RepViT: Revisiting Mobile CNN From ViT Perspective

Ao Wang[1,2] Hui Chen[1,2*] Zijia Lin[1] Jungong Han[3] Guiguang Ding[1,2*]
[1]Tsinghua University [2]BNRist [3]The University of Sheffield
wa22@mails.tsinghua.edu.cn huichen@mail.tsinghua.edu.cn linzijia07@tsinghua.org.cn
jungonghan77@gmail.com dinggg@tsinghua.edu.cn

2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-5300-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/CVPR52733.2024.01506

Abstract

Recently, lightweight Vision Transformers (ViTs) demonstrate superior performance and lower latency, compared with lightweight Convolutional Neural Networks (CNNs), on resource-constrained mobile devices. Researchers have discovered many structural connections between lightweight ViTs and lightweight CNNs. However, the notable architectural disparities in the block structure, macro, and micro designs between them have not been adequately examined. In this study, we revisit the efficient design of lightweight CNNs from ViT perspective and emphasize their promising prospect for mobile devices. Specifically, we incrementally enhance the mobile-friendliness of a standard lightweight CNN, i.e., *MobileNetV3, by integrating the efficient architectural designs of lightweight ViTs. This ends up with a new family of pure lightweight CNNs, namely RepViT. Extensive experiments show that RepViT outperforms existing state-of-the-art lightweight ViTs and exhibits favorable latency in various vision tasks. Notably, on ImageNet, RepViT achieves over 80% top-1 accuracy with 1.0 ms latency on an iPhone 12, which is the first time for a lightweight model, to the best of our knowledge. Besides, when RepViT meets SAM, our RepViT-SAM can achieve nearly 10× faster inference than the advanced MobileSAM. Codes and models are available at* https://github.com/THU-MIG/RepViT.

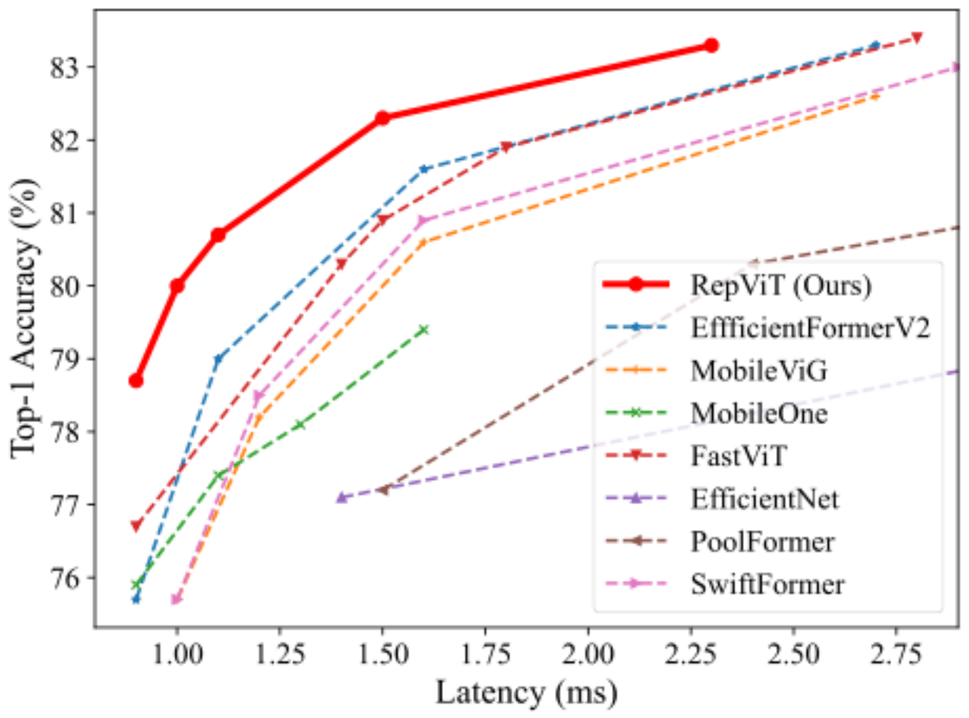

Figure 1. Comparison of latency and accuracy between RepViT (Ours) and others. The top-1 accuracy is tested on ImageNet-1K and the latency is measured by iPhone 12 with iOS 16. RepViT achieves the best trade-off between performance and latency.

1. Introduction

In the field of computer vision, designing lightweight models has been a major focus for achieving superior model performance with reduced computational costs. This is particularly important for resource-constrained mobile devices to enable the deployment of visual models at the edge. Over the past decade, researchers have primarily focused on lightweight convolutional neural networks (CNNs) and have made significant progress. Many efficient design principles have been proposed, including separable convolutions [27], inverted residual bottlenecks [53], channel shuffling [44, 75], and structural re-parameterization [13, 14], *etc*. These design principles have led to the development of representative models like MobileNets [26, 27, 53], ShuffleNets [44, 75], and RepVGG [14].

In recent years, Vision Transformers (ViTs) [18] have emerged as a promising alternative to CNNs for learning visual representations. They have demonstrated superior performance compared to CNNs on a variety of vision tasks, like image classification [39, 63], semantic segmentation [6, 66] and object detection [4, 34]. However, the trend of increasing the number of parameters in ViTs to improve performance results in large model sizes and high latency [11, 40], making them unsuitable for resource-constrained mobile devices [36, 46]. Although it is possible to directly reduce the model size of ViT models to match the constraints of mobile devices, their performance often becomes inferior to that of lightweight CNNs [5]. Therefore, researchers have embarked on exploring the lightweight de-

*Corresponding author.

2575-7075/24/$31.00 ©2024 IEEE
DOI 10.1109/CVPR52733.2024.01506

sign of ViTs, aiming to achieve performance surpassing that of lightweight CNNs.

Many efficient design principles have been proposed to enhance the computational efficiency of ViTs for mobile devices [5, 35, 46, 49]. Some approaches propose innovative architectures that combine convolutional layers with ViTs, resulting in hybrid networks [5, 46]. Additionally, novel self-attention operations with linear complexity [47] and dimension-consistent design principles [35, 36] are introduced to improve the efficiency. These studies demonstrate that lightweight ViTs [35, 47, 49] can achieve lower latency on mobile devices while outperforming lightweight CNNs [26, 53, 60], as shown in Figure 1.

Despite the success of lightweight ViTs, they continue to face practical challenges due to inadequate hardware and computational library support [60]. Additionally, ViTs are susceptible to inputs with high resolution, resulting in high latency [3]. In contrast, CNNs utilize highly optimized convolution operations with linear complexity relative to the input, making them advantageous for deployment on edge devices [54, 73]. Therefore, designing high-performance lightweight CNNs becomes imperative, compelling us to meticulously compare existing lightweight ViT and CNNs.

