PlainMamba: Improving Non-Hierarchical Mamba in Visual Recognition

Chenhongyi Yang*[1]
chenhongyi.yang@ed.ac.uk
Zehui Chen*[2]
lovesnow@mail.ustc.edu.cn
Miguel Espinosa*[1]
miguel.espinosa@ed.ac.uk
Linus Ericsson[1]
linus.ericsson@ed.ac.uk
Zhenyu Wang[3]
2301210449@stu.pku.edu.cn
Jiaming Liu[3]
jiamingliu@stu.pku.edu.cn
Elliot J. Crowley[1]
elliot.j.crowley@ed.ac.uk

[1] School of Engineering,
University of Edinburgh
[2] University of Science and Technology of China
[3] Peking University

Abstract

We present PlainMamba: a simple non-hierarchical state space model (SSM) designed for general visual recognition. The recent Mamba model has shown how SSMs can be highly competitive with other architectures on sequential data and initial attempts have been made to apply it to images. In this paper, we further adapt the selective scanning process of Mamba to the visual domain, enhancing its ability to learn features from two-dimensional images by (i) a *continuous 2D scanning* process that improves spatial continuity by ensuring adjacency of tokens in the scanning sequence, and (ii) *direction-aware updating* which enables the model to discern the spatial relations of tokens by encoding directional information. Our architecture is designed to be easy to use and easy to scale, formed by stacking identical PlainMamba blocks, resulting in a model with constant width throughout all layers. The architecture is further simplified by removing the need for special tokens. We evaluate PlainMamba on a variety of visual recognition tasks, achieving performance gains over previous non-hierarchical models and is competitive with hierarchical alternatives. For tasks requiring high-resolution inputs, in particular, PlainMamba requires much less computing while maintaining high performance. Code and models are available at: https://github.com/ChenhongyiYang/PlainMamba.

1 Introduction

Developing high-performing visual encoders has always been one of the most important goals in computer vision [22, 23, 38, 60, 75, 82, 96]. With high-quality visual features, a broad range of downstream tasks, such as semantic segmentation [11, 86, 95, 107], object

* Equal Contribution

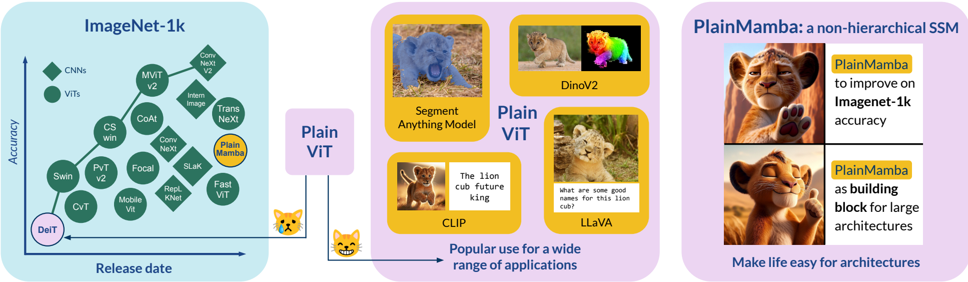


Figure 1: While hierarchical visual encoders may demonstrate superior accuracy on open-source visual recognition benchmarks, the plain non-hierarchical models have had more widespread use because of their simple structure. We investigate the potential of the plain Mamba model in visual recognition.

recognition [38, 61, 82, 96] and detection [39, 54, 69] can be tackled with relative ease. Early methods for extracting visual representations relied on hand-crafted features such as SIFT [62] and SURF [3]. A big breakthrough then came with the adoption of convolutional neural networks (CNNs) that process images with local contexts and enforce spatial equivariance [38, 47, 75]. Recently, vision transformers (ViTs) [23] obviated the need for such enforced inductive biases in favour of learnable contexts that operate on image patches [60, 82, 84]. However, despite the overwhelming success of transformers and their self-attention mechanism [7, 20, 100], the quadratic cost of attention has proved to be an obstacle to further scaling such models.

This has invigorated interest in state space models (SSMs) [31, 34, 59, 92, 108]. Due to their close ties to linear recurrent networks, SSMs have the benefits of potentially infinite context lengths while maintaining linear complexity in the input sequence length [31], offering substantial speedups compared to attention. However, it took several notable advances to make SSMs effective at learning competitive representations, including enforcing state space variables to be orthogonal basis projections [34]. The recent Mamba [31] architecture further aligned SSM-based models with modern transformers, such as making the state space variables input-dependent — much like queries, keys and values in self-attention. When being used for NLP, those designs led to a state space model that could scale to the sizes and performances of modern transformer-based LLMs [7, 83], while improving inference effiency.

