School of CIT, TUM, Munich, August 2024

# A Review on Vision Transformers: Limitations and Advancements

**Oscar Breiner (oscar.breiner@tum.de), Technical University Munich**

## Abstract

Vision Transformers have revolutionized computer vision by applying the self-attention mechanism, initially successful in natural language processing (NLP), to image processing. Despite their potential, this transition encounters challenges such as decreasing the computational complexity of self-attention while maintaining global dependency modeling. Various strategies have emerged to mitigate these limitations, driving ongoing advancements towards unified models capable of handling both vision and language tasks.

## 1. Introduction

Transformers, driven by self-attention [1], have emerged as the standard for NLP while Convolutional Neural Networks (CNNs) have historically dominated computer vision. Examples include the evolution of CNN architectures from LeNet-5 [2] and AlexNet [3] to more recent advancements such as ResNet [4] and EfficientNet [5]. Inspired by the scalability of Transformers in NLP, researchers have explored their potential in vision tasks, aiming to either replace or complement convolutional networks [6]. However, this transition has faced many challenges, prompting the development of various solution strategies that can surpass traditional CNNs.

This paper provides an introduction to the attention-mechanism in Section 2 and what modifications have to be done to apply a transformer architecture to vision tasks in Section 3. Furthermore, Section 4 reviews state-of-the-art advancements in vision transformers, emphasizing the contributions of the Swin Transformer and FasterViT in addressing challenges within the field.

## 2. Background

Self-attention is an important mechanism in transformer architectures, enabling the model to weigh the importance of different elements in an input sequence. This mechanism is particularly useful in language tasks for extracting contextual information.

The first step in this process involves mapping word tokens into fixed-size, context-independent embedding vectors. Common pre-trained embeddings include Word2Vec [7] and FastText [8].

$$\mathbf{X} = \text{Embed}(\text{tokens}) \tag{1}$$

Since the attention mechanism of transformer does not contain any recurrence or convolution, fixed Positional Encodings are usually added to the input embeddings $\mathbf{X}' = \mathbf{X} + \mathbf{PE}$ to compensate the lack of positional information:

$$\mathbf{PE}(pos, i) = \begin{cases} \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) & \text{if } i \text{ even} \\ \cos\left(\frac{pos}{10000^{2(i-1)/d_{model}}}\right) & \text{if } i \text{ odd} \end{cases} \tag{2}$$

Here, $pos$ is the token position, $i$ is the ith embedding dimension, and $d_{model}$ is the model's dimension size [1].

To determine the relevance of input tokens to each other, the attention mechanism derives three vector embeddings: Query ($\mathbf{Q}$), Key ($\mathbf{K}$), and Value ($\mathbf{V}$) using weight matrices $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ learned during backpropagation. Afterwards, the Scaled Dot-Product calculates the attention scores for evaluating the importance of each token in a given context [1]:

$$\mathbf{Q} = \mathbf{X}'\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}'\mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}'\mathbf{W}_V \tag{3}$$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{4}$$

Here, $\mathbf{Q}\mathbf{K}^T$ computes the similarity scores, scaled by $\sqrt{d_k}$ to manage the effect of large values, and softmax normalizes these scores. Figure 1 illustrates nicely, how this kind of global feature extraction differs from the local feature extraction of CNN.

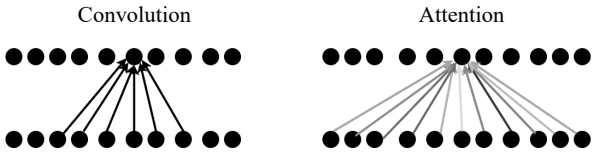


*Figure 1.* Convolution vs. Attention

Transformer encoder utilize this self-attention mechanism across multiple layers in combination with feed-forward neural networks, and residual connections with layer normalization [1]. Depending on the task, the output of the encoder can be used for classification, regression, or feed into a decoder for sequence-to-sequence tasks.

## 3. ViT - Vision Transformer

The ViT Vision Transformer applies a standard Transformer architecture with minimal modifications directly to image data, challenging the traditional reliance on CNNs in image classification [6]. These modifications involve transforming the input image into embeddings that the self-attention mechanism can process to capture spatial information.

