Multimedia Systems (2024) 30:109
https://doi.org/10.1007/s00530-024-01312-0

REGULAR PAPER

Vision transformer models for mobile/edge devices: a survey

Seung Il Lee[1] · Kwanghyun Koo[1] · Jong Ho Lee[1] · Gilha Lee[1] · Sangbeom Jeong[1] · Seongjun O[1] · Hyun Kim[1]

Received: 31 July 2023 / Accepted: 4 March 2024
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract
With the rapidly growing demand for high-performance deep learning vision models on mobile and edge devices, this paper emphasizes the importance of compact deep learning-based vision models that can provide high accuracy while maintaining a small model size. In particular, based on the success of transformer models in natural language processing and computer vision tasks, this paper offers a comprehensive examination of the latest research in redesigning the Vision Transformer (ViT) model into a compact architecture suitable for mobile/edge devices. The paper classifies compact ViT models into three major categories: (1) architecture and hierarchy restructuring, (2) encoder block enhancements, and (3) integrated approaches, and provides a detailed overview of each category. This paper also analyzes the contribution of each method to model performance and computational efficiency, providing a deeper understanding of how to efficiently implement ViT models on edge devices. As a result, this paper can offer new insights into the design and implementation of compact ViT models for researchers in this field and provide guidelines for optimizing the performance and improving the efficiency of deep learning vision models on edge devices.

Keywords Vision transformer · Mobile/edge devices · Survey

1 Introduction

Recently, there has been a surge in demand for deep learning-based high-performance vision models on edge devices, such as mobile platform [1–5]. These devices require the ability to perform complex tasks such as applications of human action recognition (HAR) based on inertial sensor analysis, vision, and processing areas while providing user-friendly interfaces and portability. This includes tracking and detection [6, 7], computer engineering [8], and physical sciences [9]. However, compared to servers or cloud-based systems, edge devices have limited computational resources [10, 11], making the need for compact vision models that maintain small sizes while delivering high performance increasingly important.

Meanwhile, the transformer model is flourishing in both the natural language processing (NLP) and computer vision fields. Initially proposed by Vaswani et al. [12], the transformer model introduced the self-attention mechanism to solve the input sequence length problem in NLP. This mechanism generates outputs by considering the relevance between all elements of the input sequence, effectively solving the problem of input sequence length. It effectively addressed the issues with existing sequence models, such

Communicated by J. Gao.

Seung Il Lee and Kwanghyun Koo contributed equally to this work.

✉ Hyun Kim
hyunkim@seoultech.ac.kr

Seung Il Lee
seungil66@seoultech.ac.kr

Kwanghyun Koo
kookh0414@seoultech.ac.kr

Jong Ho Lee
jhlees@seoultech.ac.kr

Gilha Lee
gilha1@seoultech.ac.kr

Sangbeom Jeong
sangbeom@seoultech.ac.kr

Seongjun O
seongjun_o@seoultech.ac.kr

[1] Department of Electrical and Information Engineering, Research Center for Electrical and Information Technology, Seoul National University of Science and Technology, 232 Gongneung-ro, Nowon-gu, Seoul 01811, Korea

Published online: 01 April 2024

as the recurrent neural network (RNN) [13, 14] and long short-term memory (LSTM) [15], and consequently, the transformer model has demonstrated excellent performance in the NLP field. Based on its success, various research has been conducted to apply the transformer model to vision tasks. The vision transformer (ViT) [16], which applies the transformer model from NLP to vision tasks, performs effective feature extraction considering the relationships between pixels within the input image by utilizing the self-attention mechanism and multilayer perceptron (MLP). This approach considers that each pixel in the image is not independent but is related to other pixels. In addition, DeiT [17] fine-tunes the pre-trained ViT and improves the model's generalization performance through data augmentation. This approach has proven that transformers also demonstrate excellent performance in image classification tasks. However, these structures have high computational complexity, making them challenging to use in power-limited edge devices [18, 19].

To address this limitation, various model compression techniques have been proposed. In particular, pruning [20, 21] and quantization [22, 23] successfully reduce the model size, but these compression techniques [19, 24] can degrade network performance depending on the dataset or model size. In addition, while model compression through pruning and quantization is relatively easy in traditional convolutional neural networks (CNNs), applying these conventional compression techniques to ViT is challenging. Therefore, as an alternative, this paper focuses on ViT models that are not compressed using ViT model compression techniques [25] but are designed from the start with compact architectures that are inherently more suitable for edge devices. Figure 1a and b present the accuracy according to the parameter and FLOPs size of various lightweight ViT models, respectively. This paper classifies these compact models into three categories: (1) architecture and hierarchy restructuring, (2) encoder block enhancements, and (3) integrated approaches. The first includes research that redesigns the basic architecture and hierarchical structure for convolutional vision transformer (CvT) [26] model design, focusing on reducing the size and complexity of the model. The second includes research that improves the encoder block in various ways, focusing on enhancing the computational efficiency of the model. Finally, integrated methodologies include research that applies both methodologies and proposes new models. Based on these classifications, this review paper aims to provide an in-depth understanding of the efficient implementation of ViT models on edge devices by examining the state-of-the-art (SOTA) research trends in compact ViT models and analyzing how each methodology contributes to the performance and efficiency of the model. In addition, this review paper aims to provide substantial insights into

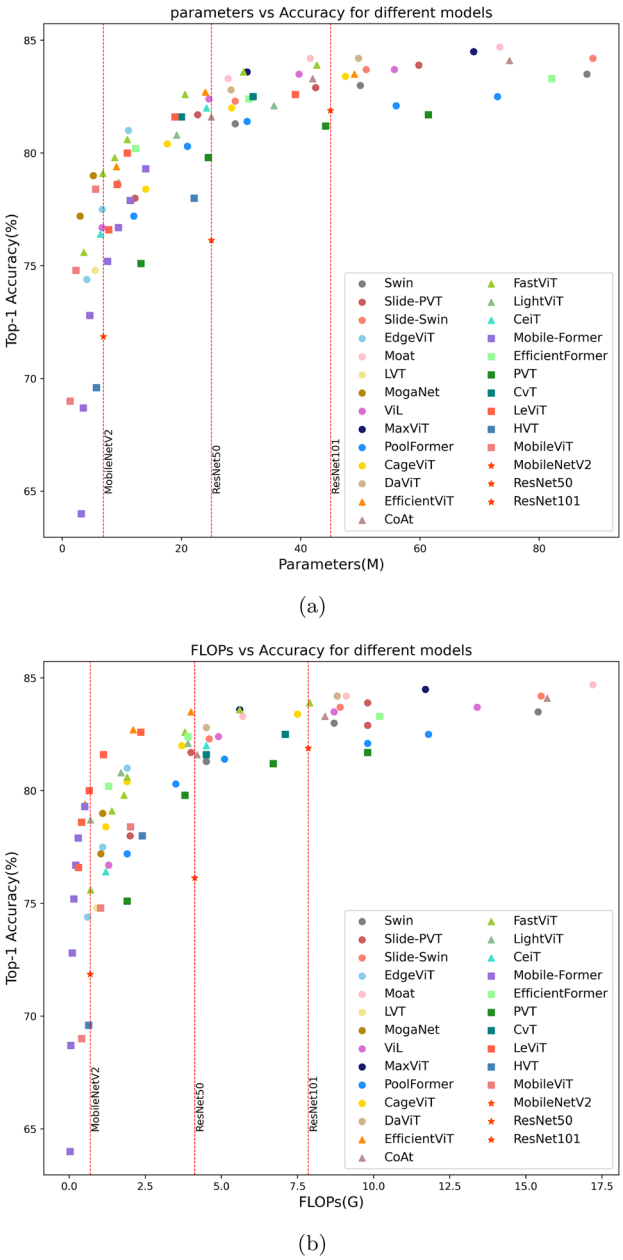


Fig. 1 Performance comparison of compact vision transformer models

how these models can be utilized on edge devices by comprehensively reviewing the SOTA research on the design and implementation of compact models. This will play a crucial role in expanding the potential for deep learning-based vision models on edge devices and maximizing the performance and efficiency of these models.

The contributions of this paper can be summarized as follows:

- Comprehensive examination of ViT models: We offer a detailed examination of the latest research in redesigning ViT models into more compact forms, making them more suitable for use on mobile and edge devices.
- Classification of compact ViT models: This research categorizes compact ViT models into three main groups: (1) architecture and hierarchy restructuring, (2) encoder block enhancements, and (3) integrated approaches.
- Detailed overview of each category: For each of these categories, this paper provides an in-depth overview, analyzing how each contributes to the model performance and computational efficiency.
- Insights and guidelines for researchers involved in this field: We present new insights into the design and implementation of compact ViT models and provide practical guidelines for improving the performance and efficiency of deep learning-based vision models on edge devices.

This paper is structured as follows: Sect. 2 provides necessary background information by discussing the general architecture and characteristics of the ViT model and introducing many compact ViT architectures. Section 3 then categorizes recent compact ViT models into three main groups and analyzes the distinct features and innovations of models within each category. Next, Sect. 4 thoroughly evaluates various models from each category using various performance metrics on image classification and other vision tasks. Finally, Sect. 5 synthesizes the current research landscape of compact ViT models based on the analysis in Sects. 3 and 4, and discusses remaining challenges and promising future research directions in designing resource-efficient yet accurate ViT models tailored for edge devices. The organization enables a comprehensive survey of the SOTA ViT architectures.

