arXiv:2202.09741v5 [cs.CV] 11 Jul 2022

Visual Attention Network

Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng and Shi-Min Hu

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 surpasses similar size vision transformers(ViTs) and convolutional neural networks(CNNs) 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 set 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. Code is available at https://github.com/Visual-Attention-Network.

Index Terms—Attention, Vision Backbone, Deep Learning, ConvNets.

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], [2], [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], [7], [8], stronger multi-scale ability [9], [10], [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 results 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 times and estimated signal transmission times 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] been proposed, self-attention models (*a.k.a.,* Transformer) quickly becomes the dominated architecture [28], [29] in natural language processing (NLP). Recently, Dosovitskiy *et al.* [13] propose the vision transformer (ViT), which introduces transformer backbone into computer vision and outperforms well-known CNNs on image classification tasks. Benefited from its 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], [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], [31], [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. Based on LKA, we further introduce a simple vision backbone called VAN.
- We show that VANs outperform the similar level ViTs and CNNs in extensive experiments on various tasks, including image classification, object detection, semantic segmentation, instance segmentation, pose estimation, *etc*.

- *M.-H. Guo and S.-M. Hu are with the Department of Computer Science, Tsinghua University, Beijing, China. Emails:gmh20@mails.tsinghua.edu.cn, shimin@tsinghua.edu.cn. Shi-Min Hu is the corresponding author.*
- *C.-Z. Lu and M.-M. Cheng are with Nankai University University, Tianjin, China. Emails: czlu919@outlook.com, cmm@nankai.edu.cn.*
- *Z.-N. Liu are with Fitten Tech, Beijing, China. Emails: lzhengning@gmail.com.*

Manuscript received April 19, 2005; revised August 26, 2015.

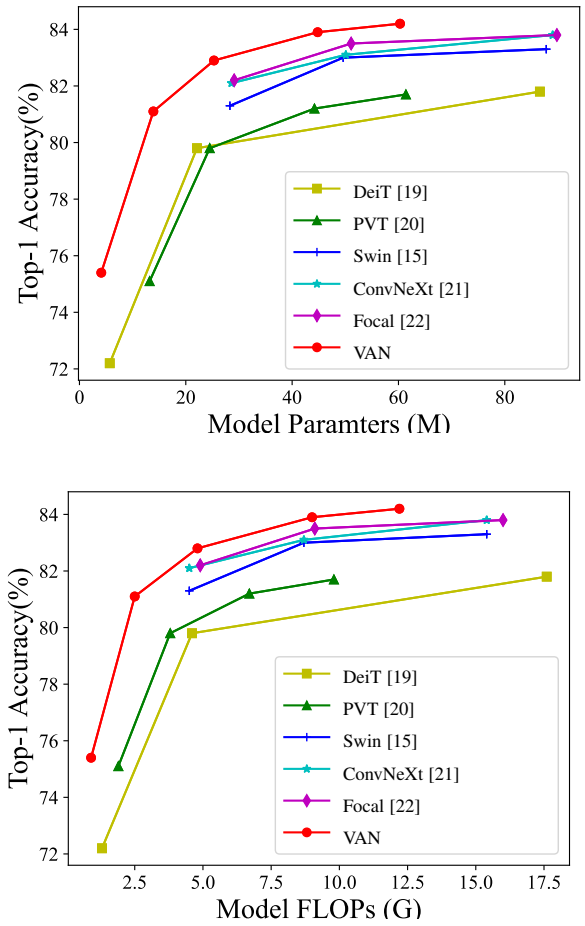


Fig. 1. Results of different models on ImageNet-1K validation set. Comparing the performance of recent models DeiT [19], PVT [20], Swin Transformer [15], ConvNeXt [21], Focal Transformer [22] and our VAN. Above: Accuracy-Parameters trade-off diagram. Under: Accuracy-FLOPs trade-off diagram.

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 (*a.k.a.,* 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.

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 [39]. It has provided benefits in many visual tasks, such as image classification [12], [30], object detection [16], [40] and semantic segmentation [41], [42]. Attention in computer vision can be divided into four basic categories [43], 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 [44], [45], [46], [47], [48], [49], [50]. Various deep self-attention networks (*a.k.a.,* vision transformers) [13], [15], [20], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64] 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 [43], [65]. Channel adaptability is also proven important for visual tasks [12], [30], [31], [65], [66]. 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) [67], [68] 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 [69], [70], [71], [72]. Such decomposition allows significant computational cost and parameters reduction, which release the amazing performance of MLP. Readers are referred to recent surveys [73], [74] for a more comprehensive review of MLPs. The most related MLP to our method is the gMLP [72], 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 ignore their local structure. Our method can make full use of its advantages and avoid its shortcomings.

