technologies

MDPI

Review

Vision Transformers for Image Classification: A Comparative Survey

Yaoli Wang [1], Yaojun Deng [2], Yuanjin Zheng [2], Pratik Chattopadhyay [3] and Lipo Wang [2,4,*]

[1] College of Electronic Information Engineering, Taiyuan University of Technology, Taiyuan 030600, China
[2] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore
[3] Department of CSE, Indian Institute of Technology (BHU), Varanasi 221005, India
[4] Institute for Digital Molecular Analytics and Science, Nanyang Technological University, Singapore 639798, Singapore
* Correspondence: elpwang@ntu.edu.sg

Abstract: Transformers were initially introduced for natural language processing, leveraging the self-attention mechanism. They require minimal inductive biases in their design and can function effectively as set-based architectures. Additionally, transformers excel at capturing long-range dependencies and enabling parallel processing, which allows them to outperform traditional models, such as long short-term memory (LSTM) networks, on sequence-based tasks. In recent years, transformers have been widely adopted in computer vision, driving remarkable advancements in the field. Previous surveys have provided overviews of transformer applications across various computer vision tasks, such as object detection, activity recognition, and image enhancement. In this survey, we focus specifically on image classification. We begin with an introduction to the fundamental concepts of transformers and highlight the first successful Vision Transformer (ViT). Building on the ViT, we review subsequent improvements and optimizations introduced for image classification tasks. We then compare the strengths and limitations of these transformer-based models against classic convolutional neural networks (CNNs) through experiments. Finally, we explore key challenges and potential future directions for image classification transformers.

Keywords: computer vision; pattern recognition; artificial intelligence; machine learning

check for updates

Academic Editors: Juan Gabriel Avina-Cervantes and Pedro Antonio Gutiérrez

Received: 19 January 2024
Revised: 23 December 2024
Accepted: 8 January 2025
Published: 12 January 2025

Citation: Wang, Y.; Deng, Y.; Zheng, Y.; Chattopadhyay, P.; Wang, L. Vision Transformers for Image Classification: A Comparative Survey. *Technologies* **2025**, *13*, 32. https://doi.org/10.3390/technologies13010032

1. Introduction

Transformer [1] was originally introduced for Natural Language Processing (NLP) tasks, leveraging the *self-attention* mechanism to capture relationships within sequences. Models like BERT [2] and RoBERTa [3] demonstrated their effectiveness, inspiring their adoption in computer vision (CV) tasks, where they achieved notable success.

Traditional convolutional neural networks (CNNs) served as the foundation for CV tasks, with their layered convolutional and pooling operations effectively extracting and modeling image features across multiple levels [4–6]. While initially integrated with CNN architectures [7], ViTs have evolved into standalone models, replacing convolutional layers with self-attention mechanisms.

Previous surveys [8,9] discussed the use of transformers in image classification [10,11], segmentation [12], and video processing [13]. The introduction of the ViT [10] demonstrated the viability of applying transformer architectures directly to CV tasks. ViT's success spurred extensive research on improving transformer architectures for CV tasks. For example, Azad et al. [14] and Liu et al. [15] explored transformers' applications in medical image analysis, covering tasks like classification, segmentation, and detection. Khalil et al. [16] conducted a comprehensive survey on ViTs for image classification, presenting a chronological overview of model advancements and covering a wide range of architectures from early ViT designs to modern lightweight transformer models. Their work highlights dataset dependencies, key milestones in transformer development, and broad trends across transformer-based models. In contrast, our manuscript focuses specifically on the technical advancements and architectural optimizations of ViTs for image classification. We delve into model-level comparisons, such as ViT-L vs. ViT-B, and emphasize improvements in training efficiency, lightweight models (e.g., DeiT, T2T-ViT), and benchmark experimental results. Additionally, we provide an in-depth discussion on challenges and future research directions, offering targeted insights for ongoing developments in transformer-based image classification.

This survey focuses specifically on image classification with transformers. Unlike previous reviews such as the comprehensive work by Maurício et al. [17], which offers a high-level comparison between ViTs and CNNs (CNNs), our study delves deeper into model-specific improvements and experimental insights. We begin with an introduction to the foundational concepts and the pioneering ViT. Building on the ViT, we provide an in-depth review of advancements such as DeiT [11] and T2T-ViT [18] models, analyze the comparative strengths of ViTs against traditional CNNs [6,19,20], and discuss challenges and future research directions, focusing on technical innovations and practical applications.

This paper is organized as follows: Section 2 delves into the fundamental concepts of the ViT, with a particular focus on the key mechanisms such as self-attention and multi-head attention that underpin the ViT architecture. We will also introduce the pioneering ViT model in detail, laying the groundwork for understanding its subsequent evolution. Section 3 is dedicated to the improved models of transformers. It begins with a detailed discussion of the baseline models, ViT-L and ViT-B, highlighting their characteristics and significance. We then provide a comprehensive classification and overview of the various improvement strategies, followed by in-depth examinations of each category of improved models. This includes models that enhance data efficiency, combine CNN concepts, adopt lightweight designs, deepen the transformer structure, explore aggregating nested transformers and cross-attention mechanisms, introduce new loss functions, and other miscellaneous improvements. In Section 4, we conduct a detailed comparison of the introduced ViT models with some representative CNN models. This comparison is presented through experimental results on different datasets, providing insights into the relative strengths and weaknesses of these models in terms of parameters, FLOPs (floating point operations), and classification accuracies. Finally, in Section 5, we discuss the challenges that currently face ViTs in image classification, such as the lack of inductive biases and computational inefficiencies. We also explore potential future directions, including the application of convolutions in novel ways, the exploration of alternative mechanisms, and the utilization of image–text training, which could potentially address some of the existing limitations and open up new avenues for research in this field.

2. ViT Concepts

2.1. Self-Attention

The first key concept of transformer architectures is the self-attention mechanism [1]. It serves as a foundational layer in transformer network structures, designed to capture relationships between sequence elements (e.g., words in a sentence in NLP tasks). This mechanism can effectively attend to entire sequences and learn long-range dependencies.

Self-attention operates as a sequence-to-sequence (Seq2Seq) structure, akin to recurrent neural networks (RNNs) [21,22]. However, unlike RNNs, it eliminates dependency on previous states, allowing transformers to process sequence elements in parallel and significantly enhance computational efficiency. Compared to convolutional neural networks (CNNs), self-attention excels in learning global, adaptive-length relationships, whereas CNNs are constrained to manually defined kernel regions. Structurally, self-attention can be viewed as a generalized form of convolution layers applied in CNNs [23], which accounts for its ability to effectively extract image information. Notably, the property of capturing image features with dynamic range has also been integrated into CNNs [24]. An example of a visual self-attention [25] mechanism structure block is shown in Figure 1.

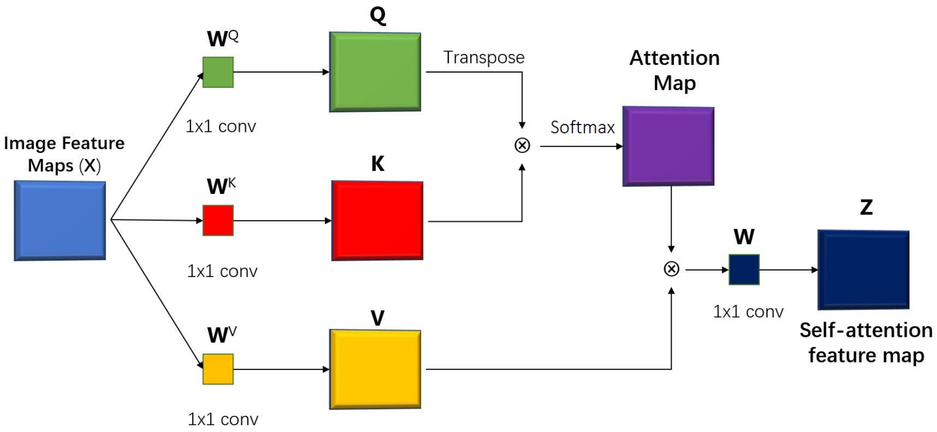


Figure 1. The block figure of the self-attention mechanism. $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ are query, key and value matrices which are used in the computation of self-attention features. $\mathbf{W}^q$, $\mathbf{W}^k$ and $\mathbf{W}^v$ are corresponding linear projections from input images to the 3 matrices. $\boldsymbol{Z}$ is the self-attention feature map.

In the self-attention layer, an input image is linearly transformed into three matrices: a query matrix $\boldsymbol{Q}$, a key matrix $\boldsymbol{K}$, and a value matrix $\boldsymbol{V}$ as illustrated in Figure 1. In the classic self-attention mechanism, the three matrices are generated by multiplying each vector in the input sequence with three individual learnable transformation matrices (or weights), $\mathbf{W}^q$, $\mathbf{W}^k$, and $\mathbf{W}^v$. The visual self-attention uses a sequence of the convolutional features of an image to calculate the three matrices. The matrix $\boldsymbol{Q}$ contains "query" information to match the other elements in the input sequence. The matrix $\boldsymbol{K}$ contains "key" information to be matched by the other elements in the input sequence. The matrix $\boldsymbol{V}$ extracts the actual information in the original input. A typical single-head visual self-attention feature map $\boldsymbol{Z}$ can be calculated based on input $\boldsymbol{X}$ as follows:

$$\boldsymbol{Z} = \textbf{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_q}})\boldsymbol{V}, \tag{1}$$

where d_q is the embedding dimensionof the input entity. The complete self-attention process extracts and computes matrix $\boldsymbol{Q}$ and $\boldsymbol{K}$ with normalization and translates the result into probability form. $\boldsymbol{V}$ is multiplied by the sum of vectors, which means it has larger attention in the next network layer.

This attention mechanism extracts information from input content irrespective of the positions of individual elements. As a result, self-attention outputs remain consistent across sequences with identical content but varying positional arrangements.

To address this limitation, an additional step, *position encoding*, is introduced to capture the positional information of sequence elements. The original NLP transformer model [1] proposes using permanent positional encoding, defined by the following equations:

$$PE(pos, 2i) = sin(\frac{pos}{10000^{\frac{2i}{d_{model}}}}) \tag{2}$$

$$PE(pos, 2i+1) = cos(\frac{pos}{10000^{\frac{2i}{d_{model}}}}) \tag{3}$$

where d_{model} denotes the dimension of embedding, *pos* denotes the position of an element, i denotes the current dimension of positional embedding. Several adaptive positional encoding mechanisms [7,26–28] were introduced and proved to make improvements compared to original permanent positional encoding.

2.2. Multi-Head Attention

In transformer model design, a simple single-head self-attention layer shows limitations in computing multiple complex relationships between elements in a sequence. Multi-head attention divides an entire task into multiple self-attention blocks, with individual groups of W^{Q_i}, W^{K_i}, and W^{V_i} for each head. Each group of the three weights focuses on different scopes of the input sequence, extracts attention features, and concludes a final attention result. For an input X, a multi-head attention layer computes h outputs and concatenates them into a single matrix $[Z_0, Z_1, \ldots, Z_{h-1}]$.

In early research, researchers made efforts to combine self-attention with traditional CNN architectures [29–31] as an additional processing step and achieved exciting progress.

2.3. ViT

ViT [10] is a pure transformer designed specifically for image classification tasks. The ViT model closely follows the architecture of the original NLP transformer, as illustrated in Figure 2.

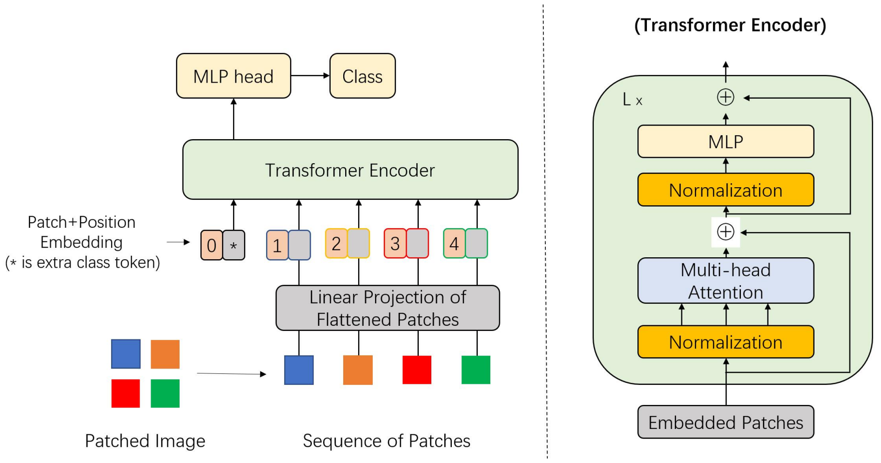


Figure 2. A block figure of the Vision Transformer (ViT). Input images are divided into patches (shown in various colors) and embedded with position information. The embedded patch sequence is applied to the transformer encoder with normalization, self-attention processing, and MLP. The MLP head output is used for classification.

