Token Merging with Class Importance Score

Kwang-Soo Seol
Department of Electronic Engineering
Hanyang University
Seoul, Korea
kwang4010@hanyang.ac.kr

Si-Dong Roh
Department of Electronic Engineering
Hanyang University
Seoul, Korea
sdroh1027@hanyang.ac.kr

Ki-Seok Chung*
Department of Electronic Engineering
Hanyang University
Seoul, Korea
kchung@hanyang.ac.kr

***Abstract*—Vision Transformers have achieved high performance in computer vision tasks, but their high computational cost and low throughput are weaknesses. Therefore, much research has been done to reduce the size of Vision Transformers. Among them, studies on pruning unnecessary tokens are being actively conducted to reduce the number of tokens used for self-attention computation inside the Vision Transformer. Recently, token merging has been proposed as a new alternative approach. These studies aim to increase throughput with a small accuracy drop by merging similar tokens instead of pruning them. A previous study finds similar tokens using cosine similarity and merges them with a weighted average. However, merging a large number of tokens at once may lead to an accuracy drop because of the underestimating of important information. In this paper, we propose ToMeCIS, a method that merges similar tokens through a weighted average using the class importance score of tokens to reduce the accuracy drop. When ToMeCIS is applied to a pretrained DeiT-S and evaluated on the ImageNet-1k dataset, the throughput is increased by about 50% with an accuracy drop of less than 1% without additional training. In addition, importance scores were evaluated with different metrics to find the best accuracy versus throughput trade-off.**

***Index Terms*—computer vision, deep learning, model compression**

I. Introduction

The Transformer, a state-of-the-art deep learning model, employs self-attention mechanisms to process input sequences concurrently and capture long-range dependencies. Many Transformers, such as BERT [1], have achieved state-of-the-art performance in various natural language processing tasks. However, Transformers have not been extensively utilized in vision tasks due to the importance of local information over global information in images. Nevertheless, the recent emergence of Vision Transformer (ViT) [2] triggered active research of Transformers in vision tasks. ViT divides the image into small patches and represents each patch as a single token vector, which enables the image to be processed using the encoder structure of Transformers while preserving its local information. ViT has often demonstrated higher accuracy than traditional Convolutional Neural Networks (CNNs). However, the self-attention operation in ViT suffers from high computational cost and low throughput due to its quadratic complexity to the number of tokens.

Recently, there has been a surge of research on compressing ViTs to reduce computational costs and increase throughput. One of the key research aspects is reducing the number of tokens. These approaches employ token pruning methods to select unnecessary tokens. Primary strategies for identifying unnecessary tokens have been twofold. The first strategy exploits a learnable mask [3], [4], [5] to determine the tokens that should be pruned. During the model training phase, the mask is learned to identify tokens for pruning, which are then selected and removed using the learned mask during inference. The second strategy prunes tokens based on specific scores [6], [7], [8]. Each token is assigned an importance score, and tokens with low scores are regarded as unnecessary and subsequently pruned.

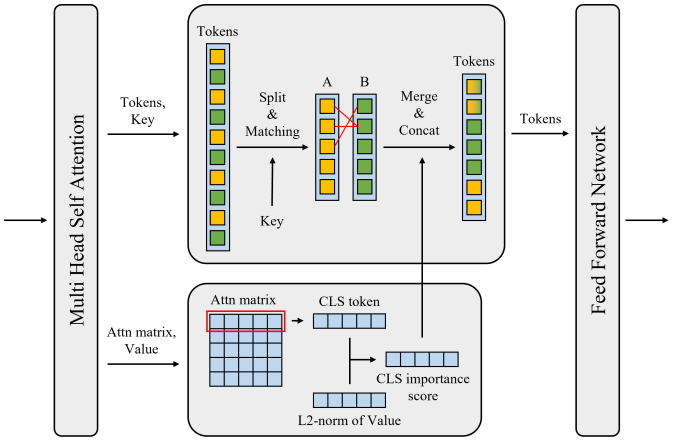

Fig. 1. Overall structure of the ToMeCIS

While these studies improve throughput by pruning tokens, they suffer the drawback of discarding meaningful information in the removed tokens. To overcome this limitation, recent studies explored reducing the number of tokens by merging similar tokens [9], [10]. These studies either learn networks to summarize token information [9] or directly calculate the similarity between tokens [10] to identify similar tokens. In the latter approach, the merging process involves a weighted averaging of tokens, which takes the number of tokens combined in previous layers into account. However, the study does not factor in the importance of individual tokens. An equal mix of important and less important tokens can incur the loss of information contained in the important tokens.

