Wave-ViT: Unifying Wavelet and Transformers for Visual Representation Learning

Ting Yao[1], Yingwei Pan[1(✉)], Yehao Li[1], Chong-Wah Ngo[2], and Tao Mei[1]

[1] JD Explore Academy, Beijing, China
tingyao.ustc@gmail.com, panyw.ustc@gmail.com, yehaoli.sysu@gmail.com, tmei@jd.com
[2] Singapore Management University, Singapore, Singapore
cwngo@smu.edu.sg

Abstract. Multi-scale Vision Transformer (ViT) has emerged as a powerful backbone for computer vision tasks, while the self-attention computation in Transformer scales quadratically w.r.t. the input patch number. Thus, existing solutions commonly employ down-sampling operations (*e.g.*, average pooling) over keys/values to dramatically reduce the computational cost. In this work, we argue that such over-aggressive down-sampling design is not invertible and inevitably causes information dropping especially for high-frequency components in objects (*e.g.*, texture details). Motivated by the wavelet theory, we construct a new Wavelet Vision Transformer (**Wave-ViT**) that formulates the invertible down-sampling with wavelet transforms and self-attention learning in a unified way. This proposal enables self-attention learning with lossless down-sampling over keys/values, facilitating the pursuing of a better efficiency-vs-accuracy trade-off. Furthermore, inverse wavelet transforms are leveraged to strengthen self-attention outputs by aggregating local contexts with enlarged receptive field. We validate the superiority of Wave-ViT through extensive experiments over multiple vision tasks (*e.g.*, image recognition, object detection and instance segmentation). Its performances surpass state-of-the-art ViT backbones with comparable FLOPs. Source code is available at https://github.com/YehLi/ImageNetModel.

Keywords: Vision transformer · Wavelet transform · Self-attention learning · Image recognition

1 Introduction

Recently, leveraging Transformer architecture [53] for visual representation learning has achieved widespread dominance in computer vision field. Transformer architecture has brought forward milestone improvement for a series of downstream vision tasks [4,9,13,29,35–37,42,43,45,48,56,58,59,67,71,74], including
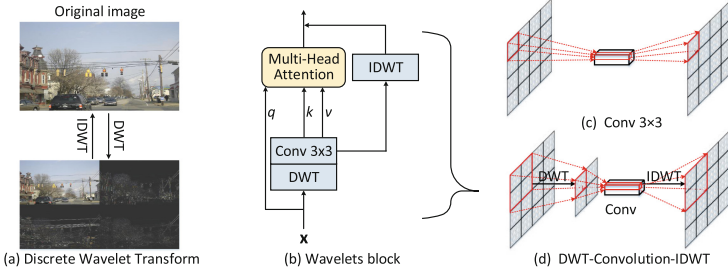
Fig. 1. An illustration of (a) Discrete Wavelet Transform (DWT) and Inverse DWT (IDWT) over an image, (b) our Wavelets block, and the comparison between (c) a single 3 × 3 convolution and (d) DWT-Convolution-IDWT process in our Wavelets block.

both image recognition and dense prediction tasks (*e.g.*, object detection and semantic segmentation). At its heart is a basic self-attention block that triggers long-range interaction among visual tokens. The Vision Transformer (ViT) [13] is one of the early attempts that directly employs a pure Transformer over image patches, and manages to attain competitive image recognition performance against CNN counterparts. However, applying the primary ViT architecture using its outputs of single-scale and low-resolution feature map for the pixel-level dense prediction tasks (*e.g.*, instance/semantic segmentation) is not trivial. Therefore, considering that visual patterns commonly occur at multiple scales in natural scenery, there has been research efforts pushing the limits of ViT backbones by aggregating contexts from multiple scales (*e.g.*, "pyramid" strategy). For example, Pyramid Vision Transformer (PVT) [59,60] integrates pyramid structure into Transformer framework, yielding multi-scale feature maps for dense prediction tasks. Multiscale Vision Transformers (MViT) [14] learns multi-scale feature hierarchies in Transformer architecture by hierarchically expanding the channel capacity while reducing the spatial resolution.

One primary challenge of applying self-attention over multi-scale feature maps is the quadratical computational cost that scales w.r.t the number of input patches (*i.e.*, spatial resolution). Thus, typical multi-scale ViT approaches usually perform down-sampling operations (*e.g.*, average pooling in [59] or pooling kernels in [14]) over keys/values to reduce computational cost. Nevertheless, these pooling based operations inevitably result in information dropping (*e.g.*, the high-frequency components of object texture details), and thus adversely affect the performances especially for dense prediction tasks. Furthermore, the recent studies (*e.g.*, [72]) also have shown that applying pooling operations in CNNs would hurt the shift-equivariance of deep networks.

In this paper, we propose **Wavelets block** to perform invertible down-sampling through wavelet transforms, aiming to preserve the original image details for self-attention learning while reducing computational cost. Wavelet transform is a fundamental time-frequency analysis method that decomposes

input signals into different frequency subbands to address the aliasing problem. In particular, Discrete Wavelet Transform (DWT) [40] enables invertible down-sampling by transforming 2D data into four discrete wavelet subbands (Fig. 1 (a)): low-frequency component (I_{LL}) and high-frequency components (I_{LH}, I_{HL}, I_{HH}). Here the low-frequency component reflects the basic object structure at coarse-grained level, while the high-frequency components retain the object texture details at fine-grained level. In this way, various levels of image details are preserved in different subbands of lower resolution without information dropping. Furthermore, inverse DWT (IDWT) can be applied to reconstruct the original image. The information preserving transformation motivates the design of an efficient Transformer block with lossless and invertible down-sampling for self-attention learning over multi-scale feature maps.

Technically, as shown in Fig. 1 (b), Wavelets block first employs DWT to transform each input key/value to four subbands of lower resolution. After stacking the four subbands into a down-sampled feature map, a 3 × 3 convolution is performed to further impose spatial locality over the frequency subbands. This leads to locally contextualized down-sampled keys/values. The multi-head self-attention learning is conducted on the down-sampled keys/values and input query. Meanwhile, IDWT can be applied over the down-sampled keys/values to reconstruct high-resolution feature map that preserves image details. Compared to the single 3 × 3 convolution (Fig. 1 (c)), the process of DWT-Convolution-IDWT (Fig. 1 (d)) enables a stronger local contextualization via enlarged receptive field, with negligible increase in computation and memory. Finally, we combine the attended feature map via self-attention learning and the reconstructed feature map with local contextualization as the outputs of Wavelets block.

By operating Wavelets block over multi-scale features in the multi-stage Transformer framework, we present a new Wavelet Vision Transformer (**Wave-ViT**) for visual representation learning. Wave-ViT has been properly analyzed and verified through extensive experiments over different vision tasks, which demonstrate its superiority against state-of-the-art ViTs. More remarkably, under a comparable number of parameters, Wave-ViT achieves 85.5% top-1 accuracy on ImageNet for image recognition, which absolutely improves PVT (83.8%) with 1.7%. For object detection and instance segmentation on COCO, Wave-ViT absolutely surpasses PVT with 1.3% and 0.5% mAP, with 25.9% less parameters.

