CascadeMambaSeg: Mamba-based Medical Image Segmentation for Prostate with Cascaded Decoding

Hyeonwook Kim and Ming Ma*
Department of Graduate Computer Science and Engineering, Yeshiva University, New York, NY, USA

***Abstract*—While Transformers have long been the standard, Mamba-based models are emerging as a powerful rival. They offer a major efficiency boost by processing data with linear time complexity, yet they face a specific hurdle: Mamba inherently views images as one-dimensional sequences. Hybrid Mamba-CNN framework have been shown to bridge this gap and avoid overly complex scanning techniques. These models blend Mamba's computational speed with the Convolutional Neural Networks (CNNs) ability to local features in 2D structures. Meanwhile, loss aggregation from different stages of the decoder has been effectively used in medical segmentation for better stability and prediction power. Building on this progress, we propose a novel Mamba-CNN hybrid framework named CascadeMambaSeg with combined loss for prostate segmentation. Our architecture leverages a MambaMixer layer for efficient extraction of low-level features from prostate scans, while standard convolutional layers are utilized for capturing high-level features and forming the final decoder structure. Experimental results on four publicly available prostate datasets show that our method outperforms existing methods in various benchmarks.**

***Index Terms*—Mamba, cascaded decoder, medical image segmentation, prostate segmentation**

I. Introduction

Prostate cancer's global impact was significant in 2020, with 1.4 million new diagnoses and 375,000 deaths, positioning it as the second most prevalent cancer among men worldwide [1]. Accurately defining the boundaries of the prostate through segmentation is a fundamental step in diagnosing and mapping treatment strategies for prostate cancer. AI-based segmentation methods have shown success in the field of medical segmentation, but current transformer-based methods result in models that are data hungry and computationally expensive.

Mamba-SSM has emerged as a high-speed alternative to Transformers. The Mamba architecture have seen success in computer vision tasks [2], but requires scanning the image from different directions due to its interpretation of the data as 1-dimensional signal. Mamba-CNN hybrid models [3] combine the strengths of both architectures to create more robust vision systems, where CNN layers handle local features while Mamba layers handle global features.

Models typically work with a single loss function at the output, but auxiliary heads can also be attached to intermediate hidden layers to produce their own predictions and loss values. Adding the additional loss values into training has shown to increase performance in segmentation models [4], [5]. However, existing methods for aggregating losses distributes weights equally across all decoder outputs or uses methods that introduce additional computation overhead.

Building on this progress, we propose a novel Mamba-CNN hybrid framework with weighted loss aggregation. We use MambaVision's MambaMixer layer to extract lower level features of the prostate scans, while convolutional layers will be responsible for high level features as well as the decoder. Each decoder will produce a segmentation prediction, of which their loss values are weighted to give more importance to later decoder predictions. The contributions of the paper can be summarized as follows:

- We propose a novel Mamba-CNN hybrid method termed CascadeMambaSeg for medical image segmentation.
- We introduce a cascaded decoder architecture with weighted loss aggregation to give more emphasis on later decoder layers while still allowing earlier layer predictions to contribute to the training process
- We demonstrate the results of our method on four publicly available prostate datasets, outperforming existing methods in various benchmarks.

II. Related Works

A. Convolution based Methods for Segmentation

Initial breakthroughs in neural-based image segmentation were driven by the U-Net's use of skip connections to transfer fine-grained information across its encoder-decoder structure [6]. To address the semantic gap inherent in these connections, U-Net++ implemented a nested dense skip pathway architecture [7]. Additionally, research indicates that the integration of residual learning within these blocks further optimizes the model by accelerating convergence and stabilizing the learning process [8].

B. Mamba & Hybrid Methods for Segmentation

While Transformers are hindered by the quadratic complexity of self-attention, the Mamba architecture [9] offers a more efficient path forward. This has led to the development of several vision-centric adaptations. Notable among these is Vim [10], which maintains linear time complexity by using SSM modules to scan input sequences both forward and backward. Building on this foundation, VMamba [2] introduced a 2D-Selective-Scan mechanism, enabling the model to capture spatial information by scanning in four directions. While Mamba based vision models have seen success, pure Mamba-based approaches require special scanning strategies or additional heuristics to add spacial context to the input data.