2 Background

2.1 Vision transformer

The ViT model depicted in Fig. 2 is a deep learning model architecture that has recently gained attention in the computer vision field. Unlike traditional CNNs, ViT utilizes a sequence model based on transformers to process images. It divides an image into small patches and employs them as inputs to the transformer model, enabling the training of global image information. The key components of ViT are described in the following subsections.

2.1.1 Multihead attention

The multi-head attention (MHA) illustrated in (1) of Fig. 2 is composed of multiple attention heads. Each attention head processes the input feature map from different perspectives and trains different relationships. Subsequently, the outputs of each attention head are combined to generate the final result. Each attention head consists of three main stages. First, the input feature map is transformed into a Query (Q), Key (K), and Value (V) through linear transformations, which are performed using weight matrices. Second, an attention score is calculated for each of Q, K, and V. The attention score represents the inter-dependencies between Q and K and is typically calculated using a dot product or a similarity function. Finally, the attention output is computed by applying a weighted sum to V using the attention scores. These processes are performed independently in each attention head, and the outputs of the attention heads are computed in parallel and combined to form the final result. This mechanism allows MHA to understand the global context of images by modeling the inter-dependencies within the input feature maps and identifying important relationships in the inputs.

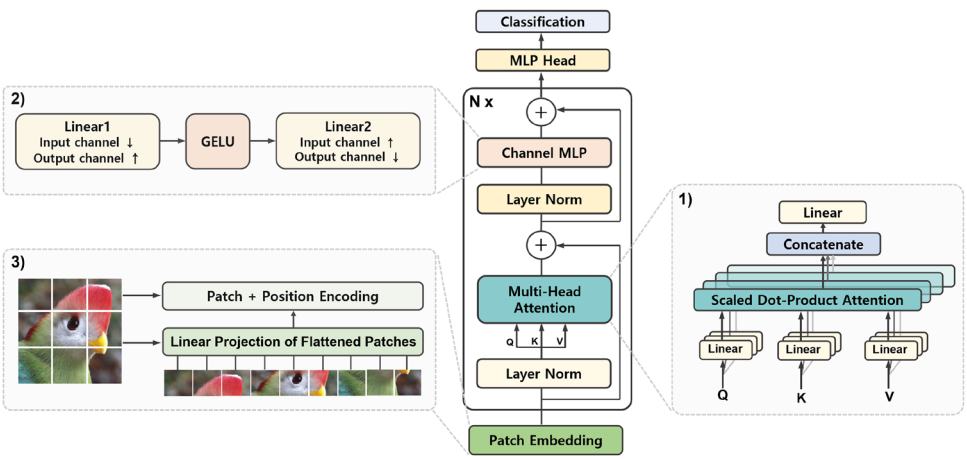


Fig. 2 Depiction of the standard ViT architecture, segmented into three key components: (1) The multi-head attention mechanism that processes input features, (2) The MLP head for classification, and (3) The patch and position encoding process, where images are divided into patches and projected for the attention mechanism

2.1.2 Multi layer perceptron

The MLP in ViT presented in (2) of Fig. 2 is a mechanism for sharing information between channels. The MLP typically consists of two linear layers and an activation function. The first linear layer expands the dimensions, and the second linear layer reduces the embeddings back to the original dimension. This allows the MLP to add nonlinearity and model relationships between channels, enabling it to train a wider range of information. Consequently, when used in conjunction with attention, the MLP demonstrates superior performance in image processing tasks.

2.1.3 Patch embedding

Patch embedding, as shown in (3) of Fig. 2, refers to the process in ViT, where the input image is divided into small patches and embedding is performed for each patch. This captures the local information (LI) of the image and provides the input to the transformer encoder. Patch embedding operates consistently regardless of the size or aspect ratio of the input image, making it capable of handling various images. Furthermore, by capturing LI and conveying it to ViT, patch embedding helps in understanding the global context. After patch embedding, ViT utilizes positional embedding to encode the positional information (PI) of each patch. This involves adding a vector representing each position in the flattened patch embeddings, thereby indicating specific locations. This allows ViT to understand and process the relative PI of the input patches.

With these three key stages in its architecture, ViT offers several advantages compared to traditional CNN-based models. First, ViT can utilize the global information (GI) of the image, enabling it to capture the overall context of objects and exhibit superior performance. Second, ViT processes images at the patch level, allowing better generalization over images of different sizes. Finally, ViT tends to achieve high performance when trained on large-scale datasets and powerful computing resources. However, although ViT models have shown superior results on various computer vision tasks, their intensive computational requirements make real-world application on mobile devices difficult. To enable the use of these powerful models on resource-limited mobile platforms, reducing their computational complexity is an important challenge that needs to be addressed.

2.2 Compact CNN models

In this subsection, we introduce compact CNN models that are widely used as backbones in compact ViT models for utilization on edge devices. Compact CNN models refer to models designed from the beginning with small model size and low computational complexity without lightweight techniques such as Kim and Kim [27, 28]. These models aim to achieve high performance even under resource-constrained environments, such as mobile/edge devices.

MobileNetV1 [2] proposes an efficient architecture called depth-wise separable convolution (CONV) for computer vision tasks. It consists of a combination of depth-wise and pointwise CONVs. The depth-wise CONV applies separate kernels to each input channel to generate outputs, training the correlation between input channels. Through this approach, MobileNetV1 reduces model size and decreases computational cost by sharing parameters. The pointwise CONV uses the 1×1 CONV operation to transform the output channels of depth-wise CONV into the desired channel size, allowing it to train high-dimensional features and extract various feature maps. MobileNetV2 [3] introduces inverted residuals and linear bottlenecks into the network structure, further optimizing the lightweightness compared to MobileNetV1. The inverted residual expands the channel size first and then reduces the model size, in contrast to the existing residual connection. Through this approach, the information stored in lower-dimensional layers, where necessary information is compressed and stored, is better preserved. Additionally, the linear bottleneck reduces the computational cost by removing non-linearity in the 1×1 CONV operation. With these structures, MobileNetV2 achieves high performance while maintaining a small model size with low computational cost. ShuffleNet [29] additionally introduces group CONV and a shuffling operation based on the structure of MobileNet. It divides input data into groups and shuffles the channels between groups to enable information exchange. Furthermore, EfficientNet [4] leverages AutoML to find the optimal combination of three factors: network depth, channel width, and input image resolution. In conclusion, these models contribute to efficient computer vision tasks on mobile devices and in resource-constrained environments.

3 Compact vision transformer models

3.1 Overview of lightweighting method for vision transformer models

The implementation of transformer models in resource-limited environments, such as mobile devices, is a significant challenge due to their high memory and computational resource demands. To address this challenge, a variety of methods have been developed to compress and accelerate transformer models, making them more feasible for efficient deployment. These methods include a variety of strategies targeting different aspects of model optimization. Among them, network pruning [18, 19] focuses on reducing the model size and complexity by eliminating redundant

parameters [30] or tokens [31], improving efficiency and reducing computing resource requirements. On the other hand, knowledge distillation [32] trains a smaller student model to mimic a larger teacher model by transferring knowledge without the bulk. Another critical approach is network quantization [33], which reduces the precision of the numerical values in the model. By representing weights and activations with fewer bits, quantization can significantly decrease the memory and computational demands with a minimal loss in performance.

This paper focuses on designing compact ViT architectures, an emerging approach that enhances efficiency by fundamentally restructuring the model to be inherently more efficient, while maintaining or even improving performance. Additionally, these compact architectures exhibit excellent compatibility with other optimization methods, making them particularly well-suited for deployment on edge devices. It should be noted that while our focus is on the architectural redesign, the other mentioned methods contribute significantly to the broader goal of creating lightweight and efficient transformer models. Each technique offers unique benefits and can be used in conjunction with others, depending on the specific requirements and constraints of the deployment environment.

3.2 A taxonomy of compact vision transformer models

In this section, we categorize compact ViT models into three main groups and introduce each core technology in the following subsections. First, research has been conducted to redesign the basic architecture and hierarchical structure belonging to "Architecture & Hierarchy" in Fig. 3 for designing compact ViT models. These studies propose new model structures, such as the pyramid architecture shown in Fig. 4. Second, research has been conducted to enhance the encoder block in various ways for compact models within the "Encoder Block" category shown in Fig. 3. This approach focuses on obtaining an excellent model in the trade-off between model performance and size by optimizing the structure and function of the encoder block as shown in Fig. 5. Finally, studies have been conducted to propose a novel architecture and encoder block, both belonging in the intersection in Fig. 3, with the purpose of identifying the optimal structure. These approaches aim to enhance model efficiency through overall structural changes in the model.

3.3 Architecture and hierarchy restructuring

3.3.1 Limitations of ViT models and introduction of new hierarchical structures

The existing ViT model [16] handles images by fixing their size and dividing them into small patches. However, this structure is not suitable for processing various sizes of features. Additionally, the non-hierarchical plain structure makes it difficult to learn low-level features and imposes constraints on detecting small objects. To overcome these challenges, there have been attempts [34–37] to explore the integration of hierarchical structures, such as residual blocks [38] and mobile bottleneck CONV(MBConv) [2, 3], to increase computational efficiency and improve low-level feature extraction.

