CvT: Introducing Convolutions to Vision Transformers

Abstract

Convolutional Neural Networks (CNNs) and Transformers are widely employed in computer vision, but they face challenges in capturing high-resolution and multi-scale features, particularly struggling in dense prediction tasks like segmentation and detection. To overcome this limitation, this paper introduces a novel architecture, the Convolutional Vision Transformer (CvT), aiming to preserve the global context and dynamic attention mechanism of Transformers while leveraging beneficial characteristics of Convolutional Neural Networks, such as translation, scale, and distortion invariance. Building on CvT, improvements are made by integrating lightweight multihead self-attention modules (LMHSA) and inverse residual feed-forward networks (IRFFN) to enhance the capture of global feature information. Extensive experiments on datasets such as CIFAR-10 demonstrate that the proposed architecture not only has a lower parameter count and computational complexity but also achieves higher accuracy, validating its advantages.

Keywords: Transformer, CNN, LHASA, IRFFN.

1 Introduction

Transformer model has recently achieved significant success in natural language processing tasks. However, in the field of computer vision, convolutional neural networks (CNNs) continue to dominate. To fully leverage the advantages of both Transformer and CNN, this paper proposes a novel architecture named Convolutional vision Transformer (CvT) [7]. CvT combines the design principles of Vision Transformer (ViT) and convolution by introducing convolutional operations and projections. This approach aims to preserve the global context and dynamic attention mechanisms of Transformers while acquiring beneficial properties of convolutional networks, such as translation, scaling, and distortion invariance. The extensive experiments in this paper demonstrate the state-of-the-art performance of CvT on the ImageNet-1k and ImageNet-22k datasets, with lower parameter count and computational complexity. Additionally, CvT exhibits adaptability to input images of different resolutions without the need for positional encoding. Improvements to CvT include a lightweight multihead self-attention module (LMHSA) and an inverted residual feedforward network (IRFFN), enhancing the capture of global feature information. The experimental results confirm that the proposed enhancements lead to higher accuracy.

2 Related works

2.1 Introducing Convolutions to Transformers

The Vision Transformer (ViT) [3]has achieved performance comparable to traditional convolutional neural networks such as ResNets and EfficientNet in image classification tasks, demonstrating that a pure Transformer architecture can achieve state-of-the-art results with sufficient data. ViT divides each image into a series of tokens (non-overlapping image patches) and applies multiple standard transformer layers, including Multi-Head Self-Attention (MHSA) and Feed-Forward Networks (FFN), to model these tokens. DeiT [5] further explores methods for training on small datasets and distillation specifically for ViT. CPVT [2] replaces predefined position embeddings in ViT with Conditional Position Embedding (CPE), enabling the Transformer to handle input images of arbitrary sizes without interpolation. TNT [4] employs external Transformer blocks to process image patch embeddings, while internal Transformer blocks simulate relationships between embeddings at the block and pixel levels to model representations at both levels. T2T [8] improves tokenization approach by connecting multiple tokens within a sliding window into a single token. PVT [6] introduces a multi-stage design (without convolution) in the Transformer, similar to the multiscale design in CNNs, suitable for dense prediction tasks. In comparison to these approaches, CvT integrates the advantages of both convolutional neural networks and Transformers, incorporating Convolutional Token Embedding and Convolutional Projection operations to achieve the best of both worlds.

3 Method

3.1 Overview

CvT introduces two convolution-based operations in the Vision Transformer: Convolutional Token Embedding and Convolutional Projection. Figure 1 (a) illustrates the CvT architecture employing a multi-stage hierarchical design, inspired by CNN principles, with a total of three stages. Each stage consists of two parts. Initially, the input image (or the 2D reshaped token map) undergoes processing through the Convolutional Token Embedding layer. This layer utilizes convolutional operations to reshape tokens into a 2D spatial grid, employing overlapping input. Additional layer normalization is applied to the tokens, allowing each stage to gradually reduce the number of tokens (feature resolution) while increasing their width (feature dimension). This achieves spatial downsampling and enriches representations, similar to CNN designs. Unlike other Transformer-based architectures, CvT does not sum temporary position embeddings into the tokens.

Subsequently, each stage comprises a set of Convolutional Transformer Blocks. Figure 1 (b) illustrates the structure of the Convolutional Transformer Block, incorporating depth-wise separable convolution operations (Convolutional Projection) applied independently to the embeddings of queries, keys and values. This is in contrast to ViT's standard position-wise linear projection. Only the last stage adds classification tokens, and the final output undergoes prediction through an MLP (fully connected layer) Head. The Convolutional Token Embedding layer is discussed in detail, followed by a demonstration of how Convolutional Projection is applied in the Multi-Head Self-Attention module, elucidating its efficient design for managing computational costs.

