SegResMamba: An Efficient Architecture for 3D Medical Image Segmentation

Badhan Kumar Das[1,2] BADHAN.DAS@FAU.DE
Ajay Singh[2] AJAY.SINGH@FAU.DE
Saahil Islam[1,2] SAAHIL.ISLAM@FAU.DE
Gengyan Zhao[3] GENGYAN.ZHAO@SIEMENS-HEALTHINEERS.COM
Andreas Maier[2] ANDREAS.MAIER@FAU.DE
[1] *Siemens Healthineers AG*
[2] *FAU Erlangen Nuremberg*
[3] *Siemens Medical Solutions USA, Inc.*

Editors: Under Review for MIDL 2025

Abstract

The Transformer architecture has opened a new paradigm in the domain of deep learning with its ability to model long-range dependencies and capture global context and has outpaced the traditional Convolution Neural Networks (CNNs) in many aspects. However, applying Transformer models to 3D medical image datasets presents significant challenges due to their high training time, and memory requirements, which not only hinder scalability but also contribute to elevated CO_2 footprint. This has led to an exploration of alternative models that can maintain or even improve performance while being more efficient and environmentally sustainable. Recent advancements in Structured State Space Models (SSMs) effectively address some of the inherent limitations of Transformers, particularly their high memory and computational demands. Inspired by these advancements, we propose an efficient 3D segmentation model for medical imaging called SegResMamba, designed to reduce computation complexity, memory usage, training time, and environmental impact while maintaining high performance. Our model uses less than half the memory during training compared to other state-of-the-art (SOTA) architectures, achieving comparable performance with significantly reduced resource demands.

Keywords: Mamba, State Space Models, Vision Transformer, Medical Image Segmentation

1. Introduction

The Transformer architecture has revolutionized deep learning by effectively modeling long-range dependencies and capturing global context. However, its application to 3D medical imaging datasets presents significant challenges, including high memory requirements, computational complexity, and prolonged training times. These challenges are particularly pronounced in tasks involving large datasets like BraTS (Baid et al., 2021) and BTCV Segmentation (Landman et al., 2015), where training Transformer-based models such as UNETR (Hatamizadeh et al., 2021b) and SwinUnetr (Hatamizadeh et al., 2021a) demands substantial resources. Furthermore, transformer models often struggle with smaller datasets, such as Spleen Segmentation (Antonelli et al., 2022), where their performance is suboptimal. The environmental impact of Transformers, driven by their elevated training times, has raised concerns. This has led to a growing interest in alternative architectures such as structured state space models (SSMs), which reduce computational demands and training time, offering a more efficient solution for medical image analysis.

State-space architectures like Mamba (Gu and Dao, 2024), S4 (Gu et al., 2022), and S4nd (Nguyen et al., 2022) have gained popularity due to their solid foundation in Kalman Filters (Kalman, 1960). In contrast, CNN-based models like U-Net (Ronneberger et al., 2015) and SegResNet (Myronenko, 2018) are effective but have a limited receptive field. Hybrid models like UNETR and SwinUnetr(Hatamizadeh et al., 2021a) combine CNNs and Transformers(Vaswani et al., 2023) to enhance performance, though Transformers remain computationally demanding, limiting their practicality in resource-constrained clinical settings. Numerous studies have adapted Mamba to address this issue by modeling long-range dependencies with innovative selection mechanisms (Zhu et al., 2024; Wang et al., 2024; Liu et al., 2024; Liao et al., 2024; Wang and Ma, 2024).

3D image segmentation methods, such as U-Mamba(Ma et al., 2024) and SegMamba(Xing et al., 2024), leverage hybrid CNN-SSM blocks to combine the local feature extraction capabilities of convolutions with the ability of SSMs to capture long-range dependencies. Inspired by these models, we propose SegResMamba, which uses the benefits of Mamba while further reducing memory consumption and computational requirements, thereby enhancing training efficiency. SegResMamba is a lightweight Mamba-based 3D image segmentation model that offers comparable performance to other state-of-the-art (SOTA) models while significantly increasing overall efficiency. Our approach employs Tri-orientated Mamba (ToM) to enhance long-range contextual understanding, combined with CNNs for effective local feature extraction. A convolution mamba mixed block (CMMB) efficiently captures both local and global features, starting with a convolutional bottleneck and leveraging Mamba's global modeling capabilities.

