guo, M. H.; Liu, Z. N.; Mu, T. J.; Liang, D.; Martin, R. R.; Hu, S. M. Can attention enable MLPs to catch up with CNNs? *Computational Visual Media* Vol. 7, No. 3, 283–288, 2021.

[80] Liu, R.; Li, Y.; Tao, L.; Liang, D.; Zheng, H. T. Are we ready for a new paradigm shift? A survey on visual deep MLP. *Patterns* Vol. 3, No. 7, 100520, 2022.

[81] Wang, F.; Jiang, M. Q.; Qian, C.; Yang, S.; Li, C.; Zhang, H. G.; Wang, X. G.; Tang, X. O.

Residual attention network for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6450–6458, 2017.

[82] Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Vedaldi, A. Gather-excite: Exploiting feature context in convolutional neural networks. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 9423–9433, 2018.

[83] Park, J.; Woo, S.; Lee, J. Y.; Kweon, I. S. Bam: Bottleneck attention module. *arXiv preprint* arXiv:1807.06514, 2018.

[84] Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning, Vol. 1, 448–456, 2015.

[85] Hendrycks, D.; Gimpel, K. Gaussian error linear units (GELUs). *arXiv preprint* arXiv:1606.08415, 2016.

[86] Wang, W. H.; Xie, E. Z.; Li, X.; Fan, D. P.; Song, K. T.; Liang, D.; Lu, T.; Luo, P.; Shao, L. PVT v2: Improved baselines with pyramid vision transformer. *arXiv preprint* arXiv:2106.13797, 2021.

[87] Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, 618–626, 2017.

[88] Deng, J.; Dong, W.; Socher, R.; Li, L. J.; Kai, L.; Li, F. F. ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 248–255, 2009.

[89] Lin, T. Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. L. Microsoft COCO: Common objects in context. In: *Computer Vision – ECCV 2014. Lecture Notes in Computer Science, Vol. 8693*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 740–755, 2014.

[90] Zhou, B. L.; Zhao, H.; Puig, X.; Xiao, T. T.; Fidler, S.; Barriuso, A.; Torralba, A. Semantic understanding of scenes through the ADE20K dataset. *International Journal of Computer Vision* Vol. 127, No. 3, 302–321, 2019.

[91] Zhou, B. L.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2921–2929, 2016.

[92] Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An imperative style, high-performance deep learning library. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Article No. 721, 8026–8037, 2019.

[93] Hu, S. M.; Liang, D.; Yang, G. Y.; Yang, G. W.; Zhou, W. Y. Jittor: A novel deep learning framework with meta-operators and unified graph execution. *Science China Information Sciences* Vol. 63, No. 12, 222103, 2020.

[94] Müller, R.; Kornblith, S.; Hinton, G. E. When does label smoothing help? In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Article No. 422, 4694–4703, 2019.

[95] Zhang, H. Y.; Cisse, M.; Dauphin, Y. N.; Lopez-Paz, D. Mixup: Beyond empirical risk minimization. *arXiv preprint* arXiv:1710.09412, 2017.

[96] Yun, S.; Han, D.; Chun, S.; Oh, S. J.; Yoo, Y.; Choe, J. CutMix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 6022–6031, 2019.

[97] Zhong, Z.; Zheng, L.; Kang, G. L.; Li, S. Z.; Yang, Y. Random erasing data augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 34, No. 7, 13001–13008, 2020.

[98] Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint* arXiv:1412.6980, 2014.

[99] Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv preprint* arXiv:1711.05101, 2017.

[100] Loshchilov, I.; Hutter, F. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint* arXiv:1608.03983, 2016.

[101] Touvron, H.; Cord, M.; Sablayrolles, A.; Synnaeve, G.; Jégou, H. Going deeper with image transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 32–42, 2021.

[102] Polyak, B. T.; Juditsky, A. B. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization* Vol. 30, No. 4, 838–855, 1992.

[103] Dai, Z.; Liu, H.; Le, Q.; Tan, M. CoAtNet: Marrying convolution and attention for all data sizes. In: Proceedings of the 35th Conference on Neural Information Processing Systems, 2021.

[104] Tan, M.; Le, Q. Efficientnetv2: Smaller models and faster training. In: Proceedings of the 38th International Conference on Machine Learning, 10096–10106, 2021.

[105] Yang, J.; Li, C.; Dai, X.; Gao, J. Focal modulation networks. In: Proceedings of the 36th Conference on Neural Information Processing Systems, 2022.

[106] Luo, W.; Li, Y.; Urtasun, R.; Zemel, R. Understanding the effective receptive field in deep convolutional neural networks. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, 4905–4913, 2016.

[107] Lin, T. Y.; Goyal, P.; Girshick, R.; He, K. M.; Dollár, P. Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, 2999–3007, 2017.

[108] Yu, W. H.; Luo, M.; Zhou, P.; Si, C. Y.; Zhou, Y. C.; Wang, X. C.; Feng, J. S.; Yan, S. C. MetaFormer is actually what You need for vision. *arXiv preprint* arXiv:2111.11418, 2021.

