EfficientVMamba: Atrous Selective Scan for Light Weight Visual Mamba

Xiaohuan Pei*, Tao Huang*, Chang Xu

School of Computer Science, Faculty of Engineering, The University of Sydney, Australia
{xiaohuan.pei, t.huang, c.xu}@sydney.edu.au

Abstract

Prior efforts in light-weight model development mainly centered on CNN and Transformer-based designs yet faced persistent challenges. CNNs adept at local feature extraction compromise resolution while Transformers offer global reach but escalate computational demands $\mathcal{O}(N^2)$. This ongoing trade-off between accuracy and efficiency remains a significant hurdle. Recently, state space models (SSMs), such as Mamba, have shown outstanding performance and competitiveness in various tasks such as language modeling and computer vision, while reducing the time complexity of global information extraction to $\mathcal{O}(N)$. Inspired by this, this work proposes to explore the potential of visual state space models in light-weight model design and introduce a novel efficient model variant dubbed EfficientVMamba. Concretely, EfficientVMamba integrates a atrous-based selective scan approach by efficient skip sampling, constituting building blocks designed to harness both global and local representational features. Additionally, we investigate the integration between SSM blocks and convolutions, and introduce an efficient visual state space block combined with an additional convolution branch, which further elevate the model performance. Experimental results show that, EfficientVMamba scales down the computational complexity while yields competitive results across a variety of vision tasks. For example, EfficientVMamba-S with 1.3G FLOPs improves Vim-Ti with 1.5G FLOPs by a large margin of 5.6% accuracy on ImageNet.

Code — https://github.com/TerryPei/EfficientVMamba

Introduction

Convolutional networks, exemplified by models such as ResNet (He et al. 2016), Inception (Szegedy et al. 2017), and EfficientNet (Tan and Le 2019a), and Transformer-based networks, such as Swin-Transformer (Liu et al. 2021), Beit (Bao et al. 2021), and Resformer (Zamir et al. 2022) have been extensively applied to visual tasks including image classification, detection, and segmentation, achieving remarkable results. Recently, Mamba (Gu and Dao 2023), a network based on state-space models (SSMs) (Gu et al. 2021; Gu, Goel, and Ré 2021; Gupta, Gu, and Berant 2022; Li et al. 2022a), has demonstrated competitive performance to Transformers (Vaswani et al. 2017) in sequence modeling tasks such as language modeling. Inspired by this, some works (Zhu et al. 2024; Liu et al. 2024b; Ruan and Xiang 2024; Xing et al. 2024; Liu et al. 2024a) are pioneering in introducing SSMs into vision tasks. Among these methods, Vmamba (Liu et al. 2024b) stands out by introducing an SS2D method to preserve 2D spatial dependencies by scanning images from multiple directions.

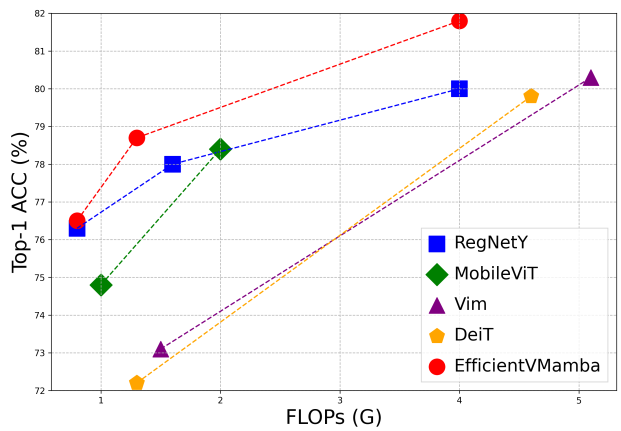

Figure 1: Lightweight Model Performance Comparison on ImageNet. EfficientVMamba outperforms previous work across various model variants in terms of both accuracy and computational complexity.

However, the impressive performance achieved by these various architectures usually comes from the scaling up of model sizes, making a critical challenge in applying them on resource-constrained devices. In pursue of light-weight models, many studies have been conducted to reduce the resource consumption of vision models while keeping a competitive performance. Early works on efficient CNNs mainly focus on narrowing the original convolutional block with efficient group convolutions (Howard et al. 2017; Sandler et al. 2018), light skipping connections (Ma et al. 2018; Han et al. 2020), *e.t.c*. While recently, due to the remarkable success of including the global representation ability of Transformers into vision tasks, some works are proposed to reduce the computation complexity of ViTs (Liu et al. 2021; Tu et al.

*Equal contributions.

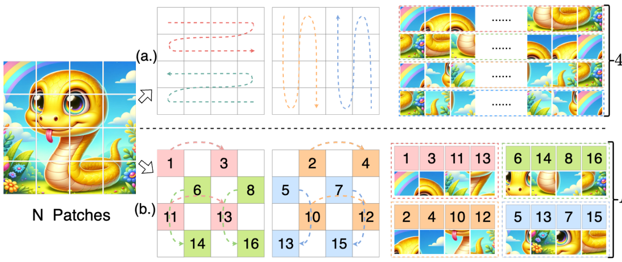

Figure 2: Illustration of efficient 2D scan methods (ES2D). (a.) Vmamba (Liu et al. 2024b) employs SS2D method in vision tasks, traversing entire row or column axes, which incurs heavy computational resources. (b.) We present an efficient 2D scanning method, ES2D, which organizes patches by omitting sampling steps, and then proceeds with an intra-group traversal (with a skipping step of 2 in the Figure). The proposed scan approach reduces computational demands ($4N \rightarrow N$) while preserving global feature maps (*e.g.* Each group contains eye-related patches).