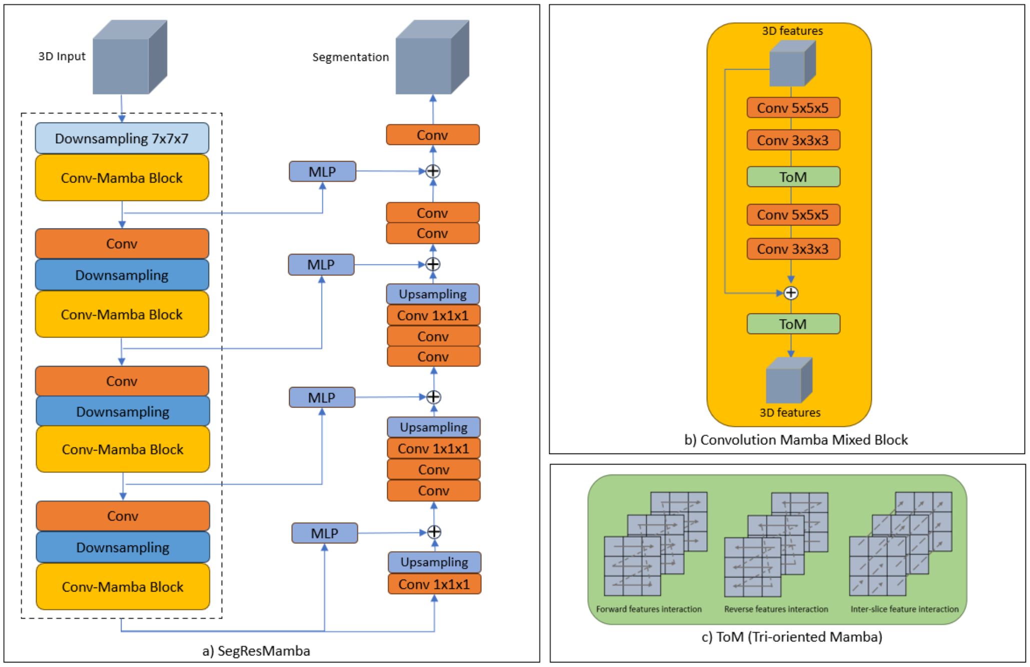
2. Methodology

Our model consists of an encoder, a decoder, and skip connections between the encoder and decoder (Ronneberger et al., 2015) as shown in Figure 1. The encoder utilizes encoder blocks that consist of downsampling layers, convolution mamba mixed blocks consisting of convolution, and tri-oriented Mamba blocks (Xing et al., 2024).

2.1. Encoder

The encoder architecture comprises four cascaded blocks, each meticulously designed to extract hierarchical features from the input 3D medical image while progressively downsampling the spatial dimensions. Each block sequentially employs Downsampling Layers, convolution mamba mixed blocks, and Multi-Layer Perceptron (MLP) (Haykin, 1994) blocks.

Downsampling Layers efficiently reduce the spatial resolution, enabling the capture of essential features with reduced computational cost. Convolution mamba mixed blocks integrates convolutional layers for local feature extraction with Mamba layers for capturing long-range dependencies. This dual approach ensures a comprehensive representation of both local and global image characteristics. Subsequently, another ToM Layer is used with a skip connection from 3D features to further enhance long-range dependency handling,



a) SegResMamba

b) Convolution Mamba Mixed Block

c) ToM (Tri-oriented Mamba)

Figure 1: a) Overview of SegResMamba architecture, b) Convolution Mamba mixed block, and c) Tri-oriented Mamba

which is crucial for accurate 3D image modeling. Finally, the output is normalized and refined through MLP blocks before being fed into a residual connection (He et al., 2015) to facilitate gradient flow during training.

The encoder commences with a downsampling layer applied to the input 3D medical image. The initial downsampling layer employs a Conv3D layer with a larger kernel size of $7 \times 7 \times 7$, a stride of $2 \times 2 \times 2$, and padding of $3 \times 3 \times 3$. This expansive receptive field facilitated by the larger kernel aids in capturing more comprehensive contextual information during feature abstraction.