Intuitively, a 2D input image can be represented as $x \in \mathbb{R}^{H \times W \times C}$, where $H \times W$ denotes the resolution of the original image, and C represents the number of image channels. In the initial step of the ViT model, the input image is reshaped into a sequence of N 2D patches with dimensions $P \times P$, expressed as $x_p \in \mathbb{R}^{N \times (P^2 \times C)}$ where $N = HW/P^2$. This reshaping step allows the ViT model to treat an image as a sequence input and leverage self-attention to extract relationships between patches. The model dimension D remains constant across all ViT layers, enabling the application of a trainable linear projection to map each vectorized patch, referred to as patch embedding.

Inspired by the class token concept from BERT [2], ViT introduces an additional learnable classification token into the sequence of embedded patches. Position tokens are then concatenated with these embedded image patches and the classification token, forming the complete input sequence for the ViT model. Dosovitskiy et al. [10] investigated several variants of 2D positional embedding methods but found no significant performance improvement over 1D positional embeddings. Unlike traditional NLP transformers, which have an encoder–decoder structure [1] designed for tasks like machine translation, the ViT architecture includes only encoding blocks. The output from the Multilayer Perceptron (MLP) head serves directly as the final classification result, simplifying the design while remaining effective for image classification tasks.

In training, ViT adopts a pre-training and fine-tuning strategy [32–34]. The model is pre-trained on large datasets [3,35] or combinations of multiple datasets [36,37] using self-supervised learning techniques [38,39]. Fine-tuning on smaller datasets tailors the model to specific tasks, reducing computational costs and enhancing accuracy. The experimental results [10] demonstrate the impact of dataset size during pre-training. When trained on mid-sized datasets such as ImageNet [40], ViT achieves accuracy comparable to ResNet [19,20], a widely recognized CNN model. However, when pre-trained on larger datasets like JFT-300M [41], ViT surpasses ResNet in downstream fine-tuning tasks, including ImageNet, CIFAR-100, and VTAB. These results emphasize that ViT is highly data-dependent and outperforms traditional CNNs only when provided with sufficient training data.

The ViT model represents the first successful application of a pure transformer architecture to replace convolutional layers in CNNs entirely. This milestone has inspired significant advancements in the field of Vision Transformers, establishing ViT as a cornerstone for ongoing research in image classification transformers.

3. Improved Models on Transformer

This section is dedicated to exploring the improved models of transformers in the context of image classification. We begin by introducing the baseline models, namely ViT-B and ViT-L, which have been the cornerstones in the evolution of ViTs. These models have demonstrated the potential of applying transformer architectures to image classification tasks, yet they also possess certain limitations that have spurred further research.

Subsequently, we will present a comprehensive roadmap that outlines how various improved models have been developed to overcome these limitations. This will involve categorizing the improvement strategies into key areas such as enhancing data efficiency, integrating CNN concepts, designing lightweight architectures, deepening the transformer structure, and exploring novel attention mechanisms. Each of these directions represents a significant effort in the pursuit of more powerful and efficient image classification models.

Finally, we will delve into the details of these improved models, examining their unique contributions, architectures, and experimental results. By following this structure, readers will gain a clear understanding of the progression from the foundational ViT models to the state-of-the-art improvements, and how each step has contributed to the advancement of ViTs in image classification.

3.1. The Baseline ViT-B and ViT-L Models

The ViT-B (ViT-Base) and ViT-L (ViT-Large) models [10] in the ViT family are crucial benchmark models. The ViT-B model has a relatively moderate parameter scale and strikes a good balance between model complexity and performance. When handling image classification tasks, it can effectively capture the feature information in images by dividing the input image into multiple patches and using the self-attention mechanism of the transformer to model the relationships between these patches. After pre-training on large-scale datasets, the ViT-B model has demonstrated good classification accuracy on datasets such as ImageNet, proving the feasibility of the transformer architecture in image classification tasks.

The ViT-L model, on the other hand, has a larger parameter scale and can learn more complex image feature representations. It has an advantage in processing high-resolution images or tasks that require high feature extraction capabilities. For example, in some scenarios that require fine-grained classification, the ViT-L model can conduct more in-depth exploration of the details and semantic information in images through its deeper structure and more parameters. However, the larger parameter scale also brings higher computational costs and a higher demand for data volume. In practical applications, the appropriate model needs to be selected based on the specific requirements of the task and the limitations of computing resources. These two benchmark models provide important reference standards for the design and performance evaluation of subsequent improved models, promoting the continuous development of transformers in the field of image classification.

3.2. Classification and Overview of Improved Transformer Architectures

While the ViT has demonstrated the potential of applying transformer architectures to image classification tasks, it also faces significant challenges. These include the need for large amounts of training data, high computational costs, and limitations in capturing positional information. Since its introduction, ViT has driven considerable research efforts to address these challenges.

In the context of image classification, numerous improved transformer-based architectures have been proposed to overcome the limitations of traditional CNNs and enhance classification performance. These improvements primarily focus on key aspects aimed at mitigating the constraints of transformers and optimizing their effectiveness.

This subsection categorizes and explains the main directions and objectives behind these architectural improvements. It serves as a roadmap to help readers better understand how different approaches address CNN limitations and contribute to advancing classification performance.

3.2.1. Improving Data Efficiency

Although the ViT has demonstrated the potential of the transformer architecture in visual tasks, its dependence on large-scale datasets has limited its application range. Therefore, some research efforts have been dedicated to reducing data requirements and improving data utilization efficiency. For example, the Data-efficient image Transformer (DeiT) achieved good training results on the medium-sized dataset ImageNet by introducing a distillation strategy, significantly reducing the dependence on large-scale data. The Tokens-to-Token ViT (T2T-ViT) improved data utilization efficiency and reduced computational complexity by aggregating adjacent tokens and improving the model structure.

3.2.2. Combining CNN Concepts

CNNs have advantages in extracting local features of images, while transformers excel in capturing long-range dependencies. Combining the strengths of the two has become an important direction for improving transformer performance. The Conditional Position-encoding Vision Transformer (CPVT) introduced a new position-encoding layer and utilized the characteristics of CNNs to compute adaptive position embeddings, enhancing model performance. The Token-based Visual Transformer (VT) sampled input images into semantic visual tokens using CNNs and then extracted relationships using transformers, improving feature extraction capabilities. In addition to this, models such as the Convolutional Vision Transformer (CvT) and CeiT integrated the structures of CNNs and transformers to varying degrees, effectively reducing computational complexity and improving classification accuracy.

3.2.3. Lightweight Design

Transformer models have high computational costs and a large number of parameters, limiting their application in specific tasks. To address this issue, researchers have explored lightweight design methods. LeViT introduced a pyramid structure and convolutional layers, significantly improving inference speed while maintaining high accuracy. The Compact Convolutional Transformer (CCT) replaced the traditional patching process with convolutions, preserving local spatial information and reducing data requirements. Other methods, such as dynamic token sparsification, introducing a progressive shift mechanism, and reducing computational resource requirements, have also made certain progress in improving model efficiency.

3.2.4. Deepening the Transformer Structure

Inspired by the deep structures of CNNs, researchers have attempted to deepen the transformer structure to enrich the ability to extract image features. The Class-Attention in Image Transformers (CaiT) improved training efficiency by improving the normalization method and introducing a class-attention layer. DeepViT addressed the attention collapse problem that occurs when the depth increases by improving the attention mechanism. These improvements indicate that a reasonably designed deeper transformer structure can help improve image classification performance.

3.2.5. Aggregating Nested Transformers and Cross-Attention

To improve the training efficiency of transformers and reduce the dependence on large-scale data, some studies have explored aggregating nested transformers and cross-attention mechanisms. The Aggregating Nested Transformers (NesT) stacked transformer layers and aggregated cross-block self-attention, improving accuracy without significantly increasing parameters and computational complexity. The Cross-Attention Multi-Scale Vision Transformer (CrossViT) fused features at different scales through a cross-attention mechanism, balancing computational complexity and accuracy.

3.2.6. Other Improvement Directions

In addition to the above aspects, some research has improved transformers from other perspectives. For example, introducing new loss mechanisms (such as Patch-wise Loss) to address the over-smoothing problem and improve model performance; exploring the performance of abstracted transformer architectures (such as MetaFormer) and their different variants; designing specialized transformer architectures for specific tasks (such as multi-object tracking, image enhancement, image retrieval, etc.); and improving activation functions and optimizing details in the training process (such as attention selection strategies). These diverse improvement methods provide broader development space for the application of transformers in image classification and other visual tasks.

3.3. Improvement on Data Efficiency

Initially, ViT researchers focused on addressing the significant dependency on large datasets. While ViT demonstrated the potential of full transformer models in vision tasks, applying this architecture remains challenging due to the limited availability of sufficiently large datasets for many specific tasks [41].

3.3.1. Data-Efficient Image Transformer

The Data-efficient Image Transformer (DeiT) [11] significantly reduces the dataset size requirement for training ViT models. Pre-training is performed using only the mid-sized ImageNet dataset, yet experimental results show that DeiT achieves better accuracy compared to the classic ViT while significantly reducing computational complexity and parameter size. An additional distillation token further enhances the DeiT model, enabling it to outperform EfficientNet [6,42] on the same training dataset. Notably, DeiT maintains a pure transformer architecture, without incorporating any convolutional layers. A standout feature of DeiT is the integration of the training strategy known as *distillation* [43]. This approach assumes the presence of an efficient classifier, referred to as the *teacher model*, which guides the loss computation process. RegNet-16GF [44] is recommended as a teacher model.

The modified DeiT structure is illustrated in Figure 3.

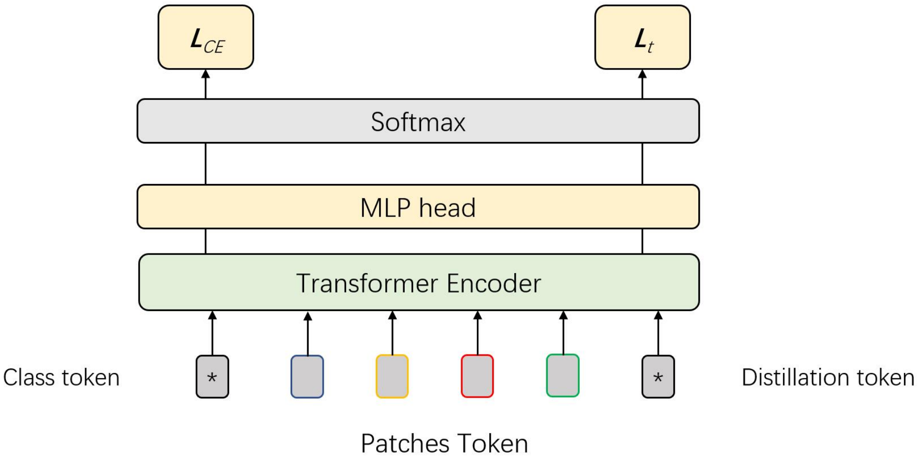


Figure 3. The block figure of DeiT. A distillation token is introduced together with the class token, and processed individually. They are applied to the computation of teacher-model loss and cross-entropy loss, respectively.

In this setup, L_{CE} represents the cross-entropy loss, and L_t denotes the loss between the distillation tokens and the output labels from the teacher model. Touvron et al. [11] conducted experiments comparing soft distillation and hard distillation [43], with results showing that hard distillation offers significant advantages over its soft counterpart. The mechanism of hard distillation is as follows:

$$\begin{gathered} \boldsymbol{L}^{hard} = \frac{1}{2}\boldsymbol{L}_{CE}(\varphi(Z_s), y) + \frac{1}{2}\boldsymbol{L}_{CE}(\varphi(Z_s), y_t) \\ y_t = argmax_c Z_t(c) \end{gathered} \tag{4}$$

where φ stands for the softmax function, and $\mathbf{Z}_s$ represents the output from the student network (referred to as the transformer).

To simplify the hard distillation process in DeiT, the CE loss is computed using the outputs of the student model with both the actual labels and the labels from the teacher model. The global loss is obtained by summing these two CE losses. The class token aims to minimize the loss with respect to the actual labels, while the distillation token learns from the teacher model, effectively supplementing the information provided by the class token. Touvron et al. observed that these two tokens converge in different directions [11]. The average similarity between the two tokens starts at 0.06 in the early layers and increases to 0.93 in the final layer. This indicates that the two tokens capture both the similarities and differences between the actual labels and the teacher model's predicted labels.

The DeiT model demonstrates that transformer-based networks can achieve impressive accuracy even with a mid-sized training dataset. This achievement has inspired further research into modifying transformer architectures for broader applications across various tasks.

3.3.2. Tokens-to-Token ViT

The classic ViT model has two notable limitations that contribute to its data-hungry nature and high computational complexity: (1) Input images are split into tokens in a way that fails to effectively model local structures, such as edges and corners. (2) The self-attention backbone introduces redundancy, limiting feature diversity and training efficiency.

To address these limitations, Yuan et al. proposed the Tokens-to-Token (T2T) module [45]. This module aggregates neighboring tokens to capture local structural features, reconstructs them into image-like representations, and then performs a soft token-splitting process for subsequent layers. The enhanced model is referred to as Tokens-To-Token Vision Transformer (T2T-ViT), as shown in Figure 4.