2   Related Work

Visual Representation Learning. Early studies predominantly focused on exploring CNN for visual representation learning, leading to a series of CNN backbones, *e.g.*, [21,26,27,46,50]. Most of them stack low-to-high convolutions by going deeper, targeting for producing low-resolution and high-level representations tailored for image recognition. However, dense prediction tasks like semantic segmentation require high-resolution and even pixel-level representations. To tackle this, several multi-scale CNNs are established. For example,

Res2Net [16] presents a multi-scale building block that contains hierarchical residual-like connections. HRNet [55] connects high-to-low resolution convolution streams in parallel and meanwhile exchanges the information across resolutions repeatedly.

Recently, due to the powerful long-range interaction modeling in Transformer [53], Transformer has advanced natural language understanding. Inspired by this, numerous Transformer-based architectures for vision understanding have started. A few attempts augment convolutional operators with the global self-attention [2] or local self-attention [22,44,47,75], yielding a hybrid backbone of CNN and Transformer. On a parallel note, Vision Transformer (ViT) [13] first employs a pure Transformer over the sequence of image patches for image recognition. DETR [4] also leverages a pure Transformer to construct an end-to-end detector for object detection. Different from ViT that solely divides input image into patches, TNT [19] further decomposes patches into sub-patches as "visual words". A sub-transformer is additionally integrated into Transformer to perform self-attention over smaller "visual words". Subsequently, to facilitate dense prediction tasks, multi-scale paradigm is introduced into Transformer structure, leading to multi-scale Vision Transformer backbones [14,35,59,60]. In particular, Swin Transformer [35] upgrades ViT by constructing hierarchical feature maps via merging image patches in deeper layers. Pyramid Vision Transformer (PVT) [60] designs a pyramid structure Transformer that produces multi-scale feature maps in a four-stage architecture. PVTv2 [59] further improves PVT by using average pooling to reduce spatial dimension of keys/values, rather than convolutions in PVT. Multiscale Vision Transformers (MViT) [14] integrates Transformer architecture with multi-scale feature hierarchies, and pooling kernels is employed over query/keys/values for spatial reduction.

Our Wave-ViT is also a type of multi-scale ViT. Existing multi-scale ViTs (*e.g.*, [14,59,60]) commonly adopt irreversible down-sampling operations like average pooling or pooling kernels for spatial reduction. In contrast, Wave-ViT capitalizes on wavelet transforms to reduce spatial dimension of keys/values via invertible down-sampling for self-attention learning over multi-scale features, leading to a better trade-off between computation cost and performance.

Wavelet Transform in Computer Vision. Wavelet Transform is effective for time-frequency analysis. Considering that Wavelet Transform is invertible and capable of preserving all information, Wavelet Transform has been exploited in CNN architectures for performance boosting in various vision tasks. For example, in [1], Bae *et al.* validate that learning CNN representations over wavelet subbands can benefit the task of image restoration. DWSR [18] takes low-resolution wavelet subbands as inputs to recover the missing details for image super-resolution task. Multi-level wavelet transform [34] is utilized to enlarge receptive field without information dropping for image restoration. Williams *et al.* [61] utilize Wavelet Transform to decompose input features into a second level decomposition, and discard first-level subbands to reduce feature dimensions for image recognition. Haar wavelet CNNs is integrated with multi-resolution analysis in [15] for texture classification and image annotation. In [41], ResNet is

remoulded by combining the first layer with a wavelet scattering network, which achieves comparable performances on image recognition with less parameters.

Although wavelet transform has been exploited as down-sampling/up-sampling operations in CNNs, it is never explored for Transformer architecture. In this work, our Wave-ViT goes beyond existing CNNs that operate wavelet transform over feature maps across different stages, and leverages wavelet transform to down-sample keys/values within Transformer block, making the impact more thorough for feature learning.
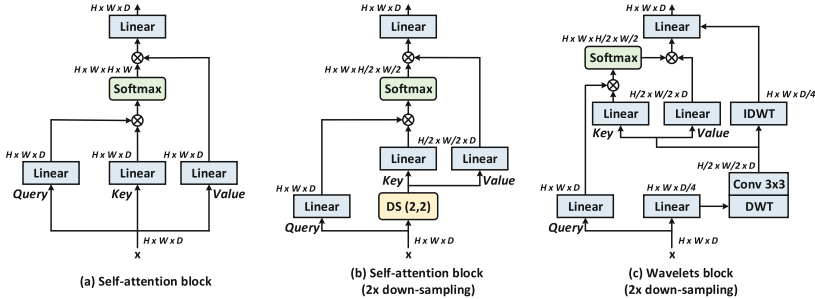
Fig. 2. The detailed architectures of (a) basic self-attention block in ViT Backbones, (b) self-attention block with down-sampling operation (*i.e.*, DS(2, 2)) that reduces the spatial scale of both height and width by half, and (c) our Wavelets block that capitalizes on wavelet transforms to enable lossless down-sampling.

3   Our Approach: Wavelet Vision Transformer

This section starts by briefly reviewing the most typical multi-head self-attention block in ViTs, particularly on how the self-attention block is scaled down for reducing computational cost in the existing multi-scale ViTs. After that, a novel principled Transformer building block, named Wavelets block, is designed to integrate self-attention learning with wavelet transforms in a unified fashion. Such design upgrades typical self-attention block by exploiting wavelet transforms to perform invertible down-sampling, which elegantly reduces spatial dimension of keys/values without information dropping. Furthermore, this block applies inverse wavelet transforms over down-sampled keys/values to enhance outputs with enlarged receptive field. Finally, after applying Wavelets block over multi-scale features in the multi-stage Transformer architecture, we elaborate a new multi-scale ViT backbone, *i.e.*, Wavelet Vision Transformer.

3.1   Preliminaries

Multi-head Self-attention in ViT Backbones. Mainstream Transformer architectures, especially Vision Transformer backbones [13], often rely on the

typical multi-head self-attention that captures long-range dependencies among inputs in a scalable fashion. Here we present a general formulation of multi-head self-attention as illustrated in Fig. 2 (a). Technically, let $X \in \mathbb{R}^{H \times W \times D}$ be the input 2D feature map, where $H/W/D$ denote the height/width/channel number, respectively. Here X can be reshaped as a patch sequence, consisting of $n = H \times W$ image patches and the dimension of each patch is D. We linearly transform the input patch sequence X into three groups in parallel: queries $Q \in \mathbb{R}^{n \times D}$, keys $K \in \mathbb{R}^{n \times D}$, and values $V \in \mathbb{R}^{n \times D}$. After that, the multi-head self-attention (**MultiHead**) module [53] decomposes each query/key/value into N_h parts along channel dimension, leading to queries $Q_j \in \mathbb{R}^{n \times D_h}$, keys $K_j \in \mathbb{R}^{n \times D_h}$, and values $V_j \in \mathbb{R}^{n \times D_h}$ for the j-th head. Note that N_h is head number and D_h denotes the dimension of each head. Then, we perform self-attention learning (**Attention**) over queries, keys and values for each head, and the outputs of each head are concatenated, followed by a linear transformation to compose the final outputs:

$$
\begin{aligned}
\textbf{MultiHead}(Q, K, V) &= \textbf{Concat}(head_0, head_1, ..., head_{N_h})W^O, \\
head_j &= \textbf{Attention}(Q_j, K_j, V_j), \\
\textbf{Attention}(Q_j, K_j, V_j) &= \textbf{Softmax}(\frac{Q_j K_j^T}{\sqrt{D_h}})V_j,
\end{aligned}
\tag{1}
$$

where **Concat**$(\cdot)$ denotes concatenation and W^O is the transformation matrix. According to the general formulation in Eq.(1), the computational cost of multi-head self-attention for the input feature map $X \in \mathbb{R}^{H \times W \times D}$ is $\mathcal{O}(H^2 W^2 D)$, which scales quadratically w.r.t. the input patch number. Such design inevitably leads to a sharp rise in computational cost especially for high-resolution inputs.

Self-attention with Down-sampling in Multi-scale ViT Backbones. To alleviate the heavy self-attention computation overhead for high-resolution inputs, the existing multi-scale ViT backbones commonly adopt the down-sampling operations (*e.g.*, average pooling in [59] or pooling kernels in [14]) over keys/values for spatial reduction. Taking the self-attention block with 2 $\times$ down-sampling in Fig. 2 (b) as an example, the input 2D feature map X is first down-sampled by a factor r ($r = 2$ in this case). Here the down-sampling operator is denoted as DS(2, 2), that reduces the spatial scale of both height and width by half. Next, the down-sampled feature map is linearly transformed into keys $K^d \in \mathbb{R}^{\frac{n}{r^2} \times D}$ and values $V^d \in \mathbb{R}^{\frac{n}{r^2} \times D}$ to trigger multi-head self-attention learning. As such, the overall computational cost of multi-head self-attention is dramatically reduced by a factor of r^2 (*i.e.*, $\mathcal{O}(\frac{H^2 W^2 D}{r^2})$).

3.2 Wavelets Block

Although the aforementioned multi-scale ViT backbones reduce self-attention computation via down-sampling, the commonly adopted down-sampling operations like average pooling are irreversible, and inevitably result in information dropping. To mitigate this issue, we design a principled self-attention

block, named **Wavelets block**, that novelly capitalizes on wavelet transforms to enable invertible down-sampling for self-attention learning. Such invertible down-sampling is seamlessly incorporated into the typical self-attention block, pursuing efficient multi-head self-attention learning with lossless down-sampling. Figure 2 (c) details the architecture of our Wavelets block.

Formally, given the input 2D feature map $X \in \mathbb{R}^{H \times W \times D}$, we first linearly transform it into $\widetilde{X} = XW_d$ with reduced channel dimension via embedding matrix $W_d \in \mathbb{R}^{D \times \frac{D}{4}}$. Next, we employ Discrete Wavelet Transform (DWT) to down-sample the input $\widetilde{X} \in \mathbb{R}^{H \times W \times \frac{D}{4}}$ by decomposing it into four wavelet subbands. Note that here we choose the classical Haar wavelet for DWT as in [33] for simplicity. Concretely, DWT applies the low-pass filter $f_L = (1/\sqrt{2}, 1/\sqrt{2})$ and high-pass filter $f_H = (1/\sqrt{2}, -1/\sqrt{2})$ along the rows to encode $\widetilde{X}$ into two subbands X_L and X_H. Next, the same low-pass filter f_L and high-pass filter f_H are employed along the columns of the learnt subbands X_L and X_H, leading to all the four wavelet subbands: $X_{LL} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times \frac{D}{4}}$, $X_{LH} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times \frac{D}{4}}$, $X_{HL} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times \frac{D}{4}}$, and $X_{HH} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times \frac{D}{4}}$. X_{LL} refers to the low-frequency component that reflects the basic object structure at coarse-grained level. X_{LH}, X_{HL}, and X_{HH} represent the high-frequency components that retain the object texture details at fine-grained level. In this way, each wavelet subband can be regarded as the down-sampled version of $\widetilde{X}$, and all of them cover every detail of inputs without any information dropping.

We concatenate the four wavelet subbands along the channel dimension to form $\hat{X} = [X_{LL}, X_{LH}, X_{HL}, X_{HH}] \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times D}$. A 3×3 convolution is further applied to impose spatial locality over $\hat{X}$, yielding the locally contextualized down-sampled feature map X^c. Next, this down-sampled feature map X^c is linearly transformed into down-sampled keys $K^w \in {}^{m \times D}$ and values $V^w \in {}^{m \times D}$, where $m = \frac{H}{2} \times \frac{W}{2}$ is the number of patches. Similarly, the wavelet-based multi-head self-attention learning **Attention$^{\mathbf{w}}$** is thus performed over the queries and the corresponding down-sampled keys/values for each head:

$$head_j = \textbf{Attention}^{\textbf{w}}(Q_j, K_j^w, V_j^w) = \textbf{Softmax}(\frac{Q_j K_j^{wT}}{\sqrt{D_h}})V_j^w, \qquad (2)$$

where K_j^w / V_j^w denotes the down-sampled keys/values for the j-th head, respectively. Here the aggregated output of self-attention learning for each head ($head_j$) can be interpreted as the long-range contextualized information of inputs.

As a beneficial by-product, we additionally apply inverse DWT (IDWT) over the locally contextualized down-sampled feature X^c. According to the wavelet theory, the reconstructed feature map X^r is able to retain every detail of primary input $\widetilde{X}$. It is worthy to note that compared to a single 3×3 convolution, such process of DWT-Convolution-IDWT in Wavelets block triggers a stronger local contextualization with enlarged receptive field, with negligible increase in computational cost/memory.

Finally, we concatenate all the long-range contextualized information of each head plus the reconstructed locally contextualized information X^r, followed by a linear transformation to compose the outputs of our Wavelets block:

$$\mathbf{WaveletsBlock}(X) = \mathbf{MultiHead^w}(XW^q, X^c W^k, X^c W^v, X^r),$$
$$\mathbf{MultiHead^w}(Q, K, V, X^r) = \mathbf{Concat}(head_0, head_1, ..., head_{N_h}, X^r)\widetilde{W}^O,$$

(3)

where $\widetilde{W}^O$ is the transformation matrix.