In this paper, we propose ToMeCIS, a novel method for merging similar tokens while considering the importance of each token. We utilize a metric called `class importance`

`score` (CIS) to represent the importance of each token. The CIS of each token is calculated using the value of the token and the relationship between the class token and the corresponding token. These values can be easily obtained during self-attention of Vision Transformers. ToMeCIS identifies similar tokens using cosine similarity between tokens and merges them into a new token. The values of the new tokens are obtained by computing a CIS-based weighted average of the merged tokens. Fig. 1 shows the overall structure of ToMeCIS. `Attn` and `CLS` refer to attention and class, respectively. We apply ToMeCIS to DeiT and evaluate its performance on image classification tasks with the popular ImageNet-1k dataset [11]. Experimental results demonstrate that ToMeCIS increases throughput by 50% with an accuracy drop of less than 1% compared to DeiT-S without additional training. We also compare the performance of ToMeCIS with other ViTs with a similar number of parameters and different token reduction methods. Furthermore, we explore various ways of scoring token importance in ablation studies and measure their accuracy and throughput.

II. Related works

A. Transformer in vision tasks

The ViT [2] is a Transformer model adapted for vision tasks. ViT converts input images into a sequence of tokens and utilizes the Transformer encoder to process them. ViT classifies the images by taking the output, which includes the class token containing class-specific information, from multiple encoder layers and feeding it into an MLP layer. ViT has demonstrated superior performance to traditional Convolutional Neural Networks. However, it has the drawback of requiring training on a large dataset of hundreds of millions of samples such as JFT-300M [12] to achieve good performance. To address this limitation, DeiT [13] leverages knowledge distillation via distillation tokens and achieves competitive performance by training on ImageNet-1k, a dataset of 1 million samples.

B. Token reduction

The self-attention mechanism in Transformers incurs a quadratic increase in computational cost with respect to the number of tokens. Consequently, a lot of studies have focused on reducing the number of tokens. DynamicViT [3] prunes tokens by learning a binary decision mask that distinguishes between necessary and unnecessary tokens. EViT [6] determines token importance by computing attention scores based on self-attention values and prunes tokens with low attention scores. ATS [8] assigns importance scores to tokens by multiplying the attention score with the L2-norm of each token's value and selectively prunes tokens through sampling. TokenLearner [9] reduces the number of tokens by generating a smaller set of new tokens through a learnable linear layer that selects information from input image pixels. ToMe [10] calculates token similarity using cosine similarity and reduces the number of tokens by merging similar tokens through a weighted average using bipartite soft matching. In ToMe, the merging weight of each token is the token size. The proposed method of this paper is similar to ToMe, but the main difference is that our method calculates the merging weight based on the importance of tokens which can be obtained during self-attention.

III. Proposed Method

In this section, we outline the ViT procedure briefly and explain the bipartite soft matching algorithm employed in ToMe [10]. Next, we describe our novel token merging process based on class importance scores.

A. Vision Transformer

In the ViT, the input image is initially fed into the embedding layer to generate tokens. The image is divided into patches of uniform size. Each patch is then projected into a vector, and a token is created by incorporating positional information. A set of tokens generated after embedding is as follows:

$$Z_0 = [z_{0,cls}; z_{0,1}; z_{0,2}; ...; z_{0,N}] + E_{pos}, \tag{1}$$

where the first token ($z_{0,cls}$) is a class token that is a learnable value after initialization with random values. The generated set of tokens Z_0 serves as an input to the Transformer encoder blocks. A Transformer encoder block is computed as follows:

$$Z'_l = MSA(LN(Z_{l-1})) + Z_{l-1}, \tag{2}$$

$$Z_l = FFN(LN(Z'_l)) + Z'_l. \tag{3}$$

A Transformer encoder block consists of a multi-head self-attention (MSA) layer and a feed-forward network (FFN). LN represents a layer normalization, and l represents the block number. To explain MSA, we introduce the attention mechanism:

$$Attention(Q, K, V) = A \times V, \tag{4}$$

$$A = softmax(\frac{Q \times K^T}{\sqrt{d}}). \tag{5}$$

Q, K, and V represent the query, key, and value matrices constructed from the input tokens to facilitate attention calculation, respectively. A and d denote the attention matrix and the scaling factor, respectively. The attention output is generated by multiplying the attention matrix A with the value matrix V. The attention matrix A is computed by multiplying the query matrix with the transpose of the key matrix. The value will be scaled by a scaling factor and a softmax operation will follow. The scaled factor is derived by taking the square root of the dimensions of the query and key matrices.

After tokens traverse a total of L Transformer encoder blocks sequentially, the final classification result y is obtained as follows:

$$y = LN(Z'_{L,cls}) \tag{6}$$

Once the final Transformer encoder block operation is complete, only the class token among the output tokens is passed as the input to the linear layer. Then the result y is obtained using the output of the linear layer.

Algorithm 1 Bipartite Soft Matching algorithm

Input: Key tokens K, The number of reducing tokens r
Output: Indices of not matched tokens T_{ui}, Indices of matched tokens T_{si}, T_{di}

$A \leftarrow$ values at the even indices of K
$B \leftarrow$ values at the odd indices of K
$S \leftarrow \frac{A \times B^T}{\|A\| \times \|B\|}$ ▷ cosine similarity
$M_v \leftarrow max(S, dim = 1)$
$M_i \leftarrow argmax(S, dim = 1)$
$M_{vs} \leftarrow argsort(M_v, order = descending)$
$T_{si} \leftarrow M_{vs}[: r]$ ▷ top r
$T_{ui} \leftarrow M_{vs}[r :]$
for $i \leftarrow 0$ to N **do**
    $T_{di}[i] \leftarrow M_i[T_{si}[i]]$
end for

Algorithm 2 Merge algorithm

Input: Input tokens T, Indices of not matched tokens T_{ui}, Indices of matched tokens T_{si}, T_{di}
Output: Merged tokens T_m

$T_e \leftarrow$ tokens at the even indices of T
$T_d \leftarrow$ tokens at the odd indices of T
$N_u \leftarrow$ number of index in T_{ui}
$N_s \leftarrow$ number of index in T_{si}
for $i \leftarrow 0$ to N_u **do**
    $T_u[i] \leftarrow T_e[T_{ui}[i]]$
end for
for $i \leftarrow 0$ to N_s **do**
    $T_s[i] \leftarrow T_e[T_{si}[i]]$
end for
for $i \leftarrow 0$ to N_s **do**
    $T_d[T_{di}[i]] \leftarrow T_d[T_{di}[i]] + T_s[i]$
end for
$T_m \leftarrow concat(T_d, T_u)$

B. Bipartite Soft Matching

Bipartite soft matching is an algorithm proposed in ToMe [10] for selecting and merging similar pairs of tokens. The algorithm proceeds as follows:

1 The total set of the Key tokens is divided into two sets of almost equal size, A and B.
2 For each token in set A, the most similar token from set B is determined by calculating the cosine similarity between their respective keys.
3 The token pairs are sorted based on similarity, and the r most similar pairs are selected for merging. This step may involve combining one token from set B with multiple tokens from set A.
4 The tokens in a selected pair are merged to create a new token. The new token will have values that correspond to the weighted averages of the values of the merged tokens, and the token size will be used as the weight. The token size is the number of tokens merged in the previous layers for each token.

The size of a token indicates how many tokens have been cumulatively merged into the token. Prior to the merging process, all tokens have never been merged with any other token. So, all elements in the token size vector S are initialized to 1. After each merging process is complete, the token size vector S will be updated depending on the previous token size vector and token matching information. In ToMe, the value of the new token is computed by token size-based weighted averaging, which means that tokens that have a bigger token size have more influence on the final value. The detailed process of the bipartite soft matching algorithm is described in Algorithms 1, 2, and 3.