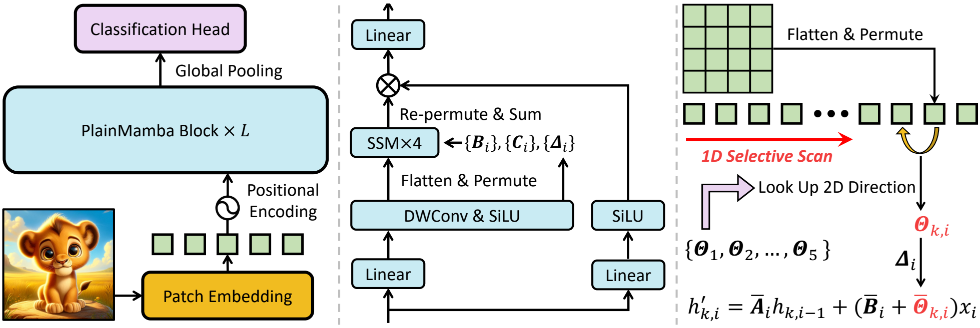
There is now understandable interest in adapting the Mamba architecture to the visual domain [49, 59, 108]. However, before we start doing that, we need to think about under what guidelines should we design our new model. As we show in Figure 1, by examining the development of recently proposed visual encoders, we find that adding more inductive biases, e.g., hierarchical structure, to the plain model such as DeiT can indeed improve a model's performance on open-source benchmarks like ImageNet. However, we should not ignore the fact that the plain ViT [23] is widely used by several popular vision foundation models [26, 46, 56, 65, 67], which suggests that simplicity in architecture design is key for multiple reasons. Firstly, maintaining a constant model width (i.e. non-hierarchical) makes it much easier to integrate features from multiple levels, as is common in dense prediction tasks such as semantic segmentation [46]. It also becomes easier to combine features across different modalities such as in CLIP [67] or LLaVa [56] or as parts of increasingly complex AI-powered systems. Furthermore, simpler components can be more easily optimized for hardware acceleration [16]. In addition, it has also been observed that the over-crafted models may lead to a significant gap between the performance on commonly used benchmarks and downstream tasks [5, 25]. This means benchmark performance may no longer reflect real-world usefulness, as over-engineering tends to increase model complexity and thus make it harder for others to re-use.

Motivated by the above findings, we propose **PlainMamba**: a simple Mamba architecture for visual recognition. This model integrates ideas from CNNs, Transformers and novel SSM-based models with an aim to providing easy-to-use models for the vision modality. Compared to previous visual state space models [59, 108], we simplify the architecture by maintaining constant model width across all layers of the network via stacking identical blocks as well as removing the need for CLS tokens. This allows for easy scaling and model re-use, while achieving competitive performances.

Our contributions are as follows: **(1)** We propose a new visual state space model we call *PlainMamba*. This architecture improves and simplifies previous attempts at extending the Mamba architecture to the visual modality. **(3)** We improve the SSM block by adapting selective scanning to better process 2D spatial inputs, in two ways. (i) Our **continuous 2D scanning** approach ensures that the scanning sequence is spatially continuous to improve semantic continuity. (ii) Our **direction-aware updating**, inspired by positional encoding, allows the model to encode the directionality of each scanning order to further improve spatial context. **(3)**We test our *PlainMamba* architecture using three different sizes (7M, 26M and 50M) and show how they perform competitively on a range of tasks, from ImageNet1K classification to semantic segmentation and object detection. Specifically, we show that PlainMamba outperforms its non-heretical counterparts, including SSMs and Transformers, while performing on par with the hierarchical competitors.

2 Related Work

Visual Feature Extractors. How to effectively extract visual features from images has been a long-standing challenge in computer vision. In the early years of deep learning, CNNs [38, 47, 75, 96] dominated the model architecture landscape. Their induced spatial prior, through the use of convolutional filters, exploits the locality of visual features. Furthermore, stacking multiple layers increases their receptive field. Many different CNN backbone architectures have been proposed over the years [12, 44, 47], introducing new ways of exploiting spatial information [75, 96], building deeper models [38, 79], improving efficiency [68, 73, 80], adding multi-scale connections [71], scaling architectures [93], and introducing attention mechanisms [4, 9, 42, 77, 85, 87]. In recent years, ViTs have become a powerful tool for image modeling [23]. Compared to CNNs, they make fewer assumptions about data (feature locality [99], translation and scale invariance). By replacing the convolutional layers with self-attention modules, transformers can capture global relationships and have achieved state-of-the-art results on many common image benchmarks [19, 52, 107]. To adapt the original transformer architecture [84] for vision tasks, images are split into patches and converted into tokens before being fed into the transformer encoder. Within this framework, numerous works have focused on pushing the performance (e.g. LeViT, [30] combining transformer encoder layers and convolutions), or on reducing the costly quadratic complexity of self-attention [15, 17]. Another popular extension to ViT architectures has been the addition of hierarchical structures [27, 60, 88, 94, 98], similar to the multi-scale feature pyramids used in CNNs. The Swin Transformer [60], for instance, uses shifted windows to share feature information across scales. These multi-scale features are then used for a wide range of downstream tasks. Recent research has explored ways of using these hierarchical features within ViTs themselves [8, 22, 24, 36, 37, 48, 51, 70, 74]. Some works [60] have examined the use of multi-resolution features as attention keys and values to learn multi-scale information. However, these extensions add complexity to the model and make it harder to



(a) Architecture of PlainMamba (b) PlainMamba Block Architecture (c) Direction-aware Updating

Figure 2: (a) The overall architecture of the proposed PlainMamba. PlainMamba does not have a hierarchical structure, it instead stacks L identical PlainMamba block to form the main network. For image classification, it uses global average pooling instead of the CLS to gather global information. (b) Architecture of PlainMamba block, which is similar to the Mamba [31] block where the selective scanning is combined with a gated MLP. (c) The proposed *Direction-Aware Updating*, where a series of learnable parameters $\mathbf{\Theta}_k$ are combined with the data-dependent updating parameters to explicitly inject relative 2D positional information into the selective scanning process.

effectively use its features in later stages, thus hindering widespread adoption. Indeed, recent works [50, 100] return to the original ViT architecture, as its non-hierarchical nature greatly simplifies the use of its features. In particular, the plain ViT provides greater flexibility for pre-training and fine-tuning on different tasks.