3.3   Wavelet Vision Transformer

Recall that our Wavelets block is a principled unified self-attention block, it is feasible to construct multi-scale ViT backbones with Wavelets blocks. Following the basic configuration of existing multi-scale ViTs [35,60], we present three variants of our Wavelet Vision Transformer (Wave-ViT) with different model sizes, *i.e.*, Wave-ViT-S (small size), Wave-ViT-B (base size), and Wave-ViT-L (large size). Note that Wave-ViT-S/B/L shares similar model size and computational complexity with Swin-T/S/B [35]. Specifically, given the input image (size: 224 × 224), the entire architecture of Wave-ViT consists of four stages, and each stage is comprised of a patch embedding layer, and a stack of Wavelets blocks followed by feed-forward layers. We follow the design principle of ResNet [21] by progressively increasing the channel dimensions of all the four stages and meanwhile shrinking the spatial resolutions. Table 1 details the architectures of all the three variants of Wave-ViT, where E_i, $Head_i$, and C_i is the expansion ratio of feed-forward layer, head number, and the channel dimension in stage i.

Table 1. Detailed architecture specifications for three variants of our Wave-ViT with different model sizes, *i.e.*, Wave-ViT-S (small size), Wave-ViT-B (base size), and Wave-ViT-L (large size). E_i, $Head_i$, and C_i represents the expansion ratio of feed-forward layer, the head number, and the channel dimension in each stage i, respectively.

	Output Size	Wave-ViT-S	Wave-ViT-B	Wave-ViT-L
Stage 1	$\frac{H}{4} \times \frac{W}{4}$	$\begin{bmatrix} E_1 = 8 \\ Head_1 = 2 \\ C_1 = 64 \end{bmatrix} \times 3$	$\begin{bmatrix} E_1 = 8 \\ Head_1 = 2 \\ C_1 = 64 \end{bmatrix} \times 3$	$\begin{bmatrix} E_1 = 8 \\ Head_1 = 3 \\ C_1 = 96 \end{bmatrix} \times 3$
Stage 2	$\frac{H}{8} \times \frac{W}{8}$	$\begin{bmatrix} E_2 = 8 \\ Head_2 = 4 \\ C_2 = 128 \end{bmatrix} \times 4$	$\begin{bmatrix} E_2 = 8 \\ Head_2 = 4 \\ C_2 = 128 \end{bmatrix} \times 4$	$\begin{bmatrix} E_2 = 8 \\ Head_2 = 6 \\ C_2 = 192 \end{bmatrix} \times 6$
Stage 3	$\frac{H}{16} \times \frac{W}{16}$	$\begin{bmatrix} E_3 = 4 \\ Head_3 = 10 \\ C_3 = 320 \end{bmatrix} \times 6$	$\begin{bmatrix} E_3 = 4 \\ Head_3 = 10 \\ C_3 = 320 \end{bmatrix} \times 12$	$\begin{bmatrix} E_3 = 4 \\ Head_3 = 12 \\ C_3 = 384 \end{bmatrix} \times 18$
Stage 4	$\frac{H}{32} \times \frac{W}{32}$	$\begin{bmatrix} E_4 = 4 \\ Head_4 = 14 \\ C_4 = 448 \end{bmatrix} \times 3$	$\begin{bmatrix} E_4 = 4 \\ Head_4 = 16 \\ C_4 = 512 \end{bmatrix} \times 3$	$\begin{bmatrix} E_4 = 4 \\ Head_4 = 16 \\ C_4 = 512 \end{bmatrix} \times 3$

4   Experiments

We evaluate the effectiveness of Wave-ViT via various empirical evidence on several mainstream CV tasks. Concretely, we consider the following evaluations to compare the quality of learnt features obtained from various vision backbones: (a) Training from scratch for image recognition task on ImageNet1K [12]; (b) Fine-tuning the backbones (pre-trained on ImageNet1K) for downstream tasks, *i.e.*, object detection and instance segmentation on COCO [32], and semantic segmentation on ADE20K [77]; (c) Ablation studies that support each design in our Wavelets block; (d) Visualization of learnt representation by Wave-ViT.

4.1   Image Recognition on ImageNet1K

Dataset and Optimization Setups. In the task of image recognition, we adopt the ImageNet1K benchmark, which comprises 1.28 million training images and 50K validation images from 1,000 classes. All vision backbones are trained

Table 2. The performances of various vision backbones on ImageNet1K for image recognition. ⋆ indicates that the backbone is additionally trained with Token Labeling objective with MixToken [24] and convolutional stem [57] for patch encoding. We group all runs into three categories, and all backbones within each category shares similar GFLOPs: Small (GFLOPs < 6), Base (6 ≤ GFLOPs < 10), Large (10 ≤ GFLOPs < 22).

Method	Params	GFLOPs	Top-1	Top-5	Method	Params	GFLOPs	Top-1	Top-5
Small					Large				
ResNet-50 [21]	25.5M	4.1	78.3	94.3	ResNet-152 [21]	60.2M	11.6	81.3	95.5
BoTNet-S1-50 [47]	20.8M	4.3	80.4	95.0	ResNeXt101 [64]	83.5M	15.6	81.5	–
Swin-T [35]	29.0M	4.5	81.2	95.5	DeiT-B [51]	86.6M	17.6	81.8	95.6
ConViT-S [11]	27.8M	5.4	81.3	95.7	SE-ResNet-152 [23]	66.8M	11.6	82.2	95.9
T2T-ViT-14 [68]	21.5M	4.8	81.5	95.7	ResNeSt-101 [70]	48.3M	10.2	82.3	–
RegionViT-Ti+ [6]	14.3M	2.7	81.5	–	ConViT-B [11]	86.5M	16.8	82.4	95.9
SE-CoTNetD-50 [30]	23.1M	4.1	81.6	95.8	T2T-ViTt-24 [68]	64.1M	15.0	82.6	95.9
Twins-SVT-S [9]	24.1M	2.9	81.7	95.6	TNT-B [19]	65.6M	14.1	82.9	96.3
CoaT-Lite Small [65]	20.0M	4.0	81.9	95.5	DeepViT-L [78]	58.9M	12.8	83.1	–
PVTv2-B2 [59]	25.4M	4.0	82.0	96.0	RegionViT-B [6]	72.7M	13.0	83.2	96.1
Wave-ViT-S	19.8M	4.3	82.7	96.2	CaiT-S36 [52]	68.4M	13.9	83.3	–
Wave-ViT-S⋆	22.7M	4.7	**83.9**	**96.6**	CrossViT-15-384 [5]	28.5M	21.4	83.5	–
Base					BoTNet-S1-128 [47]	75.1M	19.3	83.5	96.5
ResNet-101 [21]	44.6M	7.9	80.0	95.0	Swin-B [35]	88.0M	15.4	83.5	96.5
BoTNet-S1-59 [47]	33.5M	7.3	81.7	95.8	PVTv2-B4 [59]	62.6M	10.1	83.6	96.7
T2T-ViT-19 [68]	39.2M	8.5	81.9	95.7	Twins-SVT-L [9]	99.3M	15.1	83.7	96.5
CvT-21 [62]	32.0M	7.1	82.5	–	RegionViT-B+ [6]	73.8M	13.6	83.8	–
Swin-S [35]	50.0M	8.7	83.2	96.2	Focal-Base [66]	89.8M	16.0	83.8	96.5
Twins-SVT-B [9]	56.1M	8.6	83.2	96.3	PVTv2-B5 [59]	82.0M	11.8	83.8	96.6
SE-CoTNetD-101 [30]	40.9M	8.5	83.2	96.5	SE-CoTNetD-152 [30]	55.8M	17.0	84.0	97.0
PVTv2-B3 [59]	45.2M	6.9	83.2	96.5	LV-ViT-M⋆ [24]	55.8M	16.0	84.1	96.7
RegionViT-M+ [6]	42.0M	7.9	83.4	–	VOLO-D2⋆ [69]	58.7M	14.1	85.2	–
VOLO-D1⋆ [69]	26.6M	6.8	84.2	–	VOLO-D3⋆ [69]	86.3M	20.6	85.4	–
Wave-ViT-B⋆	33.5M	7.2	**84.8**	**97.1**	Wave-ViT-L⋆	57.5M	14.8	**85.5**	**97.3**