PVT [34] is an innovative hybrid model that combines the strengths of both CNN and transformer. To address the limitations of existing ViT models, which are vulnerable in dense prediction tasks such as object detection and segmentation due to their fixed image size utilization, PVT employs

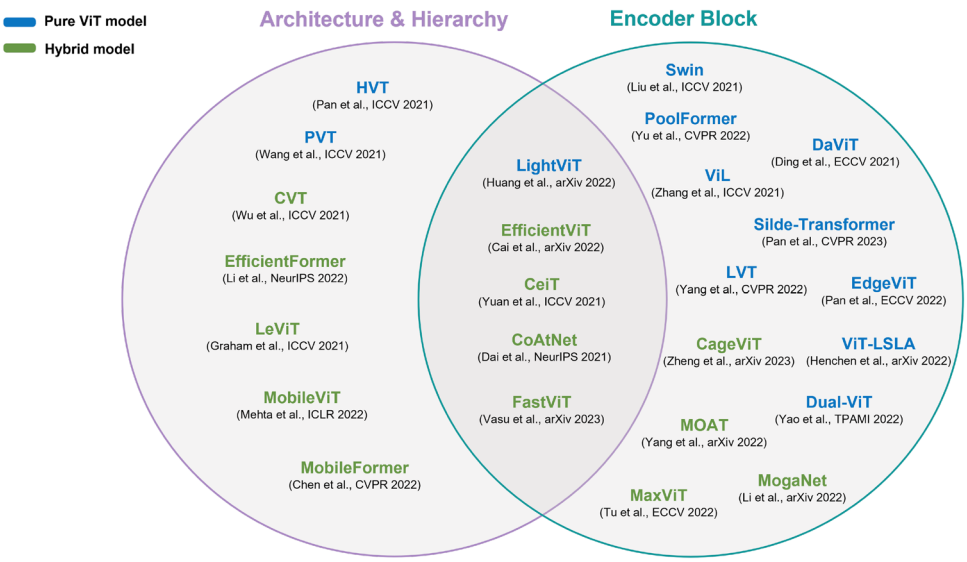


Fig. 3 Taxonomy of compact vision transformer architectures

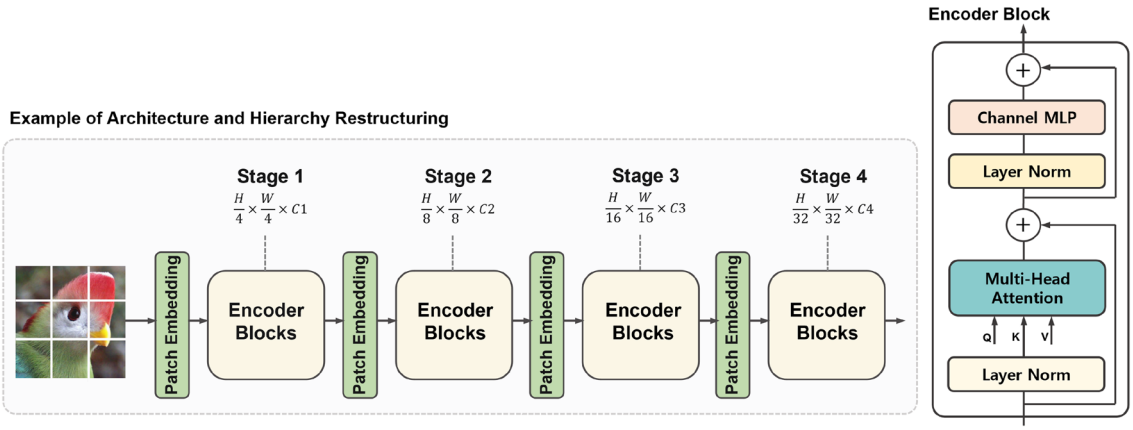


Fig. 4 Hierarchy architecture example of vision transformer model and encoder block

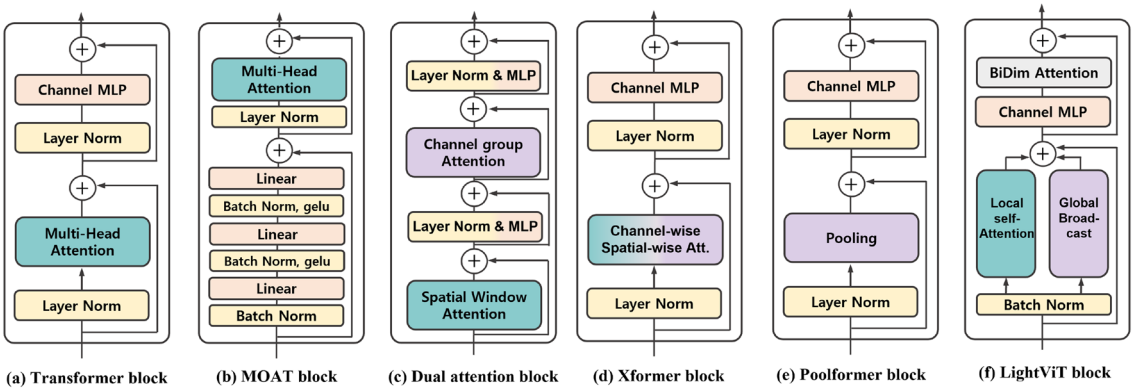


Fig. 5 Various vision transformer encoder block enhancement approaches

a pyramid structure that combines the hierarchical structure of the CNN with the transformer. This efficient architecture consists of four stages, each composed of a patch embedding layer and a transformer encoder layer. In addition, PVT employs spatial-reduction attention (SRA), instead of the conventional MHA, to effectively process high-resolution feature maps while minimizing computation and memory overhead. Therefore, PVT provides an alternative to traditional CNN backbones in tasks including object detection and semantic segmentation.

Another related approach, HVT [35], is proposed around the same time. HVT introduces a novel hierarchical approach to complement the shortcomings of ViT models, which cannot capture multilevel representations. HVT primarily achieves compression of sequential resolutions through hierarchical pooling, leading to a reduction in computational cost and enhanced model scalability. HVT demonstrates that not only single-class but also visual tokens are crucial for accurate class predictions. Consequently, HVT allows for expansion in multiple dimensions, such as depth, width, resolution, and patch size, while achieving high performance without an increase in floating point operations per second (FLOPs). MobileViT [36] proposes an efficient processing method for GI by appropriately combining the features of the transformer and CONV, leveraging the advantages of both approaches. Additionally, MobileViT demonstrates the effective application of transformers in mobile vision tasks through a simple training process; this shows the potential of ViT in the mobile environment by surpassing the performance of compact CNN models. However, in these models, the complexity of applying the hierarchical CNN structure with the transformer can pose challenges in model training and optimization.

LeViT [37] is a hybrid ViT model that combines the strengths of CNN and transformer. It introduces the hierarchical structure of LeNet to optimize the balance between accuracy and efficiency in image classification tasks. LeViT modifies the plain structure of the transformer into a hierarchical one. Instead of using class tokens, it implements average pooling at the final stage of the feature map. This

adaptation improves the trade-off between accuracy and computational efficiency. At the patch embedding stage, LeViT applies four 3×3 CONV layers, which reduces the input size of the feature map while preserving important information. Moreover, LeViT introduces attention blocks after the 1×1 CONV layers with batch normalization (BN), providing runtime advantages compared to using layer normalization. Additionally, LeViT enhances positional embedding by injecting relative PI into the attention map through attention bias. By employing these techniques, LeViT successfully achieves faster inference speed.

3.3.2 Overcoming ViT limitations in capturing local contexts

These approaches enhance the performance and compactness of ViT by introducing a hierarchical structure. However, despite these advancements, ViT still faces a limitation in capturing local contexts effectively. To mitigate this limitation, research has been conducted to incorporate the strengths of CNN in local patterns into ViT [26, 39]. This research aims to integrate CNN-based approaches with ViT to strengthen complementary capabilities and improve the ability to capture both global and local features. These studies exploit the advantages of both approaches (i.e., CNN and ViT) and further improve the performance and versatility of the ViT model.

While CNN models demonstrate superior performance in extracting local patterns, ViT does not have this ability. To address this limitation, CvT [26] combines the structure of ViT with the CONV layer. CvT adds CNN features to ViT by utilizing CONV token embeddings and projections. Through this approach, CvT can recognize LI and represent complex visual patterns. Although CvT successfully brings the LI extraction capability of CNNs to ViT, its improvements are limited because CONV layers are used only at the beginning of transformer blocks or between block connections.

To mitigate this, Mobileformer [39] configures CNN and ViT in parallel, exchanging information between them through a two-way bridge. The proposed architecture consists of four main components: mobile sub-block, former sub-block, mobile→former block, and former→mobile block. The mobile sub-block is responsible for extracting local features and adopts a structure similar to an inverted bottleneck. It takes the output from the former→mobile block as input and produces the extracted local features. The output of the mobile sub-block then serves as input to the mobile→former block, which encodes the global features. The mobile→former block acts as a two-way cross-attention block, incorporating both the local features from the mobile sub-block and global tokens from the former sub-block. This block enables the fusion of LI and GI, enhancing the model's ability to capture comprehensive features. In contrast, the former→mobile block receives global tokens from the former sub-block and combines them with the local features of the mobile sub-block. Mobile→former achieves high performance and lightweight effect through the proposed four blocks by efficiently utilizing local features and GI. This architecture is an innovative model structure that enables the fusion of LI and GI. Nevertheless, during the learning and optimization process, this two-way bridge can compromise the balance between efficiency and accuracy.

