SSMamba: Superpixel Segmentation With Mamba

Xiaohong Jia, Yonghui Li, Jianjun Jiao, Yao Zhao, *Fellow, IEEE*, and Zhiwei Xia

***Abstract*—Deep convolutional networks have achieved remarkable success in superpixel segmentation. However, they only focus on local features ignoring global attributes. The visual Mamba demonstrates an exceptional capability to capture long-range dependencies and offers a lower computational cost compared to the Transformer. Building on this inspiration, we propose a novel superpixel segmentation with Mamba, termed SSMamba. In SSMamba, Mamba is integrated into a global-local architecture, enabling efficient interaction between global attributes and local features to produce high-quality superpixels. The designed activation function further enhances the effectiveness of SSMamba. Extensive experiments on four public datasets demonstrate that SSMamba outperforms existing state-of-the-art methods, achieving competitive average values of ASA = 0.9541, BR = 0.8768, BP = 0.2124, UE = 0.0910, and CO = 0.3698.**

***Index Terms*—Convolutional neural network, mamba, superpixel segmentation.**

I. Introduction

SUPERPIXEL algorithms have become a powerful technique for simplifying image complexity by dividing it into meaningful, coherent regions based on properties such as color, texture, and location proximity. Compared to pixel-based image representation, superpixels better align with human visual perception and reduce redundancy, serving as essential spatial support in various computer vision applications, such as saliency detection [1], [2], target tracking [3], [4], semantic segmentation [5], [6], and others [7], [8]. Driven by their strong adherence to object boundaries and robust generalization capabilities, recent research has increasingly focused on developing efficient algorithms for superpixel segmentation. Superpixel algorithms reported in existing literature can generally be categorized into two groups based on their methodologies: traditional and deep superpixel algorithms.

The traditional superpixel algorithms typically initialize grid cells or seed points, and then iteratively optimize the pixel assignments by k-means, graph cutting, and watershed transform [9], [10], [11], [12]. However, these methods suffer from limitations associated with hand-crafted features, making them challenging to embed into trainable deep learning frameworks. Fortunately, researchers have leveraged deep neural networks for superpixel segmentation, offering a more efficient alternative to traditional algorithms. Early representative algorithms, such as SEAL [13] and SSN [14], utilize a fully convolutional network to predict pixel associations, specifically estimating 9-way probability distributions for all pixels. The backpropagation process in SEAL and SSN is complex due to the combination of deep feature extraction and traditional clustering methods, which involves two distinct stages.

To address this shortcoming, Yang et al. proposed SCN [15], which utilizes a UNet architecture [16] to directly predict the membership between image pixels and predefined regular grid cells. Inspired by SCN, AINet [17] with an association implantation module and ESNet [18] with a Pyramid-gradient structure were subsequently proposed. Compared to AINet, ESNet incorporates edge constraints to suppress contour noise, thereby enhancing superpixel segmentation. Recently, Xu et al. proposed the content disentanglement superpixel (CDS) algorithm [19], which selectively isolates invariant inter-pixel correlations and statistical properties, hence improving the performance of pixel grouping.

The aforementioned deep superpixel algorithms utilize a convolutional neural network (CNN) as their backbone. While CNN is effective for extracting local features, it is limited in capturing global representations. Although Transformer [20], [21] resolves long-range dependencies, its high computational cost poses significant challenges. The visual Mamba [22], [23] inherits the advantages of the Transformer and has higher execution efficiency. Based on this, we propose a superpixel segmentation with Mamba (SSMamba), which utilizes a two-stage strategy to capture both global contextual information and local detailed information. The computational cost of Mamba increases proportionally with feature resolution due to the demands of long-range modeling. Therefore, we adopt CNN to extract high-resolution features that complement the limitation of Mamba.

