EfficientViT: Enhanced Linear Attention for High-Resolution Low-Computation Visual Recognition

Han Cai[1], **Chuang Gan**[2], **Muyan Hu**[3], **Song Han**[1]
[1]MIT [2]MIT-IBM Watson AI Lab [3]Tsinghua University
https://tinyml.mit.edu

Abstract

Vision Transformer (ViT) has achieved remarkable performance in many vision tasks. However, ViT is inferior to convolutional neural networks (CNNs) when targeting high-resolution mobile vision applications. The key computational bottleneck of ViT is the softmax attention module which has quadratic computational complexity with the input resolution. It is essential to reduce the cost of ViT to deploy it on edge devices. Existing methods (e.g., Swin, PVT) restrict the softmax attention within local windows or reduce the resolution of key/value tensors to reduce the cost, which sacrifices ViT's core advantages on global feature extractions. In this work, we present *EfficientViT*, an efficient ViT architecture for high-resolution low-computation visual recognition. Instead of restricting the softmax attention, we propose to replace softmax attention with linear attention while enhancing its local feature extraction ability with depthwise convolution. EfficientViT maintains global and local feature extraction capability while enjoying linear computational complexity. Extensive experiments on COCO object detection and Cityscapes semantic segmentation demonstrate the effectiveness of our method. On the COCO dataset, EfficientViT achieves 42.6 AP with 4.4G MACs, surpassing EfficientDet-D1 by 2.4 AP while having 27.9% fewer MACs. On Cityscapes, EfficientViT reaches 78.7 mIoU with 19.1G MACs, outperforming SegFormer by 2.5 mIoU while requiring less than 1/3 the computational cost. On Qualcomm Snapdragon 855 CPU, EfficientViT is 3$\times$ faster than EfficientNet while achieving higher ImageNet accuracy.

1 Introduction

Vision Transformer (ViT) [1] has recently demonstrated great success in various computer vision tasks and received considerable attention [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]. Compared to convolutional neural networks (CNNs), ViT enjoys a stronger ability to capture global information and long-range interactions, showing superior accuracy to CNNs, especially when scaling up the training data size and model size [1, 6].

Despite the great success of ViT in the *low-resolution & high-computation* region, ViT is still inferior to CNNs for *high-resolution & low-computation* scenarios. For instance, Figure 1 (left) compares current CNN-based and ViT-based one-stage detectors on the COCO dataset [18]. There is more than an order of magnitude efficiency gap between ViT-based detectors (160G MACs) and CNN-based detectors (6G MACs). This hinders deploying ViT on real-time high-resolution vision applications on edge devices.

The root computational bottleneck of ViT is the softmax attention module, whose computational cost grows quadratically with the input resolution. For example, as shown in Figure 1 (middle), the computational cost of ViT-Small [19] quickly becomes significantly larger than ResNet-152's computational cost as the input resolution increases.

Preprint. Under review.

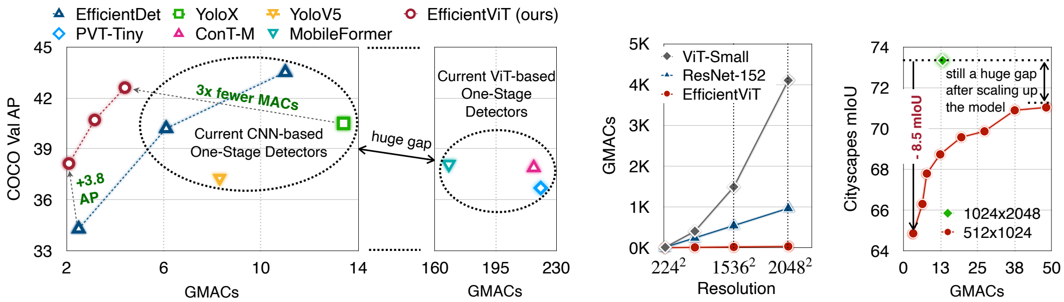

Figure 1: *Left:* Existing ViT-based one-stage detectors are still inferior to current CNN-based one-stage detectors for real-time object detection, requiring an order of magnitude more computation. We introduce the first ViT-based real-time object detector to close the gap. On COCO, EfficientViT achieves 3.8 higher AP than EfficientDet while having lower MACs. Compared with YoloX, EfficientViT saves 67.2% computational cost while providing higher AP. *Middle:* The computational cost of ViT grows quadratically as the input resolution increases, making it unable to handle high-resolution vision applications efficiently. *Right:* High resolution is important for image segmentation. If the input resolution is reduced from 1024x2048 to 512x1024, the mIoU of MobileNetV2 drops by 12% (8.5 mIoU). Scaling up the model size without scaling up the resolution can not close the performance gap.

A straightforward approach to address this issue is to reduce the input resolution. However, high-resolution visual recognition is essential in many real-world computer vision applications such as autonomous driving, medical image processing, etc. Small objects and fine details in the images will vanish when reducing the input resolutions, causing dramatic performance loss in object detection [20] and semantic segmentation [21]. Figure 1 (right) shows the performance of MobileNetV2 [22] under different input resolutions and width multipliers on the Cityscapes dataset [23]. For instance, decreasing the input resolution from 1024x2048 to 512x1024 hurts the performance by 12% (8.5 mIoU) on Cityscapes. Only scaling up the model size without increasing the resolution cannot recover this performance loss, even with 3.6x higher MACs.

Apart from reducing the resolution, another representative approach is to restrict the softmax attention by limiting its scope within fixed-size local windows [4, 24] or reducing the dimension of key/value tensors [5, 9]. However, it hurts ViT's non-local attention capability and reduces the global receptive field (the most crucial merit of ViT), making ViT less distinguishable from large-kernel CNNs [25, 26, 27].