Hybrid ViT models like LeViT [37] achieve remarkable performance by leveraging the CONV-Feed Forward Network (FFN) structure. These models effectively utilize both local features and GI, resulting in high performance while maintaining a lightweight design. However, during this process, a dimension mismatch problem arises between the 4-D operations handled by CONV and 3-D operations handled by attention. This reduces network efficiency, leading to decreased training and inference speeds. To solve this problem, EfficientFormer [40] identifies inefficient operators present in existing ViT models and proposes a new design that ensures dimensional consistency throughout the model. In this process, EfficientFormer introduces the CONV attention-based MetaBlock4D to handle 4-D tensors in stages 1 and 2. By leveraging the MetaBlock4D, EfficientFormer achieves dimensional consistency and maintains efficient processing of information. Additionally, in stages 3 and 4, EfficientFormer employs linear operation-based attention mechanisms to extract GI.

3.4 Encoder block enhancements

3.4.1 Efficient design of encoder block in ViT models

Research is ongoing to efficiently design the encoder block to make ViT models more compact, considering its significant contribution to the overall operation within the ViT network [41–44]. The encoder block primarily consists of two parts: self-attention and FFN. The self-attention mechanism used in the original ViT to model dependencies between different positions in the input sequence poses a significant computational cost. Furthermore, it suffers from a quadratic increase in computation as the input resolution grows. This issue becomes a major factor contributing to the reduced efficiency of a model, especially when dealing with large input image sizes. To address the computational challenge associated with the quadratic increase in self-attention operations, ViL [41] is proposed as a solution. ViL introduces a modified attention mechanism that enables more efficient modeling of long-range dependencies in vision tasks. ViL introduces a multiscale vision longformer that reduces the memory and computational complexity required for encoding high-resolution images. Additionally, it incorporates an efficient multiscale model architecture by introducing a

specified number of global tokens to facilitate global memory operations. Moreover, ViL replaces the original 1-D positional embedding with 2-D positional embedding and incorporates relative positional biases. Consequently, the memory complexity of the multiscale self-attention block is linearly proportional rather than quadratic.

CageViT [42] challenges the limitations of existing transformer acceleration techniques, such as low-rank projection and sparsity of the attention matrix, which can result in loss of fine-grained token-level information or severe constraints on the functionality of self-attention layers. To address these limitations, CageViT leverages class activation maps provided by Grad-CAM++ [45] to identify the key tokens and incidental tokens based on their importance and feeds them into the linear projection layer. Swin Transformer [43] addresses the issue of quadratic computation increase in self-attention from a fresh perspective by limiting the receptive field. It enhances the influence of transformers in vision tasks by employing a hierarchical transformer model and calculating representations through shift windows. This design provides flexibility in modeling at various scales while maintaining linear computational complexity with respect to image size. Consequently, Swin Transformer demonstrates high performance across tasks, including image classification, object detection, and semantic segmentation. Similarly, Slide-Transformer [44] addresses the issue of quadratic increase in computational complexity in self-attention based on input resolution by proposing a novel local attention module called slide attention. This module diverges from traditional self-attention approaches by utilizing local attention to restrict the receptive field while combining the benefits of CONV layers and self-attention. Additionally, extensive experiments have demonstrated the strong compatibility of the slide attention module with various SOTA ViT models.

3.4.2 Capturing global context information in ViT models

While the local attention introduced by Swin Transformer successfully enhances the performance of ViT, it does not effectively capture global context information owing to its restricted receptive field. Consequently, various studies [46, 47] have been conducted to extract global context information. MaxViT [46] addresses the attention computation problem by decomposing attention into a sparse form and reconstructing it into two types: window and grid attention. Furthermore, it introduces a multi-axis approach that divides the overall attention size into local and global components. The MaxViT module adopts a hierarchical architecture by stacking multi-axis self-attention (Max-SA) and MBConv alternately, effectively transforming the quadratic complexity associated with input resolution into linear complexity. While approaches like ViT, Swin Transformer, and PVT deal with patch-level self-attention, they suffer from both a quadratic increase in computation and loss of global context information.

To solve these problems, DaViT [47] introduces an image-level self-attention mechanism that effectively captures GI while maintaining efficient processing in terms of spatial size. DaViT proposes an encoder module that combines spatial window self-attention and channel group self-attention, resulting in improved performance across various tasks with reduced computational cost. In conventional models like Swin Transformer and ViT, a fixed scale value is used to handle large values in dot product calculations. However, this approach has limitations in simplifying computations or capturing PI through self-attention mechanisms. To address these issues, ViT-LSLA [48] applies inner relative position bias to the existing model to better capture PI. This method focuses each Q on the relevant patches with higher relevance. The inner relative position bias refers to the relative PI between each Q and K. It enables Q to allocate more attention to K that is in proximity, allowing Q to better capture information around specific patches. As a result, the self-attention mechanism better captures the structural information of the image and allows Q to assign more attention to patches that are truly relevant, rather than just nearby ones. Lite ViT (LVT) [49] introduces a compact transformer backbone, specifically designed for mobile applications. Most existing compact ViT models, such as PVT, suffer from performance degradation when scaled down. To address this issue, LVT employs two new self-attention modules to design a lightweight yet effective transformer. LVT introduces convolutional self-attention (CSA) and recursive atrous self-attention (RASA). CSA enhances the processing of low-level features by integrating local self-attention into CONV kernels, while RASA leverages multiscale context to compute the similarity map. By utilizing CSA and RASA, LVT improves representation ability without additional cost.

3.4.3 Overcoming limitations and exploring new directions in ViT models

Prominent studies on ViT, such as Swin Transformer [43] and LeViT [37], have attempted to decompose the local and global feature extraction procedures to reduce the computational cost of the self-attention mechanism. However, such LI leads to unwanted information loss owing to downsampling, ultimately resulting in a significant decrease in accuracy rather than reducing computational complexity. By contrast, Dual-ViT [50] employs a new transformer encoder to alleviate the accuracy degradation. Dual-ViT is composed of four stages (i.e., two dual block and two merge block stages), where the resolution of the feature map is reduced at each stage. Firstly, the dual block divides the input feature map into two paths to reduce the computational cost

of self-attention. The semantic path compresses the input feature map into semantic tokens, while the pixel path utilizes these semantic tokens to capture detailed features at the pixel level of the input. The merge block enables interactions between local tokens by performing self-attention between pixel tokens and semantic tokens, allowing for the utilization of information among all tokens. Ultimately, Dual-ViT minimizes unwanted information loss and prevents accuracy degradation while reducing computational cost.

Previous research on transformer encoder blocks generally recognizes the attention-based token mixer module as a key factor for achieving high performance. However, Poolformer [51] assumes that the general architecture of the transformer model is more crucial for the model's performance than a specific token mixer module. To validate this assumption, it deliberately replaces the attention module of the transformer model with a simple pooling layer, performing only the most basic token mixing. Despite the simplicity of the token mixer, Poolformer has achieved superior performance compared to transformer/MLP baselines on various benchmarks. As a result, Poolformer presents a direction of research that emphasizes improving the MetaFormer architecture rather than focusing on the token mixer module. Despite various efforts to address the computational issues of ViT, they remain insufficient to meet the resource constraints of mobile devices.

EdgeViT [52] resolves this problem by integrating the advantages of self-attention and CONV layers while leveraging an efficient local-global-local (LGL) bottleneck phenomenon. The LGL bottleneck phenomenon acknowledges that attending to every token in downsampled feature maps is highly inefficient owing to the spatial redundancy present in images. By contrast, LGL enables the calculation of self-attention for only a subset of tokens while still allowing for overall spatial interaction, similar to the traditional MHA. Furthermore, LGL consists of local aggregation, global sparse attention, and local propagation, enabling information exchange between all token pairs within the same feature map. Through these techniques, EdgeViT achieves an optimal trade-off between accuracy and efficiency. MogaNet [53] attributes the transformer's success to its macro architecture rather than self-attention. To this end, MogaNet employs a novel network that stacks CONV layers in a macro architecture similar to self-attention. It employs spatial and channel aggregation blocks in the macro architecture, which operate by emphasizing crucial information within the input sequence. This enables the capture of diverse spatial relationships across the input sequence, ultimately leading to the development of a superior model with lower computational complexity than self-attention-based models, using only CONV layers. To optimize the combination of transformers and CONV layers, previous approaches, such as MobileViT and MogaNet, adopt a macro-level network design by individually stacking MBConv and transformer blocks to strike a balance between MBConv's efficiency and the transformer's capacity. By contrast, MOAT [54] employs a micro-level building of MBConv and transformer blocks. While the MLP module shares similarities with MBConv owing to its inverted bottleneck design, it is less efficient than MBConv. Therefore, MOAT introduces modifications by replacing the MLP layer of the transformer block with MBConv and designing the order of attention and MBConv in reverse. Consequently, MOAT presents a more efficient and less complex architecture, offering a novel approach that harnesses the strengths of both transformers and ConvNets.

