A Survey on Efficient Training Methods for Vision Transformers on Edge Devices

Prachi Natu[1], Shachi Natu[2]
Dept. of Data Science, UMIT, SNDT University, Mumbai, India[1]
Dept. of Computer Engineering, TSEC, Bandra, India[2]

Abstract— *With rapid developments in deep learning models Vision Transformers (ViT) are being widely used for computer vision applications. Self-attention mechanism enables vision transformers to extract global information from input data and capture long range dependencies in images in contrast to convolutional neural networks (CNN). Hence vision transformers perform better than CNN. But trainable parameters of vision transformers and hence model size of vision transformers are significantly larger than CNN. Also, memory consumption and hence resultant cost of the model increases with increase in input image or video size. This causes a restriction of using ViT in edge devices as these devices have limited storage and computational capacity. Also inference latency is more due to larger model size. Using artificial intelligence algorithms on edge devices results in processing data locally and results in faster response time, low power consumption and improved security. Deploying these models on edge devices is tough due to restrictions on memory as well as computing abilities. This paper reviews various methods used to reduce the vision transformer model size and trainable parameters and to optimize the performance of ViTs on edge devices and also identifies challenges in it.*

Keywords— *Vision transformer; Pruning; Quantization; Knowledge distillation; edge devices.*

I. INTRODUCTION

Since last decade, use of deep learning-based architectures has considerably increased in industrial vision applications as well as in language applications due to their ability to process large and complex data. Convolutional neural networks (CNN) [1,2], Recurrent neural network (RNN) [3], Long short term memory (LSTM) [4] have come into picture for image and language applications. Further, transfer learning [5,6] provided the facility to reuse models trained on one task for another task. In 2014 Bahdanau [7] proposed an attention mechanism in deep learning for machine translation, where the model focuses on selected parts of input data that are relevant to input while processing. Authors have introduced the model which aligns and translates jointly. It means that when the model translates a specific word in input sentence, it also searches for the position, where the most relevant information in the input sentence is present i.e. it captures long term dependency in sequence-to-sequence models. Using context vectors associated with position in input sentence and all previously generated target words, the model predicts translation of a word under consideration. Hence, improved translation is obtained in the encoder-decoder model even if sentences are longer than the one in training corpus. Further in 2017 Vaswani et al. [8] proposed self-attention mechanism, which led to development of transformer architecture. Transformer does not contain recurrent neural network (RNN) layers and convolutional neural network (CNN) layers. Transformer processes all tokens parallelly increasing efficiency and scalability. Due to their computational efficiency and scalability, transformers became the milestone in natural language processing (NLP) applications.

Transformer architecture used in NLP inspired their use for computer vision applications too. Many researchers tried to combine transformers with CNN [9,10]. Replacing CNN by transformer architecture was also experimented [11,12] but scalability was not achieved because of specific attention mechanisms used in transformers. Dosovitskiy et. al. [13] proposed transformer for computer vision applications in 2021 by applying it for training image classification task first. This started a remarkable shift in computer vision where convolution operation is replaced by attention mechanisms. But use of vision transformers for computer vision applications needs a very large amount of training data to get closer or even better performance as compared to CNN. Vision transformers are also being used in many computer vision applications like object detection, semantic segmentation, action recognition, image generation are some of them.

Organization of the remaining paper is as follows: Section II describes details of Vision Transformer with self attention mechanism used in it. Section 3 summarizes how vision transformer can be used for different types of vision applications and limitations of using them on resource constrained devices. Section 4 details about designing and training methods of transformers for resource constrained devices.

II. VISION TRANSFORMER

Transformers capture long term dependency in the input and process tokens parallelly leading to better speed of operation. Due to self-attention mechanism transformers require high computational resources and need large amount of dataset to get optimized performance. Dosovitskiy et al. [14] proposed vision transformer in 2021, in which instead of combining attention with CNN, a pure transformer can be directly applied to images. For 2D images they have patchified the image and then flattened each small patch to make it 1D input to vision transformer. Detail architecture of proposed vision transformer by authors is shown in Figure1.

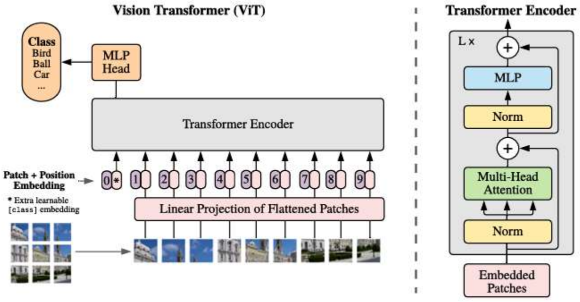

Fig.1 Vision Transformer Architecture Proposed by Dosovitskiy

It is further mapped to linear projection of dimension D. Output of this linear projection is called patch embedding. A special CLS token is prepended to this patch embedding, similar to BERT in NLP. Positional embedding is added to patch embedding and then it is passed as input to encoder. Each encoder block [9] contains stack of normalizing layer followed by Multi head Self attention (MSA) and Multi-Layer Perceptron (MLP) block. Self-attention mechanism finds the relevance of one token with other tokens which leads to extraction of global information from input features in patch embedding.

