PWLT: Pyramid Window-based Lightweight Transformer for Image Classification

Yuwei Mo[a,b,*], Pengfei Zuo[a,b,*], Quan Zhou[a,b,**], Zhiyi Mo[b], Yawen Fan[a], Suofei Zhang[c], Bin Kang[c]

[a]National Engineering Research Center of Communications and Networking, Nanjing University of Posts & Telecommunications, Nanjing, P. R. China
[b]Guangxi Colleges and Universities Key Laboratory of Intelligent Software, Wuzhou University, Wuzhou, P. R. China
[c]Department of Internet of Things, Nanjing University of Posts & Telecommunications, Nanjing, P. R. China

Abstract

Recently, vision Transformers (ViTs) have achieved remarkable progress for image classification. As the computational cost of self-attention adopted in ViTs is quadratic with respect to the number of input tokens, some window-based ViTs have been proposed to solve this issue. However, these methods limit the computation of self-attention into a spatial-constrained local windows, losing capability to encode image-based global interactions. Additionally, using fixed-size window always suffers the limitation of single-scale representation that is unsuitable for object recognition with variable scales. To address these problems, this paper describes a Pyramid Window-based Lightweight Transformer, namely PWLT, for image classification. Specifically, to address the need for multi-scale information, we employ windows of different sizes to encode objects with varying scales. To restore the relationships between different windows and explore global context, we introduce a dual self-attention scheme that utilizes local-to-global attention to reestablish these relationships. The extensive experiments on ImageNet-1K and CIFAR100 datasets demonstrate the effectiveness of our PWLT for image classification.

Keywords:
Image Classification, Lightweight Vision Transformer, Pyramid Window, Self-attention

1. Introduction

In last decade, Convolutional Neural Networks (CNNs) have played a significant role in the field of computer vision. A series of prominent network architectures, such as VGG [1], ResNet [2] and EfficientNet [3] have convincingly demonstrated the exceptional effectiveness of CNNs. However, traditional CNNs primarily focus on encoding local features using filtering kernels, overlooking the capture of global information across the entire image. In contrast, recent Transformers[4, 5, 6] have achieved remarkable success in establishing long-range dependencies among features through the use of the multi-head self-attention (MHSA) mechanism. These approaches have demonstrated outstanding performance in tasks such as image classification[7], object detection[8], and semantic segmentation[9]. The Vision Transformer (ViT)[5], as demonstrated in Fig. 1 (a), firstly designs Transformer-based backbone for image classification task, and has achieved remarkable performance. However, the computational cost of MHSA is proportional to the square of the number of input tokens, limiting Transformer-based backbone to be easily deployed for image classification [7] and downstream tasks [8, 9]. In order to mitigate this problem, several window-based ViTs, such as Swin [10] shown in Fig. 1 (b), have been proposed. These approaches adopt a localized window strategy for self-attention calculations, effectively speeding up the computation of MHSA. Nevertheless, there are two limitations associated with this method[6, 10, 11]. Firstly, they are short to capture global context across the entire scene, which can be attributed to their primary emphasis on capturing localized infor-

*The authors are with equal contributions.
**Corresponding author
Email address: quan.zhou@njupt.edu.cn (Quan Zhou)

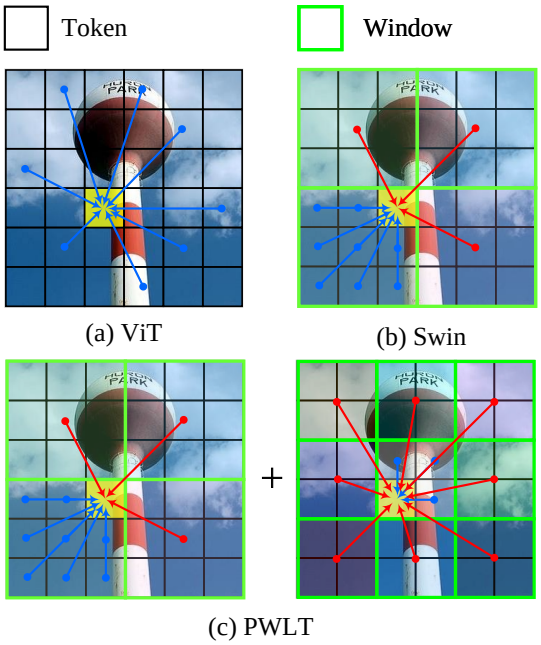

Figure 1: The comparison of ViT [5], Swin [10], and PWLT. The blue and red arrows denote information originating from tokens and windows, respectively.(a) ViT calculates the similarity among all tokens to encode global interaction.(For visual appeal, we have drawn only a few arrows pointing to the yellow token. However, in reality, every token is related to the yellow token.) (b) Swin confines attention calculation to local windows to reduce computational complexity, and employs the shift window strategy to establish global context. (c) PWLT not only establishes the relationship among all tokens but also employ window pyramid to encode multi-scale dependencies.(Best viewed in color)

mation within predefined windows. Secondly, relying solely on a single-scale window is inadequate for extracting multi-scale features, particularly considering the variations in scale that commonly exist among objects in images.