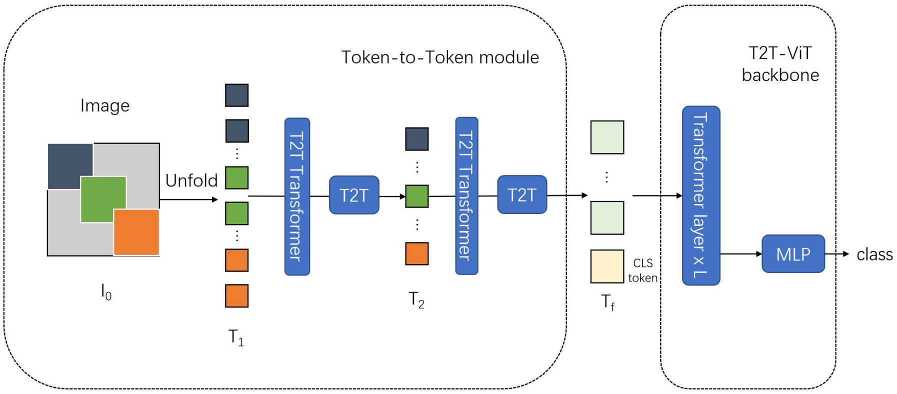


Figure 4. A block figure of Tokens-To-Token ViT. Input images are embedded into different tokens, which extract local structural features. Token patches are further introduced into classic transformer blocks.

An outstanding modification in the T2T-ViT is the soft-split process. Structured images are divided into overlapping tokens that maintain strong correlations with their neighboring tokens. This overlapping strategy allows local features to be effectively extracted by aggregating information from tokens within each split patch.

Mathematically, the T2T module can be expressed as follows:

$$\begin{aligned} T_i' &= MLP(MSA(T_i)), \\ I_i &= Reshape(T_i'), \\ T_{i+1} &= softsplit(I_i). \end{aligned} \tag{5}$$

In addition, the authors point out that the traditional ViT backbone is not efficient for computer vision (CV) tasks due to redundancy across many channels. To address this limitation in feature maps, the T2T-ViT introduces a new backbone, mathematically expressed as follows:

$$\begin{aligned} T_{f_0} &= [t_{cls}; T_f] + E, \\ T_{f_i} &= MLP(MSA(T_{f_{i-1}})), \\ y &= fc(LN(T_{fb})). \end{aligned} \tag{6}$$

This backbone adopts a deep–narrow structure, effectively reducing the embedding dimension. Experimental results [45] demonstrate that this new backbone efficiently enhances feature representation while lowering computational complexity.

Although the T2T-ViT does not exhibit significant advantages over contemporary models such as DeiT, it introduces an orthogonal strategy for token aggregation. This approach has inspired subsequent research focused on improving local feature representation and combining the strengths of transformers and CNNs.

3.4. Combination with CNN Concepts

Some researchers [30,31,46] have suggested that self-attention can be viewed as a more generalized and complex version of convolutional neural networks (CNNs). While self-attention excels at capturing global information with learnable receptive fields, CNNs focus on spatially neighboring pixels, which are highly correlated. CNNs also excel at extracting multi-level feature information through dimension conversion [47].

Interestingly, CNNs can be interpreted as multi-head self-attention networks restricted to a fixed receptive field. Their ability to efficiently capture image features allows them to outperform transformers, especially when pre-trained on less extensive datasets.

Due to these inherent properties and similarities between CNNs and transformers, researchers have explored two main strategies: 1. Using CNNs to pre-extract effective image features before introducing them into transformer models [18,48]. 2. Designing hybrid architectures that integrate CNN and transformer components into a unified framework [49,50].

3.4.1. Conditional Position-Encoding Vision Transformer

The classic self-attention mechanism [1] relies on position encoding to embed positional information into patches, where the length of the position embedding remains fixed. Interpolation is commonly employed to adjust the length of the position encoding to accommodate varying dataset sizes. However, subsequent research has highlighted limitations in these conventional position-embedding strategies, leading to proposed improvements such as relative position representation [7,26].

The Conditional Position-Encoding Vision Transformer (CPVT) [18] revisits the effectiveness of positional information derived from CNNs [27,28] and introduces a new layer called the Positional-Encoding Generator (PEG) to replace the original position-embedding step. The PEG computes efficient, adaptive-length positional embeddings for the ViT model. The mathematical steps of PEG are as follows:

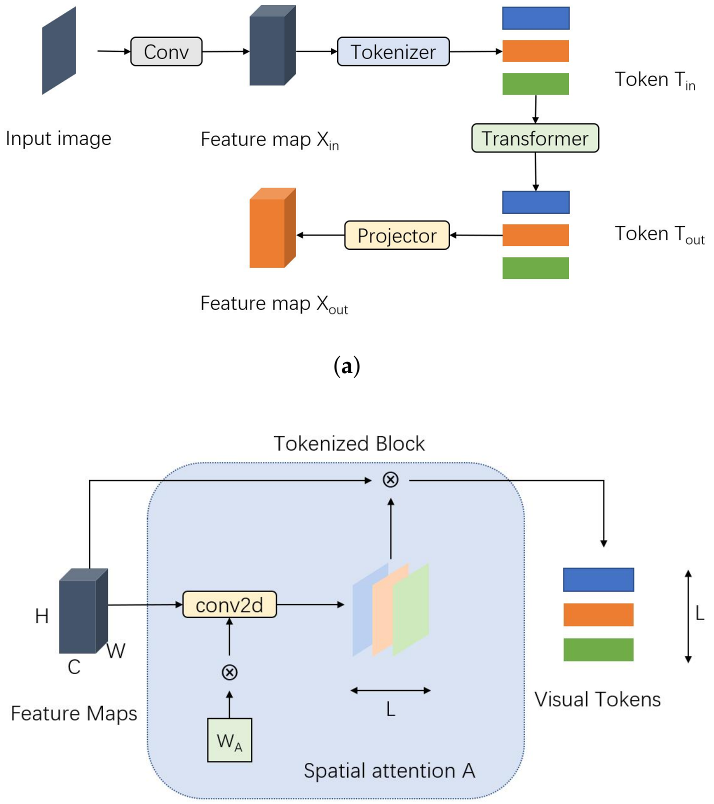
1. Keep the class token invariant and reshape embedded patches $X \in \mathbb{R}^{B \times N \times C}$ back into a 2D tensor $X' \in \mathbb{R}^{B \times H \times W \times C}$.
2. Apply a convolution kernel $k(k \geq 3)$ and perform 2D convolutions on X' to produce an output tensor $X'' \in \mathbb{R}^{B \times N \times C}$ with zero-padding of $\frac{k-1}{2}$.
3. Concatenate the class token with X'' to form the PEG output, which serves as the embedded input sequence for the transformer encoder.

In PEG blocks, zero-padding in convolution acts as a reference point for each patch. The convolution operation efficiently extracts relative positional information between patches and their reference point, enabling the generation of adaptive-length position embeddings suitable for transformers.

Experimental results [18] reveal that the PEG introduces only a negligible increase in computational complexity. Despite this minimal overhead, CPVT achieves better performance and exhibits higher adaptability compared to the DeiT model. Notably, PEG does not alter the core self-attention mechanisms, which means it can seamlessly integrate with other improvement strategies.

3.4.2. Token-Based Visual Transformer

CNNs excel at extracting low-level image features such as corners and edges. Wu et al. [48] introduced Visual Transformers (VT), leveraging CNNs to sample input images into semantic visual tokens using spatial attention, while transformers model relationships between these tokens. The objective is to constrain transformers to operate within the semantic token space, which proves highly efficient for representing and processing high-level concepts. Similar to DeiT [11], ImageNet is sufficient for pre-training this model. The block diagram of VT is shown in Figure 5.



(**a**)

(**b**)

Figure 5. Visual Transformers model: (**a**) Block figure of VT. Convolution extends the dimension of input images and extracts local features, which enable them to be tokenized. (**b**) Block figure of Tokenizer. Visual tokens are formed by projecting feature maps.

The most significant modification in VT compared to the original ViT model is the introduction of the Tokenizer. The operation steps can be summarized as follows:

1. Given an input image $X \in \mathbb{R}^{HW \times C}$, apply a convolution kernel $W_A \in \mathbb{R}^{C \times L}$ to extract L groups of vectors.
2. Apply the *softmax* function on the L groups of vectors to obtain a tensor $A \in \mathbb{R}^{HW \times L}$. The purpose is to project each pixel $X_p \in \mathbb{R}^C$ into one of the L semantic groups.
3. Compute $A^T X = T \in \mathbb{R}^{L \times CT}$, where T represents visual tokens and $L \ll HW$.

The Tokenizer enables transformers to focus on L tokens rather than all HW pixels. This design significantly reduces the computational burden while improving the ability to extract high-level features.

Wu et al. also introduced an improved version of the Tokenizer, called the Recurrent Tokenizer [48]. This mechanism uses visual tokens T' from the previous layer to guide the formation of the current tokens T:

$$\begin{aligned}\mathbf{W}_R &= \mathbf{T}'\mathbf{W}_{T\to R},\\ \mathbf{T} &= softmax_{HW}(\mathbf{X}\mathbf{W}_R)^T\mathbf{X}.\end{aligned} \tag{7}$$

In the output step, a projector re-transforms visual tokens back into a feature map:

$$\mathbf{X}_{out} = \mathbf{X}_{in} + softmax_L((\mathbf{X}_{in}\mathbf{W}_Q)(\mathbf{T}\mathbf{W}_K)^T)\mathbf{T}, \tag{8}$$

which can then be applied to downstream pixel-level tasks, such as segmentation.

It is worth noting that VT can function both as an independent network architecture or as a component integrated into existing networks. Experimental results [48] demonstrate significant improvements in VT-based ResNet models compared to the original ResNet architecture. The semantic extraction capability of the Tokenizer also helps reduce the reliance on extensive datasets, making transformers more applicable to tasks with limited training data.

3.4.3. Refiner

Previous models have explored combining transformers with CNN architectures. However, a significant limitation in applying certain deep neural network (DNN) properties to transformers is the issue of over-smoothing [18,51]. The Refiner [52] addresses this challenge by introducing a convolution-based Refiner-Attention mechanism, which enhances attention mapping and reduces dependency on extensive datasets. The structure is illustrated in Figure 6.

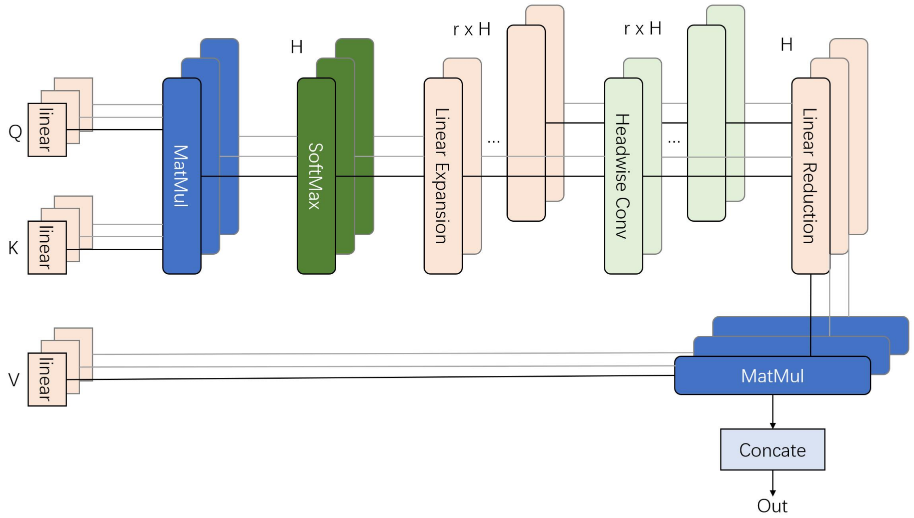


Figure 6. The block figure of Refiner-Attention. Extended $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are linearly projected for subsequent distributed local attention. The feature maps from each head are modeled individually, preserving diversity.

The Refiner-Attention mechanism introduces a linear projection on $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ instead of the conventional subdivision used in classic Multi-Head Self-Attention (MHSA). Expansions on feature maps enhance the quality of extracted features and reduce the dependency on large datasets.

Mathematically, Refiner-Attention can be expressed as follows:

$$A^{h*}_{i,j} = \sum_{a,b=1}^{k} w_{a,b} \cdot A^{h}_{i-\lfloor\frac{k}{2}\rfloor+a,j-\lfloor\frac{k}{2}\rfloor+b} \tag{9}$$

The first step of Refiner-Attention involves introducing linear expansion on the classic attention map. For each head, Refiner-Attention applies convolutions individually to compute distributed local attention (DLA). A subsequent linear reduction step reshapes the DLA map back into the classic attention map format, similar to the ViT, extracting refined attention features from each head.

By integrating convolution operations with self-attention, Refiner-Attention efficiently achieves feature aggregation while retaining the global feature modeling capabilities inherent to transformers.