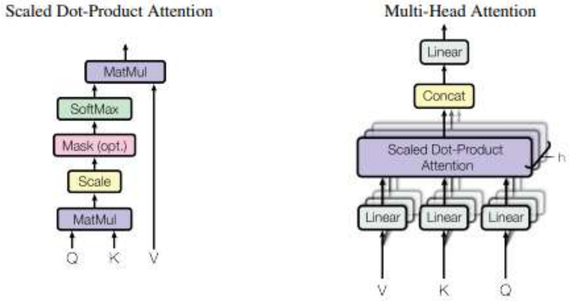

Fig.2 Self attention in transformers (left) and Multi head attention in transformers (right)

III. VISION TRANSFORMERS FOR VARIOUS COMPUTER VISION APPLICATIONS AND LIMITATIONS

For complex real-life applications like autonomous driving, surveillance applications where occlusion can be major issue and can not be resolved to expected level by CNN, ViTs can be used. In medical image applications of segmentation and classification where higher accuracy is important, ViTs can be preferred over CNNs. Also, in satellite imagery segmentation and classification ViTs can process these complex data faster and generate more accurate results than CNNs. Some of the real-world applications which use variants of ViTs are discussed below:

In smartphone, wearables and IoT devices biometric authentication is used for secure unlocking. MobileViT [21], EfficientViT [22] are used for such applications. EfficientViT is also used in real time augmented reality filters in apps like Snapchat. For real time threat (anomaly) detection edge AI cameras use EfficientViT. In autonomous car driving applications, accurate real time object detection (pedestrians, traffic signs, other vehicles on the road etc.) is very important. Tesla’s Full Self Driving System (FSD) uses vision transformers. Scanning of real time medical images on smart devices uses EfficientViT. Smartwatches with the inclusion of TinyViT [14] can detect skin diseases and other anomalies. For real time arial image analysis, ViTs are also used in drones and drone detection [15], in processing thermal images as well as night time drone footages. For plant health assessment [16], pest detection [17] in agriculture field, ViTs are used. For example, AgriViT is used for pest detection, yield prediction [18]. In industrial applications like defect detection [19], obstacle detection by robot ViTs perform major role [20]. The range of applications is very wide and ViTs can be suitable than CNNs for many of these real time applications on edge devices.

Edge devices like IoT devices, embedded systems etc. as the name suggests have limited memory and computational capacity. Vision transformers use image patchification as first step. For high resolution images, number of patches increase. Further when self-attention mechanism is used, it results in very high computational complexity O(N^2) where N denotes the number of patches. In turn it makes the model computationally expensive. Also, each layer involves matrix multiplication which need more processing power. ViTs need extensive amount of data for efficient performance. Hence model size is high with millions of parameters. It requires high memory capacity which is limitation of the resource constrained devices. Due to heavy computation in transformers, high power consumption may require which can make these devices to fail while processing. Inference latency in real time applications is another issue observed in such devices due to heavy computations. These constraints pose a need to efficiently design and train ViTs [21].

This paper focuses on details of major designing and training methods for efficient working of ViTs for resource constrained devices.

IV. DESIGNING AND TRAINING METHODS OF TRANSFORMERS FOR RESOURCE CONSTRAINED DEVICES

Issues in using ViTs on resource constrained devices can be addressed using various methods so that ViT can perform on such devices and with optimized performance. The methods which show significant impact on optimizing the performance of ViTs on Edge devices are:

- Designing compact architecture: Designing transformer architecture which can work in resource constrained environment
- Model pruning: Removing less critical parameters in ViT to reduce the model size
- Quantization: Lowering the precision of computations by using 8-bit integer instead of 32-bit floating-point number format in mathematical computations in transformer.
- Knowledge distillation: Using a shallow model by sharing and compressing knowledge of deep network model

A. Designing Compact Architecture

Many researchers have focused on different architectures of vision transformers to avoid high computational cost. Multi head self-attention (MSA) mechanism mainly contributes in high computation cost in vision transformers. Researchers have tried to linearize the cost of self attention module to reduce high computation cost. In 2021 Wang et al. [22] proposed pyramid transformer. It controls the scale of feature maps by patch embedding layers which uses progressively shrinking strategy and reduces the computations of larger feature maps. Authors have used spatial reduction attention (SRA) layer instead of MSA in encoder architecture. It reduces complexity of attention operation and resource consumption of vision transformers. Such transformers can be trained on dense partitions of an image to obtain high resolution outputs.

Liu et al. have proposed swin transformer in 2021 [23]. Authors have proposed local multi head self attention within shifted non overlapping window in their paper that gives linear computational complexity. When it is used for inference in real world applications it gave good improvements in latency. Lu et al. [24] have proposed Softmax free transformer called as SOFT. Koohpayegani and Pirsiavash in 2024 [25] have proposed softmax free attention block. Instead of using softmax layer, normalization of query and key matrices is done using L1 normalization. Hence instead of computing exponential operations using softmax, simple multiplication of matrices is performed which reduces computation cost of model.

Mehta and Rastegari have proposed MobileViT [26] which is light weight and low latency transformer for mobile devices. Low latency on mobile devices is decided by different factors like memory access, FLOPS, degree of parallelism and platform characteristics etc. Authors have introduced MobileViT block that replaces local processing in convolutions with global processing using transformers. This allows MobileViT block to learn better representations with a smaller number of parameters and simple training. Its performance was tested on mobile device iphone12 for different patch sizes and it was observed that for smaller patch size it gives better accuracy.