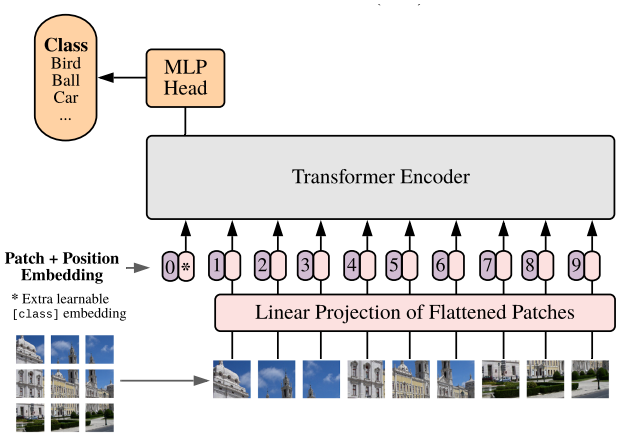


*Figure 2.* ViT Design Overview [6]

The following steps and Figure 2 give a detailed explanation of the ViT architecture:

1. **Image Splitting**: The input image $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ is split into non-overlapping patches of size $P \times P$, where $H$ and $W$ are the height and width of the image, and $C$ is the number of channels.

2. **Flattening**: Each patch is flattened into a 1D vector $x_p \in \mathbb{R}^{P^2 \cdot C}$.

3. **Linear Projection**: The flattened patches are then passed through a linear layer (dense layer) and mapped into a lower-dimensional space $\mathbf{E}_p \in \mathbb{R}^D$:

$$\mathbf{E}_p = \mathbf{W}_p \mathbf{x}_p + \mathbf{b}_p \tag{5}$$

   where $\mathbf{W}_p \in \mathbb{R}^{D \times (P^2 \cdot C)}$ and $\mathbf{b}_p \in \mathbb{R}^D$ are the learnable parameters of the linear layer.

4. **Positional Encoding**: To retain the spatial information, learnable positional embeddings $\mathbf{P}_e \in \mathbb{R}^{N \times D}$ (where $N$ is the number of patches) are added to the patch embeddings. Each positional embedding vector encodes the position of its corresponding patch in the original image. Unlike in NLP, where fixed positional encodings are used (see Section 2), these embeddings are randomly initialized and optimized during training via backpropagation, similar to other model parameters:

$$\mathbf{Z} = \mathbf{E} + \mathbf{P}_e \tag{6}$$

   where:

   - $\mathbf{E} \in \mathbb{R}^{N \times D}$ is the matrix of patch embeddings.
   - $\mathbf{P}_e \in \mathbb{R}^{N \times D}$ is the positional embedding matrix.
   - $\mathbf{Z} \in \mathbb{R}^{N \times D}$ is the final input to the encoder.

The main difference between ViT and NLP Transformers lies in how embeddings are calculated. ViT uses learnable positional embeddings, which adaptively capture the spatial relationships within image data. This is advantageous over fixed sinusoidal functions in NLP, as it allows the model to prioritize key features, like the arrangement of eyes, nose, and mouth in facial recognition, enhancing performance in image classification and object detection.

### Limitations and Problems

**Isotropic Architecture**: Maintaining the same feature resolution across all transformer layers limits the model's ability to capture hierarchical, multi-scale features needed for tasks like detection and segmentation [9].

**Quadratic Scaling Problem**: Self-attention requires computing attention scores for every pair of patches, leading to a quadratic complexity. This is a problem in vision tasks such as segmentation and detection which require high-resolution images with many patches [10, 9].

**No Inductive Bias**: ViT underperforms on mid-sized datasets compared to ResNet and EfficientNet of similar size, due to the absence of an inductive bias inherent in CNNs. This bias refers to built-in assumptions about the data, such as spatial hierarchies and locality, which helps CNN to learn more effectively from smaller datasets [6].

### Strengths and Advantages

**Pre-Training**: ViT models can match or surpass state-of-the-art CNNs, when pre-trained on large datasets. This suggests that large-scale training offsets the lack of inductive bias [6]. Furthermore, as shown in Table 1, when both model types are pre-trained on datasets with up to 300M entries, ViT not only achieves higher accuracy but also requires fewer computational resources during training.

**Global Context**: The self-attention mechanism in ViT captures global dependencies and relationships within the image. This can be beneficial for tasks requiring a comprehensive understanding of the entire image. Without built-in inductive biases, ViT can learn directly from data, potentially leading to better generalization across different tasks and datasets once sufficiently trained.