Algorithm 3 Merge with token size

Input: Input tokens T, Key tokens K, Token size S, The number of reducing tokens r
Output: Merged tokens T_m

$T_{ui}, T_{si}, T_{di} \leftarrow BipartiteSoftMatching(K, r)$
$T \leftarrow Merge(T \times S, T_{ui}, T_{si}, T_{di})$ ▷ Algorithm 2
$S \leftarrow Merge(S, T_{ui}, T_{si}, T_{di})$ ▷ update token size
$T \leftarrow T/S$ ▷ weighted average

C. CIS-based Merging

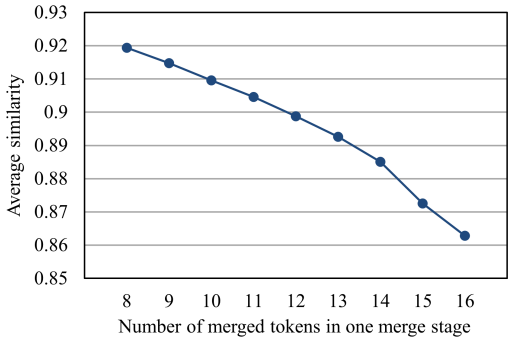

Fig. 2. Average cosine similarity with respect to the number of merged tokens in one merge stage

ToMe conducts token merging using bipartite soft matching and token size-based weighted averaging. However, using the token size as the weight may underestimate the importance of each token for classification during the merging process, especially if you are merging a large number of tokens in a single merging stage. By underestimation, we mean that when important tokens are merged with less important tokens, the important information needed for classification may be lost, leading to a significant accuracy drop.

Fig. 2 shows the average cosine similarity of matched tokens with respect to the number of tokens merged in a single merge

step. As depicted in the figure, there is a decrease in the average similarity of matched tokens as the number of tokens merged in one merge step increases. This observation hints that merging a large number of tokens in one merge step may result in merging dissimilar tokens where some distinctive information in the merged tokens may be lost.

To alleviate this concern, we propose a novel method named ToMeCIS. Our method computes values of the merged tokens as the weighted average where class importance scores (CISs) are used as weights. It is important that a score value should represent each token's importance, and we utilize the scoring mechanism employed in token pruning domain [8]. We call this class importance score (CIS) in the paper. CIS is denoted as S_{cls} and computed as follows:

$$A_{cls} = softmax(\frac{Q_{cls} \times K^T}{\sqrt{d}}), \tag{7}$$

$$S_{cls} = A_{cls} \odot \|V\|_2 , \tag{8}$$

where Q_{cls} is the first row of the query matrix Q. Since the ViT relies solely on class tokens for classification, we use A_{cls}, the first row of the attention matrix, to assign CIS. The attention layer generates an output by multiplying V with the attention matrix. Note that A_{cls} can be obtained without computation because it was already calculated in (5). Then we apply the l2-norm to transform matrix V and perform element-wise multiplication with A_{cls} to get CIS (S_{cls}). By incorporating information from both the attention matrix and the V of each token, we can express the importance of each token more accurately than the case where only the attention matrix is used. If the self-attention process is multi-headed, the CIS is computed by averaging S_{cls} of all heads. The detailed steps of the proposed CIS-based merging are described in Algorithm 4.

Algorithm 4 Merge with class importance score

Input: Input tokens T, Key of input tokens K, Value of input tokens V, Attention matrix A

Output: merged tokens T_m

$T_{ui}, T_{si}, T_{di} \leftarrow BipartiteSoftMatching(K, r)$
$A_{cls} \leftarrow$ first row of A
$S_{cls} \leftarrow A_{cls} \odot \|V\|_2$ ▷ class importance score
$T \leftarrow Merge(T \times S_{cls}, T_{ui}, T_{si}, T_{di})$ ▷ Algorithm 2
$T \leftarrow T / S_{cls}$ ▷ weighted average

IV. Experiments

A. Experimental settings

To evaluate the performance of the proposed method, we conducted a classification task on the ImageNet-1k dataset [11]. The ImageNet-1k dataset consists of 1 million training samples and 50,000 validation samples of 1,000 classes. As the base model, we used DeiT [13]. The ToMeCIS was applied to the pretrained DeiT-Ti and DeiT-S by merging 13 tokens in each Transformer block. We conducted experiments without fine-tuning the model after applying ToMeCIS to the pretrained model. The evaluation was conducted with a batch size of 128, using FP32 precision on a single GeForce RTX 3090 GPU. Using the ImageNet-1k validation set, the performance in terms of the accuracy and the throughput of the models was evaluated.

