# RapidNet: Multi-Level Dilated Convolution Based Mobile Backbone

Mustafa Munir
The University of Texas at Austin
mmunir@utexas.edu

Md Mostafijur Rahman
The University of Texas at Austin
mostafijur.rahman@utexas.edu

Radu Marculescu
The University of Texas at Austin
radum@utexas.edu

## Abstract

*Vision transformers (ViTs) have dominated computer vision in recent years. However, ViTs are computationally expensive and not well suited for mobile devices; this led to the prevalence of convolutional neural network (CNN) and ViT-based hybrid models for mobile vision applications. Recently, Vision GNN (ViG) and CNN hybrid models have also been proposed for mobile vision tasks. However, all of these methods remain slower compared to pure CNN-based models. In this work, we propose Multi-Level Dilated Convolutions to devise a purely CNN-based mobile backbone. Using Multi-Level Dilated Convolutions allows for a larger theoretical receptive field than standard convolutions. Different levels of dilation also allow for interactions between the short-range and long-range features in an image. Experiments show that our proposed model outperforms state-of-the-art (SOTA) mobile CNN, ViT, ViG, and hybrid architectures in terms of accuracy and/or speed on image classification, object detection, instance segmentation, and semantic segmentation. Our fastest model, RapidNet-Ti, achieves 76.3% top-1 accuracy on ImageNet-1K with 0.9 ms inference latency on an iPhone 13 mini NPU, which is faster and more accurate than MobileNetV2x1.4 (74.7% top-1 with 1.0 ms latency). Our work shows that pure CNN architectures can beat SOTA hybrid and ViT models in terms of accuracy and speed when designed properly*[1].

## 1. Introduction

The field of deep learning has witnessed remarkable advancements in computer vision in the last decade [48], from image classification and object detection to generative vision tasks, such as image synthesis [2, 10, 26] and video synthesis [14, 37] using generative adversarial networks (GANs) [15] and diffusion models [21]. This evolution has been fueled by diverse architectural paradigms, including Convolutional Neural Networks (CNNs) [19, 24, 25, 31, 39], Vision Transformers (ViTs) [3, 13, 38], and Multi-Layer Perceptron (MLP)-based [55, 56] models. CNNs and MLPs interpret images as pixel grids, while ViTs represent them as sequences of patches [13], enabling them to be processed by transformers [59]. ViTs also have global receptive fields and capture distant interactions within images, unlike CNNs which have local receptive fields [13].

The emergence of Vision Graph Neural Networks (ViGs), exemplified by models like ViG [16], ViHGNN [17], and MobileViG [1, 46], introduced graph-based approaches, which connect image patches through graph structures. While ViG-based models demonstrate their potential in capturing global object interactions, they incur large computational costs due to graph construction [47].

The demand for deploying powerful AI applications directly on mobile devices has led to the exploration of lightweight models [5, 24, 52, 65]. Early efforts with CNNs on mobile platforms paved the way for hybrid CNN-ViT architectures, but the computational cost of the self-attention operation in ViTs poses significant challenges for mobile applications [43, 44]. The exploration of ViTs, ViGs, and hybrid architectures for mobile devices has led to many advances in accuracy, but state-of-the-art (SOTA) results can still be achieved using only CNN-based models.

One avenue to make CNN-based models competitive is dilated convolutions [64]. Dilated convolutions can increase the receptive field of a convolution operation, but with a lower cost than increasing the kernel size. This is because a dilated convolution effectively increases the kernel's receptive field by inserting ”gaps” in between elements of the kernel by a dilation factor [64]. This means that a 3 × 3 convolution with a dilation factor of 2 will have a receptive field equal to a 5 × 5 convolution while using less parameters. Thus, we can use dilated convolutions for increasing the receptive field in our network, similar to how hybrid CNN-ViT and CNN-ViG architectures use ViTs and ViGs

[1] Code: https://github.com/mmunir127/RapidNet-Official

to increase their receptive field.