State Space Models. State Space Models (SSMs) have emerged as efficient alternatives to transformers and CNNs due to their ability to scale linearly with sequence length [32, 78]. SSMs transform the state space to effectively capture dependencies over extended sequences. To alleviate the initial computational cost of such models, S4 [33] enforced low-rank constraints on the state matrix and S5 [76] introduced parallel scanning to further improve efficiency. Furthermore, H3 [29] achieved competitive results on common benchmarks by improving the hardware utilization. Lastly, Mamba [31] parameterized the SSM matrices as functions of the input, thus allowing it to act as a learnable selection mechanism and providing greater flexibility. Follow-up works have extended selective SSMs for images [2, 57, 58, 63, 66, 72, 90, 91] and videos [64] using a hierarchical structure [59] and bidirectional blocks [108], while Mamba-ND [49] introduces an architecture for multi-dimensional data. MambaIR [35] tackles image restoration, and Pan-Mamba [40] works on pan-sharpening. DiS [28] introduces SSMs to diffusion models by replacing the U-Net with an SSM backbone. While drawing inspiration from the above works, PlainMamba improves Mamba's [31] selective SSM block by adding wider depth-wise convolutions. In contrast to the Cross-Scan Module (CSM) [59] and Mamba-ND [49], PlainMamba respects the spatio-sequential nature of image patches (see Figure 2). As opposed to [108], we do not use the CLS token.

Simplifying Visual Feature Extractors. Simplifying and unifying existing methods is equally important as improving performance. Plain architectures are robust, conceptually simpler, and scale better. ViTs [23] remove the pyramid structure of CNNs by converting images into patched tokens. This way, they easily adapt the transformer architecture for visual tasks. Another trick that stems form sequence modeling is the usage of CLS tokens for prediction, which have proven to be unnecessary for visual tasks [103]. FlexiVit [6] unified into a single architecture images with different input resolutions, and GPViT [100] improved feature resolution with a non-hierarchical transformer. Similarly, ConvNext [61] introduced a simple CNN model that competed with state-of-the-art transformer methods. Other works, like MLP-Mixer [81] and follow-up works [41], have introduced simple architectures using only multi-layer perceptrons. The plain non-hierarchical ViT [23] has served as a simple building block for many diverse tasks. SAM [46] uses a pre-trained ViT as image encoder with minimal changes for image segmentation at large scale. DinoV2 [18, 65] uses a ViT to learn general-purpose visual features by pretraining models on curated datasets with self-supervision. Similarly, the image encoder for the CLIP [67] model consists of a basic ViT with minor modifications, allowing image-text representations to be learned with a contrastive objective. DALLE-2 [26] incorporates a ViT image encoder to extract visual features that are used for text-conditional image generation. LlaVA [55, 56] combines a vision encoder (pretrained ViT from CLIP) and an LLM for vision-language tasks.

3 Method

3.1 Preliminaries

State Space Models. SSMs are typically used to model a continuous linear time-invariant (LTI) system [92] where an input signal $x(t) \in \mathbb{R}$ is mapped to its output signal $y(t) \in \mathbb{R}$ through a state variable $h(t) \in \mathbb{R}^m$ with the following rules:

$$h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t), \quad y(t) = \mathbf{C}h'(t) + \mathbf{D}x(t) \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{m\times m}$, $\mathbf{B} \in \mathbb{R}^{m\times 1}$, $\mathbf{C} \in \mathbb{R}^{1\times m}$ and $\mathbf{D} \in \mathbb{R}^{1\times 1}$ are parameters. To make the above system usable for a discrete system, e.g., a sequence-to-sequence task, a timescale parameter $\mathbf{\Delta}$ is used to transform the parameters $\mathbf{A}$ and $\mathbf{B}$ to their discretized counterparts $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$. In Mamba [31] and its following works [59, 108], this is achieved with the following zero-order hold (ZOH) rule:

$$\bar{\mathbf{A}} = \exp(\mathbf{\Delta}\mathbf{A}), \quad \bar{\mathbf{B}} = (\mathbf{\Delta}\mathbf{A})^{-1}(\exp(\mathbf{\Delta}\mathbf{A}) - \mathbf{I}) \cdot \mathbf{\Delta}\mathbf{B} \tag{2}$$

Afterwards, an input sequence $\{x_i\}$ (for $i = 1, 2, \ldots$) can be mapped to its output sequence $\{y_i\}$ in a similar way:

$$h'_i = \bar{\mathbf{A}}h_{i-1} + \bar{\mathbf{B}}x_i, \quad y_i = \mathbf{C}h'_i + \mathbf{D}x_i \tag{3}$$

Mamba. Since SSMs are often used to model LTI systems, their model parameters are shared by all time steps i. However, as found in Mamba [31], such time-invariant characteristics severely limit the model's representativity. To alleviate this problem, Mamba lifts the time-invariant constraint and makes the parameters $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{\Delta}$ dependent on the input sequence $\{x_i\}$, a process they refer to as the *selective scan*, resulting in the token-dependent $\{\mathbf{B}_i\}$, $\{\mathbf{C}_i\}$ and $\{\mathbf{\Delta}_i\}$. Moreover, the SSM is combined with a gated MLP [43] to gain better representation ability. Specifically, the output sequence $\{y_i\}$ is computed from the $\{x_i\}$ as the following:

$$x'_i = \sigma\big(\text{DWConv}\big(\text{Linear}(x_i)\big)\big), \quad z_i = \sigma\big(\text{Linear}(x_i)\big) \tag{4}$$

$$\mathbf{B}_i, \mathbf{C}_i, \mathbf{\Delta}_i = \text{Linear}(x'_i), \quad \bar{\mathbf{A}}_i, \bar{\mathbf{B}}_i = \text{ZOH}(\mathbf{A}, \mathbf{B_i}, \mathbf{\Delta_i}) \tag{5}$$

$$h'_i = \bar{\mathbf{A}}_i h_{i-1} + \bar{\mathbf{B}}_i x'_i, \quad y'_i = \mathbf{C}_i h'_i + \mathbf{D}x'_i, \quad y_i = y'_i \odot z_i \tag{6}$$

where σ denotes the SiLU activation, and $\odot$ denotes element-wise multiply.