[109] Guo, J. Y.; Han, K.; Wu, H.; Tang, Y. H.; Chen, X. H.; Wang, Y. H.; Xu, C. CMT: Convolutional neural networks meet vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12165–12175, 2022.

[110] He, K. M.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, 2980–2988, 2017.

[111] Lee, Y.; Kim, J.; Willette, J.; Hwang, S. J. MPViT: Multi-path vision transformer for dense prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7277–7286, 2022.

[112] Cai, Z. W.; Vasconcelos, N. Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 43, No. 5, 1483–1498, 2021.

[113] Zhang, S. F.; Chi, C.; Yao, Y. Q.; Lei, Z.; Li, S. Z. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9756–9765, 2020.

[114] Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, Article No. 1763, 21002–21012, 2020.

[115] Kirillov, A.; Girshick, R.; He, K. M.; Dollár, P. Panoptic feature pyramid networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6392–6401, 2019.

[116] Xiao, T. T.; Liu, Y. C.; Zhou, B. L.; Jiang, Y. N.; Sun, J. Unified perceptual parsing for scene understanding. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11209*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 432–448, 2018.

[117] Chen, K.; Wang, J. Q.; Pang, J. M.; Cao, Y. H.; Xiong, Y.; Li, X. X.; Sun, S. Y.; Feng, W. S.; Liu, Z. W.; Xu, J. R.; et al. MMDetection: Open MMLab detection toolbox and benchmark. *arXiv preprint* arXiv:1906.07155, 2019.

[118] Sun, P. Z.; Zhang, R. F.; Jiang, Y.; Kong, T.; Xu, C. F.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z. H.; Wang, C. H.; et al. Sparse R-CNN: End-to-end object detection with learnable proposals. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 14449–14458, 2021.

[119] MMSegmentation: OpenMMLab semantic segmentation toolbox and benchmark. 2020. Available at https://github.com/open-mmlab/mmsegmentation.

[120] Cheng, B. W.; Misra, I.; Schwing, A. G.; Kirillov, A.; Girdhar, R. Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1280–1289, 2022.

[121] Xiao, B.; Wu, H. P.; Wei, Y. C. Simple baselines for human pose estimation and tracking. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11210*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 472–487, 2018.

[122] OpenMMLab pose estimation toolbox and benchmark. 2020. Available at https://github.com/open-mmlab/mmpose.

[123] Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; Perona, P. Caltech-UCSD Birds 200. Computation & Neural Systems Technical Report 2010-001. California Institute of Technology, 2010.

[124] OpenMMLab pre-training toolbox and benchmark. 2020. Available at https://github.com/open-mmlab/mmclassification.

[125] Wu, Y. H.; Liu, Y.; Zhang, L.; Cheng, M. M.; Ren, B. EDN: Salient object detection via extremely-downsampled network. *IEEE Transactions on Image Processing* Vol. 31, 3125–3136, 2022.

[126] Wang, L. J.; Lu, H. C.; Wang, Y. F.; Feng, M. Y.; Wang, D.; Yin, B. C.; Ruan, X. Learning to detect salient objects with image-level supervision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3796–3805, 2017.

[127] Yang, C.; Zhang, L. H.; Lu, H. C.; Ruan, X.; Yang, M. H. Saliency detection via graph-based manifold ranking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3166–3173, 2013.

[128] Li, Y.; Hou, X. D.; Koch, C.; Rehg, J. M.; Yuille, A. L. The secrets of salient object segmentation. In:

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 280–287, 2014.

[129] Bai, S. J.; Kolter, J. Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint* arXiv:1803.01271, 2018.

Meng-Hao Guo is a Ph.D. candidate under the supervision of Prof. Shi-Min Hu in the Department of Computer Science and Technology, Tsinghua University. His research interests include conputer vision and computer graphics.

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. His research interests include deep learning and computer vision.

Zheng-Ning Liu received his bachelor degree and Ph.D. degree in computer science from Tsinghua University in 2017 and 2022 respectively. He is currently a research scientist in Fitten Technology Co., Ltd. His research interests include computer vision, 3D reconstruction, and computer graphics. He has published papers in some journals and conferences such as IEEE TPAMI, ACM TOG, ECCV, etc.

Ming-Ming Cheng received his Ph.D. degree from Tsinghua University in 2012, and then worked with Prof. Philip Torr in Oxford for 2 years. He is now a professor at Nankai University, leading the Media Computing Lab. His research interests include computer vision and computer graphics. He received awards including ACM China Rising Star Award, IBM Global SUR Award, etc. He is a senior member of the IEEE and on the editorial boards of IEEE TPAMI and IEEE TIP.

Shi-Min Hu is currently a professor in computer science at Tsinghua University. He received his Ph.D. degree from Zhejiang University in 1996. His research interests include geometry processing, image & video processing, rendering, computer animation, and CAD. He has published more than 100 papers in journals and refereed conferences. He is the Editor-in-Chief of *Computational Visual Media*, and on the editorial boards of several journals, including *Computer Aided Design* and *Computer & Graphics*.