Mamba-Adaptor: State Space Model Adaptor for Visual Recognition
–Supplementary Material–

Fei Xie[1] Jiahao Nie[2*] Yujin Tan[1] Wenkang Zhang[3] Hongshen Zhao[3]
[1] Shanghai Jiao Tong University
[2] Hangzhou Dianzi University
[3] Southeast University
jaffe0319@gmail.com, jhnie@hdu.edu.cn

1. Overview of Supplementary Materials

In the appendix, we first provide additional architectural details in Section 2. Next, we present further experimental settings in Section 3. We also include more information about the proposed Mamba-Adaptor in Section 4. Finally, we present additional experiments on semantic segmentation in Section 5.

2. Detailed Architecture

In Figure 1, we present a comprehensive illustration of the model architectures related to our proposed Mamba-Adaptor. This figure highlights not just the overall architecture but also the individual components constituting the Mamba-Adaptor Block. The subfigure provides a detailed view of the specific features and functions of the Mamba-Adaptor Block, promoting a clearer understanding of its design and role within the overall model architecture.

3. Experimental Settings

3.1. Training Settings on ImageNet1K Benchmark

The training configurations for ImageNet1K [11] utilized in this study align closely with the approaches detailed in [7]. We establish a consistent input image resolution of 224 x 224 pixels for all model variants, which serves as our default for the initial training phase. For different resolutions, such as 384 x 384 pixels, we implement a fine-tuning method. This involves taking models previously trained on the 224 x 224 resolution and adapting them for the larger input size, rather than retraining from scratch. This approach, which is supported by findings from [7], is particularly advantageous as it reduces the overall GPU resources needed during training. By leveraging earlier training, we not only attain efficient performance but also minimize computing expenses.

*Corresponding author.

3.2. Training Settings on COCO Benchmark

Our emphasis is on the popular object detection framework, Cascade Mask R-CNN [1, 5], which has garnered significant attention in scholarly discussions, especially regarding Mask R-CNN and the work by Cai et al. (2018). Implementation of this framework is achieved through MMDetection [2], a readily available open-source toolbox for object detection.

To optimize our model, we use the AdamW [9] optimizer, which is well-known for its effectiveness in training deep learning architectures. We start with an initial learning rate of 0.0001, a common starting point that allows for a gradual optimization process. Our training process utilizes a batch size of 16, which strikes a balance between memory usage and training efficiency.

4. Detail of Mamba-Adaptor

4.1. Analysis of State Space Models.

To simplify calculations, the repeated application in SSM can be effectively performed simultaneously using a global convolution method.

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{x} \circledast \overline{\boldsymbol{K}} \\ \text{with} \quad \overline{\boldsymbol{K}} &= (\boldsymbol{C}\overline{\boldsymbol{B}}, \boldsymbol{C}\overline{\boldsymbol{A}\boldsymbol{B}}, ..., \boldsymbol{C}\overline{\boldsymbol{A}}^{L-1}\overline{\boldsymbol{B}}), \end{aligned} \tag{1}$$

where $\circledast$ denotes convolution operation, and $\overline{\boldsymbol{K}} \in \mathbb{R}^L$ is the SSM kernel.

Exponential Decay. As shown in Equation 1, the output sequence $\{y_1, \ldots, y_N\}$ can also be computed as the convolution results of the input with the convolutional kernel [4]:

$$f = [\boldsymbol{C}\boldsymbol{B}, \boldsymbol{C}\boldsymbol{A}\boldsymbol{B}, \boldsymbol{C}\boldsymbol{A}^2\boldsymbol{B}, \ldots, \boldsymbol{C}\boldsymbol{A}^{N-1}\boldsymbol{B}].$$

That is, from an initial condition x_0, we have $y_i = \boldsymbol{C}\boldsymbol{A}^i\boldsymbol{B}x_0 + (f * u)_i + \boldsymbol{D}u_i$, where $(f * u)$ denotes a linear convolution between f and u. By setting the initial condition x_0 to zero, y becomes a linear convolution of u, enhanced by a residual connection. In broader terms, every