In this work, we propose Multi-Level Dilated Convolution blocks to create a CNN-based architecture competitive with SOTA CNN, ViT, ViG, and hybrid models. Our newly introduced architecture, RapidNet, is faster, less computationally expensive in terms of GMACS, and/or more accurate compared to other mobile architectures as shown in Figure 1. Indeed, our experimental results show that our proposed RapidNet architecture outperforms competing SOTA models across all model sizes for the tasks of image classification, object detection, and semantic segmentation. We summarize our contributions as follows:

1. We propose using Multi-Level Dilated Convolutions (MLDC) to enable processing features at different levels of dilation in parallel. MLDC expands the receptive field of convolutions, thus allowing for an efficient CNN-based alternative to ViG and ViT-based models.
2. We propose a novel efficient CNN-based architecture, RapidNet, which uses MLDC, reparameterizable large kernel depthwise convolutions [6, 58], and a large kernel feedforward network (FFN) [58].
3. We conduct comprehensive experiments to demonstrate the effectiveness of the RapidNet architecture, which beats the existing efficient ViG, CNN, and ViT architectures in terms of top-1 accuracy, GMACs, and/or latency on ImageNet-1k [9] image classification, COCO [36] object detection, COCO [36] instance segmentation, and ADE20K [70] semantic segmentation. Specifically, our RapidNet-M model achieves a top-1 accuracy of 81.0% on ImageNet classification, 42.0 Average Precision (AP) on COCO object detection, and 41.5 mean Intersection over Union (mIoU) on ADE20K semantic segmentation.

The paper is organized as follows. Section 2 covers related work on dilated convolutions and efficient computer vision. Section 3 describes Multi-Level Dilated Convolutions, the usage of a large kernel FFN, our RapidNet architecture, and the network configuration of RapidNet for different model sizes. Section 4 describes our experimental setup and results for ImageNet-1k image classification, COCO object detection, COCO instance segmentation, and ADE20K semantic segmentation. Lastly, Section 5 summarizes our main contributions.

## 2. Related Work

In this section, we review dilated convolutions and previous work in the mobile computer vision space.

### 2.1. Dilated Convolution

Dilated convolutions [23] introduce ”gaps” between kernel elements by a dilation factor. The ”gaps” are introduced by inserting zeros between each pixel in the convolutional kernel [64], thus allowing for an expanded receptive field without increasing the number of parameters [45, 64].

The expanded receptive field enables the model to capture a broader range of contextual information, thus facilitating more effective feature extraction. Dilated convolutions enhance the model's ability to capture long-range dependencies and improve its overall performance in tasks such as semantic segmentation [45, 64].

In Figure 2, we show how a 3 × 3 dilated convolution and 3 × 3 regular convolution differ as the dilated convolution introduces ”gaps” in between the kernel elements. In Figure 2a we can see the theoretical receptive field of the 3 × 3 convolution is 3 × 3, while in Figure 2b we see that the theoretical receptive field of the 3× 3 dilated convolution with dilation factor of 2 is 5 × 5. For a $k \times k$ dilated convolution with a dilation factor of $d$, the theoretical receptive field (TRF) of the kernel is:

$$TRF = (((k-1) \times d) + 1) \times (((k-1) \times d) + 1) \quad (1)$$

where $d - 1$ is the number of ”gaps” between pixels; thus, for a regular convolution, $d = 1$. Dilated convolutions decrease computational cost as only $k \times k$ pixels participate in the convolution even though theoretical receptive field of the convolution is increased [45].

Past work on large kernel convolutions and dilated convolutions have shown the effectiveness of increasing the receptive field of a convolutional filter [11, 39]. Large kernel convolutions are computationally expensive, thus one way to decrease the computation needed for increasing the receptive field is to use dilated convolutions. ESPNet [45] employs pointwise convolutions to reduce the computational cost of dilated convolutions and uses dilated convolutions to learn representations from the larger receptive field.

### 2.2. Mobile Vision

We can break up the past approaches in the mobile vision space into CNN-based approaches, CNN-ViT approaches, and CNN-ViG approaches.