3.5 Integrated approaches

To achieve further performance improvements in ViT, researchers have pursued studies that incorporate both aforementioned approaches, as seen in works such as CeiT [55] and CoaTNet [56]. CeiT presents a novel ViT architecture that combines the strengths of CNNs and transformers to overcome the limitations of each component. To enhance the generalizability of transformers, CeiT [55] incorporates an image-to-tokens (I2T) module, locally-enhanced feed-forward (LeFF) layers, and layer-wise class token attention (LCA). The I2T module enables patches from feature maps to be obtained rather than raw input images, while the LeFF layers extract LI akin to CNNs. Additionally, the LCA integrates information across multiple layers within the architecture. These improvements strengthen the generalizability and loss convergence of CeiT, surpassing previous ViT models and even SOTA CNNs in various benchmarks. Notably, CeiT demonstrates outstanding performance without requiring extensive training data or additional supervised CNN models. CoAtNets [56] is another hybrid model that combines the strengths of CNNs and transformers. It demonstrates that transformers may exhibit poorer generalization compared to CNNs owing to the absence of the correct inductive bias. To effectively integrate the advantages of both architectures, CoAtNets combines depth-wise CONVs with simple relative attention instead of traditional self-attention. Furthermore, it vertically stacks CONV and attention layers to enhance both generalization and efficiency. LightViT [57] employs a new approach to overcome the limitations of local attention introduced by Swin Transformer. While window-based attention effectively reduces computational complexity, it has limitations in extracting GI.

To address this problem, LightViT incorporates a learnable global token into the PVT [34] structure. To leverage the information from the global token, LightViT introduces the LightViT block, where the attention mechanism combines image tokens with the global token during the global aggregation process, effectively utilizing both LI and GI. Unlike traditional FFNs, which only utilize channel information, the proposed bi-dimensional FFN in LightViT can incorporate spatial information, enabling the extraction of global dependencies in a more effective manner. Meanwhile, to address the increase in computational complexity of the hybrid ViT model, FastViT [58] introduces a new token-mixing operator called RepMixer, which removes skip connections, BN, and linear activation during inference, effectively reducing the computational cost. Consequently, FastViT significantly reduces computational complexity while maintaining the same performance. EfficientViT [59] employs lightweight multiscale attention modules for on-device applications to improve the hardware inefficiency of self-attention in transformer models and large kernel CONVs. It replaces the conventional self-attention with ReLU-based attention to simplify the attention computation that previously involved exponential operations. Additionally, it proposes an aggregation process with independent small kernel CONVs for each head's Q, K, and V to obtain multiscale tokens, thereby enhancing hardware efficiency.

4 Performance analysis of various ViT models

4.1 Performance comparison of various ViT models

We compare the performance of various compact ViT models on the ImageNet dataset [60]. Models are evaluated based on image size, number of parameters (M), GFLOPs, Top-1 accuracy, and latency/throughput on GPU platforms. It should be noted that based on latency and throughput results on GPU platforms, the inference speed and throughput on the embedded platform can be easily estimated by comparing the specifications of the GPU and the specifications of each embedded platform. To facilitate a smooth comparison, we categorize the compact ViT models into four categories (i.e., tiny, small, base, and large) based on the number of parameters in representative CNN models, such as MobileNetV2, ResNet50, and ResNet101. Each category is presented in Tables 1, 2, 3, and 4, respectively. Models belonging to the 'Tiny' category have fewer parameters than MobileNetV2 (i.e., 6.9M). These are the simplest and lightest models, convenient for use in environments with limited computing resources. Models in the 'Small' category have numbers of parameters between those of MobileNetV2 and ResNet50 (i.e., 25M), while the 'Base' category includes models with parameters less than ResNet101 (i.e., 45M).

Table 1 Performance comparison of lightweight ViT-Tiny models

Type	Models	ImageSize	#Params (M)	GFLOPs	Top-1 acc (%)	Latency (ms)	Throughput (images/s)
Architecture and hierarchy	MobileViT-XS [36]	224	2.3	0.7	74.8	11.7(A100)	1581(V100)
	Mobile-Former-26M [39]	224	3.2	0.026	64.0	–	–
	Mobile-Former-52M [39]	224	3.5	0.052	68.7	–	–
	Mobile-Former-96M [39]	224	4.6	0.19	72.8	–	–
	HVT-Ti-1	224	5.74	0.64	69.6	–	–
Encoder block	MogaNet-XT [53]	256	3.0	1.04	77.2	–	–
	Slide-PVTv2-B0 [44]	256	3.3	0.6	71.4	–	–
	EdgeViT-XXS [52]	224	4.1	0.6	74.4	–	–
	MogaNet-T [53]	224	5.2	1.1	79.0	–	–
	LVT [49]	224	5.5	0.9	74.8	–	1545(V100)
	ViL-Tiny-APE [41]	224	6.7	1.3	76.3	–	–
	ViL-Tiny-RPB [41]	224	6.7	1.3	76.7	–	–
	EdgeViT-XS	224	6.7	1.1	77.5	–	–
Integrated approaches	FastViT-T8 [58]	256	3.6	0.7	75.6	–	1.7(A100)
	CeiT-T [55]	224	6.4	1.2	76.4	–	–
	FastViT-T12 [58]	256	6.8	1.4	79.1	–	2.1

Table 2 Performance comparison of lightweight ViT-Small models

Type	Models	ImageSize	#Params (M)	GFLOPs	Top-1 acc (%)	Latency (ms)	Throughput (images/s)
Architecture and hierarchy	LeViT-128S [37]	224	7.4	0.305	71.9	–	–
	Mobile-Former-151M [39]	224	7.6	0.15	75.2	–	–
	Mobile-Former-214M [39]	224	9.4	0.214	76.7	–	–
	Mobile-Former-294M [39]	224	11.4	0.294	77.9	–	–
	EfficientFormer-L1 [40]	224	12.3	1.3	80.2	6.2(A100)	–
	PVT-Tiny [34]	224	13.2	1.9	75.1	–	–
	Mobile-Former-508M [39]	224	14.0	0.508	79.3	14.6(A100)	–
	CvT-13-NAS [26]	224	18.0	4.1	71.3	–	–
	CvT-13 [26]	224	20.0	4.5	70.4	–	–
	HVT-S-1 [35]	224	22.1	2.4	78.0	–	–
	Scale HVT-Ti-4 [35]	224	22.1	1.39	75.2	–	–
	PVT-Small	224	24.5	3.8	79.8	–	–
Encoder block	EdgeViT-S [52]	224	11.1	1.9	81.0	–	–
	PoolFormer-S12 [51]	224	12.0	1.9	77.2	14.5(A100)	–
	Slide-PVT-T [34]	256	12.2	2.0	78.0	–	–
	Slide-PVTv2-B1 [34]	256	13.0	2.2	79.5	–	–
	CageViT-T [42]	224	14.0	1.2	78.4	–	1341(V100)
	CageViT-S [42]	224	17.6	1.9	80.4	–	1052(V100)
	ViT-LSLA [48]	224	18.9	3.5	–	–	–
	Slide-NAT-M [44]	256	20.0	2.7	82.4	–	–
	PoolFormer-S24 [51]	224	21.0	3.5	80.3	28.2(A100)	–
	Slide-PVT-S [44]	256	22.7	4.0	81.7	–	–
	Slide-PVTv2-B2 [44]	256	22.8	4.2	82.7	–	–
	Slide-CSwin-T [44]	256	23.0	4.3	83.2	–	–
	ViL-Small-APE [41]	224	24.6	4.9	82.0	–	–
	ViL-Small-RPB	224	24.6	4.9	82.4	–	–
Integrated approaches	FastViT-S12 [58]	256	8.8	1.8	79.8	2.2(A100)	–
	EfficientViT-B1 [59]	224	9.1	0.52	79.4	–	–
	LightViT-T [57]	224	9.4	0.7	78.7	–	2578(V100)
	FastViT-SA12 [58]	256	10.9	1.9	80.6	2.5(A100)	–
	LightViT-S [57]	224	19.2	1.7	80.8	–	1467(V100)
	FastViT-SA24 [58]	256	20.6	3.8	82.6	3.8(A100)	–
	EfficientViT-B2 [59]	256	24.0	2.1	82.7	–	–
	CeiT-S [55]	224	24.2	4.5	83.3	–	–
	CoAtNet-0 [56]	224	25.0	4.2	81.6	–	–

Finally, models in the 'Large' category have more parameters than ResNet101. The trends and patterns shown in each table provide important insights into optimizing model performance and efficiency. This analysis can suggest directions for future research and help in developing more efficient and higher-performing models.

