Mamba vs SOTA for Vision Tasks

Clay Crews†
Department of Computer Science
University of South Carolina
Columbia, SC, United States
jccrews@email.sc.edu

Lexington Whalen†
Department of Computer Science
University of South Carolina
Columbia, SC, United States
LAWHALEN@email.sc.edu

***Abstract*—Abstract This paper explores the performance of the Mamba architecture, a variant of Structured State Space Sequence Models (SSMs), in comparison to state-of-the-art models such as Transformers, U-Net, and ResNet for vision tasks. We focus on image segmentation and classification, with the goal of maintaining high accuracy while keeping parameter counts low. Developing compact models with fewer parameters is crucial for the future of AI, not only to reduce energy consumption but also to enable deployment on resource-constrained edge devices. By leveraging the selective state spaces in Mamba blocks, we aim to achieve efficient and effective segmentation and classification performance while maintaining the linear scalability of SSMs. We implement Mamba-based architectures and compare their results to popular models like U-Net, ResNet, and Transformer-based approaches on standard vision benchmarks. In addition to accuracy, we place a strong emphasis on model size, targeting compact architectures suitable for resource-constrained environments. Through careful design choices and optimizations, we strive to develop Mamba-based models that achieve competitive accuracy with significantly fewer parameters compared to existing state-of-the-art models. This work highlights the potential of Mamba and SSMs for efficient vision tasks, contributing to the development of compact yet accurate models in image segmentation and classification. We open-source our code to facilitate further research and exploration of these architectures: https://github.com/lxaw/mamba-vs-else-vision**

***Index Terms*—Mamba, TinyML, Image Segmentation, Image Classification**

I. Introduction

The rapid advancement of deep learning has led to remarkable achievements in various domains, including computer vision, natural language processing, and speech recognition. However, this progress has been accompanied by a significant increase in the size and complexity of neural network models. State-of-the-art models often have hundreds of millions or even billions of parameters, making them computationally expensive and challenging to deploy on resource-constrained devices [1], [18]. This trend poses concerns regarding energy consumption and the feasibility of deploying these models on edge devices, which have limited memory and processing power.

The growth of edge devices and the emerging field of TinyML have highlighted the need for more efficient and compact models [3]. Edge devices, such as smartphones, wearables, and IoT sensors, have become increasingly prevalent in our daily lives. These devices often require real-time processing and decision-making capabilities, but their limited resources make it challenging to run large, complex models. TinyML aims to address this challenge by developing machine learning techniques that can operate efficiently on resource-constrained devices, enabling a wide range of intelligent applications at the edge [4].

Many state-of-the-art models for vision tasks, such as image classification and segmentation, rely on architectures like Transformers [18], U-Net [5], and ResNet [14]. While these models have achieved impressive performance, they often come with a high parameter count. Transformers, in particular, have gained significant attention due to their ability to capture long-range dependencies and achieve superior results in various tasks [7]. However, the self-attention mechanism in Transformers scales quadratically with the input sequence length, making them computationally expensive and difficult to apply to long sequences or high-resolution images [20].

To address the challenges of large parameter models and enable efficient deployment on edge devices, there is a growing interest in developing more compact and computationally efficient architectures. One promising approach is the Mamba architecture, which is based on Structured State Space Sequence Models (SSMs) [23]. Mamba combines the modeling power of Transformers with the linear scalability of SSMs, allowing it to efficiently process long sequences while maintaining high performance [25]. By leveraging selective state spaces, Mamba can achieve competitive results with significantly fewer parameters compared to Transformers and other large models.

In this paper, we explore the application of Mamba for vision tasks, specifically image segmentation and classification. We aim to demonstrate that Mamba-based models can achieve high accuracy while keeping parameter counts low, making them suitable for deployment on edge devices and contributing to the development of more sustainable and efficient AI solutions. We compare the performance of Mamba against state-of-the-art models and discuss the potential of this approach for enabling intelligent applications on resource-constrained devices.