The contributions of SSMamba are summarized as follows: (1) We propose a novel Mamba-based superpixel segmentation model, named SSMamba, to obtain high-quality superpixels, To the best of our knowledge, SSMamba is the first exploration of a Mamba-based approach for superpixel segmentation. (2) SSMamba is designed to capture long-range coarse-grained contextual information in the first stage, and to mine short-range

Received 6 December 2024; revised 25 March 2025; accepted 31 March 2025. Date of publication 9 April 2025; date of current version 5 May 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62366029, in part by Gansu Province Youth Science and Technology Fund under Grant 23JRRA855, in part by the Key Laboratory of Big Data & Artificial Intelligence in Transportation (Beijing Jiaotong University), Ministry of Education under Grant BATLAB202302, in part by the Key Research and Development Project of Lanzhou Jiaotong University under Grant ZDYF2304, and in part by the Young Scholars Science Foundation of Lanzhou Jiaotong University under Grant 2023006. The associate editor coordinating the review of this article and approving it for publication was Dr. Yuanfang Guo. *(Corresponding author: Yao Zhao.)*

Xiaohong Jia is with the School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China, and also with the Key Laboratory of Big Data and Artiffcial Intelligence in Transportation, Ministry of Education, Beijing Jiaotong University, Beijing 100044, China (e-mail: jiaxhm@163.com).

Yonghui Li, Jianjun Jiao, and Zhiwei Xia are with the School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China (e-mail: li2635339989@163.com; jiaojianjun@lzjtu.edu.cn; xiazhiwei2024@163.com).

Yao Zhao is with the Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China (e-mail: yzhao@bjtu.edu.cn).

Source code is available at https://github.com/jiaxhm/SSMamba.

Digital Object Identifier 10.1109/LSP.2025.3559425

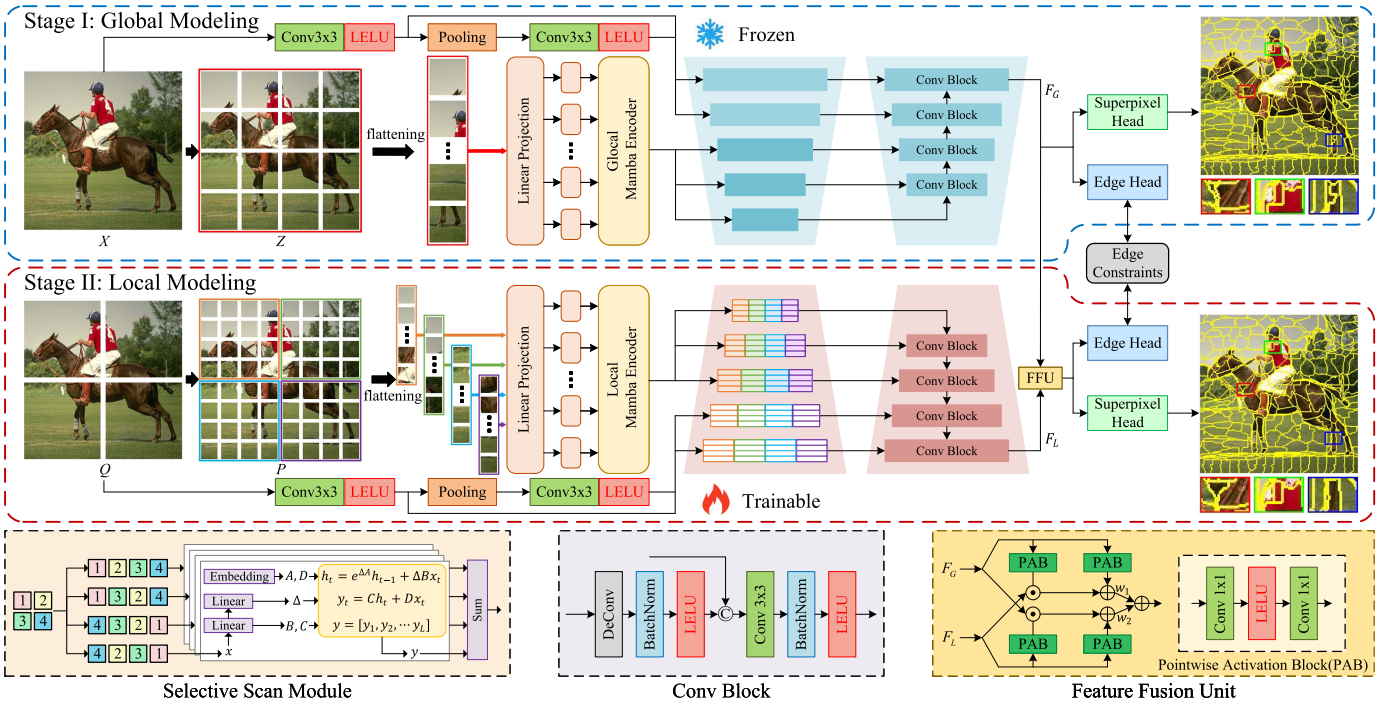


Fig. 1. Overall framework of the proposed SSMamba. In the first stage, we employ CNN and Mamba to compute global attention. In the second stage, similar to the first stage, we use CNN and Mamba to generate local attention. The superpixel head, guided by the edge head, generates the final superpixels by integrating features from both stages using the FFU. In FFU, w_1 and w_2 are learnable weights.

fine-grained local cues in the second stage. (3) The feature fusion unit seamlessly integrates the information extracted from both stages. Meanwhile, the proposed activation function further enhances the overall effectiveness of the SSMamba.