2022; Wang et al. 2021; Huang et al. 2022) and fuse ViTs with CNNs in light-weight models (Mehta and Rastegari 2021; Li et al. 2022b; Tu et al. 2022). However, the lightening of ViTs are usually obtained with the lost of global capture capability in self-attention. Due to the $\mathcal{O}(N)$ time complexity of global self-attention, its computation and memory costs increase dramatically at large resolutions. As a result, existing efficient ViT methods have to perform local self-attentions within partitioned windows (Liu et al. 2021; Tu et al. 2022; Huang et al. 2022), or only conduct global self-attentions in deeper stages with low resolutions (Mehta and Rastegari 2021; Li et al. 2022b). The embarrassing trade-off and rollback of ViTs to CNNs hinders the ability to further improve the light-weight models.

In this paper, recalling the previously mentioned linear scaling complexity in SSMs, we are inspired to obtain efficient global capture ability in light-weight vision models by involving SSMs into model design. Its outstanding performance is demonstrated in Figure 1. We achieve this by first introducing a skip-sampling mechanism, which reduces the number of tokens that need to be scanned in the spatial dimension, and saves multiple times of computation cost in sequence modeling of SSMs while keeping the global receptive field among tokens, as illustrated in Figure 2. On the other hand, acknowledging that the convolutions provide a more efficient way for feature extraction in the case when only local representations suffice, we introduce a convolution branch in supplement of the original global SSM branch, and perform feature fusion of them through the channel attention module, SE (Hu, Shen, and Sun 2018). Finally, for an optimal allocation of capabilities of various block types, we construct our network with global SSM blocks in the shallow and high-resolution layers, while adopting efficient convolution blocks (MobileNetV2 blocks (Sandler et al. 2018)) in the deeper layers. The final network, achieving efficient SSM computation and efficient integration of convolutions, has showcased significant improvements compared to previous CNN and ViT based light-weight models through our experiments on image classification, object detection, and semantic segmentation tasks.

In summary, the contributions of this paper are as follows.

1. We propose an atrous-based selective scanning strategy, which is realized through a novel skip sampling and regrouping patched in the spatial respective field. The strategy refines the building blocks to efficiently extract global dependencies while reducing computation complexity ($\mathcal{O}(N) \rightarrow \mathcal{O}(N/p^2)$) with step p.
2. We introduce a dual-pathway module that combines our efficient scanning strategy for global feature capture and a convolution branch for efficient local feature extraction, along with a channel attention module to balance the integration of both global and local features. Besides, we propose a better allocation of SSM and CNN blocks by promoting SSMs in early stages with high resolutions for better global capture, while adopting CNNs in low resolutions for better efficiency.
3. We conduct extensive experiments on image classification, object detection, and semantic segmentation tasks. The results and illustration shown in Figure 2 demonstrate that, EfficientVMamba effectively reduces the FLOPs while achieving significant performance improvements compared to existing light-weight models.

Related Work

Light-weight Vision Models In recent years, the realm of vision tasks has been predominantly governed by Convolutional Neural Networks (CNNs) and Visual Transformer (ViT) architectures. The focus on making these architectures lightweight to enhance efficiency has become a promising research direction. Notable advancements in CNNs, such as ResNet (He et al. 2016), RegNet (Schneider et al. 2017), and DenseNet (Iandola et al. 2014), have significantly improved image classification accuracy. These advancements introduced a need for lightweight architectures (Wang et al. 2018), which has been addressed through various factorization-based methods, making CNNs more mobile-friendly. For instance, separable convolutions introduced by Xception have been instrumental in this regard, leading to the development of state-of-the-art lightweight CNNs, such as MobileNets (Howard et al. 2017), ShuffleNetv2 (Ma et al. 2018), ESPNetv2 (Mehta et al. 2019), MixConv (Tan and Le 2019b), MNASNet (Tan et al. 2019), and GhostNets (Han et al. 2022). These models are not only versatile but also relatively simpler to train. Following CNNs, Transformers have gained significant traction in various vision tasks. The lightweight versions of Transformers have been achieved through diverse methods. On the training front, sophisticated data augmentation strategies and techniques like Mixup (Zhang et al. 2017), CutMix (Yun et al. 2019), and RandAugment (Cubuk et al. 2020) have been employed, as seen in models like CaiT (Touvron et al. 2021b) and DeiT-III (Touvron, Cord, and Jégou 2022), which demonstrate exceptional performance without the need for large proprietary datasets. From the architectural design perspective, efforts have been concentrated on optimizing self-attention input resolution and devising attention mechanisms. Innovations like PVT-v1 (Wang et al. 2021)'s emulation of CNN's feature map pyramid, Swin-T (Liu et al. 2021) and LightViT (Huang et al. 2022)'s hierarchical feature map and shifted-window mechanisms, and the introduction of (multi-scale) deformable attention modules in Deformable DETR (Zhu et al. 2020) exemplify these advancements. There is also NAS for ViTs (Su et al. 2022).

State Space Models The State Space Model (SSM) (Gu et al. 2021; Gu, Goel, and Ré 2021; Gupta, Gu, and Berant 2022; Li et al. 2022a) is a family architecture encapsulates a sequence-to-sequence transformation has the potential to handle tokens with long dependencies, but it is challenging to train due to its high computational and memory usage. Nevertheless, recent works (Gu, Goel, and Ré 2021; Gupta, Gu, and Berant 2022; Gu et al. 2022; Gu and Dao 2023) have enabled deep State Space Models to become progressively more competitive with CNN and Transformer. In particular, S4 (Gu, Goel, and Ré 2021) employs a Normal Plus Low-Rank (NPLR) representation to efficiently compute the convolution kernel by leveraging the Woodbury identity for matrix inversion. Mamba (Gu and Dao 2023) enhances SSMs with input-specific parameterization and a scalable, hardware-optimized algorithm, achieving simpler design and superior efficiency in processing long sequences for language and genomics. Following success of SSM, there has been a surge in applying the framework to computer vision tasks. S4ND (Nguyen et al. 2022) first introduce the SSM blocks into vision tasks, facilitating the modeling of visual data across 1D, 2D, and 3D as continuous signals. Vim (Zhu et al. 2024) pioneers a Mamba-based backbone for vision tasks by leveraging bidirectional state space modeling for global visual context. Similarly, Vmamba (Liu et al. 2024b) introduces a cross-scan module to address the direction-sensitivity issue arising from the differences between 1D sequences and multi-channels images.

