ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10888610

MambaInst: Lightweight State Space Model for Real-Time Instance Segmentation

Zeyu Wang
Computer Science and Technology
Zhejiang Normal University
JinHua, China
14797857499@zjnu.edu.cn

Chen Li
Computer Science and Technology
Zhejiang Normal University
JinHua, China
LilSodaChen@zjnu.edu.cn

Huiying Xu*
Computer Science and Technology
Zhejiang Normal University
JinHua, China
xhy@zjnu.edu.cn

Xinzhong Zhu
Computer Science and Technology
Zhejiang Normal University
JinHua, China
zxz@zjnu.edu.cn

Xiao Huang
College of Education
Zhejiang Normal University
JinHua, China
huangxiao@zjnu.edu.cn

Hongbo Li
Beijing Geekplus Technology
BeiJing, China
Jason.li@geekplus.com

***Abstract*—In this paper, we propose a lightweight and efficient state-space model-based instance segmentation network named MambaInst, which extracts deep semantic features through a LightSSM Block consisting of gating mechanisms and residual connectivity to model long-distance spatial dependencies with linear computational complexity. We design a novel downsampling method called FRDown to efficiently capture contextual information, thereby improving the network's local information perception. With its excellent model architecture and simple training method, MambaInst-B achieves 40.8% in Mask mAP on a single 4090 GPU with an inference time of 2.28 ms on the COCO-seg. Our proposal demonstrates first proof of SSM's effectiveness in real-time instance segmentation, setting a new performance benchmark for Mamba-based techniques in this particular application.**

***Index Terms*—Instance Segmentation, State Space Model**

I. Introduction

Instance segmentation is complex and requires both accurate detection of each and every objects and precise delineation of their boundaries at the same time, which is a very challenging task in computer vision. **C**onvolutional **N**eural **N**etwork (**CNN**) based instance segmentation models [1]–[8] are used in processing image data to extract local features in the image via convolutional kernels, and some models usually have a better balance of speed and performance [9]–[11], however, due to the limited sensory field of CNNs, they have a relative lack of feature coherence to associate instances in high-level visual semantic information, which may lead to poor segmentation results on large objects. **Vi**sion **T**ransformer (**ViT**) based instance segmentation models [12]–[16], which are able to capture global features and naturally model remote semantic dependencies through the self-attention mechanism [17], perform well in distinguishing overlapping instances with the same semantic categories, but they require significant computational resources both in the training and inference.

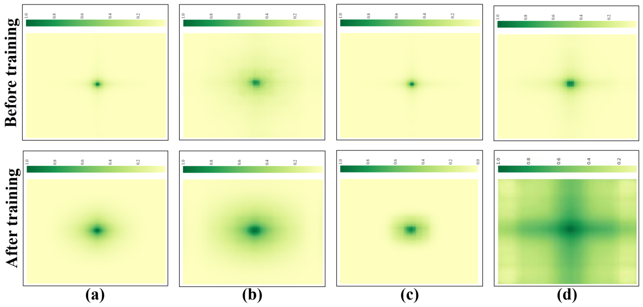

Fig. 1. Effective Receptive Field (ERF) visualization results. High-intensity green pixels indicate a greater response to the center pixel. (a) Mask-RCNN [1]. (b) RTMDet [11]. (c) YOLOv8-seg [31]. (d) MambaInst (Ours).

Recent advancements in the field of natural language processing have seen the emergence of designs based on State Space Model (SSMs) [18]–[20], exemplified by the Mamba [21], it ensure that models maintain high efficiency during selective information processing and achieve linear scalability in sequence length, thereby enhancing computational efficiency when handling long sequence data in language modeling. Despite the successful application of SSMs in various tasks, such as classification [22]–[24], object detection [30], and semantic segmentation [27]–[29], the efficacy of SSM-based approaches in the context of instance segmentation remains underexplored. In this paper, we introduce MambaInst, an innovative model based on the SSM, tailored to the task of instance segmentation, that effectively integrates local and global information while reducing parameter introductions and computational resource consumption. As shown in Fig. 1, we compared MambaInst with representative instance segmentation methods using the **E**ffective **R**eceptive **F**ield (**ERF**) [42], we measured the ERF of the weights before and after model training, and only MambaInst showed global image perception. We propose the **Light**weight **S**tate **S**pace **M**odel (**LightSSM**) Block consisting of a gating mechanism and residual connection, and devise a novel downsampling method called **F**usion