Lightweight ViTs and lightweight CNNs exhibit certain structural similarities. For example, both of them employ convolutional modules to learn spatially local representations [46, 47, 49, 61]. For learning global representations, lightweight CNNs usually enlarge the kernel size of convolutions [73], while lightweight ViTs generally employ the multi-head self-attention module [46, 47]. However, despite these structural connections, there remain notable differences in the block structure, macro/micro designs between them, which have yet to receive sufficient inspection. For example, lightweight ViTs usually adopt the MetaFormer block structure [69], while lightweight CNNs favors the inverted residual bottleneck [53]. This naturally leads us to a question: *Can architectural designs of lightweight ViTs enhance lightweight CNNs' performance?* To answer this question, we revisit the design of lightweight CNNs from the ViT perspective in this study. Our research aims to bridge the gap between lightweight CNNs and lightweight ViTs and highlight the promising prospect of the former for deployment on mobile devices compared to the latter.

To accomplish this objective, following [41], we begin with a standard lightweight CNN, *i.e.*, MobileNetV3-L [26]. We gradually "modernize" its architecture by incorporating the efficient architectural designs of lightweight ViTs [35, 36, 38, 46]. Finally, for resource-constrained mobile devices, we obtain a new family of lightweight CNNs, namely **RepViT**, which is composed entirely of **Re**-**p**arameterization convolutions in a **ViT**-like MetaFormer structure [36, 69, 70]. As a pure lightweight CNN, RepViT presents superior performance and efficiency compared with existing state-of-the-art lightweight ViTs [35, 49] on various computer vision tasks, including image classification on ImageNet [12], object detection and instance segmentation on COCO-2017 [37], and semantic segmentation on ADE20k [78]. Notably, RepViT reaches over 80% top-1 accuracy on ImageNet, with 1.0 ms latency on an iPhone 12, which is the first time for a lightweight model, to the best of our knowledge. Our largest model, RepViT-M2.3, obtains 83.7% accuracy with only 2.3 ms latency. After incorporating RepViT with SAM [33], our RepViT-SAM can obtain nearly 10× faster inference speed than the state-of-the-art MobileSAM [71] while enjoying significantly better zero-shot transfer capability. We hope that RepViT can serve as a strong baseline and inspire further research into lightweight models for edge deployments.

2. Related Work

In the past decade, Convolutional Neural Networks (CNNs) have emerged as the predominant approach for computer vision tasks [16, 23, 24, 43, 62, 67] due to their natural inductive locality biases and translation equivalence. However, the extensive computation of standard CNNs renders them unsuitable for deployment on resource-constrained mobile devices. To overcome this challenge, numerous techniques have been proposed to make CNNs more lightweight and mobile-friendly, including separable convolutions [27], inverted residual bottleneck [53], channel shuffle [44, 75], and structural re-parameterization [14], *etc*. These methods have paved the way for the development of several widely used lightweight CNNs, like MobileNets [26, 27, 53], ShuffleNets [44, 75], and RepVGG [14].

Subsequently, the Vision Transformer (ViT) [18] was introduced, which adapts the transformer architecture to achieve state-of-the-art performance on large-scale image recognition tasks, surpassing that of CNNs [18, 58]. Building on the competitive performance of ViTs, subsequent works have sought to incorporate spatial inductive biases to enhance their stability and performance [10, 22], design more efficient self-attention operations [17, 79], and adapt ViTs to a diverse range of computer vision tasks [19, 74].

Although ViTs have shown superior performance over CNNs on various vision tasks, most of them are heavyweighted, requiring substantial computation and memory footprint [39, 58]. That makes them unsuitable for mobile devices with limited resources [46, 49]. Consequently, researchers have dedicated to exploring various techniques to make ViTs more lightweight and more friendly for mobile devices [47, 59]. For example, MobileViT [46] adopts a hybrid architecture, combining lightweight MobileNet blocks and multi-head self-attention (MHSA) blocks. EfficientFormer [36] proposes a dimension-consistent design paradigm to enhance the latency-performance boundary. These lightweight ViTs have demonstrated new state-of-

the-art performance and latency trade-offs on mobile devices, outperforming previous lightweight CNNs [53, 60].

The success of lightweight ViTs is usually attributed to the multi-head self-attention module with the capability of learning global representations. However, the notable architectural distinctions between lightweight CNNs and lightweight ViTs, including their block structures, as well as macro and micro elements, are generally overlooked. As such, distinguished from existing works, our primary goal is to revisit the design of lightweight CNNs by integrating the architectural designs of lightweight ViTs. We aim to bridge the gap between lightweight CNNs and lightweight ViTs, and emphasize the mobile-friendliness of the former.

3. Methodology

In this section, we begin with a standard lightweight CNN, *i.e.*, MobileNetV3-L, and then gradually modernize it from various granularities, by incorporating the architectural designs of lightweight ViTs. We first introduce the metric to measure the latency on mobile devices, and then align the training recipe with existing lightweight ViTs in Section 3.1. Based on the consistent training setting, we explore the optimal block design in Section 3.2. We further optimize the performance of MobileNetV3-L on mobile devices from macro-architectural elements in Section 3.3, *i.e.*, stem, downsampling layers, classifier and overall stage ratio. We then tune the lightweight CNN through layer-wise micro designs in Section 3.4. Figure 2 shows the whole procedure and results we achieve in each step. Finally, we obtain a new family of pure lightweight CNNs designed for mobile devices in Section 3.5, namely RepViT. All models are trained and evaluated on ImageNet-1K.

3.1. Preliminary

Latency metric. Previous works [5, 57] optimize the inference speed of models based on metrics like floating point operations (FLOPs) or model sizes. However, these metrics do not correlate well with real-world latency in mobile applications [36]. Hence, following [35, 36, 46, 60], we measure the actual on-device latency as the benchmark metric. Such a strategy can provide a more accurate performance evaluation and fair comparisons among different models on real-world mobile devices. In practice, we utilize the iPhone 12 as the test device and Core ML Tools [1] as the compiler, like [35, 36, 60]. Besides, to avoid unsupported functions with Core ML Tools, we employ the GeLU activation in the MobileNetV3-L model, following [36, 60].

We measure the latency of MobileNetV3-L to be 1.01 ms.

Aligning training recipe. Recent lightweight ViTs [35, 36, 46, 49] generally adopt the training recipe from DeiT [58]. Specifically, they use the AdamW optimizer [42] and the cosine learning rate scheduler to train the models from scratch for 300 epochs, with a teacher of RegNetY-16GF [51] for distillation. Besides, they adopt Mixup [72], auto-augmentation [8], and random erasing [77] for data augmentation. Label Smoothing [56] is also employed as the regularization scheme. For fair comparisons, we align the training recipe of MobileNetV3-L with the existing lightweight ViTs, with the exception of excluding knowledge distillation for now. Consequently, MobileNetV3-L obtains 71.5% top-1 accuracy.

We will now use this training recipe by default.