This paper introduces *EfficientViT*, an efficient ViT architecture to address these challenges. We find that it is not necessary to stick to softmax attention. Instead, we propose to substitute softmax attention with linear attention [28]. The key benefit of linear attention is that it maintains the full N^2 attention map like softmax attention. Meanwhile, it leverages the associative property of matrix multiplication to avoid explicitly computing the full attention map while preserving the same functionality (Section 3.1). As such, it maintains the global feature extraction capacity of softmax attention with only linear computational complexity. Another key merit of linear attention is that it avoids softmax, making it much more efficient on mobile (Figure 2 left).

However, directly applying linear attention has drawbacks. Previous studies [29, 30, 31, 32, 33, 34] suggest that there is a significant performance gap between linear attention and softmax attention (Figure 5 middle). Delving into the detailed formulations of linear attention and softmax attention (Section 3.1), one key difference is that linear attention lacks the non-linear attention score normalization scheme. It makes linear attention unable to effectively concentrate its attention distribution on high attention scores produced by local patterns (Figure 2 middle, Figure 6), thereby weakening its local feature extraction capacity. We argue this is the main limitation of linear attention, making its performances inferior to softmax attention. We propose a simple yet effective solution to address this limitation while maintaining linear attention's advantages in low complexity and low hardware latency. Specifically, we propose to enhance linear attention by inserting an extra depthwise convolution in each FFN layer (Figure 2 right). As such, we do not need to rely on linear attention for local feature extraction, avoiding its weakness in capturing local features and taking advantage of its strength in capturing global features.

We extensively evaluated the effectiveness of EfficientViT on various vision tasks under low computation budget, including COCO [18] object detection, Cityscapes [23] semantic segmentation and ImageNet [35] classification. We would like to highlight the efficient backbone design, so we didn't include any add-on techniques that are orthogonal (e.g., knowledge distillation [36], neural architecture search [37]). Still, EfficientViT provides 2.4 higher AP than EfficientDet-D1 [38] on COCO val2017 while saving 27.9% computational cost. On Cityscapes, EfficientViT provides 2.5 higher mIoU than SegFormer [9] while reducing the computational cost by 69.6%. On ImageNet, EfficientViT achieves 79.7% top1 accuracy with 584M MACs, outperforming EfficientNet-B1 [39]'s accuracy while saving 16.6% computational cost.

Unlike existing mobile ViT models [40, 41, 42] that target reducing the parameter size or MACs, we target latency reduction on mobile devices. Our model does not involve complicated dependency [40] or hardware inefficient operations [41, 42]. Thus, our computational cost reduction can easily translate to latency reduction on mobile devices. On Qualcomm Snapdragon 855 CPU, EfficientViT runs 3× faster than EfficientNet while providing higher ImageNet accuracy. Our code and pre-trained models will be released to the public upon publication. We hope our study can facilitate the development of ViT for mobile vision. We summarize our contributions below:

- We are the first to investigate high-resolution low-computation visual recognition using ViT architecture. We perform an in-depth analysis on the bottlenecks of ViT and show that linear attention is a strong alternative to softmax attention and is more hardware-friendly. It alerts us to rethink the necessity of softmax attention in ViT.
- We propose enhanced linear attention to address linear attention's limitation of local feature extraction. Our enhanced linear attention shows strong capacity in visual feature extraction while maintaining low complexity and high hardware efficiency (Figure 5 left).
- We build EfficientViT based on our enhanced linear attention. On three representative vision tasks (COCO object detection, Cityscapes semantic segmentation, ImageNet classification), EfficientViT provides significant improvements over state-of-the-art methods (e.g., EfficientDet, SegFormer, EfficientNet) without add-on techniques (e.g., neural architecture search and knowledge distillation). To the best of our knowledge, EfficientViT is the first ViT-based model that outperforms state-of-the-art CNN-based models in mobile object detection.

2 Related Work

Vision Transformer. Inspired by the great success of Transformer in natural language processing (NLP), Vision Transformer recently gained lots of interest and has been applied to various computer vision tasks, including image classification [1, 3, 4, 43], object detection [2, 7, 44], semantic segmentation [5, 8, 9, 45], pose estimation [10], etc. Unlike CNNs, ViT relies on the softmax attention module that directly models the interaction between each pair of tokens in the feature map to aggregate spatial information. Therefore, ViT can better capture long-range interaction and global information than CNNs. However, this does not come for free. ViT has a higher computational complexity with the input resolution than CNNs (Figure 1 middle), making it computationally prohibitive to use ViT in high-resolution vision applications.

One representative approach for tackling this challenge is restricting softmax attention within fixed-size (e.g., 7x7) local windows [4, 24], reducing the computational complexity from quadratic to linear. Another representative approach [5, 9] is to decrease the resolution of the key/value tensors, which reduces the cost by a fixed factor. Apart from these two representative approaches, [46] employs structured sparse softmax attention, and [47] approximates the softmax attention by factorizing it into two functions to reduce the cost. While these models can handle high-resolution images, they sacrifice ViT's core advantages on global feature extractions. In addition, they still rely on softmax in the attention modules, making them unsuitable for mobile vision (Figure 2 left, Figure 5). Our extensive experiments on object detection (Table 1), semantic segmentation (Table 2), and image classification (Table 3) demonstrate that our models are more effective for high-resolution low-computation visual recognition than these models.

