ECE1512 - Digital Image Processing and Applications - Project B

Swapnil Patel
University of Toronto
swap.patel@mail.utoronto.ca
Code: Swapnil949/ECE1512_2024F_ProjectRepo_SwapnilPatel

1. State Space Models (SSMs) [Mamba: Linear-Time Sequence Modeling with Selective State Spaces]

In this section, Mamba, as introduced in the paper ”Mamba: Linear-Time Sequence Modeling with Selective State Spaces” [2], is discussed. Mamba is a novel sequence modeling architecture that leverages Selective State Space Models (SSMs) to address the computational inefficiencies of Transformers while maintaining their ability to perform complex reasoning. The architecture introduces input-dependent dynamics in SSM parameters, enabling selective propagation or forgetting of information along sequences.

This work represents a significant step forward in designing models that combine efficiency and performance, achieving linear scaling in sequence length and demonstrating up to 5× higher inference throughput compared to Transformers. Mamba eliminates the need for attention or MLP blocks, offering a simplified design suitable for multimodal tasks. It achieves state-of-the-art results across various domains, including language, audio, and genomics, and matches the performance of Transformers twice its size in language modeling tasks.

Mamba is designed to solve the computational inefficiency and scalability issues of Transformers, particularly for handling long sequence data across various modalities (e.g., language, audio, and genomics). The primary challenges that Mamba addresses are:

Inefficiency of Self-Attention in Transformers.

- **Quadratic Complexity:** The self-attention mechanism in Transformers scales quadratically with sequence length, making it computationally expensive and memory-intensive for long sequences [9].
- **Limited Context:** Transformers can only process a finite context window efficiently, which limits their ability to model long-range dependencies.

Limitations of Existing Efficient Architectures.

- Several subquadratic-time models (e.g., linear attention [5], gated convolution [8], recurrent models [1], and structured state space models [3]) have been proposed to improve efficiency. However they have their own drawbacks.
  - They often fail to match the performance of Transformers, especially in discrete and information-dense data such as natural language.
  - They lack content-based reasoning capabilities, which are crucial for tasks like language modeling and in-context learning.

1.1. Mamba Architecture

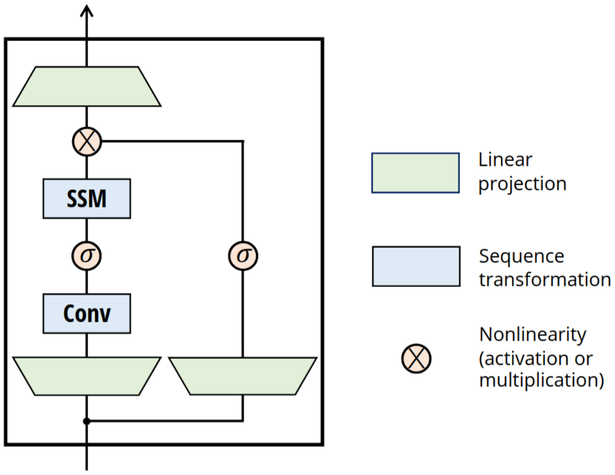

Figure 1. Mambavision Architecture[2]

1.2. MambaVision

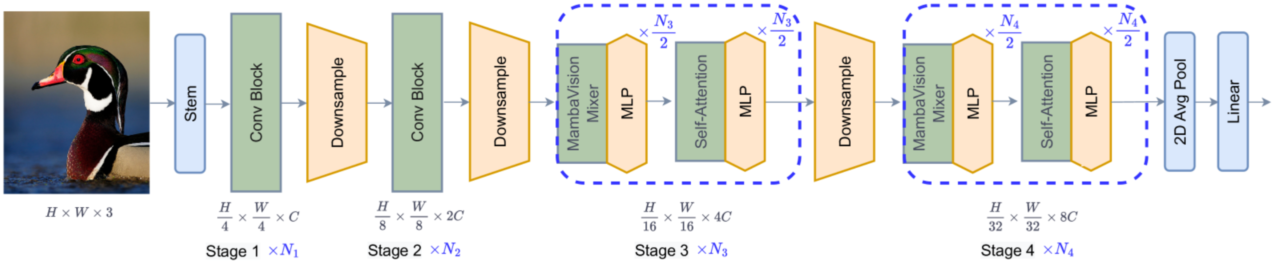

Figure 2. Mambavision Architecture[4]