from scratch on the training set, and both top-1 and top-5 accuracies metrics are used to evaluate the trained backbones on the validation set. During training, we follow the setups in [69] by applying RandAug [10], CutOut [76], and Token Labeling objective with MixToken [24] for data augmentation. We adopt the AdamW optimizer [39] with a momentum of 0.9. In particular, the optimization process includes 10 epochs of linear warm-up and 300 epochs with cosine decay learning rate scheduler [38]. The batch size is set as 1,024, which is distributed on 8 V100 GPUs. We fix the learning rate and weight decay as 0.001 and 0.05.

Performance Comparison. Table 2 summarizes the performance comparisons between the state-of-the-art vision backbones and our Wave-ViT variants. Note that the most competitive ViT backbones VOLO variants (*i.e.*, VOLO-D1*, VOLO-D2*, and VOLO-D3*) are trained with additional Token Labeling objective with MixToken [24] and convolutional stem (conv-stem) [57] for better patch encoding. We also adopt the same upgraded strategies to train our Wave-ViT, yielding the variants in each size (*i.e.*, Wave-ViT-S*, Wave-ViT-B*, Wave-ViT-L*). Moreover, for fair comparison with other vision backbones without these strategies, we also implement a degraded version of Wave-ViT in Small size without Token Labeling objective and conv-stem (*i.e.*, Wave-ViT-S). As shown in this table, under the similar GFLOPs for each group, our Wave-ViT variants consistently achieve better performances against the existing vision backbones, including both CNN backbones (*e.g.*, ResNet and SE-ResNet), single-scale ViTs (*e.g.*, TNT, CaiT, and CrossViT), and multi-scale ViTs (*e.g.*, Swin, Twins-SVT, PVTv2, VOLO). In particular, under the Base size, the Top-1 accuracy score of Wave-ViT-B* can reach 84.8%, which leads to the absolute improvement of 0.6% against the best competitive VOLO-D1* (Top-1 accuracy: 84.2%). Moreover, when removing the upgraded strategies as in VOLO for training, our Wave-ViT-S still manages to outperform the best multi-scale ViT in Small size (PVTv2-B2). These results generally demonstrate the key advantage of unifying self-attention learning and invertible down-sampling with wavelet transforms to facilitate visual representation learning. Most specifically, under the same Large size, compared to ResNet-152 and SE-ResNet-152 that solely capitalize on CNN architectures, the single-scale ViTs (*e.g.*, TNT-B, CaiT-S36, and CrossViT-15-384) outperform them by capturing long-range dependency via Transformer structure. However, the performances of CaiT-S36 and CrossViT-15-384 are still lower than most multi-scale ViTs (PVTv2-B5 and VOLO-D3*) that aggregates multi-scale contexts for image recognition. Furthermore, instead of using irreversible down-sampling for self-attention learning in PVTv2-B5, our Wave-ViT-L* enables invertible down-sampling with wavelet transforms, and thus achieves better efficiency-vs-accuracy trade-off. It is worthy to note that VOLO-D3* does not employ down-sampling operations to reduce computational cost for high-resolution inputs, but instead directly reduces the input resolution (28×28) at initial stage. In contrast, Wave-ViT-L* keeps the high-resolution inputs (56×56), and exploits wavelet transforms to trigger lossless down-sampling for multi-scale self-attention learning, leading to performance boosts.

4.2 Object Detection and Instance Segmentation on COCO

Dataset and Optimization Setups. Here we examine the pre-trained Wave-ViT's behavior on COCO for two tasks that localize objects ranging from bounding-box level to pixel level, *i.e.*, object detection and instance segmentation. Two mainstream detectors, *i.e.*, RetinaNet [31] and Mask R-CNN [20], are employed for each downstream task, and we replace the CNN backbones in each detector with our Wave-ViT for evaluation. Specifically, each vision backbone is first pre-trained over ImageNet1K, and the newly added layers are initialized with Xavier [17]. Next, we follow the standard setups in [35] to train all models on the COCO train2017 (∼118K images). Here the batch size is set as 16, and AdamW [39] is utilized for optimization (weight decay: 0.05, initial learning rate: 0.0001). All models are finally evaluated on the COCO val2017 (5K images). For the downstream task of object detection, we report the Average Precision(AP) at different IoU thresholds and for three different object sizes (*i.e.*, small, medium, large (S/M/L)). For the downstream task of instance segmentation, both bounding box and mask Average Precision (*i.e.*, AP^b, AP^m) are reported. During training, we resize each input training image by fixing the shorter side as 800 pixels and meanwhile making the longer side not exceeding 1,333 pixels. Note that for RetinaNet and Mask R-CNN, 1 × training schedule

Table 3. The performances of various vision backbones on COCO val2017 for object detection and instance segmentation tasks. For object detection, we employ RetinaNet as the object detector, and the Average Precision(AP) at different IoU thresholds or three different object sizes (*i.e.*, small, medium, large (S/M/L)) are reported for evaluation. For instance segmentation, we adopt Mask R-CNN as the base model, and the bounding box and mask Average Precision (*i.e.*, AP^b and AP^m) are reported for evaluation. We group all vision backbones into two categories: Small size and Base size.