*Corresponding author.

feature **R**esidual **D**ownSampling (**FRDown**). In summary, our contributions are as follows:

- We design a **LightSSM Block** consisting of gating mechanism and residual connection to extract deep semantic features, modeling long distance spatial dependencies with linear computational complexity.
- We design an efficient downsampling module to effectively capture contextual information named **FRDown**, aiming to achieve effective preservation of key details such as boundaries and textures through more efficient feature filtering. Textures through more efficient feature filtering and reorganization mechanisms.
- We design **MambaInst-Tiny/Base (T/B)**, exhaustive experiments on the MSCOCO [39] dataset, and quantitative comparisons with other state-of-the-art methods show the robust performance of MambaInst on various metrics.

II. Related Work

A. *Instance Segmentation Framework*

The rapid advancement of CNNs and their hierarchical feature extraction approach have seen widespread application in instance segmentation. Mask R-CNN [1] established a foundation by generating candidate regions through the Region Proposal Network, while YOLACT [9] introduced a model capable of fast, high-quality mask predictions without requiring resampling operations. SOLQ [2] simplified models by transforming instance segmentation into a single classification problem via the introduction of instance categories. CondInst [5] and QueryInst [7] further improved accuracy and speed by employing dynamic instance-aware networks to generate convolution kernels and driving dynamic mask heads through a query mechanism, respectively. RTMDet [11] enabled efficient instance segmentation using compatible backbone and neck architectures alongside fundamental building blocks for deep convolutional layers.

Although CNNs excel in extracting local features, they inherently struggle with capturing long-range dependencies, a critical challenge in accurate instance segmentation. In contrast, Transformers [12]–[14] have excelled in this area due to their superior long-range modeling capabilities, achieving significant success across various domains, from general-purpose visual backbones to instance segmentation models. Mask DINO [15] builds upon DINO [33] by adding a branch for binary mask prediction in instance segmentation, whereas Mask Frozen-DETR [16] minimizes training time and GPU resource consumption by training an additional lightweight mask network to predict instance masks within bounding boxes provided by a frozen DETR [32] object detector. Nonetheless, Transformer-based models face substantial computational overhead due to the quadratic complexity of the attention mechanism, and their superior performance is highly dependent on large-scale pre-training schemes.

B. *Visual State Space Model*

Mamba [21] utilizes a linear time series modeling approach to effectively balance computational efficiency and model flexibility in language modeling and audio processing tasks. Based on this foundation, SSM [22]–[24] was soon introduced to the vision domain, demonstrating high efficiency in visual representation learning such as image categorization. MambaOut [25] concludes the existence of advantages of Mamba in the downstream dense prediction task of vision from the interpretability of SSM. In the field of medical image segmentation, VM-UNet [26] is the first medical image segmentation framework based on a pure SSM model. RS-Mamba [27] introduces Co-Completion Module to enhance the fusion of dual-encoder features, and demonstrates excellent performance for semantic segmentation on high-resolution remote sensing images. The proliferation of subsequent work on Mamba on various visual tasks [28]–[30] has greatly demonstrated its good scalability, and our work starts from the idea of simplicity and flexibility of the model, and ease of training, with the hope of establishing an SSM based baseline on instance segmentation.

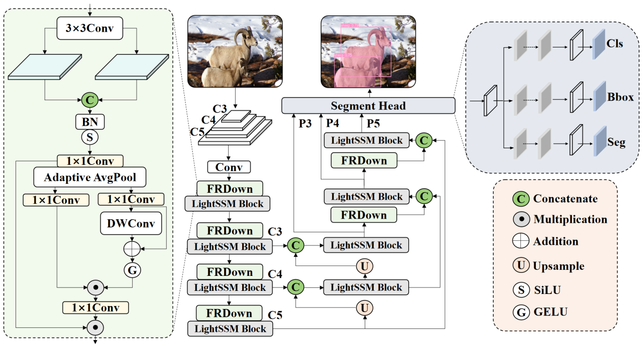

Fig. 2. Illustration of MambaInst architecture.