3.4.4. UniFormer

Li et al. [53] introduced UniFormer, a novel Unified Transformer that seamlessly integrates the strengths of convolution and self-attention, addressing challenges in learning discriminative representations from visual data. Unlike traditional transformer blocks, UniFormer employs relation aggregators with local and global token affinity in shallow and deep layers, effectively managing redundancy and dependency for efficient representation learning.

The flexible stacking of UniFormer blocks forms a robust backbone suitable for various vision tasks. UniFormer achieves a top-1 accuracy of 86.3 on ImageNet-1K without requiring additional training data. It also demonstrates state-of-the-art performance across multiple downstream tasks using only ImageNet-1K pre-training. Furthermore, an efficient variant of UniFormer with a concise hourglass design achieves 2–4 times higher throughput compared to recent lightweight models.

3.4.5. Self-Supervised Masked Convolutional Transformer

Madan et al. [54] introduced a self-supervised masked convolutional transformer block (SSMCTB) designed for anomaly detection across various domains, including medical images and thermal videos. Unlike traditional reconstruction-based methods, SSMCTB integrates reconstruction functionality directly into its core architecture. Its design allows flexible information masking at any layer of a neural network.

Building upon their previous work, the authors extended the block with a 3D masked convolutional layer, a transformer for channel-wise attention, and a novel self-supervised objective based on Huber loss. SSMCTB demonstrates its generality and adaptability by improving performance on five widely recognized benchmarks: MVTec AD, BRATS, Avenue, ShanghaiTech, and Thermal Rare Event. These improvements are achieved when SSMCTB is integrated into state-of-the-art neural models for anomaly detection.

3.4.6. Dynamic Unary Convolution in Transformers

Duan et al. [55] proposed a parallel design approach to integrate transformer architectures with CNNs, diverging from the prevalent sequential structures. Recognizing that multi-head self-attention on convolutional features primarily captures global correlations, they introduced two parallel modules to enhance transformers.

The dynamic local enhancement module uses convolution to capture local information dynamically, enhancing positive patches while suppressing less informative ones. Meanwhile, the unary co-occurrence excitation module focuses on mid-level structural features, leveraging convolution to actively identify local co-occurrence patterns between patches.

The resulting Dynamic Unary Convolution in Transformer (DUCT) blocks are assembled into a deep architecture and comprehensively evaluated across multiple computer vision tasks, including image classification, segmentation, retrieval, and density estimation. Experimental results demonstrate that DUCT outperforms existing sequentially designed structures, showcasing superior performance and efficiency across diverse tasks.

3.4.7. Transformers for Image Segmentation

Gustavo et al. [56] conducted a meta-analysis on the use of multi-modal medical transformers for oncology image segmentation, focusing on the BraTS2021 and HECKTOR2021 datasets. Two modalities, single-stream and multiple-stream, are explored using visio-linguistic representations. Fourteen architectures are evaluated based on dice similarity coefficient (DSC) and average symmetric surface distance (ASSD) metrics, along with cost indicators such as trainable parameters and multiply-accumulate operations (MACs). Results indicate that multi-path hybrid CNN-transformer models improve segmentation accuracy but may require more computational time and larger model sizes compared to traditional methods.

Shiri et al. [57] discussed a federated learning (FL) framework for multi-institutional PET/CT image segmentation, addressing challenges in sharing datasets across different centers due to legal, ethical, and privacy issues. The dataset consists of 328 head and neck cancer patients from 6 centers, and a pure transformer network is implemented for segmentation. Seven FL algorithms, including clipping, zeroing, federated averaging, lossy compression, robust aggregation, secure aggregation, and Gaussian differentially private FedAvg, are evaluated. Results show comparable performance between centralized and FL algorithms, with SeAg and GDP-AQuCl performing slightly better. Overall, FL-based algorithms demonstrate promising performance for head and neck tumor segmentation from PET/CT images, outperforming single center-based approaches.

Ding et al. [58] introduced the FTransCNN model, combining a CNN and Transformer for medical image segmentation. The model employs a fuzzy fusion strategy through a new fuzzy fusion module to jointly utilize features extracted by both CNN and transformer. Channel attention enhances global information from the transformer, spatial attention refines local details from CNN features, and a Hadamard product captures fine-grained interactions. The Choquet fuzzy integral suppresses heterogeneity and uncertainty in fused features. FTransCNN incorporates a fuzzy attention fusion module for hierarchical upsampling, effectively capturing low-level spatial features and high-level semantic context. Experimental results on Chest X-ray and Kvasir-SEG datasets demonstrate superior segmentation performance compared to state-of-the-art models.

Li et al. [59] presented UCFilTransNet, a transformer-based model with a Cross-Filter Transformer (CFTrans) block for enhanced segmentation accuracy. UCFilTransNet redesigns the transformer structure in the frequency domain to improve local information and long-range dependencies, considering various frequencies. To boost global information, it incorporates a residual pyramid squeeze-excitation (RPSA) module in the bottleneck. UCFilTransNet outperforms state-of-the-art methods on two datasets with minimal parameters (24.88 M) and low computational complexity (19.71 G). Experimental results affirm the effectiveness of the proposed CFTrans and RPSA modules for CT image segmentation.

3.4.8. Image Fusion Transformers

Karacan [60] introduced a Multi-image Transformer (MiT) for Multi-Focus Image Fusion (MFIF), inspired by the Spatial–Temporal Transformer Network (STTN). Unlike previous MFIF approaches that primarily rely on CNNs, the author leverages the global connectivity of Vision Transformers. Named MiT-MFIF, the model achieves effective global connection modeling across multiple input images. Various modifications to the baseline transformer enable the utilization of ViTs in MFIF tasks. Comprehensive experiments on standard MFIF datasets demonstrate the effectiveness of MiT-MFIF, outperforming state-of-the-art methods without requiring post-processing steps, as seen in GAN-based competitors.

Yang et al. [61] presented the Semantic Perceptive Infrared and Visible Image Fusion Transformer (SePT). The proposed SePT employs a combination of CNN modules for local feature extraction and transformer-based modules for learning long-range dependencies. Additionally, it incorporates two semantic modeling modules using transformer architecture to handle high-level semantic information. One module maps shallow features to deep semantics, while the other learns deep semantic information from various receptive fields. The fused results are obtained through a combination of local features, long-range dependencies, and semantic features. Extensive comparison experiments showcase the superiority of SePT over other advanced fusion approaches.

Mustafa et al. [62] proposed a multisensor image fusion framework combining visible (VI) and infrared (IR) images using Vision Transformers and graph attention. The framework leverages the internal patch-recurrence property of source images, enhancing feature representation and texture recovery. Transformer blocks capture high-frequency domain-specific information, while the graph attention mechanism utilizes similarity and symmetry information across patches to guide feature learning. The introduced graph attention fusion block (GAFB) improves selectivity and effectiveness in feature learning by identifying significant corresponding local and global details. The GAFB combines complementary information across domains, resulting in a fused image that preserves appropriate apparent intensity. Extensive evaluations on benchmark datasets demonstrate superior performance, with the proposed approach achieving higher SSIM scores than state-of-the-art techniques, such as 0.7552 on the TNO dataset and 0.7673 on the RoadScene dataset.

Zhao et al. [63] introduced Patch-RegNet, a hierarchical deformable image registration (DIR) framework designed to enhance the accuracy and speed of CT-MR and MR-MR registration in head-and-neck MR-Linac treatments. The framework involves whole-volume global registration, patch-based local registration, and patch-based deformable registration. The ViT-Morph model, a combination of CNN and ViT, is employed for patch-based DIR, using a modality-independent neighborhood descriptor as the similarity metric. Trained and tested on image pairs from 42 patients, Patch-RegNet outperforms traditional and deep learning-based registration methods, demonstrating significantly improved DIR accuracy for both CT-MR and MR-MR registration in head-and-neck MR-guided adaptive radiotherapy.

3.5. Deeper Incorporation

CPVT [18] and VT [48] have demonstrated that certain properties of convolutions can enhance the performance of transformers. These improvements are achieved by introducing CNNs to either augment or replace specific layers within transformer models. Building on these insights, researchers have explored deeper combinations between the two neural network architectures [49,50].

The Convolutional Vision Transformer (CvT) [49] serves as a representative model that integrates CNNs and transformers. CvT leverages key benefits of CNNs, including local receptive fields, shared weights, and spatial subsampling, to enhance transformer performance. The structure of CvT is shown in Figure 7.

Similar to CPVT [18] and VT [48], CvT introduces a Convolutional Token Embedding layer before passing image patches into transformer blocks. This layer extracts features directly from the original image or tokens. Due to the ability of convolution to implicitly capture relative positional information [18,27,28], CvT eliminates the need for explicit positional encoding on feature tokens.

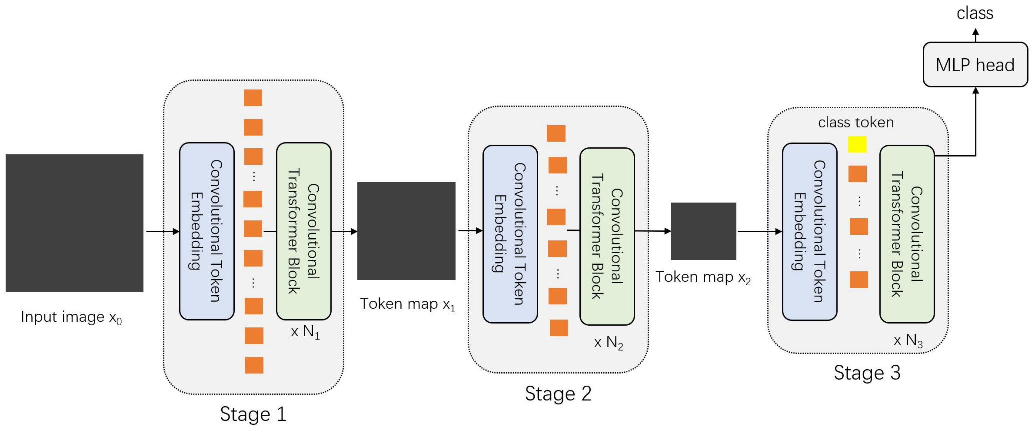


Figure 7. Structure of CvT. The whole model is structured by several convolutional embedding and transformer blocks with decreasing sizes. The class token is only introduced in the last block.

At each stage, the feature resolution (number of tokens) decreases while the feature dimension (token width) increases. The Convolutional Transformer Blocks in CvT resemble the ViT encoder (Figure 2), but with linear projections replaced by a Convolutional Projection process. In essence, CvT employs three convolution kernels to compute the Q, K, and V matrices, replacing the traditional linear transformations used in ViT.

Experimental results [49] show that CvT achieves superior performance compared to ViT and DeiT models when trained on the same datasets. Additionally, CvT significantly reduces computational complexity and parameter counts, demonstrating the practical effectiveness of incorporating CNN structures into transformer models.

The key difference between CeiT [50] and CvT lies in the placement of convolution operations. While CvT applies convolutions across three stages (as shown in Figure 7), CeiT introduces convolutions only once before images are segmented into patches. This approach illustrates that even minimal integration of CNN structures, without extensive architectural modifications, can yield substantial performance improvements in classic transformer models.

3.6. Lite Transformers

A significant challenge of transformer models is their expensive computation cost. It is common for transformer models to require extensive training time and have a large number of parameters. This limitation makes it difficult to apply transformers to tasks with limited datasets, such as medical image analysis [64]. Several lightweight transformer models [65–67] have been introduced, offering an alternative approach to improving data efficiency and inspiring further transformer research.

3.6.1. LeViT

LeViT [68] is an improved version of DeiT [11], introducing a pyramid structure that significantly increases inference speed across various devices. The network structure is shown in Figure 8. Similar to VT [48] and CvT [49], LeViT incorporates CNNs into transformers to extract feature tokens, with a GELU activation function applied after each convolution process. Attention blocks between stages also introduce sub-sampling, a concept borrowed from CNNs. Linear transformations in traditional ViT are replaced with 1×1 convolutions combined with batch normalization, improving computational efficiency.

A significant modification from the original ViT model is the absence of a class token in LeViT. Instead, the patch sequence is transformed into a tensor with dimensions of $512 \times 4 \times 4$. An average pooling layer then reshapes this tensor into a vector with a dimension of 512, which serves as the input for the supervised and distillation classifiers inherited from DeiT (Figure 3). The final predicted label is calculated by averaging the outputs from both classifiers.

