Token Adaptive Vision Transformer with Efficient Deployment for Fine-Grained Image Recognition

Chonghan Lee†, Rita Brugarolas Brufau‡, Ke Ding‡, Vijaykrishnan Narayanan†
†*Dept. of Electrical Engineering and Computer Science, The Pennsylvania State University, State College, USA*
‡*Intel Corporation, Santa Clara, USA*
†{cvl5361, vxn9}@psu.edu
‡{rita.brugarolas.brufau, ke.ding}@intel.com

***Abstract*—Fine-grained Visual Classification (FGVC) aims to distinguish object classes belonging to the same category, e.g., different bird species or models of vehicles. The task is more challenging than ordinary image classification due to the subtle inter-class differences. Recent works proposed deep learning models based on the vision transformer (ViT) architecture with its self-attention mechanism to locate important regions of the objects and derive global information. However, deploying them on resource-restricted internet of things (IoT) devices is challenging due to their intensive computational cost and memory footprint. Energy and power consumption varies in different IoT devices. To improve their inference efficiency, previous approaches require manually designing the model architecture and training a separate model for each computational budget. In this work, we propose Token Adaptive Vision Transformer (TAVT) that dynamically drops out tokens and can be used for various inference scenarios across many IoT devices after training the model once. Our adaptive model can switch among different token drop configurations at run time, providing instant accuracy-efficiency trade-offs. We train a vision transformer with a progressive token pruning scheme, eliminating a large number of redundant tokens in the later layers. We then conduct a multi-objective evolutionary search with the overall number of floating point operations (FLOPs) as its efficiency constraint that could be translated to energy consumption and power to find the token pruning schemes that maximize accuracy and efficiency under various computational budgets. Empirical results show that our proposed TAVT dramatically speeds up the GPU inference latency by up to 10× and reduces memory requirements and FLOPs by up to 5.5 × and 13× respectively while achieving competitive accuracy compared to prior ViT-based state-of-the-art approaches.**

I. Introduction

Recognizing discriminative local parts and features from objects with subtle visual variation plays a crucial role in FGVC task. Recently, inspired by successes in Natural Language Processing tasks [1]–[3], transformer models have been introduced into the computer vision domain and demonstrated high performance in various vision tasks [4], [5]. Specifically, some works have proposed ViT-based models for FGVC and resulted in superior performance compared to traditional Convolutional Neural Network approaches [6]–[10]. ViT splits an image into a series of ordered patches and utilizes a self-attention mechanism to capture important parts in the image that contribute to image recognition [4]. One of the proposed frameworks generates overlapping patches with a sliding window to avoid harming the local neighboring structures when splitting images and introduces a patch selection module to focus on the important local patches [6]. Another work introduces a patch fusion module that aggregates important patches from early ViT layers to extract low-level and middle-level information [7]. However, these models suffer from huge computational costs, even more than the baseline ViT, which blocks their deployment on resource-limited devices such as mobile phones and various IoT devices. To design compact models specialized for FGVC, a repeated neural network design process would take place and result in enormous computational costs. According to [11], the repeated process of designing and training to find a specialized transformer model on modern tensor processing hardware could cause 626,000 pounds of CO_2 emission equivalent to that of 5 cars' lifetime. Despite their exceptional performance, these power hungry models pose a great hindrance to sustainable systems. It is crucial to make the large-scale networks efficient and sustainable to be deployed across various hardware with different computational budgets and make them suitable for real-world applications.

There have been efforts to reduce the computational cost of ViT models at runtime by pruning the image patch tokens across the transformer layers [12]–[15]. Existing works introduce additional probability measures either by reusing a portion of the learnable parameters [13]–[15] or by introducing new modules to the ViT architecture to selectively prune tokens [12]. However, these models are conceived for certain scenarios where inference budget is known a priori. In order to deploy these existing adaptive models on various hardware, it is inevitable to go through a repetitive process of designing and training compact version of the models suitable to the deployment constraints. Less effort has been placed in developing sustainable models that although trained just once are suitable for diverse inference computational budgets by selecting a subconfiguration from the original model that meets the deployment constraints with optimal accuracy.

