Hierarchical Vision Transformers with Shuffled Local Self-Attentions

Xuan-Thuy Vo, Duy-Linh Nguyen, Adri Priadana and Kang-Hyun Jo
Department of Electrical, Electronic and Computer Engineering,
University of Ulsan
Ulsan (44610), South Korea
Email: xthuy@islab.ulsan.ac.kr; {ndlinh301, priadana3202}@mail.ulsan.ac.kr; acejo@ulsan.ac.kr

2023 International Workshop on Intelligent Systems (IWIS) | 979-8-3503-0504-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/IWIS58789.2023.10284686

***Abstract*—Vision Transformers have reached breakthrough improvements in addressing computer visual fields, for instance, object classification, bounding box localization, semantic/instance pixel-wise predictions, single/multiple tracking, and generative AI models such as GPT-4, SAM, and UniAD. The key success of the Transformers is derived from the flexibility in fulfilling long-range dependencies from raw data and the generalization capability of input-dependent weight adaption. With these properties, Transformer models operated with the self-attention heart and without inductive biases become the new paradigm in processing multiple-modality data. However, the main bottleneck of the Transformer is that global multi-head self-attention layers have high computational costs with the input lengths, e.g., quadratic complexity. When exploiting Transformer-based models on pixel-wise predictions, the cost is not affordable. To deal with this issue, recent methods try to calculate attention weights in local non-overlapped areas and require extra designs that exchange information across windows, for example, window shifting, window expanding, and window sliding. Although these strategies improve accuracy, their implementation is unfriendly and produces additional inference time. Following a line of this research, this paper introduces a new block that consists of non-overlapped local self-attention and overlapped local self-attention. Non-overlapped local self-attention learns interactions inside each window and overlapped local self-attention captures relationships among non-overlapped windows to boost receptive fields and modeling abilities. To be more efficient, both layers are performed in parallel in which each half of the heads is assigned to each layer. Therefore the diversity of the model is enhanced since conventional methods treat all heads equally. Experimental results are conducted and evaluated on the medium dataset, ImageNet-1K. As a result, the proposed approach achieves 77.2% Top-1 accuracy at 5.1M parameters and 0.5 GFLOPs, surpassing lightweight models by clear rooms.**

***Index Terms*—Vision Transformer, Local Self-attention, Image classification**

I. Introduction

Transformer [1] was originally designed for natural language processing and established an important milestone in AI research. With strong modeling capacities, Transformer is scaled to the big models such as BERT [2] and other language models, and achieved remarkable results. In the vision field, DETR [3] explores the benefits of the Transformer encoder and decoder in performing object detection and gets promising performances in both mAP and efficiency. Combining the Transfomer decoder with prior knowledge, e.g., object queries, mask queries, latent queries and track queries, opens flexible directions in solving multi-task learning or foundation models and discards hand-crafted designs of the head such as anchor boxes, NMS, and data association in video data.

Inspired by this trend, Vision Transformer [4] considers a 16×16 square grid as a token and mixes spatial information across patch tokens using the vanilla Transformer encoder. ViT is a non-hierarchical backbone where single-scale features are kept across layers and each layer captures global receptive fields. This paradigm is quite different from existing methods, Convolutional Neural Networks (CNNs) networks. While CNN-based models learn global features at later stages, ViT-based models capture high-level information from the input features at earlier stages and do not require multi-scale features. Convolution layers have strong inductive biases like locality and translation-invariant. Without inductive biases, self-attention layers can learn patch relations and do not degrade performance much. The main advantages of self-attention layers in the ViT model are that it has flexibility in mixing patches globally and the potential to achieve generalization modeling originating from input-dependent weights adaptation. Although ViT has great properties, there are two main problems with the Transformer: quadratic complexity and data-hungry issues.