Cai et al. [27] proposed EfficientViT in 2023. An efficient ViT architecture is designed with linear computational cost to handle high-resolution dense prediction tasks such as semantic segmentation and super-resolution. The authors leverage the use of the ReLU-based global attention [28] to achieve both the global receptive field of ViT and linear computational complexity. Setyawan et al. [29] have proposed micro-ViT architecture suitable for edge devices. They have used single head attention which uses group convolutions to eliminate feature redundancy and processes only a fraction of the channels. It reduces the burden of self-attention mechanism. When experimented on ImageNet and COCO dataset, it was observed that operation speed of this micro-ViT is 3.6X faster with 40% higher efficiency with reduction in energy consumption.

In [30], Hu et al. have used Tiny ViT-5 m architecture for classification in agricultural application. It has shown nearly 60.16% decrease in the number of parameters, a 41.03% decrease in FLOPs, and a 1.84% rise in accuracy of model classification when used for agricultural application of red jujube defect classification.

B. Model pruning

Pruning methods have been found to be very useful to reduce model storage size and computational cost of the model. Pruning methods are of two categories: structured pruning and unstructured pruning. In structured pruning, entire structure like layer, attention head or attention block is removed whereas in unstructured pruning individual element at smaller level like neuron or weight is eliminated from model architecture.

In earlier literature, Frankle and Carbin [31] and Han et al. in [32] have proposed unstructured pruning where weights were pruned. These methods can reduce model size considerably provided specialized hardware is used for running these models and get the inference faster. In transformers, feed forward network (FFN) sublayer and multihead attention (MHA) sublayer contribute higher number of parameters. Score based pruning models proposed by Ramanujan et al. [33] assigns a score for each parameter and replaces original matrix of weight parameter by masked version of weight matrix where the low absolute value parameters are replaced by zero by the mask. In case of vision transformers, Zhu et al. [34] have presented pruning approach which identifies the impact of dimensions in each layer of the transformer and then performs pruning accordingly. Dimensions with less importance score were discarded to obtain high pruning ratio without affecting the accuracy of the model. In vision transformer, Multi-Head Self-Attention (MHSA) and multilayer perceptron (MLP) requires large number of parameters. Hence the aim is to decrease the FLOPs of MHSA and MLP. Authors have proposed to prune the dimension of the linear projection by learning their associated importance scores. L1 regulation is applied to sparse the dimensions of the transformer. Lu Yu and Wei Xiang [35] have proposed explainable pruning framework in 2023. They have proposed explainability of the model which learns each prunable unit's contribution to predict each target class. Further layer wise threshold is searched to distinguish between pruned and unpruned units based on their explainability aware mask values.

Kaixin Xu et al. [36] have proposed block structured pruning to handle resource intensive issues of vision transformers and call it as low power ViT (LPVIT). Authors have proposed hardware aware learning to maximize speed of operation and minimize power consumption during inference. Dachuan Shi et al. [37] have proposed pruning for multi-modal transformer models specially for vision-language transformers. This model gives better convergence and higher compression ratio for all compressible components. Yogi et al. [38] have applied sparse regularization technique along with pruning and obtained increase in accuracy by different values for different datasets. Fatih Ilhan et al.[39] have proposed Resource and Computation efficient pruning framework called as RECAP with the goal to reduce GPU memory footprint without affecting the performance of the model. Authors have experimented on various vision-based tasks as well as language-based tasks and observed that their proposed technique offers 65% reduction in GPU footprint.

Grouped Structural Pruning proposed by hamza and Alan analyzes the network using a dependency graph to identify and remove interdependent groups of neurons, weights, filters, or attention heads. The pruning rate is determined by importance score metrics, and the pruned models are typically fine-tuned afterward. This approach is hardware-friendly and maintains the structural integrity of the model by removing inter-dependent parameter groups simultaneously [40]. Experiments show significant speedups and reduced fine-tuning time with minimal accuracy loss, particularly effective for Domain Generalization tasks. Accuracy losses become more pronounced as the pruning rates increase.

A structured pruning method that focuses on pruning channels (dimensions) within the fully connected layers of ViTs is proposed by Yuhan Chao [41]. It incorporates L1 regularization on Batch Normalization (BN) layer scale factors during training to make them sparse, effectively learning importance scores for dimensions. Dimensions with lower importance scores are then cropped. It effectively reduces computational cost and model parameters, particularly in the computationally heavy FC layers of self-attention and MLP modules. But accuracy drops slightly as the pruning rate increases. Patch Pruning Methods proposed by Shui Xiuying [42] involve various strategies for selecting which patches to remove. Static methods use pre-defined criteria like heuristics or saliency, while dynamic methods adapt based on the input, often using attention scores, learned importance modules, or reinforcement learning. Pruning can occur before, during, or after transformer processing, each with different trade-offs. It shows direct impact on reducing input tokens for self-attention. Dynamic methods offer flexibility based on the input image.

Token Reduction via an Attention-based Multilayer (TRAM) network has been proposed by Marchetti et. al. in 2025[43]. This method represents the ViT's attention matrices as a multilayer network and defines a Token Relevance Centrality measure based on this representation. Tokens are pruned based on this centrality score, gradually reducing the number of tokens in each layer. It can work with most ViTs without needing fine-tuning. It doesn't introduce additional parameters, making it lightweight. It does not rely on the CLS token for importance evaluation, making it compatible with ViTs that do not use one. It can potentially speed up both training and inference. TRAM is designed for general ViTs and is not applicable to ViTs with hierarchical structures like the Swin Transformer.