In this paper, we propose a framework capable of training the Token Adaptive Vision Transformer (TAVT) that progressively eliminates redundant image patch tokens to maximize accuracy and minimize required computational resources. We conduct a multi-objective evolutionary search to find a full Pareto-frontier of image patch pruning schemes that provides optimal

This work was supported in part by Semiconductor Research Corporation, Research in Intelligent Storage and Processing in Memory, NSF Grant 1822923, and 1955815.

978-3-9819263-7-8/DATE23/© 2023 EDAA

accuracy-efficiency trade-offs given any computational budget constraint in the inference time.

The main contribution of this work includes:

- We introduce progressive dropout of the image patch tokens based on attention scores for training a Token Adaptive Vision Transformer to reduce the computational cost of running inference on FGVC task without introducing additional parameters.
- We apply a sandwich rule with an in-place distillation training technique to make the model adaptive to arbitrary token drops and allows us to apply different token drop schemes in the inference time without compromising accuracy.
- We conduct a multi-objective evolutionary search on TAVT model to automatically find optimal accuracy-efficiency tradeoffs for various computational budgets for efficient model deployment on various hardware. In contrast to other adaptive token dropping techniques that offer input dependent runtime efficiencies, the efficiency savings obtained using this submodel configurations are independent of the input.

Empirical results show that our proposed model significantly cuts down computational and memory costs while achieving superior performance on four popular fine-grained benchmarks compared to the ViT baseline and state-of-the-art (SOTA) models.

II. Background

In this section, we review Vision Transformer and its self-attention mechanism. Our framework uses the attention mechanism to decide which tokens to prune at each layer. Then we briefly review existing transformer-based models on the FGVC task.

A. Vision Transformer with self-attention

Vision Transformer is the first model that directly applies the transformer architecture for the image classification task. The model consists of a patch embedding layer and multiple encoder layers. First, the model splits and converts input images to sequences of non-overlapping image patches. Then, the image patches are linearly projected into a d-dimensional latent embedding space in the patch embedding layer and a position embedding is added to the patch embedding to retain positional information. An additional class token is added to extract image representations by correlating with other image patches and fed to the classification head. The sequence of embedded tokens is fed to encoder layers. An encoder layer includes a multi-head self-attention layer and a feed forward network that consists of two fully connected layers for non-linear transformation. The multi-head self-attention layer decomposes the scaled dot-product attention to extract independent features from the input sequence in parallel. Let $\mathbf{x} = (x_1, x_2, \ldots, x_T)$ be a sequence of T image patch embeddings where $x_t \in \mathbb{R}^d$. A self-attention mechanism is defined as

$$\mathrm{SA}(\mathbf{x}) = W_o \sum_{t=1}^{T} \alpha_t(x_t) W_v x_t \tag{1}$$

where

$$\alpha_t(x_t) = \mathrm{softmax}\left(\frac{x_t^\top W_q^\top W_k x_t}{\sqrt{d}}\right) \tag{2}$$

$W_o, W_v, W_q, W_k \in \mathbb{R}^{d\times d}$ are weights for linear transformation. Then the multi-head self-attention is defined as

$$\mathrm{MSA}(\mathbf{x}) = \sum_{h=1}^{H} \mathrm{SA}_h(\mathbf{x}) \tag{3}$$

where H is a set of attention heads h and SA_h is a decomposed low-rank attention from the head h. All the representation outputs from SA_h are created from the same input and merged to produce a single output. In our framework, the attention score matrix α from (2) is utilized to select which tokens to prune.

B. ViT on Fine-Grained Visual Classification

Recent works proposed ViT-based models and achieved SOTA performance in FGVC [6]–[10]. TransFG is the first work to extend the ViT into FGVC [6]. The framework generates overlapping patches to minimize local visual information loss from splitting an input image to multiple patches and introduces a Part Selection Module (PSM) before the last transformer layer to select important tokens with discriminative features based on the aggregated attention weights. Another framework FFVT proposes Mutual Attention Weight Selection (MAWS) module to select image patch tokens that are similar to the class tokens to extract different levels of global and local information in images [7]. However, the existing models are computationally expensive, having quadratic compute costs based on the input sequence length. Designing an efficient ViT model is a crucial task to enable deploying the powerful models on various hardware for real-world fine-grained applications.