**Scalability**: ViT models are highly scalable, benefiting from advances in Transformer architectures. They can easily scale up with increased computational resources and larger datasets, making them adaptable to a wide range of tasks and data sizes [6].

## 4. Review Advancements after ViT

Building on the success of ViT, several new architectures have emerged to address its limitations. In the following section, we will discuss the core ideas of Swin Transformer and FasterViT, focusing on their methodological innovations rather than specific implementation details.

### 4.1. Swin Transformer

The Swin Transformer architecture addresses ViT's limitations in dense prediction tasks caused by the quadratic scaling of self-attention in relation to image size [10]. It functions as a general-purpose transformer backbone that achieves linear complexity while maintaining strong global feature extraction performance.

#### Improvements

Linear complexity is achieved by utilizing windowed self-attention, reducing the number of pairwise interactions between image patches [10]. Here is a simplified explanation: If an image has $h \times w$ windows and each window contains a fixed number of patches $M \times M$, the complexity is reduced from $O((hw)^2)$ for global self-attention to a linear complexity of $O(h \cdot w \cdot M^2)$ for windowed self-attention. Since $M^2$ is a constant and does not increase with the image size, this reduction allows the model to handle high-resolution images efficiently, overcoming the quadratic scaling issue of ViT. As the image size increases, more windows are created, but the complexity remains linear.

However, local self-attention cannot capture long-range dependencies across multiple window partitions. Therefore, the Swin Transformer introduces shifting window partitioning and hierarchical feature maps to allow cross window connections [10]. Figure 3 presents a simplified overview of how these concepts interact with each other within the Swin architecture.

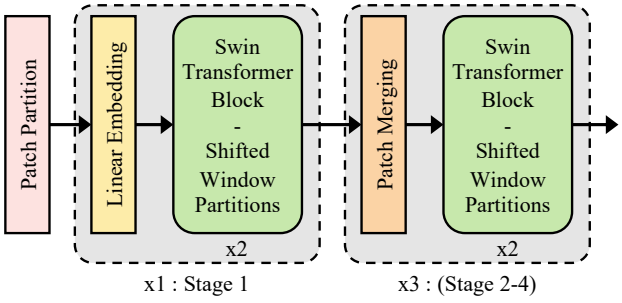


*Figure 3.* Simplified Swin Transformer Architecture

**Shifted Window Mechanism** introduces connections between neighboring windows, by introducing different partitions between consecutive transformer layers within a single stage. This modification is visually highlighted in green in Figure 3. For instance, if one layer's windows are arranged in a regular grid, the next layer shifts these windows to create a staggered grid, as shown in Figure 4. This shift allows self-attention to span across different windows from the previous layer, enabling the model to capture relationships between tokens that were originally in separate windows.

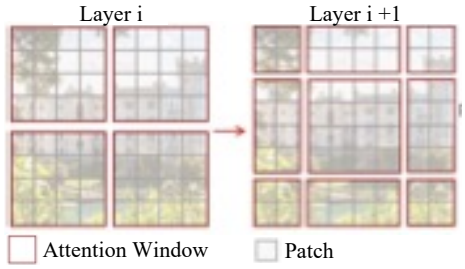


*Figure 4.* Swin: Shifted Window Partitions [10]

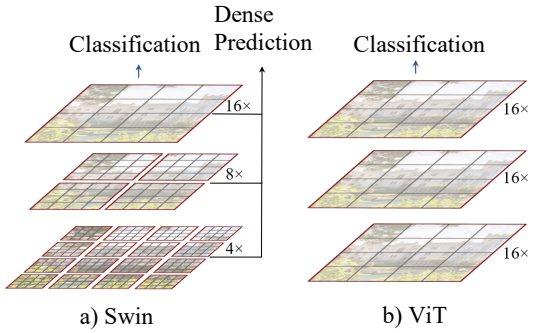


*Figure 5.* Hierarchical Feature Maps using Merging Patches [10]

**Patch Merging** is applied between each stage to create hierarchical feature maps, thereby providing additional connections between local windows. Figure 5 illustrates how the Swin Transformer gradually merges neighboring small patches into larger, more abstract patches in deeper layers. This process offers multiple benefits:

- It provides additional connections between local windows to capture long-range and cross-window dependencies.
- The number of tokens is reduced progressively in deeper layers, thus reducing the computational complexity.
- Tasks like segmentation or detection benefit from multi-scale features, as they require understanding objects at different scales [10]. This approach is similar to CNN U-Net [11] and Pyramid Vision Transformer [12].