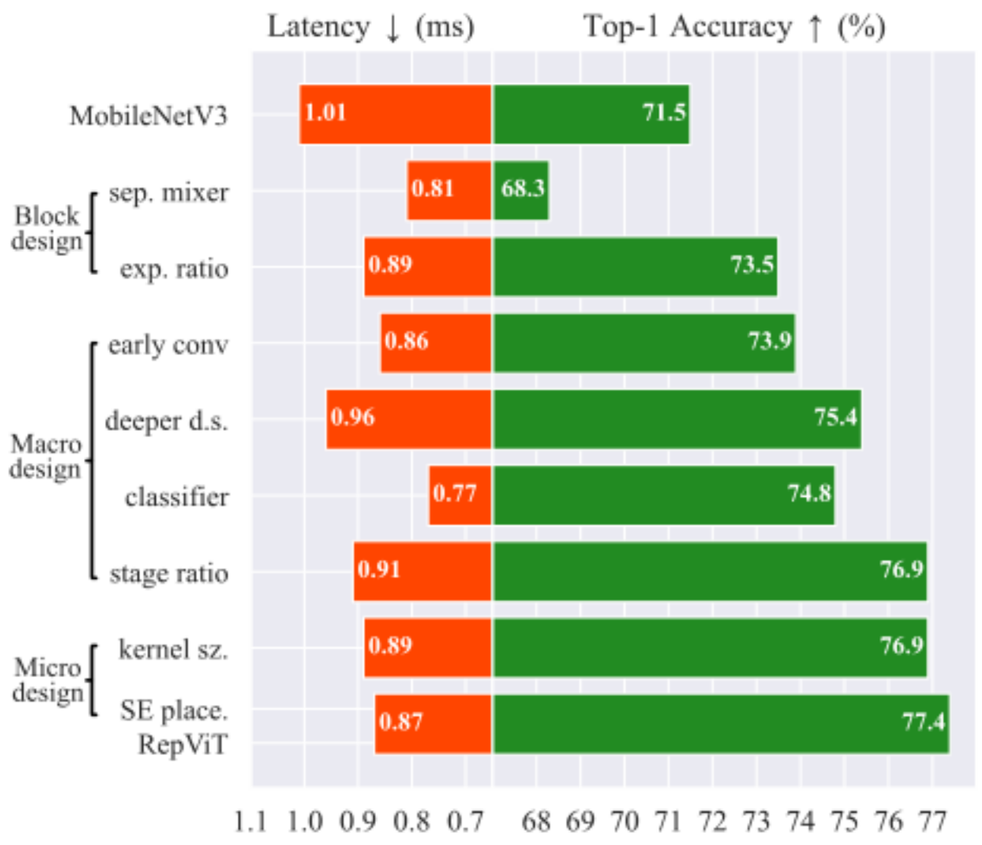

Figure 2. We modernize MobileNetV3-L from various granularities. We mainly consider the latency on mobile devices and the top-1 accuracy on ImageNet-1K. Finally, we obtain a new family of pure lightweight CNNs, namely RepViT, which can achieve lower latency and higher performance. Note that these results are obtained without the distillation.

3.2. Block design

Separate token mixer and channel mixer. The block structure of lightweight ViTs [35, 36, 47] incorporates an important design feature, namely the separate token mixer and channel mixer [70]. According to the recent research [69], the effectiveness of ViTs primarily originates from their general token mixer and channel mixer architecture, *i.e.*, the MetaFormer architecture, rather than the equipped specific token mixer. In light of this finding, we aim to emulate the existing lightweight ViTs by splitting the token mixer and channel mixer in MobileNetV3-L.

Specifically, as depicted in Figure 3.(a), the original MobileNetV3 block adopts a 1×1 expansion convolution, and a 1×1 projection layer to enable interaction among channels (*i.e.*, channel mixer). A 3×3 depthwise (DW) convolution is equipped after the 1×1 expansion convolution for the fusion of spatial information (*i.e.*, token mixer). Such a design makes the token mixer and channel mixer coupled

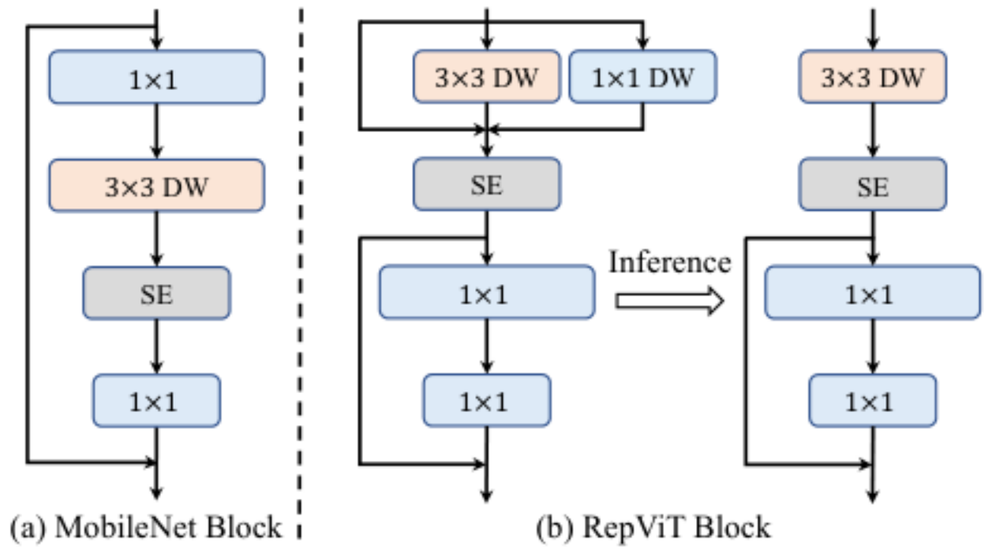

Figure 3. **Block design**. (a) is a MobileNetV3 block with an optional squeeze-and-excitation (SE) layer. (b) is the designed RepViT block, which separates the token mixer and channel mixer through the structural re-parameterization technique. The SE layer is also optional in RepViT block. The norm layer and nonlinearity are omitted for simplicity.

together. In order to separate them, we first move up the DW convolution. The optional squeeze-and-excitation (SE) layer is also moved up to be placed after the DW, as it depends on spatial information interaction. Consequently, we can successfully separate the token mixer and channel mixer in the MobileNetV3 block. We further employ a widely used structural re-parameterization technique [7, 14] for the DW layer to enhance the model learning during training. Thanks to the structural re-parameterization technique, we can eliminate the computational and memory costs associated with the skip connection during inference, which is especially advantageous for mobile devices. We name such a block as RepViT block (Figure 3.(b)), which reduces the latency of MobileNetV3-L to 0.81 ms, together with a temporary performance degradation to 68.3%.