3 Method

3.1 Large Kernel Attention

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

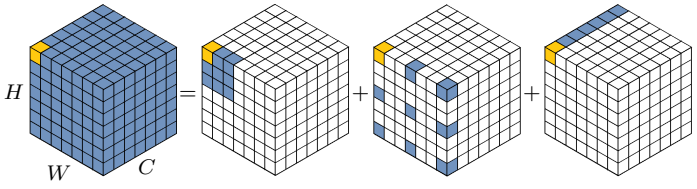


Fig. 2. Decomposition diagram of large-kernel convolution. A standard convolution can be decomposed into three parts: a depth-wise convolution (DW-Conv), a depth-wise dilation convolution (DW-D-Conv), and a pointwise convolution (1×1 Conv). The colored grids represent the location of convolution kernel 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 above figure.

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], [44], [48], [49] to capture long-range dependence. There are three obvious shortcomings for self-attention applied in computer vision which have been listed in Sec. 2.2. The second one is to use large kernel convolution [30], [75], [76], [77] 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 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 Tab. 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 Tab. 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 [43], [65] and adaptability in the channel dimension is also important for visual tasks.

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

Properties	Convolution	Self-Attention	LKA
Local Receptive Field	✓	✗	✓
Long-range Dependence	✗	✓	✓
Spatial Adaptability	✗	✓	✓
Channel Adaptability	✗	✗	✓
Computational complexity	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$

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 $133\times$ and $4.5\times$ more parameters than our decomposition respectively.

	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

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 $\frac{H}{32} \times \frac{W}{32}$ respectively. Here, H and W denote the height and width of the input image. With the decreasing of resolution, the number of output channels is increasing. The change of output channel C_i is presented in Tab. 5.

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 [78], 1×1 Conv, GELU activation [79], large kernel attention and feed-forward network (FFN) [80] 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 Tab. 5.

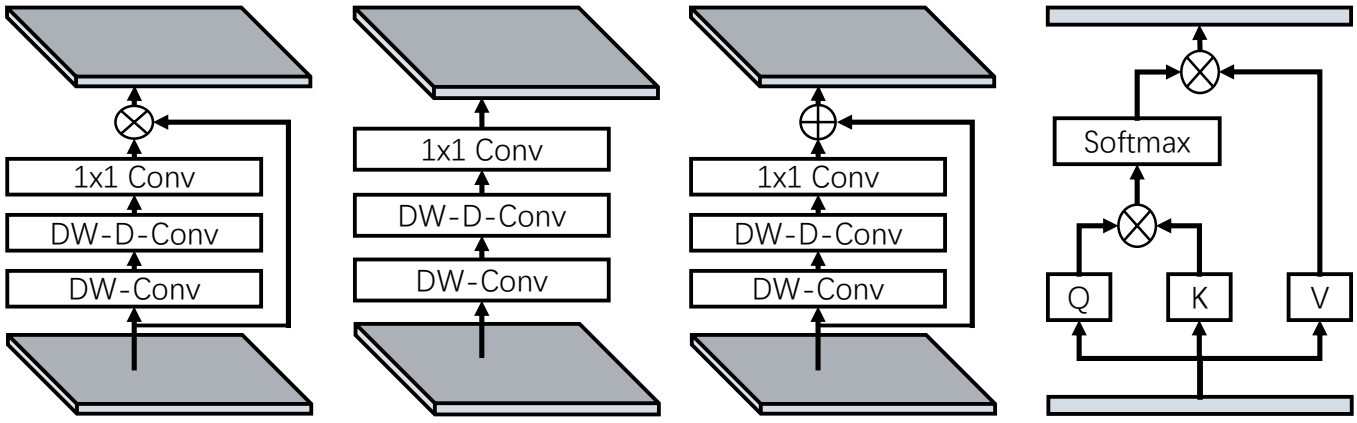
Complexity analysis. We present the parameters and floating point operations (FLOPs) of our decomposition. Bias is omitted in the computation process for simplifying format. We assume that the input and output features have same size $H \times W \times C$. The number of parameters $P(K, d)$ and FLOPs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

$$P(K, d) = C(\lceil \frac{K}{d} \rceil^2 \times C + (2d-1)^2) + C^2, \quad (3)$$

$$F(K, d) = P(K, d) \times H \times W. \quad (4)$$

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

Implementation details. We adopt K = 21 by default. For $K = 21$, the Equ. (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



(a) LKA (b) Non-Attention (c) Non-Attention(Add) (d) Self-Attention

Fig. 3. The 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.

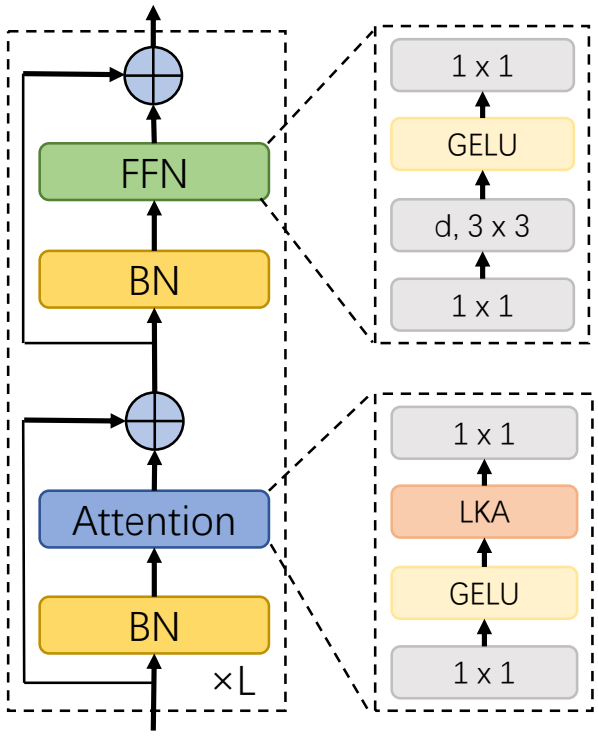


Fig. 4. A stage of VAN. d means depth wise convolution. k × k denotes k × k convolution.