To address these problems, this paper designs a novel Transformer backbone, called Pyramid Window-based Lightweight Transformer (PWLT), for image classification. As shown in Fig. 1 (c), PWLT establishes interconnections among windows and employs a pyramid window scheme to effectively encode information at multiple scales. Specifically, for the first issue, we propose a Dual Self-Attention(DSA) scheme, consisting of two major steps: Window-based Self-Attention (WSA), and Pooling-based Self-Attention (PSA). The WSA, which confines attention into windows, can effectively establish relationships within these localized regions. However, as previously mentioned, this method fails to capture global context. Therefore, the PSA follows, which utilizes the tokens pooling from the windows to establish the relationship between these areas. Due to the smaller number of windows, only a relatively small amount of computations are required. For the second issue, we propose to use different size windows in various attention head groups to capture scale-specific features. In contrast to introduce parallel pathways[12, 13], our method simplifies the network architecture. In nutshell, the major contributions of our paper are three-fold:

- A novel module DSA, containing two components: WSA and PSA, is designed to capture global context and keep low computational cost. WSA and PSA both follow the paradigm of self-attention, where WSA restricts the computation of MHSA within the divided window, while PSA recovers window connections to capture image-based global interactions.
- Instead of using single-scale features [10, 11], a window pyramid is adopted in PWLT to encode feature representation from multiple scales. In addition, rather than employing multi-path ViT [12, 13] that have sophisticated architecture, window pyramid is constructed in each group of heads, significantly simplifying network structure.
- We have evaluated our PWLT on two widely-used image classification datasets: ImageNet-1K [7] and CIFAR100 [14]. The exhausted experimental results demonstrate that our method achieves promising trade-off in terms of classification accuracy and implementing efficiency. Specifically, PWLT achieves 79.2%/82.1%/83.2% Top-1 accuracy on ImageNet-1K[7], yet with only 12.2/29.8/52.2M model size and 1.8/4.9/9.3 GFLOPs, respectively. PWLT achieves 78.4% Top-1 accuracy on CIFAR100[14] with 11.8M model size and 0.5GFLOPs.

2. Related work

2.1. Window-based Lightweight Transformers

In order to reduce the computational cost of Transformers, researchers have proposed numerous window-based Transformer backbones [10, 11, 6, 15]. Generally, these approaches confine the calculations of MHSA within pre-defined windows,

avoiding global computations over all feature tokens, and thus effectively alleviating the computational burdens. Swin [10] first evenly divides the image into non-overlapped windows and limits the MHSA computation in these independent windows, leading to the linear computational complexity of MHSA. However, it relies on the shifting window to recover global interactions, resulting in complicated window mask operations. SepViT [6] also restricts attention computation into windows, yet it establishes window connections using window tokens, to some extent overcoming the limitation using shifting windows [10]. VSA [11] employs a window regression module to predict the size and location of target window, which is adaptive to objects with different sizes. Shuffle Transformer [15] is a CNN-Transformer hybrid that introduces the depth-wise convolution [16] to complement the spatial shuffle for enhancing neighbor-window connections. NAT [17] limits the computation of MHSA for each query to its nearest neighboring tokens, which is essentially equivalent to the window-based Transformer. DaViT [18] leverages WSA and channel self-attention to capture global context while maintaining fast running speed. Couplformer[19] proposes to decouples the attention map into submatrices and generates the alignment scores from spatial information to improve time and memory efficiency.

Unlike these approaches that only investigate local relationships or employ complex operations, e.g., shifting windows, to encode global context, we explore window interactions using pooling-based tokens. The token used in SepViT[6] is similar with our PSA, but it is initialized randomly at first and then rectified throughout the training process. Moreover, instead of relying solely on a single-scale window representation, PWLT generates a window pyramid, allowing for the adaptive extraction of multi-scale information.

2.2. Multi-scale information in Transformer

Multi-scale information has been successfully deployed in Transformers [13, 20, 21]. For example, Shunted Transformer [21] merges tokens to represent larger object features while keeping certain tokens to preserve fine-grained features. This novel scheme enables the self-attention to learn relationships between objects with different scales. P2T [20] adopts an adaptive pyramid pooling to MHSA, capturing powerful contextual features. MPViT [12] simultaneously embeds features of patches that have different sizes using overlapping convolution patch embedding, enabling both fine and coarse feature representations at the same level. An alternative approaches to integrate multi-scale information is using image pyramid [13, 22]. For instance, MMViT [13] encodes input at different resolutions in parallel from multiple views perspective. CrossViT [22] designs a dual-branch Transformer backbone to produce stronger features by combining image patches of different sizes.

Despite achieving impressive classification results, advanced Transformers employ a multi-path architecture to introduce multi-scale information, requiring very high computational cost. In contrast, our PWLT allocates windows of different size to separate groups of attention heads for the extraction of multi-scale features, resulting in significant computational savings.

3. Our Method

This section firstly introduces the DSA, used to capture global context. Thereafter, we elaborate on the detailed architecture of PWLT.

3.1. Dual Self-Attention (DSA)

To effectively and efficiently capture multi-scale information, the DSAs layer is built to establish the window pyramid. Specifically, in this layer, each DSA works in different group of heads and captures the feature with distinct window partition criterion, such as 2×2, 3×3 and so on. Although all groups of heads have distinct window size, DSA works in a similar way. Without loss of generality, we thus only introduce DSA using 2×2 window division. As shown in Fig. 2, DSA consists of two components, WSA and PSA. Firstly, the input features are fed into WSA to calculate the local attention in windows, which is beneficial to capture local dependencies. And then, the outputs of WSA pass through PSA, where window interactions are well established to capture global context. Immediately below, we elaborate on the details of WSA and PSA, respectively.

3.1.1. WSA

The computation of WSA follows traditional window-based Transformers [10, 12]. More specifically, given input feature $\mathrm{X} \in \mathbb{R}^{H \times W \times C}$, where H, W and C stand for the height, width, and channel number of X, respectively. As shown in Fig. 2,