III. Method

A. *Overall Architecture: MambaInst*

The MambaInst proposed in this paper has a 5-layer structure with the number of channels set to $\{64,\ 128,\ 256,\ 512,\ 1024\}$, and adopts the Ushaped architecture, which mainly consists of three parts: the backbone, the PAFPN and the segmentation head. Specifically, the first layer uses the standard convolution with convolution kernel 3, the second, third, fourth and fifth layers adopt the stacked form of: FRDown and LightSSM Block, where the repetition number of LightSSM Block in backbone is $\{3,\ 6,\ 6,\ 3\}$, FRDown performs multi-level and multi-scale information fusion, preserving a significant portion of contextual information. The detailed designs of MambaInst is shown in Figure 2. Notably, unlike traditional methods, MambaInst is trained from scratch, avoiding any pre-training programs.

B. *LightSSM Block*

Our proposed LightSSM Block is a feature extraction unit designed for instance segmentation, as shown in Figure 3. The LightSSM Block structure starts from the LayerNorm layer of the input data, after which it passes through a linear layer and is fed into DWConv to operate on each input channel, focusing on the extraction of the spatial features while reducing the need for additional parameter calculations. The introduction of additional parameter calculations is minimized. The output of Selective Scan 2D is then linearly transformed and gated, and the linearly transformed output is adjusted to depth features using normalized features, and finally, the features are passed to the final linear layer, where LightSSM Block uses residual connection to implement **C**hannel attention **F**usion (**CF**) to make the depth features add up with the channel localized features. DWConv is used to introduce locality in FFN to provide good location information.The design of LightSSM Block focuses on reducing the parameter introduction and computational resource consumption while enhancing the ability to extract rich spatial features from the image, making the whole MambaInst flexible and lightweight in construction.

C. *Fusion feature Residual DownSampling*

Conventional downsampling methods usually result in the loss of important spatial information [35]–[38], especially boundary and texture details, but this is crucial for instance segmentation that require pixel-level accuracy. We implement multi-scale feature extraction through gating mechanisms and residual connections, and design a downsampling module that preserves as much contextual information as possible: FRDown, as shown in Figure 2 (left), which utilizes a segmented, lightweight feature processing unit to extract multi-scale feature maps into the Mamba modules.

$$x_i^{k-1} = W_p\left(\mathrm{Conv}_{3\times 3}(x^{k-2})\right) \tag{1}$$

$$x^k = \Phi\left(\mathrm{BN}\left(\mathrm{Concat}(x_1^{k-1}, x_2^{k-1})\right)\right) \tag{2}$$

In Equation (1), the input feature x^{k-2} is passed through a 1×1 convolutional layer using W_p Depthwise Convolution (DWConv) on different paths, to efficiently extract features without introducing too many parameters, and then the resulting two-channel features x_1^{k-1} and x_2^{k-1} are spliced together and nonlinearly transformed by batch normalization and Φ (SiLU activation function) to enhance the global context representation. x^{k+1} denotes the input features of the feature processing unit, and the global semantic information is preserved during downsampling by adaptive average pooling. Then the number of channels is adjusted by the 1×1 convolutional layer. x^{k+1} is defined as:

$$x^{k+1} = W_d\left(\mathrm{Avgpool}\left(W_d\left(x^k\right)\right)\right) \tag{3}$$

The feature processing unit uses Gating$(\cdot)$ for adaptive scaling. x_2^{k+1} applies W_p to sum the result with the original input to form a residual connection and multiplies it with the activated features via the φ (GELU activation function), x_1^{k+1} as coefficients, $\odot$ represents the element-wise multiplication of matrices. Then the global context-guided $\mathrm{Conv}_{1\times1}(x^k)$ is multiplied with the feature processing unit to obtain the sampled output x^{k+2}, realizing the global enhancement of the feature.