channels, we show the specific parameters in Tab. 2. It shows that our decomposition owns significant advantages in decomposing large kernel convolution in terms of parameters and FLOPs.

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 [81] and ImageNet-22K image classification dataset, COCO [82] benchmark for object detection, instance segmentation, panoptic segmentation and pose estimation, and ADE20K [83] semantic segmentation dataset. Furthermore, we visualize the experimental results and class activation mapping(CAM) [84] by using Grad-CAM [85] on ImageNet validation set. Experiments are based on Pytorch [86] and Jittor [87].

TABLE 3
Ablation study of different modules in LKA. Top-1 accuracy (Acc) on ImageNet validation set suggest 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
VAN-B0	4.1	0.9	75.4

TABLE 4
Throughput of Swin transformer and VAN on RTX 3090.

Method	FLOPs(G)	Throughput (Imgs/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

4.1 Image Classification

4.1.1 *ImageNet-1K Experiments*

Settings. We conduct image classification on ImageNet-1K [81] dataset. It contains 1.28M training images and 50K validation images from 1,000 different categories. The whole training scheme mostly follows [19]. We adopt random clipping, random horizontal flipping, label-smoothing [88], mixup [89], cutmix [90] and random erasing [91] to augment the training data. In the training process, we train our VAN for 300 epochs by using AdamW [92], [93] optimizer with momentum=0.9, weight decay=5×10^{-2} and batch

TABLE 5
The 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

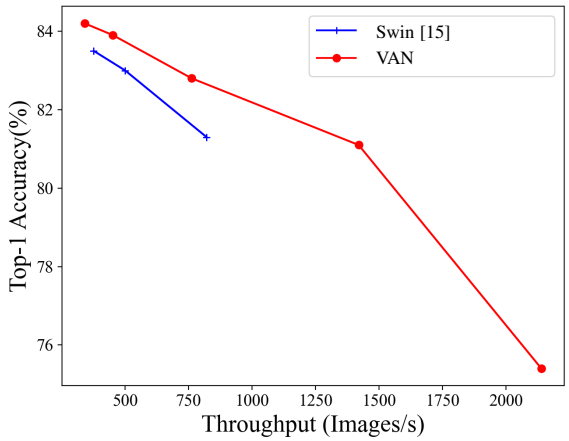


Fig. 5. Accuracy-Throughput Diagram. It claerly shows that VAN achieves a better trade-off than swin transformer [15].

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)	GFLOPs	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

size = 1,024. Cosine schedule [94] 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 [95] in attention layer which replaces $x_{out} = x + diag(\lambda_1, \lambda_2, ..., \lambda_d)f(x)$ with $x_{out} = x + diag(\lambda_1, \lambda_2, ..., \lambda_d)(f(x) + x)$ with initial value 0.01 and achieves a better performance than original LayerScale. Exponential moving average (EMA) [96] 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 the Tab. 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

TABLE 7
Compare with the state-of-the-art methods on ImageNet validation set. Params means parameter. GFLOPs denotes floating point operations. Top-1 Acc represents Top-1 accuracy.FLOPs is

Method	Params. (M)	GFLOPs	Top-1 Acc (%)
PVTv2-B0 [80]	3.4	0.6	70.5
T2T-ViT-7 [54]	4.3	1.1	71.7
DeiT-Tiny/16 [19]	5.7	1.3	72.2
TNT-Ti [97]	6.1	1.4	73.9
VAN-B0	4.1	0.9	**75.4**
ResNet18 [5]	11.7	1.8	69.8
PVT-Tiny [20]	13.2	1.9	75.1
PoolFormer-S12 [98]	11.9	2.0	77.2
PVTv2-B1 [80]	13.1	2.1	78.7
VAN-B1	13.9	2.5	**81.1**
ResNet50 [5]	25.6	4.1	76.5
ResNeXt50-32x4d [7]	25.0	4.3	77.6
RegNetY-4G [99]	21.0	4.0	80.0
DeiT-Small/16 [19]	22.1	4.6	79.8
T2T-ViT$_t$-14 [54]	21.5	6.1	81.7
PVT-Small [20]	24.5	3.8	79.8
TNT-S [97]	23.8	5.2	81.3
ResMLP-24 [71]	30.0	6.0	79.4
gMLP-S [72]	20.0	4.5	79.6
Swin-T [15]	28.3	4.5	81.3
PoolFormer-S24 [98]	21.4	3.6	80.3
Twins-SVT-S [100]	24.0	2.8	81.7
PVTv2-B2 [80]	25.4	4.0	82.0
Focal-T [22]	29.1	4.9	82.2
ConvNeXt-T [21]	28.6	4.5	82.1
VAN-B2	26.6	5.0	**82.8**
ResNet101 [5]	44.7	7.9	77.4
ResNeXt101-32x4d [7]	44.2	8.0	78.8
Mixer-B/16 [69]	59.0	11.6	76.4
T2T-ViT$_t$-19 [54]	39.2	9.8	82.4
PVT-Medium [20]	44.2	6.7	81.2
Swin-S [15]	49.6	8.7	83.0
ConvNeXt-S [15]	50.1	8.7	83.1
PVTv2-B3 [80]	45.2	6.9	83.2
Focal-S [22]	51.1	9.1	83.5
VAN-B3	44.8	9.0	**83.9**
ResNet152 [5]	60.2	11.6	78.3
T2T-ViT$_t$-24 [54]	64.0	15.0	82.3
PVT-Large [20]	61.4	9.8	81.7
TNT-B [97]	66.0	14.1	82.8
PVTv2-B4 [80]	62.6	10.1	83.6
VAN-B4	60.3	12.2	**84.2**