Preliminaries

State Space Models (S4)

State Space Models (SSMs) are a general family of sequence model used in deep learning that are influenced by systems capable of mapping one-dimensional sequences in a continuous manner. These models transform input D-dimensional sequence $x(t) \in \mathbb{R}^{L \times D}$ into output sequence $y(t) \in \mathbb{R}^{L \times D}$ by utilizing a learnable latent state $h(t) \in \mathbb{R}^{N \times D}$ that is not directly observable. The mapping could be denoted as:

$$\begin{aligned} h'(t) &= \boldsymbol{A}h(t) + \boldsymbol{B}x(t), \\ y(t) &= \boldsymbol{C}h(t), \end{aligned} \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{N \times N}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are with shape $D \times N$.

Discretization. Discretization aims to convert the continuous differential equations into discrete functions, aligning the model to the input signal's sampling frequency for more efficient computation (Gu et al. 2021). Following the work (Gupta, Gu, and Berant 2022), the continuous parameters ($\boldsymbol{A}$, $\boldsymbol{B}$) can be discretized by zero-order hold rule with a given sample timescale $\Delta \in \mathbb{R}^{D}$:

$$\begin{aligned} &\bar{\boldsymbol{A}} = e^{\Delta \boldsymbol{A}}, \quad \bar{\boldsymbol{B}} = (e^{\Delta \boldsymbol{A}} - \boldsymbol{I})\boldsymbol{A}^{-1}\boldsymbol{B}, \quad \bar{\boldsymbol{C}} = \boldsymbol{C}, \\ &\bar{\boldsymbol{B}} \approx (\Delta \boldsymbol{A})(\Delta \boldsymbol{A})^{-1}\boldsymbol{A}\boldsymbol{B} = \Delta \boldsymbol{B}, \\ &h(t) = \bar{\boldsymbol{A}}h(t-1) + \bar{\boldsymbol{B}}x(t), \\ &y(t) = \bar{\boldsymbol{C}}h(t), \end{aligned} \tag{2}$$

where $\bar{\boldsymbol{A}} \in \mathbb{R}^{N \times N}$, $\bar{\boldsymbol{B}} \in \mathbb{R}^{D \times N}$ and $\bar{\boldsymbol{C}} \in \mathbb{R}^{D \times N}$.

Selective State Space Models (S6)

Mamba (Gu and Dao 2023) improves the performance of SSM by introducing Selective State Space Models (S6), allowing the continuous parameters to vary with the input enhances selective information processing across sequences, which extend the discretization process by selection mechanism:

$$\begin{aligned} &\bar{\boldsymbol{B}} = s_{\boldsymbol{B}}(x), \quad \bar{\boldsymbol{C}} = s_{\boldsymbol{C}}(x), \\ &\Delta = \tau_{\boldsymbol{A}}(\text{Parameter} + s_{\boldsymbol{A}}(x)), \end{aligned} \tag{3}$$

where $s_{\boldsymbol{B}}(x)$ and $s_{\boldsymbol{C}}(x)$ are linear functions that project input x into an N-dimensional space, while $s_{\boldsymbol{A}}(x)$ broadens a D-dimensional linear projection to the necessary dimensions. In terms of visual tasks, VMamba proposed the 2D Selective Scan (SS2D) (Liu et al. 2024b), which maintains the integrity of 2D image structures by scanning four directed feature sequences. Each sequence is processed independently within an S6 block and then being combined to form a comprehensive 2D feature map.

Method

In order to design light-weight models that are friendly to resource-limited devices, we propose EfficientVMamaba, which is summarized in Figure 3. We introduce an efficient selective scan approach to reduce the computational complexity in Section , and build a block considering both global and local feature extraction with the integration of SSMs and CNNs in Section . Regarding the design of architecture, Section then offers an in-depth look at various architectural variations tailored to different model sizes.

Efficient 2D Scanning (ES2D)

In deep neural networks, downsampling via pooling or strided convolution is employed to broaden the receptive field with a lower computational cost; however, this comes at the expense of spatial resolution. Previous works (Yu and Koltun 2015; Tu et al. 2022) demonstrate apply atrous-based strategy benefits broadening the receptive field without sacrificing resolution. Inspired by this observation and aiming to alleviate and light the computational complexity of selective scanning, we propose an efficient 2D scanning (ES2D) method to scale down the visual selective scan block (SS2D) via skipping sampling for each patches on the feature map. Given a input feature map $\boldsymbol{X} \in \mathbb{R}^{C \times H \times W}$, instead of cross-scan whole patches, we skip scan patches with a step size p and partition into selected spatial dimensional features