For the data-hungry issue, ViT-based models need larger data training, ImageNet-21K [5] and JFT-300M [6], to converge the models. To overcome this issue, DeiT [7] applies distillation and strong data augmentations for training pure ViT on only ImageNet-1K and achieves similar results with ViT pre-trained on big datasets. Another disentanglement is to integrate inductive biases of convolution into Transformer: inserting convolution into self-attention operation internally [8]–[11], and combining convolution and self-attention layers externally [12]–[15].

In recent years, the main line of the research has attempted to reduce the model complexity of the Transformer and its adaptation to downstream tasks. PVT [16], DAT [17] downsamples the sizes of key and value features when computing attention maps and introduces multi-scale patch embeddings to build the hierarchical backbone. These designs are well adapted to object localization and semantic/instance segmentation that require high input lengths. However, self-attention in these methods still has quadratic complexity and a query attends to agnostic sets of key and value tokens. Swin Trans-

former [18] constraints self-attention into local windows and information across windows is exchanged through window shifting. Swin Transformer is simple and efficient but the implementation of the window shifting requires extra inference times and stacking more blocks is used to achieve information exchanges and better receptive fields. With these issues, many methods are introduced for efficiently communicating information among windows and enlarging receptive fields, such as window expanding [19], and window sliding in internal [20] and external [21], [22] ways.

Motivated by this direction, this paper proposes a complementary local self-attention to efficiently communicate information across non-overlapped window self-attention. Similar to the baseline Swin Transformer, firstly, the input feature is separated into isolated windows, and global multi-head self-attention is performed in each window. Secondly, in parallel with non-overlapped local self-attention processing, the input feature is shuffled and information between windows is mixed together. Then, the shuffled feature is also partitioned into windows and attention maps are computed by self-attention operation. This step can provide communication among windows in an efficient way through only $\mathrm{reshape}()$ operation. In this work, we implement two layers, non-overlapped window attention and shuffled window attention, in parallel. The number of heads is split into two sub-heads and each half is designated for each attention, respectively. This strategy can increase the diversity of the Transformer since each head extracts specific information from the input. Compared to existing methods, they treat all heads equally, and hence, weaken the diversity of the models because all the heads have similar patterns.

To clarify the capabilities of the proposed Transformer, extensive experiments are conducted and evaluated on the benchmark ImageNet-1K for image classification. Based on our attention designs, the model with 5.1M and 0.5 GFLOPs is proposed. As a result of the model on the validation set, the proposed methods achieve 77.2% Top-1 accuracy that outperforms recent methods by clear margins in both accuracy and throughput, such as MobileViTv2-0.5 by 7%, EdgeViT-XXS by 2.8%, and EMO-2M by 2.1%.

II. Literature Review

A. Vision Transformer

ViT [4] separates the input feature into a set of patches using convolution with kernel 16×16 and stride of 16. A patch 16×16 is viewed as one token and employing a Transformer encoder for globally mixing information across tokens can result in long-range dependencies from the input. DeiT [7] trains the ViT model on smaller datasets [23] by using distillation and strong data augmentation. To make the ViT model affordable for downstream tasks, PVT [16] constructs the hierarchical backbone based on a multi-scale Transformer and patches with various sizes. DAT [17] replaces spatial reduction attention in PVT with deformable attention where key and value features are sampled by adaptive reference points and bilinear interpolation.

Another improvement of ViT-based models is to combine the best from vanilla convolution and self-attention layers. CMT [8] takes full advances of convolution and Transformer by using depthwise convolutions before processing self-attention layers and in Feed-Forward Networks (FFN). ResTv1 [9] investigates the diversity of original multi-head self-attention and utilizes a project layer to mix information along the head dimension. ResTv2 [10] recovers information lost in spatial reduction attention by the up-sampling module. Seaformer [11] reduces the complexity of global ViT to be linear by introducing axial self-attention layers. NextViT [12] combines 3×3 group convolution and global multi-head self-attention layers to build real-time models on edge devices. Similar to CMT, EdgeViT [14] also develops hybrid models that can be applied to mobile devices. MobileViT [13] inserts multi-head self-attention between stages of the MobileNetv2 [24]. iFormer [15] proposes an inception mixer that balances the locality of convolution, max-pooling, and global receptive field of multi-head self-attention.