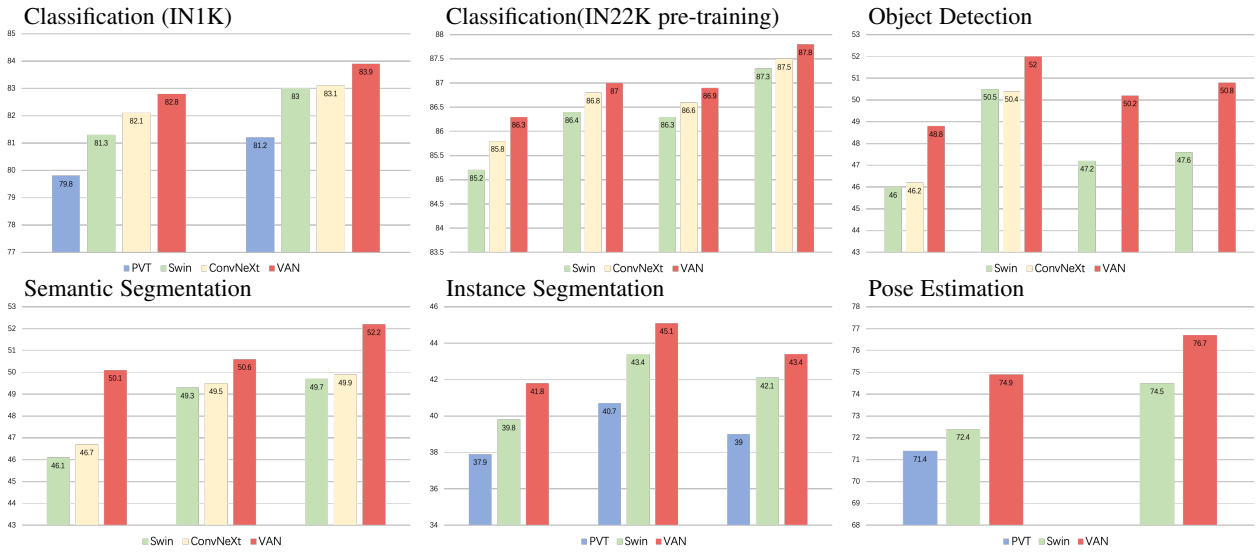


Fig. 6. Comparing with similar level PVT [20], Swin Transformer [15] and ConvNeXt [21] on various tasks, including image classification, object detection, semantic segmentation, instance segmentation and pose estimation.

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

Method	Params. (M)	Input size	GFLOPs	Top-1 Acc (%)
Swin-S [15]	50	224^2	8.7	83.2
ConvNeXt-S [21]	50	224^2	8.7	84.6
VAN-B4	60	224^2	12.2	**85.7**
ConvNeXt-S [21]	50	384^2	25.5	85.8
VAN-B4	60	384^2	35.9	**86.6**
Swin-B [15]	88	224^2	15.4	85.2
ConvNeXt-B [21]	89	224^2	15.4	85.8
VAN-B5	90	224^2	17.2	**86.3**
EffNetV2-L [101]	120	480^2	53.0	86.8
ViT-B/16 [13]	87	384^2	55.5	85.4
Swin-B [15]	88	384^2	47.0	86.4
ConvNeXt-B [21]	89	384^2	45.1	86.8
VAN-B5	90	384^2	50.6	**87.0**
Swin-L [15]	197	224^2	34.5	86.3
ConvNeXt-L [21]	198	224^2	34.4	86.6
VAN-B6	200	224^2	38.9	**86.9**
EffNetV2-XL [101]	208	480^2	94.0	87.3
CoAtNet-3 [102]	168	384^2	107.4	87.6
Swin-L [15]	197	384^2	103.9	87.3
ConvNeXt-L [21]	198	384^2	101.0	87.5
VAN-B6	200	384^2	114.3	**87.8**

confirms our viewpoint of 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 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 Tab. 1.