A data-independent pruning method specifically for hierarchical Vision Transformers like the Swin Transformer was proposed by He and Zhou [44]. It analyzes modules (e.g., QKV, MLP) and uses a weight importance metric based on information distortion to guide pruning. Since it is data-independent, it does not require input images for pruning decisions. It effectively addresses challenges specific to hierarchical ViTs and can achieve better accuracy-computation trade-offs compared to uniform magnitude pruning. Guang Yang et. al. have proposed a framework that combines two pruning strategies: self-adaptive token pruning and Hessian-aware layer-wise N:M weight sparsity [45]. It determines token pruning based on attention probability and importance scores, while weight pruning uses the Average Hessian Trace (AHT) to set N:M sparsity ratios layer-wise. This approach is designed for efficient deployment on edge devices. By combining token and weight pruning, it can achieve greater overall efficiency gains. The N:M weight sparsity is hardware-friendly. It demonstrates significant energy efficiency improvements with minimal accuracy loss on edge devices. The reported accuracy loss, while called "negligible," is still a decrease in performance. The layer-wise N:M pruning might fix parameters like N to a single value across layers for hardware simplicity. Instead of manually setting pruning rates, automatic pruning rate adjustment technique was proposed by Ishibashi and Meng [46]. It proposes automatically adjusting the rate during training based on factors like training loss convergence or gradients.

Thus, pruning basically reduces the number of active neuron connections by setting weights to zero. This is useful at inference time because number of non-zero multiplications are reduced to obtain the prediction of class label at the end. But it may not affect the number of parameters in the model as these can be saved with weight as zero value.

C. *Quantization*

Quantization is an efficient technique to reduce memory requirement and computational cost of neural networks [47]. In quantization, 32-bit floating-point values are quantized to 8-bit integer values. When it is used in neural network model, model parameters i.e. weights and activations of model are quantized i.e. represented by lower bit precision to save memory space and computational complexity of the model while maintaining the accuracy.

Two major categories of quantization are Post training quantization (PTQ) [48] and Quantization aware training [59]. Post training Quantization (PTQ) compresses the weights and activations once the model is fully trained. Hence retraining of a model is not required with simulated quantization effect. PTQ is useful when retraining of a model is not feasible or expensive due to its large size. PTQ gave good results on CNN but for the large and complex models like ViTs it degrades accuracy of the model as ViT is very sensitive to quantization of weights, activations and layer normalizations. It results in drop in accuracy with 8-bit quantization and it can be up to 1% [49].

Zhihang Yuan et al. [50] have proposed twin uniform quantization method for vision transformers to reduce quantization errors on activation values after softmax and GELU functions. Here they collected output during forward propagation and gradient of output during backward propagation at each layer before quantization and then optimal scaling factor was searched for each layer to quantize activation values and weight values in that layer. Hessian guided metric was used to minimize the difference in weights before and after quantization based on layer wise reconstruction method [51] and it has been observed that it improved the performance. Zikai Li and Quingyi Gu [52] proposed integer only quantization (I-ViT). Authors have used integer arithmetic and bit shifting to perform entire computational graph of inference. No floating-point arithmetic operations were used. Huihong Shi et al. proposed power of two post training quantization method in [53]. Instead of rounding the scaling factor directly, they used adaptive rounding post training. They have also scaled up hardware resources for accelerators.

Liu et al. in 2023 [54] have proposed post quantization method for transformer activation functions. Proposed strategy was named as NoisyQuant and it adds NoisyBias to GELU activation function. This operation is performed before quantization. It flattens the data peak and makes quantization process smooth and it follows original data distribution with lower bit rate. Some quantization methods focus on compression strategies for specific hardware devices and not only compressing activation functions and attention layers. Lit et al. [55] have proposed a framework for ViT quantization for inferencing on FPGA powered devices. Quantization is applied on FNN module of attention block to speed up inference process. Jiang et. al. have proposed post training quantization method in 2025 known as Architecture Informed post training quantization which is specifically tailored for ViTs. Basically, it incorporates an architecture-informed low-rank compensation mechanism that introduces learnable low-rank weights to mitigate the degradation caused by weight quantization. This mechanism uses network architecture search to determine the appropriate rank for these compensatory weights [56].

Zhuguanyu Wu et. al. proposed a novel PTQ approach that leverages importance estimation based on the Average Perturbation Hessian (APH). It proposes an improved average perturbation Hessian loss to address the issue of inaccurate output importance estimation in block reconstruction [57]. APHQ-ViT also tackles the degradation associated with quantizing post-GELU activations through its MLP Reconstruction (MR) method. MR replaces the GELU activation function in MLP layers with ReLU, which helps reduce the activation range and alleviate imbalanced distributions, making these activations more suitable for quantization.

For specific applications, such as real-time EEG classification using the Brain Signal Vision Transformer (BSVT), dynamic post-training quantization has been explored by Khadka et. al in 2025. This method converts the model to INT8 precision. It dynamically determines the scale factor for activations during runtime based on the data range of each batch, simplifying the tuning process. Symmetric quantization is applied to weights. Certain high-precision operations like Softmax, Layer Normalization, and GELU retain FP32 precision in this approach [58]. Quantization aware training (QAT) [59] applies quantization effect during training. It exposes the model to quantization errors during training. It helps model to adapt to precision loss and learning representation that are more robust to quantization. In QAT, it quantizes weights, activations and attention scores too. In vision transformers layer normalizations and attention mechanism are highly sensitive to precision changes. QAT increases training complexity and computation compared to PTQ. It significantly improves model robustness and enables the deployment of ViTs on edge devices, embedded systems, and specialized accelerators like TPUs and NPUs without substantial accuracy loss.