*Corresponding author: ming.ma@yu.edu

Hybrid models combining Mamba and CNNs have emerged as a solution for integrated feature learning. EfficientVMamba [3] prioritizes computational efficiency by embedding convolutional layers within the early stages of the network, achieving competitive results on vision benchmarks. Conversely, MambaVision [11] adopts an inverted strategy, placing Mamba-Transformer hybrid blocks in the lower layers and shifting convolutions to the higher stages. Furthermore, MambaVision incorporates a symmetric, SSM-free path specifically designed to bolster global context modeling. Hybrid methods have also seen success in segmentation tasks. U-Mamba [12] represents a hybrid approach that enhances the standard UNet framework through the integration of Mamba-derived scanning modules. Specifically, by utilizing these modules within the encoder, the architecture achieves superior performance in segmenting complex medical structures compared to purely convolutional counterparts.

C. Cascading Decoders & Loss Aggregation

Deep supervision facilitates the learning of multi-level representations by introducing auxiliary losses at multiple scales. Each architectural pathway is tasked with capturing features at distinct resolutions, which allows for a more comprehensive analysis that bridges the gap between local spatial information and high-level global semantics [13]. MicroSegNet [4] takes the combined loss of each decoder layer's segmentation and interpolates the ground truth image to each decoder's scale for calculation. Meanwhile, AttentionUNet [14] concatenates the outputs of each decoder layer before being processed through a convolution layer, allowing the model to learn the relationship between each output by itself. Deep supervision guides intermediate layers during training, but current methods have drawbacks: additive aggregation weights all auxiliary outputs equally regardless of merit, while concatenation increases computational overhead by expanding feature dimensions.

Multi-scale Hierarchical Vision Transformer (MERIT) [15] introduced the multi-stage feature-mixing loss aggregation (MUTATION) strategy for image segmentation. This approach combines all subsets of prediction maps from every network stage and aggregates their respective losses. This approach is computationally efficient, requiring no additional parameters or inference-time overhead. However, this also results in the exponential growth (2^n-1) of prediction maps and gives equal importance to all decoder outputs.

III. Methodology

A. Overview

The architecture of CascadeMambaSeg is shown in Figure 1. Integrating a hybrid design, this architecture utilizes a single CNN encoder followed by four MambaVision encoder layers. This structure is mirrored by cascading decoder layers that incorporate skip connections for enhanced input. As the data progresses through the decoder stages, each level generates an increasingly refined segmentation mask. While MambaVision blocks form the core of the encoder stack, CNN layers are strategically used for the initial encoding phase, the bridge, and the entire decoding sequence. Each decoder produces a segmentation prediction which gets more detail as it moves up the decoder stack.

B. Mamba Layers

In Mamba, a 1D continuous input $x(t) \in \mathbb{R}^{d_{in}}$ is modeled as the output of a linear SSM. The process can be defined through the following equations:

$$\begin{aligned} h'(x) &= Ah(t) + Bx(t) \\ y(t) &= Ch(t) \end{aligned} \tag{1}$$

where $h(t) \in \mathbb{R}^N$ represents the hidden state vector, A, B, C represents the learnable system matrices and $y(t)$ represents the output signal. To handle discrete-time sequences, Mamba uses a discretized version of the continuous SSM. The discretized parameters are defined by:

$$\begin{aligned} \bar{A} &= \exp(\Delta A) \\ \bar{B} &= (\exp(\Delta A) - I)A^{-1}B \end{aligned} \tag{2}$$

Where Δ is the step size (learnable or fixed). The discretized SSM ensures that the model is properly normalized and allows for better computational efficiency. This allows the model to be written as a recurrence, similar to an RNN.

$$\begin{aligned} h_t &= \bar{A}h_{t-1} + \bar{B}x_t \\ y_t &= Ch_t \end{aligned} \tag{3}$$

In standard SSM, the input matrices are time-invariant, limiting the model's ability to focus on specific tokens. To solve for this, Mamba introduces input-dependent gating of the parameters:

$$\begin{aligned} B_t &= f_B(x_t) \\ C_t &= f_C(x_t) \\ \Delta_t &= f_\Delta(x_t) \end{aligned} \tag{4}$$

where f_Δ, f_B, f_C represent neural networks such as linear projections, allowing for dynamic adaptation of the state-space kernel to each token. Because $\bar{A}_t$ is now a function of Δ_t, the model can effectively reset its state or ignore irrelevant inputs by scaling Δ toward zero or infinity. The unrolled recurrence is equivalent to a convolutional form:

$$y_t = \sum_{k=0}^{t} K_{t-k}(x_t)x_k \tag{5}$$

where K is the data-dependent kernel implicitly defined by the recurrence, allowing Mamba's kernel to vary with the input, enabling non-linear, selective long-range interactions.