#### Limitations on Large Images

The Swin Transformer excels with high-resolution images and performs well in dense prediction tasks like segmentation and detection [10]. However, it may struggle with vision tasks requiring global feature extraction due to the limited receptive field of its local windows [8]. This is because the shifted window mechanism and hierarchical

feature maps primarily capture dependencies between neighboring windows and patches. The farther apart the patches are, the weaker their connection becomes.
Additionally, larger images increase the number of windows, leading to linearly higher attention costs at earlier stages before patch merging is applied [9]. The resulting decrease in image throughput is shown in Table 2.

### 4.2. FasterViT

FasterViT is designed to enhance the throughput of vision transformers, particularly for large high-resolution images, addressing limitations seen in Swin Transformer and ViT [9]. Its architecture, visible in Figure 7, includes a hybrid CNN-Transformer design and Hierarchical Attention that balances local and global feature extraction.

#### Improvements

**The Hybrid Design** leverages CNN in early stages to downsample higher resolution data. The later stages transition to transformer blocks using self-attention. This approach offers several advantages:

- **Data Efficiency:** FasterViT achieves high performance without requiring large scale training, by utilizing the inductive bias information of CNNs [9].
- **High Throughput:** Early downsampling before applying attention creates high-level tokens, reducing the computational load in the initial stages. This contrasts with the Swin Transformer, which processes high-resolution inputs longer before downsampling [9]. This design choice allows FasterViT to maintain high image throughput even with large model sizes, as shown in Table 3.
- **Improved Feature Extraction:** FasterViT combines a multi-scale CNN architecture to capture local features with the global modeling ability of transformers, enhancing overall feature extraction [9].

After downsampled feature maps are generated by the CNN stages, Transformer layers apply windowed attention to reduce computational complexity, similar to the Swin Transformer. While the Swin Transformer uses a shifted window mechanism and hierarchical feature maps through patch merging to capture cross-window connections, FasterViT introduces Hierarchical Attention [9].

**Hierarchical Attention** implements a second attention layer using carrier tokens on top of the local attention windows, where embedded image patches serve as tokens. Figure 6 illustrates how each carrier token learns a summary of a single local window by attending to all patch tokens within that window. These carrier tokens then undergo full self-attention to enable cross-window communication. This mechanism enables the application of windowed attention, which is necessary to maintain computational efficiency for large images, while still capturing global dependencies across multiple windows [9]. This edge in performance is evident in Table 3, where FasterViT outperforms competiters in throughput and accuracy.

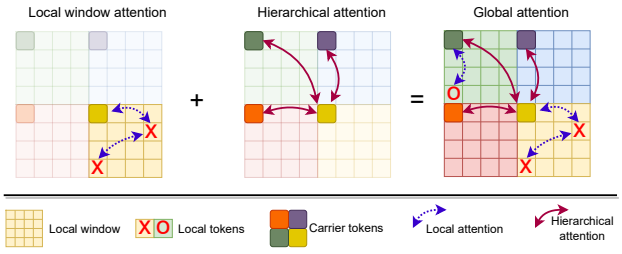


*Figure 6.* Hierarchical Attention in FasterViT [9]

### 4.3. Alternative Methods

There are various solution strategies, some similar to Swin and FasterViT, like T2T-ViT [13] and PVT [12], while others differ significantly. DeiT, for example, focuses on knowledge distillation to compensate for the lack of inductive bias without large-scale training [14]. Here, the Transformer learns from a pre-trained teacher model, typically a CNN, by using a distillation token alongside class and patch tokens. This approach allows the student model (DeiT) to mimic the teacher's output, thereby enhancing its performance.

## 5. Conclusion & Research Direction

The rapid development of Vision Transformers suggests a promising future for their application in computer vision. Early models like ViT demonstrated that Transformers could replace or complement CNNs, motivating further research. A key trend in advancing ViT is reducing the quadratic complexity of attention while maintaining high performance. One of the most promising advancements is the FasterViT architecture, which combines near-linear complexity with competitive performance in classification and dense predictions. This, along with its high image throughput, makes it particularly appealing for real-world applications.