D. Knowledge Distillation

Due to stacking of self-attention modules on multiple heads, ViTs become expensive. In such case, compressing ViT to reduce the computational cost is necessary. Hinton et al. [60] in 2015, proposed knowledge distillation. Knowledge distillation is a process of transforming large complex model into smaller one without compromising performance. It improves learning efficiency and model understanding without requiring high amount of computational power. Knowledge distillation process has three parts: knowledge, distilling algorithm and teacher-student architecture. There are three types of knowledge distillation: response based, feature based and relation-based knowledge distillation. Each of them is briefly described below [61].

- Response based knowledge distillation: It focuses on the response i.e. final output of teacher model. Student model learns to mimic the response of teacher model. It is done by calculating and minimizing the distillation loss. Distillation loss is the difference between logits of teacher and student model.
- Feature based knowledge distillation- Here, intermediate layers in the deep networks learn and discriminate specific features. These features can be used as knowledge which student model learns from teacher model. Here distillation loss is calculated as the difference between feature activation of teacher model and that of student model.
- Relation based knowledge distillation- Here relationship between feature maps or different data points is used as knowledge. This relationship is captured using different ways like correlation between feature maps, graphs, similarity matrix, feature embeddings or probabilistic distribution based on feature representations.

Various approaches have been used to make student model learn from teacher model in terms of the above three types. Few of them are discussed below:

In [62], Xie et al. have proposed knowledge distillation method where student learns representational features from teacher model using parametric correlation. For larger feature maps it becomes difficult to calculate correlations between spatial locations. But this issue was overcome by splitting the feature maps into many patches. Then distillation is performed for each patch using one to all patch. In [63], Sucheng et al. have proposed a method based on cross inductive bias. Authors have stated that inductive bias in teacher model matters more than their accuracy. If multiple such teacher models with different indictive bias are used then student model can learn variety of knowledge from them. Cross inductive bias vision transformer was proposed which gave better performance using cross inductive bias distillation method. In [64], Touvron et al. have proposed a knowledge distillation method based on distillation tokens specific to transformers. Distillation tokens which are same as that of the class tokens reproduce the labels estimated by teacher model using attention. Both these tokens interact within the transformer. Here distillation tokens pre learn from ConvoNet teacher. In [65], Shixinng et al have proposed logit based knowledge distillation combined with pruning and layer skipping. They have observed 50 % decrease in FLOPs without affecting the accuracy of the model. Yufan liu et al have proposed cross architecture knowledge distillation in [66]. It can distill the knowledge between cross architectures like from CNN to transformers or vice versa. Xiao et al.in [67] and Lin et al. in [68], authors have proposed supervised masked knowledge distillation which is suitable for few shot transformers. This method enables intra class knowledge distillation at class level as well as token level by incorporation label information into self-distillation.

Knowledge distillation framework for semantic segmentation has been proposed by R Liu et al. in 2022 [69]. Here student transformers learn by distilling feature maps and patch embeddings of teacher transformers. It skips the long pre-training process and decreases the FLOPs by more than 85.0%. Authors have proposed two modules to realize feature map distillation and patch embedding distillation, respectively: 1) Cross Selective Fusion (CSF): This module makes knowledge transfer possible between cross-stage features through channel attention and feature map distillation within hierarchical transformers; 2) Patch Embedding Alignment (PEA) performs dimensional transformation enables the patch embedding distillation.

Maohui et al. have proposed knowledge distillation in ViTs for semantic segmentation purpose in 2024 [70]. In this paper, authors match the output queries of the student and teacher models to enable a query-based knowledge distillation scheme. Using this approach it has been possible to perform knowledge distillation where the student models can have a lower number of queries and the backbone can be changed from a Transformer architecture to a convolutional neural network architecture. Feature-based method known as ViTKD, which mimics the shallow layers and generates the deep layer in the teacher is proposed by Yang et al. [71]. ViTKD leads to consistent and significant improvements in the students.

Omar et al. [72] have proposed Hybrid Data-efficient Knowledge Distillation (HDKD) model. Here student is hybrid model and teacher is CNN model. Hybrid student leverages the strengths of both convolutions and transformers and shares the convolutional structure with the teacher model. This enables the direct application of feature distillation without losing any information or any kind of computational burden.

There are two major challenges in applying KD on ViTs: Positional encoding and lack of inductive bias in ViTs.

- **Positional Encoding in ViTs**: It provides order and type of information about input tokens. ViTs do not have inherent spatial locality like in CNN. Hence ViTs rely on positional encoding.
- **Lack of inductive bias in ViTs**: Like in CNN, ViTs don't have inductive bias. It makes their generalization ability poor and learning ability lesser when model is trained on less amount of data.

V. CHALLENGES AND FUTURE RESEARCH DIRECTIONS

Deployment of ViTs on resource constrained devices is a big challenge due to heavy computational load in ViTs that incurs due to self-attention mechanism. Now pruning, quantization, knowledge distillation and compact design architecture make it possible to deploy these models on edge devices but some areas need to be explored for real world applications to avoid complexities. Pruning reduces number of weights by eliminating less important weights. It makes model sparse. But it does not always lead to less memory usage or faster computation on GPU hardware. Because GPUs are meant to run dense matrix operations faster. For sparse operations GPU hardware needs to specially optimized which is rare. Hence frameworks with reconfigurations in both hardware and software is required so that it will be able to tackle the challenge of sparsity.