**1. CNN-Based Approaches** In the domain of mobile vision, CNNs have historically been the mainstream architecture, with notable contributions from MobileNets [24, 52], EfficientNets [53, 54], ShuffleNets [41, 68], and SqueezeNet [27]. MobileNet [24] introduced depthwise separable convolutions, achieving comparable performance to standard convolutions with significantly lower computational cost by splitting a full convolution into a factorized version using a depthwise convolution and pointwise convolution. MobileNetv2 [52] enhanced this with the introduction of inverted residuals and linear bottlenecks. EfficientNet [53, 54] leveraged neural architecture search to produce fast and accurate models. ShuffleNet introduced pointwise group

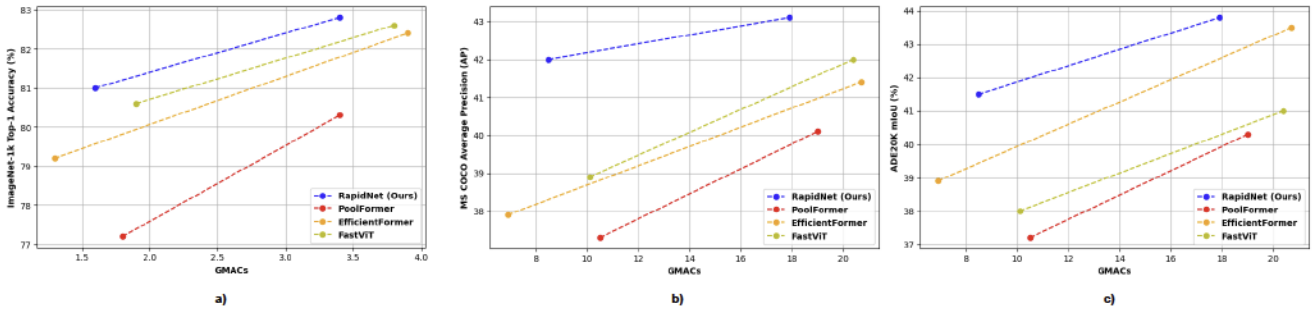

Figure 1. **Comparison of accuracy on ImageNet-1K, Average Precision (AP) on MS COCO, and mean Intersection over Union (mIoU) on ADE20K**. a) RapidNet achieves the best accuracy-GMACs tradeoff on all model sizes compared. b) RapidNet achieves the best AP-GMACs tradeoff on all model sizes compared. c) RapidNet achieves the best mIoU-GMACs tradeoff on all model sizes compared. GMACs are computed using a resolution of 224 × 224 for a) and a resolution of 512 × 512 for b) and c).

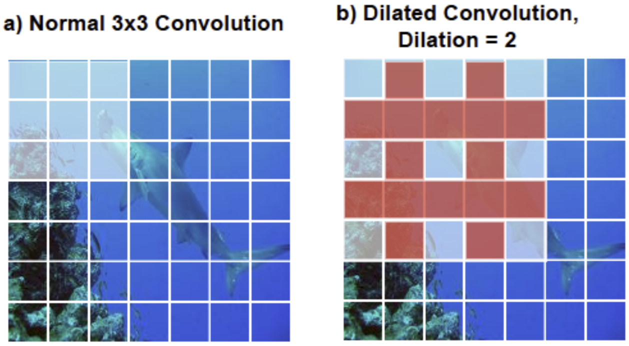
a) Normal 3x3 Convolution
b) Dilated Convolution, Dilation = 2

Figure 2. **Comparison of regular and dilated convolution.** a) In a regular convolution with a kernel size of 3 in a 7 × 7 image, we can see the convolution is applied to the 3 × 3 patches in the grid. b) In a dilated convolution with a kernel size of 3 and a dilation factor of 2 in a 7 × 7 image, we can see the convolution is applied to the 5 × 5 patches in the grid thereby expanding the receptive field. This is done by skipping the patches in red in the convolution, preserving the parameters needed for 3 × 3 convolution, but expanding the receptive field to that of a 5 × 5 convolution.

convolution and channel shuffle [68]. SqueezeNet helped push smaller networks with model compression achieving AlexNet level accuracy with 50× fewer parameters [27].

**2. CNN-ViT-Based Approaches** Recent advancements in the efficient computer vision space have led to the emergence of hybrid CNN-ViT-based models, particularly focusing on high accuracy while reducing the latency of the self-attention operation. The EfficientFormer family of models [34, 35] combine local processing using CNNs with multi-head self-attention (MHSA) operations for global processing. MobileViT and MobileViTv2 [43, 44] are also notable examples of such hybrid models that combine MobileNetv2 [52] blocks and MHSA blocks, aiming to effectively capture both local and global information.

