Vision Transformers for Edge Devices - An Overview

Nahid Alam*
Cohere for AI Community
shapla@gmail.com

Simardeep Sethi*
IIT Delhi
Simardeep2792@gmail.com

Steven Kolawole*
ML Collective
steven@mlcollective.org

Nishant Bansali
Cohere for AI Community
nishantbhansali80@gmail.com

Karina Nguyen
UC Berkeley
karinanguyen@berkeley.edu

Abstract

Vision Transformers (ViTs) have demonstrated state-of-the-art performance on many Computer Vision Tasks. This has led to a proliferation of research, resulting in many variants that address the challenges of vanilla ViT in different ways. To the extent that it can be overwhelming to understand different approaches and research directions. In this paper, we study and categorize Vision Transformers in a few different ways based on their architectures. We specifically focus on the state of Vision Transformers targeted for edge devices. Since Transformers are more compute intensive, existing research tries to incorporate Convolutional Neural Networks (CNNs) into the Transformer architecture to reduce the model parameters and increase the inference speed. We found out that ViTs can either be a pure transformer or a combination of CNNs and transformers. We compare and contrast the architectural differences to understand specific changes that make ViTs more suitable for mobile devices. This paper aims to serve as a baseline for future research on Vision Transformers for edge devices.

1 Introduction

Transformers are a type of Neural Network Vaswani et al. [1] introduced in 2017 for Natural Language Processing (NLP) applications. Language models such as BERT [2] pre-trained bidirectional encoders - in this case, transformers. BERT achieved state-of-the-art performance in 11 downstream NLP tasks when it was published. Generative Pre-trained Transformer 3 (GPT-3) is another language model that uses transformers to generate human-like text [3] In contrast to the LSTM [4] network, where inputs are processed by the model in sequences, transformers process input data in parallel.

Inspired by the success of Transformer models in NLP applications, A. Dosovitskiy et al. [5] introduced Vision Transformer (ViT) for Computer Vision (CV) applications. ViT takes a 16x16 patch of image and passes it through a Transformer architecture in sequence to classify an image.

A common challenge with Vision Transformers is that although they perform well in web-scale applications running on servers, they are not suitable for mobile and edge devices. Mobile devices typically have much less compute power and memory bandwidth, requiring models with fewer parameters and smaller sizes. The massive number of parameters of these models leads to model weights that are in the order of 500MB to GB and very high inference latency. As a result, these models are unsuitable for resource-constrained devices such as security cameras, mobile devices, and

* Equal Contribution.

Preprint. Under review.

various other edge sensors. Compared to transformers, CNNs are multiple times faster [6], [7], and that's why they are the architecture of choice for mobile devices.

Various architectures have recently been proposed to target CV tasks in mobile devices that combine transformers with CNNs. In addition, techniques such as quantization, pruning, distillation, etc., are applied to an existing big model to fit it in mobile devices [8].

This paper aims to explore transformer-based architectures optimized for edge devices, identify a few common patterns in design choices and serve as a guide for edge-optimized Vision Transformer research.

2 Vision Transformer

A. Dosovitskiy et al. [5] proposed a Vision Transformer that used 300MB of an in-house dataset at Google. The algorithm uses a 16x16 patch of images and flattens them to form a sequence. Each 16x16 patch is treated in parallel, passing through a linear layer and then through a Transformer. This is a quadratic operation since the transformer used here is essentially an attention network. The success of the Vision Transformer was followed up with DeiT [9] - a data-efficient Vision Transformer with knowledge distillation. The authors used 30MB of the ImageNet dataset - a 10x reduction of the training dataset to train the DeiT architecture. Liu et al. [10] authored the Swin Transformer that takes 4x4 image patches and uses a shifted window self-attention network instead of the typical self-attention network. The shifted window approach helps reduce the quadratic complexity to a linear one. These Vision Transformer models have shown 77.9% to 81.3% Top-1 accuracy on ImageNet dataset [9],[5],[10] and have been used in many downstream image recognition task such as classification [11],[12], object detection [13], [14] and segmentation [15],[16].

Although the Top-1 accuracy scores of these Vision Transformer models are impressive, they are generally too big and too slow to deploy on edge devices [6]. The Multi-Head Self-Attention (MHSA) operation in Transformers is quadratic in nature. The MLP module in ViT projects the embedding space by a factor of four, applies non-linearity, and then projects it back to its original shape. As a result, the latency of ViT models, in general, is unrealistic for edge devices such as iPhone, Google Edge TPU [17], NVIDIA Jetson Nano, Intel Edge devices [18], [19], Hexagon DSP, etc. Therefore we need architectural changes that can result in a Vision Transformer model with a smaller number of parameters and lower inference latency.