Therefore, developing a training-efficient or finetune free pruning techniques may be a possible solution and need more attention in the near future for efficient deployment on the edge. Automating the model design by applying techniques of pruning, quantization with the help of Neural Architecture Search (NAS) [73] that uses reinforcement learning, can be a possible path for further research.

VI. CONCLUSION

This paper surveys about how vision transformers can be deployed on edge devices, what are the different challenges posed while using such computation intensive models on edge devices and how to overcome these challenges.

Compact architectures are designed from scratch such that ViT can be used on edge devices with fewer parameters and less computational cost. Pruning eliminates unwanted weights, neurons or attention heads from trained model of ViT to reduce model size and computational cost without affecting the accuracy of the model. Quantization reduces 32-bit floating-point operations to 8-bit integer operations. It speeds up the computations and reduces memory requirement of ViT. In knowledge distillation, smaller model is trained using larger (parent) model with comparatively fewer parameters. Mostly used parameters to evaluate the performance of ViT models include number of trainable parameters, accuracy and FLOPs. Each of these four methods results in reduction in trainable parameters, computational cost and memory usage while maintaining the acceptable accuracy of the model. Maintaining trade-off between performance of the model and compression of the model is very crucial for real world applications.

To handle these challenges, optimized hardware design, reconfiguring hardware-software to make optimum use of model for diverse applications, automating the model design by neural architecture search are possible directions for future research.

REFERENCES

[1] Krizhevsky A., Sutskever I., Hinton G.E., “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, 2012, pp. 1097–1105.

[2] Karpathy A., Toderici G., Shetty S., Leung T., Sukthankar R., Fei-Fei L, “Largescale video classification with convolutional neural networks,” Computer Vision and Pattern Recognition (CVPR), pp. 1725–1732, 2014.

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” California University, La Jolla Institute for Cognitive Science, Technical Report, pp. 318-362, 1985.

[4] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, “ImageNet classification with deep convolutional neural networks,” Communications of the ACM, Vol. 60, No. 6, pp. 84–90, 2017.

[6] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” IEEE Transactions on Knowledge and Data Engineering, Vol. 22, no. 10, pp. 1345-1359, 2010.

[7] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” International Conference on Learning Representations, pp. 1-15, 2014.

[8] A. Vaswani et al., “Attention is all you need,” In Proceedings of International Conference on Neural Information Processing Systems, pp. 6000–6010, 2017.

[9] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He, “Non-local neural networks,” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7794-7803, 2018.

[10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, “End-to-end object detection with transformers,” European conference on computer vision, pp. 213-229, 2020.

[11] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens, “Stand-alone self-attention in vision models,” Advances in neural information processing systems 32, pp. 1-13, 2019.

[12] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen, “Axial-deeplab: Stand-alone axial-attention for panoptic segmentation,” European conference on computer vision, pp. 108-126, 2020.

[13] Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit and Neil Houlsby, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.” International Conference on Learning Representation, 2020.

[14] Wu, Kan, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan, “Tinyvit: Fast pretraining distillation for small vision transformers,” European conference on computer vision, pp. 68-85, 2022.

[16] Jamil, Sonain, Muhammad Sohail Abbas, and Arunabha M. Roy, “Distinguishing Malicious Drones Using Vision Transformer” AI, Vol.3, No. 2, pp. 260-273, 2022.

[16] Barman Utpal, Parismita Sarma, Mirzanur Rahman, Vaskar Deka, Swati Lahkar, Vaishali Sharma, and Manob Jyoti Saikia, “ViT-SmartAgri: Vision Transformer and Smartphone-Based Plant Disease Detection for Smart Agriculture,” Agronomy, Vol. 14, No. 2, pp. 1-14.

[17] T. Saranya, C. Deisy, S. Sridevi, “Efficient agricultural pest classification using vision transformer with hybrid pooled multihead attention,” Computers in Biology and Medicine, Volume 177, Issue C, 2024.

[18] Bi Luning, Wally Owen, Hu Guiping, Tenuta Albert U, Kandel Yuba R, Mueller Daren S., “A transformer-based approach for early prediction of soybean yield using time-series images”, Frontiers in Plant Science, Vol.14, 2023.

[19] Liangshan Lou, Ke Lu, Jian Xue, “Multi-Scale Vision Transformer for Defect Object Detection,” Procedia Computer Science, Vol. 222, pp. 397-406, 2023.

[20] Bayat, Nasrin, Jong-Hwan Kim, Renoa Choudhury, Ibrahim F. Kadhim, Zubaidah Al-Mashhadani, Mark Aldritz Dela Virgen, Reuben Latorre, Ricardo De La Paz, and Joon-Hyuk Park, “Vision Transformer Customized for Environment Detection and Collision Prediction to Assist the Visually Impaired,” Journal of Imaging, Vol. 9, No. 8, pp. 1-18, 2023.

[21] B. Zhuang, J. Liu, Z. Pan, H. He, Y. Weng, and C. Shen, “A survey on efficient training of transformers,” Proceedings of 32nd International Joint Conference on Artificial Intelligence, pp. 6823–6831, 2023.

[22] W. Wang et al., "Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions,” International Conference on Computer Vision (ICCV), pp. 548-558, 2021.

[23] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proceedings of IEEE/CVF International Conference on Computer Vision, pp. 10012–10022, 2021.

