SlimFormer-3D: A Layer-Adaptive Lightweight Transformer for Efficient 3D Medical Image Segmentation

Yang Hong[1]*, Lei Zhang[2]*, Xujiong Ye[2], and Jianqing Mo[1(✉)]

[1] Guangdong University of Technology, China
[2] University of Exeter, UK
momolon@gdut.edu.cn

Abstract. Transformer-based architectures demonstrate strong performance in medical image segmentation but face challenges due to computational redundancy and overparameterization, limiting their deployment in resource-constrained settings. This study identifies redundant computations at the block level, particularly in the deeper layers of transformer encoders, as well as in the token mixer and MLP within each layer, as quantified by cross-layer activation similarity. To operationalize these insights, we propose SlimFormer-3D, a lightweight U-shaped encoder-decoder framework that prunes redundant computations at a granular level. Using feature similarity metrics: Angular Distance and Centered Kernel Alignment (CKA), we locate minimally impactful layers and introduce gating factors to control token mixer and MLP module activations selectively. Experiments on BTCV, AMOS, and AbdomenCT-1K 3D abdominal CT datasets show SlimFormer-3D achieves competitive Dice scores while significantly reducing computational redundancy by 3.5× and cutting model parameters by approximately 83% compared to UNETR. Ablation studies confirm its balance between accuracy and efficiency, making it a promising solution for real-time 3D medical image segmentation.

Keywords: Computational Redundancy · Transformer Architecture · Medical Image Segmentation.

1 Introduction

Recent studies [6, 5, 12] have integrated Transformers with U-shaped architectures to improve segmentation accuracy and maintain topological continuity in medical images by leveraging long-range dependency modeling.

However, empirical studies have demonstrated that neural networks often suffer from over-parameterization, with a substantial number of weights that contribute minimally to the final output. This observation has spurred widespread research into model lightweighting. In vision tasks, MetaFormer [19] replaced

* These authors contributed equally to this work.

the attention mechanism in the Transformer architecture with the pooling operation, achieving comparable performance and suggesting that the effectiveness of the Transformer is primarily driven by its structural design rather than the attention mechanism. While metaUNETR [13] prunes deeper encoder blocks to reduce costs without sacrificing segmentation performance. Similarly, Zhong et al. [20] proposed PMFSNet, which simplifies the UNet architecture by incorporating PMFS blocks to balance global and local feature processing, thereby reducing the computational complexity of self-attention mechanisms. In a parallel effort, 3D UX-Net [12] employs large-kernel volumetric convolution within the Transformer framework to capture multiscale contextual information, achieving competitive segmentation performance with enhanced efficiency. As the same Transformer architecture is used in large language models (LLMs), a parallel challenge has emerged in optimizing efficiency for LLMs. Gromov et al. [4] proposed a hierarchical pruning strategy for pretrained large language models, offering new insights into network structure optimization. He et al. [8] found that despite the critical role of attention layers in Transformers, many layers exhibit high similarity and can be pruned without degrading performance. Additionally, Bobby et al. [7] demonstrated that removing Layer Normalization (LN) accelerates model convergence and improves quantization outcomes. Although these two domains employ different embedding strategies. Both findings highlight that the existing Transformer architecture exhibits unnecessary computational inefficiencies. To address model redundancy and enhance efficiency, researchers have proposed various methods, including pruning [10], distillation [18], and quantization [16]. Among these, pruning is of particular interest, as a well-designed pruning strategy can take advantage of the inherent sparsity supported by modern accelerators to enhance memory utilization and computational efficiency.

Current lightweight approaches primarily focus on block-level design, leveraging different variants of self-attention (token mixer) or redundant feature elimination. This strategy may inadvertently discard some essential features and overlook the intrinsic advantages of the Transformer architecture [3]. In this case, our research questions are as follows: Compared to large language models (LLMs) with deep architectures (e.g., 32 layers), does a smaller Transformer-based backbone (e.g., a 12-layer ViT) for vision segmentation tasks exhibit a similar redundancy profile? How can redundancy be accurately identified at a finer, layer-level scale and effectively pruned while preserving model efficiency and performance?