$$x^{k+2} = W_d \mathrm{Gating}(x^{k+1}) \odot x^{k+1} \tag{4}$$

$$\mathrm{Gating}(X) = \varphi(W_d W_p(X) + X) \odot W_d(X) \tag{5}$$

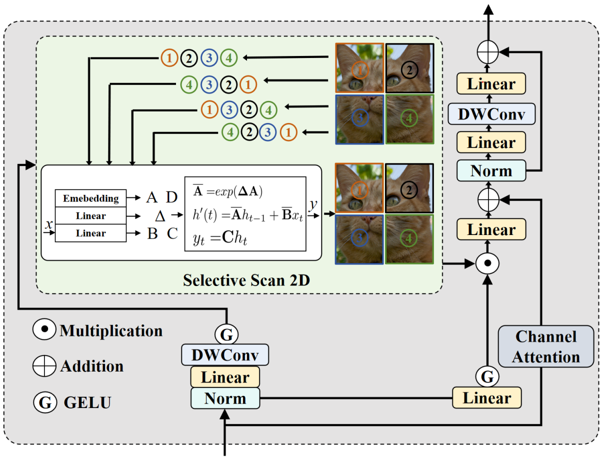

Fig. 3. Illustration of LightSSM Block architecture. LightSSM Block scans the input image in four directions (horizontally up and down, vertically left and right) and then sums the scanned features to capture complex spatial relationships and provide a comprehensive understanding of the context to enable global modeling.

FRDown represents a downsampling methodology that is engineered to preserve and augment critical information, concurrently reducing the associated complexity. This approach facilitates the integration of local details with global context within downsampled features, thereby elevating the performance of instance segmentation.

IV. Experiment

A. *Implementation Details*

All experiments are implemented based on Python 3.8 and CUDA 12.1, and run on 8 × NVIDIA H800 GPUs, trained and validated on the MSCOCO with an input image size of 640×640. All models use official pre-trained models tested for latency on NVIDIA 4090 GPU using the half-precision floating-point format (FP16), with the TensorRT version 8.4.3 and cuDNN version 8.2.0.

B. *Experimental Results*

We compared MambaInst with previous state-of-the-art methods on the val set of MSCOCO. As shown in Table I, **bold** indicates the best, **–** indicates that the paper does not provide data or does not publicly provide the verification weights. We first compare MambaInst with the traditional approach using the standard ResNet [41] backbone, and MambaInst-B possesses a definite performance advantage, with a 2.6% increase in Mask mAP and fewer parameters compared to CondInst-R50 [5]. YOLOv8-seg [31] and RTMDe-Inst [11] were augmented with a large amount of data and trained from scratch, and for a fair comparison MambaInst used almost the same training hyper-parameters, and without any additional complex optimizations, MambaInst-T outperformed the original YOLOv8-N-seg, improving the Mask mAP by 9.5%. Compared to the state-of-the-art RTMDet-Ins-T, MambaInst-T also maintains a 1.8% advantage. We also compared the MambaInst on the val set of MSCOCO results for more detailed Mask AP. Our model also has an absolute advantage in terms of latency and FLOPs.

TABLE I
SEGMENTATION RESULTS ON THE COCO SEGMENTATION DATASET.

Method	Mask mAP(%)	Mask mAP50(%)	Params	FLOPs	Latency
YOLOv8-N-seg [31]	29.6	48.2	3.4M	12.6G	1.78 ms
RTMDet-Ins-T [11]	35.4	–	5.6M	23.6G	1.89 ms
RTMDet-Ins-S [11]	**38.7**	–	10.2M	43.0G	2.26 ms
YOLOv8-S-seg [31]	36.0	56.8	11.8M	42.6G	2.11 ms
SparseInst-R50 [40]	34.2	55.3	31.6M	99.1G	11.60ms
MambaInst-T (ours)	37.2	**58.7**	5.2M	16.0G	1.81 ms
Mask-RCNN-R50 [1]	34.7	–	44.4M	240.0G	34.40ms
CondInst-R50 [5]	38.2	59.1	33.9M	240.8G	33.30ms
QueryInst-R50 [7]	39.9	62.2	173.0M	158.0G	– ms
SOLOv2-Lite [3]	37.5	57.7	64.4M	253.5G	29.10ms
Mask-RCNN-R101 [1]	36.1	–	63.4M	308.0G	42.10ms
DiffusionInst [43]	37.5	–	– M	– G	– ms
EASInst [44]	37.9	57.7	– M	– G	– ms
MambaInst-B (ours)	**40.8**	**63.9**	18.1M	55.3 G	2.28 ms

C. Ablation Study

To validate our analysis of LightSSM Block and to evaluate the efficacy of FRDown, we examined each module in MambaInst, focusing on Bbox mAP and Mask mAP, and the experimental results are shown in Table II. Among them, CF and FRDown cannot work well when they appear in the model alone, but they work well in combination with LightSSM Block, which actually comes from the local details and global context information provided by FRDown and the effective retention of key details such as boundaries and textures by CF.