Backbone	RetinaNet 1x [31]						Mask R-CNN 1x [20]					
	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet50 [21]	36.3	55.3	38.6	19.3	40.0	48.8	38.0	58.6	41.4	34.4	55.1	36.7
Swin-T [35]	41.5	62.1	44.2	25.1	44.9	55.5	42.2	64.6	46.2	39.1	61.6	42.0
Twins-SVT-S [9]	43.0	64.2	46.3	28.0	46.4	57.5	43.4	66.0	47.3	40.3	63.2	43.4
RegionViT-S [6]	43.9	–	–	–	–	–	44.2	–	–	40.8	–	–
PVTv2-B2 [59]	44.6	65.6	47.6	27.4	48.8	58.6	45.3	67.1	49.6	41.2	64.2	44.4
Wave-ViT-S	**45.8**	**67.0**	**49.4**	**29.2**	**50.0**	**60.8**	**46.6**	**68.7**	**51.2**	**42.4**	**65.5**	**45.8**
ResNet101 [21]	38.5	57.8	41.2	21.4	42.6	51.1	40.4	61.1	44.2	40.4	61.1	44.2
ResNeXt101-32x4d [64]	39.9	59.6	42.7	22.3	44.2	52.5	41.9	62.5	45.9	37.5	59.4	40.2
Swin-S [35]	44.5	65.7	47.5	27.4	48.0	59.9	44.8	66.6	48.9	40.9	63.4	44.2
Twins-SVT-B [9]	45.3	66.7	48.1	28.5	48.9	60.6	45.2	67.6	49.3	41.5	64.5	44.8
RegionViT-B [6]	44.6	–	–	–	–	–	45.4	–	–	41.6	–	–
PVTv2-B3 [59]	45.9	66.8	49.3	28.6	49.8	61.4	47.0	68.1	51.7	42.5	65.7	45.7
Wave-ViT-B	**47.2**	**68.2**	**50.9**	**29.7**	**51.4**	**62.3**	**47.6**	**69.1**	**52.4**	**43.0**	**66.4**	**46.0**

Table 4. The performances of various vision backbones on COCO val2017 for object detection. Four kinds of object detectors, *i.e.*, GFL [28], Sparse RCNN [49], Cascade Mask R-CNN [3], and ATSS [73] in mmdetection [7], are adopted for evaluation. We report the bounding box Average Precision (AP^b) in different IoU thresholds.

Backbone	Method	AP^b	AP^b_{50}	AP^b_{75}	Backbone	Method	AP^b	AP^b_{50}	AP^b_{75}
ResNet50 [21]	GFL [28]	44.5	63.0	48.3	ResNet50 [21]	Sparse R-CNN [49]	44.5	63.4	48.2
Swin-T [35]		47.6	66.8	51.7	Swin-T [35]		47.9	67.3	52.3
PVTv2-B2 [59]		50.2	69.4	54.7	PVTv2-B2 [59]		50.1	69.5	54.9
Wave-ViT-S		**50.9**	**70.2**	**55.4**	Wave-ViT-S		**50.7**	**70.4**	**55.5**
ResNet50 [21]	Cascade Mask R-CNN [3]	46.3	64.3	50.5	ResNet50 [21]	ATSS [73]	43.5	61.9	47.0
Swin-T [35]		50.5	69.3	54.9	Swin-T [35]		47.2	66.5	51.3
PVTv2-B2 [59]		51.1	69.8	55.3	PVTv2-B2 [59]		49.9	69.1	54.1
Wave-ViT-S		**52.1**	**70.7**	**56.6**	Wave-ViT-S		**50.7**	**69.8**	**55.5**

(*i.e.*, 12 epochs) is adopted to train the two mainstream detectors. In addition to RetinaNet, we also include four state-of-the-arts detectors (GFL [28], Sparse RCNN [49], Cascade Mask R-CNN [3], and ATSS [73]) for object detection task. Following [35,59], we utilize 3 × schedule (*i.e.*, 36 epochs) with multi-scale strategy for training, and the shorter side of each input image is randomly resized within the range of [480, 800] while the longer side is forced to be less than 1,333 pixels.

Performance Comparison. Table 3 lists the performance comparisons across different pre-trained vision backbones under the base detector of RetinaNet and Mask R-CNN for object detection and instance segmentation, respectively. Note that we follow the evaluation for image recognition by grouping all the pre-trained backbones into two categories (*i.e.*, Small size and Base size). As shown in this table, the performance trends in each downstream task are similar to those in image recognition task. Concretely, under the similar model size for each group, the multi-scale ViT backbones (*e.g.*, Swin-T/S and PVTv2-B2/B3) consistently exhibit better performances than CNN backbones (*e.g.*, ResNet50/101) across all evaluation metrics. Furthermore, by capitalizing on wavelet transforms to enable lossless down-sampling in multi-scale self-attention learning, Wave-ViT variants outperform PVTv2-B2/B3 that explore sub-optimal down-sampling with pooling kernels. The results confirm that unifying self-attention learning and lossless down-sampling with wavelet transforms can improve the transfer capability of pre-trained multi-scale representations on dense prediction tasks.

To further verify the generalizability of the pre-trained multi-scale features via Wave-ViT for object detection, we evaluate various pre-trained vision backbones on four state-of-the-arts detectors (GFL, Sparse RCNN, Cascade Mask R-CNN, and ATSS). Table 4 shows the detailed performances of four object detectors with different pre-trained vision backbones under Small size. Similar to the observations in the base detector of RetinaNet, our Wave-ViT-S achieves consistent performance gains against both CNN backbone (ResNet50) and multi-scale ViT backbones (Swin-T and PVTv2-B2) across all the four state-of-the-

Table 5. The performances of various vision backbones on ADE20K validation dataset for semantic segmentation. We employ the commonly adopted base model (UPerNet) and report the mean IoU (mIoU) averaged over all classes for evaluation. We group all vision backbones into two categories: Small size and Base size.
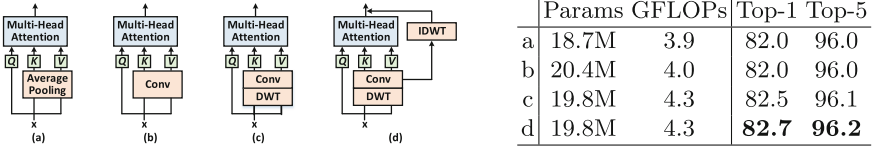
Small			Base		
Method	Backbone	mIoU	Method	Backbone	mIoU
UPerNet [63]	ResNet-50 [21]	42.8	UPerNet [63]	ResNet-101 [21]	44.9
UPerNet [63]	DeiT-S [51]	43.8	UPerNet [63]	DeiT-B [51]	47.2
DeeplabV3 [8]	ResNeSt-50 [70]	45.1	DeeplabV3 [8]	ResNeSt-101 [70]	46.9
Semantic FPN [25]	PVTv2-B2 [59]	45.2	Semantic FPN [25]	PVTv2-B3 [59]	47.3
UPerNet [63]	RegionViT-S [6]	45.3	UPerNet [63]	RegionViT-B [6]	47.5
UPerNet [63]	Swin-T [35]	45.8	UPerNet [63]	Twins-SVT-B [9]	48.9
UPerNet [63]	Twins-SVT-S [9]	47.1	UPerNet [63]	Swin-S [35]	49.5
UPerNet [63]	Wave-ViT-S	**49.6**	UPerNet [63]	Wave-ViT-B	**51.5**

arts detectors. This again validates the advantage of integrating multi-scale self-attention with invertible down-sampling in our Wave-ViT for object detection.