Reducing expansion ratio and increasing width. In vanilla ViTs, the expansion ratio in the channel mixer is typically set to 4, making the hidden dimension of the Feed Forward Network (FFN) module $4\times$ wider than the input dimension. It thus consumes a significant portion of the computation resource, thereby contributing substantially to the overall inference time [76]. To alleviate this bottleneck, recent works [21, 30] employ a narrower FFN. For instance, LV-ViT [30] adopts an expansion ratio of 3 in FFN. LeViT [21] sets the expansion ratio to 2. Besides, Yang *et al*. [68] point out that there exists a significant amount of channel redundancy in FFN. Therefore, it is reasonable to use a smaller expansion ratio.

In MobileNetV3-L, the expansion ratio ranges from 2.3 to 6, with a concentration of 6 in the last two stages that have a greater number of channels. For our RepViT block, we set the expansion ratio to 2 in the channel mixer for all stages, following [21, 30, 38]. This results in a latency reduction to 0.65 ms. Consequently, with the smaller expansion ratio, we can increase the network width to remedy the large parameter reduction. We double the channels after each stage, ending up with 48, 96, 192, and 384 channels for each stage, respectively. These modifications can increase the top-1 accuracy to 73.5% with a latency of 0.89 ms.

Note that, by directly adjusting the expansion ratio and network width on the original MobileNetV3 block, we obtain inferior performance with 73.0% top-1 accuracy under a similar latency of 0.91 ms. Therefore, by default, for the block design, *we employ the new expansion ratio and network width with the RepViT block.*

3.3. Macro design

In this part, we carry out optimizations with a specific focus on its macro architecture for mobile devices, from the front to the back of the network.

Early convolutions for stem. ViTs typically use a patchify operation as the stem, dividing the input image into non-overlapping patches [18]. This simple stem corresponds to a non-overlapping convolution with a large kernel size (*e.g.*, kernel size = 16) and a large stride (*e.g.*, stride = 16). Hierarchical ViTs [39, 63] adopt the same patchify operation, but with a smaller patch size of 4. However, recent work [65] shows that such a patchify operation easily causes the substandard optimizability and sensitivity to training recipes for ViTs. To mitigate these problems, they suggest using a small number of stacked stride-two 3×3 convolutions as an alternative for the stem, known as early convolutions, which improves the optimization stability and performance, and is thus widely adopted by lightweight ViTs [35, 36].

In contrast, MobileNetV3-L adopts a complex stem that involves a 3×3 convolution, a depthwise separable convolution, and an inverted bottleneck, as shown in Figure 4.(a). Since the stem module processes the input image at the highest resolution, a complex architecture can suffer from severe latency bottlenecks on mobile devices. Therefore, as a trade-off, MobileNetV3-L reduces the initial number of filters to 16, which in turn limits the representation power of the stem. To address these issues, following [35, 36, 38, 65], we employ the way of early convolutions [65] and simply equip two 3×3 convolutions with stride = 2 as the stem. As shown in Figure 4.(b), the number of filters in the first convolution is set to 24 and the one in the second is set to 48. The overall latency is reduced to 0.86 ms. Meanwhile, the top-1 accuracy is improved to 73.9%.

We will now use early convolutions as the stem.

Deeper downsampling layers. In ViTs, spatial downsampling is typically achieved by a separate patch merging layer. As demonstrated in [41], such a separation-based downsampling layer facilitates an increase in network depth and mitigates the information loss due to the resolution reduction. Therefore, EfficientViT [38] adopts a sandwich layout to deepen the downsampling layer, achieving efficient and effective downsampling. In contrast,

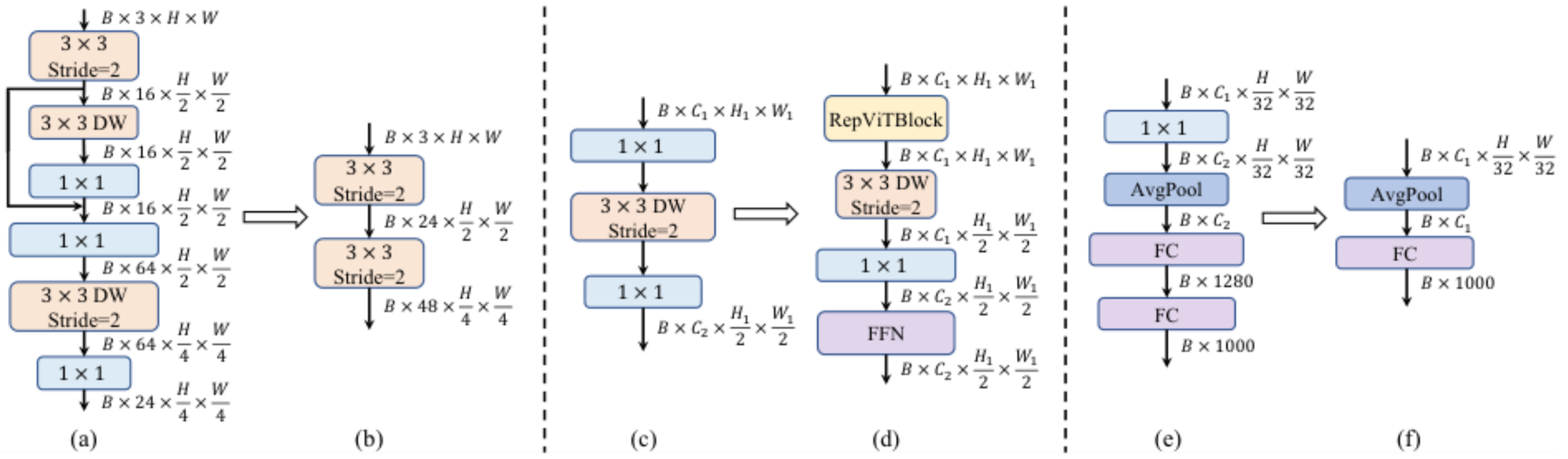

Figure 4. **Macro design**. (a) and (b), (c) and (d), (e) and (f) indicate the designs for stem, downsampling layer and classifier, respectively. RepViT has four stages with $\frac{H}{4} \times \frac{W}{4}$, $\frac{H}{8} \times \frac{W}{8}$, $\frac{H}{16} \times \frac{W}{16}$ and $\frac{H}{32} \times \frac{W}{32}$ resolutions respectively, where H and W denote the width and height of the input image. C represents the channel dimension and B denotes the batch size. The norm layer and nonlinearity are omitted.

MobileNetV3-L achieves downsampling only by an inverted bottleneck block with the DW convolution of stride = 2, as illustrated in Figure 4.(c). This design may lack adequate network depth, resulting in information loss and negative impact on the model performance. Therefore, to achieve a separate and deeper downsampling layer, we first use a DW convolution with stride = 2 and a pointwise 1×1 convolution to perform the spatial downsampling and modulate the channel dimension, respectively, as shown in Figure 4.(d). Besides, we prepend a RepViT block to further deepen the downsampling layer. A FFN module is placed after the 1×1 convolution to memorize more latent information. As a result, such deeper downsampling layers bring the top-1 accuracy to 75.4% with a latency of 0.96 ms.