**3. CNN-ViG-Based Approaches** Graph Neural Networks (GNNs) have traditionally been used in research on biological, social, and citation networks, but have grown in usage in the computer vision domain [16] too. Vision GNN (ViG) [16] used GNNs as a general-purpose vision backbone by splitting an image into patches and connecting the patches based on the K-Nearest Neighbors (KNN) algorithm. MobileViG [46] introduces a hybrid CNN-GNN architecture, utilizing Sparse Vision Graph Attention (SVGA) for efficient graph construction to make ViG fast on mobile devices. While MobileViG achieves high accuracy and low latency, it limits its usage of graph convolutions to the lowest resolution stage to decrease the impact on latency.

## 3. Method

In this section, we describe how we use Multi-Level Dilated Convolutions and provide details on the RapidNet architecture design. More precisely, Section 3.1 describes why we use dilated convolutions, Section 3.2 describes Multi-Level Dilated Convolutions. Section 3.3 describes our usage of a large kernel FFN. Section 3.4 describes how we combine the MLDC blocks, large kernel FFN, and inverted residual blocks for local processing to create the RapidNet architecture. Lastly, Section 3.5 describes our RapidNet network architecture for different model sizes.

### 3.1. Why Dilated Convolutions?

MobileViG [46] leverages graph convolution to perform global processing in its lowest resolution stage. However, if the cost of dilated convolution is no greater than the cost of graph convolution, then can we achieve better performance through the use of dilated convolutions? In MobileViG [46], the authors propose a static graph construction method called Sparse Vision Graph Attention (SVGA) to connect to every $K^{th}$ pixel in the row and column of the graph. Since SVGA and graph convolution are only used in the lowest resolution stage of MobileViG [46] and $K = 2$ in the MobileViG implementation, the graph convolution would have an theoretical receptive field over all