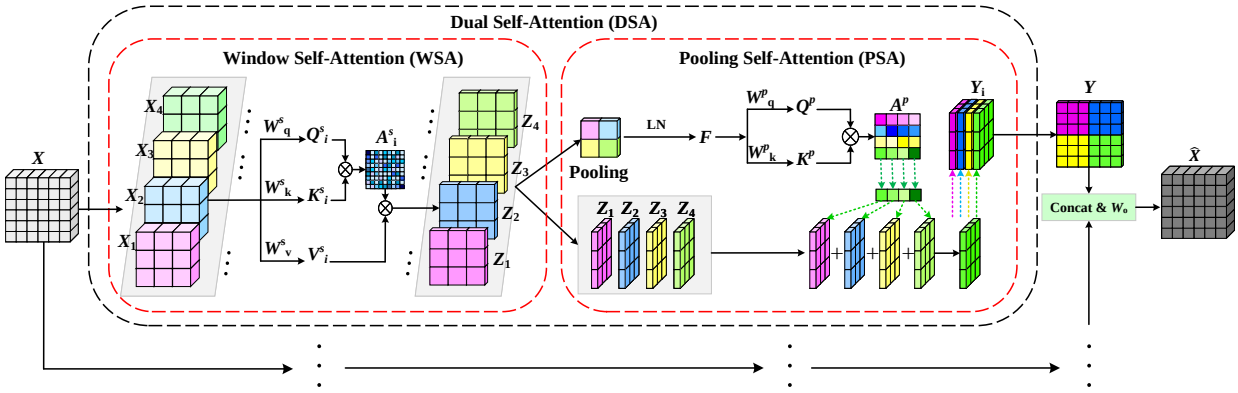

Figure 2: An illustration of DSA. DSA consists of two components: WSA and PSA. Note DSA is performed within group of heads, where input features may be partitioned into different number of windows, resulting in pyramid window architecture of PWLT. LN denotes layer normalization. (Best viewed in color)

input feature X is evenly divided into 2×2 windows, where each window corresponds to a feature block $\mathrm{X}_i \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times C}, i \in \{1,2,3,4\}$. Following [4], the self-attention is independently calculated in each window. Specifically, three linear projections $\{\mathrm{W}_q^s \in \mathbb{R}^{C \times d}, \mathrm{W}_k^s \in \mathbb{R}^{C \times d}, \mathrm{W}_v^s \in \mathbb{R}^{C \times d}\}$ are firstly used to map X_i into three low-dimensional embeddings, including query $\mathrm{Q}_i^s \in \mathbb{R}^{m \times d}$, key $\mathrm{K}_i^s \in \mathbb{R}^{m \times d}$, and value $\mathrm{V}_i^s \in \mathbb{R}^{m \times d}$:

$$\mathrm{Q}_i^s = \mathrm{Re}(\mathrm{X}_i)\mathrm{W}_q^s,\ \mathrm{K}_i^s = \mathrm{Re}(\mathrm{X}_i)\mathrm{W}_k^s,\ \mathrm{V}_i^s = \mathrm{Re}(\mathrm{X}_i)\mathrm{W}_v^s,\ \forall i \tag{1}$$

where $\mathrm{Re}(\cdot)$ denotes reshape operation, m is the number of tokens in X_i, and $d << C$ indicates the number of channels in each group of heads. Note projections $\{\mathrm{W}_q^s, \mathrm{W}_k^s, \mathrm{W}_v^s\}$ are shared parameters for all X_i to save model size. Thereafter, an attention map $\mathrm{A}_i^s \in \mathbb{R}^{m \times m}$ that represents the dependencies of tokens in a X_i is produced using $\mathrm{SoftMax}(\cdot)$:

$$\mathrm{A}_i^s = \mathrm{SoftMax}(\frac{\mathrm{Q}_i^s \mathrm{K}_i^{s\top}}{\sqrt{d}}) \tag{2}$$

Finally, the output intermediate feature $\mathrm{Z}_i \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times d}$ of WSA is produced by reweighting V_i^s with attention map A_i^s:

$$\mathrm{Z}_i = \mathrm{Re}(\mathrm{A}_i^s \mathrm{V}_i^s) \tag{3}$$

So far, we have successfully extracted local features using WSA. However, there is a lack of information interactions between different windows. Therefore, we introduce the PSA module to establish these window connections.

3.1.2. PSA

Given the intermediate features $\mathrm{Z}_i, i \in \{1,2,3,4\}$ derived from WSA, a global pooling is firstly used to produce four window-based token features that represent the global information of each window, respectively. Collecting all token features together, PSA is conducted by sequentially feeding token features into layer normalization, reshape module, linear projection, attention calculation, and feature reweighting. Let $\mathrm{F} \in \mathbb{R}^{4 \times d}$ be normalized token features, where 4 stands for the number of divided windows. Then, F undergoes two linear projections $\{\mathrm{W}_q^p \in \mathbb{R}^{d \times d}, \mathrm{W}_k^p \in \mathbb{R}^{d \times d}\}$, resulting in query $\mathrm{Q}^p \in \mathbb{R}^{4 \times d}$ and key $\mathrm{K}^p \in \mathbb{R}^{4 \times d}$, respectively:

$$\mathrm{Q}^p = \mathrm{F}\mathrm{W}_q^p \qquad \mathrm{K}^p = \mathrm{F}\mathrm{W}_k^p \tag{4}$$

Next, an attention map $\mathrm{A}^p \in \mathbb{R}^{4 \times 4}$ that encodes the dependencies of windows is also produced using a softmax function:

$$\mathrm{A}^p = \mathrm{SoftMax}(\frac{\mathrm{Q}^p \mathrm{K}^{p\top}}{\sqrt{d}}) \tag{5}$$

As each row of A^p encodes the relations between one specific window to all windows, the final step of PSA is to update intermediate features Z_i, producing reweighted features $\mathrm{Y}_i \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times d}$ that harvested information from all windows through A^p:

$$\mathrm{Y}_i = \sum_j \mathrm{A}_{ij}^p \mathrm{Z}_j \qquad i, j \in \{1,2,3,4\} \tag{6}$$

where A_{ij}^p denotes the element of i^{th} row and j^{th} column in A^p. Putting all Y_i together produces features $\mathrm{Y} \in \mathbb{R}^{H \times W \times d}$ as the output of DSA. Recalling

that DSAs are actually conducted on window pyramid within different group of heads, thus we have to concatenate all groups together and feed them into a linear projection $\mathrm{W}_o \in \mathbb{R}^{C\times C}$, resulting in the final output $\hat{\mathrm{X}} \in \mathbb{R}^{H\times W\times C}$ that has the same shape of input feature X.