B. Results

Table I presents the performance of ToMeCIS with DeiT compared with other Vision Transformers (ViTs) in terms of accuracy, GFLOPs, and throughput. In the case of DeiT-Ti, ToMeCIS reduced GFLOPs by 46% and increased throughput by 40% with a 1.6% accuracy loss compared to the base model. In comparison to other ViTs with a similar number of parameters, the accuracy change ranged from -1.1% to -2.8% while achieving a GFLOPs decrease of 0% to 1.1% and a throughput increase of 30% to 109%. In the case of DeiT-S, ToMeCIS reduced GFLOPs by 41% and increased throughput by 52% with a 0.9% accuracy loss compared to the base model. Compared to other ViTs with a similar number of parameters, the throughput was increased by 30% to 105%, with an accuracy decrease of 2% to 4.4%.

TABLE I
Performance Comparison of Vision Transformers

Model	Params (M)	Top1-Acc (%)	GFLOPs	Throughput (img/s)
DeiT-Ti [13]	5.7	72.2	1.3	3342.9
T2T-ViT-7 [14]	4.3	71.7	1.1	2498.4
PiT-Ti [15]	4.9	73.0	0.7	3594.9
CrossViT-Ti [16]	6.9	73.4	1.8	2235.1
DeiT-Ti-ToMeCIS	5.7	70.6	0.7	4672.4
DeiT-S [13]	22.1	79.8	4.6	1282.5
PiT-S [15]	23.5	80.9	2.9	1494.7
CrossViT-S [16]	26.7	81.0	5.6	905.1
Swin-T [17]	29.0	81.3	4.5	914.1
T2T-ViT-14 [14]	22.0	81.5	1.1	1017.8
LV-ViT-S [18]	26.2	83.3	6.6	951.2
DeiT-S-ToMeCIS	22.1	78.9	2.7	1949.6

TABLE II
Performance Comparison of token reduction methods on DeiT-S

Methods	Acc (%)	GFLOPs	Throughput (img/s)
Baseline [13]	79.82	4.6	1282.5
EViT [6]	78.52 (-1.30%)	3.0 (-35%)	1872.3 (+46%)
ToMe [10]	78.79 (-1.03%)	2.7 (-42%)	1956.1 (+53%)
ToMeCIS	78.90 (-0.92%)	2.7 (-42%)	1949.6 (+53%)

Table II presents the performance of ToMeCIS when compared with other token reduction methods in terms of accuracy, GFLOPs, and throughput. All methods were applied to the pretrained DeiT-S without additional training. Among all the methods, ToMeCIS achieved the highest accuracy. Both ToMeCIS and ToMe achieved the lowest GFLOPs while ToMe achieved the highest throughput. However, the difference in throughput between ToMeCIS and ToMe is negligible. In

conclusion, ToMeCIS achieves the highest performance in terms of accuracy, GFLOPs, and throughput.

C. Ablation Study

Several scoring mechanisms have been proposed to accurately estimate the importance of each token [6], [7], [8]. In this section, we evaluate the performance of ToMeCIS using various scores while gradually increasing the number of merged tokens for each layer. The scores we considered include the token size, the class attention score (A_{cls}), and the class importance score (S_{cls}). Additionally, a new scoring method called attention mean score was evaluated as well. The attention mean score represents the average influence of a token on the other tokens. It was obtained by averaging the attention matrix's row corresponding to each token. We applied ToMeCIS to DeiT-S for our experiments and merged 8 to 16 tokens for each layer. That is, the 8 to 16 most similar token pairs were selected for merging in a bipartite soft matching process. The range of the number of merged tokens in one merge process was determined based on the observation that merging more than 8 tokens at once led to a significant throughput improvement of approximately 20% or more, resulting in meaningful acceleration. If the number of merged tokens was bigger than 16, it turned out that an accuracy drop of more than 0.3% was observed with an increment in the number of merged tokens by one. Considering the trade-off between accuracy and increased throughput, this accuracy drop was deemed significant.