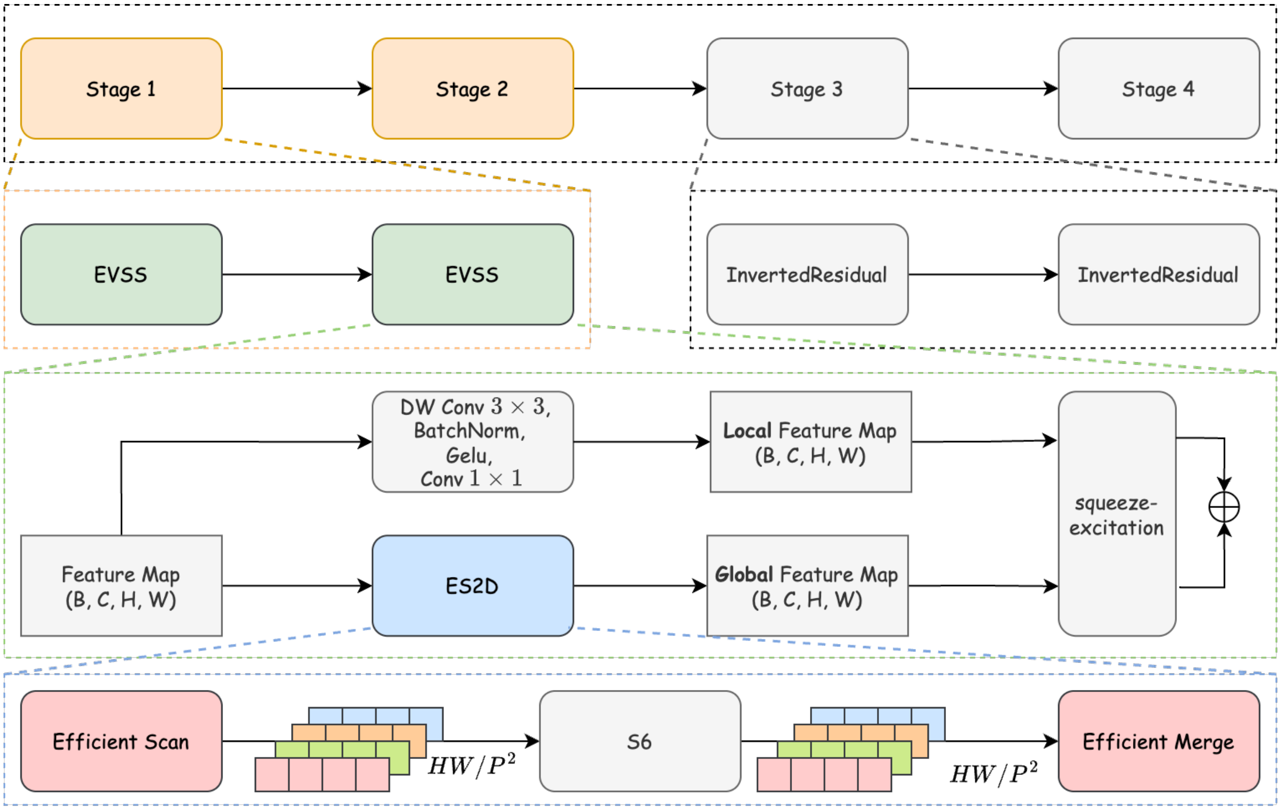

Figure 3: Architecture overview of EfficientVMamba. We hightlight our contributions with corresponding colors in the Figure. (1) ES2D : Atrous-based selective scanning strategy via skip sampling and regrouping in the spatial space. (2) EVSS : The EVSS block merges global and local feature extraction with modified ES2D and convolutional approaches enhanced by Squeeze-Excitation blocks for refined dual-pathway feature representation. Inverted Fusion : Inverted Fusion places local-representation modules in deep layers, deviating from traditional designs by utilizing EVSS blocks early for global representation and inverted residual blocks later for local feature extraction.

$\{\boldsymbol{O}_i\}_{i=1}^4$:

$$\begin{aligned}
&\boldsymbol{O}_i \xleftarrow{\text{scan}} \boldsymbol{X}[:, m :: p, n :: p], \\
&\{\tilde{\boldsymbol{O}}_i\}_{i=1}^4 \leftarrow \mathrm{SS2D}(\{\boldsymbol{O}_i\}_{i=1}^4), \\
&\boldsymbol{Y}[:, m :: p, n :: p] \xleftarrow{\text{merge}} \tilde{\boldsymbol{O}}_i,
\end{aligned} \tag{4}$$

$$\text{with} \quad (m, n) = \left(\left\lfloor \frac{1}{2} + \frac{1}{2}\sin\left(\frac{\pi}{2}(i-2)\right)\right\rfloor, \left\lfloor \frac{1}{2} + \frac{1}{2}\cos\left(\frac{\pi}{2}(i-2)\right)\right\rfloor\right),$$

where $\boldsymbol{O}_i, \tilde{\boldsymbol{O}}_i \in \mathbb{R}^{C \times \frac{H}{p} \times \frac{W}{p}}$ and the operation $[:, m :: p, n :: p]$ represents slicing the matrix for each channel, starting at m on height (H) and n on width (W), skipping every p steps. The process decomposes the fully scanning method into both local and global sparse forms. Skip sampling for local receptive fields reduces computational complexity by selectively scanning smaller patches of the feature map. With a step size p, we sample the $(C, H/p, W/p)$ patches at intervals of p, compared to (C, H, W) in the SS2D, decreasing the number of tokens processed from N to $\frac{N}{p^2}$ for each scan and merge operation, which improves feature extraction efficiency. Re-grouping for global spatial feature maps in ES2D involves combining the processed patches to reconstruct the global structure of the feature map. This integration captures broader contextual information, balancing local detail and global context in feature extraction. Accordingly, our design is intended to streamline the scanning and merging modules while maintaining the essential benefit of global integration in the state-space architecture, ensuring that the feature extraction remains comprehensive on the spatial axis.

Efficient Visual State Space Block (EVSS)

Based on the efficient selected scan approach, we introduce the Efficient Visual State Space (EVSS) block, which is designed to synergistically merge global and local feature representations while maintaining computational efficiency. It leverages a SqueezeEdit-modified ES2D for global information capture and a convolutional branch tailored to extract critical local features, with both branches undergoing a subsequent Squeeze-Excitation (SE) block (Hu, Shen, and Sun 2018). The ES2D module aims to efficiently abstract global contextual information by implementing an intelligent skipping mechanism presented in . It selectively scans the map with a step size p, reducing redundancy without sacrificing the representational quality of the global context in the resultant spatial dimensional features. Parallel to this, empirical evidence concurs that convolutional operations offer a more proficient approach to feature extraction, particularly in scenarios where local representations are adequate. We introduce the convolutional branch concentrates on discerning fine-grained local details through a 3×3 convolution of stride 1. The subsequent SE block adaptively recalibrates the features, allowing the network to auto re-balanced the local and global respective field on the feature map.