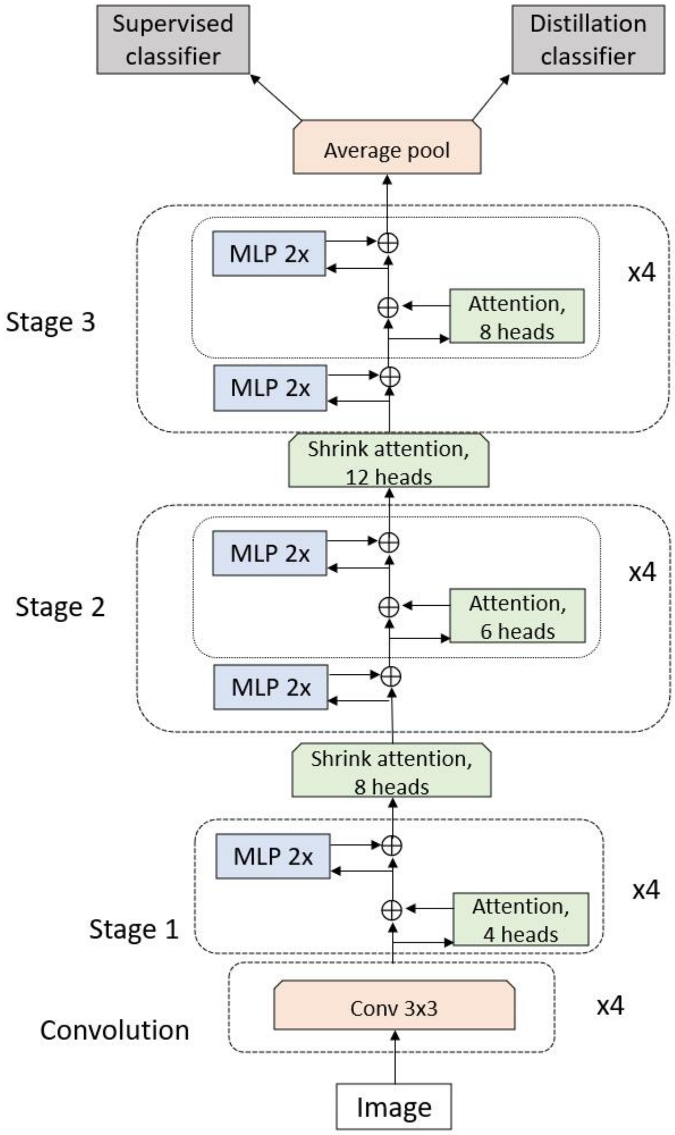


Figure 8. Block figure of LeViT. Input images are first processed by convolution to model local features. The feature map is extended by transformer and MLP blocks with increasing sizes. The average pooling layer reshapes the output into a vector with a dimension of 512 that is introduced to supervised and distillation classifiers.

Another highlighted modification is the replacement of traditional position encoding with attention bias:

$$\mathbf{A}^{h}_{(x,y),(x',y')} = \mathbf{Q}_{(x,y),:} \cdot \mathbf{K}_{(x',y'),:} + \mathbf{B}^{h}_{|x-x'|,|y-y'|} \tag{10}$$

where **B** is the bias tensor, which encodes relative positional information between sequence elements. The objective is to incorporate positional embedding in every layer rather than just at the input stage, preventing the loss of positional information during deeper processing.

Experimental results [68] demonstrate that LeViT requires less than half the computational cost of DeiT while maintaining similar accuracy. Furthermore, LeViT achieves a training speed approximately three times faster than EfficientNet when trained on a CPU.

3.6.2. Compact Convolutional Transformer

The Compact Convolutional Transformer (CCT) [69] is another improved model based on ViT, focusing on overcoming the data-hungry nature of transformer models. CCT replaces the traditional patching process in ViT with convolution layers, preserving local spatial information and implicitly embedding positional information.

The most significant modification in CCT is the introduction of a new layer called Sequence Pooling. The process can be described by the following steps:

1. Use pooling to concentrate data information in the entire sequence: $x_L = f(x_0) \in \mathbb{R}^{b \times n \times d}$.
2. Introduce a linear layer g and softmax activation function on x_L: $x'_L = softmax(g(x_L)^T) \in \mathbb{R}^{b \times 1 \times n}$.
3. Compute $z = x_L x'_L \in \mathbb{R}^{b \times 1 \times d}$.

Sequence Pooling can be interpreted as a weighted average process on input data sequences. The pooled sequences are then passed directly into the MLP classifier layer, eliminating the need for a class token.

Although the CCT structure is extremely simple, experimental results [69] demonstrate significant improvements in both speed and accuracy. A notable feature of this model is its ability to deliver satisfying results when trained on relatively small datasets, such as CIFAR-10, without requiring extensive pre-training or additional guidance. Researchers are encouraged to explore applying similar models to specialized image classification tasks involving limited datasets.

3.6.3. More Lite Transformers

Rao et al. [70] introduced a novel approach to accelerate vision models by leveraging spatial sparsity in visual data. Their dynamic token sparsification framework progressively prunes redundant tokens based on their importance, recognizing that accurate image recognition depends on a subset of informative regions. The framework employs a lightweight prediction module to estimate token importance and applies hierarchical pruning across layers. While inspired by sparse attention in ViTs, this method extends to various architectures, including CNNs and hierarchical ViTs, demonstrating effectiveness across diverse visual tasks. Hierarchical token pruning achieves a significant reduction in FLOPs (31–35 percent) and improves throughput by over 40 percent with less than a 0.5 percent drop in accuracy for ViTs.

Wu et al. [71] introduced Progressive Shift Ladder Transformer (PSLT), a lightweight transformer backbone designed to reduce computational resources. PSLT employs a ladder self-attention block with multiple branches and a progressive shift mechanism. The ladder self-attention block models local self-attention in each branch, while the progressive shift mechanism expands the receptive field through branch interaction. Despite having nearly one-third of the parameters and FLOPs, PSLT effectively models long-range interactions and achieves a top-1 accuracy of 79.9 percent on ImageNet-1K with only 9.2 million parameters and 1.9 G FLOPs.

Wang et al. [72] proposed Quantformer, an extremely low-precision ViT designed for efficient inference. Unlike conventional quantization techniques, Quantformer ensures self-attention rank consistency and applies group-wise discretization for patch features, minimizing rounding and clipping errors. Experimental results show that Quantformer outperforms existing quantization techniques across multiple ViT architectures in tasks like image classification and object detection.

Mou and Zhang [73] introduced TransCL, a transformer-based compressive learning framework tailored for large-scale images with arbitrary compressive sensing ratios. TransCL employs a learnable block-based compressed sensing strategy and processes image blocks as sequences through a transformer backbone. Extensive experiments show state-of-the-art performance in image classification and semantic segmentation, even at extremely low sensing ratios.

With these lightweight transformer models addressing efficiency concerns, we now turn our attention to deeper architectural innovations in transformers.

3.7. Deeper Transformer

In recent years, CNNs have undergone significant improvements through deeper structures [4–6]. Stacking convolution layers allows deep CNNs to generate richer and more complex representations for input images. Inspired by the success of deep learning, researchers have explored whether increasing the depth of transformer architectures could similarly enrich extracted image features and yield improvements comparable to deep CNNs.

3.7.1. Class-Attention in Image Transformers

The Class-Attention in Image Transformers (CaiT) represents a refinement of the DeiT architecture [11]. Touvron et al. introduced an improved normalization strategy referred to as LayerScale and a novel architecture for processing class embeddings known as the class-attention layer.

The mathematical formulation of LayerScale is given as follows:

$$\begin{aligned} x_l' &= x_l + diag(\lambda_{l,1}, \ldots, \lambda_{l,d}) \times SA(LN(x_l)) \\ x_{l+1} &= x_l' + diag(\lambda_{l,1}', \ldots, \lambda_{l,d}') \times FFN(LN(x_l')) \end{aligned} \tag{11}$$

where *diag* denotes diagonal matrices, λ represents learnable weights, and SA and FFN correspond to self-attention and feed-forward network layers in the ViT structure (Figure 2). The weights λ are initialized and progressively reduced with increasing network depth.

In simpler terms, LayerScale can be interpreted as a multiplication between the self-attention output and a diagonal matrix. Experimental results [74] indicate that this normalization strategy enhances training efficiency without compromising the representational capacity of the transformer architecture.

Touvron et al. also revised the concept of class embedding and identified a contradiction in classic transformer architectures: the learned weights must guide self-attention while simultaneously summarizing classification information. To address this conflict, class embedding is processed separately from self-attention in CaiT, as illustrated in Figure 9.

In CaiT, self-attention operates without the class token in the early layers, focusing solely on updating patch embeddings through deeper structures. Towards the end of the self-attention process, patch embeddings remain unchanged, and the class token is introduced into the sequence. In the subsequent class-attention layers, only the class token is updated, while patch embeddings remain invariant.

This design decouples image feature extraction and classifier training into distinct stages. Self-attention focuses on deep feature extraction without being influenced by the class token, while the class token serves as a summary representation for downstream classification tasks. Experimental results suggest that two class-attention layers are sufficient for image classification tasks.

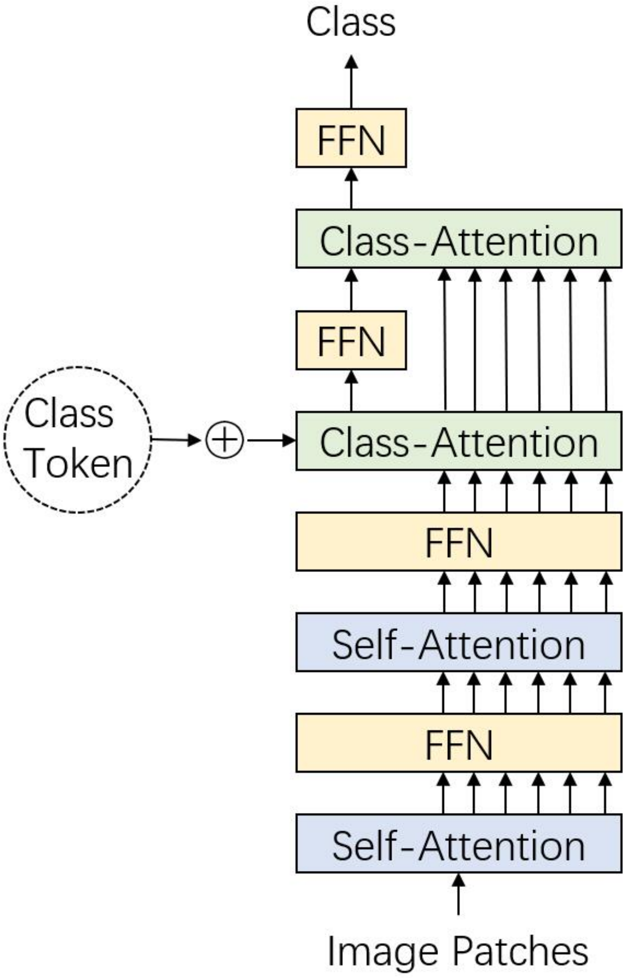


Figure 9. Structure of CaiT. In early layers of self-attention, the class token is not introduced and embedded image patches are modeled. In the following layers of class attention, the class token is introduced and modeled, while the image tokens are kept invariant.

3.7.2. DeepViT

Wang et al. [51] investigated the effects of increasing the depth of the original ViT model. They observed that as layers increased, the attention maps became increasingly similar across layers, indicating that image feature representations ceased to evolve effectively. This phenomenon is referred to as attention collapse or over-smoothing. Similar observations have been reported in other studies exploring deeper transformer architectures [74].

Wang et al. noted that in multi-head self-attention, attention maps from different heads tend to exhibit low similarity. Based on this observation, they introduced an improved attention mechanism referred to as re-attention. This mechanism exchanges information between attention heads using a linear transformation Θ and reconstructs attention maps as follows:

$$ReAtt(Q,K,V) = Norm(\Theta^T(Softmax(\frac{QK^T}{\sqrt{d}})))V \tag{12}$$

Re-attention serves as an alternative to traditional self-attention in transformer models (Figure 2). Notably, re-attention employs BatchNorm instead of the traditional LayerNorm.

This reconstruction process leverages the low-similarity information present between different attention heads to refine the attention maps. Experimental results [51] show that re-attention effectively mitigates attention collapse by maintaining diversity across attention heads. Furthermore, these improvements are achieved without a significant increase in parameters or computational complexity.

3.8. Aggregating Nested Transformers

One explanation for the data-hungry nature of transformer models is their lack of inductive bias. Research suggests that reducing the range of self-attention can improve the training efficiency of transformers [75]. Some researchers have proposed using local self-attention within specific image blocks instead of global attention. This approach led to the development of complex transformer models and mechanisms, including the TNT [76], HaloNets [31], and Swin Transformer [77]. However, these models often suffer from increased computational complexity and larger parameter sizes.

The Aggregating Nested Transformers (NesT) [78] address these challenges with a simplified and more efficient approach to local self-attention. In NesT, basic transformer layers are stacked to process non-overlapping image blocks individually. The aggregation function facilitates cross-block self-attention, enabling interaction between image block patches. The architecture is illustrated in Figure 10.