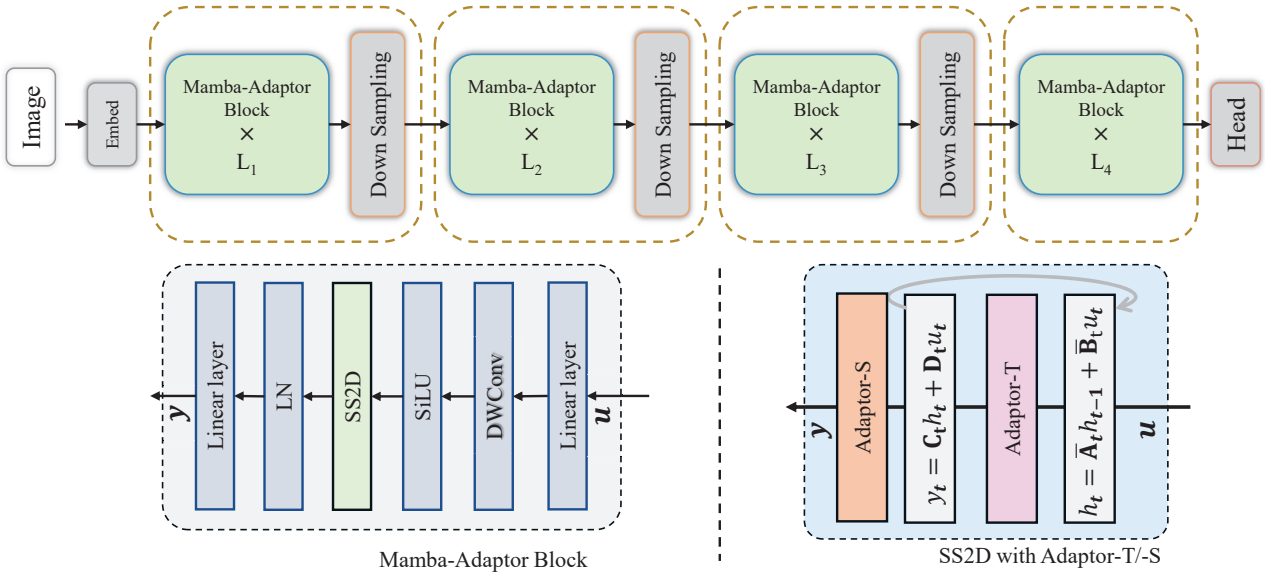

Figure 1. Detailed model architectures of the proposed Mamba-Adaptor. The detailed architecture of the Mamba-Adaptor Block is also presented.

linear time-invariant system (including SSMs as a specific example) can be expressed in convolution form.

4.2. Improvements from Adaptor.

Calculating convolutional weights requires cumulatively multiplying an exponential matrix, represented as (mA). This approach introduces an exponential decay effect on the input mx. As the convolutional operation advances, the impact of preceding input values diminishes swiftly due to this decay. To mitigate the challenges posed by this exponential decay, we have developed an innovative solution called Adaptor-S. Adaptor-S effectively addresses the decay issue by focusing on aggregating the input data $\boldsymbol{x}$ from various locations within the input space rather than solely relying on the decayed values. This approach allows for retaining more meaningful information from the input, enhancing the overall effectiveness of the convolutional operation. By directly gathering data from multiple locations, Adaptor-S ensures that the model can utilize a broader range of information, leading to more accurate and robust outputs.

Sequential Format. Given a 1D input sequence $u \in \mathbb{R}^N$ of length N, we denote the 1D output sequence $y \in \mathbb{R}^N$ of an SSM parameterized by matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$ as

$$y = \mathrm{SSM}_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}}(u).$$

To simplify notation, we omit the reference to $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$ and write $y = \mathrm{SSM}(x)$ if they are clear from context. The input signal x is transformed into a 1D sequence, a method that frequently neglects the spatial structural details in image data. This conversion may result in the loss of vital contextual information, which is necessary for effective image analysis. To mitigate this issue, we employ convolutional filters, which take advantage of the 2D visual inductive bias. These filters operate by conducting spatial aggregation within a specified local window. By concentrating on small sections of the image, convolutional filters can identify local patterns and features, maintaining the spatial relationships that are crucial for interpreting visual data. This method enables us to keep more of the spatial details while processing the information efficiently.

5. Additional Experiments

We conduct additional experiments in semantic segmentation on the ADE20K [15] dataset. ADE20K is a renowned semantic segmentation dataset comprising 150 diverse semantic categories. It includes 25,000 images in total: 20,000 for training, 2,000 designated for validation, and 3,000 reserved for testing. For our implementation, we use UperNet [13] within the mmsegmentation framework [3]. To align with the baseline model, we use a training setup similar to previous works [6]. During the inference phase, a multi-scale evaluation is conducted using resolutions that vary from 0.5 to 1.75 times the resolution used for training. The test scores encompass both training and validation images, which is standard practice [14].

In Table 1, we present the performance results of our Mamba-Adaptor variants, specifically b1 and b2. These variants have demonstrated outstanding performance, achieving state-of-the-art results in comparison to existing models. They enhance the capabilities of the

Backbone	Params(M)	FLOPs(G)	mIoU-SS(%)	mIoU-MS(%)
EffVMamba-S [10]	29M	505G	41.5	42.1
MSVMamba-M [12]	42M	875G	45.1	45.4
Mamba-Adaptor-b1	38M	708G	45.4	46.0
Swin-T [7]	60M	945G	44.4	45.8
ConvNeXt-T [8]	60M	939G	46.0	46.7
VMamba-T [6]	55M	964G	47.3	48.3
EffVMamba-B [10]	65M	930G	46.5	47.3
MSVMamba-T [12]	65M	942G	47.6	48.5
Mamba-Adaptor-b2	58M	971G	47.8	48.6

Table 1. Additional experiments on the semantic segmentation in ADE20K [15] benchmark. SS and MS denote single-scale and multi-scale inference settings.

baseline model, VMamba [6], with minimal additional costs in model complexity and resources. This enhancement demonstrates the efficiency and effectiveness of the Mamba-Adaptor, making it an attractive option for researchers and practitioners in the field.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 1

[2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, et al. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint*, 2019. 1

[3] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 2

[4] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. In *NeurIPS*, 2021. 1

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1

[6] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *NeurIPS*, 2024. 2, 3

[7] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *CVPR*, 2021. 1, 3

[8] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, et al. A convnet for the 2020s. In *CVPR*, 2022. 3

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint*, 2017. 1

[10] Xiaohuan Pei, Tao Huang, and Chang Xu. Efficientvmamba: Atrous selective scan for light weight visual mamba. *NeurIPS*, 2024. 3

[11] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. ImageNet Large scale visual recognition challenge. *IJCV*, 2015. 1

[12] Yuheng Shi, Minjing Dong, and Chang Xu. Multi-scale vmamba: Hierarchy in hierarchy visual state space model. *arXiv preprint arXiv:2405.14174*, 2024. 3

[13] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 2

[14] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In *ECCV*, 2020. 2

[15] Bolei Zhou, Hang Zhao, Xavier Puig, et al. Scene parsing through ade20k dataset. In *CVPR*, 2017. 2, 3