However, research is not just about improving performance metrics but also about expanding the range of applications. For instance, the success of Transformers in both vision and language tasks underscores the potential for unified modeling, as evidenced by promising results from models like CLIP [15] and ALIGN [16]. These models leverage large-scale datasets to learn visual and textual representations simultaneously, enabling them to perform tasks such as image captioning and visual question answering with high accuracy. Additionally, they have shown proficiency in zero-shot learning, where the model can make predictions about classes it has never seen before. As this research progresses, we can expect more groundbreaking advancements, unlocking new capabilities across multiple domains.

## References


[1] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[2] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[4] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[5] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[6] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[7] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems* 26 (2013).

[8] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[9] Ali Hatamizadeh et al. "Fastervit: Fast vision transformers with hierarchical attention". In: *arXiv preprint arXiv:2306.06189* (2023).

[10] Ze Liu et al. "Swin transformer: Hierarchical vision transformer using shifted windows". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022.

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer. 2015, pp. 234–241.

[12] Wenhai Wang et al. "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 568–578.

[13] Li Yuan et al. "Tokens-to-token vit: Training vision transformers from scratch on imagenet". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 558–567.

[14] Hugo Touvron et al. "Training data-efficient image transformers & distillation through attention". In: *International conference on machine learning*. PMLR. 2021, pp. 10347–10357.

[15] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[16] Chao Jia et al. "Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 4904–4916. URL: https://proceedings.mlr.press/v139/jia21b.html.

[17] Alexander Kolesnikov et al. "Big transfer (bit): General visual representation learning". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer. 2020, pp. 491–507.

[18] Qizhe Xie et al. "Self-training with noisy student improves imagenet classification". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10687–10698.

[19] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[20] Ross Wightman, Hugo Touvron, and Hervé Jégou. "Resnet strikes back: An improved training procedure in timm". In: *arXiv preprint arXiv:2110.00476* (2021).

[21] Zhengsu Chen et al. "Visformer: The vision-friendly transformer". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 589–598.

[22] Zhengzhong Tu et al. "Maxvit: Multi-axis vision transformer". In: *European conference on computer vision*. Springer. 2022, pp. 459–479.

# APPENDIX

## A. Classification Benchmarks

This appendix section provides a detailed comparison of benchmarks from various papers on ImageNet classification. These experiments are listed separately to ensure consistency, since metrics are only comparable if they come from the same experimental setup.

### A.1. ViT Results

| | ViT-Large | ViT-Large | BiT-L (ResNet) | Noisy Student (EfficientNet) |
|---|---|---|---|---|
| **Pretrained dataset** | JFT-300M | ImageNet-21k | weakly labelled JFT | unlabelled JFT |
| **Model Parameter** | 307M | 307M | 928M | 480M |
| **ImageNet** | $87.76 \pm 0.03$ | $85.30 \pm 0.02$ | $87.54 \pm 0.02$ | 88.4/88.5 |
| **ImageNet ReaL** | $90.54 \pm 0.03$ | $88.62 \pm 0.05$ | 90.54 | 90.55 |
| **CIFAR-10** | $99.42 \pm 0.03$ | $99.15 \pm 0.03$ | $99.37 \pm 0.06$ | - |
| **CIFAR-100** | $93.90 \pm 0.05$ | $93.23 \pm 0.05$ | $93.51 \pm 0.08$ | - |
| **Oxford-IIIT Pets** | $97.32 \pm 0.11$ | $94.67 \pm 0.15$ | $96.62 \pm 0.23$ | - |
| **Oxford Flowers-102** | $99.74 \pm 0.00$ | $99.61 \pm 0.02$ | $99.63 \pm 0.03$ | - |
| **TPUv3-core-days** | 0.68k | 0.23k | 9.9k | 12.3k |

*Table 1.* Comparison of ViT models with state-of-the-art CNNs such as Big Transfer (BiT) [17] and Noisy Student [18] on popular image classification benchmarks. The accuracy results are derived from the paper "An Image is Worth 16x16 Words" [6]. The mean and standard deviation of accuracies is averaged over three fine-tuning runs. Large ViT models pre-trained on the JFT-300M dataset outperform ResNet-based baselines on all datasets, while requiring substantially fewer computational resources for pre-training. The computational cost in TPUv3-core-days refers to the total training computation, calculated as the number of TPU v3 cores used multiplied by the training duration in days. The ViT model pre-trained on the smaller ImageNet-21k dataset performs well but not better than BiT-L.