The outputs of the respective SE blocks are combined via element-wise summation to construct the EVSS's output and the dual pathway could be denoted as:

$$\boldsymbol{X}^{l+1} = \mathrm{SE}(\mathrm{ES2D}(\boldsymbol{X}^l)) + \mathrm{SE}(\mathrm{Conv}(\boldsymbol{X}^l)), \quad (5)$$

where $\boldsymbol{X}^l$ represent the feature map of the l-layer and $\mathrm{SE}(\cdot)$ is the Squeeze-Excitation operation. With each pathway utilizing a SE block, the EVSS ensures that the respective features of global and local information are dynamically rebalanced to emphasize the most salient features. This fusion aims to preserve the integrity of both the expansive global perspective and the intricate local specifics, facilitating a comprehensive feature representation.

Inverted Insertion of EfficientNet Blocks

As a well-established consensus, the computational efficiency of convolutional operations is more efficient than that of the global-based block such as Transformer. Prior light-weight work efforts have predominantly employed computation-efficient convolutions in the former stages to scale down the token numbers to reduce computational complexity, subsequently integrating global-based blocks (*e.g.*, Transformer with the computational complexity of $\mathcal{O}(N^2)$) to capture global context in the latter stages. For example, MobileViT (Mehta and Rastegari 2021) adopts pure MobileNetV2 blocks in the first two downsampling stages, while only integrating self-attention operations in the latter stages at low resolutions. EfficientFormer (Li et al. 2022b) introduces two types of base blocks, the convolution-based blocks with local pooling are used in the first three stages, and the transformer-like self-attention blocks are only leveraged in the last stage.

However, the observation is contrast on the Mamba-based block. In the SSM framework, the computational complexity for global representation is $\mathcal{O}(N)$, indicating that placing local representation modules at either the front or the back of the stage could be reasonable. Through empirical observation in Table 5, we found positioning these local-representation modules towards the latter layers of the stage yields better results. This discovery significantly deviates from the design principles of previous CNN-based and Transformer-based lightweight models, thereby we call it `inverted` insertion. Consequently, our designed L stages architecture is an inverted insertion of EfficientNet Blocks (MobileNetV2 blocks with SE modules), which utilizes EVSS blocks in the former two stages to capture global-representation and Inverted Residual blocks (InRes) (Sandler et al. 2018) in the subsequent stages to extract local feature maps:

$$\boldsymbol{X}^{l+1} = \begin{cases} \mathrm{EVSS}(\boldsymbol{X}^l) & \text{if} \quad X^l \in \{\mathrm{stage1}, \mathrm{stage2}\}; \\ \mathrm{InRes}(\boldsymbol{X}^l) & \text{otherwise}, \end{cases} \quad (6)$$

where $\boldsymbol{X}^l$ is the feature map in the l-layer. The inverted insertion design of using the shortcuts directly between the bottlenecks is considerably more memory efficient (Sandler et al. 2018).

	EfficientVMamba-T	EfficientVMamba-S	EfficientVMamba-B
#Dims	[48, 96, 192, 384]	[96, 192, 384, 768]	[96, 192, 384, 768]
Stage 1	EVSS $\times$ 2	EVSS $\times$ 2	EVSS $\times$ 2
Stage 2	EVSS $\times$ 2	EVSS $\times$ 2	EVSS $\times$ 2
Stage 3	InRes $\times$ 4	InRes $\times$ 4	InRes $\times$ 9
Stage 4	InRes $\times$ 2	InRes $\times$ 2	InRes $\times$ 2
Params. (M)	6	11	33
FLOPs (G)	0.8	1.3	4.0

Table 1: Model variants of EfficientVMamba.

Model Variants

To sufficiently demonstrate the effectiveness of our proposed model, we detail architectural variants rooted in plain structures as referenced in (Zhu et al. 2024). These variants are designated as EfficientVMamba-T, EfficientVMamba-S, and EfficientVMamba-B, shown as Table 1, corresponding to different scales of the model. EfficientVMamba-T is the most lightweight with 6M parameters, followed by EfficientVMamba-S with 11M, and EfficientVMamba-B being the most complex with 33M. In terms of computational load, measured in FLOPs, the models exhibit a parallel increase with 0.8G for EfficientVMamba-T, 1.3G for EfficientVMamba-S, and 4.0G for EfficientVMamba-B, correlating directly with their complexity and feature size.

Experiments

To rigorously evaluate the performance of our diverse model variants, we demonstrate the results of image classification, object detection, and semantic segmentation tasks. Furthermore, we pursued ablation study to comprehensively examine the effects of atrous selective scanning, the impact of SSM-Conv fusion blocks, and the implications of incorporating convolution blocks at different stages of the models.

ImageNet Classification

Training strategies. Following previous works (Touvron et al. 2021a; Liu et al. 2021; Zhu et al. 2024; Liu et al. 2024b), we train our models for 300 epochs with a base batch size of 1024 and an AdamW optimizer, a cosine annealing learning rate schedule is adopted with initial value 10^{-3} and 20-epoch warmup. For training data augmentation, we use random cropping, AutoAugment (Cubuk et al. 2019) with policy *rand-m9-mstd0.5*, and random erasing of pixels with a probability of 0.25 on each image, then a MixUp (Zhang et al. 2017) strategy with ratio 0.2 is adopted in each batch. An exponential moving average on model is adopted with decay rate 0.9999.

Tiny models (<1 GFLOPs). In the pursuit of efficiency, the results of tiny models are shown in Table 2. EfficientVMamba-T achieves state-of-art performance with a Top-1 accuracy of 76.5%, rivalling its counterparts that demand higher computational costs. With a modest expenditure of only 0.8 GFLOPs, our model surpasses the PVTv2-B0 by a 6% margin in accuracy and outperforms the MobileViT-XS by 1.7%, all with less computational demand.