4.2 Correlation of results

The performance of transformer models is significantly influenced by their structure and the number of parameters. The design of hierarchical structure and encoder blocks determines how the model processes and transforms input data, which

Table 3 Performance comparison of lightweight ViT-Base models

Type	Models	ImageSize	#Params (M)	GFLOPs	Top-1 acc (%)	Latency (ms)	Throughput (images/s)
Architecture and hierarchy	EfficientFormer-L3 [40]	224	31.3	3.9	82.4	13.9(A100)	–
	CvT-21 [26]	224	32.0	7.1	71.3	–	–
	PVT-Medium [34]	224	44.2	6.7	81.2	–	–
Encoder block	MOAT-0 [54]	224	27.8	5.7	83.3	–	536(V100)
	Slide-NAT-T [44]	256	28.0	4.3	83.6	–	–
	DaViT-Tiny [47]	224	28.3	4.5	82.8	–	–
	CageViT-B [42]	224	28.4	3.7	82.0	–	704(V100)
	Swin-T [43]	224	29.0	4.5	81.3	–	755(V100)
	Slide-Swin-T [44]	256	29.0	4.6	82.3	–	–
	PoolFormer-S36 [51]	224	31.0	5.1	81.4	41.2(A100)	–
	MaxViT-T [46]	224	31.0	5.6	83.6	–	350(V100)
	Slide-CSwin-S [44]	256	35.0	6.9	84.0	–	–
	ViL-Medium-APE [41]	224	39.7	8.7	83.3	–	–
	ViL-Medium-RPB [41]	224	39.7	8.7	83.5	–	–
	MOAT-1 [54]	224	41.6	9.1	84.2	–	339(V100)
	CoAtNet-1 [56]	224	42.0	8.4	83.3	–	–
	Slide-PVT-M [44]	256	42.5	9.8	82.9	–	–
	Slide-PVTv2-B3 [44]	256	42.5	7.1	83.8	–	–
Integrated approaches	FastViT-SA36 [58]	256	30.4	5.6	83.6	5.2(A100)	–
	LightViT-B [57]	224	35.5	3.9	82.1	–	827(V100)
	FastViT-MA36 [58]	256	42.7	7.9	83.9	6.7(A100)	–

directly affects the ability to recognize complex patterns and features. As a result, a well-designed hierarchical structure can achieve high performance even with a small number of parameters. To quantitatively measure this relationship, we calculate the correlation between the number of parameters and the accuracy of the model. By measuring the correlation between these two variables, we can better understand the relationship between the model's structure and performance, and gain insights for more efficient model design. The equation for measuring the correlation is as follows:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{1}$$

where x_i and y_i represent individual values for the parameters and accuracy of the model, respectively, while $\bar{x}$ and $\bar{y}$ denote the mean values within each category. Increasing the number of parameters indiscriminately can increase the complexity of the model, which leads to the risk of overfitting. On the other hand, by efficiently designing the architectures and appropriately adjusting the number of parameters, a high correlation between the number of parameters and the accuracy of the model can be maintained. This allows us to quantitatively understand how the accuracy of the model changes as the number of parameters increases. The correlation coefficient ranges between −1 and 1, where a value close to 1 signifies a strong positive correlation, while a value close to −1 represents a strong negative correlation. Furthermore, a value close to 0 indicates little correlation between the variables. The average values for the correlation coefficients of the tiny and small category models are 0.474 and 0.430, respectively, while those of the base and large categories are 0.208 and 0.472, respectively. This indicates that there is a certain correlation between the number of parameters and top-1 accuracy in all categories. Therefore, optimizing a model's performance involves not only increasing the number of parameters but also selecting and optimizing an appropriate model structure. Through this process, the model's performance improves while preventing overfitting, even as the number of parameters increases. In addition, we analyze the correlation between FLOPs and performance. In the tiny model, the correlation coefficient between these two variables is very high at 0.805. This implies that within a limited model size, complexity greatly impacts performance. However, as the model size increases, this correlation decreases. The correlation coefficients in the small, base, and large models are 0.315, 0.076, and 0.492, respectively. This indicates that as the model size

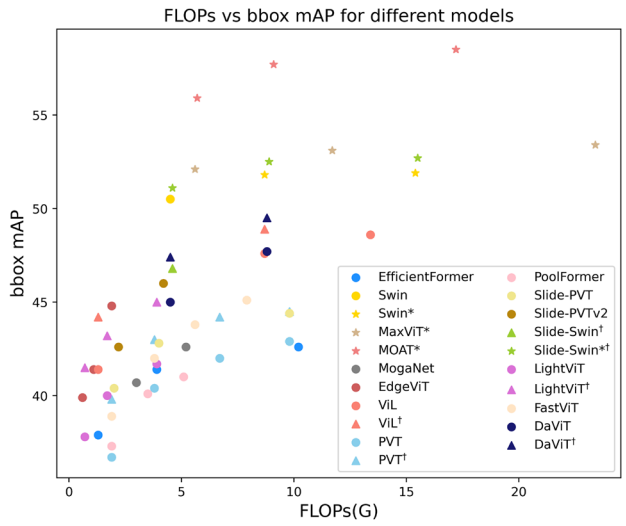
Table 4 Performance comparison of lightweight ViT-Large models

Type	Models	ImageSize	#Params (M)	GFLOPs	Top-1 acc (%)	Latency (ms)	Throughput (images/s)
Architecture and hierarchy	PVT-Large [34]	224	61.4	9.8	81.7	–	–
	EfficientFormer-L7 [40]	224	82.1	10.2	83.3	30.7(A100)	–
Encoder block	CageViT-L [42]	224	47.5	7.5	83.4	–	481(V100)
	DaViT-Small [47]	224	49.7	8.8	84.2	–	–
	Swin-S [43]	224	50.0	8.7	83.0	–	437(V100)
	Slide-Swin-S [44]	256	51.0	8.9	83.7	–	–
	Slide-NAT-S [44]	256	51.0	7.8	84.3	–	–
	ViL-Base-APE [41]	224	55.7	13.4	83.2	–	–
	ViL-Base-RPB [41]	224	55.7	13.4	83.7	–	–
	PoolFormer-M36 [51]	224	56.0	9.8	82.1	–	–
	Slide-PVT-L [44]	256	59.8	9.8	83.9	–	–
	Slide-PVTv2-B4 [44]	256	59.8	10.3	84.2	–	–
	MaxViT-S [46]	224	69.0	11.7	84.5	–	243(V100)
	PoolFormer-M48 [51]	224	73.0	11.8	82.5	–	–
	MOAT-2 [54]	224	73.4	17.2	84.7	–	209(V100)
	CoAtNet-2 [56]	224	75.0	15.7	84.1	–	–
	Slide-CSwin-B [44]	256	78.0	15.0	84.7	–	–
	Slide-PVTv2-B5 [44]	256	78.9	12.1	84.3	–	–
	Swin-B [43]	224	88.0	15.4	83.5	–	278(V100) / 5325(A100)
	Slide-Swin-B [44]	256	89.0	15.5	84.2	–	–
	MaxViT-B [46]	224	120.0	23.4	84.9	–	134(V100)
	MaxViT-L [46]	224	212.0	43.9	85.2	–	99(V100)
Integrated	EfficientViT-B3 [59]	224.0	49.0	4.0	83.5	–	3797(A100)

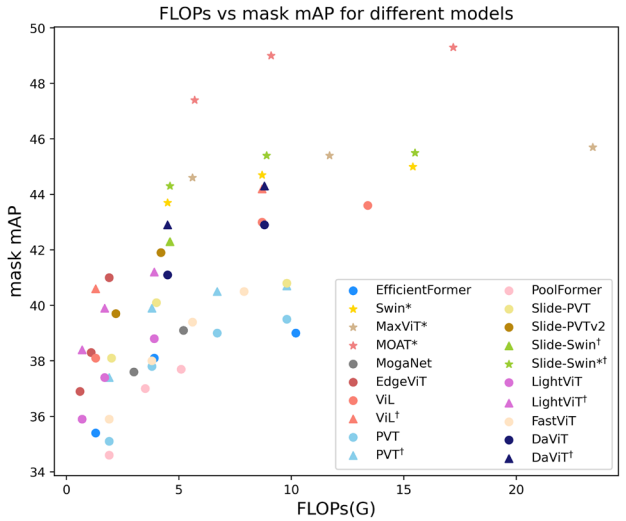
increases, it becomes challenging to improve performance solely through increasing complexity, and other factors such as the model's architecture and hierarchy gain increasing importance. These results prove that to optimize a model's performance, it is more critical to appropriately design the model's architecture and hierarchical structure, rather than merely increasing the number of parameters. In other words, it is possible to achieve high performance with fewer parameters by efficiently utilizing the model's hierarchy or encoder blocks. This plays an important role in maintaining a high correlation between the number of parameters and the accuracy of the model while optimizing the model's performance.

4.3 Analysis according to taxonomy

As can be seen from Tables 1, 2, 3 and 4, architecture and hierarchy-type models are mainly concentrated in the 'Tiny' and 'Small' categories, while the encoder block-type dominated in the 'Base' and 'Large' categories. This implies that selecting optimized lightweight methods based on model size (i.e., parameters) is crucial for enhancing the performance and efficiency of compact ViT models. In other words, in resource-constrained environments, such as edge devices, it is advantageous to design the architecture and hierarchy-type models effectively to make the model compact. For example, as shown in Table 2, the LeViT-128S and Mobile-Former-151M models have similar numbers of parameters but differed in accuracy by 3.3%. This clearly demonstrates that the architecture and hierarchy of the model have a significant impact on performance. Meanwhile, to maximize accuracy within a given model size, it is necessary to design the encoder block appropriately. Therefore, it is important to choose the best architecture and design strategies according to the requirements of specific tasks and deployment environments. This can reduce model size and complexity while maintaining or improving accuracy. For example, looking at ViL-Tiny-APE and ViL-Tiny-RPB models in Table 1, which have the same number of parameters and FLOPs but different encoder block structures, Top-1 accuracy differs by approximately 0.4%. Similarly, as shown in Table 4, ViL-Base-APE and ViL-Base-RPB have the same numbers of parameters and FLOPs, but the Top-1 accuracy



(a)



(b)

Fig. 6 bbox and mask mAP according to FLOPs of various ViT models. Mask R-CNN is used as the base model, where * indicates cascaded R-CNN and † denotes models trained with 3× longer schedule

differs by approximately 0.5%. This indicates that even with similar size and complexity, the design of the encoder block can affect the model's performance.