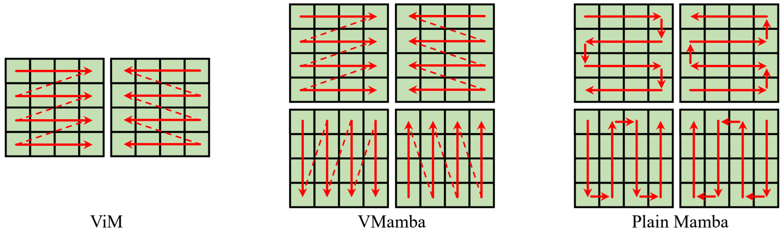


Figure 3: Comparison between our Continuous 2D Scanning and the selective scan orders in ViM [108] and VMamba [59]. Our method makes sure that every scanned visual token is spatially adjacent to its predecessor, avoiding potential spatial and semantic discontinuity.

3.2 Overall architecture of PlainMamba

In Figure 2, we present the model architecture of PlainMamba. Our model is divided into three main components: (1) a convolutional tokenizer that transforms an input 2D image into visual tokens, (2) the main network with a series of L identical PlainMamba blocks to learn visual representations, and (3) a task-specific head for downstream applications.

In more detail, the tokenizer will downsample the input image $I \in \mathbb{R}^{H_I \times W_I \times 3}$ into a list of visual tokens $x \in \mathbb{R}^{H \times W \times C}$, where C is the channel number. We set the default down-sampling factor to 16, following ViT [23]. After combining the initial visual tokens with positional embeddings [84] for retaining spatial information, the tokens undergo a series of transformations through the L PlainMamba blocks, which are designed to simplify usage by maintaining the input-output shape consistency. The final stage of the architecture involves a task-specific head, which is dependent on the particular downstream application. For instance, in image classification tasks, the image tokens are globally pooled into a vector, which is then fed into a linear classification head to produce the final output.

PlainMamba distinguishes itself from existing vision transformers [23, 57] and concurrent vision Mamba [59, 108] architectures in several key aspects. Firstly, it does not use any special tokens, such as the commonly used CLS token. Secondly, in contrast to approaches that adopt a hierarchical structure to manage feature resolution [51, 60, 89], Instead, PlainMamba maintains a constant feature resolution across all blocks. This design choice considers the recent progress made in various visual foundation models [46, 65, 67] where the plain non-hierarchical ViT is used rather than its hierarchical counterparts.

3.3 PlainMamba Block

The overall architecture comprises several identical PlainMamba blocks, forming the backbone for learning high-quality visual features. We present the structure of the PlainMamba block in Figure 2, in which we make several key adjustments to the original Mamba block to fully exploit the two-dimensional nature of image inputs. This adaptation is crucial for effectively transitioning from the inherently 1D processing paradigm of language models to the 2D domain of images. To this end, we introduce two novel techniques: (1) *Continuous 2D Scanning* and (2) *Direction-Aware Updating*. The first technique ensures that each visual token is always adjacent to the previous scanned token. By doing so, it mitigates positional bias and encourages a more uniform understanding of the image space, enhancing the model's ability to learn from visual inputs. The second technique explicitly embeds the 2D relative positional information into the selective scanning process, which allows the model to better interpret the positional context of flattened visual tokens.

Table 1: PlainMamba variants. FLOPs are measured using input size 224×224.

Model	Depth	Channels	Params	FLOPs
PlainMamba-L1	24	192	7.3M	3.0G
PlainMamba-L2	24	384	25.7M	8.1G
PlainMamba-L3	36	448	50.5M	14.4G

Continuous 2D Scanning. The selective scan mechanism is inherently designed for sequential data, such as text. Adapting this mechanism for 2D image data requires flattening the 2D visual tokens into a 1D sequence to apply the State Space Model (SSM) updating rule. Prior research, e.g., VisionMamba [108] and VMamba [59], has demonstrated the efficacy of using multiple scanning orders to enhance model performance — such as both row-wise and column-wise scans in multiple directions. However, as shown in Figure 3 (a) and (b), in these approaches, each scanning order can only cover one type of 2D direction, e.g., left to right, causing spatial discontinuity when moving to a new row (or column). Moreover, as the parameter $\mathbf{A}$ in Equation 3 serves as a decaying term, such spatial discontinuity can also cause adjacent tokens to be decayed to different degrees, compounding the semantic discontinuity and resulting in potential performance drop.

Our *Continuous 2D Scanning* addresses this challenge by ensuring a scanned visual token is always adjacent (in the 2D space) to the previously scanned token. As shown in Figure 3 (c), in our approach, the visual tokens are also scanned in four distinct orders. When reaching the end of a row (or column), the next scanned token will be its adjacent, *not the opposite*, token in the next column (or row). Then, the scanning continues with a reversed direction until it reaches the final visual token of the image. As a consequence, our method preserves spatial and semantic continuity and avoids potential information loss when scanning non-adjacent tokens. Furthermore, in practice the model usually takes input images of the same size, meaning our method can be easily implemented and efficiently run by pre-computing the permutation indexes.