III. Token Adaptive Vision Transformer

In this section, we describe the progressive token pruning strategy based on the attention importance score to dropout redundant tokens. We train the Token Adaptive Vision Transformer with the pruning strategy to make the final model robust to arbitrary token drops at inference time. Fig. 1 shows how tokens are progressively pruned throughout the encoder layers according to the attention scores. Then we conduct a multi-object evolutionary search on the trained model to find the optimal token pruning scheme under the target computational budgets.

A. Attention score for token pruning

To drop out the patch tokens based on the pruning schemes, (2) is used as the importance scoring function for a sequence of tokens. The equation measures the attention imposed by x_t on the other image patches $x \in \mathbf{x}$. A patch token x with a high importance value is likely to contain discriminative visual features that have a strong local and global correlation with other important tokens. These tokens with high scores influence the final classification. The significance score of x is the overall attention score aggregated over the heads. We identify the top-k attentive tokens based on the significance score and retain the highest value tokens based on the pruning scheme.

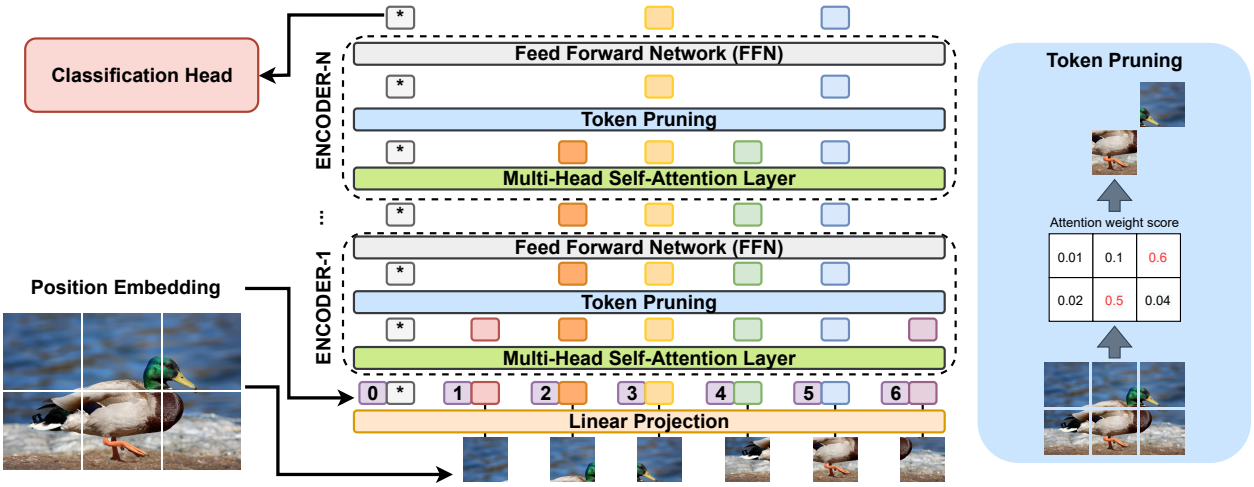

Fig. 1. The overview of our proposed Token Adaptive Vision Transformer. The input sequence of image patch tokens are projected into the embedding space and fed to the encoder blocks. The progressive token pruning based on attention scores show the number of selected token features are gradually decreased as they pass to the next layer.

B. Training Token Adaptive Vision Transformer

To train our adaptive model to be used for different scenarios with various computational budgets, we assign random token drop schemes to the model during the training process. For each iteration, we sequentially sample the number of retained patch tokens n_{i+1} at the $(i+1)$-th layer within the range$[(1-p)n_i, n_i]$ where n_i is the number of retained tokens in the previous layer and p is the token dropout rate. This way, the tokens are progressively pruned across the layers to reduce computational costs. Additionally, we perform random layer skipping for each iteration to make the model robust to the random token drop.