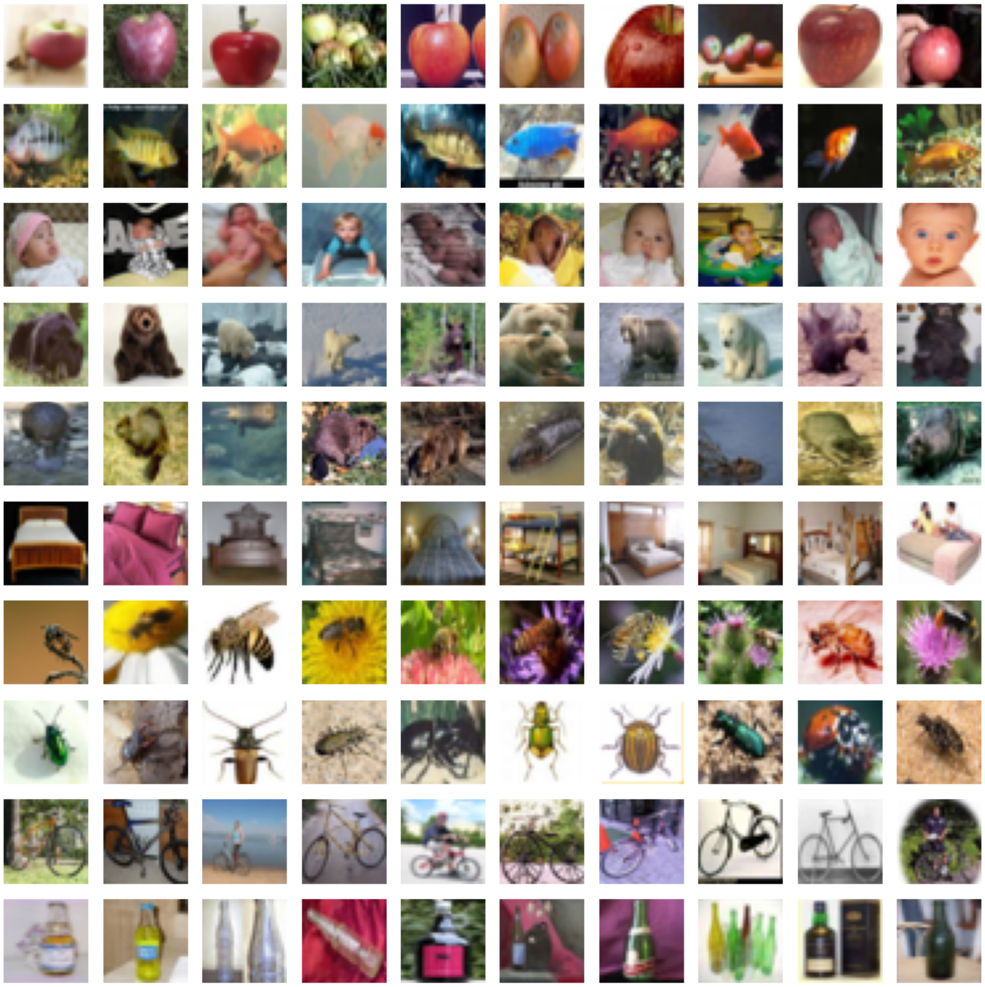

Figure 3. Sample of CIFAR100 Dataset[6]

2. LLaVA - Large Language and Vision Assistant

In this section, LLaVA (Large Language and Vision Assistant), as presented in the paper ”Visual Instruction Tuning” [7], is discussed. LLaVA is an end-to-end trained multimodal AI system that connects a vision encoder with a large language model (LLM) to interpret and follow human instructions involving both visual and linguistic contexts. The work in this paper represents the first attempt to use language-only GPT-4 to generate multimodal language-image instruction-following data, enabling instruction tuning for multimodal tasks.

The paper demonstrates that LLaVA achieves remarkable capabilities in multimodal chat, often exhibiting behaviors similar to multimodal GPT-4 on unseen images and instructions. Evaluations show an 85.1% relative score compared to GPT-4 on a synthetic multimodal instruction-following dataset. Furthermore, when fine-tuned on ScienceQA, LLaVA achieves a new state-of-the-art accuracy of 92.53%, highlighting its effectiveness in reasoning and answering visual and textual queries. To facilitate further research, the authors release the GPT-4-generated visual instruction tuning data, their model, and associated codebase to the public.