Direction-Aware Updating. As shown in Equation 3, the contribution of a token x_i to the hidden state h_i in the selective scan is determined by the parameter $\bar{\mathbf{B}}_i$, derived from x_i itself. In language models, the sequential order naturally dictates the positional relationship between tokens, allowing the model to "*remember*" their relative positions. However, in our Continuous 2D Scanning, the current token can be in one of four possible directions relative to its predecessor. This challenges the model's ability to discern the precise spatial relationship between consecutive tokens based on $\mathbf{B}_i$ alone. Our *Direction-Aware Updating* is therefore proposed to address this challenge. Drawing inspiration from the relative positional encoding mechanisms in vision transformers [23], we employ a set of learnable parameters $\{\mathbf{\Theta}_k \in \mathbb{R}^{m\times 1}\}$ (for $k = 1,2,...,5$), representing the four cardinal directions plus a special `BEGIN` direction for the initial token. These parameters are summed with the data-dependent $\mathbf{B}_i$ to enrich the selective scan process with directional information. Specifically, with x_i and z_i following Equation 3, our *Direction-Aware Updating* is formulated as follows:

$$h'_{k,i} = \bar{\mathbf{A}}_i h_{k,i-1} + (\bar{\mathbf{B}}_i + \bar{\mathbf{\Theta}}_{k,i})x_i \tag{7}$$

$$y'_i = \sum_{k=1}^{4}\left(\mathbf{C}_i h'_{k,i} + \mathbf{D}x_i\right), \quad y_i = y'_i \odot z_i \tag{8}$$

where k spans the four distinct scanning directions introduced by our *Continuous 2D Scanning*. Alternatively, for the initial token of each scan, we instead add the final $\bar{\mathbf{\Theta}}_{k=5}$ vector. The term $\bar{\mathbf{\Theta}}_{k,i}$ represents the discretized $\mathbf{\Theta}_{k,i}$ using $\mathbf{\Delta}_i$.

3.4 Model Variants of PlainMamba

As shown in Table 1, we present three different model variants of PlainMamba. Specifically, from PlainMamba-L1 to PlainMamba-L2, we scale the model width, i.e., feature channel numbers, and keep the model depth to 24. From PlainMamba-L2 to PlainMamba-L3, we scale both model width and depth. The FLOPs are measured using 224×224 inputs, and we follow the official Mamba codebase to compute the FLOPs of the selective scan process.

4 Experiments

In the main paper, we quantitatively compare PlainMamba with previously proposed models on four visual recognition tasks: image classification, object detection, instance segmentation, and semantic segmentation. Please refer to our supplementary materials for further ablation studies.

Table 2: Comparison between PlainMamba and other models on ImageNet-1K. (* denotes best epoch result.)

Model	Hierarchical	Params	FLOPs	Top-1
CNN				
ResNeXt101-32×4 [97]	✓	44M	8.0G	78.6
ResNeXt101-32×8 [97]	✓	88M	16.5G	79.3
RegNetY-4G [68]	✓	21M	4.0G	80.0
RegNetY-8G [68]	✓	39M	8.0G	81.7
ConvNeXt-T [61]	✓	29M	4.5G	82.1
ConvNeXt-S [61]	✓	50M	8.7G	83.1
Transformer				
DeiT-Tiny [82]	✗	5M	1.3G	72.2
DeiT-Small [82]	✗	22M	4.6G	79.9
DeiT-Base [82]	✗	86M	16.8G	81.8
Swin-Tiny [60]	✓	29M	4.5G	81.3
Swin-Small [60]	✓	50M	8.7G	83.0
PVT-Tiny [88]	✓	13M	2G	75.1
PVT-Small [88]	✓	25M	3.8G	79.8
PVT-Medium [88]	✓	44M	6.7G	81.2
Focal-Tiny [101]	✓	29M	4.9G	82.2
Focal-Small [101]	✓	51M	9.1G	83.5
State Space Modeling				
ViM-T [108]	✗	7M	-	76.1
ViM-S [108]	✗	26M	-	80.5
LocalViM-T [45]	✗	8M	1.5G	76.2
LocalViM-S [45]	✗	28M	4.8G	81.2
Mamba-ND-T [49]	✗	24M	-	79.2
Mamba-ND-S [49]	✗	63M	-	79.4
S4ND-ViT-B [64]	✗	89M	-	80.4
S4ND-ConvNeXt-T [64]	✓	30M	-	82.2
VMamba-T [59]	✓	22M	5.6G	*82.2
VMamba-S [59]	✓	44M	11.2G	*83.5
PlainMamba-L1	✗	7M	3.0G	77.9
PlainMamba-L2	✗	25M	8.1G	81.6
PlainMamba-L3	✗	50M	14.4G	82.3

4.1 Experiment Settings

ImageNet Classification. We build our codebase following [100], which is a commonly used training recipe. Specifically, for the ImageNet-1k experiments, we train all PlainMamba models for 300 epochs using AdamW optimizer. Following [60], we set the batch size to 2048, weight decay to 0.05, and the peak learning rate to 0.002. Cosine learning rate scheduling is used. For data augmentation, we used the commonly used recipe [21, 22, 60, 82], which includes Mixup [104], Cutmix [102], Random erasing [106] and Rand augment [14].

ADE20K Semantic Segmentation. We follow common practice [22, 60, 100] to use UperNet [95] as the segmentation network. Unlike XCiT [1], we do not explicitly resize the constant resolution feature maps into multi-scale. Following [60], we train all models for 160 iterations with batch size 16 and set the default training image size to 512×512.