4.3   Semantic Segmentation on ADE20K

Dataset and Optimization Setups. We next evaluate our pre-trained Wave-ViT in the downstream task of semantic segmentation on ADE20K dataset. This dataset is the most typical benchmark for evaluating semantic segmentation techniques, which consists of 25K images (20K training images, 2K validation images, and 3K testing images) derived from 150 semantic categories. Here we choose the commonly adopted UPerNet [63] as the base model for this task and the CNN backbone in primary UPerNet is replaced with our Wave-ViT. During training, we train the models on 8 GPUs for 160K iterations via AdamW [39] optimizer (batch size: 16, initial learning rate: 0.00006, weight decay: 0.01). Both linear learning rate decay scheduler and a linear warmup of 1,500 iterations are utilized for optimization. The scale of input images is fixed as 512×512. We perform the random horizontal flipping, random photometric distortion, and random re-scaling within the ratio range [0.5, 2.0] as data augmentations. We report the metric of mean IoU (mIoU) averaged over all classes for evaluation. For fair comparison with other vision backbones for semantic segmentation downstream task, we set all the hyperparameters and detection heads as in Swin [35].

Performance Comparison. Table 5 shows the mIoU scores of different pre-trained vision backbones under the base models (*e.g.*, UPerNet, DeeplabV3, Semantic FPN) for semantic segmentation. As in the evaluation for image recognition, object detection, and instance segmentation tasks, we group all the pre-trained backbones into two categories (*i.e.*, Small size and Base size). Similarly, by upgrading multi-scale self-attention learning with Wavelet based invertible down-sampling, our Wave-ViT variants yield consistent gains against both CNN

|	Params	GFLOPs	Top-1	Top-5
a	18.7M	3.9	82.0	96.0
b	20.4M	4.0	82.0	96.0
c	19.8M	4.3	82.5	96.1
d	19.8M	4.3	**82.7**	**96.2**

Fig. 3. Performance comparisons across different ways on designing self-attention blocks with down-sampling in multi-scale ViT backbones (under Small size): (a) self-attention block with irreversible down-sampling operation of average pooling, (b) self-attention block with irreversible down-sampling operation of pooling kernels, (c) a degraded version of Wavelets block that solely equips self-attention block with invertible down-sampling via wavelet transforms (DWT), and (d) the full version of our Wavelets block with inverse wavelet transforms (IDWT).

backbones (*e.g.*, ResNet-50/101 and ResNeSt-50/101) and existing multi-scale ViT backbones (*e.g.*, Swin-T/S and Twins-SVT-S/B). Concretely, under a comparable model size within each group, Wave-ViT-S/B achieves 49.6%/51.5% mIoU on ADE20K validation dataset, which absolutely improves the best competitor Twins-SVT-S/Swin-S (47.1%/49.5%) with 2.5%/2.0%. The results basically demonstrates the superiority of Wave-ViT for semantic segmentation task.

4.4 Ablation Study

We investigate how each design in Wavelets block influences the overall performance on ImageNet1K for image recognition, as summarized in Fig. 3. All the variants here are constructed under Small size for fair comparison.

Block (a) is a typical self-attention block with irreversible down-sampling. By directly operating average pooling over the input keys/values, (a) significantly reduces the computational cost for self-attention learning and the Top-1 score achieves 82.0%. **Block (b)** is another typical self-attention block with irreversible down-sampling via pooling kernels (convolution), rather than average pooling in (a). (b) reduces the spatial dimension of keys/values through pooling kernels, leading to the same performances as in (a). However, the number of parameters is inevitably increased. **Block (c)** can be regarded as a degraded version of our Wavelets block, that solely equips self-attention block with invertible down-sampling based on wavelet transforms (DWT). Compared to the most efficient (a) with irreversible down-sampling, the Top-1 score of (c) increases from 82.0% to 82.5%. This validates the effectiveness of unifying self-attention block and invertible down-sampling without information dropping. **Block (d)** (*i.e.*, the full version of Wavelets block) further upgrades (c) by additionally exploiting inverse wavelet transforms (IDWT) to strengthen outputs with enlarged receptive field. Such design leads to performance boosts in Top-1 and Top-5 scores, with negligible increase in computational cost/memory.
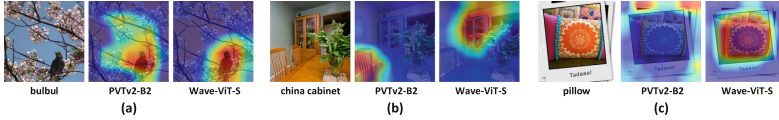
Fig. 4. Visualization of Score-CAM [54] for PVTv2-B2 [59] and our Wave-ViT-S on six images in ImageNet1K dataset.

4.5   Visualization of Learnt Visual Representation

In order to further explain the visual representations learnt by our Wave-ViT, we produce the saliency map through Score-CAM [54] to identify the importance of each pixel in presenting the class discrimination of the input image. Figure 4 visualizes the saliency map derived from the visual representations learnt by two backbones with similar model size (PVTv2-B2 and our Wave-ViT-S). As illustrated in the figure, Wave-ViT-S consistently shows higher concentration at the semantically relevant object than PVTv2-B2, which validates that the representations learnt by Wave-ViT-S are more robust.

5   Conclusions

In this paper, we delve into the idea of unifying typical Transformer module and invertible down-sampling, thereby pursuing efficient multi-scale self-attention learning with lossless down-sampling. To verify our claim, we present a new principled Transformer module, *i.e.*, Wavelets block, that capitalizes on Discrete Wavelet Transform (DWT) to perform invertible down-sampling over keys/values in self-attention learning. In addition, we adopt inverse DWT (IDWT) to reconstruct the down-sampled DWT outputs, which are utilized to strengthen the outputs of Wavelets block by aggregating local contexts with enlarged receptive field. Our Wavelets block is appealing in view that it is feasible to construct multi-scale ViT backbone with Wavelets blocks, with light computational cost/memory budget. In particular, by operating stacked Wavelets blocks over multi-scale features in four-stage architecture, a series of Wavelet Vision Transformer (Wave-ViT) are designed with different model sizes. We empirically validate the superiority of Wave-ViT over the state-of-the-art multi-scale ViTs for several mainstream CV tasks, under comparable numbers of parameters.

References

1. Bae, W., Yoo, J., Chul Ye, J.: Beyond deep residual learning for image restoration: persistent homology-guided manifold simplification. In: CVPR Workshops (2017)
2. Bello, I., Zoph, B., Vaswani, A., Shlens, J., Le, Q.V.: Attention augmented convolutional networks. In: ICCV (2019)
3. Cai, Z., Vasconcelos, N.: Cascade r-cnn: delving into high quality object detection. In: CVPR (2018)