3.2. Complexity Analysis

In the previous section, we have elaborated on the implementation of DSA. To better illustrate the advantages of DSA, this section provides an analysis of the parameter count and computation cost in DSA. It is worth mentioning that we assume the size of input feature map is $N \times C$, where N and C represent the number of tokens and channels, respectively.

3.2.1. Parameters in DSA

As shown in Fig. 2, the parameters in DSA primarily originate from three components: WSA, PSA, and the linear projection layer of W_o. In WSA, the total number of parameters is $3C^2$, encompassing contributions from W_q^s,W_k^s and W_v^s. The parameter count of PSA is upon the number of groups of attention heads. Assuming that there are D groups and each group consists of C/D channels, the parameter count of PSA is $2C^2/D$(there are $2C^2/D^2$ parameters in each group). The linear mapping projection W_o has a parameter count of C^2. Consequently, the overall parameter count of DSA is $(4 + 2/D)\,C^2$. Compared with the parameter count of $4C^2$ in ViT[5] and Swin[10], our DSA has caused an extra parameter, but it is negligible and makes our model establish the relationship between different windows without shifting operations.

3.2.2. Computational cost of DSA

Computational cost plays a crucial role in determining the efficiency of a network. The computation complexity of MHSA in ViT[5] is $4NC^2+2N^2C$. Specifically, $4NC^2$ is the cost of generating query, key, value and the computation from W_o. $2N^2C$ is the major cost of MHSA and indicates the computation involved in generating the attention map and performing the multiplication with the value matrix.

Compared with MHSA, our DSA decreases the major cost from $2N^2C$ to $2N^2C/m$. Therefore, our method has effectively decreased the workload of ViT. Specifically, the computational workload of DSA primarily arises from WSA, PSA, and the linear mapping layer W_o. For simplicity of representation, we assume that all heads are divided into m windows. The cost in WSA is $3NC^2 + 2N^2C/m$, where the $3NC^2$ is the cost of generating query Q_i^s, key K_i^s and value V_i^s. Otherwise, $2N^2C/m$ is the cost of generating attention map and performing the multiplication with the value matrix in m windows(each window is $2N^2C/m^2$). The cost in PSA is $2mC^2 + m^2C + mNC$. $2mC^2$ indicates the cost of generating query $\mathrm{Q}^p$and key K^p . m^2C is the cost of generating the attention map. mNC represents the the matrix multiplication between the attention map and the feature maps. Finally, the computation cost in linear projection layer W_o is NC^2. Therefore, the cost of DSA is $(4N + 2m)\,C^2 + \left(2N^2/m + m^2 + mN\right)C$.

However, assuming that there are M tokens in each window, the computation complexity of WSA in Swin[10] is $4NC^2 + 2MNC$. $4NC^2$ indicates the cost of generating query, key, value and the computation from W_o. $2MNC$ is the cost of generating the attention map and performing the multiplication with the value matrix in all windows(each window is $2M^2C$, and there are N/M windows). Although our method has introduced extra computation cost compared with Swin[10], it avoids the use of shifting window operations. Instead, it leverages an attention mechanism to establish comprehensive connections, offering a simpler solution and getting better performance.

3.3. Network Architecture of PWLT

As shown in Fig. 3, PWLT inherits the hierarchical architecture [10, 23] that has four stages, where feature resolutions are gradually reduced, and channel numbers are expanded by factor 2. Therefore, it produces a series of feature maps that have the shape of $\frac{H}{4}\times\frac{W}{4}\times C$, $\frac{H}{8}\times\frac{W}{8}\times 2C$, $\frac{H}{16}\times\frac{W}{16}\times 4C$, $\frac{H}{32}\times\frac{W}{32}\times 8C$, where H and W represent the width and height of input image, and C denotes the number of channels. Except the first stage that contains patch embedding module [10], the rest stages are composed by patch merging module [23] and repeated PWLT blocks. As illustrated in the right panel of Fig. 3, the DSAs layer is equipped with Feed-Forward Networks (FFN), residual connection, and Layer Normalization (LN), following the traditional ViT-based design [5, 10].

Based on the preceding Transformer architecture[6, 10, 21], we have developed PWLT of varying depths by stacking different numbers

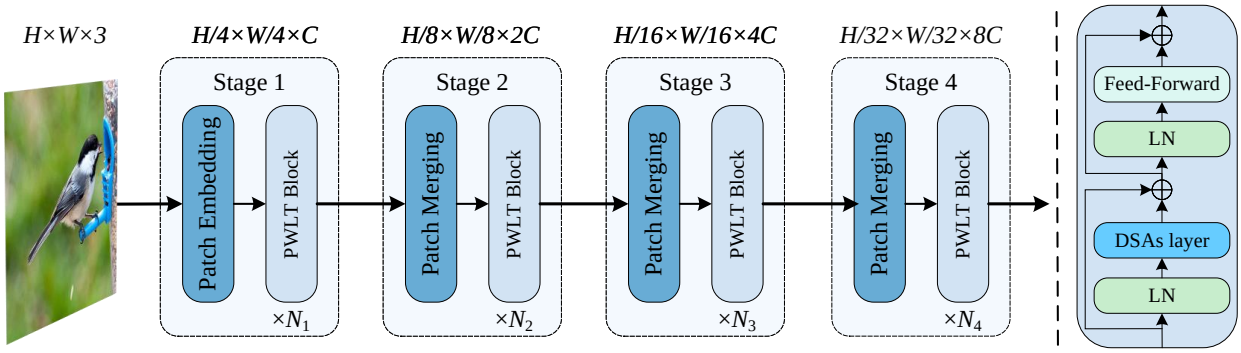

Figure 3: Overview of proposed Pyramid Window-based Light-weight Transformer (PWLT). The architecture is divided into four stages, each of which consists of a patch embedding block or patch merging block and N_i PWLT blocks. The right column is the detailed implementation of PWLT block. (Best viewed in color)

Table 1: The detailed architecture of backbone in proposed PWLT. $Conv(k \times k, s)$ represents convolution using $k \times k$ filter kernel size with stride s. $n \times n$ denotes window partition size on the grid of feature maps. h is the index of head, and D denotes total number of group heads in each PWLT block.