COCO Object Detection and Instance Segmentation. Following [100], we test PlainMamba's ability on COCO object detection and instance segmentation using both the two-stage Mask R-CNN [39] and the single-stage RetinaNet [54]. For both models, we report the results of both 1× schedule. Following [100], we use ViTAdapter [10] to compute multi-scale features to fit the FPN network structure. We use the commonly used training settings proposed in [60] to keep a fair comparison.

4.2 Main Results

ImageNet-1K Classification. In Table 2, we report the ImageNet-1K experiment results. We compare PlainMamba with three different kinds of visual feature extractors: CNNs, vision transformers, and SSMs. In addition, the comparison includes both hierarchical and non-hierarchical models. Firstly, when comparing with SSMs, our model is doing better than the recently proposed Vision Mamba [108] and Mamba-ND [49]. For example, PlainMamba-L2 achieves a 2.4% higher accuracy than Mamba-ND-T while they share a similar model size. These results validate PlainMamba's effectiveness as a non-hierarchical SSM. Secondly, when compared with CNNs and transformers, our model achieves better performance than the non-hierarchical counterparts. For example, PlainMamba-L2 achieves 1.7% better accuracy with DeiT-Small. Moreover, PlainMamba also achieves similar performance when compared with hierarchical models. For example, when the model size is around 25M, our model achieves 0.3% better accuracy than Swin-Tiny, validating PlainMamba's ability as a general feature extractor. On the other hand, the hierarchical VMamba [59], together with other hierarchical transformers, do achieve a better accuracy than ours. As we explained in Section 1, hierarchical models tend to perform better than non-hierarchical ones in visual recognition. As the main motivation of our work is to develop a simple Mamba architecture, a bit inferior ImageNet accuracy is acceptable.

ADE20K Semantic Segmentation We report our model's ADE20K semantic segmentation performance in Table 3. Similar to the ImageNet-1k and COCO experiments, here the competing models include both hierarchical and non-hierarchical backbones in three types of visual feature extractors. The results again suggest that PlainMamba achieves the best performance among the non-hierarchical models. For example, with similar parameter amounts, PlainMamba-L2 outperforms the high-resolution (patch size of 8) XCiT-S12/8 model [1] with a much lower computation cost. Moreover, PlainMamba-L2 also outperforms the hierarchical Swin-Transformer-Tiny [60], achieving better mIoU while having a lower model size and FLOPs. At the same time, PlainMamba is also doing better than the concurrent Vision Mamba [108]. For instance, PlainMamba-L2 achieves a 1.9 higher mIoU than ViM-S. This result verifies our model's effectiveness in extracting fine-grained visual features, which is essential for the pixel-wise semantic segmentation task.

Table 3: ADE20K semantic segmentation using UperNet. The FLOPs are computed using input size 512×2048.

Backbone	Hierarchical	Params	FLOPs	mIoU
CNN				
ResNet-50 [38]	✓	67M	953G	42.1
ResNet-101 [38]	✓	85M	1030G	44.0
ConvNeXt-T [61]	✓	60M	939G	46.7
Transformer				
DeiT-S+MLN [82]	✗	58M	1217G	43.8
DeiT-B+MLN [82]	✗	144M	2007G	45.5
XCiT-T12/8 [1]	✗	34M	-	43.5
XCiT-S12/8 [1]	✗	52M	1237G	46.6
XCiT-S24/8 [1]	✗	74M	1587G	48.1
Swin-Tiny [60]	✓	60M	945G	44.5
Swin-Small [60]	✓	81M	1038G	47.6
Focal-Tiny [101]	✓	62M	998G	45.8
Focal-Small [101]	✓	85M	1130G	48.0
Twins-SVT-Small [13]	✓	54M	912G	46.2
Twins-SVT-Small [13]	✓	88M	1044G	47.7
State Space Modeling				
ViM-T [108]	✗	13M	-	41.0
ViM-S [108]	✗	46M	-	44.9
LocalVim-T [45]	✗	36M	181G	43.4
LocalVim-S [45]	✗	58M	297G	46.4
VMamba-T [59]	✓	55M	964G	47.3
VMamba-S [59]	✓	76M	1081G	49.5
PlainMamba-L1	✗	35M	174G	44.1
PlainMamba-L2	✗	55M	285G	46.8
PlainMamba-L3	✗	81M	419G	49.1

COCO Object Detection and Instance Segmentation. We report the results of Mask R-CNN object detection and instance segmentation in Table 4. With similar FLOPs and many fewer parameters, PlainMamba-L1 achieves 44.1 AP^{bb} and 39.1 AP^{mk} when using 1× training schedule, while Swin-Small achieves 44.8 AP^{bb} and 40.9 AP^{mk}. We also observe that

Table 4: Mask R-CNN object detection and instance segmentation on MS COCO *mini-val* using 1× schedule. We use ViTAdapter [10] to compute multi-scale features. FLOPs are computed using input size 1280×800.