Method	Image size	Params (M)	FLOPs (G)	Top-1 (%)
RegNetY-800M (Radosavovic et al. 2020)	224^2	6	0.8	76.3
PVTv2-B0 (Wang et al. 2022)	224^2	3	0.6	70.5
MobileViT-XS (Mehta and Rastegari 2021)	224^2	2	1.0	74.8
LVT (Yang et al. 2022)	224^2	6	0.9	74.8
EfficientVMamba-T	224^2	6	0.8	76.5
RegNetY-1.6G (Radosavovic et al. 2020)	224^2	11	1.6	78.0
DeiT-Ti (Touvron et al. 2021a)	224^2	6	1.3	72.2
MobileViT-S (Mehta and Rastegari 2021)	224^2	6	2.0	78.4
Vim-Ti (Zhu et al. 2024)	224^2	7	1.5	73.1
EfficientVMamba-S	224^2	11	1.3	78.7
RegNetY-4G (Radosavovic et al. 2020)	224^2	21	4.0	80.0
DeiT-S (Touvron et al. 2021a)	224^2	22	4.6	79.8
Swin-T (Liu et al. 2021)	224^2	29	4.5	81.3
Vim-S (Zhu et al. 2024)	224^2	26	5.1	80.3
VMamba-T (Liu et al. 2024b)	224^2	22	5.6	82.2
EfficientVMamba-B	224^2	33	4.0	81.8

Table 2: Comparison of different backbones on ImageNet-1K classification.

Small models (1-2 GFLOPs). Our EfficientVMamba-S exhibits a significant improvement in accuracy, achieving a Top-1 accuracy of 78.7%. This represents a substantial increase over DeiT-Ti and MobileViT-S, which achieve 72.2% and 78.4% respectively. Notably, EfficientVMamba-S maintains this high accuracy level with computational efficiency, requiring only 1.3 GFLOPs, which is on par with DeiT-Ti and lower than MobileViT-S's 2.0 GFLOPs.

Base models (4-5 GFLOPs). EfficientVMamba-B achieves an impressive Top-1 accuracy of 81.8%, surpassing DeiT-S by 2% and Vim-S by 1.5%, as indicated in the third group of the Table 2. This base model demonstrates the feasibility of coupling a substantial parameter count of 33 M with a modest computational demand of 4.0 GFLOPs. In comparison, VMamba-T, with a similar parameter count of 22 M requires a higher 5.6 GFLOPs.

Object Detection

Training strategies. We evaluate the efficacy of our EfficientVMamba model for object detection tasks on the MSCOCO 2017 (Lin et al. 2014) dataset. Our evaluation framework relies on the mmdetection library (Chen et al. 2019). For comparisons with light-weight backbones, we follow PvT (Wang et al. 2021) to use RetinaNet as the detector and adopt $1\times$ training schedule. While for comparisons with larger backbones, our experiment follows the hyperparameter settings detailed in Swin (Liu et al. 2021) We use the AdamW optimization method to refine the weights of our pre-trained networks on ImageNet-1K for durations of 12 and 36 epochs. We apply drop path rates of 0.2% across the board for EfficientVMamba-T/S/B variants. The learning rate begins at $1e-5$ and is decreased tenfold at epochs 9 and 11. Multi-scale training and random flipping are implemented during training with a batch size of 16, adhering to standard procedures for evaluating object detection systems.

Results. We summarize the results of RetinaNet detector in Table 3. Remarkably, each variants competitively reducing the sizes while simultaneously exhibits a performance enhancement. The EfficientVMamba-T model stands out with 13M parameters and an AP of 37.5%, slightly higher by 5.7% compared to the ResNet-18, which has $21.3M$ parameters. The performance of EfficientVMamba-T also surpasses PVTv1-Tiny by 0.8% while matching it in terms of parameter count. EfficientVMamba-S, with only 19M parameters, achieves a commendable AP of 39.1%, outstripping the larger ResNet50 model, which shows a lower AP of 36.3% despite having 37.7M parameters. In the higher echelons, EfficientVMamba-B, which boasts 44M parameters, secures an AP of 42.8%, signifying a significant lead over both ResNet101 and ResNeXt101-32x4d, highlighting the efficiency of our models even with a smaller parameter footprint. Notably, PVTv2-b0 with 13M parameters achieves an AP of 37.2%, which EfficientVMamba-T closely follows, indicating competitive performance with a similar parameter budget. For the comparisons with other backbones on Mask R-CNN, see Appendix.

Model	Params (M)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet18 (He et al. 2016)	21.3	31.8	49.6	33.6	16.3	34.3	43.2
PVTv1-Tiny (Wang et al. 2021)	23.0	36.7	56.9	38.9	22.6	38.8	50.0
EfficientViT-M4 (Liu et al. 2023)	8.8	32.7	52.2	34.1	17.6	35.3	46.0
PVTv2-b0 (Wang et al. 2022)	13.0	37.2	57.2	39.5	23.1	40.4	49.7
EfficientVMamba-T	13.0	37.5	57.8	39.6	22.6	40.7	49.1
ResNet50 (He et al. 2016)	37.7	36.3	55.3	38.6	19.3	40.0	48.8
PVTv1-Small (Wang et al. 2021)	34.2	40.4	61.3	43.0	25.0	42.9	55.7
PVTv2-b1 (Wang et al. 2022)	23.8	41.2	61.9	43.9	25.4	44.5	54.3
EfficientVMamba-S	19.0	39.1	60.3	41.2	23.9	43.0	51.3
ResNet101 (He et al. 2016)	56.7	38.5	57.8	41.2	21.4	42.6	51.1
ResNeXt101-32x4d (Xie et al. 2017)	56.4	39.9	59.6	42.7	22.3	44.2	52.5
PVTv1-Medium (Wang et al. 2021)	53.9	41.9	63.1	44.3	25.0	44.9	57.6
EfficientVMamba-B	44.0	42.8	63.9	45.8	27.3	46.9	55.1

Table 3: COCO detection results on RetinaNet.