Subsequently, the remaining three downsampling layers utilize smaller Conv3D kernels with dimensions of $2 \times 2 \times 2$, a stride of $2 \times 2 \times 2$, and padding of $3 \times 3 \times 3$. This adjustment in kernel size balances feature extraction granularity with computational efficiency as the network progresses through deeper layers. We also added an additional convolutional layer before these downsampling operations to preserve essential features.

Following the downsampling layer, feature maps are processed through the convolution mamba mixed block. This block integrates both convolution and Mamba layers to capture multi-scale features. It begins with a larger $5 \times 5 \times 5$ convolution kernel that effectively reduces the spatial dimensions while extracting coarse-grained features. Subsequently, these features pass through a $3 \times 3 \times 3$ convolution layer to refine local details. A ToM Layer is then applied to this refined representation, enabling the abstraction of long-range dependencies and creating a more comprehensive understanding of the local context learned by the convolution filters. The ToM module computes feature dependencies in three distinct directions: forward (z_f), reverse (z_r), and inter-slice (z_s). The ToM module begins by flattening the 3D input features into sequences correspond- ing to each of these directions. This transformation enables the module to effectively capture sequential dependencies within the high-dimensional feature space.

To reverse the spatial reduction process a $3 \times 3 \times 3$ convolution layer and a $5 \times 5 \times 5$ convolution layer are used which restore the feature map's spatial dimensions to their original size. Following that, the extracted features are added to the original features passed through a skip connection. The combined feature maps are then fed into another ToM layer to further capture long-range dependencies and refine the representation.

After this, we apply an MLP block coupled with Instance Normalization (Ulyanov et al., 2017) to normalize the activations and improve stability during training. Notably, these intermediate outputs are directly utilized in the decoder.

2.2. Decoder

The decoder leverages both the encoded features from the encoder and the intermediate results from the encoding process. The decoder is structured with three distinct upsampling stages, designed to progressively refine and expand the spatial resolution of the features. The main input to the decoder has a shape of 768 in the channel dimension. At each stage, the feature map is upsampled and its channel count is halved. This process uses a $1 \times 1 \times 1$ convolution operation followed by an upsampling layer. Inside the upsampling layer, we use non-trainable linear interpolation from Monai(Cardoso et al., 2022).

Upsampled results are combined with the corresponding intermediate results received during the encoding process. The combined features are processed through a sequence of residual blocks. The residual block consists of the ReLU activation function, Group Norm, and convolution kernel of $3 \times 3 \times 3$. We use two of these blocks and a skip connection from the input of these residual blocks to get an output. This architecture combines efficient upsampling with skip connections and residual learning, allowing it to reconstruct detailed spatial information while maintaining the ability to learn complex features at multiple scales. After getting the output from three decoder blocks we use a transposed convolution layer to get the final segmented output. This design is lightweight, being both memory and computation-efficient.

3. Experiments & Results

3.1. Dataset and Implementation Details

BraTS 2021: The BraTS 2021 dataset(Baid et al., 2021) comprises 1,251 multi-parametric magnetic resonance (mpMRI) brain scans, each annotated with segmentation masks delineating tumorous regions. Each scan includes four modalities: Fluid Attenuated Inversion Recovery (FLAIR), native T1-weighted (T1), post-contrast T1-weighted (T1Gd), and T2-weighted (T2) images. Three recombined regions—the tumor core, the entire tumor, and the enhancing tumor—are used to quantify performance using 5-fold cross-validation.

3D Multi-organ Segmentation (BTCV Challenge): The 3D Multi-organ Segmentation dataset from the BTCV Challenge (Landman et al., 2015) focuses on the segmentation of 13 abdominal organs. The dataset comprises 30 volumetric images, with 24 volumes allocated for training and the remaining 6 reserved for testing and evaluation. Each volumetric image provides detailed 3D representations of abdominal structures, essential for medical imaging and diagnosis. The task involves accurately delineating each of the 13 specified organs within these scans.

Spleen 3d Segmentation: The Spleen 3D Segmentation dataset (Antonelli et al., 2022) focuses on segmenting spleens within portal-venous phase CT scans from patients undergoing chemotherapy treatment for liver metastases. The dataset consists of 61 volumetric CT scans, with 41 scans designated for training and the remaining 20 reserved for testing and evaluation. Each scan provides detailed 3D representations of abdominal anatomy, emphasizing the spleen and its surrounding structures during the portal-venous phase. The segmentation task involves accurately delineating the spleen, which is critical for assessing spleen-related conditions and treatment responses in oncology patients.