To address these challenges, we employ a validation method based on feature map similarity, leveraging both Angular Distance and Centered Kernel Alignment (CKA) to jointly verify and locate redundant layers. Furthermore, we shift the focus of network optimization toward pruning and refining the Transformer's internal structure, proposing a lightweight architecture SlimFormer-3D. This design implements differentiated architectural modifications for specific layers, enabling layer-level pruning while preserving overall model performance. Our main contributions are summarized as follows: 1)Our findings reveal that redundant computations exist at a fine-grained (layer) level within individual blocks, even

in a small Transformer. Notably, the second layer within a Transformer block exhibits significantly higher redundancy than the first. 2) Based on our findings, we propose SlimFormer-3D, a novel lightweight framework for 3D segmentation that optimizes the Transformer architecture to enhance resource efficiency. 3) Our model achieves competitive performance across different datasets, using only 34% of the parameters required by state-of-the-art methods.

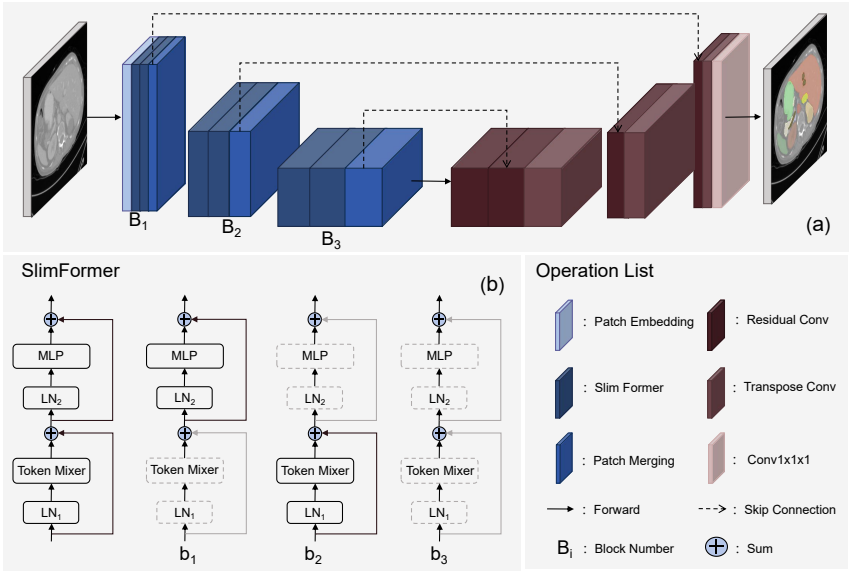

Fig. 1: (a) The overall network architecture of SlimFormer-3D. We use three encoder stages as our backbone. B_1 B_2 and B_3 represent three coding blocks from shallow to deep, respectively. (b) Illustration of three variants of the SlimFormer layer. Specifically, b_1 denotes the SlimFormer layer with the first LayerNormalization and token mixer pruned, b_2 denotes the layer with the second LayerNormalization and MLP pruned, and b_3 corresponds by pruning both sets of components.

2 Method

2.1 Architecture Overview

We propose a lightweight encoder-decoder architecture. As shown in Fig. 1, the encoder stage comprises three encoder blocks built upon the SlimFormer design. By flexibly selecting among various SlimFormer variants, we systematically eliminate redundant components at the layer level and contribute to an overall lightweight design.

Given a CT image $X \in \mathbb{R}^{H\times W\times D\times C}$ with resolution H, W, D, and C channels. The input image is partitioned into a sequence of 3D tokens with a patch size of 2 and projected into an embedding dimension of 48 via the Patch Embedding Layer.

In the encoder stage ($i = 1, 2, 3$), each encoder employs two Slim Former layers to process the input embedding features. This module is designed to reduce computational redundancy while preserving critical local and global semantic relationships. Patch merging operations between blocks downsample the feature maps by half while doubling the channel count.

Residual skip connections between the encoder and decoder help preserve information during upsampling, which is performed via transposed convolution followed by a $1 \times 1 \times 1$ convolution to generate the final segmentation.