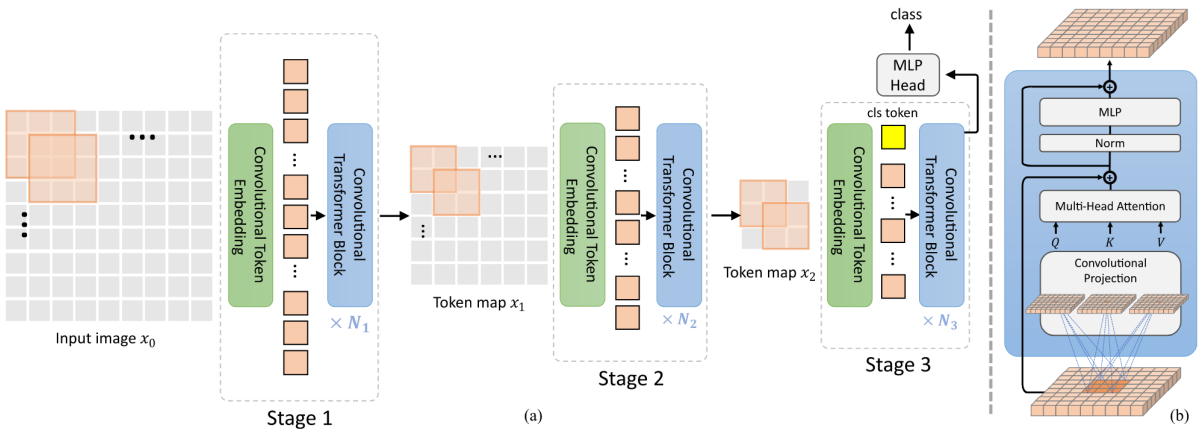

Figure 1. Overview of the CvT

3.2 Convolutional Token Embedding

The convolutional operations in CvT are designed to emulate local spatial context, progressing from low-level edges to high-level semantic primitives using a multi-stage hierarchical approach, similar to CNNs. Formally, given the 2D image or 2D reshaped output token map $x_{i-1} \in \mathbb{R}^{H_{i-1} \times W_{i-1} \times C_{i-1}}$ from the previous stage as input for stage i, CvT learns a mapping function $f(\cdot)$ that maps x_{i-1} to new tokens $f(x_{i-1})$ with channel size C_i, where $f(\cdot)$ is a 2D convolution operation with a kernel size $s \times s$, stride $s - o$, and padding p (handling boundary conditions). The new token map $f(x_{i-1}) \in \mathbb{R}^{H_i \times W_i \times C_i}$ has height and width

$$H_i = \left\lfloor \frac{H_{i-1} + 2p - s}{s - o} + 1 \right\rfloor, W_i = \left\lfloor \frac{W_{i-1} + 2p - s}{s - o} + 1 \right\rfloor \tag{1}$$

$f(x_{i-1})$ is flattened into a vector of size $H_i W_i \times C_i$, normalized through layer normalization, and input into subsequent Transformer blocks in stage i. The Convolutional Token Embedding layer allows CvT to adjust the feature dimension and token count for each stage by varying the parameters of the convolutional operation. Consequently, CvT gradually reduces the length of the token sequence while increasing the feature dimension of the tokens in each stage,. This enables tokens to represent increasingly complex visual patterns, covering a larger spatial range, similar to feature layers in CNNs.

3.3 Convolutional Projection for Attention

The goal of the Convolutional Projection layer is to provide additional modeling of local spatial context and offer efficiency advantages by allowing undersampling of the K and V matrices. Essentially, the proposed Transformer block with Convolutional Projection is a generalization of the original Transformer block. We suggest replacing the original position-wise linear projection in Multi-Head Self-Attention (MHSA) with depth-wise separable convolution, forming the Convolutional Projection layer.

Fig.2 illustrates the original position-wise linear projection used in ViT, while Fig.2 demonstrates our proposed s Œ s Convolutional Projection. As shown in Fig.2, tokens are first reshaped into a 2D token map. Subsequently, a depth-wise separable convolution layer with a size of s is applied to achieve Convolutional Projection. Finally, the projected tokens are flattened to 1D for further processing. This can be represented as:

$$x^i_{q/k/v} = \text{Flatten}(\text{Conv2d}(\text{Reshape2D}(x_i), s)) \tag{2}$$

where $x^i_{q/k/v}$ is the token input for the Q/K/V matrices of layer i, x_i is the unprocessed token before Convolutional Projection, Conv2d is an operation implemented by depth-wise separable convolution [1]: **Depth-wise Conv2d BatchNorm2d Point-wise Conv2d**, and s represents the convolution kernel size.