Efficient Vision Transformer. Improving the efficiency of ViT is essential for deploying ViT on resource-constrained edge platforms, such as mobile phones, IoT devices, etc. While ViT provides impressive performances in the high-computation region, it is usually inferior to previous efficient

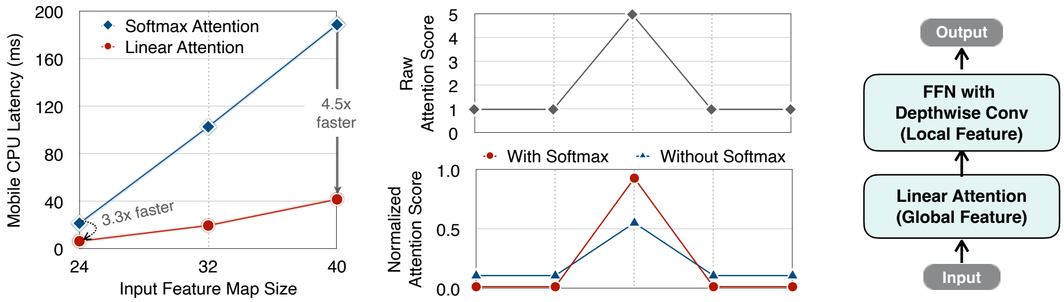

Figure 2: *Left:* Linear attention is 3.3-4.5× faster than softmax attention under similar MACs, thanks to removing the hardware-inefficient softmax function. Latency is measured on Qualcomm Snapdragon 855 CPU with TensorFlow-Lite. We increase the number of heads of linear attention to ensure it has similar MACs as softmax attention. *Middle:* However, linear attention cannot effectively concentrate its attention distribution without the non-linear attention score normalization used in softmax attention, weakening its local feature extraction capacity. Visualization is provided in Figure 6. *Right:* We enhance linear attention with depthwise convolution to address the limitation of linear attention. Depthwise convolution can effectively capture local features, while linear attention can focus on capturing global information. The enhanced linear attention shows strong performances on various vision tasks (Figure 4) while maintaining the efficiency and simplicity of linear attention.

CNNs [39, 48, 49] when targeting the low-computation region. To close the gap, MobileViT [41] proposes to combine the strength of CNN and ViT by replacing local processing in convolutions with global processing using transformers. MobileFormer [40] proposes to parallelize MobileNet and Transformer with a two-way bridge in between for feature fusing. NASViT [42] proposes to leverage neural architecture search to search for efficient ViT architectures. These models provide highly competitive accuracy-efficiency trade-offs on ImageNet. However, they are not suitable for high-resolution vision tasks (Table 1), as they still rely on the softmax attention.

3 Method

In this section, we first review linear attention [28] in NLP and discuss its merits and drawbacks. Next, we introduce a simple yet effective solution to overcome the limitations of linear attention. Finally, we present the detailed architecture of EfficientViT.

3.1 Review of Linear Attention in NLP

Given input $x \in \mathbb{R}^{N \times f}$, the generalized form of softmax attention can be written as:

$$O_i = \sum_{j=1}^{N} \frac{Sim(Q_i, K_j)}{\sum_{j=1}^{N} Sim(Q_i, K_j)} V_j, \quad \text{where } Q = xW_Q, K = xW_K, V = xW_V. \tag{1}$$

$W_Q/W_K/W_V \in \mathbb{R}^{f \times d}$ is the learnable projection matrix. O_i represents the i-th row of matrix O. $Sim(\cdot, \cdot)$ is the similarity function. When using $Sim(Q, K) = \exp(\frac{QK^T}{\sqrt{d}})$, Eq. (1) becomes the softmax attention [1].

While softmax attention has been highly successful in vision [1] and NLP [50, 51], it is not the only choice. For example, linear attention [28] proposes the following similarity function:

$$Sim(Q, K) = \phi(Q)\phi(K)^T, \tag{2}$$

where $\phi(\cdot)$ is the kernel function. In this work, we choose ReLU as the kernel function as it is friendly for hardware. With $Sim(Q, K) = \phi(Q)\phi(K)^T$, Eq. (1) can be rewritten as:

$$O_i = \sum_{j=1}^{N} \frac{\phi(Q_i)\phi(K_j)^T}{\sum_{j=1}^{N} \phi(Q_i)\phi(K_j)^T} V_j = \frac{\sum_{j=1}^{N} (\phi(Q_i)\phi(K_j)^T) V_j}{\phi(Q_i) \sum_{j=1}^{N} \phi(K_j)^T}. \tag{3}$$

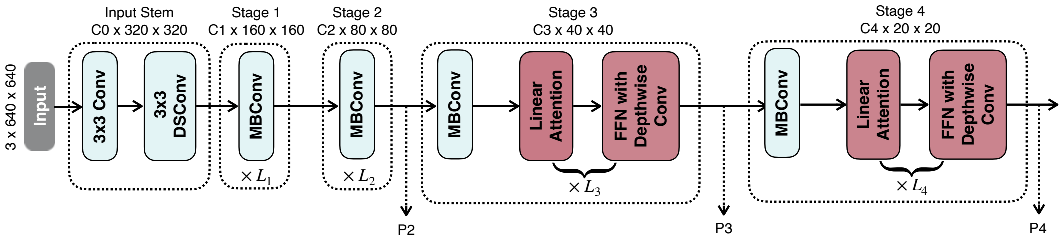

Figure 3: EfficientViT macro architecture. We use the enhanced linear attention starting from stage 3. P2, P3, and P4 form a pyramid of feature maps, which are used in detection and segmentation. P4 is used for classification.

One key merit of linear attention is that it allows leveraging the associative property of matrix multiplication to reduce the computational complexity from quadratic to linear without changing the functionality:

$$O_i = \frac{\sum_{j=1}^{N}(\phi(Q_i)\phi(K_j)^T)V_j}{\phi(Q_i)\sum_{j=1}^{N}\phi(K_j)^T} = \frac{\sum_{j=1}^{N}\phi(Q_i)(\phi(K_j)^TV_j)}{\phi(Q_i)\sum_{j=1}^{N}\phi(K_j)^T} = \frac{\phi(Q_i)(\sum_{j=1}^{N}\phi(K_j)^TV_j)}{\phi(Q_i)(\sum_{j=1}^{N}\phi(K_j)^T)}. \quad (4)$$

According to Eq. (4), we only need to compute $(\sum_{j=1}^{N}\phi(K_j)^TV_j) \in \mathbb{R}^{d\times d}$ and $(\sum_{j=1}^{N}\phi(K_j)^T) \in \mathbb{R}^{d\times 1}$ once, then can reuse them for each query, thereby only requires $\mathcal{O}(N)$ computational cost and $\mathcal{O}(N)$ memory.