2.2 Slim Former: Layer-Adaptive Redundancy Reduction

To reduce redundant computations and more effectively exploit the diverse components within the Transformer architecture, the proposed SlimFormer incorporates two variants: TokenMixer Drop and MLP Drop, is illustrated in Fig. $1(b_1)$ and Fig. $1(b_2)$, respectively. The processing of each layer is determined by the gating factors α_1 and α_2 introduced to control the activation of the token mixer and MLP modules. Specifically, the output of the j th layer of the i th block is given by:

$$Y_{i,j} = \begin{cases} \alpha_1 \cdot \text{TokenMixer}\left(\text{LN}_1(X_{i,j})\right) + X_{i,j}, & \text{if } \alpha_1 = 1, \alpha_2 = 0 \\ \alpha_2 \cdot \text{MLP}\left(\text{LN}_2(X_{i,j})\right) + X_{i,j}, & \text{if } \alpha_1 = 0, \alpha_2 = 1 \end{cases} \tag{1}$$

where $X_{i,j}$, $Y_{i,j}$ represent the input and output of the j th layer of the i th block, respectively. The token mixer module can be replaced with various token mixer architectures, such as Mamba [21] or Attention, and is coupled with a residual connection. LN is applied to stabilize training and ensure consistent scaling of activations across layers.

The gating factor α_1 and α_2 is introduced to flexible control over the processing route: when $\alpha_1 = 1$ and $\alpha_2 = 0$, SlimFormer employs the TokenMixer Drop variant, where the input is processed solely by LN and the token mixer. Conversely, when $\alpha_1 = 0$ and $\alpha_2 = 1$, the MLP Drop variant is activated, and the input is processed only by LN and the MLP. When both α_1 and α_2 are set to zero, the layer is effectively pruned as shown in Fig $1(b_3)$, resulting in no computation. In the case where both $\alpha_1 = 1$ and $\alpha_2 = 1$ in all block(Fig. 1), SlimFormer defaults to the vanilla Transformer architecture, processing the input through both the token mixer and the MLP as in standard practice. This layer-adaptive approach removes redundant computations while maintaining the model's ability to capture complex data relationships in Sec. 3.2.

2.3 Feature Similarity Analysis for Redundancy Localization

We evaluate each layer's contribution by measuring the similarity between its input and output feature maps using Cosine Similarity and Centered Ker-

nel Alignment (CKA). High similarity indicates that a layer makes little change—suggesting redundancy—while low similarity implies significant transformation and importance.

We primarily quantify feature similarity with an Angular Distance-based metric [17, 15]:

$$D(X_{i,j}, Y_{i,j}) = 1 - \frac{1}{\pi} \arccos\left(\frac{x_{i,j} \cdot y_{i,j}}{\|x_{i,j}\|\|y_{i,j}\|}\right) \tag{2}$$

where $||\cdot||$ denotes the L^2-norm and the factor, $\frac{1}{\pi}$ serves as a normalization constant. A lower value of D indicates a lower similarity between the two feature maps, suggesting that the module is high importance and should be preserved; conversely, a higher D implies high similarity, indicating potential redundancy that may be pruned.

To ensure robust detection, we further introduce a CKA-based method [2] to compute the similarity between features:

$$\mathrm{CKA}(X_{i,j}, Y_{i,j}) = \frac{\|X_{i,j}^\top Y_{i,j}\|_F^2}{\|X_{i,j}^\top X_{i,j}\|_F \cdot \|Y_{i,j}^\top Y_{i,j}\|_F} \tag{3}$$

where $X_{i,j}^\top Y_{i,j}$ represents the inner product between the input and output feature matrices, while $\|\|_F$ denotes the Frobenius norm. CKA measures directional similarity by quantifying the correlation between $X_{i,j}^\top Y_{i,j}$. A high CKA value implies high similarity (redundancy), while a low value indicates that the layer significantly transforms its input and should be preserved.

3 Experiments

3.1 Dataset and implementation details

In these experiments, we employ three publicly available 3D abdominal CT datasets with increasing data scales: BTCV [11], AMOS [9] and AbdomenCT-1K [14]. The datasets are strictly divided into training and validation sets following the default splitting ratios.

We implemented SlimFormer-3D using the PyTorch and MONAI frameworks, and all experiments were executed on four RTX 3090 GPUs. The AdamW optimizer was used to train the model for 16K iterations with a learning rate 1e-5 and a weight decay of 1e-4. The input images were cropped to a size of $96 \times 96 \times 96$. The widely used medical image segmentation metric, DICE [1] was employed for evaluation.

We employed two different token mixer backbones with four encoder blocks to precisely identify the locations of redundant computations in Sec. 3.2. Given Mamba's lower parameter count, reduced computational complexity, and highly competitive segmentation performance (Sec. 3.3), we selected Mamba as the token mixer backbone with three encoder blocks. Based on the redundancy analysis at the layer-level in Sec. 3.2, the first layer of each block utilizes a vanilla Transformer layer, while the second layer integrates the MLP Drop version of SlimFormer.