MambaVision [11] modified the original Mamba mixer to be more suitable for vision tasks. A key change was substituting the causal convolution with a standard convolution, driven by the fact that causal layers only look in one direction, a constraint that limits performance when processing visual data. In addition, MambaVision also utilizes a parallel, SSM-free symmetric branch to prevent content degradation, compensating for the sequential limitations of the SSM. The model merges the outputs from the SSM and symmetric branch

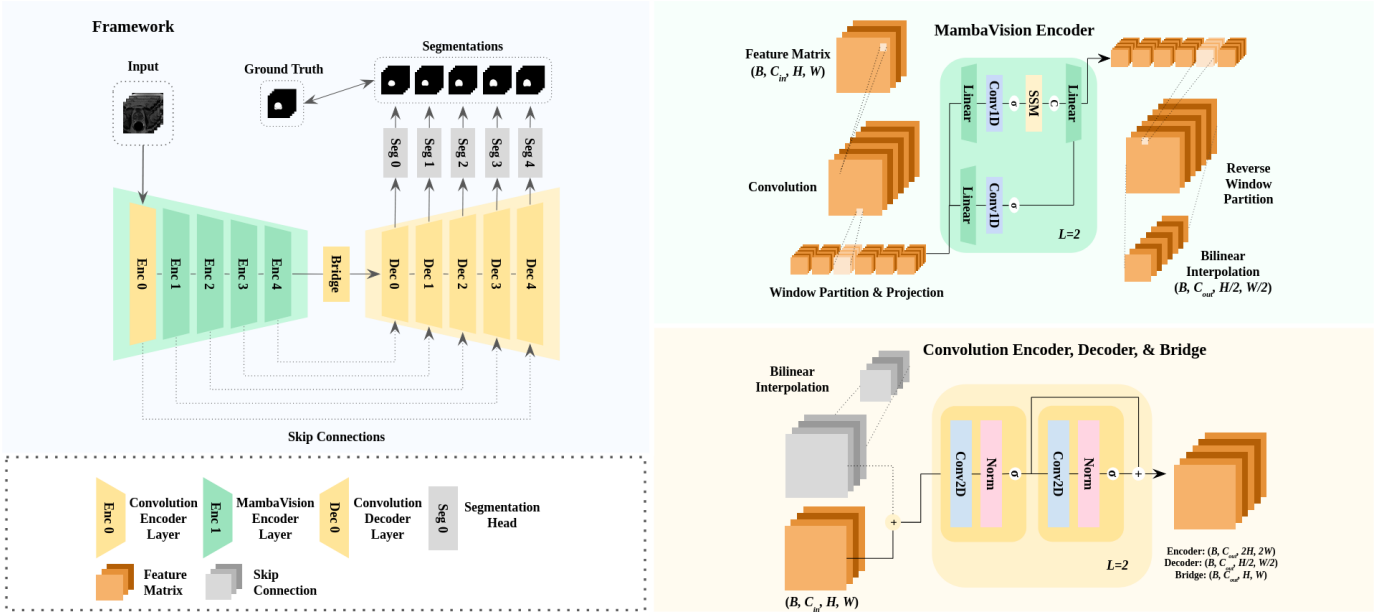

Fig. 1. Overview of the proposed CascadeMambaSeg method. σ denotes the SiLU activation function and $+$ denotes residual connection through addition.

via concatenation, subsequently passing them through a final linear layer for projection. Given an input x_{in}, the output of the MambaVision mixer x_{out} is computed as follows:

$$\begin{aligned} x_1 &= \text{Scan}(\text{SiLU}(\text{Conv1D}(\text{Linear}(C, \frac{C}{2})(x_{in})))) \\ x_2 &= \text{SiLU}(\text{Conv1D}(\text{Linear}(C, \frac{C}{2})(x_{in}))) \\ x_{out} &= \text{Linear}(\frac{C}{2} \times 2, C)\,(\text{Concat}(x_1, x_2)) \end{aligned} \tag{6}$$

$\text{Linear}(C_{\text{in}}, C_{\text{out}})(\cdot)$ denotes a linear transformation with input and output embedding dimensions C_{in} and C_{out}, respectively. The operator Scan refers to the selective scan procedure, and SiLU denotes the activation function. Conv and Concat correspond to one-dimensional convolution and concatenation operations. MambaVision's original implementation is also followed by transformer encoders, but our method chooses to omit them and only use the mamba blocks.