TABLE II
ABLATION STUDY ON MAMBAINST.

Method	LightSSM Block	CF	FRDown	Box mAP(%)	Mask mAP(%)
1				36.2	29.6
2	✓			43.4	35.9
3		✓		36.1	29.9
4			✓	38.4	31.5
5	✓		✓	45.1	36.5
6	✓	✓	✓	**45.7**	**37.2**

We explored the number of repetitions of four different variants of LightSSM Block in the backbone. The experimental results are shown in Table III. Blocks indicate the number of repetitions of LightSSM Block in the backbone. ✓indicates that the LightSSM Block is used in the neck, ✗indicates that the LightSSM Block is not used in the neck. In output feature map size, P2=20 × 20, P3=40 × 40, P4=80 × 80, P5=160 × 160. The low accuracy of $\{3,\ 3,\ 3,\ 3\}$ is caused by the insufficient number of repetitions of LightSSM Block in the backbone. The $\{3,\ 6,\ 9,\ 3\}$ and $\{3,\ 9,\ 9,\ 3\}$ bring additional computational overhead but do not yield the corresponding level of accuracy improvement, which is in fact a kind of redundancy due to the duplication of LightSSM Block.

In the Neck, although we can realize a lighter model by removing the LightSSM Block, it will lead to an inevitable decrease in the accuracy, and the experimental results prove that the LightSSM Block in the Neck part can also show a certain richness of gradient flow and feature fusion. The output feature map selection of $\{P2, P3, P4, P5\}$ variant can substantially improve the accuracy, but it inevitably results in the increase of GFLOPs. Above all, the Blocks are $\{3,\ 6,\ 6,\ 3\}$ and the Feature Maps are $\{P3, P4, P5\}$, using LightSSM Block in the neck. This scheme can better achieve the balance between accuracy and complexity, and better adapt to efficient instance segmentation.

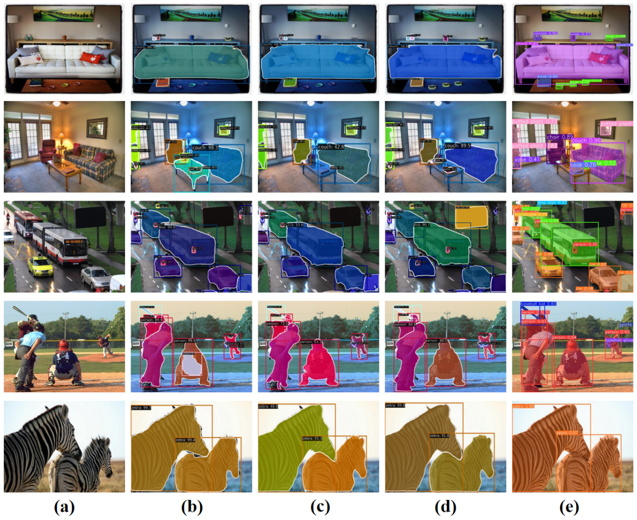

Fig. 4. Visualization results on the MSCOCO dataset. (a) Original images. (b) Mask-RCNN-R50 [1]. (c) CondInst-R50 [5]. (d) RTMDet-Ins-T [11]. (e) MambaInst-T (Ours). Zoom in for better visualization.

TABLE III
THE CONFIGURATIONS OF MAMBAINST VARIANTS.

Blocks	w/o Mamba(Neck)	Feature Map	Box mAP(%)	Mask mAP(%)	FLOPs
$\{3,\ 6,\ 9,\ 3\}$	✓	$\{P3, P4, P5\}$	46.0	37.4	20.8G
$\{3,\ 9,\ 9,\ 3\}$	✓	$\{P3, P4, P5\}$	46.2	37.9	12.8G
$\{3,\ 6,\ 6,\ 3\}$	✗	$\{P3, P4, P5\}$	44.1	36.3	17.6G
$\{3,\ 6,\ 6,\ 3\}$	✓	$\{P2, P3, P4, P5\}$	46.3	38.2	39.6G
$\{3,\ 6,\ 6,\ 3\}$	✓	$\{P3, P4, P5\}$	45.7	37.2	16.0G

The visualized instance segmentation comparison result graph is displayed in Figure 4. Because of the superior global-awareness domain of the MambaInst architecture and the effective preservation of key details such as boundaries and textures, which allows the model to obtain more comprehensive and fine-grained boundaries and dimensions with higher confidence, which further proves the significant advantage of MambaInst in the domain of instance segmentation.