We applied the sandwich rule training technique introduced by [16] to effectively train our adaptive model. First, we update the model with the upper bound configuration where none of the tokens are dropped but layers are skipped uniformly at random. Second, we apply in-place distillation to update the model with the lower bound with the maximum token pruning configuration and other randomly sampled intermediate pruning configurations to transfer the knowledge from the full model to the sparse models with various pruning schemes. In each iteration, both the full and sparse models are optimized simultaneously to make the model adaptive to arbitrary token drops at inference time. With the same number of training steps as the baseline ViT, our approach results in a superior accuracy-latency trade-off.

C. Evolutionary Search on Token pruning configurations

After training our Token Adaptive Vision Transformer, we conduct a multi-objective evolutionary search to find the optimal token pruning scheme that maximizes accuracy and efficiency for the target computational budgets. Evolutionary search on our adaptive model requires a significantly less computational cost since our model provides instant accuracy-efficiency trade-offs without additional training and the search process only takes inference on a small validation dataset compared to a repeated design and training process to find specialized models for limited scenarios.

First, we initialize the population of token pruning schemes with constant drop ratios that are evenly spaced. That way, the initial population is uniformly distributed between the upper bound and the lower bound pruning configurations. At each iteration, we evolve the population to only retain configurations with the optimal accuracy-efficiency trade-offs that lie on a new Pareto frontier. Then we apply mutation and crossover to generate more population from the current optimal configurations to find better trade-offs. A mutation transforms an original pruning configuration $(g_1, \cdots, g_L)$ to $(g'_1, \cdots, g'_L)$ where an arbitrary element g_i for i-th layer in the original pruning configuration $(g_1, \cdots, g_L)$ is updated to a new value g'_i sampled from the uniform distribution $\left(g'_{i-1}, g_{i+1}\right)$ to retain progressive pruning. A crossover randomly selects two pruning configurations from the population and averages the pruning values at each layer. In each evolution iteration, we maintain n_m mutated configurations and n_c configurations from the crossover. The final iteration will generate the furthest Pareto frontier that consists of configurations with optimal accuracy-efficiency trade-offs.

IV. Experiments

We evaluate our framework for FGVC task and compare it to the baseline ViT model and SOTA Transformer-based models.

A. Model and Datasets

We base our framework on the ViT model (ViT-B/16) that is pre-trained on the Imagenet21K dataset [17]. The model consists of an embedding layer and 12 encoder layers with 12 self-attention heads in each layer. The embedding layer is pre-trained to process sequences of patches of size 16. We used four widely used fine-grained benchmarks including CUB-200-2011, Stanford Cars, Stanford Dogs, and NABirds datasets to evaluate our method [18]–[21].

B. Evaluation metrics

We compare our approach to the baseline ViT model fine-tuned with the four mentioned datasets and the two SOTA

ViT-based models including TransFG and FFVT. We focus on comparing the inference efficiency with three different metrics: the overall memory usage, latency, and the number of floating operations (FLOPs), which is independent of hardware and could be used as a proxy for efficiency [22]. We measured both the average GPU and CPU latency across the validation dataset with a single Nvidia Tesla V100 GPU and a Intel i9-9980XE (18 threads) CPU using Pytorch and APEX. The input images with a batch size of 8 are fed to measure all three efficiency metrics. The same setup is shared across all comparing models and our proposed method for a fair comparison.

C. Experimental Setup

For data preparation, we performed the data augmentation used by TransFG and FFVT. We applied random cropping for training and center cropping for testing to have 448×448 size input images and adopted extra color augmentation. We fine-tune the pre-trained ViT-B/16 model for the four fine-grained benchmarks without any token pruning to have the baseline ViT models. We used SGD optimizer to optimize the network with a momentum of 0.9 and applied the cosine annealing scheduler. The initial learning rate is set to 0.03 except 0.003 for Stanford Dog benchmark and the batch size is set to 16. Then we further fine-tuned the baseline ViT model on our token adaptive framework with a token drop rate set to 0.2. The same training setup for training the baseline ViT model was used to train our adaptive model. We fine-tuned with 2 randomly sampled intermediate token pruning configurations in addition to upper and lower bound configurations to apply the sandwich training technique.