4.4 Detection and segmentation results

In Fig. 6, we evaluate the models on the COCO dataset [61] and depict bbox and mask mAP according to FLOPs. Mask R-CNN [62] is used as the base model, where * indicates cascaded R-CNN and † denotes models trained with 3x longer schedule. These compact ViT models demonstrate remarkable capability in both detection and segmentation, achieving competitive AP. This performance particularly emphasizes the potential of compact ViTs in efficiently balancing model size and computational demands without significantly compromising task-specific effectiveness. The results encourage further exploration into advanced model compression techniques and the integration of hybrid architectures for enhancing real-time application capabilities on mobile and edge devices. Detailed accuracy results according to ViT model size are presented in Tables 5, 6, 7, and 8.

5 Conclusions

While the ViT initially achieved significant success in the field of computer vision, it has the disadvantages of high model size and computational cost, making its use in general mobile/edge environments a challenge. Therefore, various compact ViT models are being researched to overcome this limitation. In this paper, we classified recent research related to compact ViT models into three categories: architecture and hierarchy restructuring, encoder block enhancements, and integrated approaches, and analyzed the development trends of the studies in each category. We believe that this paper will significantly contribute to improving the performance of compact ViT models and exploring their practical applicability.

Table 5 Performance comparison of ViT-Tiny models for object detection and instance segmentation on the COCO dataset

Type	Models	Schedule	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Encoder block	MogaNet-XT [53]	1×	40.7	62.3	44.4	37.6	59.6	40.2
	EdgeViT-XXS [52]	1×	39.9	62.0	43.1	36.9	59.0	39.4
	MogaNet-T [53]	1×	42.6	64.0	46.4	39.1	61.3	42.0
	ViL-Tiny-RPB [41]	1×	41.4	63.5	45.0	38.1	60.3	40.8
	ViL-Tiny-RPB [41]	3×	44.2	66.4	48.2	40.6	63.2	44.0
	EdgeViT-XS [52]	1×	41.4	63.7	45.0	38.3	60.9	41.3

Table 6 Performance comparison of ViT-Small models for object detection and instance segmentation on the COCO dataset

Type	Models	Schedule	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Architecture and hierarchy	EfficientFormer-L1 [40]	–	37.9	60.3	41.0	35.4	57.3	37.3
	PVT-Tiny [34]	1×	36.7	59.2	39.3	35.1	56.7	37.3
	PVT-Tiny [34]	3×	39.8	62.2	43.0	37.4	59.3	39.9
	PVT-Small [34]	1×	40.4	62.9	43.8	37.8	60.1	40.3
	PVT-Small [34]	3×	43.0	65.3	46.9	39.9	62.5	42.8
Encoder block	EdgeViT-S	1×	44.8	67.4	48.9	41.0	64.2	43.8
	PoolFormer-S12	1×	37.3	59.0	40.1	34.6	55.8	36.9
	Slide-PVT-T [44]	1×	40.4	63.4	43.8	38.1	60.4	41.0
	Slide-PVTv2-B1 [44]	1×	42.6	65.3	46.8	39.7	62.6	42.6
	PoolFormer-S24	1×	40.1	62.2	43.4	37.0	59.1	39.6
	Slide-PVT-S [44]	1×	42.8	65.9	46.7	40.1	63.1	43.1
	Slide-PVTv2-B2 [44]	1×	46.0	68.2	50.3	41.9	65.1	45.4
Integrated approaches	LightViT-T [57]	1×	37.8	60.7	40.4	35.9	57.8	38.0
	LightViT-T [57]	3×	41.5	64.4	45.1	38.4	61.2	40.8
	FastViT-SA12 [58]	1×	38.9	60.5	42.2	35.9	57.6	38.1
	LightViT-S [57]	1×	40.0	62.9	42.6	37.4	60.0	39.3
	LightViT-S [57]	3×	43.2	66.0	47.4	39.9	63.0	42.7
	FastViT-SA24 [58]	1×	42.0	63.5	45.8	38.0	60.5	40.5

Table 7 Performance comparison of ViT-Base models for object detection and instance segmentation on the COCO dataset

Type	Models	Schedule	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Architecture and hierarchy	EfficientFormer-L3 [40]	–	41.4	63.9	44.7	38.1	61.0	40.4
	PVT-Medium [34]	1×	42.0	64.4	45.6	39.0	61.6	42.1
	PVT-Medium [34]	3×	44.2	66.0	48.2	40.5	63.1	43.5
Encoder block	MOAT-0* [54]	–	55.9	73.9	60.9	47.4	70.9	52.1
	DaViT-Tiny [47]	1×	45.0	–	–	41.1	–	–
	DaViT-Tiny [47]	3×	47.4	69.5	52.0	42.9	66.8	46.4
	Swin-T [43]	–	50.5	69.3	54.9	–	–	–
	Swin-T* [43]	–	50.5	69.3	54.9	43.7	66.6	47.1
	Slide-Swin-T [44]	3×	46.8	69.0	51.6	42.3	66.0	45.8
	Slide-Swin-T* [44]	3×	51.1	69.8	55.4	44.3	67.4	48.0
	PoolFormer-S36	1×	41.0	43.1	44.8	37.7	60.1	40.0
	MaxViT-T* [46]	–	52.1	71.9	56.8	44.6	69.1	48.4
	ViL-Medium-RPB [41]	1×	47.6	69.8	52.1	43.0	66.9	46.6
	ViL-Medium-RPB [41]	3×	48.9	70.3	54.0	44.2	67.9	47.7
	MOAT-1* [54]	–	57.7	76.0	63.4	49.0	73.4	53.2
	Slide-PVT-M [44]	1×	44.4	66.9	48.6	40.8	63.9	43.8
Integrated approaches	FastViT-SA36 [58]	1×	43.8	65.1	47.9	39.4	62.0	42.3
	LightViT-B [57]	1×	41.7	64.5	45.1	38.8	61.4	41.4
	LightViT-B [57]	3×	45.0	67.9	48.8	41.2	64.8	44.2
	FastViT-MA36 [58]	1×	45.1	66.8	49.5	40.5	63.8	43.4

*Denotes cascaded R-CNN based model

Looking ahead, the future direction for compact ViT centers on enhancing efficiency and adaptability for real-world applications. As demand grows for resource-efficient yet powerful models, particularly in mobile and edge computing, the emphasis will likely be on refining ViT architectures for an optimal balance between performance and

Table 8 Performance comparison of ViT-Large models for object detection and instance segmentation on the COCO dataset

Type	Models	Schedule	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Architecture and hierarchy	PVT-Large [34]	1×	42.9	65.0	46.6	39.5	61.9	42.5
	PVT-Large [34]	3×	44.5	66.0	48.3	40.7	63.4	43.7
	EfficientFormer-L7 [40]	–	42.6	65.1	46.1	39.0	62.2	41.7
Encoder block	DaViT-Small [47]	1×	47.7	–	–	42.9	–	–
	DaViT-Small [47]	3×	49.5	71.4	54.7	44.3	68.4	47.6
	Swin-S* [43]	–	51.8	70.4	56.3	44.7	67.9	48.5
	Slide-Swin-S* [44]	3×	52.5	71.3	57.2	45.4	68.9	49.6
	ViL-Base-RPB [41]	1×	48.6	70.5	53.4	43.6	67.6	47.1
	MaxViT-S* [46]	–	53.1	72.5	58.1	45.4	69.8	49.5
	MOAT-2* [54]	–	58.5	76.6	64.3	49.3	73.9	53.9
	Swin-B* [43]	–	51.9	70.9	56.5	45.0	68.4	48.7
	Slide-Swin-B* [44]	3×	52.7	71.2	57.2	45.5	68.8	49.6
	MaxViT-B* [46]	–	53.4	72.9	58.1	45.7	70.3	50.0

*Denotes cascaded R-CNN based model

computational cost. This entails technical advancements in model design and training techniques, as well as a focus on application-specific optimizations to meet diverse environmental challenges. The continued evolution in this area promises significant contributions to both the field of computer vision and the practical deployment of AI technologies in resource-constrained settings.

Acknowledgements This work was partly supported by the National R &D Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2022M3I7A1078936) and Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2019R1A6A1A03032119).

Author Contributions S. Lee and K. Koo collaborated on the writing and review of the paper. They jointly authored the figures and text for the section titled 'Introduction', 'Background', and 'Performance Analysis of Various ViT Models'. All authors collaborated on the investigation of the compact ViT models. S. O was primarily responsible for writing for the section titled 'Architecture and Hierarchy Restructuring.' J. Lee. led the writing for the section on 'Encoder Block Enhancements.' S.J. authored the text for the 'Integrated Approaches' section of the paper. G. Lee compiled and summarized the experimental results for the paper and was also responsible for creating and formatting all tables. H. Kim supervised and reviewed the entire process. All authors reviewed, edited, and approved the final manuscript. This collective effort ensured a comprehensive and well-rounded analysis of the compact ViT models, contributing to the overall quality and depth of the review paper.