3 Vision Transformers at the Edge

A few techniques can make a model suitable for edge deployment. The techniques can be broadly classified into two categories. First, changing the model after the model is trained. For changing the trained model, techniques such as quantization, pruning [20], [21] and a mix of model sparsification techniques [8] are applied. Architectural changes can be a mix of transformer+CNN model, MLP based models [22] or sparse models [23].

In this section, we will discuss the architectural changes that make Vision Transformers more suitable for the edge.

3.1 Exploring the Architectures

Recent work has been focused on hybrid architectures to take advantage of both CNNs and Vision Transformers. The majority of these models reduce the time complexity of the MHSA module by modifying the attention operation or completely removing it. Efficient-ViT [24] proposes a high resolution low-computation image recognition model. It consists of MobileNetV2 [25] convolutional layers followed by Linear Attention layer. This is followed by a Feed Forward Network with deformable CNN. Similarly, EdgeNeXt [26] proposes a Split Depth-wise Transpose Attention (SDTA) encoder instead of the vanilla MHSA module. The SDTA encoder first learns the multi-scale feature information using 3x3 Depth-Wise convolution and feeds it to the Transpose Attention. Then the Transpose Attention layer creates the dot product of the channel dimensions instead of the spatial dimension. As a result, the SDTA encoder executes in linear time.

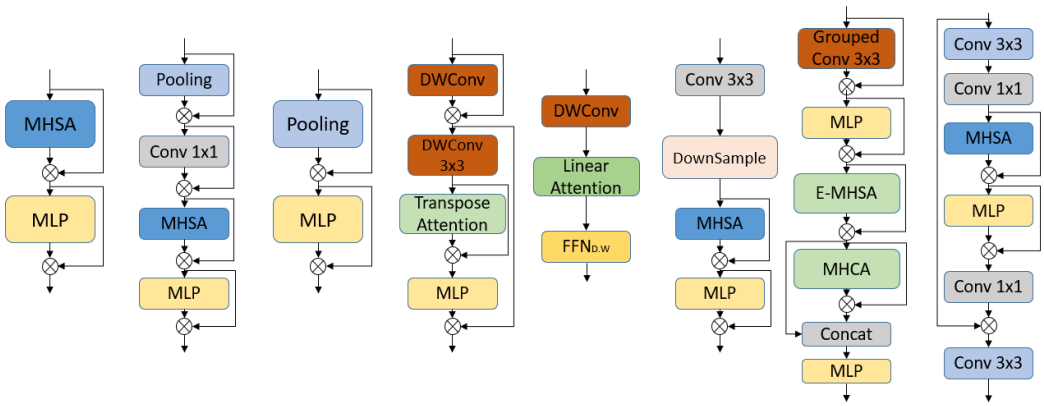

Figure 1: Abstract Vision Transformer Architectures From Left to Right: Vanilla Transformer; Efficient-Former; PoolFormer; EdgeNext; EfficientViT; LeViT; NextViT; MobileViT

NextViT [27] proposes a hybrid structure with each stage consisting of alternate Convolutional and Transformer blocks. The Convolutional block, referred to as Novel Convolutional Block (NBC), uses Multi-head Convolution Attention formed of a grouped 3X3 Convolution operation and an MLP. The Transformer block, referred to as Novel Transformer Block (NTB), uses MHSA consisting of a pooling layer for the Key and Value vectors during self-attention operation. Moreover, channel reduction is applied to improve inference speed. PoolFormer [28] suggests an alternative token mixture module - MetaFormer, where the token mixture can be modified. PoolFormer uses a pooling operation as the token mixer, resulting in a similar accuracy with fewer parameters compared to ViT.

LeViT [29] uses an attention-shrinking block before each stage, which scales the Q vector while increasing the channels moving through the soft activation. LeViT does not impose positional embedding after each block, unlike other Vision Transformers. However, they encode the location information through an attention bias term to the attention map. The MLP module of ViT is replaced by a 1x1 Convolution layer. The training process of LeViT is similar to DeiT, which uses a distillation token head after the last stage.

MobileViT [7] introduces the MobileViT module responsible for capturing the interaction of local and global features of the image. The local representation is captured using NxN convolutions with 1x1 point-wise convolution, whose output feature map serves as an input for the global feature capturing module. This module unrolls the feature map into patches, which are then used as input to ViT. The fusion of these representations is carried out using skip connections.

Next, there is a category of models that executes specific layers in parallel to improve time complexity. Mobile-Former [30] has four primary modules: Mobile block, Former block, Mobile→Former block, and Former→Mobile block. The Mobile block consists of a 3x3 depthwise convolution layer used to extract local representations from the image. The Former block consists of a basic transformer module with MHSA. This block extracts global representations that feed into the mobile block through a bridge. The Mobile→Former and Former→Mobile blocks are used to fuse the local and global features using a two-way cross-attention.

