Optimising ViT for Edge Deployment: Hybrid Token Reduction for Efficient Semantic Segmentation

Mathilde Proust[1], Martyna Poreba[1], Calvin Galagain[1,2], Michal Szczepanski[1], Karim Haroun[1,3]

[1] University Paris-Saclay, CEA, List, France

[2] ENSTA Paris, Institut Polytechnique de Paris, France

[3] University Côte d’Azur, France

Abstract

Vision Transformers (ViTs) achieve high accuracy in multiple vision-related tasks; however, substantial computational and memory demands limit their deployment on resource-constrained edge devices. ViTs process images by splitting them into uniform patches, treating each patch as a separate token. Since not all regions are equally important—detailed areas may require more tokens, while broader regions need fewer optimizing token processing is considered essential for improving efficiency. To enhance computational performance, a hybrid token reduction approach is implemented, integrating token merging and pruning strategies within MMSegmentation, a widely used open-source semantic segmentation toolbox. The strengths of CTS, which merges semantically similar and adjacent patches using a CNN-based policy network, and DToP, which halts the processing of tokens that can be predicted with sufficient accuracy in the early layers of the network, are combined in this method. A reduction in computational complexity of up to 2× is shown by the experimental results, with only an approximate 1% drop in accuracy observed on the NVIDIA Jetson AGX Orin 64GB. Exporting a pruned PyTorch model to TensorRT remains a challenging task that requires considerable effort. The difficulties involved are emphasized, and additional work needed to achieve full compatibility with ONNX export standards is outlined.

Keywords Vision transformer, semantic segmentation, token reduction, token merging, model optimization, computational efficiency, computational complexity, edge device.

List of Notations and Abbreviations

ViT: Vision Transformer

mmseg: MMSegmentation, an open-source semantic segmentation toolbox

DToP: Dynamic Token Pruning

CTS: Content-aware Token Sharing

mIoU: Mean Intersection Over Union

FPS: Frames Per Second

1.1 Introduction and Background

Vision Transformers (ViTs) have achieved outstanding results in various vision tasks, but their substantial computational and memory requirements pose major obstacles to deployment on resource-constrained edge devices. A combination of software and hardware innovations has emerged to tackle these challenges, focusing on reducing computational complexity, memory consumption, and improving power efficiency. For example, ViTA [1] introduces a dedicated hardware accelerator that optimises ViT inference for real-time applications on edge devices, reducing computational overhead and enhancing efficiency. Another approach [2] utilises an integer-only systolic array accelerator to minimise power consumption and computational demands. Additionally, the ME-ViT accelerator [3] offers a memory-efficient FPGA-based solution that optimises data flow and storage, lowering memory usage and power consumption. The 109-GOPs/W FPGA-based accelerator [4] marks significant progress by incorporating a weighted data flow mechanism that minimizes energy consumption. This approach prioritizes data reuse, optimizing resource efficiency and reducing power usage. On the other hand, researchers have explored various optimization techniques, including quantization, distillation, and pruning, to bridge the gap between the high performance of ViTs and the constraints of edge environments, making them more practical for resource- limited settings. For instance, MobileViT [5] introduces a variant of ViTs that merges convolutional neural networks (CNNs) with transformers, resulting in a lightweight model that maintains high accuracy while being suitable for mobile and edge devices. TinyViT [6] employs knowledge distillation to create a smaller, more efficient transformer model that retains high performance, making it ideal for edge applications. Similarly, EdgeViTs [7] are specifically designed for edge devices, incorporating optimized attention mechanisms and downsampling strategies.

ViTs typically generate visual patches by splitting an image into a uniform, fixed grid, where each grid cell is treated as a distinct token. Though straightforward, this approach overlooks the varying complexity of image content, as certain regions can be represented with fewer tokens due to their homogeneity. For example, in an image depicting a busy street, tasks such as identifying vehicles and pedestrians may necessitate a higher density of tokens. In contrast, broader areas of the image, such as the sidewalk or the sky, may require significantly fewer tokens. This disparity in token necessity raises an important question: is it truly essential to process such a large number of tokens at every layer of the network? Given that the computational complexity of ViT scales quadratically with the length of input sequences, a reduction in the number of tokens presents a viable strategy for decreasing computational costs. By intelligently selecting and utilising tokens based on their relevance to the task, performance can be optimised while simultaneously reducing the resource demands on the model.