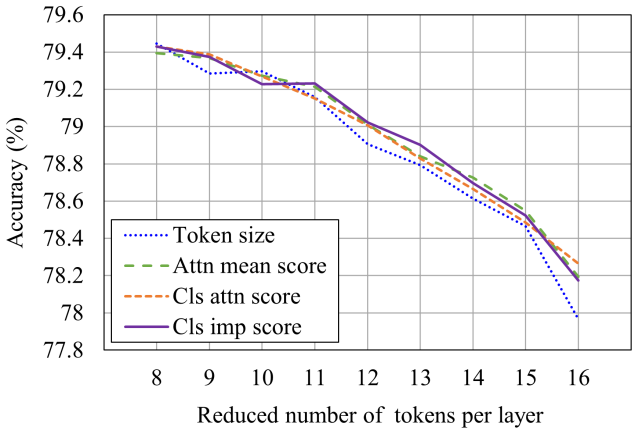

Fig. 3. Accuracy versus the reduced number of tokens per layer with respect to various score metrics

Fig. 3 represents the relationship between the accuracy and the number of merged tokens per layer for each score. When the number of tokens merged in one merge stage is less than 11, the accuracy differences do not show any particular trend. On the other hand, the accuracy when using the token size was similar to or less than the other scores when 11 or larger tokens were merged. When 11 to 13 tokens were merged, the class importance score achieved the highest accuracy. The attention mean score achieved the highest accuracy when 14 and 15 tokens were merged. When 16 tokens were merged, all methods showed a significant drop in accuracy, while the class attention score achieved the highest accuracy.

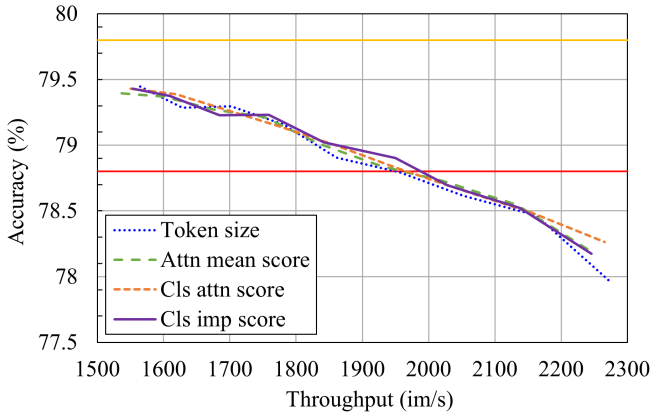

Fig. 4. Accuracy versus throughput with respect to various score metrics

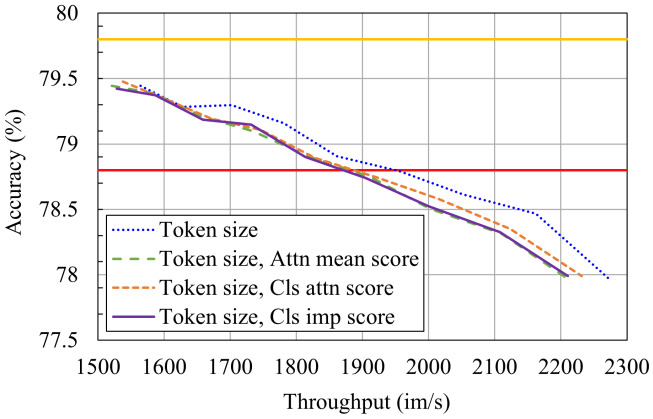

Fig. 5. Accuracy versus throughput with respect to token size and score metrics

Fig. 4 shows the relationship between accuracy and throughput with respect to various scoring metrics. The yellow line in the graph denotes the baseline accuracy, while the red line corresponds to the accuracy subtracted by 1% from the baseline. Regions between the yellow line and the red line indicate accuracy drops of less than 1% compared to the baseline. As the accuracy approaches a maximum of 1% margin, the class importance score exhibits the highest throughput, followed by the class attention score. On the other hand, the token size consistently performs similarly or worse than the other scores when the throughput is bigger than 1800 im/s. Therefore, the class importance score offers the highest accuracy-to-throughput ratio when merging is performed with a maximum accuracy margin of 1%.