B. Local Self-Attention

The point of local self-attention is to limit self-attention computations in local windows. HaloNet [20] computes attention areas in which a query attends to local overlapped key and value features. Swin Transformer [18] introduces a successive block: a window Transformer computed within each non-overlapped window and shifted window Transformer where windows are shifted. However, the implementation of shifted windows is unfriendly and results in additional costs. CSWin Transformer [19] enlarges the receptive field of the local vision Transformer by expanding windows to cross-shaped windows and achieves great improvements compared to Swin Transformer. Another line of this research is to replace shifted window Transformer with overlapped depthwise convolution. The depthwise convolution can allow the model to exchange information across non-overlapped windows. MixFormer [21] improves non-overlapped self-attention with two parts: using depthwise convolution as a communication bridge and bidirectional information exchange between depthwise convolution with shared weights across spatial dimension and local self-attention with shared weights across channel dimension. Similarly, EMO [22] unifies spatial token mixer with MLP by implementing non-overlapped self-attention and depthwise convolution in a sequential way at expanded features with high dimensions.

III. Methodology

The hierarchical feature extractor of the SLS network is sketched in Figure 1. Given the input tensor with dimension $3 \times H \times W$, the stem block shrinks the input image by a factor of 4 and increases the number of channels from 3 to C_1. Similar to conventional methods [16], [18], the token mixer - SLS mixer is performed on the feature with spatial dimension reduction by 4, 8, 16, and 32 with respect to stages. The aim of each SLS block is to learn local and global dependencies from the input via non-overlapped local self-attention and

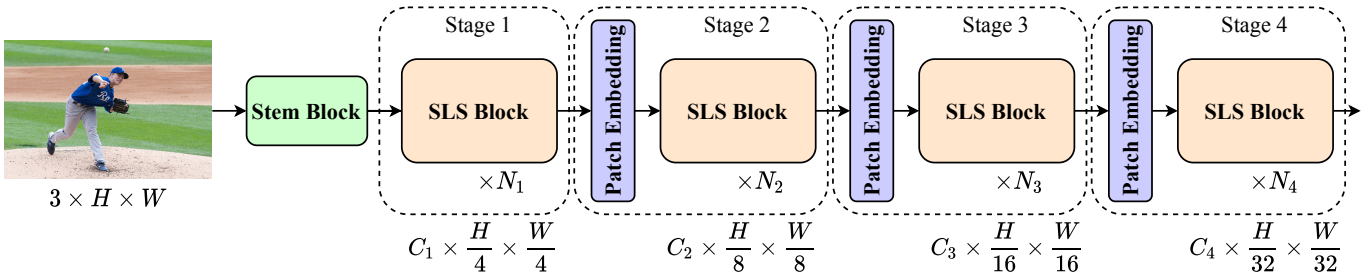

Fig. 1. Flow chart of the proposed SLS. It includes one stem block and four stages. In stem block, two consecutive 3×3 convolutions with a stride of 2 are used to down-sample the images by a factor of 4. In each stage, patch embedding implemented by 3×3 convolution with stride 2 is employed and then, a stack of SLS blocks is developed to learn full information from the input tokens via spatial token mixer - Shuffled Local Self-attention layer and channel token mixer - channel MLP. H, W, C denote 3D dimension of the input tensor, height, width, and channel dimension, respectively. $\{N_1, N_2, N_3, N_4\}$indicate number of stacked SLS blocks across 4 stages.

shuffled local self-attention layers. The overlapped local self-attention layer models spatial interaction inside each window. Meanwhile, a shuffle operation is used to shuffle information of the input feature before window partitions and self-attention computation. The shuffle operation is viewed as a bridge to exchange information across windows.