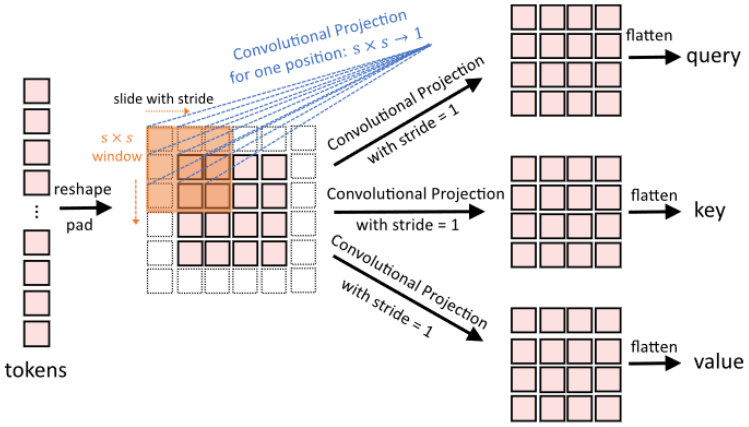

Figure 2. Convolutional projection of CvT

4 Implementation details

4.1 Comparing with the released source codes

CvT does not directly address the challenge of multi-scale feature extraction. Instead, the primary focus of CvT is on incorporating convolution into the Vision Transformer to enhance the representation of spatial information. This is achieved through the introduction of new convolutional token embeddings and convolutional Transformer blocks to improve both performance and efficiency. Transformers, due to their fixed dimensions, struggle to capture low-resolution and multi-scale features, which is particularly disadvantageous for dense prediction tasks such as segmentation and detection.

To address this limitation, we implement the Lightweight Multi-head Self-Attention (LMHSA) and Inverted Residual Feed-Forward Network (IRFFN). These components assist in capturing local and global structural information within intermediate features, thereby enhancing the representational capacity of network. LMHSA and IRFFN are instrumental in capturing both local and global contextual information, contributing to the overall improvement of the network's ability to represent complex features in diverse scales.

4.2 Experimental environment setup

To conduct our evaluation, we utilize CIFAR-10, Oxford-IIIT-Pet, and Oxford-IIIT-Flower datasets.

4.3 Lightweight Multi-head Self-attention

In the original self-attention module, the input $X \in \mathbb{R}^{n \times d}$ is linearly transformed into queries $Q \in \mathbb{R}^{n \times d_k}$, keys $K \in \mathbb{R}^{n \times d_k}$, and values $V \in \mathbb{R}^{n \times d_v}$, where $n = H \times W$ is the number of patches. For simplicity, the reshaping operation from $H \times W \times d$ to $n \times d$ tensors is omitted in Fig. 3. The symbols d, d_k, and d_v represent

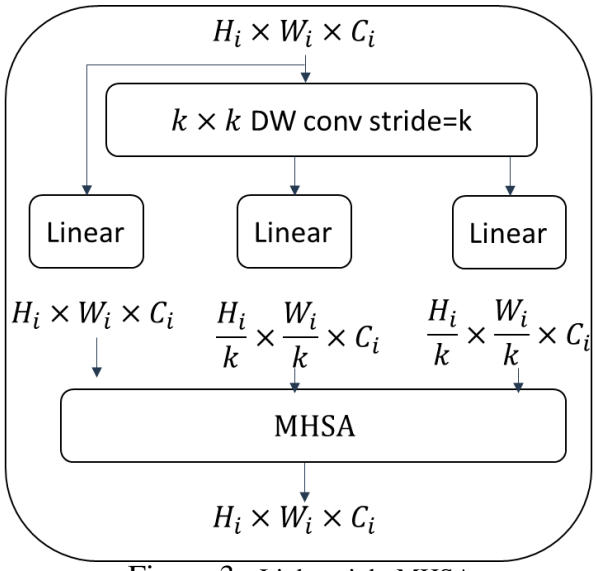

Figure 3. Lightweight MHSA

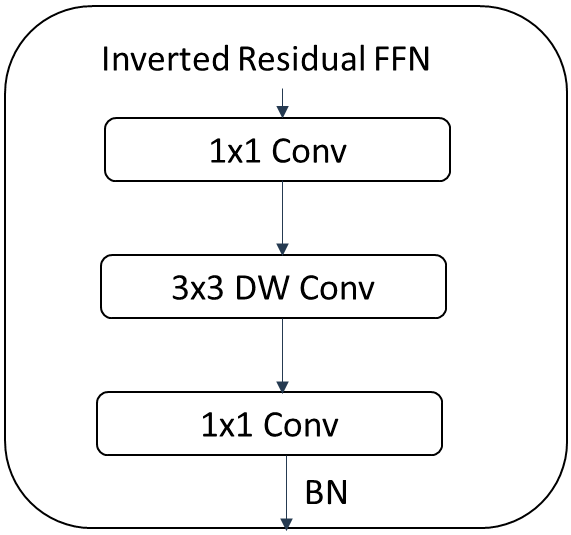

Figure 4. Inverted Residual FFN

the dimensions of the input, keys (queries), and values, respectively. The self-attention module is then applied as follows:

$$\text{Attn}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

To alleviate computational costs, we use a depth-wise convolution with a stride of k ($k \times k$ depth-wise convolution) to reduce the spatial size of K and V before the attention operation, i.e., $K' = \text{DWConv}(K) \in \mathbb{R}^{nk^2 \times d_k}$ and $V' = \text{DWConv}(V) \in \mathbb{R}^{nk^2 \times d_v}$ as shown in Fig 3. Furthermore, we add relative positional biases B to each self-attention module, and the corresponding lightweight attention is defined as follows:

$$\text{LightAttn}(Q, K, V) = \text{Softmax}\left(\frac{QK'^T}{\sqrt{d_k}} + B\right)V' \tag{4}$$

Here, $B \in \mathbb{R}^{n \times nk^2}$ is randomly initialized and learnable. The learned relative positional biases can also be easily transferred to different sizes $m_1 \times m_2$ with $B' \in \mathbb{R}^{m_1 \times m_2}$ using bilinear interpolation, i.e., $B' = \text{Bicubic}(B)$. Thus, the proposed CMT can be conveniently fine-tuned for other downstream vision tasks. Finally, the Lightweight Multi-head Self-Attention (LMHSA) module is defined by considering h "heads," applying the LightweightAttention function to the input. Each head produces a sequence of size $n \times d/h$, and these h sequences are then concatenated into an $n \times d$ sequence.

4.4 Inverted Residual Feed-forward Network

The original Feed-Forward Network (FFN) proposed in ViT [3] consists of two linear layers separated by GELU activation. The first layer expands the size by a factor of 4, and the second layer reduces it by the same factor:

$$\text{FFN}(X) = \text{GELU}(XW_1 + b_1)W_2 + b_2 \tag{5}$$

Here, $W_1 \in \mathbb{R}^{d \times 4d}$ and $W_2 \in \mathbb{R}^{4d \times d}$ represent the weights of the two linear layers, and b_1 and b_2 are bias terms. Fig.4 provides a visual representation of our design. The proposed Inverted Residual Feed-Forward Network (IRFFN) appears similar to the inverted residual block and comprises an expansion layer, depth-wise convolution, and projection layer. Specifically, we modify the location of the shortcut connection for improved

Table 1. Top-1 accuracy

Model	CIFAR-10	Oxford-IIIT-Pet	Oxford-IIIT-Flower
CvT-13	96.83	90.25	95.30
CvT-21	97.16	91.03	96.12
Proposed	97.50	90.70	95.50

performance.

$$\text{IRFFN}(X) = \text{Conv}\left(\text{F}(\text{Conv}(X))\right) \tag{6}$$

$$\text{F}(X) = \text{DWConv}(X) + X \tag{7}$$

Here, the activation layers are omitted. We also include batch normalization after the activation layer and the final linear layer. Depth-wise convolution is employed to extract local information, with negligible additional computational cost. The motivation for inserting the shortcut is similar to classical residual networks, promoting the ability to propagate the gradient across layers. We demonstrate that this method contributes to better results in our experiments.

5 Results and analysis

Table 1 presents a comparison between the proposed method and CvT on different datasets. The top two rows show the original results of CvT, where the model is trained on datasets like CIFAR-10 and tested afterward. Due to resource limitations on the server, we were only able to train and test on datasets such as CIFAR, which results in a loss of dataset diversity. However, as observed in Table 1, Top-1 accuracy has improved compared to CvT-13, although with an increase in runtime. In theory, depth-wise separable convolutions are faster, but due to constraints in lower-level operations, they may require longer runtimes. The observed improvement in Top-1 accuracy suggests a trade-off between speed and accuracy in the proposed method.

6 Conclusion and future work

In this study, we reproduce and enhance the Convolutional Vision Transformer (CvT), with a particular focus on modifying the Multi-Head Self-Attention (MHSA) and Feed-Forward Network (FFN) modules. The proposed approach exhibits increased accuracy in training on datasets such as CIFAR-10. However, this improvement comes at the cost of an extended computational time. As part of our future work, we aim to address and mitigate the computational time concerns, exploring optimizations and efficiency enhancements to ensure a more balanced trade-off between accuracy and computational efficiency in subsequent iterations of the proposed method.

References

[1] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[2] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[4] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021.

[5] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[6] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021.

[7] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22–31, 2021.

[8] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 558–567, 2021.