We will now leverage the deeper downsampling layers.

Simple classifier. In lightweight ViTs [21, 36, 46], the classifier generally consists of a global average pooling layer followed by a linear layer. Such a simple classifier is thus friendly to the latency, especially for mobile devices. In contrast, MobileNetV3-L employs a complicated classifier, which includes one extra 1×1 convolution and one extra linear layer to expand the features to a higher-dimensional space [9], as shown in Figure 4.(e). Such a design is crucial for MobileNetV3-L to generate rich predictive features [26], particularly given the small output channel in the final stage. However, it in turn results in a heavy burden to the latency on mobile devices. Considering that the final stage now has more channels after block design in Section 3.2, we thus replace it with a simple classifier, *i.e.*, a global average pooling layer and a linear layer, as shown in Figure 4.(f). This step causes an accuracy drop of 0.6% but make the latency decrease to 0.77 ms.

We will now employ the simple classifier.

Overall stage ratio. Stage ratio represents the ratio of the number of blocks in different stages, thereby indicating the distribution of computation across the stages. Previous works [50, 51] have shown that the utilization of more blocks in the third stage confers a favorable balance between the accuracy and speed. Therefore, existing lightweight ViTs generally apply more blocks in this stage. For example, EfficientFormer-L2 [36] employs a stage ratio of 1:1:3:1.5. Meanwhile, Conv2Former [25] shows that a more aggressive stage ratio and a deeper layout perform better for small models. They thus adopt the stage ratio of 1:1:4:1 and 1:1:8:1 for Conv2Former-T and Conv2Former-S, respectively. Here, we employ a stage ratio of 1:1:7:1 for the network. We then increase the network depth to 2:2:14:2, achieving a deeper layout. This step increases the top-1 accuracy to 76.9% with a latency of 0.91 ms.

We will use this stage ratio.

3.4. Micro design

In this section, we focus on the micro architecture for lightweight CNNs, including the kernel size selection and squeeze-and-excitation (SE) layer placement.

Kernel size selection. The performance and latency of CNNs are often impacted by the size of convolution kernels. For example, to capture long-range dependencies like MHSA, ConvNeXt [41] employs large kernel-sized convolutions, exhibiting the performance gain. Similarly, RepLKNet [15] shows a powerful paradigm that utilizes super large convolution kernels in CNNs. However, large kernel-sized convolution is not friendly for mobile devices, due to its computation complexity and memory access costs. Additionally, compared to 3×3 convolutions, larger convolution kernels are typically not highly optimized by compilers and computing libraries [14]. MobileNetV3-L primarily utilizes 3×3 convolutions, with a small number of 5×5 convolutions employed in certain blocks. To ensure the inference efficiency on the mobile device, we prioritize the simple 3×3 convolutions in all modules. This replacement can maintain the top-1 accuracy at 76.9% while enjoying a latency reduction to 0.89 ms.

We will now use 3×3 *convolutions.*

Table 1. **Classification performance on ImageNet-1K.** Following [21, 38], throughput is tested on a Nvidia RTX3090 GPU with maximum power-of-two batch size that fits in memory.

Model	Type	Params (M)	GMACs	Latency ↓ (ms)	Throughput ↑ (im/s)	Epochs	Top-1 (%)
MobileViG-Ti [48]	CNN-GNN	5.2	0.7	1.0	4337	300	75.7
FastViT-T8 [59]	Hybrid	3.6	0.7	0.9	3909	300	76.7
SwiftFormer-XS [55]	Hybrid	3.5	0.6	1.0	4304	300	75.7
EfficientFormerV2-S0 [35]	Hybrid	3.5	0.4	0.9	1274	300 / 450	75.7 / 76.2
RepViT-M0.9	CONV	5.1	0.8	**0.9**	4817	300 / 450	**78.7 / 79.1**
RepViT-M1.0	CONV	6.8	1.1	**1.0**	3910	300 / 450	**80.0 / 80.3**
MobileViG-S [48]	CNN-GNN	7.2	1.0	1.2	2985	300	78.2
EfficientFormer-L1 [36]	Hybrid	12.3	1.3	1.4	3360	300	79.2
SwiftFormer-S [55]	Hybrid	6.1	1.0	1.2	3376	300	78.5
EfficientFormerV2-S1 [35]	Hybrid	6.1	0.7	1.1	1153	300 / 450	79.0 / 79.7
RepViT-M1.1	CONV	8.2	1.3	**1.1**	3604	300 / 450	**80.7 / 81.2**
MobileViG-M [48]	CNN-GNN	14.0	1.5	1.6	2491	300	80.6
FastViT-S12 [59]	Hybrid	8.8	1.8	1.5	2313	300	80.9
FastViT-SA12 [59]	Hybrid	10.9	1.9	1.8	2181	300	81.9
SwiftFormer-L1 [55]	Hybrid	12.1	1.6	1.6	2576	300	80.9
EfficientFormerV2-S2 [35]	Hybrid	12.6	1.3	1.6	611	300 / 450	81.6 / 82.0
RepViT-M1.5	CONV	14.0	2.3	**1.5**	2151	300 / 450	**82.3 / 82.5**
MobileViG-B [48]	CNN-GNN	26.7	2.8	2.7	1446	300	82.6
EfficientFormer-L3 [36]	Hybrid	31.3	3.9	2.7	1422	300	82.4
EfficientFormer-L7 [36]	Hybrid	82.1	10.2	6.6	619	300	83.3
SwiftFormer-L3 [55]	Hybrid	28.5	4.0	2.9	1474	300	83.0
EfficientFormerV2-L [35]	Hybrid	26.1	2.6	2.7	399	300 / 450	83.3 / 83.5
RepViT-M2.3	CONV	22.9	4.5	**2.3**	1184	300 / 450	**83.3 / 83.7**

Squeeze-and-excitation layer placement. One advantage of self-attention module compared with convolution is the ability to adapt weights according to input, known as the data-driven attribute [29, 64]. As a channel wise attention module, SE layers [28] can compensate for the limitation of convolutions in lacking data-driven attributes, bringing better performance [73]. MobileNetV3-L incorporates SE layers in certain blocks, with a primary focus on the latter two stages. However, as shown in [52], stages with low-resolution feature maps get a smaller accuracy benefit, compared to stages with higher resolution feature maps. Meanwhile, along with performance gains, SE layers also introduce non-negligible computational costs. Therefore, we design a strategy to utilize SE layers in a cross-block manner, *i.e.*, adopting the SE layer in the 1st, 3rd, 5th, ... block in each stage, to maximize the accuracy benefit with a minimal latency increment. This step brings the top-1 accuracy to 77.4% with a latency of 0.87 ms.