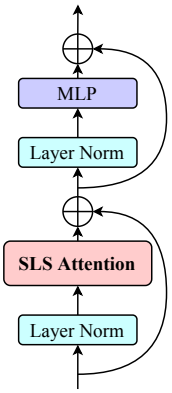

Fig. 2. The detailed architecture of the SLS block.

The SLS block consists of layer normalization, SLS attention, layer normalization, and MLP mixer, followed by common methods [4], [7], [15]–[19]. The illustration is sketched in Figure 2.

A. SLS Attention

Given the input tensor with dimension $H \times W \times C$, two sub-features are generated via channel splitting, denoted by two tensors with shape: $H \times W \times C^{'}$ and $H \times W \times C^{''}$ where $C^{'} + C^{''} = C$. Two sub-features are fed into window attention and shuffle window attention. The detailed illustration of the SLS attention is sketched in Figure 3.

1) Window Attention: Local self-attention allows the model to learn spatial interactions of the image tokens inside local regions. With the feature with dimension $H \times W \times C^{'}$, we divide this feature into non-overlapped regions with the shape

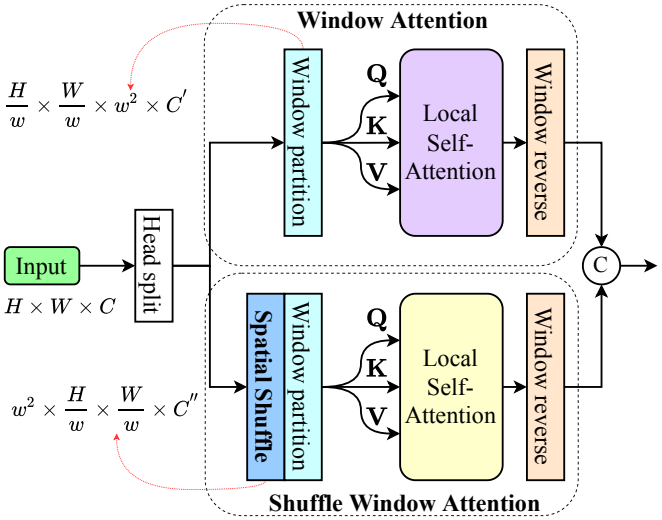

Fig. 3. The detailed architecture of the SLS attention. w is the window size and all implementation is set to 7. $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are query, key, and value tokens projected by linear transformations. C is the concatenation operation.

$\frac{H}{w} \times \frac{W}{w} \times w^2 \times C^{'}$. Then, the self-attention operation is computed within each window $w^2 \times C^{'}$, as follows:

$$\mathrm{SA}(\mathbf{x}^{'}) = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^{\mathbf{T}}}{\sqrt{C^{'}}})\mathbf{V}, \tag{1}$$

where $\mathbf{x}^{'}$ is one window with the shape $w^2 \times C^{'}$. With this formula, the local-self-attention has the locality and translation equivalent identical to convolution. However, there is no connection across windows and its drawback results in inefficient receptive fields and modeling ability.

2) Shuffle Window Attention: In the literature, four solutions introduced to complement non-overlapped local self-attentions are window expanding [19], window sliding [20], window shifting [18], and window shuffling. Figure 4 describes the attention areas of different self-attention versions. In global self-attention, each query attends to the full set of the image token and thus, its computation produces long-range dependencies. Axial self-attention [11] decomposes attention regions into vertical and horizontal areas, and each self-