Backbone	Hierarchical	Params	FLOPs	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
CNN									
ResNeXt101-32x4d [97]	✓	63M	340G	41.9	-	-	37.5	-	-
ResNeXt101-64x4d [97]	✓	102M	493G	42.8	-	-	38.4	-	-
Transformer									
ViT-Adapter-T [10]	✗	29M	349G	41.1	62.5	44.3	37.5	59.7	39.9
ViT-Adapter-S [10]	✗	49M	463G	44.7	65.8	48.3	39.9	62.5	42.8
ViT-Adapter-B [10]	✗	131M	838G	47.0	68.2	51.4	41.8	65.1	44.9
PVT-Small [88]	✓	44M	-	40.4	62.9	43.8	37.8	60.1	40.3
PVT-Medium [88]	✓	64M	-	42.0	64.4	45.6	39.0	61.6	42.1
PVT-Large [88]	✓	81M	-	42.9	65.0	46.6	39.5	61.9	42.5
Swin-Tiny [60]	✓	48M	264G	42.2	-	-	39.1	-	-
Swin-Small [60]	✓	69M	354G	44.8	-	-	40.9	-	-
ViL-Tiny [105]	✓	26M	145G	41.4	63.5	45.0	38.1	60.3	40.8
ViL-Small [105]	✓	45M	218G	44.9	67.1	49.3	41.0	64.2	44.1
ViL-Medium [105]	✓	60M	293G	47.6	69.8	52.1	43.0	66.9	46.6
State Space Modeling									
EfficientVMamba-T [66]	✓	11M	60G	35.6	57.7	38.0	33.2	54.4	35.1
EfficientVMamba-S [66]	✓	31M	197G	39.3	61.8	42.6	36.7	58.9	39.2
EfficientVMamba-B [66]	✓	53M	252G	43.7	66.2	47.9	40.2	63.3	42.9
VMamba-T [59]	✓	42M	262G	46.5	68.5	50.7	42.1	65.5	45.3
VMamba-S [59]	✓	64M	357G	48.2	69.7	52.5	43.0	66.6	46.4
PlainMamba-Adapter-L1	✗	31M	388G	44.1	64.8	47.9	39.1	61.6	41.9
PlainMamba-Adapter-L2	✗	53M	542G	46.0	66.9	50.1	40.6	63.8	43.6
PlainMamba-Adapter-L3	✗	79M	696G	46.8	68.0	51.1	41.2	64.7	43.9

hierarchical models tend to work better than non-hierarchical models. Although our model achieves lower performance than some hierarchical models, e.g., the concurrent VMamba [59], PlainMamba achieves the best performance among its non-hierarchical counterparts. For instance, when using 1× training schedule, PlainMamba achieves 3.1 higher AP^{bb} and 1.6 higher AP^{mk} than DeiT-T when they are both equipped with the ViTAdapter [10]. These results demonstrate that PlainMamba is able to extract good local features, which is important to the object-level tasks like instance segmentation. On the other hand, we also admit that PlainMamba is performing worse than the hierarchical VMamba [59]. We attribute such inferiority to the multi-resolution architecture of FPN-based [53] Mask R-CNN, which is more naturally suitable to the hierarchical designs.

5 Conclusion

We present PlainMamba, a plain SSM-based model for visual recognition. Our model is conceptually simple because it uses no special tokens or hierarchical structure, making it a perfect counterpart to the widely used plain vision transformer. The results show that PlainMamba achieves superior performance to previous non-hierarchical models, including the concurrent SSM-based models, and can perform on par with the high-performing hierarchical models. We hope our model can serve as a baseline for future research in this area.

Acknowledgements

Funding for this research is provided in part by a studentship from the School of Engineering at the University of Edinburgh, the SENSE - Centre for Satellite Data in Environmental Science CDT, and an EPSRC New Investigator Award (EP/X020703/1).

References

[1] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems*, 34:20014–20027, 2021.

[2] Ethan Baron, Itamar Zimerman, and Lior Wolf. 2-D ssm: A general spatial layer for visual transformers. *arXiv preprint arXiv:2306.06635*, 2023.

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. In *ECCV*, 2006.

[4] Irwan Bello. Lambdanetworks: Modeling long-range interactions without attention. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=xTJEN-ggl1b.

[5] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.

[6] Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. FlexiViT: One Model for All Patch Sizes. In *CVPR*, 2023.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *NeurIPS*, 2020.

[8] Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. Regionvit: Regional-to-local attention for vision transformers. In *International Conference on Learning Representations*, 2022.

[9] Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3435–3444, 2019.

[10] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022.

[11] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2022.

[12] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *CVPR*, 2016.

[13] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting Spatial Attention Design in Vision Transformers. *arXiv.org*, April 2021.

[14] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[15] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *ICLR*, 2024.

[16] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

[17] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *NeurIPS*, 2022.

[18] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2023.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

[21] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *Proceedings of the European conference on computer vision*, 2022.

[22] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12124–12134, June 2022.

[23] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[24] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*. PMLR, 2021.

[25] Linus Ericsson, Henry Gouk, and Timothy M Hospedales. How well do self-supervised models transfer? In *CVPR*, 2021.

[26] Aditya Ramesh et al. Hierarchical text-conditional image generation with clip latents, 2022.

[27] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021.

[28] Zhengcong Fei, Mingyuan Fan, Changqian Yu, and Junshi Huang. Scalable diffusion models with state space backbone. *arXiv preprint arXiv:2402.05608*, 2024.

[29] Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. Hungry Hungry Hippos: Towards Language Modeling with State Space Models. In *ICLR*, 2023.

[30] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: A vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[31] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[32] Albert Gu, Isys Johnson, Karan Goel, Khaled Kamal Saab, Tri Dao, Atri Rudra, and Christopher Re. Combining Recurrent, Convolutional, and Continuous-time Models with Linear State Space Layers. In *NeurIPS*, 2021.

[33] Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *ICLR*, 2022.

[34] Albert Gu, Isys Johnson, Aman Timalsina, Atri Rudra, and Christopher Ré. How to train your hippo: State space models with generalized orthogonal basis projections. In *ICLR*, 2023.

[35] Hang Guo, Jinmin Li, Tao Dai, Zhihao Ouyang, Xudong Ren, and Shu-Tao Xia. Mambair: A simple baseline for image restoration with state-space model. In *ECCV*, 2024.

[36] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12175–12185, June 2022.