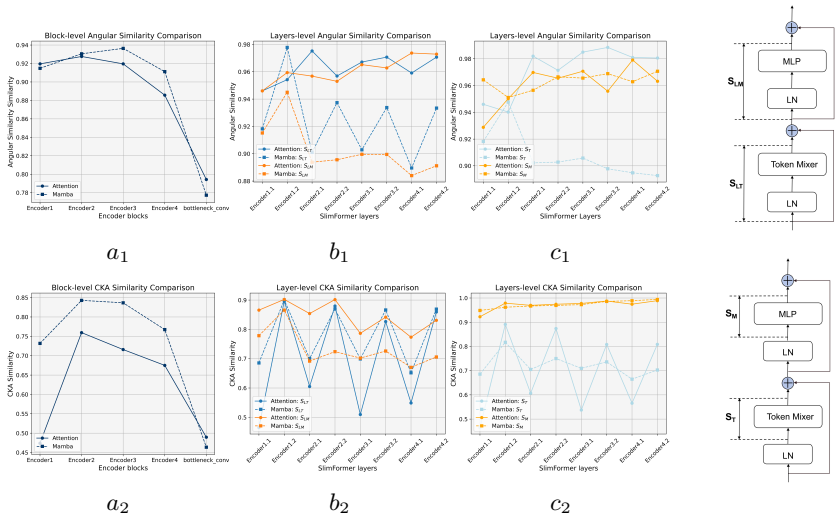

Fig. 2: Feature Similarity of block-level and layer-level: The first row shows the results of Angular similarity, and the second row presents the results of CKA similarity.

3.2 The layer level redundency computation in framework

In Fig. 2, S_{LM} or S_{LT} indicates the similarity between the inputs before LN and the outputs of the token mixer or MLP, respectively. S_M and S_T represent the similarity between the inputs and the outputs of the token mixer and MLP, respectively. The Angular Distance similarity (Fig. 2-a_1) of the translation-invariant feature representations derived from blocks with the two token mixers (Attention and Mamba) varies by only 5%, alongside their competitive performance in Table. 2 (Backbone: Attention vs. Mamba). This suggests that the Transformer architecture contributes more to performance than the token mixer.

Block-level redundancy analysis (Fig. 2-a_1, a_2)shows that representations before and after the bottleneck with convolution undergo a dramatic transition with low similarity, the encoder blocks generally exhibit high feature similarity, indicating the presence of redundant computations. Although the first block shows relatively low CKA similarity due to its translation variance the subsequent deeper blocks align with the Angular Distance similarity results, indicating that pruning deeper blocks would be a viable optimization strategy.

At the layer-level(second column in Fig. 2), we compare the features of two modules in each encoder block: one comprising a LN followed by a token mixer (LN_1–Token Mixer) and the other comprising a LN followed by a MLP (LN_2–MLP) . The results reveal that feature similarity between adjacent layers within the same block follows a pronounced sawtooth pattern (Fig. 2-b_1, b_2), where the second layer in each block exhibits significantly higher redundancy than the first. This finding indicates that redundant computations are not only

confined to deeper blocks but also occur within the individual layers of each block. Furthermore, the experimental results in (Fig. 2-c_1, c_2) demonstrate that the sawtooth pattern persists even after eliminating the influence of LN, confirming that redundancy predominantly exists in the deeper layers within individual blocks.

Table 1: Ablation studies: results of pruning different positions of the Backbone with various strategies

Pruning Strategy	**Block Position**	**Layer Position**	**Parameters (M)**	**FLOPs (G)**	**Dice**
Backbone(Attention)	×	×	81.77	198.93	0.835
Backbone(Mamba)	×	×	71.50	166.18	0.832
LN_2+MLP and LN_1+TokenMixer	4	2	69.06	165.66	0.823
LN_2+MLP and LN_1+TokenMixer	4	1,2	17.80	156.97	0.833
LN_2+MLP	1,2,3,4	2	69.93	158.48	0.828
LN_1+TokenMixer	1,2,3,4	2	69.80	165.89	0.831
LN_2+MLP	1,2,3	2	17.47	149.78	0.826
LN_1+TokenMixer	1,2,3	2	17.36	156.69	0.828

Based on the redundant computation locations revealed by the feature similarity comparisons in Figure. 2, we designed ablation experiments to test various pruning strategies.