2.1. LLaVA Architecture

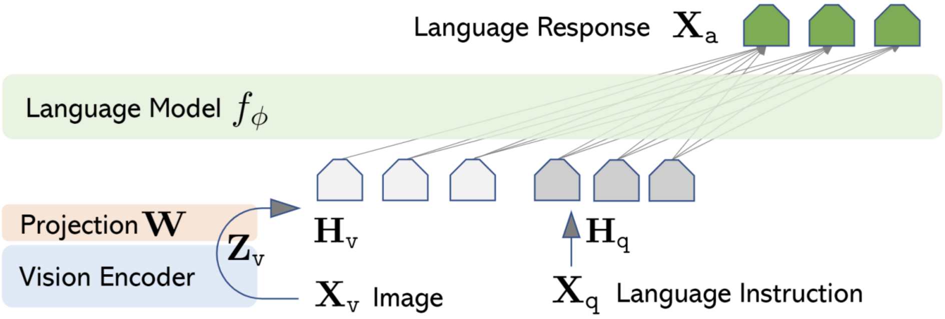

Figure 4. LLaVA Architecture [7]

The LLaVA (Large Language and Vision Assistant) model architecture is designed to integrate visual and linguistic modalities to enable advanced multimodal understanding. Its structure consists of several key components, which work together to process both image and text inputs, generating coherent and context-aware responses.

Vision Encoder. At the foundation of the architecture is the Vision Encoder, which processes the input image X_v. This encoder (e.g., CLIP) transforms the raw image data into a high-dimensional feature representation H_v. These features encapsulate the visual information in a format suitable for downstream processing.

Projection Layer. The Projection Layer serves as a critical bridge between the Vision Encoder and the Language Model. It applies a projection matrix W to convert the feature representations H_v into a format Z_v that is compatible with the language model's embedding space. This alignment ensures seamless integration of visual data with textual data for joint processing.

Language Instruction Input. Alongside the visual input, the model receives a Language Instruction input X_q, which represents the textual task or query. This input is processed by the language model to generate its own feature representation H_q, encapsulating the semantic meaning of the query.

Language Model. At the core of the architecture is the Language Model f_ϕ, which is a pre-trained large language model (e.g., Vicuna). This model takes both the projected visual features Z_v and the linguistic features H_q as input, integrating them to produce a unified understanding of the multimodal context.

Output Generation. The final output, X_a, is a language-based response that incorporates information from both the visual and textual inputs. This response can range from answering specific questions about an image to providing detailed descriptions or engaging in complex reasoning tasks that require a multimodal perspective.

References

[1] Aydar Bulatov, Yuri Kuratov, and Mikhail S. Burtsev. *Recurrent Memory Transformer*. 2022. arXiv: 2207.06881 [cs.CL]. URL: https://arxiv.org/abs/2207.06881.

[2] Albert Gu and Tri Dao. *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. 2024. arXiv: 2312.00752 [cs.LG]. URL: https://arxiv.org/abs/2312.00752.

[3] Albert Gu, Karan Goel, and Christopher Ré. *Efficiently Modeling Long Sequences with Structured State Spaces*. 2022. arXiv: 2111.00396 [cs.LG]. URL: https://arxiv.org/abs/2111.00396.

[4] Ali Hatamizadeh and Jan Kautz. *MambaVision: A Hybrid Mamba-Transformer Vision Backbone*. 2024. arXiv: 2407.08083 [cs.CV]. URL: https://arxiv.org/abs/2407.08083.

[5] Angelos Katharopoulos et al. *Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention*. 2020. arXiv: 2006.16236 [cs.LG]. URL: https://arxiv.org/abs/2006.16236.

[6] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto, 2009.

[7] Haotian Liu et al. *Visual Instruction Tuning*. 2023. arXiv: 2304.08485 [cs.CV]. URL: https://arxiv.org/abs/2304.08485.

[8] Yongming Rao et al. *HorNet: Efficient High-Order Spatial Interactions with Recursive Gated Convolutions*. 2022. arXiv: 2207.14284 [cs.CV]. URL: https://arxiv.org/abs/2207.14284.

[9] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.