Data availability Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Choi, J., Chun, D., Kim, H., Lee, H.-J.: Gaussian yolov3: an accurate and fast object detector using localization uncertainty for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 502–511 (2019)
2. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: efficient convolutional neural networks for mobile vision applications (2017). arXiv preprint. arXiv:1704.04861
3. 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, pp. 4510–4520 (2018)
4. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning, pp. 6105–6114. PMLR (2019)
5. Lee, S.I., Kim, H.: GaussianMask: uncertainty-aware instance segmentation based on gaussian modeling. In: 2022 26th International Conference on Pattern Recognition (ICPR), pp. 3851–3857. IEEE (2022)
6. Vishwakarma, D.K., Singh, T.: A visual cognizance based multi-resolution descriptor for human action recognition using key pose. AEU Int. J. Electron. Commun. **107**, 157–169 (2019)
7. Singh, T., Vishwakarma, D.K.: Video benchmarks of human action datasets: a review. Artif. Intell. Rev. **52**, 1107–1154 (2019)
8. Singh, T., Vishwakarma, D.K.: A deeply coupled ConvNet for human activity recognition using dynamic and RGB images. Neural Comput. Appl. **33**, 469–485 (2021)
9. Dhiman, C., Vishwakarma, D.K.: View-invariant deep architecture for human action recognition using two-stream motion and shape temporal dynamics. IEEE Trans. Image Process. **29**, 3835–3844 (2020)
10. Chun, D., Choi, J., Lee, H.-J., Kim, H.: CP-CNN: computational parallelization of CNN-based object detectors in heterogeneous embedded systems for autonomous driving. IEEE Access **11**, 52812–52823 (2023)
11. Lee, J., Jang, J., Lee, J., Chun, D., Kim, H.: CNN-based mask-pose fusion for detecting specific persons on heterogeneous embedded systems. IEEE Access **9**, 120358–120366 (2021)

12. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30** (2017)
13. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training. Inpreprint (2018)
14. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding (2018). arXiv preprint. arXiv:1810.04805
15. Sak, H., Senior, A.W., Beaufays, F.: Long short-term memory recurrent neural network architectures for large scale acoustic modeling. Google (2014)
16. 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 (2020). arXiv preprint. arXiv:2010.11929
17. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning, pp. 10347–10357. PMLR (2021)
18. Yu, F., Huang, K., Wang, M., Cheng, Y., Chu, W., Cui, L.: Width & depth pruning for vision transformers. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 3143–3151 (2022)
19. Chen, T., Cheng, Y., Gan, Z., Yuan, L., Zhang, L., Wang, Z.: Chasing sparsity in vision transformers: an end-to-end exploration. Adv. Neural Inf. Process. Syst. **34**, 19974–19988 (2021)
20. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.-J.: DynamicViT: efficient vision transformers with dynamic token sparsification. Adv. Neural Inf. Process. Syst. **34**, 13937–13949 (2021)
21. Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., Xie, P.: Not all patches are what you need: expediting vision transformers via token reorganizations (2022). arXiv preprint. arXiv:2202.07800
22. Liu, Z., Wang, Y., Han, K., Zhang, W., Ma, S., Gao, W.: Post-training quantization for vision transformer. Adv. Neural Inf. Process. Syst. **34**, 28092–28103 (2021)
23. Yao, Z., Yazdani Aminabadi, R., Zhang, M., Wu, X., Li, C., He, Y.: ZeroQuant: efficient and affordable post-training quantization for large-scale transformers. Adv. Neural Inf. Process. Syst. **35**, 27168–27183 (2022)
24. Tang, Y., Han, K., Wang, Y., Xu, C., Guo, J., Xu, C., Tao, D.: Patch slimming for efficient vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12165–12174 (2022)
25. Lee, J.H., Kim, H.: Discrete cosine transformed images are easy to recognize in vision transformers. IEIE Trans. Smart Process. Comput. **12**(1), 48–54 (2023)
26. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: CvT: introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 22–31 (2021)
27. Kim, N.J., Kim, H.: FP-AGL: filter pruning with adaptive gradient learning for accelerating deep convolutional neural networks. IEEE Trans. Multimed. **25**, 5279–5290 (2023)
28. Kim, S., Kim, H.: Zero-centered fixed-point quantization with iterative retraining for deep convolutional neural network-based object detectors. IEEE Access **9**, 20828–20839 (2021)
29. Zhang, X., Zhou, X., Lin, M., Sun, J.: ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6848–6856 (2018)
30. Chuanyang, Z., Li, Z., Zhang, K., Yang, Z., Tan, W., Xiao, J., Ren, Y., Pu, S.: SAViT: structure-aware vision transformer pruning via collaborative optimization. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022). https://openreview.net/forum?id=w5DacXWzQ-Q
31. Liu, Y., Gehrig, M., Messikommer, N., Cannici, M., Scaramuzza, D.: Revisiting token pruning for object detection and instance segmentation (2023). arXiv preprint. arXiv:2306.07050
32. Wu, K., Zhang, J., Peng, H., Liu, M., Xiao, B., Fu, J., Yuan, L.: TinyViT: fast pretraining distillation for small vision transformers. In: European Conference on Computer Vision, pp. 68–85. Springer, Berlin (2022)
33. Lin, Y., Zhang, T., Sun, P., Li, Z., Zhou, S.: FQ-ViT: post-training quantization for fully quantized vision transformer (2021). arXiv preprint. arXiv:2111.13824
34. Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 568–578 (2021)
35. Pan, Z., Zhuang, B., Liu, J., He, H., Cai, J.: Scalable vision transformers with hierarchical pooling. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 377–386 (2021)
36. Mehta, S., Rastegari, M.: MobileViT: light-weight, general-purpose, and mobile-friendly vision transformer (2021). arXiv preprint. arXiv:2110.02178
37. 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. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12259–12269 (2021)
38. 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, pp. 770–778 (2016)
39. Chen, Y., Dai, X., Chen, D., Liu, M., Dong, X., Yuan, L., Liu, Z.: Mobile-former: bridging MobileNet and transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5270–5279 (2022)
40. Li, Y., Yuan, G., Wen, Y., Hu, J., Evangelidis, G., Tulyakov, S., Wang, Y., Ren, J.: EfficientFormer: vision transformers at MobileNet speed. Adv. Neural Inf. Process. Syst. **35**, 12934–12949 (2022)
41. Zhang, P., Dai, X., Yang, J., Xiao, B., Yuan, L., Zhang, L., Gao, J.: Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2998–3008 (2021)
42. Zheng, H., Wang, J., Zhen, X., Chen, H., Song, J., Zheng, F.: CageViT: convolutional activation guided efficient vision transformer (2023). arXiv preprint. arXiv:2305.09924
43. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021)
44. Pan, X., Ye, T., Xia, Z., Song, S., Huang, G.: Slide-transformer: hierarchical vision transformer with local self-attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2082–2091 (2023)
45. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-CAM++: generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 839–847. IEEE (2018)
46. Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., Li, Y.: MaxViT: multi-axis vision transformer. In: Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October

23–27, 2022, Proceedings, Part XXIV, pp. 459–479. Springer, Berlin (2022)
47. Ding, M., Xiao, B., Codella, N., Luo, P., Wang, J., Yuan, L.: DaViT: dual attention vision transformers. In: Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV, pp. 74–92. Springer, Berlin (2022)
48. Hechen, Z., Huang, W., Zhao, Y.: ViT-LSLA: vision transformer with light self-limited-attention (2022). arXiv preprint. arXiv:2210.17115
49. Yang, C., Wang, Y., Zhang, J., Zhang, H., Wei, Z., Lin, Z., Yuille, A.: Lite vision transformer with enhanced self-attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11998–12008 (2022)
50. Yao, T., Li, Y., Pan, Y., Wang, Y., Zhang, X.-P., Mei, T.: Dual vision transformer. IEEE Trans. Pattern Anal. Mach. Intell. **45**(9), 10870–10882 (2023)
51. Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., Yan, S.: MetaFormer is actually what you need for vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10819–10829 (2022)
52. Pan, J., Bulat, A., Tan, F., Zhu, X., Dudziak, L., Li, H., Tzimiropoulos, G., Martinez, B.: EdgeViTs: competing light-weight CNNs on mobile devices with vision transformers. In: Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI, pp. 294–311. Springer, Berlin (2022)
53. Li, S., Wang, Z., Liu, Z., Tan, C., Lin, H., Wu, D., Chen, Z., Zheng, J., Li, S.Z.: Efficient multi-order gated aggregation network (2022). arXiv preprint. arXiv:2211.03295
54. Yang, C., Qiao, S., Yu, Q., Yuan, X., Zhu, Y., Yuille, A., Adam, H., Chen, L.-C.: MOAT: alternating mobile convolution and attention brings strong vision models (2022). arXiv preprint. arXiv:2210.01820
55. Yuan, K., Guo, S., Liu, Z., Zhou, A., Yu, F., Wu, W.: Incorporating convolution designs into visual transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 579–588 (2021)
56. Dai, Z., Liu, H., Le, Q.V., Tan, M.: CoAtNet: marrying convolution and attention for all data sizes. Adv. Neural Inf. Process. Syst. **34**, 3965–3977 (2021)
57. Huang, T., Huang, L., You, S., Wang, F., Qian, C., Xu, C.: LightViT: towards light-weight convolution-free vision transformers (2022). arXiv preprint. arXiv:2207.05557
58. Vasu, P.K.A., Gabriel, J., Zhu, J., Tuzel, O., Ranjan, A.: FastViT: a fast hybrid vision transformer using structural reparameterization (2023). arXiv preprint. arXiv:2303.14189
59. Cai, H., Gan, C., Han, S.: EfficientViT: enhanced linear attention for high-resolution low-computation visual recognition (2022). arXiv preprint. arXiv:2205.14756
60. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
61. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13, pp. 740–755. Springer, Berlin (2014)
62. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.