To find the accuracy-efficiency Pareto frontier of token pruning configurations for different compute budgets, we run 30 iterations of evolutionary search with 30 mutation configurations with a mutation probability of 0.5 and 30 crossover configurations populated on each iteration.

For training SOTA ViT-based models, we followed the repositories of TransFG and FFTV from the authors to configure training on fine-grained datasets.

V. Results and Analysis

In this section, we evaluate the experiments and compare the performance of our method to the other ViT-based models. We further analyze the pruning results in terms of the accuracy-efficiency trade-offs.

A. Pareto Frontier

We used the four fine-grained benchmarks to investigate the effect of the proposed framework on the accuracy-efficiency trade-off. Fig. 2 shows Pareto front curves of the proposed adaptive model trained on the four benchmarks. Each point in the Pareto curves corresponds to a submodel from the original configuration with a specific token pruning scheme. It is noticeable in the CUB-200-2011, Stanford Car, and NABirds Pareto curves that the sparse model with certain token drop schemes has even higher accuracy with significantly less number of floating point operations compared to the full model, which corresponds to the rightmost point on the Pareto curve. As we try to reduce the compute cost (FLOPs), our models trained with the token pruning scheme remove unnecessary distracting tokens leading to boosting the accuracy. Beyond a certain point, token pruning starts to drop essential information as well showing the trend toward lowering accuracy consequently.

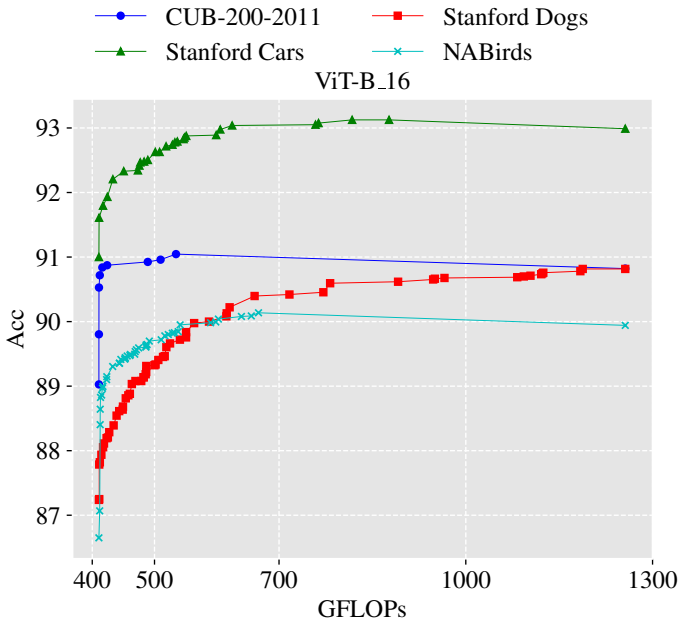

Fig. 2. Pareto frontier curves of accuracy to GFLOPs on four standard fine-grained benchmarks.

B. Maximizing efficiency gain

We compared our adaptive model with the baseline ViT and the two SOTA ViT-based models: TransFG and FFVT. Table I, II, III, and IV show the accuracy and efficiency of the models on four different fine-grain benchmarks. To validate that our proposed framework could find token pruning configurations that maximize both accuracy and efficiency, we searched two pruning configurations for each model that maximize accuracy and efficiency respectively. TAVT^{p} denotes the performance model with the pruning scheme that achieves the highest accuracy. TAVT^{e} denotes the most efficient model within 1 percent of the baseline ViT model accuracy. For all benchmarks, our performance model achieves higher accuracy and higher efficiency compared to the baseline ViT model.