We will now use this cross-block SE layer placement. This brings our final model, namely RepViT.

3.5. Network architecture

Following [36, 46], we develop multiple RepViT variants, including RepViT-M0.9/M1.0/M1.1/M1.5/M2.3. The suffix "-MX" means that the latency of the corresponding model is X ms on the mobile device, *i.e.*, iPhone 12 with iOS 16. Variants are distinguished by the number of channels and the number of blocks within each stage. Please refer to the supplementary material for more details.

4. Experiments

4.1. Image Classification

Implementation details. We conduct image classification experiments on ImageNet-1K, using a standard image size of 224×224 for both training and testing. Following [35, 36, 48, 59], we train all models from scratch for 300 epochs or 450 epochs using the same training recipe. For fair comparisons, the RegNetY-16GF model with a top-1 accuracy of 82.9% is used as the teacher model for distillation. Following [35, 36, 60], the latency is measured on iPhone 12 with models compiled by Core ML Tools under a batch size of 1. Note that RepViT-M0.9 is the outcome of the "modernizing" process applied to MobileNetV3-L. Following [35, 59], we report the performance with and without distillation in Table 1 and Table 2, respectively.

Comparison with state-of-the-arts. As shown in Table 1, RepViT consistently achieves state-of-the-art performance across various model sizes. With similar latency, RepViT-M0.9 can significantly outperform EfficientFormerV2-S0 and FastViT-T8 by 3.0% and 2.0% top-1 accuracy, respectively. RepViT-M1.1 can also enjoy 1.7% performance improvement over EfficientFormerV2-S1. It is worth noting

Table 2. Results without distillation on ImageNet-1K.

Model	Latency (ms)	Epochs	Top-1 (%)
MobileOne-S1 [60]	0.9	300	75.9
EfficientFormerV2-S0 [35]	0.9	300	73.7
FastViT-T8 [59]	0.9	300	75.6
RepViT-M0.9	**0.9**	300	**77.4**
RepViT-M1.0	**1.0**	300	**78.6**
MobileOne-S2 [60]	1.1	300	77.4
EdgeViT-XS [49]	3.6	300	77.5
EfficientFormerV2-S1 [35]	1.1	300	77.9
RepViT-M1.1	**1.1**	300	**79.4**
MobileOne-S4 [60]	1.6	300	79.4
FastViT-S12 [59]	1.5	300	79.8
EfficientFormerV2-S2 [35]	1.6	300	80.4
RepViT-M1.5	**1.5**	300	**81.2**
EfficientNet-B3 [57]	5.3	350	81.6
PoolFormer-S36 [69]	3.5	300	81.4
RepViT-M2.3	**2.3**	300	**82.5**

that RepViT-M1.0 notably achieves over 80% top-1 accuracy with 1.0 ms latency on iPhone 12, which is the first time for a lightweight model, to the best of our knowledge. Our largest model, RepViT-M2.3, obtains 83.7% accuracy with only 2.3 ms latency. The results above well demonstrate that pure lightweight CNNs can outperform existing the state-of-the-art lightweight ViTs on mobile devices by incorporating the efficient architectural designs.

Results without knowledge distillation. As shown in Table 2, even without the enhancement of knowledge distillation, our RepViT can still significantly outperform all competitor models in different levels of latency. For example, with a latency of 1.0 ms, our RepViT-M1.0 can enjoy 2.7% accuracy gain over MobileOne-S1. For larger models, our RepViT-M2.3 can obtain 1.1% performance improvement while enjoying 34.3% latency reduction (3.5 ms to 2.3 ms), compared with PoolFormer-S36. Such results further demonstrate the effectiveness of our models.

4.2. RepViT meets SAM

Segment Anything Model (SAM) [33] has shown impressive zero-shot transfer performance for various computer vision tasks recently. However, its heavy computation costs remain daunting for resource-constrained mobile devices. Here, to show the promising performance of RepViT in segmenting anything on mobile devices, following [71], we replace the heavyweight image encoder in SAM with our RepViT model, ending up with the RepViT-SAM model. RepViT-SAM employs RepViT-M2.3 as the image encoder and is trained for 8 epochs under the same setting as [71]. Like MobileSAM [71], we use only 1% data in the SAM-1B dataset [33] for training. The project page can be found at https://jameslahm.github.io/repvit-sam/.

We first compare our RepViT-SAM with MobileSAM [71] and the original SAM [33] with ViT-B image encoder, *i.e.*, ViT-B-SAM, in terms of latency. As demonstrated in Table 3, on iPhone 12, our RepViT-SAM can perform model inference smoothly, while both competitors fail to run. On Macbook M1 Pro, RepViT-SAM is nearly 10× faster than the state-of-the-art MobileSAM.

Table 3. Comparison between RepViT-SAM and others in terms of latency. The latency (ms) is measured with the standard resolution [20] of 1024×1024 on iPhone 12 and Macbook M1 Pro by Core ML Tools. OOM means out of memory.

Platform	Image encoder			Mask decoder
	RepViT-SAM	MobileSAM [71]	ViT-B-SAM [33]	
iPhone	**48.9**	OOM	OOM	11.6
Macbook	**44.8**	482.2	6249.5	11.8

Table 4. Comparison results on zero-shot edge detection (z.s. edge.), zero-shot instance segmentation (z.s. ins.), and segmentation in the wild benchmark (SegInW). Bold indicates the best, and underline indicates the second best.

Model	z.s. edge.			z.s. ins.	SegInW
	ODS	OIS	AP	AP	Mean AP
ViT-H-SAM [33]	**.768**	**.786**	**.794**	**46.8**	**48.7**
ViT-B-SAM [33]	.743	.764	.726	42.5	44.8
MobileSAM [71]	.756	.768	.746	42.7	43.9
RepViT-SAM	<u>.764</u>	**.786**	<u>.773</u>	<u>44.4</u>	<u>46.1</u>

We then evaluate the performance of our RepViT-SAM on zero-shot edge detection using BSDS500 [2, 45], zero-shot instance segmentation using COCO [37], and segmentation in the wild benchmark (SegInW), following [31, 33]. As shown in Table 4, our RepViT-SAM outperforms MobileSAM and ViT-B-SAM on all benchmarks. Compared with ViT-H-SAM, which is the largest SAM model with over 615M parameters, our small RepViT-SAM can obtain comparable performance in terms of ODS and OIS on the zero-shot edge detection. Overall, taking all the results into consideration, our RepViT-SAM model exhibits exceptional efficiency on both the iPhone 12 and Macbook M1 Pro, while maintaining remarkable transfer performance for downstream tasks. We hope that RepViT-SAM model can serve as a strong baseline for SAM on edge deployments.

