Visual-SSM: Adapting State-Space Models for Efficient Vision Tasks

SereiWathna Ros, HyungWon Kim*
Chungbuk National Univ.

{sereiwathnaros, hwkim}@chungbuk.ac.kr

Abstract

In natural language processing (NLP) problems, state-space models (SSMs) have emerged as strong competitors to Transformer-based models due to recent developments in deep learning architectures. It has been demonstrated that small-to-medium sized language modeling with SSMs such as Mamba is on par with or superior to Transformer-based models. Developing efficient and versatile vision frameworks based solely on SSMs is an intriguing approach. The hypothesis behind this study is that SSMs' sequential modeling capabilities can be successfully applied to capture spatial dependencies in image data, thereby expanding the application of SSMs to visual tasks. We introduce Visual-SSM, a novel SSM-based architecture intended for computer vision applications.

Ⅰ. Introduction

Recent research advancement, Transformers [1] have become the de facto standard architecture in many areas such as computer vision, natural language processing, robotics, and audio processing. The advantages of Transformers over other architectures are primarily due to the attention mechanisms [2] and their flexibility is well-suited for multimodal learning tasks which require integrating information from diverse datasets. However, one of the main concerns with Transformer is the quadratic complexity of attention mechanism with respect to sequence length poses a significant computational challenge. To alleviate this limitation, Mamba [3] proposed a new state space model (SSM) which achieves linear time complexity and is on par or outperform Transformers [3] in different natural language processing tasks. The key innovation of Mamba is an alternative approach of selection mechanisms which enables efficient input-dependent processing of long sequences with optimized hardware-aware configurations.

In this paper, we propose a Visual-SSM which is solely based on SSM to be more suitable for vision tasks. The model is evaluated on CIFAR-10 dataset.

Ⅱ. Methodology

The Visual-SSM is a state space model designed to provide insight into capability of the state space model for vision tasks. This approach excludes the other architectures like convolutions, which have a strong inductive bias toward local patterns of images in early layers or transformers to capture the long-range dependencies within image data. This approach allows us to focus on an examination of the state space model's intrinsic capabilities in vision tasks, which is unaugmented by other specialized architectures.

Mamba is an extension of structured state space model (S4) which is inspired by the continuous system, which transform a 1D continuous input $x(t) \in \mathbb{R}$ to $y(t) \in \mathbb{R}$ via a learnable hidden state $h(t) \in \mathbb{R}^M$ with parameters $\boldsymbol{A} \in \mathbb{R}^{M\times M}$and $\boldsymbol{C} \in R^{1\times M}$ according to:

$$h'(t) = \boldsymbol{A}h(t) + \boldsymbol{B}x(t), \quad (1)$$

$$y(t) = \boldsymbol{C}h(t) \quad (2)$$

Discretization The S4 and Mamba are the discrete versions of the continuous system, hence, the continuous parameters $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are required to be discretized for better computational efficiency [4]. Consider a timescale parameter $\boldsymbol{\Delta}$ to transform our mentioned continuous parameters to discrete parameters $\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}$, and $\overline{\boldsymbol{C}}$. In practice, zero-order hold (ZOH) is commonly used for such a transformation and is defined as follows:

$$\overline{\boldsymbol{A}} = \exp(\boldsymbol{\Delta A}), \quad (3)$$

$$\overline{\boldsymbol{B}} = (\boldsymbol{\Delta A})^{-1}(\exp \boldsymbol{\Delta A}) - \boldsymbol{I}) \cdot \boldsymbol{\Delta B}. \quad (4)$$

After the discretization, Eq. (1) and (2) can be expressed as:

$$h'(t) = \overline{\boldsymbol{A}}h(t-1) + \overline{\boldsymbol{B}}x(t), \quad (5)$$

$$y(t) = \overline{\boldsymbol{C}}h(t) \quad (6)$$

Finally, the models compute through a convolution as follows:

$$\overline{\boldsymbol{K}} = (\boldsymbol{C}\overline{\boldsymbol{B}}, \boldsymbol{C}\overline{\boldsymbol{AB}}, \ldots, \boldsymbol{C}\overline{\boldsymbol{A}}^{M-1}\overline{\boldsymbol{B}}), \quad (7)$$

$$\boldsymbol{y} = \boldsymbol{x} * \overline{\boldsymbol{K}}, \quad (8)$$

where M is the length of the input sequence **x**, and $\overline{\boldsymbol{K}} \in \mathbb{R}^M$ is a structured convolutional kernel.

Visual-SSM Our architecture consists of 2.6 million parameters including layers such as linear layer, layer normalization [5], dropout [6], global average pooling [7], and Mamba Block [3]. Mamba Block contains convolution 1D, Residual Connection [8], SSM [2], and SiLU [9]. As illustrate in Figure 1 is our Mamba block, in which σ denotes our SiLU activation. In addition, our overall architecture can be seen in Figure 2, the model takes the input of image and embed it into 1D sequence as Mamba is designed to take 1D sequence. We first transform the 2D image $\boldsymbol{Img} \in \mathbb{R}^{H\times W\times C}$ into the flattened 2D patches of size $H'W' \times (P^2 \cdot C)$ in which (H, W) is the size of input image, C is the number of channels, (H', W') is the size of image after dividing by the image patches. P is the size of image patches which in our case is equal to 4.