TABLE I
Comparison of different methods on CUB-200-2011 dataset

Model	Acc	latency(ms)	FLOPs	Mem usage (Gb)
TransFG	91.6	671	5.70x	21.1
FFVT	91.5	160	0.93x	8.96
VIT	90.6	134 / 2232⋆	1.00x	9.41
TAVT^{p}	91.1	63 / 850⋆	0.43x	3.8
TAVT^{e}	89.8	50 / 623⋆	0.33x	2.9

p is the performance model. e is the efficiency model. ⋆ on CPU

On CUB-200-2011 benchmark, our performance model achieves 0.5% higher accuracy while requiring only 43% of the number of floating operations from the baseline ViT. Other efficiency metrics show linear correlations with FLOPs. Our model is 2× and 2.6× faster on GPU and CPU respectively, and reduces the memory usage by 2.5× compared to ViT.

Our efficiency model is 2.7× and 3.6× faster on GPU and CPU respectively, and only requires 33% of the number of floating operations (FLOPs) from ViT with only a minor loss of accuracy. In Table II, our efficiency model is 2.3 × faster with 41% of FLOPs compared to ViT within 1 percent of the ViT model accuracy on Stanford Dogs benchmark. In Table III, our performance model achieves 1% higher accuracy while requiring 47% of FLOPs from ViT on Stanford Cars benchmark. NABirds benchmark from Table IV also shows both performance and efficiency models significantly reduce memory usage and FLOPs while achieving competitive accuracy.

TABLE II
COMPARISON OF DIFFERENT METHODS ON STANFORD DOGS DATASET

Model	Acc	latency(ms)	FLOPs	Mem usage (Gb)
TransFG	90.6	672	5.70x	21.1
FFVT	91.3	162	0.94x	8.79
VIT	90.5	144 / 2359⋆	1.00x	9.41
TAVT^{p}	90.8	129 / 2051⋆	0.89x	8.41
TAVT^{e}	89.6	61 / 789⋆	0.41x	3.62

p is the performance model. e is the efficiency model. ⋆ on CPU

TABLE III
COMPARISON OF DIFFERENT METHODS ON STANFORD CARS DATASET

Model	Acc	latency(ms)	FLOPs	Mem usage (Gb)
TransFG	92.4	671	5.70x	21.1
VIT	92.1	134 / 2214⋆	1.00x	9.41
TAVT^{p}	93.1	87 / 1228⋆	0.60x	5.27
TAVT^{e}	91.6	51 / 640⋆	0.34x	2.96

p is the performance model. e is the efficiency model. ⋆ on CPU

Furthermore, compared to the SOTA models, our performance model is substantially more efficient with a minor drop in accuracy. In Table I, our model is 10X faster with 13X fewer FLOPs and 5.5X less memory usage with only 0.5% accuracy drop compared to TransFG on CUB-200-2011. Compared to FFVT, which does not use overlaying patches, our model is 2.5× faster requiring 2× fewer FLOPs and 2.4× less memory usage with only 0.4% accuracy drop. For Stanford Dogs benchmark in Table II, our performance model achieves 0.2% higher accuracy while requiring only 15% of FLOPs and 40% of memory usage from TransFG. In Table III, our model outperforms the SOTA models on Stanford Cars benchmark as well. With a minor accuracy drop, our performance model is 7.7× faster and requires 9.5× and 4× smaller number of FLOPs and memory usage respectively compared to TransFG. For NABirds in Table IV, our performance model is 9× faster and improves memory usage and FLOPs by 4.7× and 11× respectively with only 0.5% accuracy drop compared to TransFG. With a drastic reduction in memory usage, our efficiency models could be deployed on a Raspberry Pi 4 with 4GB RAM or a low-profile Nvidia GTX 1650 GPU with 4GB RAM.

C. Token drop distribution and visualization

We further analyze the token pruning configurations and visualize how tokens are progressively pruned out throughout the layers. Fig. 3 shows how many tokens are retained in each layer based on the optimal token pruning configurations that maximize efficiency within 1 percent of the baseline ViT model accuracy on the four benchmarks. About half of the tokens are pruned from the first 3 layers on all the benchmarks. There is also an extreme pruning on the last layer where only 6 and 4 tokens are retained for Stanford Car and NABirds benchmarks respectively.