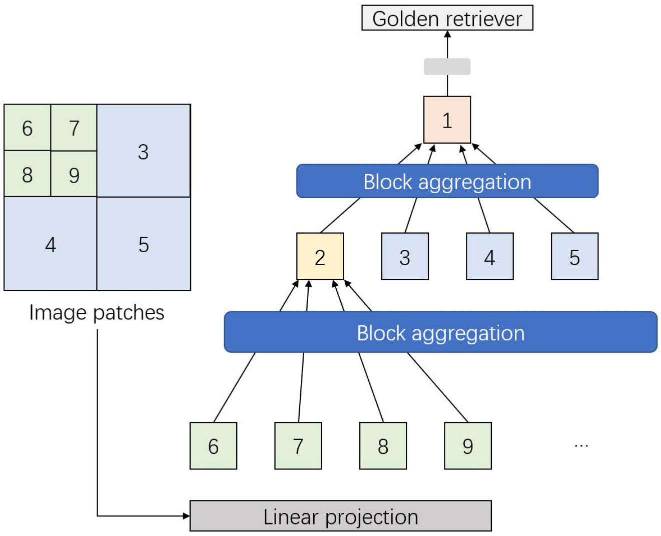


Figure 10. Architecture of the aggregation function in NesT. An input image is divided into non-overlapping blocks (illustrated with various colors and block numbers on the left) which are aggregated. The aggregated blocks are further stacked and introduced to following multi-head self-attention processing.

The architecture stacks transformer layers, performs self-attention on each image block, and nests them hierarchically using aggregation. Notably, convolution operations are introduced during the block aggregation process. Mathematically, the process can be expressed as follows:

$$\begin{aligned} MSA_{NesT}(Q,K,V) &= Stack(block_1, \ldots, block_{T_n}), \\ block_i &= MSA(Q,K,V)W^O. \end{aligned} \tag{13}$$

Each block is merged with four spatially connected blocks before aggregation.

Experimental results [78] demonstrate that the NesT model achieves higher accuracy while maintaining a comparable number of parameters to other local-attention-based transformers. Additionally, the computational complexity of NesT is significantly reduced, making it a more practical choice for large-scale image tasks.

3.9. Cross-Attention

Some researchers propose new attention processes that apply self-attention on individual image blocks while fusing tokens [79,80] to capture multi-scale feature representations [81,82]. Cross-attention has shown strong capabilities in extracting robust image features while maintaining computational efficiency.

The Cross-Attention Multi-Scale Vision Transformer (CrossViT) [80] exemplifies the application of cross-attention in image classification. It introduces an effective token fusion method tailored for transformers. Patch sizes have a significant impact on both accuracy and computational complexity. Fine-grained patches offer improved accuracy but come with higher computational costs. CrossViT attempts to strike a balance between these trade-offs. The architecture is shown in Figure 11.

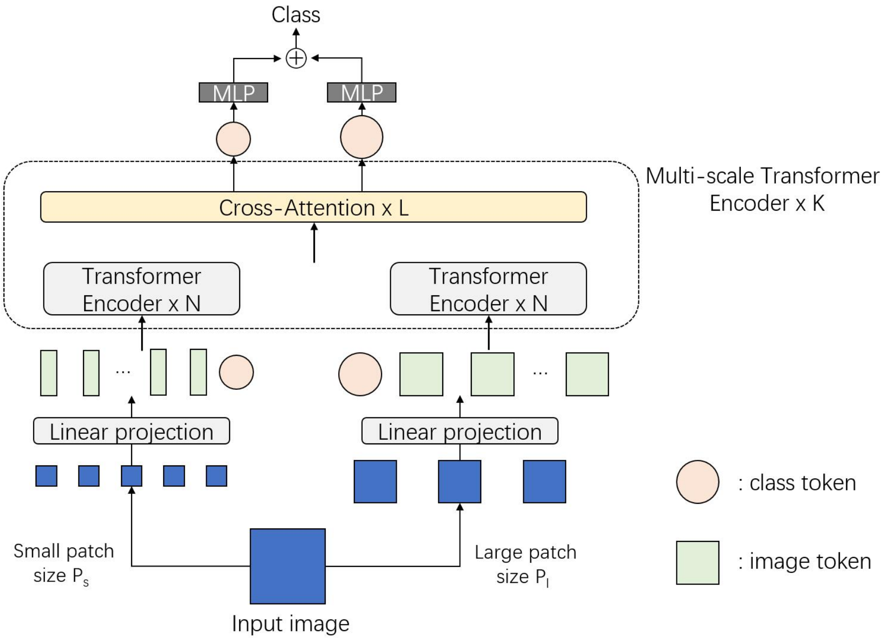


Figure 11. Architecture of CrossViT. An input image is embedded with different patch sizes. Patches with different sizes are projected and introduced to two transformer encoders individually. Cross-attention layers model feature information from attention maps and present a final MLP classifier.

The CrossViT architecture can be considered as using an additional cross-attention layer to fuse two individual ViT blocks with different patch sizes. The mechanism of cross-attention can be divided into the following steps:

1. For the large branch, concatenate the class token with patch tokens from the small branch:
$$x'^{l} = [f^{l}(x^{l}_{cls}) \| x^{s}_{patch}], \tag{14}$$
where $f^{l}(\cdot)$ is a projection function.
2. Compute cross-attention using concatenated x'^{l}:
$$\begin{gathered}\mathbf{Q} = x'^{l}_{cls}\mathbf{W}_Q, \mathbf{K} = x'^{l}\mathbf{W}_K, \mathbf{V} = x'^{l}\mathbf{W}_V, \\ \mathbf{A} = softmax(\mathbf{Q}\mathbf{K}^T/\sqrt{C/h}), AC(x'^{l}) = \mathbf{AV}.\end{gathered} \tag{15}$$

It is noted that only the class token is applied in the computation of Q. This modification improves the efficiency of the attention process.

The authors also suggest using multi-head cross-attention (MCA) without any feed-forward network (FFN) layer after attention:

$$\begin{gathered} y'^{l}_{cls} = f^{l}(x^{l}_{cls}) + MCA(LN([f^{l}(x^{l}_{cls}) \| x^{s}_{patch}])), \\ z^{l} = [g^{l}(y^{l}_{cls}) \| x^{l}_{patch}]. \end{gathered} \tag{16}$$

where $g^{l}(\cdot)$ is a back-projection function.

Experimental results [80] demonstrate that CrossViT achieves similar or better accuracy with fewer parameters compared to ViT and DeiT models. This indicates that cross-attention effectively leverages fine-grained patch tokens without significantly increasing parameter size or computational complexity.

3.10. Patch-Wise Loss

Over-smoothing, also referred to as attention collapse, is recognized as a significant limitation in transformer model training. As transformer models grow deeper, the generated tokens become increasingly similar and difficult to distinguish. Consequently, self-attention across different image patches becomes less effective, leading to poor representation quality.

Zhou et al. [51] introduced the re-attention mechanism to address over-smoothing. In parallel, Gong et al. [83] suggested a different approach by focusing on the training process instead of modifying transformer structures. They introduced new loss mechanisms to counteract over-smoothing. Specifically, they use the standard deviation between attention scores of patches, referred to as layer-wise standard deviation, to measure the similarity of patch tokens. Observations indicate that this standard deviation tends to be small across layers in traditional transformer models, highlighting the root cause of over-smoothing.

The authors proposed three distinct loss functions:

Patch-wise cosine loss

$$l_{cos} = \frac{1}{n(n-1)} \sum_{i \neq j} \frac{h_i^T h_j}{||h_i||||h_j||} \tag{17}$$

Patch-wise contrastive loss

$$l_{cons} = -\frac{1}{n} \sum_{i=1}^{n} log \frac{exp(e_i^T h_i)}{exp(e_i^T h_i + exp(e_i^T (\sum_{j=1}^{n} h_i / n))} \tag{18}$$

Patch-wise mixing loss

$$l_{mix} = \frac{1}{n} \sum_{i=1}^{n} l_{ce}(g(h_i), y_i) \tag{19}$$

In these equations, $h = (h_{cls}, h_1, \ldots, h_n)$ represents the patch representations in the last layer. The cosine loss enhances the discriminative power of patch representations by maximizing their diversity. The contrastive loss ensures that corresponding patch representations between early and deeper layers maintain similarity while preserving diversity across layers. The mixing loss incorporates supervisory information for each patch, generating more informative and representative patch features.

Experimental results [83] demonstrate that these three losses improve transformer performance, enhance training stability, and allow for a higher drop path rate. These findings pave the way for further architectural refinements, as discussed in subsequent sections.

3.11. MetaFormers

Yu et al. [84] investigated MetaFormer, an abstracted transformer architecture that generalizes beyond specific token mixer designs. Various baseline models under the

MetaFormer framework, employing basic or traditional token mixers, demonstrated impressive performance.

Key observations from their study include the following: 1. IdentityFormer achieved over 80 percent accuracy on ImageNet-1K using identity mapping as the token mixer. 2. RandFormer, utilizing a random matrix as the token mixer, surpassed IdentityFormer with an accuracy exceeding 81 percent. 3. Even with older token mixers, models derived from MetaFormer consistently outperformed state-of-the-art architectures. For instance, ConvFormer outperformed ConvNeXt, and CAFormer set a new record of 85.5 percent accuracy on ImageNet-1K.

Additionally, a new activation function, StarReLU, was introduced. StarReLU reduced computational costs by 71 percent compared to GELU while simultaneously improving performance. This activation function shows potential not only for MetaFormer-like models but also for other neural network architectures.

3.12. Transformers with Dense Representations for Multiple-Object Tracking

Transformers have demonstrated superior performance across various tasks, including image classification and object detection. However, their direct application to multiple-object tracking (MOT) has been limited due to quadratic computational complexity and insufficient initialization of sparse queries.

Xu et al. [85] addressed these limitations by introducing TransCenter, a Transformer-based MOT architecture designed for accurate tracking with dense representations. TransCenter employs image-related dense detection queries for robust target inference and sparse tracking queries generated by query learning networks in the TransCenter Decoder.

This hybrid approach effectively balances global context awareness with computational efficiency. Extensive ablation studies and comparisons with alternative methods validate TransCenter's superior accuracy and efficiency on public and private MOT benchmarks.

3.13. Adversarial Transformers

Murthy and P. Murali [86] proposed a deep learning methodology for lung cancer classification using chest CT images. Their framework consists of three key stages: 1. Pre-processing with guided bilateral filtering to remove noise. 2. Feature selection using weighted least absolute shrinkage and selection vector regression for dimensionality reduction. 3. Classification using a transformer-aided generative adversarial network (T-GAN).

The model was fine-tuned using dynamic Levy flight chimp optimization. Experimental results demonstrated high accuracy (0.997), precision (0.996), specificity (0.998), and a low RMSE (0.104), with a time complexity of 120 s, underscoring the effectiveness of this method in lung cancer classification.

Zhou et al. [87] introduced a transformer-based adversarial network for image inpainting. Their approach integrates a self-supervised attention module and a hierarchical Swin Transformer in the discriminator for capturing contextual features. A depthwise over-parameterized convolutional layer in the generator further enhances feature extraction.

Experimental evaluations showed that the proposed approach outperformed existing inpainting methods, effectively addressing structural ambiguity and semantic incompleteness issues.

3.14. Other Improvements

Ren et al. [88] introduced a pooling-based Visual Transformer with low-complexity attention hashing (PTLCH) for image retrieval. Their approach combines pooling-based Vision Transformer features with low-complexity attention modules, enriching contextual information and improving retrieval accuracy.

Sun et al. [89] proposed Separable Transformer (SeT), which factorizes spatial self-attention into pixel-wise local attention and patch-wise global attention. This reduces computational costs while preserving both local and global feature interactions.

Dai et al. [90] enhanced the DETR model with Unsupervised Pre-training DETR (UP-DETR). Their approach uses a random query patch detection task for pre-training, significantly improving convergence speed and average precision in object detection tasks.

These advancements collectively demonstrate the versatility and adaptability of transformers across diverse vision tasks, paving the way for continued innovation.

4. Comparisons

In this section, we compare the introduced ViT models from Section 3 with some representative CNN models.

In Table 1, we highlight models specifically designed for relatively small datasets, such as CIFAR-10 and CIFAR-100 [91]. Both datasets contain 60,000 images and are frequently used as benchmarks for image classification tasks. The results illustrate the potential of applying lightweight ViT models on smaller datasets for specific classification tasks without requiring extensive pre-training on large image datasets.

Table 1. Comparison on ViTs and typical CNNs models on parameters, FLOPs, Top-1 accuracy (in %) with training on CIFAR [91].

Model	Params. (M)	FLOPs. (G)	CIFAR10 Top-1	CIFAR100 Top-1
CCT-2/3x2 [69]	0.28	0.03	89.17	66.90
CCT-7/3x2 [69]	3.85	0.28	93.65	74.77
CCT-7/3x1 [69]	3.76	0.95	94.72	76.67
NesT-T [78]	17	5.8	96.04	78.69
NesT-S [78]	38	10.4	96.97	81.70
NesT-B [78]	68	17.9	97.20	82.56
ResNet164-v2 [5]	1.70	0.56	94.54	75.67
ResNet1001-v2 [5]	10.33	1.55	95.08	77.29
MobileNetV2/1.0 [92]	2.24	0.007	89.07	63.69
MobileNetV2/2.0 [92]	8.72	0.024	91.02	67.44

Similarly, Table 2 compares models trained on ImageNet [40], ImageNet Real [93], and ImageNet V2 [94]. The results include parameters (in millions), FLOPs (in giga operations), and top-1 classification accuracy (in percentage). This table offers an intuitive comparison of how modifications in ViT architectures affect model accuracy, computational complexity, and parameter count.