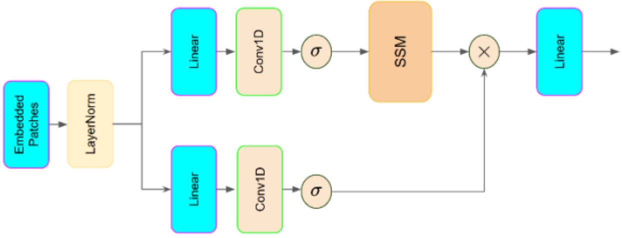

Figure 1. Architecture of **MambaBlock**. A symmetric path without SSM to enhance the modeling of global context.

Ⅲ. Results and Discussion

The experiments were conducted primarily on the CIFAR-10 dataset, which consists of 60000 images across 10 categories. During training, data augmentation techniques including

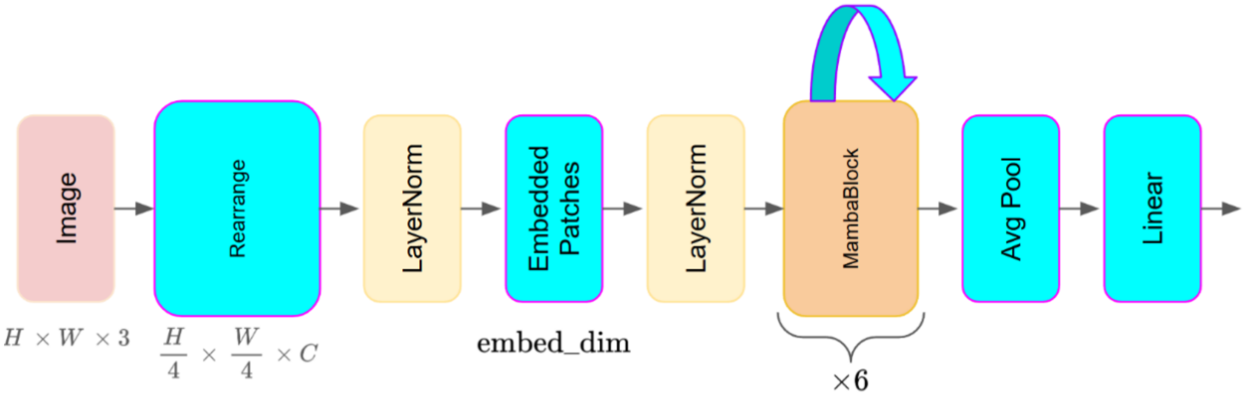

Figure 2. The **Visual-SSM** architecture. The first phase we rearrange our image with patch size of 4 and embedding layer each followed by a layer normalization respectively, then we apply our Mamba Block 6 times.

normalizing input according to their mean and standard deviation, and RandAugment [10] has been used to get our Visual-SSM to be more robust. Additionally, we compare our Visual-SSM with a plain Vision Transformer [11] containing 2.7 million parameters, with the outcome demonstrated in *Table 1*.

	Visual-SSM	**Vision Transformer**
Class	Top1 - Accuracy	
Plane	90.00%	87.20%
Car	94.50%	89.30%
Bird	82.10%	80.60%
Cat	77.90%	71.20%
Deer	88.10%	81.20%
Dog	75.80%	78.00%
Frog	90.00%	89.80%
Horse	88.70%	87.10%
Ship	93.70%	90.20%
Truck	91.30%	94.30%
Overall	87.21%	84.90%

Table 1. Demonstrate the result from the experiment with all the classes and overall top-1 accuracy.

IV. **Conclusion**

In this study, we introduce Visual-SSM, which is mainly based on SSM for vision tasks. Visual-SSM achieves 87.21% on CIFAR-10 dataset. We hope that this work contributes meaningful insight to the field of computer vision and serves as the foundation for future development involving SSM.

ACKNOWLEDGMENT

This work was supported by Regional Leading Research Center (RLRC) of the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No.2022R1A5A8026986) and supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-01304, Development of Self-Learnable Mobile Recursive Neural Network Processor Technology). It was also supported by the MSIT (Ministry of Science and ICT), Korea, under the Grand Information Communication Technology Research Center support program (IITP-2024-2020-0-01462) supervised by the IITP (Institute or Information & communications Technology Planning & Evaluation).

REFERENCES

[1] Ashish V. et al., "Attention is All You Need," 2017, (https://arxiv.org/abs/1706.03762).

[2] Dzmitry B., Kyunghyun C., and Yoshua B., "Neural Machine Translation by Jointly Learning to Align and Translate," 2014, (https://arxiv.org/abs/1409.0473).

[3] Albert G. and Tri D., "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," 2023, (https://arxiv.org/abs/2312.00752)

[4] Albert G. et al., "Combining Recurrent, Convolutional, and Continuous-time Models with Linear State-Space Layers," 2021, (https://arxiv.org/abs/2110.13985)

[5] Jimmy L.B, Jamie R.K. and Geoffrey E.H., "Layer Normalization," 2016, (https://arxiv.org/abs/1607.06450v1)

[6] Nitish S. et al., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," 2024, (https://jmlr.org/papers/v15/srivastava14a.html)

[7] Min L., Qiang C., and Shuicheng Y., "Network in Network," 2014, (https://arxiv.org/abs/1312.4400v3)

[8] Kaiming H. et al., "Deep Residual Learning for Image Recognition," 2015, (https://arxiv.org/abs/1512.03385v1)

[9] Stefan E., Eiji U., and Kenji D., "Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning," 2017, (https://arxiv.org/abs/1702.03118)

[10] Ekin D. et al., "RandAugment: Practical automated data augmentation with a reduced search space," 2019, (https://arxiv.org/abs/1909.13719)

[11] Alexey D. et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," 2020, (https://arxiv.org/abs/2010.11929)