Similarly, MixFormer [31] combines local window self-attention and depthwise convolutions in a parallel design. These two parallel branches communicate with each other through channel interactions and spatial interactions, providing complementary clues for better representation learning in both branches. The output is then concatenated and passed through a Feed Forward Network.

EfficientFormer [32] aims to replicate MobileNet in terms of latency while maintaining optimal performance. The authors provide insights into latency for different operations of vision transformers and observe that consistent feature dimension is essential for the choice of token mixer. The overall network is divided per the tensor dimension. First, based on 4D tensors. A Convolutional layer termed MB4D is applied on 4-Dimensional tensors. Second, based on 3D tensors consisting of MHSA as a token mixer for the transformer module. Embedding layers are used between stages to project the token length to a lower dimension. This alleviates the dimension mismatch, which is shown to be a major cause of slow inference speed at edge devices.

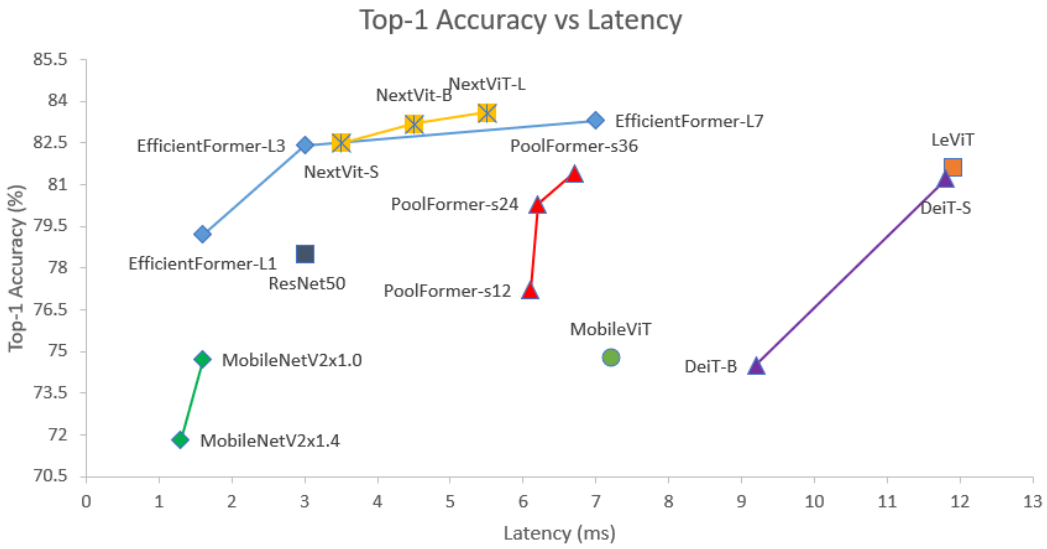

Figure 2: Top-1 Accuracy vs. Latency for Image Classification on ImageNet-1K using CoreML.

The floating point precision in Vision Transformer models increases latency at edge devices. Quantization can convert those floating point operations to integer operations. There are various methods for quantization for CNNs. However, a pure integer-only arithmetic operation on the Vision Transformer is an open challenge. I-ViT [33] addresses this gap by applying a dyadic arithmetic pipeline to efficiently use integer-only arithmetic for linear operations. For non-linear operations, I-ViT introduces novel operations Shiftmax, ShiftGelu, and I-LayerNorm. Shiftmax serves as an alternative to the softmax function. It uses the scaling factor of quantization obtained from the MatMul operation and the quantized integer weight to get an integer-only value. Similarly, ShiftGelu works as an alternative to the Gelu activation function and I-LayerNorm for an integer-only layer normalization operation that improves both the top-1 accuracy and latency for edge devices.

3.2 Comparing Results

Figure 2 displays Top-1 accuracy vs. latency on several Vision Transformer models that might be suitable for edge devices. These models are trained on ImageNet-1K for 300 epochs with AdamW optimizer on an image resolution of 224x224. Latency is measured using the CoreML framework.

Table 1 compares a few state-of-the-art models based on CNN, Hybrid, and pure transformer-based architectures. ResNet50 [34] has 78.5% top-1 accuracy on image classification tasks with a 3ms latency on CoreML. DeiT-S has 81.2% top-1 accuracy while maintaining a similar number of parameters. However, the latency of the model is much higher than that of ResNet50. Furthermore, EfficientFormer-L3 has an 82.4% accuracy with a considerably low latency of 3ms. In comparison, NextViT-S has an 82.5% accuracy on the classification task, with a 3.5ms latency on CoreML.

These results suggest that EfficientFormer or NextViT might be the most suitable models for edge devices, considering their performance based on the accuracy-latency tradeoff. Furthermore, a hybrid strategy of stacking CNNs and transformer blocks together might be better than using the traditional hybrid strategy of stacking CNNs at earlier layers and transformer blocks at later layers.