Table 2. Comparison on ViTs and typical CNNs models on parameters, FLOPs, Top-1 accuracy (in %) with training on ImageNet [40], ImageNet Real [93] and ImageNet V2 [94].

Model (Transformers)	Params. (M)	FLOPs. (G)	ImNet Top-1	Real Top-1	V2 Top-1
ViT-B/16 [10]	86.4	17.7	77.9	83.6	-
ViT-L/16 [10]	307	63.6	76.5	82.2	-
DeiT-Ti [11]	6	1.3	72.2	80.1	60.4
DeiT-S [11]	22	4.6	79.8	85.7	68.5
DeiT-B [11]	86	17.6	81.8	86.7	71.5
T2T-ViT-12 [45]	6.8	2.2	76.5	-	-
T2T-ViT-14 [45]	21.5	6.1	81.5	-	-
T2T-ViT-24 [45]	64.1	14.1	82.3	-	-

Table 2. *Cont.*

Model (Transformers)	Params. (M)	FLOPs. (G)	ImNet	Real	V2
			Top-1	Top-1	Top-1
CPVT-Ti [18]	6	-	75.9	-	-
CPVT-S [18]	22	-	81.5	-	-
CPVT-B [18]	86	-	82.3	-	-
VT-R18 [48]	11.6	1.6	76.8	-	-
VT-R34 [48]	19.2	3.2	79.9	-	-
VT-R50 [48]	21.4	3.4	80.6	-	-
VT-R101 [48]	41.5	7.1	82.3	-	-
Refined-ViT-s [52]	25	7.2	83.6	88.3	-
Refined-ViT-M [52]	55	13.5	84.6	88.9	-
Refined-ViT-L [52]	81	19.1	84.9	89.1	-
CvT-13 [49]	20	4.5	81.6	86.7	70.4
CvT-21 [49]	32	7.1	82.5	87.2	71.3
CvT-W24$_{\uparrow 384}$ [49]	277	193.2	87.7	90.6	78.8
CeiT-T [50]	6.4	1.2	76.4	83.6	-
CeiT-S [50]	24.2	4.5	82.0	87.3	-
LeViT-128 [68]	9.2	0.4	76.6	83.1	64.3
LeViT-192 [68]	10.9	0.7	80.0	85.7	68.0
LeViT-256 [68]	18.9	1.1	81.6	86.8	70.0
LeViT-384 [68]	39.1	2.2	82.6	87.6	71.3
CaiT-S36 [74]	68	13.9	83.3	88.0	72.5
CaiT-M36$_{\uparrow 384}\Upsilon$ [74]	271	173.3	86.1	90.0	76.3
CaiT-M48$_{\uparrow 448}\Upsilon$ [74]	356	329.6	86.5	90.2	76.9
DeepViT-S [51]	27	6.2	81.4	-	-
DeepViT-L [51]	55	12.5	82.2	-	-
Model (CNNs)	**Params. (M)**	**FLOPs. (G)**	**ImNet**	**Real**	**V2**
			Top-1	**Top-1**	**Top-1**
RegNetY-16GF [44]	84	16.0	82.9	88.1	72.4
ResNet-50 [19]	25	4.1	76.2	82.5	63.3
ResNet-101 [19]	45	7.9	77.4	83.7	65.7
ResNet-152 [19]	60	11	78.3	84.1	67.0
EfficientNet-B5 [4]	30	9.9	83.6	88.3	73.6
EfficientNet-B7 [4]	84.3	66	37.0	84.3	-
NFNet-F0 [95]	72	12.4	83.6	88.1	72.6
NFNet-F3 [95]	255	114.8	85.7	88.9	74.4
NFNet-F6+SAM [95]	438	377.3	86.5	89.9	75.8

Furthermore, Table 3 presents the experimental results of applying and fine-tuning pre-trained ViT models (trained on ImageNet) on downstream datasets, including CIFAR-10, CIFAR-100, Oxford-IIIT Pets [96], Oxford-IIIT Flowers, and Stanford Cars [97]. These results provide insights into the generalization capability of each improved ViT model across diverse tasks and datasets.

Table 3. Comparison of the application of pre-trained ViTs and typical CNNs models on downstream datasets.

Models	CIFAR-10	CIFAR100	Pets	Flowers	Cars
ViT-B/16 [10]	98.95	91.67	94.43	99.38	-
ViT-L/16 [10]	99.16	93.44	94.73	99.61	-
ViT-H/14 [10]	99.27	93.82	94.82	99.51	-
DeiT-B [11]	99.1	90.8	-	98.4	92.1
DeiT-B distilled [11]	99.1	90.8	-	98.5	93.9
T2T-ViT-14 [45]	97.1	87.1	-	-	-
T2T-ViT-19 [45]	98.3	89.0	-	-	-

Table 3. *Cont.*

Models	CIFAR-10	CIFAR100	Pets	Flowers	Cars
CvT-13 [49]	98.83	91.11	93.25	99.50	-
CvT-21 [49]	99.16	92.88	94.03	99.62	-
CvT-W24 [49]	99.39	94.09	94.73	99.72	-
CeiT-T [50]	98.5	88.4	93.8	-	90.5
CeiT-S [50]	99.1	90.8	94.9	-	93.2
CaiT-S36 [74]	99.2	92.2	-	98.8	93.5
CaiT-M36 [74]	99.3	93.3	-	99.0	93.5
CaiT-M36γ [74]	99.4	93.1	-	99.1	94.2

5. Conclusions and Discussions

5.1. Challenges

Although CV transformer models have demonstrated significant potential in image classification tasks using standard benchmark datasets such as ImageNet and CIFAR, their application to domain-specific tasks, such as medical imaging or traffic analysis, remains limited. A key challenge is the lack of inductive bias [10], which makes transformers heavily reliant on large, specific datasets. This reliance restricts their effectiveness in smaller, domain-specific datasets. Limitations in generalization and robustness remain open challenges for future research.

Efficiency is another critical issue. While lightweight transformer models, such as CCT [69] and LeViT [68], have been proposed, experimental results show that achieving a balance between computational efficiency, parameter reduction, and model accuracy remains challenging. Further optimization is necessary to address this trade-off effectively.

5.2. Future Directions

One promising direction is the integration of convolutional operations within transformer architectures. Models such as CCT [69] and NesT [78] demonstrated significant improvements by incorporating convolution-based tokenization instead of simple image patch division. This approach enhances inductive bias and aligns transformer architectures with traditional computer vision paradigms.

Another valuable area of research lies in exploring alternative mechanisms without altering the transformer architecture itself.

- Normalization Mechanisms: Techniques like power-normalization [98] and improved layer normalization [99] have shown advantages over traditional normalization methods in ViTs.
- Loss Mechanisms: Patch-wise loss functions [83] have been effective in improving convergence and reducing over-smoothing in deeper models.
- Hardware-Aware Transformers: Models like HAT [100] optimize transformer architectures for energy-efficient hardware implementations.

Future research could focus on combining these mechanisms into cohesive models, as they often exhibit cooperative effects [74].

Lastly, image–text training paradigms hold great potential. Models such as CLIP [101] and Frozen Pretrained Transformer (FPT) [102] leverage large-scale natural language datasets to improve visual representations. By utilizing vast online resources, such as WIT [103], these approaches reduce dependence on manually labeled datasets and enable more scalable training pipelines.

In summary, while significant progress has been made in transformer-based computer vision architectures, ongoing research must address challenges related to dataset reliance, computational efficiency, and architectural scalability. Collaborative advancements across architecture, training strategies, and hardware optimization are essential for further breakthroughs in the field.

Author Contributions: Conceptualization, L.W., Y.W. and Y.Z.; writing—original draft preparation, Y.D. and Y.W.; writing—review and editing, P.C., L.W. and Y.W.; visualization, Y.D.; supervision, L.W., Y.W. and Y.Z.; project administration, L.W.; funding acquisition, L.W. All authors have read and agreed to the published version of the manuscript.

Funding: We would like to acknowledge the support received from the Ministry of Education, Singapore, under its Research Centre of Excellence award to the Institute for Digital Molecular Analytics (IDMxS, grant: EDUN C-33-18-279-V12) (L.W.).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
2. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
3. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
4. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
5. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 630–645.
6. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
7. Bello, I.; Zoph, B.; Vaswani, A.; Shlens, J.; Le, Q.V. Attention augmented convolutional networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3286–3295.
8. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in Vision: A Survey. *arXiv* **2021**, arXiv:2101.01169. [CrossRef]
9. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A Survey on Visual Transformer. *arXiv* **2020**, arXiv:2012.12556.
10. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
11. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. *arXiv* **2020**, arXiv:2012.12877.
12. Ye, L.; Rochan, M.; Liu, Z.; Wang, Y. Cross-modal self-attention network for referring image segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 9–15 June 2019; pp. 10502–10511.
13. Sun, C.; Myers, A.; Vondrick, C.; Murphy, K.; Schmid, C. Videobert: A joint model for video and language representation learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 9–15 June 2019; pp. 7464–7473.
14. Azad, R.; Kazerouni, A.; Heidari, M.; Aghdam, E.K.; Molaei, A.; Jia, Y.; Jose, A.; Roy, R.; Merhof, D. Advances in medical image analysis with vision Transformers: A comprehensive review. *Med. Image Anal.* **2024**, *91*, 103000. [CrossRef]
15. Liu, Z.; Lv, Q.; Yang, Z.; Li, Y.; Lee, C.H.; Shen, L. Recent progress in transformer-based medical image analysis. *Comput. Biol. Med.* **2023**, *164*, 107268. [CrossRef]
16. Khalil, M.; Khalil, A.; Ngom, A. A Comprehensive Study of Vision Transformers in Image Classification Tasks. *arXiv* **2023**, arXiv:2312.01232.
17. Maurício, J.; Domingues, I.; Bernardino, J. Comparing vision transformers and convolutional neural networks for image classification: A literature review. *Appl. Sci.* **2023**, *13*, 5521. [CrossRef]

18. Chu, X.; Zhang, B.; Tian, Z.; Wei, X.; Xia, H. Do we really need explicit position encodings for vision transformers? *arXiv* **2021**, arXiv:1706.03762.
19. Kolesnikov, A.; Beyer, L.; Zhai, X.; Puigcerver, J.; Yung, J.; Gelly, S.; Houlsby, N. Big transfer (bit): General visual representation learning. *arXiv* **2019**, arXiv:1912.11370.
20. Xie, Q.; Luong, M.T.; Hovy, E.; Le, Q.V. Self-training with noisy student improves imagenet classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10687–10698.
21. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
22. Yin, W.; Kann, K.; Yu, M.; Schütze, H. Comparative study of CNN and RNN for natural language processing. *arXiv* **2017**, arXiv:1702.01923.
23. Cordonnier, J.B.; Loukas, A.; Jaggi, M. On the relationship between self-attention and convolutional layers. *arXiv* **2019**, arXiv:1911.03584.
24. Hu, H.; Zhang, Z.; Xie, Z.; Lin, S. Local Relation Networks for Image Recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
25. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-attention generative adversarial networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 7354–7363.
26. Shaw, P.; Uszkoreit, J.; Vaswani, A. Self-attention with relative position representations. *arXiv* **2018**, arXiv:1803.02155.
27. Islam, M.A.; Jia, S.; Bruce, N.D. How much position information do convolutional neural networks encode? *arXiv* **2020**, arXiv:2001.08248.
28. Islam, M.A.; Kowal, M.; Jia, S.; Derpanis, K.G.; Bruce, N.D. Position, padding and predictions: A deeper look at position information in cnns. *arXiv* **2021**, arXiv:2101.12322. [CrossRef]
29. Zhao, H.; Jia, J.; Koltun, V. Exploring Self-Attention for Image Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
30. Ramachandran, P.; Parmar, N.; Vaswani, A.; Bello, I.; Levskaya, A.; Shlens, J. Stand-Alone Self-Attention in Vision Models. *arXiv* **2019**, arXiv:1906.05909.
31. Vaswani, A.; Ramachandran, P.; Srinivas, A.; Parmar, N.; Hechtman, B.; Shlens, J. Scaling local self-attention for parameter efficient visual backbones. *arXiv* **2021**, arXiv:2103.12731.
32. Li, X.; Yin, X.; Li, C.; Zhang, P.; Hu, X.; Zhang, L.; Wang, L.; Hu, H.; Dong, L.; Wei, F.; et al. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. *arXiv* **2020**, arXiv:2004.06165.
33. Lin, K.; Wang, L.; Liu, Z. End-to-End Human Pose and Mesh Reconstruction with Transformers. *arXiv* **2020**, arXiv:2012.09760.
34. Gidaris, S.; Singh, P.; Komodakis, N. Unsupervised Representation Learning by Predicting Image Rotations. *arXiv* **2018**, arXiv:1803.07728.
35. Wang, S.; Li, B.; Khabsa, M.; Fang, H.; Ma, H. Linformer: Self-attention with linear complexity. *arXiv* **2020**, arXiv:2006.04768.
36. Su, W.; Zhu, X.; Cao, Y.; Li, B.; Lu, L.; Wei, F.; Dai, J. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. *arXiv* **2019**, arXiv:1908.08530.
37. Chen, Y.C.; Li, L.; Yu, L.; El Kholy, A.; Ahmed, F.; Gan, Z.; Cheng, Y.; Liu, J. Uniter: Universal image-text representation learning. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 104–120.
38. Liu, X.; Zhang, F.; Hou, Z.; Wang, Z.; Mian, L.; Zhang, J.; Tang, J. Self-supervised Learning: Generative or Contrastive. *arXiv* **2020**, arXiv:2006.08218. [CrossRef]
39. Jing, L.; Tian, Y. Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4037–4058. [CrossRef] [PubMed]
40. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
41. Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
42. Cubuk, E.D.; Zoph, B.; Shlens, J.; Le, Q.V. Randaugment: Practical automated data augmentation with a reduced search space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 702–703.
43. Hinton, G.E.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
44. Radosavovic, I.; Kosaraju, R.P.; Girshick, R.; He, K.; Dollar, P. Designing Network Design Spaces. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.
45. Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.; Tay, F.E.; Feng, J.; Yan, S. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv* **2021**, arXiv:2101.11986.