V. CONCLUSION

In this paper, we design the LightSSM Block with linear computational complexity modeling long-range spatial dependencies for extracting deep semantic features. we propose a new downsampling strategy, FRDown, that fuses global contextual information while enhancing local feature representation. Extensive experiments on the COCO-seg benchmark verify the extreme competitiveness of our proposed method, MambaInst, and the performance improvement is largely attributed to the design of the MambaInst architecture, and we hope that our method will bring some new ideas to the field of efficient instance segmentation.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (62376252); Key Project of Natural Science Foundation of Zhejiang Province (LZ22F030003); Zhejiang Province Leading Geese Plan (2024C02G1123882).

References

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN.” Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969, 2017.

[2] B. Dong, F. Zeng, T. Wang, and et al., “SOLQ: Segmenting Objects by Learning Queries.” Advances in Neural Information Processing Systems, vol. 34, pp. 21898–21909, 2021.

[3] X. Wang, R. Zhang, T. Kong, and et al., “SOLOv2: Dynamic and Fast Instance Segmentation.” Advances in Neural Information Processing Systems, vol. 33, pp. 17721–17732, 2020.

[4] H. Chen, K. Sun, Z. Tian, and et al., “BlendMask: Top-down Meets Bottom-up for Instance Segmentation.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8573–8581, 2020.

[5] Z. Tian, C. Shen, and H. Chen, “Conditional Convolutions for Instance Segmentation.” Computer Vision–ECCV 2020, Springer International Publishing, pp. 282–298, 2020.

[6] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image segmentation using deep learning: A survey.” IEEE transactions on pattern analysis and machine intelligence, vol. 44, no. 7, pp. 3523–3542, 2021.

[7] Y. Fang, S. Yang, X. Wang, and et al., “Instances As Queries.” Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6910–6919, 2021.

[8] G. Zhang, X. Lu, J. Tan, and et al., “RefineMask: Towards High-quality Instance Segmentation with Fine-grained Features.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6861–6869, 2021.

[9] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “Yolact: Real-time instance segmentation.” Proceedings of the IEEE/CVF international conference on computer vision, pp. 9157–9166, 2019.

[10] M. Huang, G. Xu, J. Li, and et al., “A Method for Segmenting Disease Lesions of Maize Leaves in Real Time Using Attention YOLACT++.” Agriculture, vol. 11, no. 12, p. 1216, 2021.

[11] C. Lyu, W. Zhang, H. Huang, and et al., “RTMDet: An Empirical Study of Designing Real-time Object Detectors.” arXiv preprint arXiv:2212.07784, 2022.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, and et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.” arXiv preprint arXiv:2010.11929, 2020.

[13] Z. Liu, Y. Lin, Y. Cao, and et al., “Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows.” Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022, 2021.

[14] Z. Chen, Y. Duan, W. Wang, and et al., “Vision Transformer Adapter for Dense Predictions.” arXiv preprint arXiv:2205.08534, 2022.

[15] F. Li, H. Zhang, H. Xu, and et al., “Mask DINO: Towards a Unified Transformer-Based Framework for Object Detection and Segmentation.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3041–3050, 2023.

[16] Z. Liang and Y. Yuan, “Mask Frozen-Detr: High Quality Instance Segmentation with One GPU.” arXiv preprint arXiv:2308.03747, 2023.

[17] A. Vaswani, N. Shazeer, N. Parmar, and et al., “Attention is All You Need.” Advances in Neural Information Processing Systems, vol. 30, 2017.

[18] A. Gu, K. Goel, and C. Ré, “Efficiently Modeling Long Sequences with Structured State Spaces.” arXiv Preprint arXiv:2111.00396, 2021.

[19] J. T. H. Smith, A. Warrington, and S. W. Linderman, “Simplified State Space Layers for Sequence Modeling.” arXiv preprint arXiv:2208.04933, 2022.