C. Convolutional Layers & Cascaded Decoder

Each convolution uses a 3×3 kernel with stride and padding both set to 1. Downsampling and upsampling is conducted at the end of each encoder and decoder layer, which is conducted simply through bilinear interpolation. Residuals are added at the end of each convolution to improve stability. The process for each convolution layer is as follows:

$$\begin{aligned} \hat{x} &= x + \text{SiLU}(\text{norm}(\text{Conv2D}(x))) \\ x &= \hat{x} + \text{SiLU}(\text{norm}(\text{Conv2D}(\hat{x}))) \\ x &= \text{Interpolate}(x, H, W) \end{aligned} \tag{7}$$

where norm and Interpolate denotes the , Instance Normalization and bilinear interpolation, respectively. Each interpolation takes input x and changes its height and width to (H, W). Conv2D corresponds to a 2D convolution operation with kernel size set to 3 and padding, stride set to 1.

All decoder layers, in addition to the first encoder layer, consist of the CNN layers described above. To get the segmentation outputs from each decoder layer, the height and width of each decoder layer is interpolated to the input dimension and a final convolution is applied. As a result, each decoder layer in the model outputs a predicted segmentation projection.

D. Weighted Combined Loss

The weighted combination of all loss values between each decoder output $pred_i$ and the ground truth gt serves as the loss for our method. The total loss $loss_{total}$ can be formulated as:

$$loss_{total} = \sum_{i=0}^{n} 0.25^i * loss(pred_i, gt) \tag{8}$$

Here, the loss from the later stages are given more weight compared to the loss values from the earlier stages to allow the decoders to capture details hierarchically. Since each decoder layer doubles the feature's height and width dimensions, the weight for each additional decoder's loss value is multiplied by 0.25.

IV. Experiments

A. Dataset and Settings

Evaluation was conducted on four public benchmark datasets consisting of 6697 prostate scans, totaling 5115 training and 1582 testing images. We implemented isolated training cycles for each of the four datasets to confirm patient-level splitting and observe the model's convergence across varied data. NCI-ISBI [16], ProstateX [17] and Promise12 [18] were used for MRI data while CCH-TRUSPS [19] was used for ultrasoind data. Dice Similarity Coefficient (DSC), Intersection over Union (IoU), Hausdorff Distance (HSD) and Average Surface Distance (ASD) were chosen to benchmark the proposed model. These benchmarks are categorized into overlap-based and distance-based evaluations. DSC and IoU

provide a statistical measure of the similarity between two sets, whereas HSD and ASD focus on the spatial distance between the surfaces of the segmented objects. By combining these approaches, the benchmarking process accounts for both the volumetric consistency and the boundary fidelity of the proposed model. The training was conducted for 1000 epochs over a batch size of 4 on Google Colab's T4 GPU. A combination of Binary Cross-Entropy (BCE), DSC and IoU was chosen as the loss function. AdamW was chosen as the optimizer with the learning rate set to 1e-5.

B. Segmentation Performance

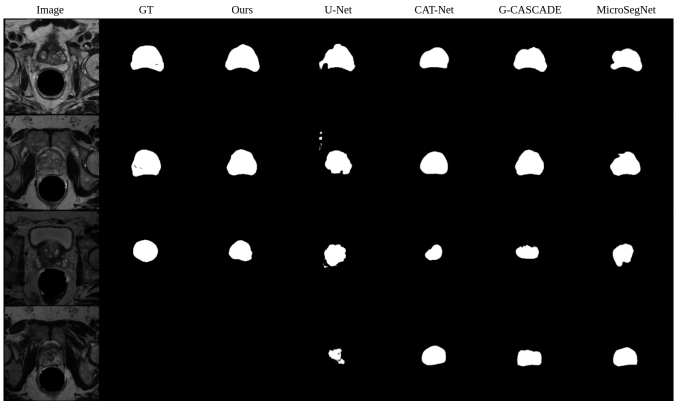

Fig. 2. Visual comparison of proposed method and existing methods.