TABLE IV
COMPARISON OF DIFFERENT METHODS ON NABIRDS DATASET

Model	Acc	latency(ms)	FLOPs	Mem usage (Gb)
TransFG	90.6	674	5.70x	21.1
VIT	89.7	136 / 2258⋆	1.00x	9.41
TAVT^{p}	90.1	74 / 1092⋆	0.51x	4.45
TAVT^{e}	88.8	55 / 639⋆	0.33x	2.96

p is the performance model. e is the efficiency model. ⋆ on CPU

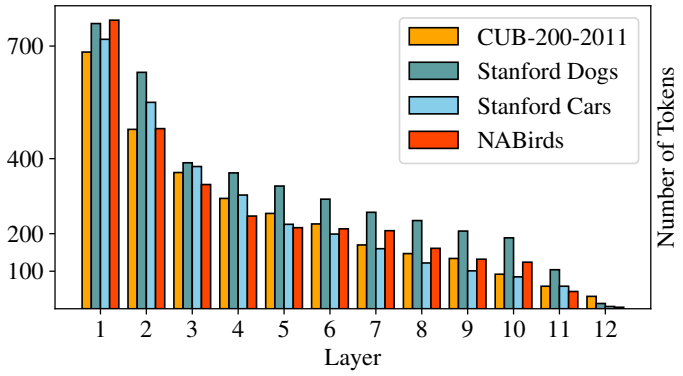

Fig. 3. The token length configuration of TAVT^{p} with the maximum efficiency gains on the fine-grained benchmarks.

To further analyze the behavior of the progressive token pruning applied in our model, we visualize the pruning procedure in Fig. 4. We show the pruning results after each layer where the red masks represent the dropped tokens. We find that our progressive token pruning scheme can gradually prune out distractor tokens such as backgrounds and local features that do not contribute to the fine-grained objects. Our adaptive model can focus on the objects and local distinctive features of the fine-grained objects in the images. This suggests that our model can hierarchically capture discriminative local parts and features from objects in the image which contribute most to the fine-grained classification.

VI. CONCLUSION

In this paper, we propose Token Adaptive Vision Transformer (TAVT), a framework that progressively drops image patch tokens. Our adaptive model can switch among different token drop configurations at runtime, providing instant accuracy-efficiency trade-offs. Our approach can significantly reduce computational effort while maintaining competitive accuracy compared to the SOTA ViT-based models for Fine-grained image classification. The multi-objective evolutionary search on TAVT model to automatically find optimal accuracy-efficiency trade-offs for various computational budgets allow the model to be deployed on various hardware without additional tuning or