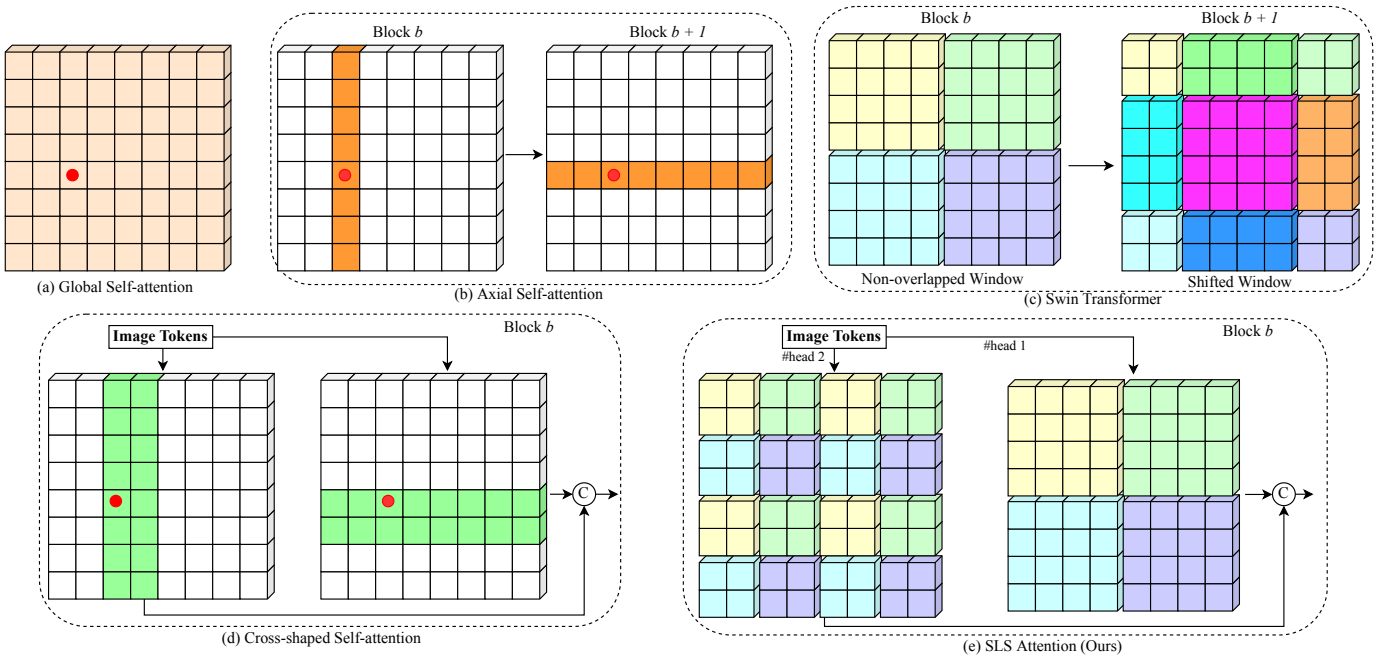

Fig. 4. Comparative architectures: (a) Global self-attention [4], (b) Axial self-attention [11], (c) Swin Transformer [18], (d) Cross-shaped Window attention [19], and (e) Our attention - SLS attention. In our design, the number of heads is divided into two subsets, and each local self-attention is computed on each set of heads: #head1 and #head2. Red points indicate a query position.

TABLE I
COMPARATIVE RESULTS WITH EFFICIENT APPROACHES ON IMAGENET-1K VALIDATION SET

Method	Model Type	Image size	Top-1 Acc. (%)	#params	GFLOPs	Throughput (images/s)
MobileViTv1-XXS [13]	Hybrid	256	69.0	1.3	0.4	7052
MobileViTv2-0.5 [25]	Hybrid	256	70.2	1.4	0.5	6748
PVTv2-B0 [26]	Hybrid	224	70.5	3.7	0.6	6036
Swin-0.7G [18]	Attn	224	74.4	4.4	0.7	2913
MobileViTv1-XS [13]	Hybrid	256	74.8	2.3	1.0	3759
MobileViTv2-0.75 [25]	Hybrid	256	75.6	2.9	1.0	4504
EdgeViT-XXS [14]	Hybrid	256	74.4	4.1	0.6	3954
tiny-MOAT-0 [27]	Hybrid	224	75.5	3.4	0.8	-
Swin-1G [18]	Attn	224	77.3	7.3	1.0	2702
SLS (Ours)	*Attn*	*224*	*77.2*	*5.1*	*0.5*	*4618*