The segmentation results for our method as well as comparisons with different CNN, Transformer and Diffusion-based models are outlined in Table I. The results of the compared method have been sourced from [29]. The DSC and IoU score for our method outperforms all others for the MRI-Based datasets and and performs the second best for CCH-TRUSPS. These are scores that fall only behind MicroSegNet [4], which also uses a loss aggregation strategy. Meanwhile, the HSD and ASD scores perform better than any of the compared methods for all datasets, indicating better alignment and edge detection compared to existing methods. On average, our method performed 1.2% better on DSC and IoU compared to the best existing method for each dataset. The NCI-ISBI and ProstateX dataset showed the most improvement compared to existing methods. The NCI-ISBI benchmark scored 0.860 DSC, 0.830 IoU, 1.87 HSD and 3.51 ASD on our method. Meanwhile, the ProstateX benchmark scored 0.874 DSC, 0.827 IoU, 2.65 HDS and 1.80 ASD. Both datasers performed over 2% improvement on DSC and IoU scores in comparison to the best performing compared method. Meanwhile, our method scored 0.895 DSC, 0.820 IoU, 3.54 HSD and 2.18 ASD on the PROMISE12 dataset. While it is the best performing model, the performance itself is comparable to MicroSegNet. Finally, our method scored 0.925 DSC, 0.878 IoU, 4.81 HSD and 4.40 ASD on the CCH-TRUSPS dataset, which is the second best performing method behind MicroSegNet. Visual comparisons between our method and existing methods can be seen in Figure 2. Our method showcases improved overlap and edge detection compared to existing methods such as U-Net [6], CAT-Net [23], G-CASCADE [5] and [23] and MicroSegNet.

C. Ablation Studies

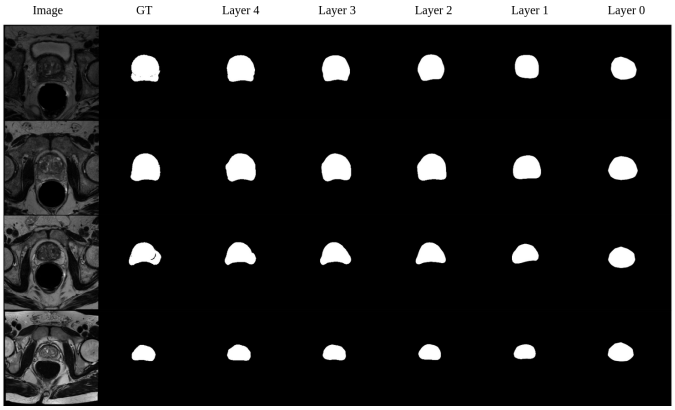

Fig. 3. Segmentation Samples per Decoder Layer.

Table II shows the segmentation performance for each decoder level on the NCI-ISBI dataset. As expected, performance for all benchmarks generally improves as features pass through more decoders before segmentation. The only exception is the Hausdorff Distance metric which shows the lowest score of 1.8543 at layer 3. The biggest performance improvements can be seen in the earlier layers, with the biggest improvement shown between layers 1 and 2 with an increase of 2.78% in Dice score and in 4.01% IoU. Our loss function assigns more favorable weights to later predictions, with decoder layer influence decaying exponentially toward the earlier stages. Nevertheless, the competitive performance of early-stage decoders suggests the model successfully captures fundamental segmentation features early on, allowing subsequent layers to concentrate on refining granular details. This is supported by the visualizations shown in Figure 3, where the initial layers predict the general location of the segmentation while the later decoder layers converge to a more confident prediction.

Table III shows the experiment results for different encoder settings on the NCI-ISBI dataset. Start Block denotes the encoder layer that transitions from CNN layers to MambaVision layers. Before the Start Block, all previous encoder layers consist of CNN layers. Enc1, the method that uses a single CNN layer followed by four MambaVision layers, shows the best performance across all benchmarks with the exception of ASD. The increase in DSC and IoU scores after adding a single CNN layer in the beginning could indicate that they are still important for extracting high-level features of the input image.The ASD scores show a trend of degradation as MambaVision layers get replaced with CNN layers, which could indicate Mamba's importance in precise edge detection.

Table IV shows the benchmarks comparing different activation functions. Using both ReLU or Leaky ReLU shows a significant degredation in performance compared to the SiLU baseline. Both activation functions score a Dice score of below 0.85 and an IoU score of less than 0.815. While ReLU and Leaky ReLU are computationally efficient due to their linear-piecewise nature, they introduce a static slope that can limit the model's representative capacity. SiLU's shape ensures that even very small negative numbers contribute to the learning