II. Methodology

A. Overview

The overall framework of the proposed SSMamba is depicted in Fig. 1. SSMamba consists of two stages, designed to effectively capture both coarse-grained abstractions and fine-grained details from the input image. In the first stage, CNN processes the original image, while Mamba analyzes a sequence of macro patches obtained through a splitting operation, capturing global contextual information. In the second stage, CNN handles quarter-sized images and Mamba analyzes multiple sequences of micro patches obtained through non-overlapping sliding window sampling, extracting local inherent attributes. Finally, the global and local features are fused by a Feature Fusion Unit (FFU) and passed into a decision head to predict the final superpixel maps.

B. Review Mamba

The Mamba encoders of SSMamba adopt a visual state space model (SSM) from [23]. This model first transforms a 2D image, $X \in \mathbb{R}^{H\times W\times 3}$, where $H \times W$ represents the image resolution, into a sequence of flattened 2D image patches of size $S \times S$. The resulting sequence is then projected into latent embeddings, which is subsequently processed by the Mamba encoder.

The SSM can be regarded as a continuous system, which maps the input sequence $x_t \in \mathbb{R} \longrightarrow y_t \in \mathbb{R}$ through a hidden state $h_t \in \mathbb{R}^N$. The system can be written as follows:

$$\begin{cases} h_t' = \boldsymbol{A}h_t + \boldsymbol{B}x_t \\ y_t = \boldsymbol{C}h_t + Dx_t \end{cases}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ represents the state matrix, $\boldsymbol{B} \in \mathbb{R}^{N\times 1}$, $\boldsymbol{C} \in \mathbb{R}^{1\times N}$, and $D \in \mathbb{R}$ denote the projection parameters, and N is the state size. To implant SSM into deep networks, the discretization operation of (1) is executed. This operation involves converting the continuous parameters $\boldsymbol{A}$ and $\boldsymbol{B}$ into their corresponding discrete counterparts, $\overline{\boldsymbol{A}}$ and $\overline{\boldsymbol{B}}$, using the zero-order hold method as shown below:

$$\begin{cases} \overline{\boldsymbol{A}} = exp\left(\triangle \boldsymbol{A}\right) \\ \overline{\boldsymbol{B}} = \left(\triangle \boldsymbol{A}\right)^{-1}\left(exp\left(\boldsymbol{A}\right) - \boldsymbol{I}\right) \cdot \triangle \boldsymbol{B} \end{cases}. \tag{2}$$

The discretized form of (1), with a step size $\triangle$, can be reformulated as follows:

$$\begin{cases} h_t = \overline{\boldsymbol{A}}h_{t-1} + \overline{\boldsymbol{B}}x_t \\ y_t = \boldsymbol{C}h_t + Dx_t \end{cases}. \tag{3}$$

Furthermore, (3) can be reformulated into an equivalent representation in the form of a CNN as follows:

$$\begin{cases} \overline{\boldsymbol{K}} = \left(\boldsymbol{C}\overline{\boldsymbol{B}}, \boldsymbol{C}\overline{\boldsymbol{A}\boldsymbol{B}}, \cdots \boldsymbol{C}\overline{\boldsymbol{A}}^{L-1}\overline{\boldsymbol{B}}\right) \\ \boldsymbol{y} = \boldsymbol{x} \circledast \overline{\boldsymbol{K}} \end{cases}, \tag{4}$$

where L is the length of the input sequence $\boldsymbol{x}$, $\circledast$ represents the convolution operation, and $\overline{\boldsymbol{K}} \in \mathbb{R}^L$ is a structured convolution kernel.