Stage	Input Size	Layer Name	PWLT-Tiny	PWLT-Base	PWLT-Medium
1	224×224	Patch Embedding	$Conv(7 \times 7, s = 4)$		
	56×56	PWLT Block	$\begin{bmatrix} D = 2 \\ 2 \times 2,\ h \in \{1\} \\ 3 \times 3,\ h \in \{2\} \end{bmatrix} \times 1$	$\begin{bmatrix} D = 2 \\ 2 \times 2,\ h \in \{1\} \\ 3 \times 3,\ h \in \{2\} \end{bmatrix} \times 2$	$\begin{bmatrix} D = 2 \\ 2 \times 2,\ h \in \{1\} \\ 3 \times 3,\ h \in \{2\} \end{bmatrix} \times 3$
2	56×56	Patch Merging	$Conv(3 \times 3, s = 2)$		
	28×28	PWLT Block	$\begin{bmatrix} D = 3 \\ 2 \times 2,\ h \in \{1,2\} \\ 3 \times 3,\ h \in \{3\} \\ 5 \times 5,\ h \in \{4\} \end{bmatrix} \times 1$	$\begin{bmatrix} D = 3 \\ 2 \times 2,\ h \in \{1,2\} \\ 3 \times 3,\ h \in \{3\} \\ 5 \times 5,\ h \in \{4\} \end{bmatrix} \times 4$	$\begin{bmatrix} D = 3 \\ 2 \times 2,\ h \in \{1,2\} \\ 3 \times 3,\ h \in \{3\} \\ 5 \times 5,\ h \in \{4\} \end{bmatrix} \times 6$
3	28×28	Patch Merging	$Conv(3 \times 3, s = 2)$		
	14×14	PWLT Block	$\begin{bmatrix} D = 3, C = 64 \\ 2 \times 2,\ h \in \{1 \sim 4\} \\ 3 \times 3,\ h \in \{5,6\} \\ 5 \times 5,\ h \in \{7,8\} \end{bmatrix} \times 4$	$\begin{bmatrix} D = 3 \\ 2 \times 2,\ h \in \{1 \sim 4\} \\ 3 \times 3,\ h \in \{5,6\} \\ 5 \times 5,\ h \in \{7,8\} \end{bmatrix} \times 16$	$\begin{bmatrix} D = 3 \\ 2 \times 2,\ h \in \{1 \sim 4\} \\ 3 \times 3,\ h \in \{5,6\} \\ 5 \times 5,\ h \in \{7,8\} \end{bmatrix} \times 30$
4	14×14	Patch Merging	$Conv(3 \times 3, s = 2)$		
	7×7	PWLT Block	$\lfloor D = 1 \rfloor \times 1$	$\lfloor D = 1 \rfloor \times 2$	$\lfloor D = 1 \rfloor \times 3$

of PWLT blocks at each stage. Consequently, we introduce three variants: PWLT-Tiny, PWLT-Base, and PWLT-Medium, whose difference only comes from the number of layers in different stage. The detailed structure of PWLT backbone is given in Table 1. More specifically, the patch embedding module utilizes a 7×7 convolution operation with a stride of 4, effectively reducing the input resolution by a factor of 4 in a single step, while the patch merging module employs a 3×3 convolution operation with a stride of 2, resulting in a down-sampling of the input size by a factor of 2. Each PWLT block consists of multiple groups of attention heads, which are employed to construct a window pyramid and perform DSA computations. Taking stage 2 as an example, except the first two heads which are regarded as a group and use 2×2 window partition, the rest two heads individually form groups, corresponding to the window partition size of 3×3 and 5×5, respectively. In the final stage, as the input resolution of features is too small to be divided (e.g., $7 \times 7 \times 512$), we resort to directly employ MHSA instead of DSA in this stage.

4. Experiments

To evaluate our PWLT, we have conducted exhausted experiments on ImageNet-1K [7] and CIFAR100 [14] datasets.

4.1. Datasets and Evaluation metrics

4.1.1. Datasets

ImageNet-1K [7] is the most popular dataset for image classification. It involves 1,000 categories with approximately 1,000 images per category. Specifically, this dataset contains 1.28 million training, and 50K testing images. Each image in this dataset varies in terms of object types, scale and backgrouds, making it a challenging and diverse dataset for evaluating the performance of image classification models. In contrast, CIFAR100 [14] is a smaller dataset that only includes 100 classes and 60K color images with resolution 32×32, which are divided into 50K/10K images for training and testing, respectively. Despite the reduced number of categories compared to ImageNet-1K, CIFAR100 compensates with a concentrated selection and lower image resolution, introducing a distinctive set of challenges for image classification models.

4.1.2. Evaluation metrics

For fair comparison with other state-of-the-art lightweight methods, we employ the standard evaluation metric of Top-1 recognition accuracy which measures the proportion of correct predictions out of the total number of predictions made by the model. It provides a straightforward evaluation of the model's ability to correctly classify instances into their respective categories. On the other hand, the commonly-used floating-point operations per second (FLOPs) is used to measure implementation efficiency.

4.2. Implementation Details

4.2.1. Training settings

PWLT is implemented in the hard-ware server platform with 8 RTX 3090 GPU cards. The software code is based on the MMPretrain toolbox that is an open-source repository for image classification. The widely-used AdamW [6] is employed to optimize PWLT, where the weight decay and initial learning rate are set to 0.05 and 1×10^{-3} respectively. Otherwise, the cosine learning policy is adopted with the minimum learning rate 1×10^{-5}. We trained our model for 300 epochs on ImageNet-1K [7] and CIFAR100 [14], where the first five epochs are used to warm up. To enhance data diversity and improve generalization, various data augmentation techniques are utilized, including random cropping, random flipping, Mixup[23] and CutMix[24] and random erasing[10].