Besides, we also conduct ablation study to decompose different size convolution kernels in Tab. 6. We can find that decomposing a 21×21 convolution works better than decomposing a 7×7 convolution which demonstrates large kernel is critical for visual tasks. Decomposing a larger 28×28 convolution, we find 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. Tab. 7 presents the comparison of VAN with other MLPs, CNNs and ViTs. VAN outperforms common CNNs (ResNet [5], ResNeXt [7], ConvNeXt [21], *etc*.), ViTs (DeiT [19], PVT [20] and Swin-Transformer [15], *etc*.) and MLPs (MLP-Mixer [69], ResMLP [71], gMLP [72], *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. 6, 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*). Compared VAN with ConvNeXt [21], VAN-B2 surpasses

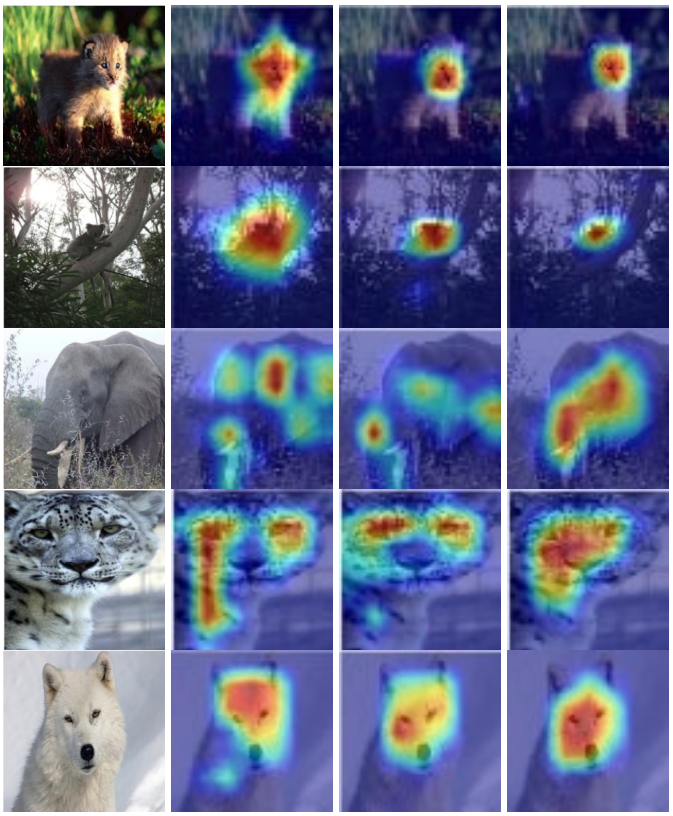


Fig. 7. Visualization results. All images come from different categories in ImageNet validation set. CAM is produced by using Grad-CAM [85]. We compare different CAMs produced by Swin-T [15], ConvNeXt-T [21] and VAN-B2.

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%). As for MLPs, we choose gMLP [72]. VAN-B2 surpass gMLP-S [72] by 3.2% (82.8% vs. 79.6%) which reflects the importance of locality.

Throughput. We test the throughput of the Swin transformer [15] and VAN on some hardware environment with the RTX 3090. Results are shown in Tab. 4. Besides, we also plots the accuracy-throughput diagram on Fig. 5, which clearly demonstrates 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 [85] to visualize the attentions on the ImageNet validation set produced by VAN-B2 model. Results in Fig. 7 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 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 in Fig. 7), which demonstrates its ability to capture long-range dependence.

TABLE 9
Object detection on COCO 2017 dataset. #P means parameter. RetinaNet 1× denotes models are based on RetinaNet [103] and we train them for 12 epochs.

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
PoolFormer-S12 [20]	21.7	36.2	56.2	38.2	20.8	39.1	48.0
PVT-Tiny [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-Small [20]	34.2	40.4	61.3	43.0	25.0	42.9	55.7
PoolFormer-S24 [98]	31.1	38.9	59.7	41.3	23.3	42.1	51.8
PoolFormer-S36 [98]	40.6	39.5	60.5	41.8	22.5	42.9	52.4
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-Medium [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 are based on Mask R-CNN [104] and we train them for 12 epochs. AP^b and AP^m refer to bounding box AP and mask AP respectively.

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
PoolFormer-S12 [98]	31.6	37.3	59.0	40.1	34.6	55.8	36.9
PVT-Tiny [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-Small [20]	44.1	40.4	62.9	43.8	37.8	60.1	40.3
PoolFormer-S24 [98]	41.0	40.1	62.2	43.4	37.0	59.1	39.6
PoolFormer-S36 [98]	50.5	41.0	63.1	44.8	37.7	60.1	40.0
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
ResNeXt101-32x4d [7]	62.8	41.9	62.5	45.9	37.5	59.4	40.2
PVT-Medium [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**

4.1.3 Pretraining on ImageNet-22K.

Settings. ImageNet-22K is a large-scale image classification dataset, which contains about 14M images and 21841 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 8,196. Other training details are 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 [101]) and ViTs(*e.g.,* Swin Transformer [15], ViT [13] and CoAtNet [102]). As shown in Tab. 8, VAN achieves 87.8% Top-1 accuracy with 200M parameters and surpasses the same level ViT [13], Swin Transformer [15], EFFNetV2 [101] and ConvNeXt [21] on different resolution, which proves the strong capability to adapt large-scale pretraining.