4.3. Downstream Tasks

Object Detection and Instance Segmentation. We evaluate RepViT on object detection and instance segmentation tasks to verify its transfer ability. Following [35], we integrate RepViT into the Mask-RCNN framework [24] and conduct experiments on MS COCO 2017 [37]. As seen in Table 5, RepViT consistently outperforms the competitor models in terms of latency, AP^{box} and AP^{mask}, under similar model sizes. Specifically, RepViT-M1.1 significantly outperforms EfficientFormer-L1 backbone by 1.9 AP^{box} and 1.8 AP^{mask}, with a smaller latency. For a

Table 5. **Object detection & instance segmentation** results on MS COCO 2017 with the Mask RCNN framework. **Semantic segmentation** results on ADE20K by integrating models into Semantic FPN. Backbone latencies are measured with image crops of 512×512 on iPhone 12 by Core ML Tools. * indicates that the model is initialized with weights pretrained for 450 epochs on ImageNet-1K.

Backbone	Latency ↓ (ms)	Object Detection			Instance Segmentation			Semantic
		AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	mIoU
ResNet18 [23]	4.4	34.0	54.0	36.7	31.2	51.0	32.7	32.9
PoolFormer-S12 [69]	7.5	37.3	59.0	40.1	34.6	55.8	36.9	37.2
EfficientFormer-L1 [36]	5.4	37.9	60.3	41.0	35.4	57.3	37.3	38.9
RepViT-M1.1	4.9	39.8	61.9	43.5	37.2	58.8	40.1	40.6
PoolFormer-S24 [69]	12.3	40.1	62.2	43.4	37.0	59.1	39.6	40.3
PVT-Small [63]	53.7	40.4	62.9	43.8	37.8	60.1	40.3	39.8
EfficientFormer-L3 [36]	12.4	41.4	63.9	44.7	38.1	61.0	40.4	43.5
RepViT-M1.5	6.4	41.6	63.2	45.3	38.6	60.5	41.5	43.6
EfficientFormerV2-S2* [35]	12.0	43.4	65.4	47.5	39.5	62.4	42.2	42.4
EfficientFormerV2-L* [35]	18.2	44.7	66.3	48.8	40.4	63.5	43.2	45.2
RepViT-M2.3*	9.9	44.6	66.1	48.8	40.8	63.6	43.9	46.1

larger model size, RepViT-M1.5 surpasses EfficientFormer-L3 with a nearly 2× faster speed while enjoying comparable performance. Compared with EfficientFormerV2-L, RepViT-M2.3 achieves comparable AP^{box} and higher AP^{mask} with a nearly 50% latency, highlighting the substantial advantage of lightweight CNNs in high-resolution vision tasks. The results above well demonstrate the superiority of RepViT in transferring to downstream vision tasks.

Semantic Segmentation. We conduct experiments on ADE20K [78] to verify the performance of RepViT on the semantic segmentation task. Following [35, 36], we integrate RepViT into the Semantic FPN framework [32]. As shown in Table 5, RepViT shows favorable mIoU-latency trade-offs across different model sizes. Specifically, RepViT-M1.1 significantly outperforms EfficientFormer-L1 by 1.7 mIoU with a faster speed. RepViT-M1.5 achieves a 1.2 higher mIoU over EfficientFormerV2-S2, along with a nearly 50% latency reduction. Compared with EfficientFormerV2-L, RepViT-M2.3 presents an increase of 0.9 mIoU while being nearly 2× faster. All results show the efficacy of RepViT as a general vision backbone.

4.4. Model Analyses

Structural re-parameterization (SR). To verify the effectiveness of SR in RepViT block, we conduct ablation studies on ImageNet-1K by removing the multi-branch topology of SR at training time. As shown in Table 6, without SR, different variants of the proposed RepViT suffer from consistent performance declines. The results well demonstrate the positive impact of SR.

SE layer placement. To verify the advantage of utilizing SE layers in a cross-block manner for all stages, we conduct ablation studies on ImageNet-1K by removing all SE layers (*i.e.*, "w/o SE") and adopting SE layer in each block (*i.e.*, "per block"). As presented in Table 7, alternatively adopting SE layers in blocks shows a more advantageous trade-off between accuracy and latency.

Table 6. Analyses on structural re-parameterization (SR).

SR	RepViT-M0.9	RepViT-M1.5	RepViT-M2.3
×	78.47%	82.09%	83.10%
✓	78.74%	82.29%	83.30%

Table 7. Analyses on SE layer placement.

SE	RepViT-M0.9		RepViT-M1.5	
	Top-1	Latency ↓	Top-1	Latency ↓
w/o SE	77.92%	0.83 ms	81.86%	1.48 ms
per block	78.75%	0.92 ms	82.29%	1.58 ms
ours	78.74%	0.87 ms	82.29%	1.52 ms

5. Conclusion

In this paper, we revisit the efficient design of lightweight CNNs by incorporating the architectural designs of lightweight ViTs. This ends up with RepViT, a new family of lightweight CNNs for resource-constrained mobile devices. RepViT outperforms existing state-of-the-art lightweight ViTs and CNNs on various vision tasks, showing favorable performance and latency. It highlights the promising prospect of pure lightweight CNNs for mobile devices. We hope that RepViT can serve as a strong baseline and inspire further research into lightweight models.

6. Acknowledgments

This work was supported by National Science and Technology Major Project 2022ZD0119401, Beijing Natural Science Foundation (No. L223023), and National Natural Science Foundation of China (Nos. 62271281, 61925107, 62021002).

References

[1] Core ml tools. https://github.com/apple/coremltools, 2021. 3

[2] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010. 7

[3] Arian Bakhtiarnia, Qi Zhang, and Alexandros Iosifidis. Efficient high-resolution deep learning: A survey. *arXiv preprint arXiv:2207.13050*, 2022. 2

[4] 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

[5] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 5270–5279, 2022. 1, 2, 3

[6] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 1

[7] Xiangxiang Chu, Liang Li, and Bo Zhang. Make repvgg greater again: A quantization-aware approach. *arXiv preprint arXiv:2212.01593*, 2022. 4

[8] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123, 2019. 3

[9] Cheng Cui, Tingquan Gao, Shengyu Wei, Yuning Du, Ruoyu Guo, Shuilong Dong, Bin Lu, Ying Zhou, Xueying Lv, Qiwen Liu, et al. Pp-lcnet: A lightweight cpu convolutional neural network. *arXiv preprint arXiv:2109.15099*, 2021. 5

[10] 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. 2

[11] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023. 1

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[13] Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungong Han. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1911–1920, 2019. 1

[14] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13733–13742, 2021. 1, 2, 4, 5

[15] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11963–11975, 2022. 5

[16] Zixuan Ding, Ao Wang, Hui Chen, Qiang Zhang, Pengzhang Liu, Yongjun Bao, Weipeng Yan, and Jungong Han. Exploring structured semantic prior for multi label recognition with incomplete labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3398–3407, 2023. 2