In this context, our work introduces a hybrid token reduction mechanism aimed at enhancing the efficiency of ViTs for semantic segmentation tasks. This method integrates two cutting-edge techniques: patch merging and early-pruning. A class-agnostic CNN-based network, trained independently from the ViT, merges semantically similar and adjacent patches, while early-pruning stops the processing of tokens that can be confidently predicted in the early layers, reducing unnecessary computations. We implement this method with semantic segmentation transformer models, specifically ViT-Base and ViT-Tiny, and perform experiments on the NVIDIA Jetson AGX Orin 64GB platform.

1.2 Related Work

Token reduction techniques are generally tailored to the specific task they address. State-of-the-art methods predominantly focus on classification. In this case, token pruning methods often permanently eliminate tokens, as they no longer affect the outcome. However, in dense prediction tasks like semantic segmentation, patches cannot be completely discarded, as each one plays a role in the pixel-level predictions needed for detailed results. For such tasks, ViTs handle a large number of tokens, where both the size and number of tokens must be carefully selected to preserve essential details while minimizing computational complexity. Given the demands of dense prediction tasks, not all token reduction methods are suitable, with merging techniques generally proving more effective than pruning approaches. Unlike pruning, which irreversibly discards tokens and risks losing critical information, merging aggregates similar patches, retaining essential details. This approach allows the model to maintain accuracy while reducing computational complexity by carefully selecting which tokens to combine based on their relevance, thereby providing the flexibility needed to adapt to the complexities of different image content. Among the token reduction methods extended to support dense prediction tasks is DynamicViT [8], [9] that employs a dynamic token selection mechanism. Similarly, ToFu [10] has produced notable results in image generation tasks, highlighting its potential in areas requiring dense, detailed predictions. The authors of TCFormer [11] propose their method as a general solution applicable to a wide range of vision tasks, such as object detection and semantic segmentation. Nonetheless, TCFormer faces a major drawback: the computational complexity of its KNN-DPC algorithm increases quadratically with the number of tokens, which undermines its efficiency, especially when handling high-resolution images.

To the best of our knowledge, only three token reduction methods have been specifically designed for the segmentation. One such approach is Content-aware Token Sharing (CTS) [12], which introduces a class-agnostic policy model using a CNN network trained separately from the ViT. CTS identifies whether adjacent image patches belong to the same semantic class; if they do, they can share a common token. This is achieved through binary classification to form fixed-size groups of patches within the input image, ensuring spatial coherence while eliminating the need to process unnecessary tokens. Another approach, Dynamic Token Pruning (DToP) [13], enables early-pruning for tokens, allowing simpler tokens to complete their predictions earlier in the network. DToP divides the transformer into distinct stages and utilizes auxiliary blocks for early prediction generation. It also incorporates the attention-to-mask (ATM) module [14] as the segmentation head, which improves its efficiency in handling dense, pixel-level predictions. Finally, SVIT [15] introduces an innovative method that utilizes a lightweight two-layer MLP (Multi-Layer Perceptron) to dynamically select tokens for processing within the transformer block. One of its key features is that it prunes tokens while retaining them in feature maps, enabling their reactivation in later layers. This ensures that important information is preserved, even if some tokens are not processed in the early stages of the network.

1.3 Methodology

Re-evaluating the traditional fixed-grid approach in ViTs paves the door to more efficient architectures that can handle diverse visual tasks with greater precision and reduced computational overhead. In the vast majority of images, there exist homogeneous regions where it is unnecessary to process redundant patches separately. By minimizing the number of input patches, we can reduce the total number of tokens handled by the ViT blocks. This approach helps prevent the system from expending resources on superfluous tokens, leading to lower energy consumption. This drives our investigation into improving the efficiency of ViTs through a token merging and pruning strategy tailored for inference on edge devices, specifically aimed at enhancing performance in semantic segmentation. Our method integrates the strengths of two state-of-the-art techniques: content-aware patch merging through CTS and early token pruning via DToP. Figure 1.1 outlines the proposed hybrid token optimization mechanism. Tokenization initiates the process, dividing the image into a regular grid of patches. To minimise the number of patches that need processing, we utilise a class-agnostic CNN network to merge neighboring, semantically similar patches. Next, the token-sharing module transforms these non-uniform size patches into tokens Zi using a linear embedding function as follows:

$$Z_i = f_{embed}(P_i) \tag{1.1}$$

where Pi represents the group of patches obtained from the image, in which each patch $pi \in P$ is defined as a sub-region of the image, and $fembed$ (.) the embedding function that maps into supertokens Zi.