We used Dice loss and weighted ADAM optimizer for training. Dice similarity coefficient was used for quantitative evaluations. Our experiments used the PyTorch framework with Monai (Cardoso et al., 2022) for model implementation.

3.2. Results

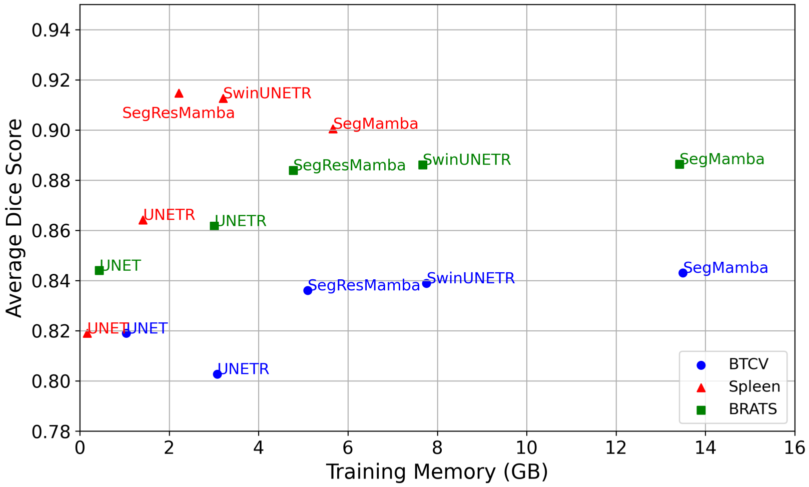


Figure 2: Average Dice Scores for BTCV, Spleen, and BraTS2021 datasets plotted against training memory (in GB) for different models using image size 128 × 128 × 128 for BTCV and BRATS dataset and 96 × 96 × 96 for Spleen dataset with batch size 1.

Figure 2 illustrates the relationship between the peak memory consumption during training and segmentation accuracy, measured by the Average Dice Score, across various models. It can be observed that our method uses comparatively less memory than other large models like Swin Unetr and SegMamba while still maintaining comparable performance.

Brain tumor segmentation performances of different SOTA models are shown in Table 1. SegResMamba achieves a competitive mean Dice score of 0.8839, which is comparable to models like SwinUNETR (0.8861) and SegMamba (0.8863). Despite this, SegResMamba operates with significantly lower Multiply-Accumulate operations (MACs), making it a more computationally efficient model. This lower computational cost, combined with its strong performance, highlights SegResMamba as an excellent choice for scenarios requiring a balance between accuracy and resource efficiency.

Model	MACs	Mean Dice	Dice TC	Dice WT	Dice ET
UNETR	203.29G	0.8617	0.8653	0.8708	0.8490
SegMamba	**1575.13G**	0.8863	0.8943	0.8962	0.8685
UNET	30.13G	0.8444	0.8435	0.8637	0.8260
SwinUnetr	792.08G	0.8861	0.8907	0.8970	0.8707
SegResMamba	340.52G	0.8839	0.8953	0.8958	0.8605

Table 1: Mean dice scores of different models on BraTS21 dataset for 5-fold cross-validation. Dice TC, Dice WT, and Dice ET represent the Dice scores for Tumor Core, Whole Tumor, and Enhancing Tumor, respectively.

Table 2 shows that SegResMamba, while having a reduced memory footprint and lower MACs, delivers performance on the BTCV dataset comparable to more memory-intensive computationally expensive models. In the spleen segmentation task, as shown in Table 3, the SegResMamba network achieved the highest average Dice score of **0.9147**, outperforming UNETR (0.8642), UNET (0.8195), and SwinUNETR (0.9126). This highlights SegResMamba's superior performance compared to transformer-based models on a small dataset.

Model	MACs	Avg Dice
UNETR	196.03G	0.8027
SegMamba	1554.86G	0.8430
UNET	60.10G	0.8192
SwinUnetr	784.46G	0.8389
SegResMamba	336.45G	0.8361

Table 2: Average Dice scores of models on the BTCV dataset.