C. Global Modeling

In the first stage, we employ CNN encoder C_{ge} on the original image X to explore local features, while utilizing Mamba encoder M_{ge} on macro patches Z to mine long-range dependencies. Since M_{ge} requires greater computational resources for high-resolution features, the image is down-sampled before its application. Simultaneously, C_{ge} compensates for the positional error caused by the down-sampling process. The global encoder CM_{ge} is computed as follows:

$$\begin{aligned} CM_{ge} &= \{\{C_{ge}(X)\}, \{M_{ge}(Z)\}\} \\ &= \left\{\left\{F_{gc}^1, F_{gc}^2\right\}, \left\{F_{gm}^1, F_{gm}^2, F_{gm}^3\right\}\right\}, \end{aligned} \tag{5}$$

where $\{F_{gc}^1, F_{gc}^2\}$ denote the low-level features produced by C_{ge}, $\{F_{gm}^1, F_{gm}^2, F_{gm}^3\}$ represent the high-level features generated by M_{ge}. To maintain precision of spatial information, the resolution of F_{gc}^1 is kept consistent with the input image X. The resolutions of $\{F_{gc}^1, F_{gc}^2, F_{gm}^1, F_{gm}^2, F_{gm}^3\}$ decreases exponentially, halving with each successive reduction.

Then, we employ a straightforward CNN decoder C_{gd} to reconstruct the feature maps, as expressed below:

$$F_G = C_{gd}\{CM_{ge}\} = C_{gd}\{\{C_{ge}(X)\}, \{M_{ge}(Z)\}\}, \tag{6}$$

where F_G represents the coarse-grained global features obtained through the integration of CNN and Mamba.

To emphasize edge-awareness, the loss function of the first stage is defined as a weighted combination of the superpixel loss L_{super}^g and the edge loss L_{edge}^g:

$$L_{global} = L_{super}^g(S_h(F_G)) + \lambda L_{edge}^g(E_h(F_G)), \tag{7}$$

where S_h represents superpixel head [15], E_h represents edge head [24], λ balances the superpixel loss L_{super}^g and the edge loss L_{edge}^g. In the subsequent stage, our focus will shift to capturing fine-grained local features.

D. Local Modeling

In the second stage, we adopt CNN encoder C_{le} on quarter images $Q \in \{X_1, X_2, X_3, X_4\} \in X$ to excavate fine-grained features, and utilize Mamba encoder M_{le} on micro patches P to capture short-range dependencies. The local encoder CM_{le} is computed as follows:

$$\begin{aligned} CM_{le} &= \{\{C_{le}(Q)\}, \{M_{le}(P)\}\} \\ &= \left\{\left\{F_{lc}^1, F_{lc}^2\right\}, \left\{F_{lm}^1, F_{lm}^2, F_{lm}^3\right\}\right\}, \end{aligned} \tag{8}$$

where $\{F_{lc}^1, F_{lc}^2\}$ denote the low-level features produced by C_{le}, $\{F_{lm}^1, F_{lm}^2, F_{lm}^3\}$ represent the high-level features generated by M_{le}. Similarly, the decoder process C_{ld} is represented as follows:

$$F_L = C_{ld}\{CM_{le}\} = C_{ld}\{\{C_{le}(Q)\}, \{M_{le}(P)\}\}, \tag{9}$$

where F_L indicates the fine-grained local features derived from the integration of CNN and Mamba.

After training in the first stage, we freeze the parameters from the first stage and proceed to the second stage. Similar to the first stage, the loss function of the second stage is defined as:

$$\begin{aligned} L_{local} = {} & L_{super}^l(S_h(FFU(F_G, F_L))) \\ & + \lambda L_{edge}^l(E_h(FFU(F_G, F_L))), \end{aligned} \tag{10}$$

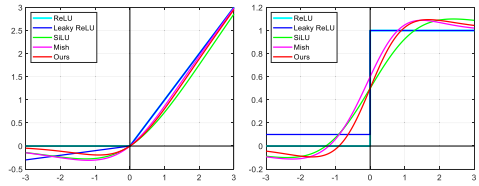


Fig. 2. Comparison of activation functions. From Left to Right: original curves and corresponding first derivatives.

where FFU represents the feature fusion unit, which is employed to incorporate coarse-grained global features and fine-grained local features, enhancing global awareness of local cues while enriching local understanding of global attributes. Here, λ serves as a scaling factor to balance the magnitude between the two losses of L_{super}^l and L_{edge}^l.