As in DToP, the ViT backbone is organised into M stages, with auxiliary heads identifying high-confidence tokens that are masked and excluded from further calculations. Let C denote the set of high-confidence tokens, where each token is determined by a confidence score $ck \in C$:

$$c_k = f(z_j) \;\; if \;\; confidence(z_j) > threshold \tag{1.2}$$

This operation is performed on carefully selected layers, specifically after a certain number of transformer blocks. Finally, the model processes the remaining tokens to generate the final output through per-token predictions.

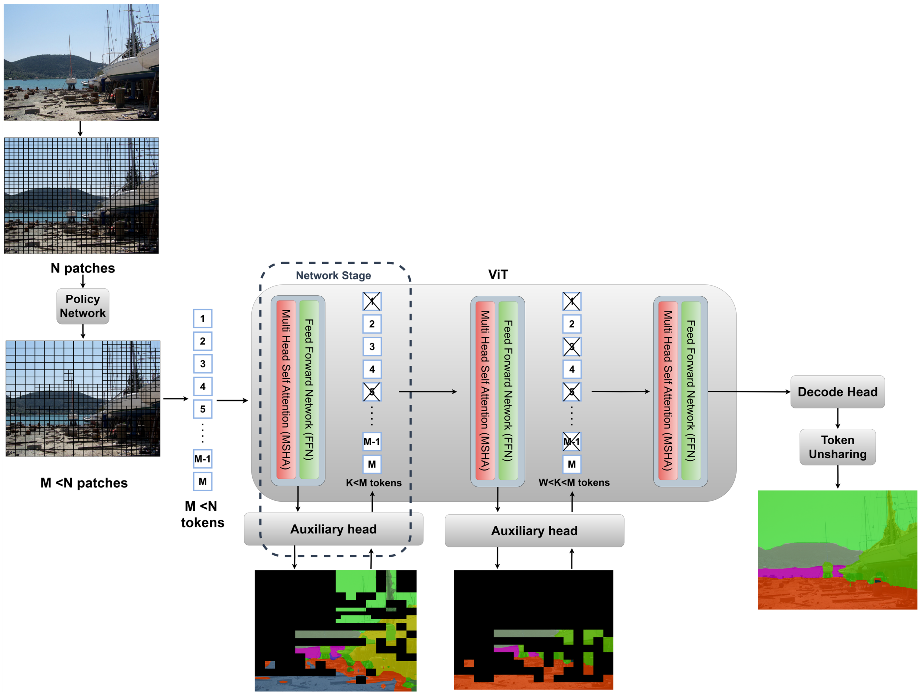

Figure 1.1 Outline of the Proposed Hybrid Token Optimization Technique

1.3.1 Content-aware Patch Merging

To apply the CTS method to any conventional transformer-based model, it is necessary to incorporate a token sharing function Zi, a token unsharing function, and a policy model. The class-agnostic policy network determines which patches are eligible to share a token prior to their entry into the ViT. It focuses on grouping only square neighboring regions, facilitating the seamless reassembly of tokens at the output of the ViT backbone. CTS comes with a lightweight CNN network to generate probability scores for each 2×2 patch group. It is based on the EfficientNetLite0 model [17], pre-trained on ImageNet-1K [18]. This model predicts a similarity score S for a window of n patches $\omega j = \{p1, p2, \dots pn\}$:

$$S = \sigma(W^T(\omega)) \tag{1.3}$$

where Wp is the learned weight matrix of the policy network and $\sigma(.\,)$ is the sigmoid activation function.

Finally, only the top 103 patch windows ωj are merged into 2x2 groups, based on the highest-ranked probabilities. As a result, the number of patches that are converted into tokens is significantly reduced (Figure 1.2). For example, a 512×512 resolution input image traditionally produces 32×32 patches, with each patch covering 16×16 pixels, resulting in 1 024 patches to process. After applying the CTS method, only 715 patches are sent to ViT, reducing the number of tokens by 30%.