First, by replacing the token mixer in the four encoder block backbone as shown in The first part (row 1,2) of Table. 1. We observed that the segmentation Dice score differed only marginally 0.003 DICE, confirming that the Transformer architecture, rather than the token mixer, is the decisive component of the model.

To further validate the presence of deep redundancy at the block level across the entire backbone, we pruned the second layer of the block 4 and an entire block as shown in The second part (row 3 and 4) part of Table. 1. The results revealed a significant reduction in the number of parameters, while the Dice value remained nearly unchanged or even increased by 0.001. This provides strong evidence that redundant computations exist in the deeper encoder blocks, consistent with the feature similarity analysis in Figure. 2.

To further investigate the redundancy within each transformer layer at the layer level—and based on the observation of higher redundancy in the deeper layers of each encoder block—we pruned the token mixer or MLP modules in the second layer of each of the four encoder blocks as shown in The third part (row 5 and 6) of Table. 1. The experimental results indicate that the model performance did not deteriorate after pruning. Subsequently, in a three-encoder-block architecture, we pruned the token mixer or MLP in the second layer of each

block(as shown in row 7 and 8 of Table. 1). With this setting, the number of parameters and the floating point operations were significantly reduced, while the model performance remained virtually unaffected. These results comprehensively demonstrate the existence of redundancy both across encoder blocks and within their deeper layers, thereby supporting the design of the SlimFormer-3D framework.

Table 2: Quantitative comparisons with multi-organ segmentation among SlimFormer-3D. We mark the best results with bold.

Models		UNETR [6]	SwinUNETR [5]	3D UX-NET [12]	SlimFormer-3D
Dice	BTCV	0.791	0.806	0.810	**0.826**
	AMOS	0.794	0.887	0.881	**0.902**
	abdomenCT-1K	0.925	**0.940**	0.938	**0.940**
FLOPs(G)		528.64	331.56	227.51	**149.78**
Parameters(M)		101.79	69.94	50.74	**17.47**

3.3 Comparison with state of the art methods on three datasets

We compared our method with state-of-the-art segmentation models—across three datasets (Table 1). Our results show that our method achieves comparable performance on AbdomenCT-1K (0.940) while significantly boosts performance on BTCV(0.825) and AMOS (0.902), demonstrating its effectiveness. Importantly, SlimFormer-3D achieves a favorable balance between accuracy and efficiency, with low parameter and computation costs.

4 Conclusion

In this study, we introduced SlimFormer-3D, a lightweight framework that reduces computational inefficiencies in Transformer-based medical image segmentation. Using Angular Distance and CKA to measure feature similarity, we identify and prune redundant computations—especially in the second layers of encoder blocks. This layer-adaptive pruning reduces model complexity and computational load by up to 3.5× compared to conventional methods like UNETR while preserving or enhancing segmentation accuracy. Ablation studies confirm that essential information is maintained despite systematic pruning. Our future work will focus on developing more objective similarity measurement metrics for redundancy detection, exploring its underlying causes, and extending our approach to other modalities.