E. Activation Function

Additionally, we design a new activation function termed Logarithmic and Exponential Linear Unit (LELU), distinguished by its smoothness, continuity, and non-monotonicity. Mathematically, it is defined as follows:

$$\mathrm{LELU}(\tau) = \frac{\tau \ln\left(1 + e^{\tau}\right)}{\ln\left(1 + e^{\tau}\right) + \ln\left(1 + e^{-\tau}\right)}, \tag{11}$$

where τ represents a neural unit in the input layer. LELU utilizes a self-gating mechanism, multiplying the raw input by the output of a non-linear operation applied to the same input. The proposed activation function exhibits a small and negative tail lengths, facilitating efficient information flow. Meanwhile, it is continuously differentiable, eliminating singularities and ensuring smooth gradient propagation. Eq. (11) can be reformulated as: $LELU(\tau) = \tau softplus(\tau)/(softplus(\tau) - ln(sigmoid(\tau)))$, where $softplus(\tau) = ln(1 + e^{\tau})$, and $sigmoid(\tau) = 1/(1 + e^{-\tau})$. LELU can be regarded as a combination of existing activations. In comparison to Leaky ReLU used in SCN, AINet, and ESNet, LELU offers greater flexibility in accommodating various scenarios.

Fig. 2 demonstrates the differences among various activation functions. As illustrated, LELU is closely related to SiLU [25] and Mish [26]. It is worth noting that LELU approaches 0 faster than both of them. The first derivative of LELU also maintains similar beneficial properties, effectively suppressing unimportant features while avoiding the occurrence of dead neurons. For convenience, we adopt the LELU as the default activation in this letter.

III. Experiments

A. Experiments Settings

Datasets: We evaluate the proposed SSMamba on four publicly available benchmarks from diverse domains: BSDS500 [27], NYUv2 [28], KITTI [29], and DRIVE [30]. BSDS500 is a widely used natural scene dataset for superpixel segmentation, which provides multiple annotated ground truths (GT) for each image. NYUv2 focuses on indoor scenes, KITTI represents street scenes, and DRIVE pertains to medical retinal vessel segmentation. Unlike BSDS500, NYUv2, KITTI,

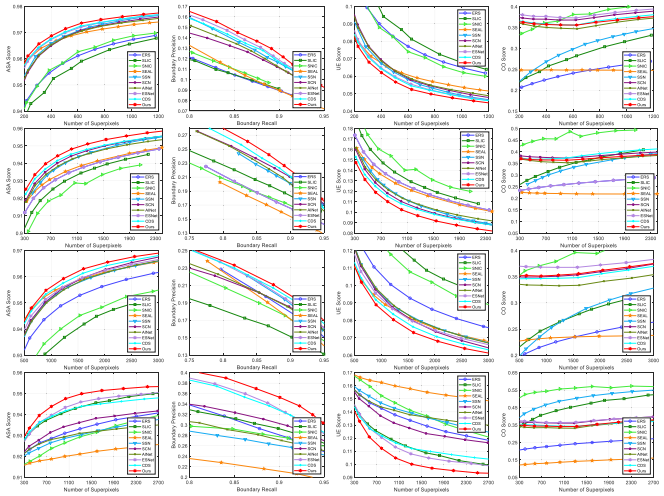
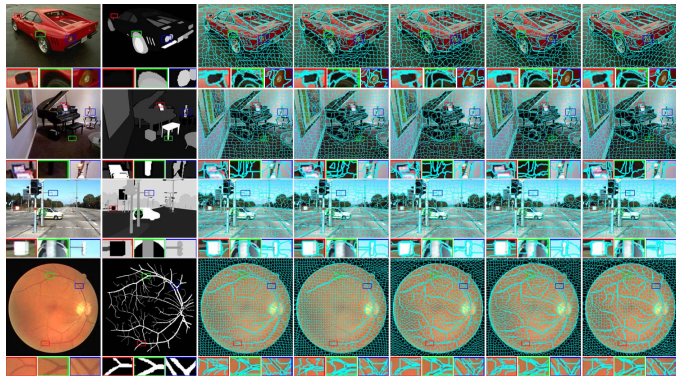


Fig. 3. Metric comparison on four datasets from different domains. From Top to Bottom: BSDS500, NYUv2, KITTI, and DRIVE datasets. From Left to Right: ASA, BR-BP, UE, and CO metrics.