[24] J. Lu et al., “SOFT: Softmax-free transformer with linear complexity,” in Proceedings of International Conference on Neural Information Processing System, pp. 21297–21309, 2021.

[25] S. A. Koohpayegani and H. Pirsiavash, “SimA: Simple softmax-free attention for vision transformers,” Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2607-2617, 2024.

[28] Mehta, Sachin, and Mohammad Rastegari, “Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer,” International Conference on Learning Representations 2022.

[27] H. Cai, J. Li, M. Hu, C. Gan, and S. Han, “EfficientViT: Lightweight multiscale attention for high-resolution dense prediction,” in Proceedings of IEEE/CVF International Conference on Computer Vision, pp. 17 302–17 313, 2023.

[28] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, “Transformers are RNNs: Fast autoregressive transformers with linear attention,” in Proceedings of International Conference on Machine Learning, pp. 5156–5165, 2020.

[29] Novendra Setyawan, Chi-Chia Sun, Mao-Hsiu Hsu, Wen-Kai Kuo, Jun-Wei Hsieh, “MicroViT: A Vision Transformer with Low Complexity Self Attention for Edge Device,” To appear at IEEE International Symposium on Circuits and Systems (ISCAS), May 25-28, 2025, London, UK.

[30] Hu, C., Guo, J., Xie, H. et al., "RJ-TinyViT: an efficient vision transformer for red jujube defect classification," Scientific Reports, Vol. 14, Article No. 27776, 2024.

[31] Frankle, Jonathan, and Michael Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks." arXiv preprint arXiv:1803.03635, 2018.

[35] Han, Song, Jeff Pool, John Tran, and William Dally, "Learning both weights and connections for efficient neural network," Proceedings of 29th, International Conference on Advances in neural information processing systems, Vol. 1, pp. 1135-1143, 2015.

[33] Ramanujan, Vivek, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari, "What's hidden in a randomly weighted neural network?" In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 11893-11902, 2020.

[34] Zhu, Mingjian, Yehui Tang, and Kai Han, "Vision transformer pruning," arXiv preprint, arXiv:2104.08500, 2021.

[35] Yu, Lu, and Wei Xiang, "X-pruner: explainable pruning for vision transformers," In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 24355-24363, 2023.

[36] Xu, Kaixin, Zhe Wang, Chunyun Chen, Xue Geng, Jie Lin, Xulei Yang, Min Wu, Xiaoli Li, and Weisi Lin, "Lpvit: Low-power semi-structured pruning for vision transformers," European Conference on Computer Vision, pp. 269-287, 2024.

[37] Shi Dachuan, Chaofan Tao, Ying Jin, Zhendong Yang, Chun Yuan, and Jiaqi Wang, "UPop: Unified and progressive pruning for compressing vision-language transformers," International Conference on Machine Learning, pp. 31292-31311, 2023.

[38] Prasetyo Yogi, Novanto Yudistira, and Agus Wahyu Widodo, "Sparse then prune: Toward efficient vision transformers," arXiv preprint, arXiv:2307.11988, 2023.

[39] Ilhan Fatih, Gong Su, Selim Furkan Tekin, Tiansheng Huang, Sihao Hu, and Ling Liu, "Resource-efficient transformer pruning for finetuning of large models," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16206-16215, 2024.

[40] Riaz Hamza, and Alan F. Smeaton. "The Effects of Grouped Structural Global Pruning of Vision Transformers on Domain Generalization." arXiv preprint arXiv:2504.04196, 2025.

[41] Cao Yuhao. "FCP_DIS_ViT: Efficient Vision Transformer with Neural Network Pruning." In 2024 IEEE 4th International Conference on Power, Electronics and Computer Applications (ICPECA), pp. 1216-1221. IEEE, 2024.

[42] Shui Xiuying, "A Unified Method for Patch Pruning in Vision Transformers to Achieve Efficient Model Deployment", hal-05011879 2025.

[43] Marchetti, Michele, Davide Traini, Domenico Ursino, and Luca Virgili. "Efficient token pruning in Vision Transformers using an attention-based Multilayer Network." Expert Systems with Applications, Volume 279: 127449, 2025.

[44] He, Yang, and Joey Tianyi Zhou. "Data-independent Module-aware Pruning for Hierarchical Vision Transformers." In The Twelfth International Conference on Learning Representations, 2024.

[45] Yang, Guang, Xinming Yan, Hui Kou, Zihan Zou, Qingwen Wei, Hao Cai, and Bo Liu. "TWDP: A Vision Transformer Accelerator with Token-Weight Dual-Pruning Strategy for Edge Device Deployment." In Proceedings of the 30th Asia and South Pacific Design Automation Conference, pp. 177-182. 2025.

[46] Ishibashi, Ryuto, and Lin Meng. "Automatic pruning rate adjustment for dynamic token reduction in vision transformer." Applied Intelligence Vol. 55, no. 5, pp 1-15, 2025.

[47] Gholami A., Kim S., Dong Z., Yao Z., Mahoney M.W., Keutzer K, "A survey of quantization methods for efficient neural network inference," CoRR abs/2103.13630, 2021.

[48] Ron Banner, Yury Nahshan, and Daniel Soudry, "Post training 4-bit quantization of convolutional networks for rapid-deployment," Proceedings of the 33rd International Conference on Neural Information Processing Systems, Article 714, pp. 7950–7958, 2019.