Model	MACs	Avg Dice
UNETR	82.52G	0.8642
SegMamba	655.32G	0.9004
UNET	11.53G	0.8195
SwinUnetr	328.68G	0.9126
SegResMamba	137.84G	0.9147

Table 3: Average Dice scores of models on the Spleen dataset.

Exp	Model	Avg Dice (BTCV)
1	SegMamba Encoder + ResNet-based Decoder	0.8164
2	Exp. 1 + Convolution Mamba Mixed Block	0.8279
3	Exp. 2 + Additional Conv before downsampling	0.8361

Table 4: Average Dice scores of different setups on the BTCV dataset.

To investigate the contribution of various components in our model, we conducted an ablation study on the BTCV dataset, with results shown in Table 4. In the first experiment, a Mamba encoder proposed by SegMamba(Xing et al., 2024) was paired with a lightweight ResNet-based decoder. This helps us to reduce computational complexity and memory efficiency. Next, we replaced the Global Spatial Context (GSC) block used in the SegMamba encoder with our convolution mamba mixed block. This modification leverages both local representation through convolution and global representation via the mamba layer and improves the segmentation performance by 1.15%. Finally, we added a convolutional layer before the downsampling operation to preserve essential features. When combined with the improvements from Experiments 1 and 2, this experiment led to significant performance gains, increasing the Dice score on the BTCV dataset from 0.8164 to 0.8361.

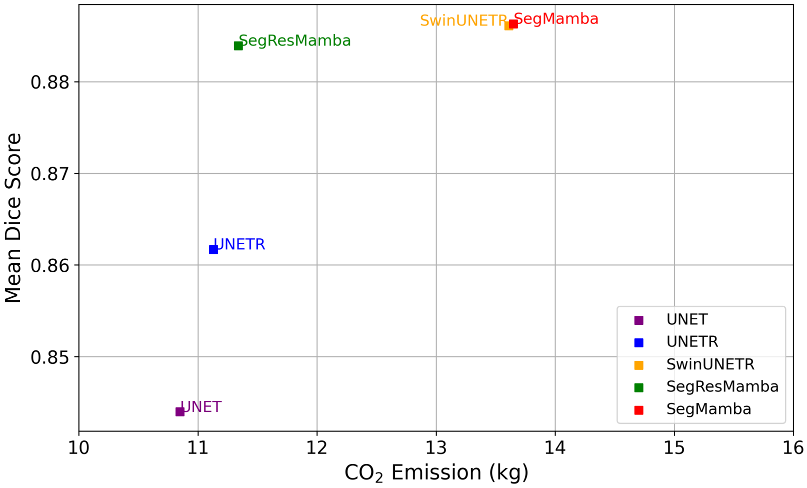


Figure 3: Mean dice score of BraTS dataset against CO_2 emission with 5-fold cross-validation settings for different models.

Furthermore, Figure 3 illustrates the relationship between CO_2 emission and segmentation accuracy for brain tumor segmentation with 5-fold cross-validation across various models. These estimations were conducted using Amazon Web Services in region eu-central-1, which has a carbon efficiency of 0.61 $kgCO_2eq/kWh$. A cumulative training hours of computation was performed on hardware of type A100 PCIe 40GB (TDP of 250W). Estimations were conducted using the Machine Learning Impact calculator presented in (Lacoste et al., 2019).

Among the high-performing models, SegResMamba demonstrates a notable advantage by achieving a balance between environmental efficiency and segmentation performance. Specifically, SegResMamba exhibits significantly lower CO_2 emissions compared to other high-performing models such as SwinUNETR and SegMamba, while maintaining a comparable dice score. Furthermore, when compared to UNET and UNETR, SegResMamba achieves superior segmentation accuracy without a substantial increase in CO_2 emissions, highlighting its efficiency and effectiveness.

4. Discussion

The experimental results demonstrate that SegResMamba is a robust and efficient model for 3D medical image segmentation tasks. It consistently delivers competitive performance across datasets while significantly reducing memory consumption and computational costs compared to state-of-the-art models like SwinUNETR and SegMamba. The model's design prioritizes memory efficiency without compromising segmentation accuracy. The reduced training memory requirements make this model an excellent choice for training and deployment on less resource-intensive hardware.