### A.2. Swin Transformer Results

| Model | Image Size | #Param. | Throughput (Image/Sec) | Top-1 Acc. (%) |
|---|---|---|---|---|
| **(a) Regular ImageNet-1K trained models** | | | | |
| **EffNet-B3 [19]** | $300^2$ | 12M | 732.1 | 81.6 |
| **EffNet-B7 [19]** | $600^2$ | 66M | 55.1 | 84.3 |
| **ViT-B/16 [6]** | $384^2$ | 86M | 85.9 | 77.9 |
| **ViT-L/16 [6]** | $384^2$ | 307M | 27.3 | 76.5 |
| **Swin-T [10]** | $224^2$ | 29M | 755.2 | 81.3 |
| **Swin-S [10]** | $224^2$ | 50M | 436.9 | 83.0 |
| **Swin-B [10]** | $224^2$ | 88M | 278.1 | 83.5 |
| **Swin-B [10]** | $384^2$ | 88M | 84.7 | 84.5 |
| **(b) ImageNet-22K pre-trained models** | | | | |
| **ViT-B [6]** | $384^2$ | 86M | 85.9 | 84.0 |
| **ViT-L [6]** | $384^2$ | 307M | 27.3 | 85.2 |
| **Swin-B [10]** | $224^2$ | 88M | 278.1 | 85.2 |
| **Swin-B [10]** | $384^2$ | 88M | 84.7 | 86.4 |
| **Swin-L [10]** | $384^2$ | 197M | 42.1 | 87.3 |

*Table 2.* Comparison of ImageNet-1K classification results based on experiments from the Swin Transformer paper [10]. The image throughput of both the Swin Transformer and ViT decreases significantly with increasing image size and model parameters. In contrast, EffNet-B3 maintains better image throughput even with larger image sizes.

### A.3. FasterViT Results

| Model | Image Size | #Param. | Throughput (Img/Sec) | Top-1 Acc. (%) |
|---|---|---|---|---|
| **CNN-Based** | | | | |
| **ResNetV2-101 [20]** | $224^2$ | 44.5M | 4019 | 82.0 |
| **Transformer-Based** | | | | |
| **Swin-T [10]** | $224^2$ | 28.3M | 2758 | 81.3 |
| **Swin-S [10]** | $224^2$ | 49.6M | 1720 | 83.2 |
| **Hybrid (CNN + Transformer)** | | | | |
| **Visformer-S [21]** | $224^2$ | 40.2M | 3676 | 82.1 |
| **MaxViT-L [22]** | $224^2$ | 212.0M | 376 | 85.1 |
| **FasterViT (Hybrid Design + Hierarchical Attention)** | | | | |
| **FasterViT-0 [9]** | $224^2$ | 31.4M | 5802 | 82.1 |
| **FasterViT-3 [9]** | $224^2$ | 159.5M | 1780 | 84.9 |
| **FasterViT-5 [9]** | $224^2$ | 957.5M | 449 | 85.6 |
| **FasterViT-6 [9]** | $224^2$ | 1360.0M | 352 | 85.8 |

*Table 3.* Comparison of ImageNet-1K classification results based on experiments from the FasterViT paper [9]. Despite increasing model parameters, FasterViT models maintain higher image throughput compared to other models. Generally, there is a trade-off between throughput and accuracy: increasing model size can improve accuracy but typically reduces throughput.

## B. Simplified FasterVit Architecture

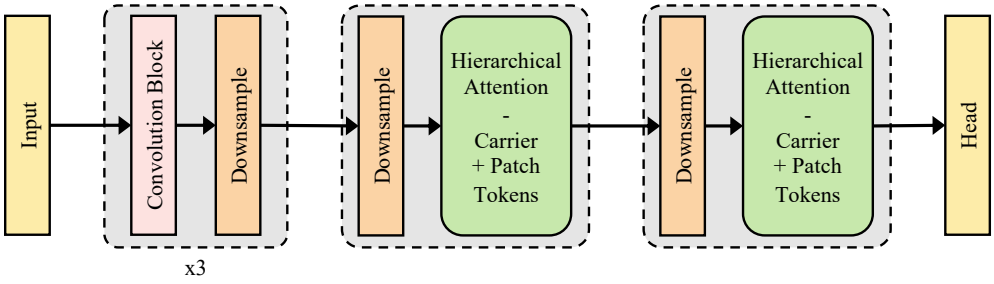


*Figure 7.* Simplified Overview of FasterViT Architecture.