[49] Liu Z., Wang Y., Han K., Zhang W., Ma S., Gao W, "Post-training quantization for vision transformer," Proceedings of the 35th International Conference on Neural Information Processing Systems, 2021, pp. 28092–28103.

[50] Yuan Zhihang et al., "PTQ4ViT: Post-training Quantization for Vision Transformers with Twin Uniform Quantization." European Conference on Computer Vision, pp. 191-207, 2022.

[51] Li Y., Gong R., Tan X., Yang Y., Hu P., Zhang Q., Yu F., Wang W., Gu S, "BRECQ: pushing the limit of post-training quantization by block reconstruction," 9th International Conference on Learning Representations, ICLR 2021.

[52] Li Zhikai, and Qingyi Gu, "I-vit: Integer-only quantization for efficient vision transformer inference," Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 17019-17029, 2023.

[53] Shi H., Cheng X., Mao W., & Wang Z, "P^2-ViT: Power-of-Two Post-Training Quantization and Acceleration for Fully Quantized Vision Transformer," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume 32, Issue 9, pp. 1704-1717, 2024.

[54] Y. Liu, H. Yang, Z. Dong, K. Keutzer, L. Du, and S. Zhang, "NoisyQuant: Noisy bias-enhanced post-training activation quantization for vision transformers," in Proceedings of IEEE/CVF Conference on Computer Vision Pattern Recognition, pp. 20 321–20 330, 2023.

[55] Z. Lit et al., "Auto-ViT-Acc: An FPGA-aware automatic acceleration framework for vision transformer with mixed-scheme quantization," in Proceedings of 32nd International Conference on Field-Programmable Logic and Applications IEEE, pp. 109–116, 2022,

[56] Jiang Runqing, Ye Zhang, Longguang Wang, Pengpeng Yu, and Yulan Guo. "AIQViT: Architecture-Informed Post-Training Quantization for Vision Transformers." arXiv preprint arXiv:2502.04628, 2025.

[57] Wu Zhuguanyu, Zhang Jiayi, Chen Jiaxin, Guo Jinyang, Huang Di, Wang Yunhong, "APHQ-ViT: Post-Training Quantization with Average Perturbation Hessian Based Reconstruction for Vision Transformers", In proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2025.

[58] Khadka R., Sengupta P., G. Lind P., Yazidi A. "Quantization of Vision Transformer-Based Model for Real-Time EEG Classification." Communications in Computer and Information Science, Springer, vol 2398. pp 17-27, 2025.

[59] Choi J., Wang Z., Venkataramani S., Chuang P.I., Srinivasan V., Gopalakrishnan K, "PACT: parameterized clipping activation for quantized neural networks," arxiv preprint, 2018.

[60] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531, 2015.

[61] Liu Y., Chen K., Liu C., Qin Z., Luo Z., and Wang, J., "Structured knowledge distillation for semantic segmentation," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2604-2613, 2019.

[62] S. Xie, H. Wang, B. Yu, K. Chang, X. Liang, and Wang, G, "Knowledge Distillation via the Target-aware Transformer," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10915-10924, 2022.

[63] Ren S., Gao Z., Hua T., Xue Z., Tian Y., He S., and Zhao H, “Co-advise: Cross inductive bias distillation,” In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16773-16782, 2022.

[64] Touvron H., Cord M., Douze M., Massa F., Sablayrolles A. and Jégou H, “Training data-efficient image transformers & distillation through attention,” International Conference on Machine Learning, pp. 10347-10357, 2021.

[65] Yu S., Chen, T. Shen J., Yuan H., Tan J., Yang S., and Wang Z, “Unified visual transformer compression,” International Conference on Learning Representations., arXiv:2203.08243, 2022.

[84] Liu Y., Cao J., Li B., Hu W., Ding J. and Li L, “Cross-Architecture Knowledge Distillation,” International Journal of Computer Vision, Vol. 132, pp. 2798–2824, 2024.

[67] Xiao T., Singh M., Mintun E., Darrell T., Dollár P. and Girshick, R, “Early convolutions help transformers see better,” Advances in Neural Information Processing Systems, Vol. 34, pp. 30392-30400, 2021.

[68] Lin H., Han G., Ma J., Huang S., Lin X. and Chang S. F, “Supervised masked knowledge distillation for few-shot transformers,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19649-19659, 2023.

[69] Liu Ruiping, Kailun Yang, Alina Roitberg, Jiaming Zhang, Kunyu Peng, Huayao Liu, Yaonan Wang, and Rainer Stiefelhagen, "TransKD: Transformer knowledge distillation for efficient semantic segmentation," IEEE Transactions on Intelligent Transportation Systems, Vol. 25, No. 12, pp. 20933-20949, 2024.

[70] Li Maohui, Michael Halstead and Chris McCool, “Knowledge distillation for efficient instance semantic segmentation with transformers,” In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5432-5439, 2024.

[71] Yang Zhendong, Zhe Li, Ailing Zeng, Zexian Li, Chun Yuan and Yu Li, “ViTKD: Feature-based knowledge distillation for vision transformers,” In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1379-1388, 2024.

[72] EL-Assiouti, Omar S., Ghada Hamed, Dina Khattab and Hala M. Ebied, “HDKD: Hybrid data-efficient knowledge distillation network for medical image classification,” Engineering Applications of Artificial Intelligence, Vol. 138, Issue PB, 2024.

[73] H. Cai, L. Zhu, S. Han, “ProxylessNAS: Direct neural architecture search on target task and hardware”, International Conference on Learning Representations, 2019.