4.2.2. Loss settings

Label Smoothing Loss (LSL) [10] is employed to supervise the entire PWLT. This loss function merges cross-entropy [2]with a label smoothness term, which generates target labels by averaging the probability distribution of the true label with those of other classes. By doing so, models are often able to achieve better generalization performance, particularly in scenarios with noisy or uncertain data. Following is the implementation details:

$$\mathcal{L} = (1 - \epsilon) \cdot \mathcal{L}_{CE} - \epsilon / N \sum \log y_{pred} \tag{7}$$

where $\mathcal{L}_{CE}$ is the cross entropy Loss, ϵ denotes non-negative smoothing parameter that leverage the trade-off between $\mathcal{L}_{CE}$ and smoothness term, N indicates the number of classes, and y_{pred} stands for the predicted probabilities.

4.3. Results on ImageNet-1K

Table 2 reports comparison results with selected state-of-the-art networks. To increase diversity of lightweight backbones, CNN-based [2, 31], and Transformer-based [23, 26, 10, 33] approaches are both adopted. Obviously, PWLT outperforms CNN-based models such as ResNet[2] and RegNetY[31]. For instance, with similar model size and FLOPs, the top-1 accuracy of PWLT-T/B/M is 9.6%/5.9%/3.4% and 1.2%/2.7%/3.3% better than ResNet32/50/101 and RegNetY-1.6G/4G/8G, respectively. Furthermore, PWLT also achieves superior results compared with recent Transformer-based models. For example, PWLT-T/B/M is 2.3%/0.9%/0.1% better than Swin/T/S[10] with similar model size and FLOPs. Compared with SepViT[6], our PWLT-T achieves better performance which can be attributed to that our model capture multi-scale information. Some methods have smaller model size and GFLOPs(e.g., DeiT-T/S/B and XCiT-T/S12/S24), but their performance has 7.0%/2.2%/1.4% and 2.1%/0.1%/0.6% drops compared with PWLT-T/B/M.

Table 2: Comparison with state-of-the-art methods in terms of classification accuracy and implementing efficiency on ImageNet-1K dataset [7]. For fair comparison, the input resolutions of all methods are fixed to 224×224.

Method	Year	Param (M)	FLOPs (G)	Top-1 (%)
ResNet18[2]	CVPR2016	11.7	1.8	69.8
DeiT-T[24]	ICML2021	5.7	1.3	72.2
PVT-T[23]	ICCV2021	13.2	1.9	75.1
MoCoVIT-D[25]	ARXIV2022	12.1	0.2	75.5
ConViT-T[26]	ICML2021	10.0	2.0	76.7
ViL-T [27]	ICCV2021	6.7	1.3	76.7
Swin[10]	ICCV2021	11.0	1.5	76.9
XCiT-T12[28]	NIPS2021	7.0	1.2	77.1
PoolFormer-S12[29]	CVPR2022	11.9	2.0	77.2
Flatten-PVT-T[30]	ICCV2023	12.2	2.0	77.8
RegNetY-1.6G[31]	CVPR2020	11.2	1.6	78.0
LocalViT-PVT[32]	ARXIV2021	13.5	4.8	78.2
PVT-V2-B1[33]	CVM2022	13.1	2.1	78.7
SepViT[6]	ARXIV2022	12.2	1.8	78.8
FasterNet-T[34]	CVPR2023	15.0	1.9	78.9
PWLT-T	-	12.2	1.8	79.2
ResNet50[2]	CVPR2016	25.0	4.1	76.2
RegNetY-4G[31]	CVPR2020	20.6	4.0	79.4
PVT-S[23]	ICCV2021	24.5	3.8	79.8
DeiT-S[24]	ICML2021	22.1	4.6	79.9
Couplformer-T[19]	WACV2023	28.0	6.4	80.5
Swin-T[10]	ICCV2021	28.3	4.5	81.2
ConViT-S[26]	ICML2021	27.0	5.4	81.3
FasterNet-S[34]	CVPR2023	31.1	4.6	81.3
PoolFormer-S36[29]	CVPR2022	31.0	5.0	81.4
Flatten-PVT-S[30]	ICCV2023	21.7	4.0	81.7
XCiT-S12[28]	NIPS2021	26.0	4.8	82.0
PVT-V2-B2[33]	CVM2022	25.4	4.0	82.0
ViL-Small[27]	ICCV2021	24.6	4.9	82.0
PWLT-B	-	29.8	4.9	82.1
ViT-B[5]	ICLR2021	86.6	17.6	77.9
ResNet-101[2]	CVPR2016	45.0	7.9	79.8
RegNetY-8G[31]	CVPR2020	39.2	18.0	79.9
PVT-M[23]	ICCV2021	44.2	6.7	81.2
PVT-L[23]	ICCV2021	61.4	9.8	81.7
DeiT-B[24]	ICML2021	86.6	17.5	81.8
ConVIT-S+[26]	ICML2021	48.0	10.0	82.2
Couplformer-S[19]	WACV2023	49.0	20.4	82.3
PoolFormer-M48[29]	CVPR2022	73.0	11.6	82.5
XCiT-S24[28]	NIPS2021	48.0	9.1	82.6
FasterNet-M[34]	CVPR2023	53.5	8.7	83.0
Swin-S[10]	ICCV2021	49.6	8.7	83.1
PWLT-M	-	52.2	9.3	83.2

4.4. Results on CIFAR100

In this section, we carry out experiments on the CIFAR100 [14] dataset to further evaluate the effectiveness of our method. All the results are reported in Table 3. Whether CNN-based or Transformer-based models, our PWLT is able to achieve best accuracy, yet with comparable even less model size and FLOPs. Particularly, compared with DeiT-S [24], PVT-T [23], and Swin-T [10], PWLT obtains remarkable Top-1 improvement (14.7%, 8.8%, and 0.4%) with less model size(11.8M vs 21.4/15.8/27.5M), and less FLOPs(0.5 vs 5.5/0.6/1.4G). With the same model size and FLOPs, PWLT improves the performance of SepViT[6] from 78.1% to 78.4%.