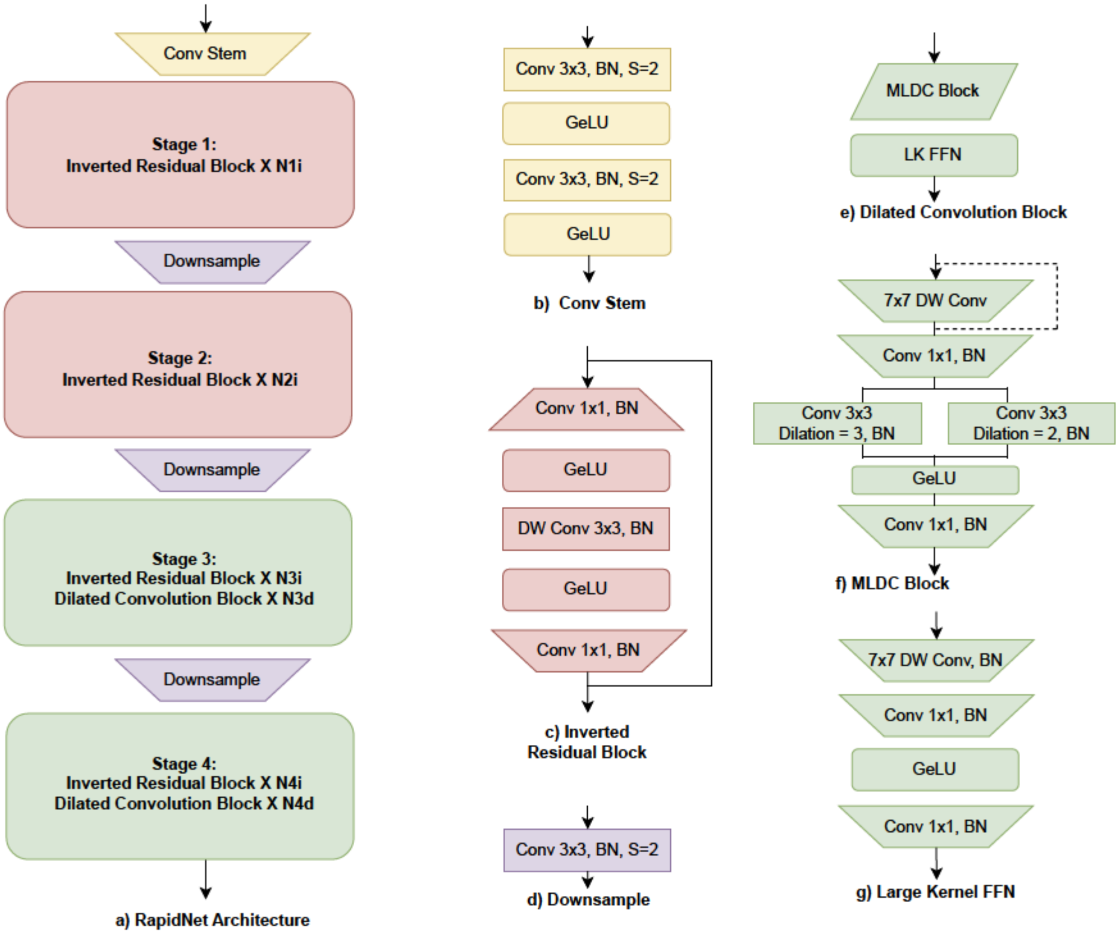

Figure 3. **RapidNet architecture.** (a) Network architecture showing the stages and layers, where N1i, N2i, N3i, N4i, N3d, and N4d represent the number of Inverted Residual Blocks and Dilated Convolution Blocks in the RapidNet-Ti, S, M, and B configurations and S represents the stride of the convolutions. (b) The Conv Stem. (c) Inverted Residual Block. (d) Downsample. (e) Dilated Convolution Block. (f) Multi-Level Dilated Convolution (MLDC) Block. (g) Large Kernel FFN.

of the patches in the image. Since the input image resolution is halved in the stem and in each downsample layer, the theoretical receptive field for SVGA in the final stage of MobileViG is:

$$TRF_{MobileViG} = 7 \times 7 \tag{2}$$

If we replace the graph convolution of MobileViG [46] with a 3 × 3 convolution and a dilation of 3 then we can achieve the same theoretical receptive field as MobileViG:

$$TRF_{Kernel=3\times3, Dilation=3} = 7 \times 7 \tag{3}$$

Using 3 × 3 convolutions has been shown to be effective in past works [12, 60] and using dilated 3 × 3 convolutions can achieve a larger theoretical receptive field, without a major hit in latency. Two dilated convolutions at different levels of dilation can be used to process features with different theoretical receptive fields in parallel. We used dilated convolutions instead of deformable convolutions due to them being better suited for our aim of computational efficiency, due to no additional learnable parameters [8]. We also provide an ablation study in the Supplementary Materials Table 6, which shows our better performance using our Multi-Level Dilated Convolution.

### 3.2. Multi-Level Dilated Convolution

We propose using two parallel dilated convolutions in place of the max-relative graph convolution [32, 46] used in MobileViG. Both of the dilated convolutions are at different levels of dilation, where dilated convolution one has a dilation factor of 2 and dilated convolution two has a dilation factor of 3. Since we use 3 × 3 convolutions instead of the pointwise convolutions of MobileViG [46], we can get a theoretical receptive field of a 5 × 5 convolution and a 7 × 7 convolution for our dilated convolutions with the parameter cost of a 3 × 3 convolution for each dilated convolution.

We note that both convolutions occur in parallel and we sum their output after batch normalization (BN) [28] and GeLU activation [20]. These parallel dilated convolutions at different dilation levels allow us to process features in different receptive fields, thus better enabling feature extraction. The usage of dilated convolutions as opposed to large kernel convolutions is to decrease the computational cost from processing via a larger receptive field.

In our RapidNet architecture, the Dilated Convolution Block consists of our MLDC Block followed by a large kernel FFN [58]. The MLDC Block shown in Figure 3f consists of a 7 × 7 depthwise convolution [6, 58] with a reparameterizable skip connection. Given an input feature $X \in \mathbb{R}^{N\times N}$, during training this can be expressed as:

$$Y = (X + DW_{7\times7}(X)) \tag{4}$$

During inference this can be expressed as:

$$Y = DW_{7\times7}(X) \tag{5}$$

where $Y \in \mathbb{R}^{N\times N}$ and $DW_{7\times7}$ is a 7 × 7 depthwise convolution. This is followed by a pointwise convolution and BN, then two dilated convolution blocks expressed as:

$$Z = \sigma(Dilated_2(Y) + Dilated_3(Y)) \tag{6}$$

where $Z \in \mathbb{R}^{N\times N}$, $Dilated_2$ and $Dilated_3$ are 3 × 3 kernel convolutions with dilation factors of 2 and 3 respectively, and $\sigma$ is a GeLU activation.