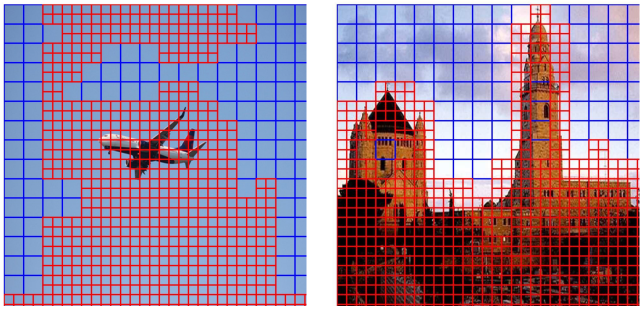

Figure 1.2 Results of Patch merging: grouped patches in blue, individual patches in red

1.3.2 Early-Pruning

The core concept of DToP is to identify easy, high-confident tokens in the intermediate layers and exclude them from further computations. After a predetermined number of attention block layers, the model directs tokens to an auxiliary segmentation head, which adapts the ATM, and applies a stopping criterion based on the confidence of its predictions. Specifically, at stage M, a confidence score $c(m)$ is calculated for each token Zi, which is formalized as follows:

$$Z(m+1) = \{zi \mid c(m) < \theta\} \tag{1.3}$$

where $Z(m+1)$ represents the set of tokens passed to the next stage. Tokens with confidence scores exceeding a predefined threshold θ are classified as high-confidence tokens and are discarded, while the low-confidence tokens proceed further through the network. This underscores the significance of strategically positioning auxiliary heads within the network. Placing them too early could make it difficult for the model to accurately predict the class of any tokens. We adopt the recommendations from the original DToP paper concerning hyperparameters and the positioning of auxiliary heads, acknowledging that they may not be optimal in all scenarios.

1.4 Experiments

We integrate our hybrid token reduction mechanism into the SegViT semantic segmentation framework [14], which serves as the baseline for our performance comparison study. All experiments are performed using the MMSegmentation (mmseg) toolbox [19], which allows for easy customization of models by combining different backbones. We integrate ViT-Base, which includes 12 encoder layers, a 768-dimensional hidden layer, and 12 attention heads, alongside ViT-Tiny, which features 12 encoder layers, a 192-dimensional hidden layer, and 3 attention heads. Both process images by dividing them into 16×16 pixel patches. We follow the standard training settings in mmseg and use the same hyperparameters as the original papers. For DToP, we adopt the configuration recommended by the authors, and split the ViT backbone into three stages with token pruning occurring at the 6th and 8th layers for ViT-Base. This setup is intended to achieve an effective balance between computational cost and segmentation accuracy. Additionally, we choose to examine a model divided into two stages and position the pruning head after the 8th layer. Since the authors did not provide configurations for ViT-Tiny, we applied the same configuration as ViT-Base, as ViT-Tiny contains the same number of blocks. Experiments are conducted on ADE20k [20], a dataset focused on semantic segmentation. Mean Intersection over Union (mIoU) assesses segmentation accuracy, while giga floating-point operations (GFLOPs), measured with fvcore package [16], reflect model complexity, and frames per second (FPS) indicates throughput.

For inference on the NVIDIA Jetson AGX Orin 64GB, we primarily use PyTorch because of its flexibility and ease of use during model development. To optimize performance and fully leverage the hardware capabilities of the NVIDIA Jetson platform, TensorRT is the preferred option. However, we encountered several challenges when exporting pruned models to ONNX and TensorRT. While PyTorch 2.4 supports all necessary layers, it presents compatibility issues with the OpenMMLab libraries. Specifically, the mmseg framework, which depends on MMCV (a foundational library for computer vision tasks) and MMEngine (a runtime engine for managing training, validation, and inference loops), complicates cross-compilation with the latest Python and the preferred CUDA version. Although we ultimately succeeded in validating the ONNX export, TensorRT indicated a size mismatch in one of the backbone layers. It appears that a specific layer contains parameters not supported by TensorRT, necessitating further investigation to find a solution.

Table 1.1 and Table 1.2 summarise the performance achieved with the model in FP32 format. The results show that integrating our hybrid token reduction method into SegViT allows us to maintain a comparable mIoU, with segmentation accuracy loss kept within a maximum of 1%. This method achieves a reduction in complexity of up to 45% for ViT-Base and 42% for ViT-Tiny. By applying only the token merging via CTS, we observe a reduction in computational complexity for ViT-Base and ViT-Tiny of 33% and 37%, respectively. The early-pruning technique via DToP impacts both computational complexity and inference speed, with the number of auxiliary heads playing a crucial role. Although placing the pruning heads at the 6th and 8th positions yields a 23% reduction in GFLOPs for

ViT-Base. This advantage comes at the expense of increased inference time, which can slow the process down by nearly a factor of two.