In terms of computational complexity, SegResMamba requires only 340.52 GMACs for the BraTS21 dataset (Table 1), a significant improvement over SegMamba (1575.13 GMACs) and SwinUNETR (792.08 GMACs). Despite its lightweight design, SegResMamba maintains a competitive mean Dice score of 0.8839, only 0.24% and 0.22% less than SegMamba and SwinUNETR respectively. This demonstrates the model's ability to achieve high segmentation accuracy while remaining computationally efficient which makes it more suitable to be deployed in energy-sensitive situations.

SegResMamba's performance across datasets further highlights its versatility. On the BTCV dataset, the model achieves Dice scores comparable to memory-intensive counterparts like SegMamba and SwinUNETR (Table 2), while attaining the highest Dice score of 0.9147 on the spleen segmentation task (Table 3). These results emphasize its effectiveness in addressing diverse segmentation challenges.

Environmental efficiency is another key aspect of the proposed model. SegResMamba demonstrates significantly lower CO_2 emissions compared to other SOTA models during training due to reduced memory and computational requirements. This aligns with sustainable AI practices, promoting the development of energy-efficient models that minimize environmental impact without compromising performance.

While SegResMamba demonstrates substantial advantages, there are a few limitations to consider. First, its segmentation performance, although competitive, is marginally lower than other high-performing models like SwinUNETR and SegMamba, as observed in the BraTS and BTCV datasets. This slight trade-off may be a consideration for applications where peak accuracy is critical. Another limitation is the training and evaluation were performed on datasets with well-defined segmentation tasks; performance on more challenging or less-structured datasets remains to be explored.

5. Conclusion

SegResMamba marks a significant advancement in 3D medical image segmentation, balancing efficiency and performance by combining Mamba's global context modeling with convolutional layers for local feature extraction. Its reduced memory overhead, along with improved computational and training efficiency, makes it well-suited for real-world clinical applications, delivering excellent results while remaining resource-efficient. Future work will focus on exploring new training strategies and data augmentation to further enhance segmentation accuracy and generalization across various datasets.

References

Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, et al. The medical segmentation decathlon. *Nature communications*, 13(1):4128, 2022.

Ujjwal Baid, Satyam Ghodasara, Suyash Mohan, Michel Bilello, Evan Calabrese, Errol Colak, Keyvan Farahani, Jayashree Kalpathy-Cramer, Felipe C Kitamura, Sarthak Pati, et al. The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification. *arXiv preprint arXiv:2107.02314*, 2021.

M Jorge Cardoso, Wenqi Li, Richard Brown, Nic Ma, Eric Kerfoot, Yiheng Wang, Benjamin Murrey, Andriy Myronenko, Can Zhao, Dong Yang, et al. Monai: An open-source framework for deep learning in healthcare. *arXiv preprint arXiv:2211.02701*, 2022.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces, 2022. URL https://arxiv.org/abs/2111.00396.

Ali Hatamizadeh, Vishwesh Nath, Yucheng Tang, Dong Yang, Holger R Roth, and Daguang Xu. Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In *International MICCAI brainlesion workshop*, pages 272–284. Springer, 2021a.

Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation, 2021b. URL https://arxiv.org/abs/2103.10504.

Simon Haykin. *Neural networks: a comprehensive foundation.* Prentice Hall PTR, 1994.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

B. Landman, Z. Xu, J. Igelsias, M. Styner, T. Langerak, and A. Klein. Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge. In *Proc. MICCAI Multi-Atlas Labeling Beyond Cranial Vault—Workshop Challenge*, 2015.

Weibin Liao, Yinghao Zhu, Xinyuan Wang, Chengwei Pan, Yasha Wang, and Liantao Ma. Lightm-unet: Mamba assists in lightweight unet for medical image segmentation. *arXiv preprint arXiv:2403.05246*, 2024.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model, 2024. URL https://arxiv.org/abs/2401.10166.

Jun Ma, Feifei Li, and Bo Wang. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024.

Andriy Myronenko. 3d mri brain tumor segmentation using autoencoder regularization, 2018. URL https://arxiv.org/abs/1810.11654.

Eric Nguyen, Karan Goel, Albert Gu, Gordon W. Downs, Preey Shah, Tri Dao, Stephen A. Baccus, and Christopher Ré. S4nd: Modeling images and videos as multidimensional signals using state spaces, 2022. URL https://arxiv.org/abs/2210.06583.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2017. URL https://arxiv.org/abs/1607.08022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Ziyang Wang and Chao Ma. Weak-mamba-unet: Visual mamba makes cnn and vit work better for scribble-based medical image segmentation. *arXiv preprint arXiv:2402.10887*, 2024.