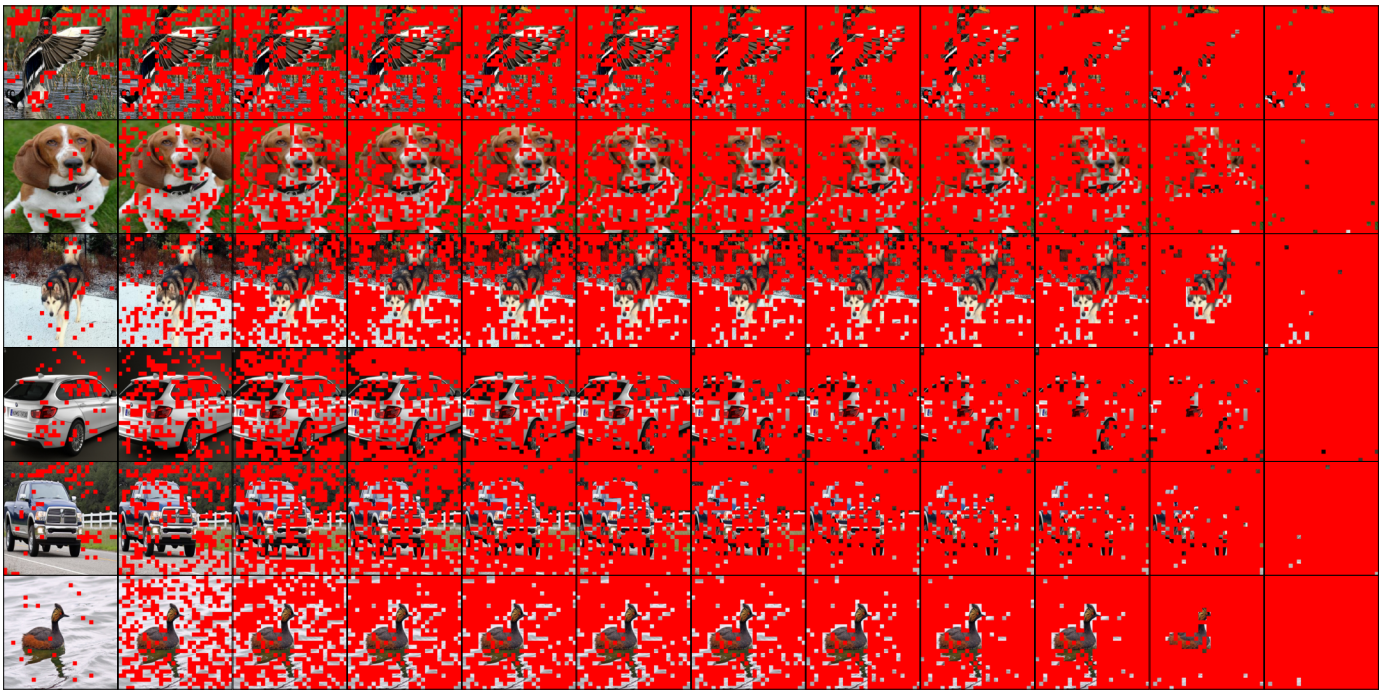

Fig. 4. Visualization of the progressive token pruning. Our model focuses on the objects and captures local discriminative features successfully in different images from various categories.

training. Furthermore, our model drastically reduces the overall memory usage, which allows inference on resource restricted devices and embedded systems.

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, Minneapolis, MN, Jun. 2019, pp. 4171–4186.

[2] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[3] T. Brown *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[4] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[5] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 10 012–10 022.

[6] J. He *et al.*, "Transfg: A transformer architecture for fine-grained recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, 2022, pp. 852–860.

[7] J. Wang, X. Yu, and Y. Gao, "Feature fusion vision transformer for fine-grained visual categorization," in *BMVC*, 2021.

[8] H. Zhu *et al.*, "Dual cross-attention learning for fine-grained visual categorization and object re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4692–4702.

[9] Y. Zhang *et al.*, "A free lunch from vit: adaptive attention multi-scale fusion transformer for fine-grained visual recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3234–3238.

[10] X. Liu, L. Wang, and X. Han, "Transformer with peak suppression and knowledge guidance for fine-grained image recognition," *Neurocomputing*, vol. 492, pp. 137–149, 2022.

[11] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern deep learning research," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 09, pp. 13 693–13 696, Apr. 2020.

[12] Y. Rao *et al.*, "Dynamicvit: Efficient vision transformers with dynamic token sparsification," *Advances in neural information processing systems*, vol. 34, pp. 13 937–13 949, 2021.

[13] H. Yin *et al.*, "A-vit: Adaptive tokens for efficient vision transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 809–10 818.

[14] Y. Tang *et al.*, "Patch slimming for efficient vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 165–12 174.

[15] Y. Liang *et al.*, "Not all patches are what you need: Expediting vision transformers via token reorganizations," in *International Conference on Learning Representations*, 2022.

[16] J. Yu and T. S. Huang, "Universally slimmable networks and improved training techniques," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1803–1811.

[17] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "Imagenet-21k pretraining for the masses," *arXiv preprint arXiv:2104.10972*, 2021.

[18] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[19] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[20] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization," in *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

[21] G. Van Horn *et al.*, "Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 595–604.

[22] G. Kim and K. Cho, "Length-adaptive transformer: Train once with length drop, use anytime with search," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 6501–6511.