[17] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12124–12134, 2022. 2

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 4

[19] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 2

[20] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2918–2928, 2021. 7

[21] 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. 4, 5, 6

[22] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12175–12185, 2022. 2

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 8

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 7

[25] Qibin Hou, Cheng-Ze Lu, Ming-Ming Cheng, and Jiashi Feng. Conv2former: A simple transformer-style convnet for

visual recognition. *arXiv preprint arXiv:2211.11943*, 2022. 5

[26] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 1, 2, 5

[27] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 1, 2

[28] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 6

[29] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015. 6

[30] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. *Advances in neural information processing systems*, 34: 18590–18602, 2021. 4

[31] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. *arXiv preprint arXiv:2306.01567*, 2023. 7

[32] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6399–6408, 2019. 8

[33] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2, 7

[34] Y Li, CY Wu, H Fan, K Mangalam, B Xiong, J Malik, and C Feichtenhofer. Improved multiscale vision transformers for classification and detection. arxiv 2021. *arXiv preprint arXiv:2112.01526*. 1

[35] Yanyu Li, Ju Hu, Yang Wen, Georgios Evangelidis, Kamyar Salahi, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Rethinking vision transformers for mobilenet size and speed. *arXiv preprint arXiv:2212.08059*, 2022. 2, 3, 4, 6, 7, 8

[36] Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35: 12934–12949, 2022. 1, 2, 3, 4, 5, 6, 8

[37] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2, 7

[38] Xinyu Liu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan. Efficientvit: Memory efficient vision transformer with cascaded group attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14420–14430, 2023. 2, 4, 6

[39] 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, 4

[40] 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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022. 1

[41] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 2, 4, 5

[42] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[43] Mengyao Lyu, Jundong Zhou, Hui Chen, Yijie Huang, Dongdong Yu, Yaqian Li, Yandong Guo, Yuchen Guo, Liuyu Xiang, and Guiguang Ding. Box-level active detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23766–23775, 2023. 2

[44] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. 1, 2

[45] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, pages 416–423. IEEE, 2001. 7

[46] Sachin Mehta and Mohammad Rastegari. Mobilevit: lightweight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021. 1, 2, 3, 5, 6

[47] Sachin Mehta and Mohammad Rastegari. Separable self-attention for mobile vision transformers. *arXiv preprint arXiv:2206.02680*, 2022. 2, 3

[48] Mustafa Munir, William Avery, and Radu Marculescu. Mobilevig: Graph-based sparse attention for mobile vision applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2210–2218, 2023. 6

[49] Junting Pan, Adrian Bulat, Fuwen Tan, Xiatian Zhu, Lukasz Dudziak, Hongsheng Li, Georgios Tzimiropoulos, and Brais Martinez. Edgevits: Competing light-weight cnns on mobile devices with vision transformers. In *European Conference on Computer Vision*, pages 294–311. Springer, 2022. 2, 3, 7

[50] Ilija Radosavovic, Justin Johnson, Saining Xie, Wan-Yen Lo, and Piotr Dollár. On network design spaces for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1882–1890, 2019. 5

[51] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design

spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436, 2020. 3, 5

[52] Tal Ridnik, Hussam Lawen, Asaf Noy, Emanuel Ben Baruch, Gilad Sharir, and Itamar Friedman. Tresnet: High performance gpu-dedicated architecture. In *proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1400–1409, 2021. 6

[53] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 1, 2, 3

[54] Arish Sateesan, Sharad Sinha, Smitha KG, and AP Vinod. A survey of algorithmic and hardware optimization techniques for vision convolutional neural networks on fpgas. *Neural Processing Letters*, 53:2331–2377, 2021. 2

[55] Abdelrahman Shaker, Muhammad Maaz, Hanoona Rasheed, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Swiftformer: Efficient additive attention for transformer-based real-time mobile vision applications. *arXiv preprint arXiv:2303.15446*, 2023. 6

[56] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 3

[57] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 3, 7

[58] 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. 2, 3

[59] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. Fastvit: A fast hybrid vision transformer using structural reparameterization. *arXiv preprint arXiv:2303.14189*, 2023. 2, 6, 7

[60] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. Mobileone: An improved one millisecond mobile backbone. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7907–7917, 2023. 2, 3, 6, 7

[61] Shakti N Wadekar and Abhishek Chaurasia. Mobilevitv3: Mobile-friendly vision transformer with simple and effective fusion of local, global and input features. *arXiv preprint arXiv:2209.15159*, 2022. 2

[62] Ao Wang, Hui Chen, Zijia Lin, Zixuan Ding, Pengzhang Liu, Yongjun Bao, Weipeng Yan, and Guiguang Ding. Hierarchical prompt learning using clip for multi-label classification with single positive labels. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5594–5604, 2023. 2

[63] 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, 4, 8

[64] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 6

[65] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *Advances in neural information processing systems*, 34:30392–30400, 2021. 4

[66] 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*, 34:12077–12090, 2021. 1

[67] Yizhe Xiong, Hui Chen, Zijia Lin, Sicheng Zhao, and Guiguang Ding. Confidence-based visual dispersal for few-shot unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11621–11631, 2023. 2

[68] Huanrui Yang, Hongxu Yin, Pavlo Molchanov, Hai Li, and Jan Kautz. Nvit: Vision transformer compression and parameter redistribution. 2021. 4

[69] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10819–10829, 2022. 2, 3, 7, 8

[70] Weihao Yu, Chenyang Si, Pan Zhou, Mi Luo, Yichen Zhou, Jiashi Feng, Shuicheng Yan, and Xinchao Wang. Metaformer baselines for vision. *arXiv preprint arXiv:2210.13452*, 2022. 2, 3

[71] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023. 2, 7

[72] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 3

[73] Haokui Zhang, Wenze Hu, and Xiaoyu Wang. Parc-net: Position aware circular convolution with merits from convnets and transformer. In *European Conference on Computer Vision*, pages 613–630. Springer, 2022. 2, 6

[74] Wenqiang Zhang, Zilong Huang, Guozhong Luo, Tao Chen, Xinggang Wang, Wenyu Liu, Gang Yu, and Chunhua Shen. Topformer: Token pyramid transformer for mobile semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12083–12093, 2022. 2

[75] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 1, 2

[76] Chuanyang Zheng, Kai Zhang, Zhi Yang, Wenming Tan, Jun Xiao, Ye Ren, Shiliang Pu, et al. Savit: Structure-aware vision transformer pruning via collaborative optimization. *Advances in Neural Information Processing Systems*, 35:9010–9023, 2022. 4

[77] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 13001–13008, 2020. 3

[78] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 2, 8

[79] Lei Zhu, Xinjiang Wang, Zhanghan Ke, Wayne Zhang, and Rynson WH Lau. Biformer: Vision transformer with bi-level routing attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10323–10333, 2023. 2