attention is computed on each long row or column. Swin Transformer [18] limits attention areas inside each local window and requires further shifted windows to establish the new set of windows. CSWin Transformer [19] expands the shape of windows from square to cross-shaped windows to significantly enlarge modeling ability. Differently, in this paper, performing with non-overlapped self-attention is a shuffled self-attention operation.

Given the input tensor with the shape $H \times W \times C''$, we partition this feature into adaptive windows with the shape $\frac{H}{w} \times \frac{W}{w}$ and apply self-attentions to grid windows with the shape $w \times w$. This corresponds to utilizing self-attention operation on dilated features and producing global spatial interactions of the features. Therefore, information across windows is exchanged significantly instead of requirements of stacked consecutive blocks of non-overlapped and shifted local self-attention. Our design is simple because only using $\mathrm{reshape}()$ and $\mathrm{permutation}()$ can achieve shuffled windows without any complex operations but enjoying global receptive fields.

B. Model Setting

Based on the design of the SLS block, the manual configuration of the model is constructed. The number of channels across 4 stages is set to $\{32, 64, 96, 192\}$, and the number of stacked SLS blocks across 4 stages is configured to $\{2, 2, 10, 6\}$. The expansion ratio in the MLP mixer is set to 4 and remained in all blocks. This kind of model generates 5.1M parameters and 0.5 GFLOPs.

IV. EXPERIMENTS

To clarify the modeling of the SLS Transformer, implementations are carried out on the large-scaled ImageNet-1K dataset. This dataset includes 1.2M/50K training/validation images. Similar to the common procedure, the comparison with other methods is measured on the validation fold.

The SLS model is trained on two A100 GPUs for 300 epochs. The optimizer AdamW is employed with a learning rate = 0.001 and weight decay = 0.005. The model uses the image with size 224×224 and a total batch size of 4096. Following conventional data augmentations [7], [18], some strategies such as Mixup [28], Cutmix [29], RandAugment [30], and stochastic depth [31] are adopted to train and evaluate the SLS Transformer. We use the Pytorch framework and the code baseline Timm [32].

Table I addresses the comparative results between our SLS and other efficient networks. Model type indicates three kinds of models that are pure convolutions (conv), pure self-attention (attn), and the combination of common convolution and global self-attention to propose hybrid deep learning models. The proposed method achieves 77.2% Top-1 accuracy with 5.1M parameters and 0.5 GFLOPs that surpasses MobileViTv1-XXS [13] by 8.2%, MobileViTv2-0.5 [25] by 7%, PVTv2-B0 [26] by 6.7%, Swin-0.7G by 2.8%, MobileViTv1-XS by 2.4%, MobileViTv2-0.75 by 1.6%, EdgeViT-XXS [14] by 2.8%, tiny-MOAT-0 [27] by 1.7%, and similar accuracy with Swin-1G. The throughput (images/second (s)) is tested on GPU V100 Tesla. As a result, the proposed SLS partially runs faster than other efficient methods while getting better accuracy. For example, with similar performance, the SLS network runs two times faster than Swin-1G [18].

V. Conclusion

This paper introduces a simple and efficient local self-attention operation that can result in local and global receptive fields in one layer. The role of non-overlapped window attention is to learn local interactions between tokens inside windows. To capture long-range dependencies from the input feature, shuffled local self-attention is proposed to perform attention on dilated features and produce global spatial interactions in the image tokens. The shuffled local self-attention can be viewed as a communication block that can exchange information across non-overlapped regions. The shuffled local multi-head self-attention is only implemented by $\mathrm{reshape}()$ and does not produce additional latency compared to unfriendly latency in window shifting, and window sliding. Both non-overlapped and shuffled local self-attention works in parallel and thus, leverage parallel computing of GPU devices. As a result, the proposed SLS Transformer achieves better performances when compared with existing efficient methods in both accuracy and throughput. For instance, we achieve similar Top-1 accuracy with Swin-1G while running two times faster than Swin-1G.