Table 1.1 Performance of Token Reduction Method integrated with ViT-Base

Method	mIoU [%]	GFLOPs	FPS
SegViT	48.3	112.8	6.8
+CTS	47.8	75.4	13.3
+DToP@[6,8]	46.1	86.3	3.9
+CTS&DToP@[6,8](ours)*	47.2	63.0	12.7
+CTS&DToP@[6,8](ours)	47.7	62.1	4.5
+CTS&DToP@[8](ours)	48.3	68.3	6.5

**on a single A100GPU*

Table 1.2 Performance of Token Reduction Method integrated with ViT-Tiny

Method	mIoU [%] e	GFLOPs	FPS
SegViT	37.8	12.0	15.6
+CTS	37.3	7.6	15.4
+DToP@[6,8]	38.0	9.9	8.2
+CTS&DToP@[6,8](ours)*	37.7	6.8	19.0
+CTS&DToP@[6,8](ours)	37.7	6.8	9.3
+CTS&DToP@[8](ours)	38.5	6.9	13.1

**on a single A100GPU*

Figure 1.3 illustrates the inference time for each layer of the model, including the auxiliary heads used for pruning. It shows that pruning tokens with segmentation heads equipped with ATM modules tends to be excessively slow, underscoring the need for future work to focus on optimization. Given this observation, a single auxiliary head presents the best trade-off between reducing complexity and time inference.

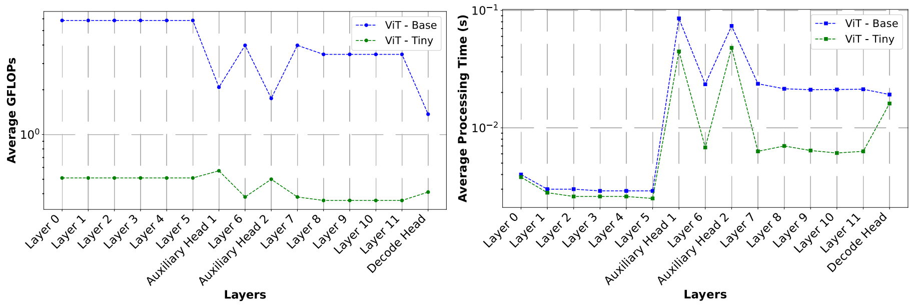

Figure 1.3 Layer-by-layer analysis considering GFLOPs and Throughput (FPS) for pruning heads placed at positions 6 and 8.

Figure 1.4 and Figure 1.5 display visualized predictions, where the number of pruned tokens increases from bottom to top. In "easy" samples, most tokens are pruned after the 6th ViT block, while in "hard" cases, the majority of tokens are retained until the final layer. The second auxiliary head (at the 8th layer) was often unable to prune a significant number of tokens, as it was placed too soon after the first head. This highlights that using two pruning heads in smaller networks like ViT-Base and ViT-Tiny is not always necessary or effective.