(a) Image (b) GT (c) SCN (d) AINet (e) ESNet (f) CDS (g) Ours

Fig. 4. Visual comparison of different algorithms. From Top to Bottom images from BSDS500, NYUv2, KITTI, and DRIVE datasets. Alternating rows display the segmentation details of each image.

and DRIVE exhibit significant domain shifts, offering a robust benchmark for assessing the generalization capability of SSMamba.

Implementation Details: Our work is based on the PyTorch framework, and its Mamba blocks are initialized using pre-trained weights from VMamba [23]. SSMamba is trained exclusively on the BSDS500 training set and directly generates superpixels for NYUv2, KITTI, and DRIVE without requiring fine-tuning. During the training phase, data augmentation techniques are employed, and random cropping to a fixed size of 208 $\times$ 208. The Adam optimizer is used to optimize the parameters of SSMamba, ensuring efficient convergence, and balancing factor λ is set to 100.

B. Comparison With State-of-The-Art Algorithms

We compare SSMamba against traditional superpixel algorithms, including ERS [9], SLIC [10], and SNIC [11], as well as deep superpixel algorithms including SEAL [13], SSN [14], SCN [15], AINet [17], ESNet [18], and CDS [19]. All the methods are executed using their source code. To assess the quality of superpixels, we utilize four metrics across all datasets: Achievable Segmentation Accuracy (ASA), Boundary Recall-Precision (BR-BP), Under-segmentation Error (UE), and Compactness (CO). For these evaluation metrics, higher ASA, BR-BP, and CO values, alongside lower UE scores, indicate better performance in superpixel segmentation.

Metric comparison. Fig. 3 illustrates the quantitative performance of various algorithms on four datasets. With the help of Mamba, our SSMamba achieves the best ASA, BR-BP, and UE scores. Furthermore, its CO score demonstrates competitive performance, ranking second only to that of CDS. Additionally, compared to global modeling, local modeling shows improvements across several metrics, with CO showing the most significant increase of 5.88% .

Generalization capability. Figs. 3 and 4 report the stable performance of different algorithms across diverse domains. Compared to state-of-the-art algorithms, such as ESNet and CDS, SSMamba exhibits a stronger generalization ability, particularly excelling in the DRIVE dataset.

TABLE I
THE AVERAGE PERFORMANCE COMPARISON OF DIFFERENT ACTIVATION FUNCTIONS ON THE BSDS500 DATASET

Activation	ASA↑	BR↑	BP↑	UE↓	CO↑
ReLU	0.9649	0.8335	0.1352	0.0694	0.3846
Leaky ReLU	0.9649	0.8358	0.1354	0.0695	0.3790
SiLU	0.9662	0.8416	0.1358	0.0668	0.3698
Mish	0.9655	0.8350	**0.1383**	0.0683	**0.3972**
LELU	**0.9668**	**0.8461**	0.1364	**0.0657**	0.3702

The range of the number of superpixels is approximately 50-1900. The best values are in bold.

Visual comparison. Fig. 4 shows the qualitative performance of five algorithms. As observed, the proposed SSMamba generates more accurate and satisfactory object contours compared to competing methods, highlighting the effectiveness of SSMamba.

C. Ablation Study

In addition to the SSMamba framework, we also introduce a new activation function, called LELU. To validate the effectiveness of LELU, our standard model is tested by replacing its activation function. Table I exhibits the average performance comparison of different activation functions. It is evident that LELU slightly outperforms Mish: the former achieves 3 best metrics, while the latter only achieves 2 best metrics.

IV. CONCLUSION

In this letter, we propose a novel superpixel segmentation with Mamba (SSMamba). In SSMamba, global modeling primarily extracts coarse-grained global features, while local modeling focuses on capturing fine-grained local features and interacting with the global features to enhance the overall representation. Additionally, a specifically designed activation function improves the model's expressive power. Experimental results demonstrate that SSMamba surpasses state-of-the-art superpixel algorithms, particularly in terms of generalizability. In future work, we aim to optimize SSMamba by integrating advanced noise filtering techniques [31] and developing adaptive global-local balance mechanisms based on input complexity [32], while also extending its application to real-time processing, such as defect detection and medical diagnosis.

REFERENCES

[1] Y. Qiu, J. Mei, and J. Xu, "Superpixel-wise contrast exploration for salient object detection," *Knowl-Based Syst.*, vol. 292, May 2024, Art. no. 111617.

[2] F. Ma, X. Sun, F. Zhang, Y. Zhou, and H.-C. Li, "What catch your attention in SAR images: Saliency detection based on soft-superpixel lacunarity cue," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5200817.