Table 3: Comparison with state-of-the-art methods in terms of classification accuracy and implementing efficiency on CIFAR100 [14]. '-' denotes that the results are not reported.

Method	Year	Param (M)	FLOPs (G)	Top-1 (%)
ConvNet				
ResNet18[2]	CVPR2016	11.2	1.8	75.6
ResNet34[2]	CVPR2016	21.3	3.7	76.7
SENet34[35]	CVPR2018	21.6	-	77.9
Transformer				
DeiT-S[24]	ICML2021	21.4	5.5	63.7
PVT-T[23]	ICCV2021	15.8	0.6	69.6
Swin-T[10]	ICCV2021	27.5	1.4	78.0
SepViT[6]	ARXIV2022	11.8	0.5	78.1
PWLT	-	11.8	0.5	78.4

Table 4: Ablation studies on the effectiveness of each component in DSA.

WSA	PSA	Params (M)	FLOPs (G)	Top-1(%)
		13.25	2.18	76.8
✓		11.79	1.46	75.1
✓	✓	12.36	1.73	76.9

4.5. Ablation Study

In order to understand the underlying behavior of PWLT, this section reports some results of a series of ablation studies.

4.5.1. Ablation studies for WSA and PSA

Table 4 reports some ablation studies on ImageNet-1K [7], which quantify the individual contribution of two main components: WSA and PSA. We first construct the baseline based on traditional ViT-based backbone, where WSA and PSA are not considered. And then, the WSA and PSA are sequentially added. When ViT-based backbone is adopted, it has similar classification accuracy

Table 5: Ablation studies on the pooling method in the PSA. AvgPooling and Conv represent the average pooling and stride convolution, respectively.

Method	Params(M)	FLOPs(G)	Top-1(%)
Conv	12.4	1.8	78.7
AvgPooling	12.2	1.8	79.2

Table 6: Ablation studies on the window pyramid, where the feature maps are split into $n \times n$ window grids.

2×2	3×3	5×5	7×7	Params (M)	FLOPs (G)	Top-1(%)
✓				12.1	0.5	77.12
✓	✓			11.9	0.5	78.07
✓	✓	✓		11.8	0.5	78.40
✓	✓	✓	✓	11.8	0.5	78.20

with PWLT, yet it has the largest number of parameters and FLOPs. When WSA is used to reduce model size (11.79M vs 13.25M) and FLOPs (1.46G vs 2.18G), the performance drops from 76.8% to 75.1%, probably due to the loss of connections between divided windows. When the final module PSA is introduced to recover window dependencies, however, we obtain the comparable results with baseline, yet with only slight growth of model size and FLOPs.

4.5.2. Ablation studies for pooling method in PSA

To assess the impact of the pooling method in PSA on the network, such as average pooling and stride convolution, we have conducted some ablation experiments on ImageNet-1K[7]. The results are shown in Table 5. Obviously, compared with average pooling, stride convolution will introduce additional parameters(12.2M vs 12.4M). However, the average pooling achieves better performance(79.2% vs 78.7%) than stride convolution, which may because the convolution did not effectively extract information from Z_i.

4.5.3. Ablation studies for window pyramid

The number of groups of heads determines how many scales of window are produced, significantly influencing the trade-off between the capability of multi-scale context representation and computational efficiency. Consequently, we evaluate the performance variance along with the changes of window size on CIFAR100 [14] dataset. The results are reported in Table 6. As different-scale window is introduced, the model's performance steadily improves, which implies that the incorporation of multi-scale information effectively aids in enhancing the model's performance. It can be observed that the best trade-off is achieved when only 3 scale windows are employed (2×2, 3×3, and 5×5), thus chosen as default setting in our PWLT. Note employing more scales of window pyramid leads to the decrease of model size, probably due to requiring few parameters used in W_q^p and W_k^p.

5. Conclusion Remarks and Future Work

In this paper, we propose a novel network, called Pyramid Window-based Lightweight Transformer(PWLT), for image classification[7]. In contrast to the previous works, our approach differs in two key aspects. First of all, our method proposes the Dual Self-Attention(DSA) mechanism to reestablish the connection between different windows, facilitating the capture of global features. Secondly, our model incorporates the window pyramid, enhancing the network's ability to extract multi-scale features for the recognition of objects of varying size. In this study, the performance of proposed PWLT has been evaluated on ImageNet-1K and CIFAR100 datasets. The results demonstrate a promising trade-off between classification accuracy and implementing efficiency. In the future, we are interested in introducing PWLT to dense prediction tasks such as object detection[8], and semantic segmentation[9]. Additionally, future research would like to focus on optimizing and enhancing PWLT to further improve its performance across a broader spectrum of computer vision applications.

CRediT authorship contribution statement

Yuwei Mo: Methodology, Formal analysis, Writing-original draft. Pengfei Zuo: Methodology, Formal analysis. Quan Zhou: Conceptualization, Resources, Writing-original draft. Zhiyi Mo: Review & editing-revised draft. Yawen Fan: Review & editing-revised draft. Suofei Zhang: Review & editing-revised draft. Bin Kang: Review & editing-revised draft.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work.

Acknowledgments

The authors would like to thank associated editor and all anonymous reviewers for their helpful and insight comments. This research was supported by the NSFC (No. 61876093,62171232,), Graduate Research & Innovation Projects of Jiangsu Province (No. KYCX22_0962), and open funding project of Guangxi Colleges and Universities Key Laboratory of Intelligent Software (No. 2023B01).

References

[1] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556 (2014).

[2] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[3] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, 2019, pp. 6105–6114.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International conference on learning Representations, 2021.