In addition to linear complexity, another key merit of linear attention is that it does not involve softmax in the attention modules. Softmax is highly inefficient on hardware. Avoiding it can significantly reduce the latency. For example, Figure 2 (left) shows the latency comparison between softmax attention and linear attention. With similar MACs, linear attention is significantly faster than softmax attention on mobile.

3.2 EfficientViT

3.2.1 Enhancing Linear Attention with Depthwise Convolution

Although linear attention is superior to softmax attention in terms of computational complexity and hardware latency, linear attention has limitations. Previous studies [29, 30, 31, 32] suggest that there is usually a significant performance gap between linear attention and softmax attention in NLP. For vision tasks, previous work [33, 34] also suggests that linear attention is inferior to softmax attention. In our experiments, we also have similar observations (Figure 5 middle).

We challenge this assumption and argue that the inferior performances of linear attention are mainly due to the loss of local feature extraction capacity. Without the non-linear score normalization used in softmax attention, it is difficult for linear attention to concentrate its attention distribution like softmax attention [32, 52, 53, 54]. Figure 2 (middle) provides an example of this difference. Given the same raw attention score, using softmax is better at concentrating than without softmax. Therefore, linear attention cannot effectively focus on high attention scores produced by local patterns (Figure 6), weakening its local feature extraction capacity.

We introduce a simple and effective solution to address this limitation. Our idea is to enhance linear attention with convolution, which is highly effective in local feature extraction. In this way, we do not need to rely on linear attention for capturing local features, and it can focus on global feature extraction. Specifically, to keep linear attention's efficiency and simplicity, we propose to insert a depthwise convolution [55] in each FFN layer (Figure 2 right), which incurs little computational overhead while greatly improving linear attention's local feature extraction capacity (Table 1, 2, 3).

3.2.2 Building Block

Figure 2 (right) demonstrates the detailed architecture of the enhanced linear attention, which consists of a linear attention layer and an FFN layer. A depthwise convolution is inserted into the middle of FFN as discussed in Section 3.2.1.

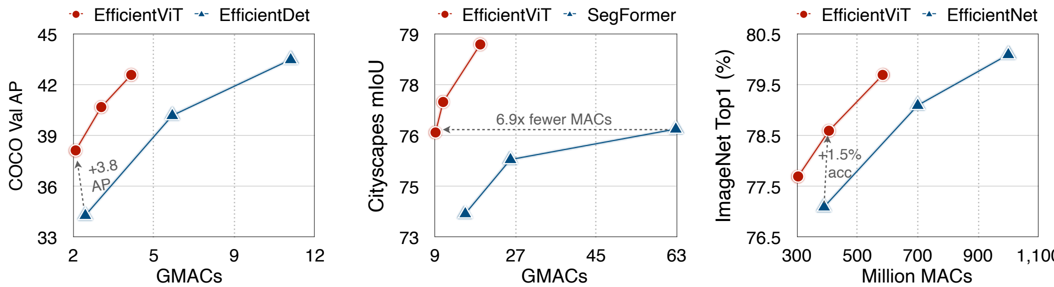

Figure 4: The trade-off between performance and computational cost. On COCO val2017, EfficientViT provides 3.8 higher AP than EfficientDet [38] while having fewer MACs. On Cityscapes, EfficientViT requires 6.9× fewer MACs than SegFormer [9] while achieving similar mIoU. On ImageNet, EfficientViT reaches 1.5% higher top1 accuracy than EfficientNet [39] with slighly higher MACs.

Unlike previous methods [4, 6], we do not use relative position bias in EfficientViT. Although relative position bias can improve the performances, it makes models fragile to the resolution change [9]. Training with multi-resolution or testing under a new resolution is common in detection and segmentation. Removing relative position bias makes EfficientViT more flexible to the input resolution. Unlike the design in previous low-computation CNNs [22,48], we add extra downsampling shortcuts for downsampling blocks. Each downsampling shortcut consists of an average pooling and a 1x1 convolution. In our experiments, these additional downsampling shortcuts can stabilize the training of EfficientViT and improve the performance.

3.2.3 Macro Architecture

Figure 3 illustrates the macro architecture of EfficientViT. It consists of the input stem and 4 stages. Recent studies [6, 56, 57] suggest that using convolution in early stages is better for ViT. We follow this design and start using the enhanced linear attention in stage 3.

To highlight the efficient backbone itself, we keep the hyper-parameters simple using the same expand ratio e for MBConv [22] and FFN ($e = 4$), the same kernel size k for all depthwise convolution ($k = 5$ except the input stem), and the same activation function (hard swish [48]) for all layers. P2, P3, and P4 denote the outputs of stages 2, 3, and 4, forming a pyramid of feature maps. We feed P2, P3, and P4 to the detection head following the common practice. We use YoloX [58] for detection. For segmentation, we fuse P2 and P4. The fused feature is fed to a lightweight head comprising several convolution layers, following Fast-SCNN [21]. For classification, we feed P4 to the lightweight head, same as MobileNetV3 [48].

4 Experiments

4.1 Setups

Datasets. We evaluated EfficientViT on three representative vision datasets, including COCO object detection [18], Cityscapes semantic segmentation [23], and ImageNet classification [35].

Model Architecture. We build our model to have around 400M MACs under a 224x224 input resolution. The macro architecture is illustrated in Figure 3. The detailed configuration is:

- C0=16, C1=24, C2=48, C3=96, C4=192; L1=2, L2=3, L3=5, L4=2

In linear attention, the key/value dimension is 16, while the number of heads is 12/24 in stage 3/4. For normalization, we follow the design of [6], using batch normalization [59] for early stages (input stem, stage 1, and stage 2) and using layer normalization [60] for later stages (stage 3 and stage 4). We refer to this model as EfficientViT-Base in the following section.