46. Gong, J.; Qiu, X.; Chen, X.; Liang, D.; Huang, X. Convolutional Interaction Network for Natural Language Inference. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 1576–1585.
47. Heo, B.; Yun, S.; Han, D.; Chun, S.; Choe, J.; Oh, S.J. Rethinking spatial dimensions of vision transformers. *arXiv* **2021**, arXiv:2103.16302.
48. Wu, B.; Xu, C.; Dai, X.; Wan, A.; Zhang, P.; Yan, Z.; Tomizuka, M.; Gonzalez, J.; Keutzer, K.; Vajda, P. Visual transformers: Token-based image representation and processing for computer vision. *arXiv* **2020**, arXiv:2006.03677.
49. Wu, H.; Xiao, B.; Codella, N.; Liu, M.; Dai, X.; Yuan, L.; Zhang, L. Cvt: Introducing convolutions to vision transformers. *arXiv* **2021**, arXiv:2103.15808.
50. Yuan, K.; Guo, S.; Liu, Z.; Zhou, A.; Yu, F.; Wu, W. Incorporating convolution designs into visual transformers. *arXiv* **2021**, arXiv:2103.11816.
51. Zhou, D.; Kang, B.; Jin, X.; Yang, L.; Lian, X.; Jiang, Z.; Hou, Q.; Feng, J. Deepvit: Towards deeper vision transformer. *arXiv* **2021**, arXiv:2103.11886.
52. Zhou, D.; Shi, Y.; Kang, B.; Yu, W.; Jiang, Z.; Li, Y.; Jin, X.; Hou, Q.; Feng, J. Refiner: Refining Self-attention for Vision Transformers. *arXiv* **2021**, arXiv:2106.03714.
53. Li, K.; Wang, Y.; Zhang, J.; Gao, P.; Song, G.; Liu, Y.; Li, H.; Qiao, Y. UniFormer: Unifying Convolution and Self-Attention for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 12581–12600. [CrossRef]
54. Madan, N.; Ristea, N.C.; Ionescu, R.T.; Nasrollahi, K.; Khan, F.S.; Moeslund, T.B.; Shah, M. Self-Supervised Masked Convolutional Transformer Block for Anomaly Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 525–542. [CrossRef]
55. Duan, H.; Long, Y.; Wang, S.; Zhang, H.; Willcocks, C.G.; Shao, L. Dynamic Unary Convolution in Transformers. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 12747–12759. [CrossRef]
56. Andrade-Miranda, G.; Jaouen, V.; Tankyevych, O.; Cheze Le Rest, C.; Visvikis, D.; Conze, P.H. Multi-modal medical Transformers: A meta-analysis for medical image segmentation in oncology. *Comput. Med. Imaging Graph.* **2023**, *110*, 102308. [CrossRef] [PubMed]
57. Shiri, I.; Razeghi, B.; Vafaei Sadr, A.; Amini, M.; Salimi, Y.; Ferdowsi, S.; Boor, P.; Gündüz, D.; Voloshynovskiy, S.; Zaidi, H. Multi-institutional PET/CT image segmentation using federated deep transformer learning. *Comput. Methods Programs Biomed.* **2023**, *240*, 107706. [CrossRef] [PubMed]
58. Ding, W.; Wang, H.; Huang, J.; Ju, H.; Geng, Y.; Lin, C.T.; Pedrycz, W. FTransCNN: Fusing Transformer and a CNN based on fuzzy logic for uncertain medical image segmentation. *Inf. Fusion* **2023**, *99*, 101880. [CrossRef]
59. Li, L.; Liu, Q.; Shi, X.; Wei, Y.; Li, H.; Xiao, H. UCFilTransNet: Cross-Filtering Transformer-based network for CT image segmentation. *Expert Syst. Appl.* **2024**, *238*, 121717. [CrossRef]
60. Karacan, L. Multi-image transformer for multi-focus image fusion. *Signal Process. Image Commun.* **2023**, *119*, 117058. [CrossRef]
61. Yang, X.; Huo, H.; Li, C.; Liu, X.; Wang, W.; Wang, C. Semantic perceptive infrared and visible image fusion Transformer. *Pattern Recognit.* **2024**, *149*, 110223. [CrossRef]
62. Mustafa, H.T.; Shamsolmoali, P.; Lee, I.H. TGF: Multiscale transformer graph attention network for multi-sensor image fusion. *Expert Syst. Appl.* **2024**, *238*, 121789. [CrossRef]
63. Zhao, Y.; Chen, X.; McDonald, B.; Yu, C.; Mohamed, A.S.; Fuller, C.D.; Court, L.E.; Pan, T.; Wang, H.; Wang, X.; et al. A transformer-based hierarchical registration framework for multimodality deformable image registration. *Comput. Med. Imaging Graph.* **2023**, *108*, 102286. [CrossRef]
64. Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, A.L.; Zhou, Y. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv* **2021**, arXiv:2102.04306.
65. Wu, Z.; Liu, Z.; Lin, J.; Lin, Y.; Han, S. Lite transformer with long-short range attention. *arXiv* **2020**, arXiv:2004.11886.
66. Mehta, S.; Koncel-Kedziorski, R.; Rastegari, M.; Hajishirzi, H. Define: Deep factorized input token embeddings for neural sequence modeling. *arXiv* **2019**, arXiv:1911.12385.
67. Mehta, S.; Ghazvininejad, M.; Iyer, S.; Zettlemoyer, L.; Hajishirzi, H. DeLighT: Very Deep and Light-weight Transformer. *arXiv* **2020**, arXiv:2008.00623.
68. Graham, B.; El-Nouby, A.; Touvron, H.; Stock, P.; Joulin, A.; Jégou, H.; Douze, M. LeViT: A Vision Transformer in ConvNet's Clothing for Faster Inference. *arXiv* **2021**, arXiv:2104.01136.
69. Hassani, A.; Walton, S.; Shah, N.; Abuduweili, A.; Li, J.; Shi, H. Escaping the Big Data Paradigm with Compact Transformers. *arXiv* **2021**, arXiv:2104.05704.
70. Rao, Y.; Liu, Z.; Zhao, W.; Zhou, J.; Lu, J. Dynamic Spatial Sparsification for Efficient Vision Transformers and Convolutional Neural Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 10883–10897. [CrossRef]
71. Wu, G.; Zheng, W.S.; Lu, Y.; Tian, Q. PSLT: A Light-Weight Vision Transformer With Ladder Self-Attention and Progressive Shift. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 11120–11135. [CrossRef] [PubMed]

72. Wang, Z.; Wang, C.; Xu, X.; Zhou, J.; Lu, J. Quantformer: Learning Extremely Low-Precision Vision Transformers. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 8813–8826. [CrossRef]
73. Mou, C.; Zhang, J. TransCL: Transformer Makes Strong and Flexible Compressive Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 5236–5251. [CrossRef]
74. Touvron, H.; Cord, M.; Sablayrolles, A.; Synnaeve, G.; Jégou, H. Going deeper with image transformers. *arXiv* **2021**, arXiv:2103.17239.
75. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The long-document transformer. *arXiv* **2020**, arXiv:2004.05150.
76. Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; Wang, Y. Transformer in transformer. *arXiv* **2021**, arXiv:2103.00112.
77. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv* **2021**, arXiv:2103.14030.
78. Zhang, Z.; Zhang, H.; Zhao, L.; Chen, T.; Pfister, T. Aggregating Nested Transformers. *arXiv* **2021**, arXiv:2105.12723.
79. Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; Liu, W. CCNet: Criss-Cross Attention for Semantic Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
80. Chen, C.F.; Fan, Q.; Panda, R. Crossvit: Cross-attention multi-scale vision transformer for image classification. *arXiv* **2021**, arXiv:2103.14899.
81. Chen, C.F.; Fan, Q.; Mallinar, N.; Sercu, T.; Feris, R. Big-little net: An efficient multi-scale feature representation for visual and speech recognition. *arXiv* **2018**, arXiv:1807.03848.
82. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
83. Gong, C.; Wang, D.; Li, M.; Chandra, V.; Liu, Q. Improve Vision Transformers Training by Suppressing Over-smoothing. *arXiv* **2021**, arXiv:2104.12753.
84. Yu, W.; Si, C.; Zhou, P.; Luo, M.; Zhou, Y.; Feng, J.; Yan, S.; Wang, X. MetaFormer Baselines for Vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 896–912. [CrossRef] [PubMed]
85. Xu, Y.; Ban, Y.; Delorme, G.; Gan, C.; Rus, D.; Alameda-Pineda, X. TransCenter: Transformers with Dense Representations for Multiple-Object Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 7820–7835. [CrossRef]
86. Murthy, S.; Krishna Prasad, P.M. Adversarial transformer network for classification of lung cancer disease from CT scan images. *Biomed. Signal Process. Control* **2023**, *86*, 105327. [CrossRef]
87. Zhou, M.; Liu, X.; Yi, T.; Bai, Z.; Zhang, P. A superior image inpainting scheme using Transformer-based self-supervised attention GAN model. *Expert Syst. Appl.* **2023**, *233*, 120906. [CrossRef]
88. Ren, H.; Guo, J.; Cheng, S.; Li, Y. Pooling-based Visual Transformer with low complexity attention hashing for image retrieval. *Expert Syst. Appl.* **2024**, *241*, 122745. [CrossRef]
89. Sun, S.; Yue, X.; Zhao, H.; Torr, P.H.; Bai, S. Patch-Based Separable Transformer for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 9241–9247. [CrossRef]
90. Dai, Z.; Cai, B.; Lin, Y.; Chen, J. Unsupervised Pre-Training for Detection Transformers. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 12772–12782. [CrossRef]
91. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. 2009. Available online: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf (accessed on 18 January 2024).
92. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
93. Beyer, L.; Hénaff, O.J.; Kolesnikov, A.; Zhai, X.; Oord, A.v.d. Are we done with ImageNet? *arXiv* **2020**, arXiv:2006.07159.
94. Recht, B.; Roelofs, R.; Schmidt, L.; Shankar, V. Do imagenet classifiers generalize to imagenet? In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 5389–5400.
95. Brock, A.; De, S.; Smith, S.L.; Simonyan, K. High-performance large-scale image recognition without normalization. *arXiv* **2021**, arXiv:2102.06171.
96. Parkhi, O.M.; Vedaldi, A.; Zisserman, A.; Jawahar, C. Cats and dogs. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3498–3505.
97. Krause, J.; Stark, M.; Deng, J.; Fei-Fei, L. 3d object representations for fine-grained categorization. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 2–8 December 2013; pp. 554–561.
98. Shen, S.; Yao, Z.; Gholami, A.; Mahoney, M.; Keutzer, K. PowerNorm: Rethinking Batch Normalization in Transformers. In Proceedings of the 37th International Conference on Machine Learning, Online, 13–18 July 2020; Volume 119, pp. 8741–8751.
99. Xiong, R.; Yang, Y.; He, D.; Zheng, K.; Zheng, S.; Xing, C.; Zhang, H.; Lan, Y.; Wang, L.; Liu, T. On Layer Normalization in the Transformer Architecture. In Proceedings of the 37th International Conference on Machine Learning, Online, 13–18 July 2020; Volume 119, pp. 10524–10533.

100. Wang, H.; Wu, Z.; Liu, Z.; Cai, H.; Zhu, L.; Gan, C.; Han, S. Hat: Hardware-aware transformers for efficient natural language processing. *arXiv* **2020**, arXiv:2005.14187.
101. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. *arXiv* **2021**, arXiv:2103.00020.
102. Lu, K.; Grover, A.; Abbeel, P.; Mordatch, I. Pretrained Transformers as Universal Computation Engines. *arXiv* **2021**, arXiv:2103.05247. [CrossRef]
103. Srinivasan, K.; Raman, K.; Chen, J.; Bendersky, M.; Najork, M. WIT: Wikipedia-based Image Text Dataset for Multimodal Multilingual Machine Learning. *arXiv* **2021**, arXiv:2103.01913.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.