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 1,280 × 800.

Backbone	Method	AP^b	AP^b_{50}	AP^b_{75}	#P (M)	GFLOPs
Swin-T [5]	Mask R-CNN [104]	46.0	68.1	50.3	48	264
ConvNeXt-T [15]		46.2	67.9	50.8	48	262
VAN-B2		**48.8**	**70.0**	**53.6**	46	273
ResNet50 [5]	Cascade Mask R-CNN [105]	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 [106]	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 [107]	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

TABLE 12
Results of semantic segmentation on ADE20K [83] validation set. The upper and lower part are obtained under two different training/validation schemes following [98] and [15]. We calculate FLOPs with input size 512 × 512 for Semantic FPN [108] and 2,048 × 512 for UperNet [109].

Method	Backbone	#P(M)	GFLOPs	mIoU (%)
Semantic FPN [108]	PVTv2-B0 [80]	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 [98]	16	31	37.2
	PVTv2-B1 [80]	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 [98]	23	39	40.3
	PVTv2-B2 [80]	29	46	45.2
	VAN-B2	30	48	**46.7**
	ResNet101 [5]	48	65	38.8
	ResNeXt101-32x4d [7]	47	65	39.7
	PVT-Medium [20]	48	61	43.5
	PoolFormer-S36 [98]	35	48	42.0
	PVTv2-B3 [80]	49	62	47.3
	VAN-B3	49	68	**48.1**
UperNet [109]	ResNet-101 [5]	86	1029	44.9
OCRNet [41]		56	923	45.3
HamNet [42]		69	1111	46.8
UperNet [109]	Swin-T [15]	60	945	46.1
	ConvNeXt-T [21]	60	939	46.7
	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
	VAN-B4	90	1098	**52.2**

4.2 Object Detection

Settings. We conduct object detection and instance segmentation experiments on COCO 2017 benchmark [82], which contains 118K images in training set and 5K images in validation set. MMDetection [110] 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 [98]. Many kinds of detection models (*e.g.,* Mask R-CNN [104], RetinaNet [103], Cascade Mask R-CNN [105], Sparse R-CNN [111] ,*etc*.) are included to demonstrate the effectiveness of our method. All backbone models are pre-trained on ImageNet training set.

TABLE 13
Compare with the state-of-the-art methods on ADE20K validation set. Params means parameter. GFLOPs 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	GFLOPs	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**

Results. According to Tab. 9 and Tab. 10, we find that VAN surpasses CNN-based method ResNet [5] and transformer-based method PVT [20] with a large margin under RetinaNet [103] 1x and Mask R-CNN [104] 1x settings. Besides, we also compare the state-of-the-art methods Swin transformer [15] and ConvNeXt [21] in Tab. 11. Results show that VAN achieves the state-of-the-art performance with different detection methods such as Mask R-CNN [104] and Cascade Mask R-CNN [105].

4.3 Semantic Segmentation

Settings. We conduct experiments on ADE20K [83], which contains 150 semantic categories for semantic segmentation. It consists of 20,000, 2,000 and 3,000 separately for training, validation and testing. MMSEG [112] is used as the base framework and two famous segmentation heads, Semantic FPN [108] and UperNet [109], are employed for evaluating our VAN backbones. For a fair comparison, we adopt two training/validating schemes following [98] and [15] and quantitative results on the validation set are shown in the upper and lower part in Tab. 12, respectively. All backbone models are pre-trained on ImageNet-1K or ImageNet-22K training set.

Results. From the upper part in Tab. 12, compared with different backbones using FPN [108], VAN-Based methods are superior to CNN-based (ResNet [5], ResNeXt [7]) or transformer-based (PVT [20], PoolFormer [98], PVTv2 [80]) methods. For instance, we surpass four PVTv2 [80] 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 Tab. 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 [109], 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 Tab. 13.

4.4 Panoptic Segmentation

Settings. We conduct our panoptic segmentation on COCO panoptic segmentation dataset [82] and choose Mask2Former [113] as our segmentation head. For fair comparison, we adopt the default settings in MMDetection [110] and same training/validating scheme as Mask2Former [113]. All backbone models are pre-trained on ImageNet-1K or ImageNet-22K set.

TABLE 14
Experimental results on COCO panoptic segmentation. * means model is pretrained on ImageNet-22K dataset. All methods are based on Mask2Former [113]. 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**

TABLE 15
Comparison with the state-of-the-art vision backbones on COCO benchmark for pose estimation. Models are based SimpleBaseline [114].

Backbone	Input size	AP	AP^{50}	AP^{75}	AR	#P (M)	GFLOPs
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

Results. As shown in Tab. 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 set new state-of-the-art performance for panoptic segmentation task.

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 [114] as our decoder part, which is same with Swin transformer [15] and PVT [20]. All experiments are based on MMPose [115].