Table 1: EfficientViT outperforms state-of-the-art one-stage detectors on COCO (val2017). 'r608' denotes the input resolution is 608x608. † denotes the best result we find for CNN-based mobile object detection, which is achieved with a bunch of additional techniques (e.g., neural architecture search, ghost module, CSP, Cycle-EMA, etc.). Compared with this strong baseline (PP-PicoDet-L [61]), EfficientViT provides 1.7 higher AP with slightly lower MACs.

Models		Params	MACs	AP	AP_{50}	AP_{75}
CNN-based	MobileDet-DSP [62]	9.2M	3.2G	29.1	-	-
	RetinaNet+ResNet50 [63]	34M	97G	39.2	-	-
	EfficientDet-D0 [38]	3.9M	2.5G	34.3	-	-
	EfficientDet-D1 [38]	6.6M	6.1G	40.2	-	-
	YOLOv4-Tiny [64]	6.1M	3.5G	21.7	40.2	-
	YOLOX-Tiny [58]	5.1M	3.2G	32.8	-	-
	YOLOv5s	7.2M	8.3G	37.2	56.0	-
	YOLOX-s [58]	9.0M	13.4G	40.5	-	-
	PP-PicoDet-L† [61]	3.3M	4.5G	40.9	57.6	-
ViT-based	RetinaNet+PVT-Tiny [5]	23.0M	221G	36.7	56.9	38.9
	RetinaNet+ConT-M [65]	27.0M	217G	37.9	58.1	40.2
	RetinaNet+MobileFormer [40]	17.9M	168G	38.0	58.3	40.3
ViT-based	EfficientViT-Det-r416 (ours)	10.6M	2.1G	38.1	55.6	40.3
	EfficientViT-Det-r512 (ours)	10.6M	3.2G	40.7	58.6	43.6
	EfficientViT-Det-r608 (ours)	10.6M	4.4G	**42.6**	**60.5**	**45.3**

Training Details. We use AdamW [66] for training our models. For simplicity, we do not use add-on techniques that are orthogonal to backbone design, such as knowledge distillation [36], neural architecture search [37], etc.

For COCO object detection [18], we train models for 300 epochs with a batch size of 192. We use weights pretrained on ImageNet [35] for initializing the backbone while the detection head is initialized randomly. For data augmentation, we use the setting suggested in [67], including color jitter, random expansion, random crop, and random horizontal flip. We also adopt detection mixup [67] to prevent overfitting.

For Cityscapes semantic segmentation [23], we train models for 485 epochs with a batch size of 16. Same as detection, we use weights pretrained on ImageNet for initializing the backbone while randomly initializing the head. Data augmentation includes random scaling with a ratio of 0.5-2.0, random horizontal flip, and random crop.

For ImageNet classification, we train models for 450 epochs with a batch size of 2048. We use RandAugment [68], Mixup [69], Cutmix [70], StochasticDepth [71] to avoid overfitting. We also use label smoothing with a factor of 0.1.

4.2 COCO Object Detection

Table 1 and Figure 4 (left) report the comparison between EfficientViT and state-of-the-art one-stage object detectors. Compared to previous ViT-based object detectors, EfficientViT provides noticeable improvements in both performance and efficiency. Specifically, EfficientViT requires 38.2× fewer MACs than MobileFormer [40] and provides 4.6 higher AP.

Compared with state-of-the-art CNN-based object detectors (e.g., YoloX [58], EfficientDet [38], PP-PicoDet-L [61]), EfficientViT also provides significant improvements. Specifically, EfficientViT-Det-r608 provides 1.7 AP improvement over PP-PicoDet-L and requires slightly fewer MACs. EfficientViT-Det-r416 provides 3.8 AP improvement over EfficientDet-D0 while reducing the computational cost by 1.2×. In addition, EfficientDet and PP-PicoDet-L are optimized with extra techniques (e.g., neural architecture search, compound scaling, etc.) that are orthogonal to backbone design. In contrast, EfficientViT does not leverage these techniques, thus still has a large room for further improvement.

Table 2: Results on Cityscapes semantic segmentation. ‘r2048’ denotes the input resolution is 1024x2048. Unlike SegFormer [9] that runs inference on 1024x1024 sliding windows, we directly run inference on high-resolution images (1024x2048), thanks to the high efficiency. This brings significant performance improvements. With 19.1G MACs, EfficientViT provides 78.7 mIoU, surpassing SegFormer by 2.5 mIoU while requiring 3.3× fewer MACs.

Models		Backbone	Params	MACs	mIoU
CNN-based	FCN [72]	MobileNetV2	9.8M	158.6G	61.5
	Fast-SCNN [21]	-	1.1M	6.9G	68.0
	PSPNet [73]	MobileNetV2	13.7M	211.7G	70.2
	DeepLabV3+ [74]	MobileNetV2	15.4M	277.7G	75.2
ViT-based	SegFormer [9]	MiT-B0	3.8M	62.8G	76.2
	NASViT [42]	-	-	-	76.1
ViT-based	EfficientViT-Seg-r1408 (ours)	EfficientViT-Base	6.5M	9.1G	76.1
	EfficientViT-Seg-r1536 (ours)	EfficientViT-Base	6.5M	10.8G	77.0
	EfficientViT-Seg-r2048 (ours)	EfficientViT-Base	6.5M	19.1G	**78.7**