[6] W. Li, X. Wang, X. Xia, J. Wu, X. Xiao, M. Zheng, S. Wen, Sepvit: Separable vision transformer, arXiv preprint arXiv:2203.15380 (2022).

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

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, 2014, pp. 740–755.

[9] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A. Torralba, Scene parsing through ade20k dataset, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5122–5130.

[10] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 10012–10022.

[11] Q. Zhang, Y. Xu, J. Zhang, D. Tao, Vsa: Learning varied-size window attention in vision transformers, in: European conference on computer vision, 2022, pp. 466–483.

[12] Y. Lee, J. Kim, J. Willette, S. J. Hwang, Mpvit: Multi-path vision transformer for dense prediction, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2022, pp. 7287–7296.

[13] Y. Liu, N. Ong, K. Peng, B. Xiong, Q. Wang, R. Hou, M. Khabsa, K. Yang, D. Liu, D. S. Williamson, et al., Mmvit: Multiscale multiview vision transformers, arXiv preprint arXiv:2305.00104 (2023).

[14] A. Krizhevsky, Learning multiple layers of features from tiny images, 2009.

[15] Z. Huang, Y. Ben, G. Luo, P. Cheng, G. Yu, B. Fu, Shuffle transformer: Rethinking spatial shuffle for vision transformer, arXiv preprint arXiv:2106.03650 (2021).

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

[17] A. Hassani, S. Walton, J. Li, S. Li, H. Shi, Neighborhood attention transformer, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2023, pp. 6185–6194.

[18] M. Ding, B. Xiao, N. Codella, P. Luo, J. Wang, L. Yuan, Davit: Dual attention vision transformers, in: European conference on computer vision, 2022, pp. 74–92.

[19] H. Lan, X. Wang, X. Wei, Couplformer: Rethinking vision transformer with coupling attention map, ArXiv abs/2112.05425 (2021).

[20] Y.-H. Wu, Y. Liu, X. Zhan, M.-M. Cheng, P2t: Pyramid pooling transformer for scene understanding, IEEE Transactions on pattern analysis and machine intelligence 45 (2021) 12760–12771.

[21] S. Ren, D. Zhou, S. He, J. Feng, X. Wang, Shunted self-attention via multi-scale token aggregation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2022, pp. 10853–10862.

[22] C.-F. R. Chen, Q. Fan, R. Panda, Crossvit: Cross-attention multi-scale vision transformer for image classification, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2021, pp. 357–366.

[23] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, L. Shao, Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2021, pp. 568–578.

[24] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in: International conference on machine learning, 2021, pp. 10347–10357.

[25] H. Ma, X. Xia, X. Wang, X. Xiao, J. Li, M. Zheng, Mocovit: mobile convolutional vision transformer, arXiv preprint arXiv:2205.12635 (2022).

[26] S. d'Ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, L. Sagun, Convit: Improving vision transformers with soft convolutional inductive biases, in: International conference on machine learning, 2021, pp. 2286–2296.

[27] P. Zhang, X. Dai, J. Yang, B. Xiao, L. Yuan, L. Zhang, J. Gao, Multi-scale vision longformer: A new vision transformer for high-resolution image encoding, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 2998–3008.

[28] 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, Advances in neural information processing systems 34 (2021) 20014–20027.

[29] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, S. Yan, Metaformer is actually what you need for vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2022, pp. 10819–10829.

[30] D. Han, X. Pan, Y. Han, S. Song, G. Huang, Flatten transformer: Vision transformer using focused linear attention, Proceedings of the IEEE/CVF international conference on computer vision (2023) 5938–5948.

[31] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, P. Dollár, Designing network design spaces, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2020, pp. 10428–10436.

[32] Y. Li, K. Zhang, J. Cao, R. Timofte, L. Van Gool, Localvit: Bringing locality to vision transformers, arXiv preprint arXiv:2104.05707 (2021).

[33] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, L. Shao, Pvt v2: Improved baselines with pyramid vision transformer, Computational visual media 8 (3) (2022) 415–424.

[34] J. Chen, S. hong Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, S.-H. G. Chan, Run, don't walk: Chasing higher flops for faster neural networks, Proceedings of the IEEE conference on computer vision and pattern recognition (2023) 12021–12031.

[35] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.

Author biography

Yuwei Mo received the B.S. degree in telecommunications engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2021, where he is currently pursuing the M.S. degree in signal and information processing. His research interests include image classification, object detection, and semantic segmentation.

Pengfei Zuo is currently pursuing the B.S. degree in computer science and technology from the Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include image classification, object detection, and semantic segmentation.

Quan Zhou received the B.S. degree in electronics and information engineering from the China University of Geosciences, Hubei, China, in 2002, and the M.S. and Ph.D. degrees in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2013, respectively. He is now a full Professor with the National Engineering Research Center of Communication and Network Technology, Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include deep learning, pattern recognition, and computer vision.

Zhiyi Mo is now a professor with the Guangxi Colleges and Universities Key Laboratory of Intelligent Software, Wuzhou University, Wuzhou, China. His research interests include object detection, and semantic segmentation.

Yawen Fan received BE and MS degrees in electronic engineering from Hehai University, Nanjing, China in 2003 and 2006, respectively. She has received the doctor degree in EE department from Shanghai Jiao Tong University, Shanghai, China. She is now an assistant professor at Nanjing University of Posts and Telecommunications, Nanjing, China. Her research interests include computer vision.

Suofei Zhang received the PhD degree in School of Information Science and Engineering from Southeast University in 2013 and the master degree in School of Mechanical Engineering from Jiangsu University in 2007. In 2013, he joined the School of Internet of Things at the Nanjing University of Posts and Telecommunications. His research interests include computer vision, video surveillance, real-time object tracking and deep learning based image processing.

Bin Kang received the Ph.D. degree from Nanjing University of Posts and Telecommunications, Nanjing, in 2016. He is currently an Associate Professor at College of Internet of Things, Nanjing University of Posts and Telecommunications. His research interests include computer vision and pattern recognition.