4 Conclusion

This paper aims to serve as a bird's-eye view of the current state of ViTs for edge devices. We hope the community finds it useful as a reference for their background research. We surveyed ten different ViT architectures that are specifically designed for mobile applications in edge devices. We identified a few different models with around 10M parameters and reasonable latency-accuracy trade-offs. Throughout this process, we developed some insights into designing and training ViTs that will be suitable for mobile devices.

Table 1: State-of-the-art models on ImageNet classification task

Model	Parameters(M)	Top-1 Accuracy (%)	Latency(ms)
ResNet50	25.5	78.5	3
MobileNetv2	6.1	74.7	1.6
DeiT-T	5.9	74.5	9.2
DeiT-S	22.5	81.2	11.8
EdgeNeXt	5.6	78.8	-
MobileViT	2.3	74.8	7.2
LeViT	18.9	81.6	11.9
EfficientFormer-L1	12.3	79.2	1.6
EfficientFormer-L3	31.3	82.4	3
EfficientFormer-L7	82.1	83.3	7
MobileFormer-214	9.4	76.7	-
PoolFormer-s12	12	77.2	6.1
PoolFormer-s24	21	80.3	6.2
PoolFormer-s36	31	81.4	6.7
EfficientViT	10.9	79.7	-
NextViT-S	31.7	82.5	3.5
NextViT-B	44.8	83.2	4.5
NextViT-L	57.8	83.6	5.5

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

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

[6] Xudong Wang, Li Lyna Zhang, Yang Wang, and Mao Yang. Towards efficient vision transformer inference: a first study of transformers on mobile devices. In *Proceedings of the 23rd Annual International Workshop on Mobile Computing Systems and Applications*, pages 1–7, 2022.

[7] Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. *CoRR*, abs/2110.02178, 2021.

[8] Mark Kurtz, Justin Kopinsky, Rati Gelashvili, Alexander Matveev, John Carr, Michael Goin, William Leiserson, Sage Moore, Bill Nell, Nir Shavit, and Dan Alistarh. Inducing and exploiting activation sparsity for fast inference on deep neural networks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5533–5543, Virtual, 13–18 Jul 2020. PMLR.

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

[10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030, 2021.

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *CoRR*, abs/2104.14294, 2021.

[12] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *CoRR*, abs/2106.13230, 2021.

[13] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *CoRR*, abs/2005.12872, 2020.

[14] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Improved multiscale vision transformers for classification and detection. *CoRR*, abs/2112.01526, 2021.

[15] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *CoRR*, abs/2105.15203, 2021.

[16] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *CoRR*, abs/2112.01527, 2021.

[17] Google Cloud. Edge TPU - Run Inference at the Edge | Google Cloud. `https://cloud.google.com/edge-tpu`, 2022.

[18] Intel Corporation. Intel Movidious VPU. `https://www.intel.com/content/www/us/en/products/docs/processors/movidius-vpu/myriad-x-product-brief.html?wapkw=movidius`, 2022.

[19] Intel Corporation. Intel AI FPGA. `https://www.intel.com/content/www/us/en/products/details/fpga/stratix/10/nx.html`, 2020.

[20] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

[21] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[22] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Resmlp: Feedforward networks for image classification with data-efficient training, 2021.

[23] William Fedus, Jeff Dean, and Barret Zoph. A review of sparse expert models in deep learning, 2022.

[24] Han Cai, Chuang Gan, and Song Han. Efficientvit: Enhanced linear attention for high-resolution low-computation visual recognition. *arXiv preprint arXiv:2205.14756*, 2022.

[25] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018.

[26] Muhammad Maaz, Abdelrahman Shaker, Hisham Cholakkal, Salman Khan, Syed Waqas Zamir, Rao Muhammad Anwer, and Fahad Shahbaz Khan. Edgenext: Efficiently amalgamated cnn-transformer architecture for mobile vision applications. *arXiv preprint arXiv:2206.10589*, 2022.

[27] Jiashi Li, Xin Xia, Wei Li, Huixia Li, Xing Wang, Xuefeng Xiao, Rui Wang, Min Zheng, and Xin Pan. Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios. *arXiv preprint arXiv:2207.05501*, 2022.

[28] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10819–10829, 2022.

[29] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12259–12269, 2021.

[30] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5270–5279, 2022.

[31] Qiang Chen, Qiman Wu, Jian Wang, Qinghao Hu, Tao Hu, Errui Ding, Jian Cheng, and Jingdong Wang. Mixformer: Mixing features across windows and dimensions, 2022.

[32] Yanyu Li, Geng Yuan, Yang Wen, Eric Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *arXiv preprint arXiv:2206.01191*, 2022.

[33] Zhikai Li and Qingyi Gu. I-vit: Integer-only quantization for efficient vision transformer inference. *arXiv preprint arXiv:2207.01405*, 2022.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.