[3] T. Li, Y. Cai, Y. Zhang, Z. Cai, G. Jiang, and X. Liu, "Superpixel prior cluster-level contrastive clustering network for large-scale urban hyperspectral images and vehicle detection," *IEEE Trans. Veh. Technol.*, vol. 72, no. 2, pp. 2019–2031, Feb. 2025.

[4] L. Wang, H. Lu, and M.-H. Yang, "Constrained superpixel tracking," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 1030–1041, Mar. 2018.

[5] X. Zeng, T. Wang, Z. Dong, X. Zhang, and Y. Gu, "Superpixel consistency saliency map generation for weakly supervised semantic segmentation of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5606016.

[6] Z. Xie, W. Jiang, Y. Yang, and H. Lu, "Superpixel guided network for weakly supervised semantic segmentation," *IEEE Signal Process. Lett.*, vol. 31, pp. 2885–2889, 2024.

[7] X. Zeng, W. Wu, G. Tian, F. Li, and Y. Liu, "Deep superpixel convolutional network for image recognition," *IEEE Signal Process. Lett.*, vol. 28, pp. 922–926, 2021.

[8] Q. Liu, X. Lu, Q. Dong, Y. Zhang, and H. Wang, "SG-SRNs: Superpixel-guided scene representation networks," *IEEE Signal Process. Lett.*, vol. 29, pp. 2038–2042, 2022.

[9] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 2097–2104.

[10] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[11] R. Achanta and S. Susstrunk, "Superpixels and polygons using simple non-iterative clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4651–4660.

[12] T. Lei, X. Jia, T. Liu, S. Liu, H. Meng, and A. K. Nandi, "Adaptive morphological reconstruction for seeded image segmentation," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5510–5523, Nov. 2019.

[13] W.-C. Tu et al., "Learning superpixels with segmentation-aware affinity loss," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 568–576.

[14] V. Jampani, D. Sun, M.-Y. Liu, M.-H. Yang, and J. Kautz, "Superpixel sampling networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 352–368.

[15] F. Yang, Q. Sun, H. Jin, and Z. Zhou, "Superpixel segmentation with fully convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13964–13973.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2015, pp. 234–241.

[17] Y. Wang, Y. Wei, X. Qian, L. Zhu, and Y. Yang, "AINet: Association implantation for superpixel segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7078–7087.

[18] S. Xu, S. Wei, T. Ruan, and Y. Zhao, "ESNet: An efficient framework for superpixel segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 7, pp. 5389–5399, Jul. 2024.

[19] S. Xu, S. Wei, T. Ruan, and L. Liao, "Learning invariant inter-pixel correlations for superpixel generation," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 6351–6359.

[20] M. Pu, Y. Huang, Y. Liu, Q. Guan, and H. Ling, "EDTER: Edge detection with transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1402–1412.

[21] X. Li et al., "Transformer-based visual segmentation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 10138–10163, Dec. 2024.

[22] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," in *Proc. Int. Conf. Mach. Learn.*, 2024, pp. 62429–62442.

[23] Y. Liu et al., "VMamba: Visual state space model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 103031–103063.

[24] Y. Liu et al., "Richer convolutional features for edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 08, pp. 1939–1946, Aug. 2019.

[25] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Netw.*, vol. 107, pp. 3–11, 2018.

[26] D. Misra, "Mish: A self-regularized non-monotonic activation function," in *Proc. Brit. Mach. Comput. Vis.*, 2020, pp. 1–14.

[27] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[28] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 746–760.

[29] H. Abu Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *Int. J. Comput. Vis.*, vol. 126, no. 9, pp. 961–972, 2018.

[30] J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *IEEE Trans. Med. Imag.*, vol. 23, no. 4, pp. 501–509, Apr. 2004.

[31] A. Al Fahoum, "Complex wavelet-enhanced convolutional neural networks for electrocardiogram-based detection of paroxysmal atrial fibrillation," in *Proc. 6th Int. Conf. Adv. Signal Process. Artif. Intell.*, 2024, pp. 158–161.

[32] A. Al Fahoum, "Enhanced cardiac arrhythmia detection utilizing deep learning architectures and multi-scale ECG analysis," *Tuijin Jishu/J. Propul. Technol.*, vol. 44, no. 6, pp. 5539–5548, 2023.