Backbone	Image size	Params (M)	FLOPs (G)	mIoU (SS)	mIoU (MS)
ResNet-50 (He et al. 2016)	512^2	67	953	42.1	42.8
ResNet-101 (He et al. 2016)	512^2	85	1030	42.9	44.0
Swin-T (Liu et al. 2021)	512^2	60	945	44.4	45.8
Swin-S (Liu et al. 2021)	512^2	81	1039	47.6	49.5
DeiT-S + MLN (Touvron et al. 2021a)	512^2	58	1217	43.8	45.1
DeiT-B + MLN (Touvron et al. 2021a)	512^2	144	2007	45.5	47.2
VMamba-T (Liu et al. 2024b)	512^2	55	964	47.3	48.3
VMamba-S (Liu et al. 2024b)	512^2	76	1095	49.5	50.5
EfficientVMamba-T	512^2	14	230	38.9	39.3
EfficientVMamba-S	512^2	29	505	41.5	42.1
EfficientVMamba-B	512^2	65	930	46.5	47.3

Table 4: Results of semantic segmentation on ADE20K using UperNet (Xiao et al. 2018). We measure the mIoU with single-scale (SS) and multi-scale (MS) testings on the *val* set. The FLOPs are measured with an input size of 512×2048. MLN: multi-level neck.

Semantic Segmentation

Training strategies. Aligning with Vmamba (Liu et al. 2024b) settings, we integrate an UperHead into the pre-trained model structure. Utilizing the AdamW optimizer, we initiate the learning rate at 6×10^{-5}. The fine-tuning stage consists of $160k$ iterations, using a batch size of 16. While the standard input resolution stands at 512×512, we also conduct experiments with 640×640 inputs and apply multi-scale (MS) testing to broaden our evaluation.

Results. The EfficientVMamba-T model yields mIoUs of 38.9% (SS) and 39.3% (MS), surpassing the ResNet-50's 42.1% mIoU with far fewer parameters. EfficientVMamba-S achieves 41.5% (SS) and 42.1 (MS) mIoUs, bettering the DeiT-S + MLN despite having a lower computational footprint. The EfficientVMamba-B reaches 46.5% (SS) and 47.3% (MS), outperforming the heavier VMamba-S. These findings attest to the EfficientVMamba series' balance of accuracy and efficiency in semantic segmentation.

Ablation Study

Effect of atrous selective scan. We implement experiment to validate the efficacy of atrous selective scan in Table 6. The upgrade from SS2D to ES2D significantly reduces the computational complexity from 0.8 GFLOPs while retains competitive accuracy at 73.6%, a 1.5% improvement on the tiny variant. Similarly, In the case of base variant, the model utilizing ES2D not only reduces the GFLOPs to 4.0 from VMamba-B's 4.2 but also exhibits an increase in accuracy from 80.2% to 80.9%. The results suggest that the incorporation of ES2D in our EfficientVMamba models is one of the key factor in achieving the reduction of computational complexity by skip sampling while preserve the global respective field to keep competitive performance. The reduction of GLOPs also reveals the potency of ES2D in maintaining, and even enhancing, model accuracy while significantly reducing computational overhead, demonstrating its viability for resource-constrained scenarios.

Effect of SSM-Conv fusion block. The integration of a convolutional branch following with a SE block enhances the performance of our model. For tiny variance, adding the local fusion feature extraction improves accuracy from 73.6% to 75.1%. In the case of EfficientVMamba-B, introducing fusion mechanism increases accuracy from 80.9% to 81.2%. The observed performance gains reveals the additional convolutional branch enhance the local feature extraction. By integrating Fusion, the models likely benefit from a more diversified feature set that captures a wider range of spatial details, improving the model's ability to generalize and thus boosting accuracy. This suggests that the strategic addition of such branches can effectively enhance the model's performance by providing a comprehensive and more nuanced respective field of the input feature map.

Comparisons of injecting convolution block in different stages. In this paper, we obtain an interesting observation that our SSM based block, EVSS, is more beneficial in the early stages of the network. In contrast, previous works on light-weight ViTs usually inject the convolution blocks in the early stages and adopt Transformer blocks in the deep stages. As shown in Table 5, we compare the performance of injecting convolution blocks in different stages of EfficientVMamba-T, and the results indicate that, adopting Inverted Residual blocks in the deep stages with performs better than that in early stages. A explanation to the opposite phenomenons between our light-weight VSSMs and ViTs is that, the self-attention in Transformers has higher computation complexity and thus its computation at high resolutions is inefficient; while the SSMs, tailored for efficient modeling of long sequences, is more efficient and beneficial on capturing information globally at high resolutions.

Model	ES2D	Fusion	InRes	Param (M)	FLOPs (G)	ACC (%)
VMamba-T				2.4	0.9	72.1
EfficientVMamba-T	✓	✗	✗	5	0.8	73.6
	✓	✓	✗	5	0.8	75.1
	✓	✓	✓	6	0.8	76.5
VMamba-B				16	4.2	80.2
EfficientVMamba-B	✓	✗	✗	25	4.0	80.9
	✓	✓	✗	26	4.0	81.2
	✓	✓	✓	33	4.0	81.8

Table 5: Comparisons of injecting convolution blocks at different stages on ImageNet dataset. We use EfficientVMamba-T in the experiments.

Layers	Stage 1	Stage 2	Stage 3	Stage 4	Params (M)	FLOPs (G)	ACC (%)
EVSS only	EVSS	EVSS	EVSS	EVSS	5	0.8	75.1
Previous	InRes	InRes	EVSS	EVSS	5	0.8	75.6
Ours	EVSS	EVSS	InRes	InRes	6	0.8	76.5

Table 6: Ablation Analysis: Evaluating the Efficacy of Enhanced Spatially Selective Dilatation (ES2D), Assessing the Synergistic Effect of Convolutional Branch Fusion Enhanced with Squeeze-and-Excitation (SE), and Investigating the Role of Inverted Residual Blocks in Model Performance. For comparison with the baseline VMamba, we adjust the dimensions and number of layers of it to match the FLOPs.

Conclusion

This paper proposed EfficientVMamba, a lightweight state-space network architecture that adeptly combines the strengths of global and local information extraction, addressing the trade-off between model accuracy and computational efficiency. By incorporating an atrous-based selective scan with efficient skip sampling, EfficientVMamba ensures comprehensive global receptive field coverage while minimizing computational load. The integration of this scanning approach with a convolutional branch, followed by optimization through a Squeeze-and-Excitation module, allows for a robust rebalancing of global and local features. The experimental results affirm that EfficientVMamba not only reduces computational complexity to $\mathcal{O}(N)$ but also delivers competitive performance in various vision tasks.