Results. Experimental results are shown on Tab. 15. For 256 × 192 input, VAN-B2 outperform Swin-T and PVT-S [20] 2.5AP (74.9 vs. 72.4) and 3.5AP (74.9 vs. 71.4) and with simliar computing and parameters. Furthermore, VAN-B2 exceeds Swin-B 2AP (74.9 vs. 72.9) and 1.8AP (76.7 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 [116], 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 [117]. Results on Tab. 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 base on EDN [118]. We replace the backbone with VAN and hold experiments on common saliency detection benchmarks, including DUTS [119]. DUT-O [120] and PASCAL-S [121]. Results on Tab. 17 show that VAN clearly surpasses other backbones ResNet [5] and PVT [20] on all datasets.

TABLE 16
Experimental results on CUB-200 fine-grain classification dataset. * means 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

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 [59], [64]. Combining the above two points, we believe 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 [122] in NLP is still worth exploring. we look forward to seeing VANs becoming a general model.

7 Conclusion

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 Sec. 6.

Acknowledgments

This work was supported by the National Key R&D Program(2018AAA0100400), the Natural Science Foundation of China (Project 61521002, 61922046), and Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv. Neural Inform. Process. Syst.*, vol. 25, pp. 1097–1105, 2012.

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

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

[7] 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.

[8] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 6848–6856.

[9] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 4700–4708.

[10] 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.

[11] 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 Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 652–662, 2021.

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

[13] 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.

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

[15] 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.

[16] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Int. Conf. Comput. Vis.*, 2017, pp. 764–773.

[17] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Adv. Neural Inform. Process. Syst.*, vol. 34, 2021.

[18] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[19] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Int. Conf. Mach. Learn.* PMLR, 2021, pp. 10 347–10 357.

[20] 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.

[21] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," 2022.

[22] 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," 2021.

[23] J. P. Gottlieb, M. Kusunoki, and M. E. Goldberg, "The representation of visual salience in monkey parietal cortex," *Nature*, vol. 391, no. 6666, pp. 481–484, 1998.

[24] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[25] J. M. Wolfe and T. S. Horowitz, "What attributes guide the deployment of visual attention and how do they do it?" *Nature reviews neuroscience*, vol. 5, no. 6, pp. 495–501, 2004.

[26] J. K. Tsotsos, S. M. Culhane, W. Y. K. Wai, Y. Lai, N. Davis, and F. Nuflo, "Modeling visual attention via selective tuning," *Artificial intelligence*, vol. 78, no. 1-2, pp. 507–545, 1995.

[27] 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.

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

[29] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Adv. Neural Inform. Process. Syst.*, vol. 33, pp. 1877–1901, 2020.

[30] 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.

[31] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," 2020.

[32] A. Ali, H. Touvron, M. Caron, P. Bojanowski, M. Douze, A. Joulin, I. Laptev, N. Neverova, G. Synnaeve, J. Verbeek *et al.*, "Xcit: Cross-covariance image transformers," *Adv. Neural Inform. Process. Syst.*, vol. 34, 2021.

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

[34] I. Bello, W. Fedus, X. Du, E. D. Cubuk, A. Srinivas, T.-Y. Lin, J. Shlens, and B. Zoph, "Revisiting resnets: Improved training and scaling strategies," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[35] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 4510–4520.

[36] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Int. Conf. Learn. Represent.*, 2014.

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

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

[39] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *Adv. Neural Inform. Process. Syst.*, 2014, pp. 2204–2212.

[40] 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.

[41] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*. Springer, 2020, pp. 173–190.

[42] Z. Geng, M.-H. Guo, H. Chen, X. Li, K. Wei, and Z. Lin, "Is attention better than matrix decomposition?" in *Int. Conf. Learn. Represent.*, 2021.

[43] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu, "Attention mechanisms in computer vision: A survey," *Computational Visual Media*, pp. 1–38, 2022.

[44] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7794–7803.

[45] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3146–3154.

[46] 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.

[47] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[48] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, "Ocnet: Object context network for scene parsing," *arXiv preprint arXiv:1809.00916*, 2018.

[49] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Int. Conf. Mach. Learn.* PMLR, 2019, pp. 7354–7363.

[50] S. Xie, S. Liu, Z. Chen, and Z. Tu, "Attentional shapecontextnet for point cloud recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 4606–4615.

[51] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Eur. Conf. Comput. Vis.* Springer, 2020, pp. 213–229.

[52] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, pp. 187–199, 2021.

[53] 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.

[54] 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.

[55] R. Liu, H. Deng, Y. Huang, X. Shi, L. Lu, W. Sun, X. Wang, J. Dai, and H. Li, "Decoupled spatial-temporal transformer for video inpainting," *arXiv preprint arXiv:2104.06637*, 2021.

[56] I. Bello, "Lambdanetworks: Modeling long-range interactions without attention," in *International Conference on Learning Representations*, 2021.

[57] Y. Xu, Q. Zhang, J. Zhang, and D. Tao, "Vitae: Vision transformer advanced by exploring intrinsic inductive bias," *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 522–28 535, 2021.