4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
5. Chen, C.F., Fan, Q., Panda, R.: Crossvit: cross-attention multi-scale vision transformer for image classification. In: ICCV (2021)
6. Chen, C.F., Panda, R., Fan, Q.: Regionvit: regional-to-local attention for vision transformers. In: ICLR (2022)
7. Chen, K., et al.: Mmdetection: open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019)
8. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
9. Chu, X., et al.: Twins: revisiting the design of spatial attention in vision transformers. In: NeurIPS (2021)
10. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: practical automated data augmentation with a reduced search space. In: CVPR Workshops (2020)
11. dAscoli, S., Touvron, H., Leavitt, M.L., Morcos, A.S., Biroli, G., Sagun, L.: Convit: improving vision transformers with soft convolutional inductive biases. In: ICML (2021)
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: CVPR (2009)
13. Dosovitskiy, A., et al.: An image is worth 16 x 16 words: transformers for image recognition at scale. In: ICLR (2020)
14. Fan, H., et al.: Multiscale vision transformers. In: ICCV (2021)
15. Fujieda, S., Takayama, K., Hachisuka, T.: Wavelet convolutional neural networks. arXiv preprint arXiv:1805.08620 (2018)
16. Gao, S.H., Cheng, M.M., Zhao, K., Zhang, X.Y., Yang, M.H., Torr, P.: Res2net: a new multi-scale backbone architecture. In: IEEE TPMAI (2021)
17. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS (2010)
18. Guo, T., Seyed Mousavi, H., Huu Vu, T., Monga, V.: Deep wavelet prediction for image super-resolution. In: CVPR Workshops (2017)
19. Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., Wang, Y.: Transformer in transformer. In: NeurIPS (2021)
20. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
22. Hu, H., Zhang, Z., Xie, Z., Lin, S.: Local relation networks for image recognition. In: ICCV (2019)
23. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR (2018)
24. Jiang, Z.H., et al.: All tokens matter: token labeling for training better vision transformers. In: NeurIPS (2021)
25. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: CVPR (2019)
26. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NeurIPS (2012)
27. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE (1998)
28. Li, X., et al.: Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In: NeurIPS (2020)
29. Li, Y., Pan, Y., Yao, T., Mei, T.: Comprehending and ordering semantics for image captioning. In: CVPR (2022)

30. Li, Y., Yao, T., Pan, Y., Mei, T.: Contextual transformer networks for visual recognition. In: IEEE TPAMI (2022)
31. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017)
32. Lin, T.Y., et al.: Microsoft coco: common objects in context. In: ECCV (2014)
33. Liu, L., et al.: Wavelet-based dual-branch network for image demoiréing. In: ECCV (2020)
34. Liu, P., Zhang, H., Zhang, K., Lin, L., Zuo, W.: Multi-level wavelet-cnn for image restoration. In: CVPR Workshops (2018)
35. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: ICCV (2021)
36. Long, F., Qiu, Z., Pan, Y., Yao, T., Luo, J., Mei, T.: Stand-alone inter-frame attention in video models. In: CVPR (2022)
37. Long, F., Qiu, Z., Pan, Y., Yao, T., Ngo, C.W., Mei, T.: Dynamic temporal filtering in video models. In: ECCV (2022)
38. Loshchilov, I., Hutter, F.: Sgdr: stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
39. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
40. Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation. In: IEEE TPAMI (1989)
41. Oyallon, E., Belilovsky, E., Zagoruyko, S.: Scaling the scattering transform: deep hybrid networks. In: ICCV (2017)
42. Pan, Y., et al.: Smart director: an event-driven directing system for live broadcasting. In: ACM TOMM (2021)
43. Pan, Y., Yao, T., Li, Y., Mei, T.: X-linear attention networks for image captioning. In: CVPR (2020)
44. Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., Shlens, J.: Stand-alone self-attention in vision models. In: NeurIPS (2019)
45. Ruan, B.K., Shuai, H.H., Cheng, W.H.: Vision transformers: state of the art and research challenges. arXiv preprint arXiv:2207.03041 (2022)
46. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
47. Srinivas, A., Lin, T.Y., Parmar, N., Shlens, J., Abbeel, P., Vaswani, A.: Bottleneck transformers for visual recognition. In: CVPR (2021)
48. Su, Z., Zhang, H., Chen, J., Pang, L., Ngo, C.W., Jiang, Y.G.: Adaptive split-fusion transformer. arXiv preprint arXiv:2204.12196 (2022)
49. Sun, P., et al.: Sparse r-cnn: end-to-end object detection with learnable proposals. In: CVPR (2021)
50. Szegedy, C., et al.: Going deeper with convolutions. In: CVPR (2015)
51. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML (2021)
52. Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H.: Going deeper with image transformers. In: ICCV (2021)
53. Vaswani, A., et al.: Attention is all you need. In: NeurIPS (2017)
54. Wang, H., et al.: Score-cam: score-weighted visual explanations for convolutional neural networks. In: CVPR workshops (2020)
55. Wang, J., et al.: Deep high-resolution representation learning for visual recognition. In: IEEE TPAMI (2020)
56. Wang, J., et al.: Objectformer for image manipulation detection and localization. In: CVPR (2022)

57. Wang, P., et al.: Scaled relu matters for training vision transformers. arXiv preprint arXiv:2109.03810 (2021)
58. Wang, R., et al.: Bevt: bert pretraining of video transformers. In: CVPR (2022)
59. Wang, W., et al.: Pvtv 2: improved baselines with pyramid vision transformer. arXiv preprint arXiv:2106.13797 (2021)
60. Wang, W., et al.: Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In: ICCV (2021)
61. Williams, T., Li, R.: Wavelet pooling for convolutional neural networks. In: ICLR (2018)
62. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: Cvt: introducing convolutions to vision transformers. In: ICCV (2021)
63. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: ECCV (2018)
64. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: CVPR (2017)
65. Xu, W., Xu, Y., Chang, T., Tu, Z.: Co-scale conv-attentional image transformers. In: ICCV (2021)
66. Yang, J., et al.: Focal self-attention for local-global interactions in vision transformers. arXiv preprint arXiv:2107.00641 (2021)
67. Yao, T., Li, Y., Pan, Y., Wang, Y., Zhang, X.P., Mei, T.: Dual vision transformer. arXiv preprint arXiv:2207.04976 (2022)
68. Yuan, L., et al.: Tokens-to-token vit: training vision transformers from scratch on imagenet. In: ICCV (2021)
69. Yuan, L., Hou, Q., Jiang, Z., Feng, J., Yan, S.: Volo: vision outlooker for visual recognition. arXiv preprint arXiv:2106.13112 (2021)
70. Zhang, H., et al.: Resnest: split-attention networks. arXiv preprint arXiv:2004.08955 (2020)
71. Zhang, H., Hao, Y., Ngo, C.W.: Token shift transformer for video classification. In: ACM Multimedia (2021)
72. Zhang, R.: Making convolutional networks shift-invariant again. In: ICML (2019)
73. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: CVPR (2020)
74. Zhang, Y., Pan, Y., Yao, T., Huang, R., Mei, T., Chen, C.W.: Exploring structure-aware transformer over interaction proposals for human-object interaction detection. In: CVPR (2022)
75. Zhao, H., Jia, J., Koltun, V.: Exploring self-attention for image recognition. In: CVPR (2020)
76. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: AAAI (2020)
77. Zhou, B., et al.: Semantic understanding of scenes through the ade20k dataset. In: IJCV (2019)
78. Zhou, D., et al.: Deepvit: towards deeper vision transformer. arXiv preprint arXiv:2103.11886 (2021)