[20] R. Bhirangi, C. Wang, V. Pattabiraman, and et al., “Hierarchical State Space Models for Continuous Sequence-to-Sequence Modeling.” arXiv preprint arXiv:2402.10211, 2024.

[21] A. Gu and T. Dao, “Mamba: Linear-time Sequence Modeling with Selective State Spaces.” arXiv preprint arXiv:2312.00752, 2023.

[22] L. Zhu, B. Liao, Q. Zhang, and et al., “Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model.” arXiv preprint arXiv:2401.09417, 2024.

[23] Y. Liu, Y. Tian, Y. Zhao, and et al., “VMamba: Visual State Space Model.” arXiv preprint arXiv:2401.10166, 2024.

[24] T. Huang, X. Pei, S. You, and et al., “LocalMamba: Visual State Space Model with Windowed Selective Scan.” arXiv preprint arXiv:2403.09338, 2024.

[25] W. Yu and X. Wang, “MambaOut: Do We Really Need Mamba for Vision?” arXiv preprint arXiv:2405.07992, 2024.

[26] J. Ruan and S. Xiang, “Vm-unet: Vision Mamba unet for medical image segmentation.” arXiv preprint arXiv:2402.02491, 2024.

[27] S. Zhao, H. Chen, X. Zhang, and et al., “RS-Mamba for Large Remote Sensing Image Dense Prediction.” arXiv preprint arXiv:2404.02668, 2024.

[28] J. Ma, F. Li, and B. Wang, “U-Mamba: Enhancing Long-range Dependency for Biomedical Image Segmentation.” arXiv preprint arXiv:2401.04722, 2024.

[29] Z. Xing, T. Ye, Y. Yang, G. Liu, and L. Zhu, “SegMamba: Long-range sequential modeling Mamba for 3d medical image segmentation.” arXiv preprint arXiv:2401.13560, 2024.

[30] Z. Wang, C. Li, H. Xu, and et al., “Mamba YOLO: SSMs-Based YOLO for Object Detection.” arXiv preprint arXiv:2406.05835, 2024.

[31] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLO.” Version 8.0.0, 2023. https://github.com/ultralytics/ultralytics.

[32] N. Carion, F. Massa, G. Synnaeve, and et al., “End-to-End Object Detection with Transformers.” European Conference on Computer Vision, Springer International Publishing, pp. 213–229, 2020.

[33] H. Zhang, F. Li, S. Liu, and et al., “DINO: DETR with Improved Denoising Anchor Boxes for End-to-End Object Detection.” arXiv preprint arXiv:2203.03605, 2022.

[34] R. Girshick, “Fast R-CNN.” Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448, 2015.

[35] G. Lin, A. Milan, C. Shen, and I. Reid, “Refinenet: Multi-path refinement networks for high-resolution semantic segmentation.” Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1925–1934, 2017.

[36] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, “Unet++: A nested u-net architecture for medical image segmentation.” Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4, pp. 3–11, 2018.

[37] G. Xu, W. Liao, X. Zhang, C. Li, X. He, and X. Wu, “Haar wavelet downsampling: A simple but effective downsampling module for semantic segmentation.” Pattern Recognition, vol. 143, p. 109819, 2023.

[38] T. Wu, S. Tang, R. Zhang, J. Cao, and Y. Zhang, “Cgnet: A light-weight context guided network for semantic segmentation.” IEEE Transactions on Image Processing, vol. 30, pp. 1169–1179, 2020.

[39] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context.” In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pp. 740–755, 2014.

[40] T. Cheng, X. Wang, S. Chen, W. Zhang, Q. Zhang, C. Huang, Z. Zhang, and W. Liu, “Sparse instance activation for real-time instance segmentation.” In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4433–4442, 2022.

[41] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.” In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.

[42] W. Luo, Y. Li, R. Urtasun, and R. Zemel, “Understanding the effective receptive field in deep convolutional neural networks.” In Advances in Neural Information Processing Systems, vol. 29, 2016.

[43] G. Zhangxuan, C. Haoxing, and X. Zhuoer. “Diffusioninst: Diffusion model for instance segmentation.” ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024.

[44] X. Renqiu, Z. Dongyuan, D. Yixin, Z. Juanping, L. Wenlong, H. Tao, Y. Junchi. “Efficient Architecture Search for Real-Time Instance Segmentation.” ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024.