TABLE I
SEGMENTATION RESULTS

Datasets	NCI-ISBI [5]				ProstateX [12]				Promise12 [13]				CCH-TRUSPS [6]			
Methods / Metric	**DSC**	**IoU**	**HSD**	**ASD**	**DSC**	**IoU**	**HSD**	**ASD**	**DSC**	**IoU**	**HSD**	**ASD**	**DSC**	**IoU**	**HSD**	**ASD**
Unet [6]	0.822	0.786	2.32	3.99	0.748	0.683	3.41	3.93	0.779	0.676	4.25	6.57	0.898	0.848	5.58	7.43
Unet++ [7]	0.814	0.777	2.30	3.76	0.741	0.682	3.44	3.80	0.81	0.715	4.12	5.06	0.882	0.824	5.92	7.80
TransUnet [20]	0.827	0.789	2.28	3.90	0.851	0.795	2.92	3.33	0.887	0.812	3.65	2.22	0.915	0.874	5.36	4.43
Swin-Unet [21]	0.821	0.782	2.39	5.01	0.792	0.727	3.32	3.49	0.839	0.744	4.09	3.60	0.908	0.857	5.78	5.83
Uctransnet [22]	0.813	0.776	2.38	4.86	0.769	0.701	3.40	2.93	0.875	0.796	3.87	4.62	0.915	0.868	5.61	5.39
G-CASCADE [5]	0.842	0.808	2.24	3.75	0.844	0.795	3.05	2.02	0.880	0.802	3.67	2.67	0.915	0.871	5.58	5.93
CAT-Net [23]	0.841	0.810	2.21	4.04	0.796	0.743	3.31	2.67	0.888	0.813	3.76	2.51	0.895	0.850	5.76	5.01
CCT-Unet [24]	0.836	0.803	2.20	4.50	0.803	0.756	3.08	2.22	0.857	0.775	3.82	3.51	0.902	0.852	5.64	6.98
MicroSegNet [4]	0.829	0.796	2.25	3.86	0.849	0.798	2.96	2.45	0.890	0.817	3.66	2.19	0.928	0.886	5.49	4.72
SegDiff [25]	0.807	0.776	2.27	4.12	0.835	0.788	3.07	1.93	-	-	-	-	0.854	0.788	5.86	7.83
EnDiff [26]	0.814	0.781	2.32	3.82	0.815	0.761	3.24	2.17	-	-	-	-	0.875	0.829	6.04	5.71
DermoSegDiff [27]	0.841	0.806	2.14	3.79	0.853	0.804	2.96	2.02	0.885	0.809	3.69	2.64	0.900	0.855	5.41	4.59
MedSegDiff-V2 [28]	0.828	0.796	2.19	3.71	0.822	0.773	3.10	2.18	0.888	0.815	3.67	2.19	0.844	0.772	6.05	8.55
Ours	**0.860**	**0.830**	**1.87**	**3.51**	**0.874**	**0.827**	**2.65**	**1.80**	**0.895**	**0.820**	**3.54**	**2.18**	**0.925**	**0.878**	**4.81**	**4.40**

TABLE II
ABLATION STUDY: SEGMENTATION PERFORMANCE PER DECODER LAYER

Decoder	DSC	IoU	HSD	ASD
Dec0	0.8027	0.7538	2.2358	5.9796
Dec1	0.8180	0.7755	2.0556	4.5776
Dec2	0.8408	0.8066	1.8790	3.9078
Dec3	0.8526	0.8213	**1.8543**	3.5396
Dec4	**0.8599**	**0.8295**	1.8728	**3.5061**

TABLE III
ABLATION STUDY: DIFFERENT ENCODER SETTINGS

Start Block	DSC	IoU	HSD	ASD	Params
Enc0	0.8453	0.8128	1.9913	3.0932	66.1M
Enc1	**0.8599**	**0.8295**	**1.8728**	**3.5061**	**66.0M**
Enc2	0.8550	0.8237	1.8759	3.5222	65.8M
Enc3	0.8547	0.8208	1.9105	3.8191	65.1M
Enc4	0.8440	0.8122	2.0495	3.4618	62.1M

TABLE IV
ABLATION STUDY: ACTIVATION FUNCTION COMPARISON

Function	DSC	IoU	HSD	ASD
ReLU	0.8497	0.8183	1.9418	3.7608
LeakyReLU	0.8426	0.8102	2.0710	3.6834
SiLU	**0.8599**	**0.8295**	**1.8728**	**3.5061**

process. By providing a nonzero gradient for a wider range of inputs, SiLU facilitates a more expressive latent space, which is particularly critical in deep architectures where vanishing gradients often stifle convergence.