II. Current Vision Task SOTA

Over the past decade, deep learning has revolutionized the field of computer vision, with various architectures achieving state-of-the-art (SOTA) performance on tasks such as image

† Equal contributions

classification and segmentation. Convolutional Neural Networks (CNNs) have been at the forefront of this revolution, with architectures like AlexNet [11], VGGNet [12], and Inception [13] pushing the boundaries of classification accuracy on large-scale datasets like ImageNet. One of the most significant advancements in CNNs came with the introduction of ResNet [14], which addressed the problem of vanishing gradients in deep networks by introducing residual connections. ResNets allowed for the training of much deeper networks, leading to improved performance on various vision tasks. The success of ResNet sparked a wave of research into more efficient and effective CNN architectures, such as MobileNet [15], EfficientNet [16], and RegNet [17]. While CNNs have been highly successful in vision tasks, they struggle to capture long-range dependencies and global context, which are crucial for understanding complex scenes and objects. To address this limitation, the Vision Transformer (ViT) [18] was introduced, adapting the self-attention mechanism from natural language processing to vision tasks. ViT treats an image as a sequence of patches and applies multi-head self-attention to learn global relationships between these patches. ViT has achieved impressive results on image classification tasks, often outperforming CNNs while requiring less inductive bias. Building upon the success of ViT, several transformer-based architectures have been proposed for other vision tasks, such as image segmentation. SegFormer [19] is a transformer-based model for semantic segmentation that employs a hierarchical structure and a novel attention mechanism called Efficient Self-Attention (ESA). ESA reduces the computational complexity of self-attention by performing attention operations in a local window and aggregating global information through a depth-wise convolution. This allows SegFormer to efficiently process high-resolution images while capturing both local and global context. Despite the impressive performance of these SOTA models, they often come with a high computational cost and large number of parameters. The self-attention mechanism in transformers scales quadratically with the input sequence length, making them challenging to apply to long sequences or high-resolution images [20]. CNNs, while more efficient than transformers, still require a significant number of parameters to achieve SOTA performance, especially for complex tasks like segmentation. The high parameter counts and scaling issues of these models pose challenges for deployment on resource-constrained devices and raise concerns about energy consumption. As the demand for efficient and sustainable AI solutions grows, there is a pressing need for more compact and computationally efficient architectures that can maintain high accuracy while reducing the parameter footprint. This has led to increased interest in techniques like model compression, quantization, and the development of novel architectures that can achieve SOTA performance with fewer parameters [21], [22].

III. State Space Models (SSMs)

State Space Models (SSMs) are a powerful mathematical framework for modeling and processing temporal signals. They map an input signal $u(t)$ to an output signal $y(t)$ via a latent state $x(t)$. The latent state evolves according to a linear differential equation involving the input signal and a set of matrices (A, B, C, D) that define the SSM's behavior [23]. When the SSM matrices are constant over time, the system is called a Linear Time-Invariant (LTI) SSM, and it can be equivalently expressed as a convolution between the input signal and the SSM's impulse response function $K(t)$. This convolutional form makes LTI SSMs particularly efficient to compute, which is crucial for their application in modern deep learning models like S4 [23]. A key concept in the SSM literature is the SSM basis, which is a set of functions defined by the matrix exponential $e^{tA}B$. The output of the SSM is a linear combination of these basis functions, weighted by the coefficients in matrix C. The choice of the SSM basis functions determines the properties and capabilities of the SSM [23]. The HiPPO framework proposed a mathematical technique for deriving SSMs with orthogonal basis functions, such that the latent state $x(t)$ maintains a compressed representation of the entire input history. By carefully designing the SSM matrices, HiPPO derived SSMs whose basis functions could reconstruct the input signal history from the latent state, making them powerful tools for modeling long-range dependencies in sequence data [24]. The Mamba model built upon the foundations laid by HiPPO, extending SSMs with a gating mechanism to enable more general sequence modeling tasks beyond signal reconstruction and memorization [25]. The success of models like S4 and Mamba have demonstrated the potential of SSMs as a fundamental building block for efficient and expressive sequence models.