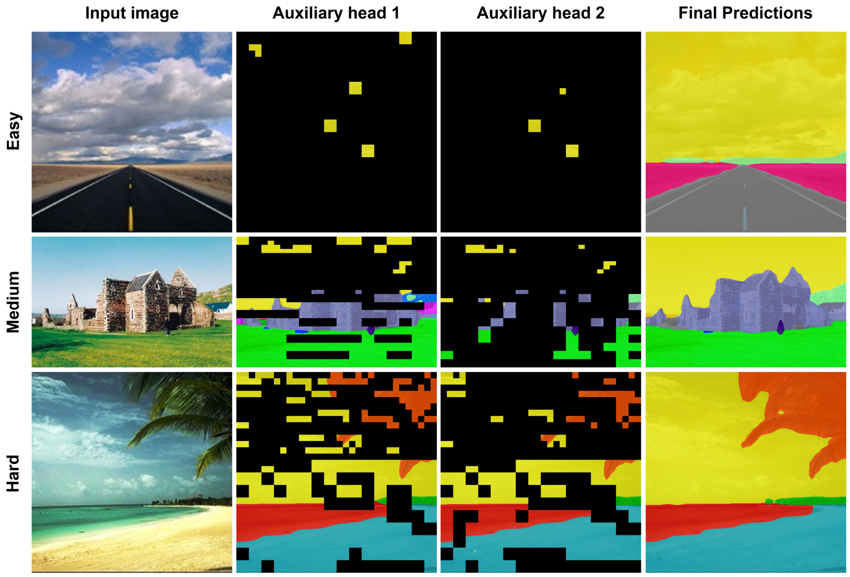

Figure 1.4 ViT-Base segmentation results with pruned tokens masked in black

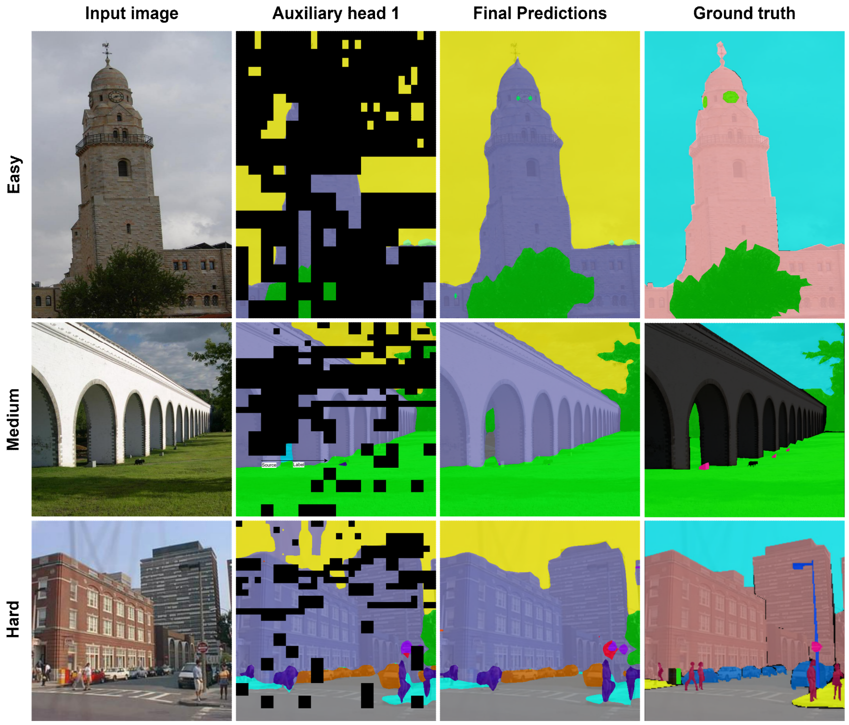

Figure 1.5 ViT-Tiny segmentation results with pruned tokens masked in black

1.5 Conclusion

We introduced a hybrid token optimization mechanism specifically designed for semantic segmentation, which merges semantically similar neighboring patches and incorporates dynamic token pruning based on an early-pruning strategy. Implementing our on-the-fly pruning approach significantly influences architectural design, requiring careful attention to resource allocation and dynamic token management. Nevertheless, proposed token reduction mechanism can seamlessly transition to a fixed-token strategy. By simply fixing the number of top-k most confident tokens pruned by each auxiliary head, rather than relying on the threshold θ, we unlock several advantages. This streamlines hardware design by providing predictable resource allocation and optimizing performance. It also enhances memory management, improves scalability, minimizes overflow risks, and enables parallel processing. Our token reduction technique has been integrated into transformer models (ViT-Base and ViT-Tiny) within the mmseg framework. Through experiments conducted on established segmentation benchmark with an NVIDIA Jetson AGX Orin 64GB, we showed that this optimization method can lower computational costs by up to 45% while maintaining accuracy with minimal impact. Nevertheless, while using auxiliary heads to prune high-confidence tokens lowers computational complexity, it significantly affects inference speed. We suggest that future work concentrate on exploring methods to optimize the architecture of auxiliary heads. Despite its advantages, the complex mmseg framework and the dynamic pruning can complicate model export, as both ONNX and TensorRT require a consistent model structure. Future work will tackle these challenges, aiming to create a more seamless and efficient export pipeline. Efforts will focus on verifying the compatibility of the pruned models with TensorRT and ensuring consistent shapes for all inputs to conditional layers. This may involve modifying the mmseg framework to include shape-alignment operations or developing custom ONNX operations to address shape mismatches.

Acknowledgements

This work was conducted as part of the AI optimization roadmap within the LIAE laboratory of the DSCIN department at CEA-LIST.