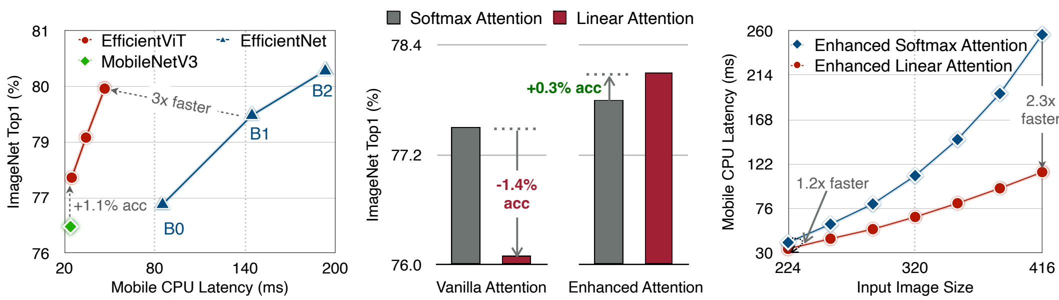

Figure 5: *Left:* Accuracy and latency trade-off on Qualcomm Snapdragon 855. EfficientViT is 3× faster than EfficientNet with higher accuracy. *Middle:* Comparison between softmax attention and linear attention on ImageNet. We observe a significant accuracy gap between softmax attention and linear attention under the same computation. However, the linear attention's accuracy is significantly improved after enhancing the models with depthwise convolution. In contrast, the accuracy of softmax attention does not change a lot. Under the same MAC constraint, the enhanced linear attention provides 0.3% higher accuracy than the enhanced softmax attention. *Right:* The enhanced linear attention is more hardware-efficient and the latency grows slower with the resolution compared with the enhanced softmax attention.

4.3 Cityscapes Semantic Segmentation

Table 2 provides the comparison between EfficientViT and state-of-the-art segmentation models on Cityscapes. Thanks to the high efficiency, EfficientViT can directly run inference on high-resolution images (1024x2048) instead of using 1024x1024 sliding windows as done in SegFormer [9]. This brings significant performance improvements. Specifically, EfficientViT provides 2.5 higher mIoU and reduces the computational cost by 3.3× compared with SegFormer. We also scale down the input resolution of EfficientViT to get multiple models under different MACs constraints. The trade-off curve is illustrated in Figure 4 (middle). Compared with SegFormer, EfficientViT requires 6.9× fewer MACs to achieve a similar mIoU.

4.4 ImageNet Classification

Table 3 and Figure 4 (right) demonstrate the comparison between EfficientViT and state-of-the-art classification models on ImageNet. NASViT [42] and LeViT [56] are not included in Table 3 as they are trained with knowledge distillation [36] and a very long training schedule (e.g., 1000 epochs).

Thanks to the strong visual feature extraction capacity, EfficientViT provides highly competitive performances, though it is not specifically designed for image classification. We highlight that EfficientViT is fast and practical on mobile device (Figure 5 left). Compared with EfficientNet [39],

Table 3: Results on ImageNet classification. 'r224' denotes the input resolution is 224x224. 'w1.2' denotes the width multiplier [22] is 1.2. ‡ denotes the result is from [40]. While EfficientViT is not specifically designed for image classification, it still provides highly competitive performances on ImageNet. With 584M MACs, EfficientViT achieves 79.7% ImageNet top1 accuracy, outperforming EfficientNet-B1 by 0.6% while saving the computational cost by 1.2×. It demonstrates the strong capacity of EfficientViT in visual feature learning.

Models		Params	MACs	Accuracy Top1 (%)	Accuracy Top5 (%)
CNN-based	MobileNetV2 [22]	3.4M	300M	72.0	-
	ShuffleNetV2 1.5x [75]	-	299M	72.6	-
	FBNet-B [76]	4.5M	295M	74.1	-
	ProxylessNAS-Mobile [77]	4.1M	320M	74.6	92.2
	MnasNet-A1 [78]	3.9M	312M	75.2	92.5
	MobileNetV3-Large 1.25x [48]	7.5M	356M	76.6	-
	EfficientNet-B0 [39]	5.3M	390M	77.1	93.3
	EfficientNetV2-B0 [79]	7.4M	700M	78.7	-
	EfficientNet-B1 [39]	7.8M	700M	79.1	94.4
ViT-based	T2T-ViT-7 [80]	4.3M	1.2G	71.7	-
	QuadTree-B-b0 [34]	3.5M	0.7G	72.0	-
	ConViT-Tiny [81]	6.0M	1.0G	73.1	-
	PVT-Tiny [5]	13.2M	1.9G	75.1	-
	CeiT-T [82]	6.4M	1.2G	76.4	93.4
	ViL-Tiny-RPB [46]	6.7M	1.3G	76.7	
	Swin-1G [4] ‡	7.3M	1.0G	77.3	-
	HVT-S-1 [83]	22.1M	2.4G	78.0	93.8
	PiT-XS [84]	10.6M	1.4G	78.1	-
	CoaT Tiny [47]	5.5M	4.4G	78.3	
	HRFormer-T [10]	8.0M	1.8G	78.5	
	MobileViT-XS [41]	2.3M	700M	74.8	-
	MobileFormer w/o DY-ReLU [40]	10.1M	290M	76.8	93.2
ViT-based	EfficientViT-Base-r192 (ours)	7.9M	304M	77.7	93.6
	EfficientViT-Base-r224 (ours)	7.9M	406M	78.6	94.2
	EfficientViT-Base-r224-w1.2 (ours)	10.9M	584M	**79.7**	**94.8**

EfficientViT is 3× faster with higher ImageNet accuracy. Compared to MobileNetV3, EfficientViT provides 1.1% higher accuracy improvement while maintaining a similar latency. The latency is measured on Qualcomm Snapdragon 855 CPU with Tensorflow-Lite, batch size 1.