TABLE V
ABLATION STUDY: LOSS RATIO COMPARISON

Base	DSC	IoU	HSD	ASD
0.0	0.8547	0.8104	2.0235	3.7384
0.125	0.8556	0.8145	1.9903	3.6570
0.25	**0.8599**	**0.8295**	**1.8728**	**3.5061**
0.5	0.8565	0.8280	1.8220	3.1909
1.0	0.8514	0.8215	1.9388	3.1451

Results from modifying the base of the loss function is shown in Table V. The defualt value, as shown in equation 8, is 0.25. Lower base values will mean the predictions from the final decoder will have greater weight, while higher values will distribute them equally. A Base of 0 will be equivalent to taking the loss only from the final decoder, while a Base of 1 will take the loss equally from all 5 decoder layers. This effectively generalizes the loss as the following equation:

$$loss_{ablation} = \sum_{i=0}^{4} Base^i * loss(pred_i, gt) \tag{9}$$

Overall, the best performing models tend to give the most amount of weight to the loss from the final decoder while giving smaller but not nonzero weights to the earlier decoder layers. Compared to the baseline of 0.25, the methods with base 0.125 and 0.5 displayed a Dice score of 0.8556 and 0.8565 respectively, and showed an IoU score of 0.8145 and 0.8280, respectively. The methods with the base set at the extreme eds scored the poorest on DSC, IoU and HSD. When the Base was set to 0.0, the model scored the worst HSD and IoU score of all compared methods at 2.0235 and 0.8104, respectively. When the Base was set at 1.0, it showed a DSC score of 0.8514, IoU of 0.8215 and HSD of 1.9388. Both ends of the extreme underperformed when compared to the methods in between for DCSC, IoU and HSD. Interestingly, ASD scores improve as the base increases and puts more weight on the last decoder layer, with the model with 1.0 as the base having 3.1451 ASD, the lowest of the compared methods. This may indicate a performance tradeoff, by taking into account lower level features the model predicts better overlapping features, but at the cost of worse edge detection.

TABLE VI
ABLATION STUDY: INTERPOLATION METHOD COMPARISON

Algorithm	DSC	IoU	HSD	ASD
Nearest	0.8565	0.8249	1.9146	3.2475
Bilinear	**0.8599**	**0.8295**	**1.8728**	**3.5061**
Bicubic	0.8500	0.8190	1.9631	3.4995

Table VI displays the performance of the model dependent on the tensor interpolation method. Nearest-Neighbor, Bilinear and Bicubic Interpolation was chosen for the ablation study with Bilinear as the baseline. Bicubic interpolation performed much worse compared to both Nearest and Bilinear Interpola-

tion with 0.85 DSC, 0.819 IoU, 1.96 HSD and 3.50 ASD. The Nearest-Neighbor parameter showed slightly worse scores for DSC, IoU and HSD with 0.856, 0.825 and 1.91, respectively. However, the ASD score performed the best out of the three interpolation methods with a score of 3.24. Bilinear and Bicubic interpolation results in smoother and softer images when upsampling resulting in a decrease in overall sharpness. This could explain why the Nearest method performs much better in ASD compared to Bilinear and Bicubic Interpolation.

V. Conclusion

In this paper, we presented CascadeMambaSeg, a novel CNN-Mamba hybrid architecture specifically designed for the precise segmentation of the prostate across both MRI and ultrasound modalities. By leveraging the local feature extraction capabilities of CNNs alongside the long-range dependency modeling of State Space Models, our model captures features that are often missed by traditional architectures. To further refine model convergence, we introduced a weighted loss aggregation method. This approach optimizes the learning process by balancing objective functions, leading to performance gains without introducing any additional computational overhead during inference. Future studies could focus on preprocessing such as wavelet transform to extract additional information from the data, or use learnable upscaling strategies in place of linear interpolation.

References

[1] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, "Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians*, vol. 71, no. 3, pp. 209–249, 2021.

[2] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, J. Jiao, and Y. Liu, "Vmamba: Visual state space model," *Advances in neural information processing systems*, vol. 37, pp. 103031–103063, 2024.

[3] X. Pei, T. Huang, and C. Xu, "Efficientvmamba: Atrous selective scan for light weight visual mamba," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 39, pp. 6443–6451, 2025.

[4] H. Jiang, M. Imran, P. Muralidharan, A. Patel, J. Pensa, M. Liang, T. Benidir, J. R. Grajo, J. P. Joseph, R. Terry, *et al.*, "Microsegnet: A deep learning approach for prostate segmentation on micro-ultrasound images," *Computerized Medical Imaging and Graphics*, vol. 112, p. 102326, 2024.

[5] M. M. Rahman and R. Marculescu, "G-cascade: Efficient cascaded graph convolutional decoding for 2d medical image segmentation," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 7728–7737, 2024.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[7] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *International workshop on deep learning in medical image analysis*, pp. 3–11, Springer, 2018.

[8] M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, "Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation," *arXiv preprint arXiv:1802.06955*, 2018.