[37] Ali Hatamizadeh, Hongxu Yin, Jan Kautz, and Pavlo Molchanov. Global context vision transformers. *arXiv preprint arXiv:2206.09959*, 2022.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

[39] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, Oct 2017.

[40] Xuanhua He, Ke Cao, Keyu Yan, Rui Li, Chengjun Xie, Jie Zhang, and Man Zhou. Pan-mamba: Effective pan-sharpening with state space model. *arXiv preprint arXiv:2402.12192*, 2024.

[41] Qibin Hou, Zihang Jiang, Li Yuan, Ming-Ming Cheng, Shuicheng Yan, and Jiashi Feng. Vision permutator: A permutable mlp-like architecture for visual recognition. *TPAMI*, 2022.

[42] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. doi: 10.1109/CVPR.2018.00745.

[43] Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc Le. Transformer quality in linear time. In *ICML*, 2022.

[44] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[45] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan. *arXiv preprint arXiv:2403.09338*, 2024.

[46] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[47] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[48] Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7287–7296, June 2022.

[49] Shufan Li, Harkanwar Singh, and Aditya Grover. Mamba-nd: Selective state space modeling for multi-dimensional data. *arXiv preprint arXiv:2402.05892*, 2024.

[50] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022.

[51] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, June 2022.

[52] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[53] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[54] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, Oct 2017.

[55] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023.

[56] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[57] Jiarun Liu, Hao Yang, Hong-Yu Zhou, Yan Xi, Lequan Yu, Yizhou Yu, Yong Liang, Guangming Shi, Shaoting Zhang, Hairong Zheng, et al. Swin-umamba: Mamba-based unet with imagenet-based pretraining. *arXiv preprint arXiv:2402.03302*, 2024.

[58] Jiarun Liu, Hao Yang, Hong-Yu Zhou, Yan Xi, Lequan Yu, Yizhou Yu, Yong Liang, Guangming Shi, Shaoting Zhang, Hairong Zheng, et al. Swin-umamba: Mamba-based unet with imagenet-based pretraining. *arXiv preprint arXiv:2402.03302*, 2024.

[59] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024.

[60] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[61] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *CVPR*, 2022.

[62] David G Lowe. Distinctive image features from scale-invariant keypoints. In *IJCV*, 2004.

[63] Jun Ma, Feifei Li, and Bo Wang. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024.

[64] Eric Nguyen, Karan Goel, Albert Gu, Gordon Downs, Preey Shah, Tri Dao, Stephen Baccus, and Christopher Ré. S4nd: Modeling images and videos as multidimensional signals with state spaces. *NeurIPS*, 2022.

[65] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[66] Xiaohuan Pei, Tao Huang, and Chang Xu. Efficientvmamba: Atrous selective scan for light weight visual mamba. *arXiv preprint arXiv:2403.09977*, 2024.

[67] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[68] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *CVPR*, 2020.

[69] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[70] Sucheng Ren, Daquan Zhou, Shengfeng He, Jiashi Feng, and Xinchao Wang. Shunted self-attention via multi-scale token aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10853–10862, June 2022.

[71] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*, 2015.

[72] Jiacheng Ruan and Suncheng Xiang. Vm-unet: Vision mamba unet for medical image segmentation. *arXiv preprint arXiv:2402.02491*, 2024.

[73] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *CVPR*, 2018.

[74] Dai Shi. TransNeXt: Robust Foveal Visual Perception for Vision Transformers. In *CVPR*, 2024.

[75] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[76] Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified State Space Layers for Sequence Modeling. In *ICLR*, 2023.

[77] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. *arXiv preprint arXiv:2101.11605*, 2021.

[78] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

[79] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[80] Mingxing Tan and Quoc Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, pages 6105–6114, Long Beach, California, USA, June 2019. PMLR.

[81] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34:24261–24272, 2021.

[82] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[83] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[84] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017.

[85] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual Attention Network for Image Classification. In *CVPR*, 2017.

[86] Kaihong Wang, Donghyun Kim, Rogerio Feris, and Margrit Betke. Cdac: Cross-domain attention consistency in transformer for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11519–11529, 2023.

[87] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[88] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[89] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 2022.

[90] Ziyang Wang and Chao Ma. Semi-mamba-unet: Pixel-level contrastive cross-supervised visual mamba-based unet for semi-supervised medical image segmentation. *arXiv preprint arXiv:2402.07245*, 2024.

[91] Ziyang Wang, Jian-Qing Zheng, Yichi Zhang, Ge Cui, and Lei Li. Mamba-unet: Unet-like pure visual mamba for medical image segmentation. *arXiv preprint arXiv:2402.05079*, 2024.

[92] Robert L Williams, Douglas A Lawrence, et al. *Linear state-space control systems*. John Wiley & Sons, 2007.

[93] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders. In *CVPR*, 2023.

[94] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[95] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision*, pages 418–434, 2018.

[96] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. In *CVPR*, 2016.

[97] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[98] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9981–9990, 2021.

[99] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *Advances in Neural Information Processing Systems*, 34:28522–28535, 2021.

[100] Chenhongyi Yang, Jiarui Xu, Shalini De Mello, Elliot J. Crowley, and Xiaolong Wang. GPViT: A High Resolution Non-Hierarchical Vision Transformer with Group Propagation. In *ICLR*, 2023.

[101] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. In *NeurIPS*, 2021.

[102] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[103] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling Vision Transformers. In *CVPR*, 2022.

[104] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[105] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-Scale Vision Longformer: A New Vision Transformer for High-Resolution Image Encoding. *arXiv.org*, March 2021.

[106] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, 2020.

[107] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.

[108] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.