IV. Mamba

Mamba [25] is a recently proposed architecture that combines the modeling power of Transformers with the linear scaling efficiency of state space models (SSMs). The key idea is to augment SSMs with a selection mechanism that allows the model parameters to vary based on the input, enabling it to perform content-aware reasoning. Standard SSMs map an input sequence $x(t)$ to an output $y(t)$ via a latent state $h(t)$ using a linear time-invariant system:

$$h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t)$$
$$y(t) = \mathbf{C}h(t) + \mathbf{D}x(t)$$

where the parameter matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ are fixed. In the original paper, the $\mathbf{D}$ matrix is considered only as a skip connection, and thus as it does not directly play a role in the differential equation, was ignored. This allows SSMs to be computed efficiently as a convolution. However, the time-invariance means the model cannot change its behavior for specific inputs, making it difficult to solve tasks requiring selective focus, such as selective copying or induction heads [25]. Mamba introduces a selection mechanism where the SSM parameters $\mathbf{B}$, $\mathbf{C}$, Δ (the state step size) vary for each input token x_t. This allows the model to selectively propagate or forget information based on the current token. The selective SSM is computed using a parallel scan operation to maintain

linear complexity. The Mamba block interleaves the selective SSM with linear projections, convolutions, and activations:

1) Linear projection to expand the embedding dimension
2) Convolution to mix information between dimensions
3) Selective SSM via parallel scan to efficiently propagate information
4) Linear projection to get back to the target embedding size

These blocks are stacked to form the Mamba architecture. Compared to Transformers, Mamba can achieve comparable modeling performance while scaling linearly in sequence length rather than quadratically [25]. Overall, the selective SSM leverages the strengths of RNNs, CNNs and Transformers - it maintains an unbounded context window like RNNs, can be parallelized like CNNs, and achieves content-aware reasoning like Transformers, while being more efficient than all three. This makes Mamba a promising foundation model for processing long sequences across various domains.

V. Application: Image Segmentation

VI. Application: Image Classsification

VII. Conclusion

References

[1] T. B. Brown et al., "Language Models are Few-Shot Learners," arXiv:2005.14165 [cs.CL], 2020.
[2] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," arXiv:2010.11929 [cs.CV], 2021.
[3] P. Warden and D. Situnayake, "TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers," O'Reilly Media, Inc., 2019.
[4] C. R. Banbury et al., "MLPerf Tiny Benchmark," arXiv:2106.07597 [cs.LG], 2021.
[5] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," arXiv:1505.04597 [cs.CV], 2015.
[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," arXiv:1512.03385 [cs.CV], 2016.
[7] A. Vaswani et al., "Attention Is All You Need," arXiv:1706.03762 [cs.CL], 2017.
[8] K. Choromanski et al., "Rethinking Attention with Performers," arXiv:2009.14794 [cs.LG], 2020.
[9] A. Gu, K. Goel, and C. Ré, "Efficiently Modeling Long Sequences with Structured State Spaces," arXiv:2111.00396 [cs.LG], 2022.
[10] A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," arXiv:2312.00752 [cs.LG], 2023.
[11] - ImageNet Classification with Deep Convolutional Neural Networks
[12] - Very Deep Convolutional Networks for Large-Scale Image Recognition
[13] - Going Deeper with Convolutions
[14] - Deep Residual Learning for Image Recognition
[15] - MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications
[16] - EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks
[17] - Designing Network Design Spaces
[18] - An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale
[19] - SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers
[20] - Rethinking Attention with Performers
[21] - A Survey of Model Compression and Acceleration for Deep Neural Networks
[22] - Distilling the Knowledge in a Neural Network
[23] A. Gu, K. Goel, and C. Ré, "Efficiently Modeling Long Sequences with Structured State Spaces," arXiv preprint arXiv:2111.00396, 2022.
[24] A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Ré, "HiPPO: Recurrent Memory with Optimal Polynomial Projections," Advances in Neural Information Processing Systems, vol. 33, pp. 1474-1487, 2020.
[25] A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with State Spaces," arXiv preprint arXiv:2302.01327, 2023.