Compared with MobileFormer [40], EfficientViT provides 0.9% higher ImageNet top1 accuracy with slightly higher MACs. Remarkably, EfficientViT does not involve a complicated two-branch design like MobileFormer, making EfficientViT more friendly for deployment on mobile. Compared with MobileNetV3-Large [48], EfficientViT provides 1.1% higher ImageNet top1 accuracy while requiring fewer MACs. Compared with EfficientNet-B1 [39], EfficientViT achieves 0.6% higher ImageNet top1 accuracy and reduces the computational cost by 1.2×.

4.5 Analysis and Discussion

Visualization. In Figure 6, we visualize the attention maps of softmax attention and linear attention on ImageNet. The input resolution is 224x224. Without the non-linear attention normalization scheme, linear attention can not produce concentrated attention distributions like softmax attention. Linear attention is weaker than softmax attention in capturing local details.

Ablation Study. We study the effectiveness of our enhanced linear attention module in Figure 5 (middle). We train models for 180 epochs with random initialization on ImageNet. We build softmax attention models by replacing linear attention with softmax attention in EfficientViT-Base, while the other modules remain unchanged. We reduce the key/value dimension of softmax attention to 8, as suggested in [42]. We also adjust the number of heads to ensure softmax attention has similar MACs

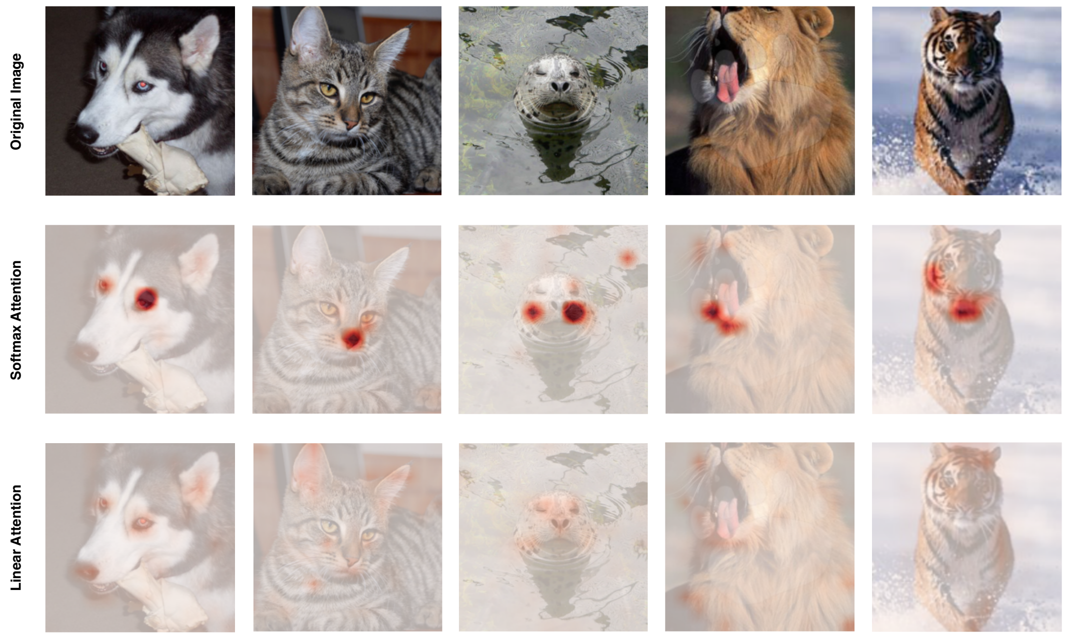

Figure 6: Visualizations of the attention maps that show the limitation of linear attention. With non-linear attention normalization, softmax attention can produce sharp attention distributions, as shown in the middle row. In contrast, linear attention's distributions are relatively smooth, making linear attention weaker at capturing local details and causing significant accuracy loss. We addressed this limitation by enhancing linear attention with depthwise convolution and effectively improved the accuracy (Figure 5 middle).

as linear attention. In addition, we increase the width multiplier [22] of vanilla attention models to ensure they have similar MACs as enhanced attention models.

Align with previous studies [29, 30, 33], we find vanilla linear attention is significantly inferior to vanilla softmax attention, as demonstrated in Figure 5 (middle). In contrast, the enhanced linear attention outperforms the enhanced softmax attention by 0.3% ImageNet top1 accuracy, while having lower complexity with the input resolution and being more efficient on mobile devices (Figure 5 right). It demonstrates the importance of enhancing the local feature extraction ability for linear attention.

5 Conclusion

We proposed EfficientViT for high-resolution low-computation visual recognition. Our study suggests that linear attention is a strong alternative to softmax attention for hardware-friendly visual recognition. However, directly applying linear attention cannot capture local information. Without sacrificing its merits, enhancing it with depthwise convolution can effectively address this limitation. Extensive experiments on three representative vision tasks (COCO, Cityscapes, ImageNet) demonstrate the effectiveness of EfficientViT, significantly outperforming state-of-the-art models.

Limitations, Future Work, and Social Impact. Though our proposed EfficientViT provides strong performances for high-resolution low-computation vision, it is not studied whether our study is still effective in the high-computation scenarios. Future work will scale up EfficientViT to study this question. Regarding negative societal impacts, our study involves GPU resources for training the models, which will result in CO2 emissions.

Acknowledgments

We thank National Science Foundation, MIT-IBM Watson AI Lab, Ford, Intel, Qualcomm, Amazon for supporting this research. Han Cai was partially supported by the Qualcomm Innovation Fellowship.