In the future, the pre-trained SLS network will be fin-tuned for object recognition tasks such as bounding box localization, and semantic/instance pixel-wise prediction. Scaling the model to bigger and smaller budgets also leaves in future works.

Acknowledgement

This result was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-003).

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[2] 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.

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[5] 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*. Ieee, 2009, pp. 248–255.

[6] 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*, 2017, pp. 843–852.

[7] 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*. PMLR, 2021, pp. 10 347–10 357.

[8] J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, and C. Xu, "Cmt: Convolutional neural networks meet vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 175–12 185.

[9] Q. Zhang and Y.-B. Yang, "Rest: An efficient transformer for visual recognition," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=6Ab68Ip4Mu

[10] ——, "Rest v2: Simpler, faster and stronger," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=2OdAggzzF3z

[11] Q. Wan, Z. Huang, J. Lu, G. YU, and L. Zhang, "Seaformer: Squeeze-enhanced axial transformer for mobile semantic segmentation," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=-qg8MQNrxZw

[12] J. Li, X. Xia, W. Li, H. Li, X. Wang, X. Xiao, R. Wang, M. Zheng, and X. Pan, "Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios," *arXiv preprint arXiv:2207.05501*, 2022.

[13] S. Mehta and M. Rastegari, "Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=vh-0sUt8HlG

[14] J. Pan, A. Bulat, F. Tan, X. Zhu, L. Dudziak, H. Li, G. Tzimiropoulos, and B. Martinez, "Edgevits: Competing light-weight cnns on mobile devices with vision transformers," in *European Conference on Computer Vision*. Springer, 2022, pp. 294–311.

[15] C. Si, W. Yu, P. Zhou, Y. Zhou, X. Wang, and S. YAN, "Inception transformer," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=qf12cWVSksq

[16] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 568–578.

[17] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4794–4803.

[18] 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*, 2021, pp. 10 012–10 022.

[19] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, and B. Guo, "Cswin transformer: A general vision transformer backbone with cross-shaped windows," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 124–12 134.

[20] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens, "Scaling local self-attention for parameter efficient visual backbones," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 894–12 904.

[21] Q. Chen, Q. Wu, J. Wang, Q. Hu, T. Hu, E. Ding, J. Cheng, and J. Wang, "Mixformer: Mixing features across windows and dimensions," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5249–5259.

[22] J. Zhang, X. Li, J. Li, L. Liu, Z. Xue, B. Zhang, Z. Jiang, T. Huang, Y. Wang, and C. Wang, "Rethinking mobile block for efficient attention-based models," 2023.

[23] 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*. Ieee, 2009, pp. 248–255.

[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[25] S. Mehta and M. Rastegari, "Separable self-attention for mobile vision transformers," *Transactions on Machine Learning Research*, 2023. [Online]. Available: https://openreview.net/forum?id=tBl4yBEjKi

[26] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pvt v2: Improved baselines with pyramid vision transformer," *Computational Visual Media*, vol. 8, no. 3, pp. 415–424, 2022.

[27] C. Yang, S. Qiao, Q. Yu, X. Yuan, Y. Zhu, A. Yuille, H. Adam, and L.-C. Chen, "MOAT: Alternating mobile convolution and attention brings strong vision models," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=H0HGljkxQFN

[28] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[29] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6023–6032.

[30] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 702–703.

[31] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer, 2016, pp. 646–661.

[32] R. Wightman, "Pytorch image models," https://github.com/rwightman/pytorch-image-models, 2019.