Acknowledgments

This work was supported in part by the Australian Research Council under Projects DP240101848 and FT230100549.

References

Bao, H.; Dong, L.; Piao, S.; and Wei, F. 2021. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*.

Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. 2019. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2019. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 113–123.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 702–703.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Gu, A.; Goel, K.; Gupta, A.; and Ré, C. 2022. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35: 35971–35983.

Gu, A.; Goel, K.; and Ré, C. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.

Gu, A.; Johnson, I.; Goel, K.; Saab, K.; Dao, T.; Rudra, A.; and Ré, C. 2021. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34: 572–585.

Gupta, A.; Gu, A.; and Berant, J. 2022. Diagonal state spaces are as effective as structured state spaces. *Advances in Neural Information Processing Systems*, 35: 22982–22994.

Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; and Xu, C. 2020. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1580–1589.

Han, K.; Wang, Y.; Xu, C.; Guo, J.; Xu, C.; Wu, E.; and Tian, Q. 2022. GhostNets on heterogeneous devices via cheap operations. *International Journal of Computer Vision*, 130(4): 1050–1069.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

Huang, T.; Huang, L.; You, S.; Wang, F.; Qian, C.; and Xu, C. 2022. Lightvit: Towards light-weight convolution-free vision transformers. *arXiv preprint arXiv:2207.05557*.

Iandola, F.; Moskewicz, M.; Karayev, S.; Girshick, R.; Darrell, T.; and Keutzer, K. 2014. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*.

Li, Y.; Cai, T.; Zhang, Y.; Chen, D.; and Dey, D. 2022a. What Makes Convolutional Models Great on Long Sequence Modeling? *arXiv preprint arXiv:2210.09298*.

Li, Y.; Yuan, G.; Wen, Y.; Hu, J.; Evangelidis, G.; Tulyakov, S.; Wang, Y.; and Ren, J. 2022b. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35: 12934–12949.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Liu, J.; Yang, H.; Zhou, H.-Y.; Xi, Y.; Yu, L.; Yu, Y.; Liang, Y.; Shi, G.; Zhang, S.; Zheng, H.; et al. 2024a. Swin-umamba: Mamba-based unet with imagenet-based pretraining. *arXiv preprint arXiv:2402.03302*.

Liu, X.; Peng, H.; Zheng, N.; Yang, Y.; Hu, H.; and Yuan, Y. 2023. Efficientvit: Memory efficient vision transformer with cascaded group attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14420–14430.

Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; and Liu, Y. 2024b. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Ma, N.; Zhang, X.; Zheng, H.-T.; and Sun, J. 2018. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, 116–131.

Mehta, S.; and Rastegari, M. 2021. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*.

Mehta, S.; Rastegari, M.; Shapiro, L.; and Hajishirzi, H. 2019. Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9190–9200.

Nguyen, E.; Goel, K.; Gu, A.; Downs, G.; Shah, P.; Dao, T.; Baccus, S.; and Ré, C. 2022. S4nd: Modeling images and videos as multidimensional signals with state spaces. *Advances in neural information processing systems*, 35: 2846–2861.

Radosavovic, I.; Kosaraju, R. P.; Girshick, R.; He, K.; and Dollár, P. 2020. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10428–10436.

Ruan, J.; and Xiang, S. 2024. Vm-unet: Vision mamba unet for medical image segmentation. *arXiv preprint arXiv:2402.02491*.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Schneider, N.; Piewak, F.; Stiller, C.; and Franke, U. 2017. RegNet: Multimodal sensor registration using deep neural networks. In *2017 IEEE intelligent vehicles symposium (IV)*, 1803–1810. IEEE.

Su, X.; You, S.; Xie, J.; Zheng, M.; Wang, F.; Qian, C.; Zhang, C.; Wang, X.; and Xu, C. 2022. Vitas: Vision transformer architecture search. In *European Conference on Computer Vision*, 139–157. Springer Nature Switzerland Cham.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; and Le, Q. V. 2019. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2820–2828.

Tan, M.; and Le, Q. 2019a. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, 6105–6114. PMLR.

Tan, M.; and Le, Q. V. 2019b. Mixconv: Mixed depthwise convolutional kernels. *arXiv preprint arXiv:1907.09595*.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021a. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, 10347–10357. PMLR.

Touvron, H.; Cord, M.; and Jégou, H. 2022. Deit iii: Revenge of the vit. In *European Conference on Computer Vision*, 516–533. Springer.

Touvron, H.; Cord, M.; Sablayrolles, A.; Synnaeve, G.; and Jégou, H. 2021b. Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 32–42.

Tu, Z.; Talebi, H.; Zhang, H.; Yang, F.; Milanfar, P.; Bovik, A.; and Li, Y. 2022. Maxvit: Multi-axis vision transformer. In *European conference on computer vision*, 459–479. Springer.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, 568–578.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2022. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3): 415–424.

Wang, Y.; Xu, C.; Xu, C.; Xu, C.; and Tao, D. 2018. Learning versatile filters for efficient convolutional neural networks. *Advances in Neural Information Processing Systems*, 31.

Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; and Sun, J. 2018. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, 418–434.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1492–1500.

Xing, Z.; Ye, T.; Yang, Y.; Liu, G.; and Zhu, L. 2024. Segmamba: Long-range sequential modeling mamba for 3d medical image segmentation. *arXiv preprint arXiv:2401.13560*.

Yang, C.; Wang, Y.; Zhang, J.; Zhang, H.; Wei, Z.; Lin, Z.; and Yuille, A. 2022. Lite vision transformer with enhanced self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11998–12008.

Yu, F.; and Koltun, V. 2015. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6023–6032.

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; and Yang, M.-H. 2022. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5728–5739.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; and Wang, X. 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.