Acknowledgments. This work was supported in part by Guangzhou Key Research and Development Program under Grant (202206010130).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bertels, J., Eelbode, T., Berman, M., Vandermeulen, D., Maes, F., Bisschops, R., Blaschko, M.B.: Optimizing the dice score and jaccard index for medical image segmentation: Theory and practice. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part II 22. pp. 92–100. Springer (2019)
2. Davari, M., Horoi, S., Natik, A., Lajoie, G., Wolf, G., Belilovsky, E.: On the inadequacy of cka as a measure of similarity in deep learning. In: ICLR 2022 Workshop on Geometrical and Topological Representation Learning (2022)
3. Ganesh, P., Chen, Y., Lou, X., Khan, M.A., Yang, Y., Sajjad, H., Nakov, P., Chen, D., Winslett, M.: Compressing large-scale transformer-based models: A case study on bert. Transactions of the Association for Computational Linguistics **9**, 1061–1080 (2021)
4. Gromov, A., Tirumala, K., Shapourian, H., Glorioso, P., Roberts, D.: The unreasonable ineffectiveness of the deeper layers. In: The Thirteenth International Conference on Learning Representations (Oct 2024)
5. Hatamizadeh, A., Nath, V., Tang, Y., Yang, D., Roth, H.R., Xu, D.: Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In: International MICCAI brainlesion workshop. pp. 272–284. Springer (2021)
6. Hatamizadeh, A., Tang, Y., Nath, V., Yang, D., Myronenko, A., Landman, B., Roth, H.R., Xu, D.: Unetr: Transformers for 3d medical image segmentation. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 574–584 (2022)
7. He, B., Noci, L., Paliotta, D., Schlag, I., Hofmann, T.: Understanding and minimising outlier features in neural network training (Nov 2024). https://doi.org/10.48550/arXiv.2405.19279
8. He, S., Sun, G., Shen, Z., Li, A.: What matters in transformers? Not all attention is needed (Oct 2024). https://doi.org/10.48550/arXiv.2406.15786
9. Ji, Y., Bai, H., Ge, C., Yang, J., Zhu, Y., Zhang, R., Li, Z., Zhanng, L., Ma, W., Wan, X., et al.: Amos: A large-scale abdominal multi-organ benchmark for versatile medical image segmentation. Advances in neural information processing systems **35**, 36722–36732 (2022)
10. Kwon, W., Kim, S., Mahoney, M.W., Hassoun, J., Keutzer, K., Gholami, A.: A fast post-training pruning framework for transformers. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. pp. 24101–24116. NIPS '22, Curran Associates Inc., Red Hook, NY, USA (Nov 2022)
11. Landman, B., Xu, Z., Igelsias, J., Styner, M., Langerak, T., Klein, A.: Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge. In: Proc. MICCAI multi-atlas labeling beyond cranial vault—workshop challenge. vol. 5, p. 12. Munich, Germany (2015)
12. Lee, H.H., Bao, S., Huo, Y., Landman, B.A.: 3d ux-net: A large kernel volumetric convnet modernizing hierarchical transformer for medical image segmentation. arXiv preprint arXiv:2209.15076 (2022)
13. Lyu, P., Zhang, J., Zhang, L., Liu, W., Wang, C., Zhu, J.: MetaUNETR: Rethinking token mixer encoding for efficient multi-organ segmentation. In: Medical Image Computing and Computer Assisted Intervention – MICCAI 2024:

27th International Conference, Marrakesh, Morocco, October 6–10, 2024, Proceedings, Part IX. pp. 446–455. Springer-Verlag, Berlin, Heidelberg (Oct 2024). https://doi.org/10.1007/978-3-031-72114-4_43
14. Ma, J., Zhang, Y., Gu, S., Zhu, C., Ge, C., Zhang, Y., An, X., Wang, C., Wang, Q., Liu, X., et al.: Abdomenct-1k: Is abdominal organ segmentation a solved problem? IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(10), 6695–6714 (2021)
15. Ontañón, S.: An overview of distance and similarity functions for structured data. Artificial Intelligence Review **53**(7), 5309–5351 (Oct 2020). https://doi.org/10.1007/s10462-020-09821-w
16. Rokh, B., Azarpeyvand, A., Khanteymoori, A.: A comprehensive survey on model quantization for deep neural networks in image classification. ACM Trans. Intell. Syst. Technol. **14**(6), 97:1–97:50 (Nov 2023). https://doi.org/10.1145/3623402
17. Tang, C., Lv, J., Chen, Y., Guo, J.: An angle-based method for measuring the semantic similarity between visual and textual features. Soft Computing **23**, 4041–4050 (2019)
18. Yu, R., Liu, S., Wang, X.: Dataset distillation: A comprehensive review. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(01), 150–170 (Jan 2024). https://doi.org/10.1109/TPAMI.2023.3323376
19. Yu, W., Si, C., Zhou, P., Luo, M., Zhou, Y., Feng, J., Yan, S., Wang, X.: MetaFormer baselines for vision. IEEE Trans. Pattern Anal. Mach. Intell. (Jan 2024)
20. Zhong, J., Tian, W., Xie, Y., Liu, Z., Ou, J., Tian, T., Zhang, L.: PMFSNet: Polarized multi-scale feature self-attention network for lightweight medical image segmentation. Computer Methods and Programs in Biomedicine **261**, 108611 (Apr 2025). https://doi.org/10.1016/j.cmpb.2025.108611
21. Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., Wang, X.: Vision mamba: Efficient visual representation learning with bidirectional state space model. ArXiv **abs/2401.09417** (2024), https://api.semanticscholar.org/CorpusID:267028142