References

[1] 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. 1, 3, 4

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1, 3

[3] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 1, 3

[4] 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*, pages 10012–10022, 2021. 1, 2, 3, 6, 9

[5] 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*, pages 568–578, 2021. 1, 2, 3, 7, 9

[6] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977, 2021. 1, 6

[7] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. 1, 3

[8] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 3

[9] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 2, 3, 6, 8

[10] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution vision transformer for dense predict. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 3, 9

[11] 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. 1

[12] 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*, pages 22–31, 2021. 1

[13] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Improved multiscale vision transformers for classification and detection. *arXiv preprint arXiv:2112.01526*, 2021. 1

[14] 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. *arXiv preprint arXiv:2107.00652*, 2021. 1

[15] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *arXiv preprint arXiv:2106.13112*, 2021. 1

[16] 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. *arXiv preprint arXiv:2107.00641*, 2021. 1

[17] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. *arXiv preprint arXiv:2201.00520*, 2022. 1

[18] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 3, 6, 7

[19] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 1

[20] Ji Lin, Wei-Ming Chen, Han Cai, Chuang Gan, and Song Han. Memory-efficient patch-based inference for tiny deep learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021. 2

[21] Rudra PK Poudel, Stephan Liwicki, and Roberto Cipolla. Fast-scnn: Fast semantic segmentation network. *arXiv preprint arXiv:1902.04502*, 2019. 2, 6, 8

[22] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 2, 6, 9, 10

[23] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2, 3, 6, 7

[24] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. *arXiv preprint arXiv:2111.09883*, 2021. 2, 3

[25] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022. 2

[26] Xiaohan Ding, Xiangyu Zhang, Yizhuang Zhou, Jungong Han, Guiguang Ding, and Jian Sun. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022. 2

[27] Yihan Wang, Muyang Li, Han Cai, Wei-Ming Chen, and Song Han. Lite pose: Efficient architecture design for 2d human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022. 2

[28] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 2, 4

[29] Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, and Luke Zettlemoyer. Luna: Linear unified nested attention. In *Advances in Neural Information Processing Systems*, volume 34, 2021. 2, 5, 10

[30] Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah Smith, and Lingpeng Kong. Random feature attention. In *International Conference on Learning Representations*, 2021. 2, 5, 10

[31] Hongyu Ren, Hanjun Dai, Zihang Dai, Mengjiao Yang, Jure Leskovec, Dale Schuurmans, and Bo Dai. Combiner: Full attention transformer with sparse computation cost. In *Advances in Neural Information Processing Systems*, volume 34, 2021. 2, 5

[32] Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. In *International Conference on Learning Representations*, 2022. 2, 5

[33] Hugo Germain, Vincent Lepetit, and Guillaume Bourmaud. Visual correspondence hallucination. In *International Conference on Learning Representations*, 2022. 2, 5, 10

[34] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *International Conference on Learning Representations*, 2022. 2, 5, 9

[35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 3, 6, 7

[36] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3, 7, 8

[37] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017. 3, 7

[38] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 3, 6, 7

[39] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 3, 4, 6, 8, 9

[40] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022. 3, 4, 7, 9

[41] Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In *International Conference on Learning Representations*, 2022. 3, 4, 9

[42] Chengyue Gong, Dilin Wang, Meng Li, Xinlei Chen, Zhicheng Yan, Yuandong Tian, qiang liu, and Vikas Chandra. NASVit: Neural architecture search for efficient vision transformers with gradient conflict aware supernet training. In *International Conference on Learning Representations*, 2022. 3, 4, 8, 9

[43] Yehui Tang, Kai Han, Chang Xu, An Xiao, Yiping Deng, Chao Xu, and Yunhe Wang. Augmented shortcuts for vision transformers. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[44] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022. 3

[45] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 3

[46] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3008, 2021. 3, 9

[47] 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. 3, 9

[48] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019. 4, 6, 9

[49] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once for all: Train one network and specialize it for efficient deployment. In *ICLR*, 2020. 4

[50] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 4

[51] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in neural information processing systems*, volume 33, pages 1877–1901, 2020. 4

[52] Titsias RC AUEB et al. One-vs-each approximation to softmax for scalable estimation of probabilities. In *Advances in Neural Information Processing Systems*, volume 29, 2016. 5

[53] Bolin Gao and Lacra Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017. 5

[54] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 5

[55] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 5

[56] 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*, pages 12259–12269, 2021. 6, 8

[57] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. In *Advances in Neural Information Processing Systems*, volume 34, pages 30392–30400, 2021. 6

[58] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 6, 7

[59] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 6

[60] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 6

[61] Guanghua Yu, Qinyao Chang, Wenyu Lv, Chang Xu, Cheng Cui, Wei Ji, Qingqing Dang, Kaipeng Deng, Guanzhong Wang, Yuning Du, et al. Pp-picodet: A better real-time object detector on mobile devices. *arXiv preprint arXiv:2111.00902*, 2021. 7

[62] Yunyang Xiong, Hanxiao Liu, Suyog Gupta, Berkin Akin, Gabriel Bender, Yongzhe Wang, Pieter-Jan Kindermans, Mingxing Tan, Vikas Singh, and Bo Chen. Mobiledets: Searching for object detection architectures for mobile accelerators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3825–3834, 2021. 7

[63] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 7

[64] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 7

[65] Haotian Yan, Zhe Li, Weijian Li, Changhu Wang, Ming Wu, and Chuang Zhang. Contnet: Why not use convolution and transformer at the same time? *arXiv preprint arXiv:2104.13497*, 2021. 7

[66] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7

[67] Zhi Zhang, Tong He, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of freebies for training object detection neural networks. *arXiv preprint arXiv:1902.04103*, 2019. 7

[68] 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. 7

[69] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 7

[70] 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. 7

[71] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 7

[72] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 8

[73] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 8

[74] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 8

[75] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, 2018. 9

[76] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *CVPR*, 2019. 9

[77] Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct neural architecture search on target task and hardware. In *ICLR*, 2019. 9

[78] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, 2019. 9

[79] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021. 9

[80] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021. 9

[81] 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*, pages 2286–2296. PMLR, 2021. 9

[82] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 579–588, 2021. 9

[83] Zizheng Pan, Bohan Zhuang, Jing Liu, Haoyu He, and Jianfei Cai. Scalable vision transformers with hierarchical pooling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 377–386, 2021. 9

[84] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11936–11945, 2021. 9