Figures and Tables Caption List

References

[1] S. Nag, G. Datta, S. Kundu, N. Chandrachoodan and P. Beerel, “ViTA: A Vision Transformer Inference Accelerator for Edge Applications”, in 2023 IEEE International Symposium on Circuits and Systems (ISCAS), pp.1-5, 2023, https://doi.org/10.1109/ISCAS46773.2023

[2] M. Huang, J. Luo, Ch. Ding, Z. Wei, S. Huang, H. Yu, “An Integer-Only and Group-Vector Systolic Accelerator for Efficiently Mapping Vision Transformer on Edge,” in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 70, no. 12, Dec. 2023, pp. 5289-5301. https://doi.org/10.1109/tcsi.2023.3312775

[3] K. Marino, P. Zhang and V. Prasanna, "ME- ViT: A Single-Load Memory- Efficient FPGA Accelerator for Vision Transformers," in 2023 IEEE 30th International Conference on High Performance Computing, Data, and Analytics (HiPC), Goa, India, 2023 pp. 213-223. https://doi.org/10.1109/HiPC58850.2023.00039

[4] S. Li, C. Chen, L. Yu, X. Wang, and H. Zhang, "A 109-GOPs/W FPGA-based Vision Transformer Accelerator with Weight-Loop Dataflow Featuring Data Reusing and Resource Saving," in Proceedings of the 2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA), 2024, pp. 1-12.

[5] S. Mehta and M. Rastegari, “MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer,” openreview.net, Mar. 04, 2022.

[6] K. Wu et al., “TinyViT: Fast Pretraining Distillation for Small Vision Transformers,” Lecture notes in computer science, pp. 68–85, Jan. 2022, doi: https://doi.org/10.1007/978-3-031-19803-8_5.

[7] J. Pan et al., “EdgeViTs: Competing Light-weight CNNs on Mobile Devices with Vision Transformers.” Accessed: Sep. 16, 2024. Available online:https://www.ecva.net/papers/eccv_2022/papers_ECCV/papers/136710294.pdf

[8] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, and C.-J. Hsieh, “DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification,” arXiv (Cornell University), vol. 34, Dec. 2021.

[9] Y. Rao, Z. Liu, W. Zhao, J. Zhou, and J. Lu, “Dynamic Spatial Sparsification for Efficient Vision Transformers and Convolutional Neural Networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 9, pp. 10883–10897, Apr. 2023, https://doi.org/10.1109/tpami.2023.3263826

[10] M. Kim, S. Gao, Y.-C. Hsu, Y. Shen, and H. Jin, “Token Fusion: Bridging

the Gap between Token Pruning and Token Merging," Jan. 2024, https://doi.org/10.1109/wacv57701.2024.00141.

[11] W. Zeng et al., "Not All Tokens Are Equal: Human-centric Visual Analysis via Token Clustering Transformer," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2022, https://doi.org/10.1109/cvpr52688.2022.01082

[12] C. Lu, D. de Geus and G. Dubbelman, "Content-aware Token Sharing for Efficient Semantic Segmentation with Vision Transformers," in 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023 pp. 23631-23640.

[13] Q. Tang, B. Zhang, J. Liu, F. Liu and Y. Liu, "Dynamic Token Pruning in Plain Vision Transformers for Semantic Segmentation," in 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2023 pp. 777-786.

[14] B. Zhang, et al., 2024. "SegViT: semantic segmentation with plain vision transformers". in Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS '22). Curran Associates Inc., Red Hook, NY, USA, Article 359, pp. 4971–4982.

[15] Y. Liu, M. Gehrig, N. Messikommer, M. Cannici and D. Scaramuzza, "Revisiting Token Pruning for Object Detection and Instance Segmentation," in 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2024 pp. 2646-2656.

[16] facebookresearch, "GitHub - facebookresearch/fvcore: Collection of common code that's shared among different research projects in FAIR computer vision team.," GitHub, 2019. Available online: https://github.com/facebookresearch/fvcore

[17] M. Tan, and Q.V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, 9-15 June 2019, pp. 6105-6114.

[18] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, Apr. 2015, https://doi.org/10.1007/s11263-015-0816-

[19] MMSegmentation Contributors, "OpenMMLab Semantic Segmentation Toolbox and Benchmark," GitHub, Jul. 01, 2020. https://github.com/open-mmlab/mmsegmentation

[20] B. Zhou et al., "Semantic Understanding of Scenes Through the ADE20K Dataset," International Journal of Computer Vision, vol. 127, no. 3, pp. 302–321, Dec. 2018, https://doi.org/10.1007/s11263-018-1140-0