[9] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in *First conference on language modeling*, 2024.

[10] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," *arXiv preprint arXiv:2401.09417*, 2024.

[11] A. Hatamizadeh and J. Kautz, "Mambavision: A hybrid mamba-transformer vision backbone," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 25261–25270, 2025.

[12] J. Ma, F. Li, and B. Wang, "U-mamba: Enhancing long-range dependency for biomedical image segmentation," *arXiv preprint arXiv:2401.04722*, 2024.

[13] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403, 2015.

[14] R. Mukisa and A. K. Bansal, "Dadu: dual attention-based deep supervised unet for automated semantic segmentation of cardiac images," in *Proceedings of the Future Technologies Conference*, pp. 546–565, Springer, 2024.

[15] M. M. Rahman and R. Marculescu, "Multi-scale hierarchical vision transformer with cascaded attention decoding for medical image segmentation," in *Medical Imaging with Deep Learning*, pp. 1526–1544, PMLR, 2024.

[16] N. Bloch, A. Madabhushi, H. Huisman, J. Freymann, J. Kirby, M. Grauer, A. Enquobahrie, C. Jaffe, L. Clarke, and K. Farahani, "Nci-isbi 2013 challenge: automated segmentation of prostate structures," *The cancer imaging archive*, vol. 370, no. 6, p. 5, 2015.

[17] G. Litjens, O. Debats, J. Barentsz, N. Karssemeijer, and H. Huisman, "SPIE-AAPM PROSTATEx Challenge Data." https://doi.org/10.7937/K9TCIA.2017.MURS5CL, 2017. Accessed on [Date].

[18] G. Litjens, R. Toth, W. Van De Ven, C. Hoeks, S. Kerkstra, B. Van Ginneken, G. Vincent, G. Guillard, N. Birbeck, J. Zhang, *et al.*, "Evaluation of prostate segmentation algorithms for mri: the promise12 challenge," *Medical image analysis*, vol. 18, no. 2, pp. 359–373, 2014.

[19] Y. Feng, C. C. Atabansi, J. Nie, H. Liu, H. Zhou, H. Zhao, R. Hong, F. Li, and X. Zhou, "Multi-stage fully convolutional network for precise prostate segmentation in ultrasound images," *Biocybernetics and Biomedical Engineering*, vol. 43, no. 3, pp. 586–602, 2023.

[20] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306*, 2021.

[21] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-unet: Unet-like pure transformer for medical image segmentation," in *European conference on computer vision*, pp. 205–218, Springer, 2022.

[22] H. Wang, P. Cao, J. Wang, and O. R. Zaiane, "Uctransnet: rethinking the skip connections in u-net from a channel-wise perspective with transformer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, pp. 2441–2449, 2022.

[23] A. L. Y. Hung, H. Zheng, Q. Miao, S. S. Raman, D. Terzopoulos, and K. Sung, "Cat-net: A cross-slice attention transformer model for prostate zonal segmentation in mri," *IEEE transactions on medical imaging*, vol. 42, no. 1, pp. 291–303, 2022.

[24] Y. Yan, R. Liu, H. Chen, L. Zhang, and Q. Zhang, "Cct-unet: A u-shaped network based on convolution coupled transformer for segmentation of peripheral and transition zones in prostate mri," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 9, pp. 4341–4351, 2023.

[25] T. Amit, T. Shaharbany, E. Nachmani, and L. Wolf, "Segdiff: Image segmentation with diffusion probabilistic models," *arXiv preprint arXiv:2112.00390*, 2021.

[26] J. Wolleb, R. Sandkühler, F. Bieder, P. Valmaggia, and P. C. Cattin, "Diffusion models for implicit image segmentation ensembles," in *International conference on medical imaging with deep learning*, pp. 1336–1348, PMLR, 2022.

[27] A. Bozorgpour, Y. Sadegheih, A. Kazerouni, R. Azad, and D. Merhof, "Dermosegdiff: A boundary-aware segmentation diffusion model for skin lesion delineation," in *International workshop on predictive intelligence in medicine*, pp. 146–158, Springer, 2023.

[28] J. Wu, W. Ji, H. Fu, M. Xu, Y. Jin, and Y. Xu, "Medsegdiff-v2: Diffusion-based medical image segmentation with transformer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, pp. 6030–6038, 2024.

[29] T. Liu, M. Zhang, L. Liu, J. Zhong, S. Wang, Y. Piao, and H. Lu, "Cridiff: Criss-cross injection diffusion framework via generative pre-train for prostate segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 102–112, Springer, 2024.