[58] R. Liu, H. Deng, Y. Huang, X. Shi, L. Lu, W. Sun, X. Wang, J. Dai, and H. Li, "Fuseformer: Fusing fine-grained information in transformers for video inpainting," in *Int. Conf. Comput. Vis.*, 2021, pp. 14 040–14 049.

[59] H. Bao, L. Dong, S. Piao, and F. Wei, "BEit: BERT pre-training of image transformers," in *Int. Conf. Learn. Represent.*, 2022.

[60] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, and L. Zhang, "DAB-DETR: Dynamic anchor boxes are better queries for DETR," in *Int. Conf. Learn. Represent.*, 2022.

[61] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *Int. Conf. Comput. Vis.*, 2021, pp. 22–31.

[62] S. Liu, L. Zhang, X. Yang, H. Su, and J. Zhu, "Query2label: A simple transformer way to multi-label classification," 2021.

[63] Y.-H. Wu, Y. Liu, X. Zhan, and M.-M. Cheng, "P2t: Pyramid pooling transformer for scene understanding," *arXiv preprint arXiv:2106.12011*, 2021.

[64] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," 2021.

[65] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 5659–5667.

[66] Z. Qin, P. Zhang, F. Wu, and X. Li, "Fcanet: Frequency channel attention networks," in *Int. Conf. Comput. Vis.*, 2021, pp. 783–792.

[67] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[68] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[69] 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.

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

[71] 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.

[72] H. Liu, Z. Dai, D. So, and Q. V. Le, "Pay attention to MLPs," in *Adv. Neural Inform. Process. Syst.*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021.

[73] M.-H. Guo, Z.-N. Liu, T.-J. Mu, D. Liang, R. R. Martin, and S.-M. Hu, "Can attention enable mlps to catch up with cnns?" *Computational Visual Media*, vol. 7, no. 3, pp. 283–288, 2021.

[74] R. Liu, Y. Li, L. Tao, D. Liang, S.-M. Hu, and H.-T. Zheng, "Are we ready for a new paradigm shift? a survey on visual deep mlp," 2021.

[75] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 3156–3164.

[76] 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.

[77] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "Bam: Bottleneck attention module," *arXiv preprint arXiv:1807.06514*, 2018.

[78] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Int. Conf. Mach. Learn.* PMLR, 2015, pp. 448–456.

[79] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," 2020.

[80] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pvtv2: Improved baselines with pyramid vision transformer," *arXiv preprint arXiv:2106.13797*, 2021.

[81] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[82] 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.

[83] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 302–321, 2019.

[84] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2921–2929.

[85] 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.

[86] 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," *Advances in neural information processing systems*, vol. 32, 2019.

[87] S.-M. Hu, D. Liang, G.-Y. Yang, G.-W. Yang, and W.-Y. Zhou, "Jittor: a novel deep learning framework with meta-operators and unified graph execution," *Science China Information Sciences*, vol. 63, no. 12, pp. 1–21, 2020.

[88] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" *Adv. Neural Inform. Process. Syst.*, vol. 32, 2019.

[89] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[90] 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.

[91] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *AAAI Conf. Artif. Intell.*, 2020, pp. 13 001–13 008.

[92] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[93] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019.

[94] ——, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[95] 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.

[96] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM journal on control and optimization*, vol. 30, no. 4, pp. 838–855, 1992.

[97] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," *Adv. Neural Inform. Process. Syst.*, vol. 34, 2021.

[98] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan, "Metaformer is actually what you need for vision," *arXiv preprint arXiv:2111.11418*, 2021.

[99] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 10 428–10 436.
[100] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, X. Wei, H. Xia, and C. Shen, "Twins: Revisiting the design of spatial attention in vision transformers," *Adv. Neural Inform. Process. Syst.*, vol. 34, 2021.
[101] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 096–10 106.
[102] 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.
[103] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
[104] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *Int. Conf. Comput. Vis.*, Oct 2017.
[105] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
[106] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 9759–9768.
[107] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," *Adv. Neural Inform. Process. Syst.*, vol. 33, pp. 21 002–21 012, 2020.
[108] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 6399–6408.
[109] 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.
[110] 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.
[111] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang *et al.*, "Sparse r-cnn: End-to-end object detection with learnable proposals," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 14 454–14 463.
[112] M. Contributors, "MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark," https://github.com/open-mmlab/mmsegmentation, 2020.
[113] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1290–1299.
[114] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 466–481.
[115] M. Contributors, "Openmmlab pose estimation toolbox and benchmark," https://github.com/open-mmlab/mmpose, 2020.
[116] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-ucsd birds 200," 2010.
[117] M. Contributors, "Openmmlab's image classification toolbox and benchmark," https://github.com/open-mmlab/mmclassification, 2020.
[118] Y.-H. Wu, Y. Liu, L. Zhang, M.-M. Cheng, and B. Ren, "Edn: Salient object detection via extremely-downsampled network," *IEEE Transactions on Image Processing*, vol. 31, pp. 3125–3136, 2022.
[119] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, "Learning to detect salient objects with image-level supervision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 136–145.
[120] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3166–3173.
[121] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 280–287.
[122] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.