We also made an attempt to use both token size and class important score to compute the weighted averages even though conceptually, there is little correlation between the token size and the class importance score. To evaluate the performance of this attempt, we make new scores obtained by multiplying

various scores with the size of each token. Fig. 5 represents the relationship between accuracy and throughput when using the token size and using the scores that consider both the token size and the importance score. Similar to Fig. 4, the yellow and the red lines represent the baseline accuracy and a maximum margin of 1%, respectively. As shown in Fig. 5, the token size achieves the best trade-off between throughput and accuracy. These results indicate that incorporating both the token size and the class importance score decreases throughput due to increased computational costs during the merging process. However, the accuracy drop is not significant compared to the token size. Consequently, achieving a decent trade-off between accuracy and throughput can be accomplished by utilizing a weighted average with either the token size or the class importance score alone.

V. Conclusion

In this paper, we proposed ToMeCIS, a new method for merging tokens based on class importance scores. We evaluated the proposed method on the DeiT models by comparing other well-known Vision Transformers. As a result, ToMeCIS achieved a 52% throughput increase with less than a 1% accuracy drop compared to DeiT-S without any additional training. Compared to the other token reduction methods, ToMeCIS achieved the best trade-off between accuracy and throughput. In addition, we verified that our scoring metric to represent the class importance outperforms other scoring metrics when they are utilized in token merging.

Acknowledgment

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No.2021-0-00131, Development of Intelligent Edge Computing Semiconductor For Lightweight Manufacturing Inspection Equipment)

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[3] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, and C.-J. Hsieh, "Dynamicvit: Efficient vision transformers with dynamic token sparsification," *Advances in neural information processing systems*, vol. 34, pp. 13937–13949, 2021.

[4] L. Meng, H. Li, B.-C. Chen, S. Lan, Z. Wu, Y.-G. Jiang, and S.-N. Lim, "Adavit: Adaptive vision transformers for efficient image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12309–12318, 2022.

[5] Z. Kong, P. Dong, X. Ma, X. Meng, W. Niu, M. Sun, X. Shen, G. Yuan, B. Ren, H. Tang, *et al.*, "Spvit: Enabling faster vision transformers via latency-aware soft token pruning," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pp. 620–640, Springer, 2022.

[6] Y. Liang, C. Ge, Z. Tong, Y. Song, J. Wang, and P. Xie, "Not all patches are what you need: Expediting vision transformers via token reorganizations," *arXiv preprint arXiv:2202.07800*, 2022.

[7] Y. Xu, Z. Zhang, M. Zhang, K. Sheng, K. Li, W. Dong, L. Zhang, C. Xu, and X. Sun, "Evo-vit: Slow-fast token evolution for dynamic vision transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 2964–2972, 2022.

[8] M. Fayyaz, S. A. Koohpayegani, F. R. Jafari, S. Sengupta, H. R. V. Joze, E. Sommerlade, H. Pirsiavash, and J. Gall, "Adaptive token sampling for efficient vision transformers," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pp. 396–414, Springer, 2022.

[9] M. Ryoo, A. Piergiovanni, A. Arnab, M. Dehghani, and A. Angelova, "Tokenlearner: Adaptive space-time tokenization for videos," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12786–12797, 2021.

[10] D. Bolya, C.-Y. Fu, X. Dai, P. Zhang, C. Feichtenhofer, and J. Hoffman, "Token merging: Your vit but faster," *arXiv preprint arXiv:2210.09461*, 2022.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[12] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proceedings of the IEEE international conference on computer vision*, pp. 843–852, 2017.

[13] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International conference on machine learning*, pp. 10347–10357, PMLR, 2021.

[14] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 558–567, 2021.

[15] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, "Rethinking spatial dimensions of vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11936–11945, 2021.

[16] C.-F. R. Chen, Q. Fan, and R. Panda, "Crossvit: Cross-attention multi-scale vision transformer for image classification," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 357–366, 2021.

[17] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

[18] Z.-H. Jiang, Q. Hou, L. Yuan, D. Zhou, Y. Shi, X. Jin, A. Wang, and J. Feng, "All tokens matter: Token labeling for training better vision transformers," *Advances in neural information processing systems*, vol. 34, pp. 18590–18602, 2021.