Ziyang Wang, Jian-Qing Zheng, Yichi Zhang, Ge Cui, and Lei Li. Mamba-unet: Unet-like pure visual mamba for medical image segmentation, 2024. URL https://arxiv.org/abs/2402.05079.

Zhaohu Xing, Tian Ye, Yijun Yang, Guang Liu, and Lei Zhu. Segmamba: Long-range sequential modeling mamba for 3d medical image segmentation, 2024. URL https://arxiv.org/abs/2401.13560.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model, 2024. URL https://arxiv.org/abs/2401.09417.

Appendix A. Additional Implementation Details

A.1. Brain Tumor Segmentation

BraTS2021 dataset (Baid et al., 2021) was used for brain tumor segmentation to compare the performance across multiple folds for the SOTA models. We trained 5-fold cross-validation for 200 epochs utilizing strategies like learning rate scheduling with CosineAnnealing, Adam optimizer with weight decay of 1e-5, and gradient scaling. We used dice metric and dice loss as metric and the loss function. Various dataset transforms like foreground cropping, random spatial cropping, random flip with probability 0.5 in each direction, and random intensity scaling were used.

A.2. Multi-organ Segmentation

We conducted experiments on the BTCV dataset for multi-organ segmentation (Landman et al., 2015). The training process ran for 25,000 steps. We utilized the Adam optimizer with a learning rate of 1e-4 for our experiments. Our data transformations included scaling intensity range, orientation adjustment (Orientationd), foreground cropping (CropForegroundd), and spacing adjustment (Spacingd). To optimize the model's performance, we employed DiceLoss as the loss function and evaluated using the dice metric for validation.

A.3. Spleen Segmentation

For the spleen segmentation task, we used the spleen 3D segmentation dataset (Antonelli et al., 2022) and we limited training to 100 epochs. Extending the training to larger epoch numbers, such as 200, results in overfitting due to the relatively small size of the dataset compared to larger datasets like BraTS. Following a similar approach to the aforementioned tasks we used Adam optimizer with a learning rate of 1e-4. Transformations like scaling intensity range, normalizing the orientations of images, foreground cropping, and spacing adjustment were used. DiceLoss was used as the loss function and dice metric as the metric for validation.

Appendix B. Training Time and CO_2 Footprint

Model	**Epoch Time**	**Total Time**	**5-Fold Time**	**CO_2 Emissions (kg)**		
	(in sec)	(in hours)	(in hours)	Azure	Google	Amazon
UNETR	262.83	14.60	73.01	10.40	11.32	11.13
Segmamba	321.50	17.86	89.31	**12.73**	**13.84**	**13.62**
UNET	255.80	14.21	71.06	10.13	11.01	10.84
SwinUNETR	321.39	17.85	89.28	12.72	13.84	13.61
Segresmamba	267.83	14.88	74.40	10.60	11.53	11.35

Table 5: Comparison of models in terms of training time, and CO_2 emissions across different cloud providers for training of 5-fold cross-validation using BraTS dataset

A detailed comparison of CO_2 emissions across different cloud providers, including Amazon Web Services, Google Cloud, and Azure, for 5-fold training of the BraTS dataset, is presented in Table 5. It is important to note that these values represent only the emissions from 5-fold training; incorporating hyperparameter optimization would result in significantly higher CO_2 emissions. These estimations were conducted using the Machine Learning Impact calculator presented in (Lacoste et al., 2019).

Appendix C. Memory Efficiency

A comparison of training memory of different models is shown in Table 6 (using image size $128 \times 128 \times 128$ for BTCV and BraTS dataset and $96 \times 96 \times 96$ for Spleen dataset with batch size 1).

Model	**BTCV (GB)**	**Spleen (GB)**	**BRATS (GB)**
UNETR	3.08	0.14	3.02
SegMamba	13.51	5.68	13.44
UNET	1.42	0.48	1.13
SwinUNETR	7.77	3.21	7.68
SegResMamba	5.10	2.22	4.78

Table 6: Training memory (in GB) for different models on BTCV, Spleen, and BraTS datasets.