Following the MLDC Block, we use the large kernel FFN module as used in [58], which can be seen in Figure 3g. The large kernel FFN module is a 7 × 7 depthwise convolution followed by a two layer MLP expressed as:

$$Out = BN(FC_2(\sigma(FC_1(BN(DW_{7\times7}(Z)))))) \tag{7}$$

where $Out \in \mathbb{R}^{N\times N}$, $FC_1$ and $FC_2$ are fully connected layers, $\sigma$ is once again GeLU, $BN$ is batch normalization, and $DW_{7\times7}$ is again a 7 × 7 depthwise convolution. We call this combination of the MLDC Block and large kernel FFN the Dilated Convolution block, as shown in Figure 3e.

A reparameterizable 7 × 7 depthwise convolution is introduced into the MLDC block to also expand the receptive field [6]. The reparameterization follows the method of [58] to eliminate a skip connection at inference time thereby decreasing latency without damaging accuracy.

### 3.3. Large Kernel FFN

Since we do not use self-attention token mixers or the graph-based mixing of MobileViG [46], we need to employ another efficient method to expand our theoretical receptive field. For the last two stages in our architecture, we do this through our MLDC Block, but using the MLDC Block in all four stages is too computationally expensive. Thus, an efficient approach to improve the receptive field of our architecture in the first two stages is to incorporate depthwise large kernel convolutions [58] in the FFN. Following the method of [58], we incorporate depthwise 7 × 7 kernel convolutions in the FFN. The architecture of the large kernel FFN (LK FFN) is shown in Figure 3g. The LK FFN block is similar to past works [39, 46], but utilizes large kernel convolutions to enhance the receptive field and bolster model robustness as shown in [62]. Convolutional FFN blocks have been shown to exhibit greater robustness compared to standard FFN blocks [58] as shown in [42]. Thus, inspired by [58] we integrate large kernel depthwise convolutions into our FFN as an effective method for elevating model performance and robustness while minimizing the impact on latency.

### 3.4. RapidNet Architecture

The RapidNet architecture shown in Figure 3a is composed of a convolutional stem and four stages, where processing occurs at a single resolution in each stage. The stem downsamples the input image by 4× using 3 × 3 convolutions with a stride of 2 as shown in Figure 3b. The output of the stem is passed to Stage 1, which consists of $N1i$ modified inverted residual blocks (IRB) as shown in Figure 3a. After each stage is another downsample consisting of a 3 × 3 convolution with a stride of 2 to half the input resolution and expand the channel dimension as shown in Figure 3d. Stage 2 consists of $N2i$ IRB blocks and has different channel dimensions from Stage 1. Stages 3 and 4 start with a sequence of $N3i$ and $N4i$ IRB blocks followed by $N3d$ and $N4d$ Dilated Convolution Blocks.

The IRB block is used for local processing at each stage and uses an expansion ratio of four following the method of MobileNetv2 [52]. Each IRB block consists of a 1 × 1 convolution, BN, GeLU, a depth-wise 3 × 3 convolution, BN, GeLU, and lastly a 1 × 1 convolution plus BN and a residual connection as shown in Figure 3c. Within the IRB blocks, we replace ReLU for GeLU following [35, 46], which show GeLU improves performance in vision tasks. The MLDC block is used for processing at a larger theoretical receptive field to better learn global object interactions.

The Dilated Convolution Block consists of the MLDC Block and the large kernel FFN shown in Figure 3e, 3f, and 3g. The MLDC Block consists of a reparameterizable 7 × 7 depthwise convolution [6] as in [58], followed by a pointwise convolution. Then, two parallel Multi-Level Dilated Convolutions with dilation factors of 2 and 3 respectively followed by another a pointwise convolution and BN. The large kernel FFN consists of a 7 × 7 depthwise convolution and BN followed by an FFN consisting of two pointwise convolutions and BN with GeLU activation in between.

Table 1. **Architecture details of RapidNet** showing configuration of the stem, stages, output size, downsample layers, and classification head. *Channels* represents the channel width. *IRB* represents the Inverted Residual Block. *DCB* represents the Dilated Convolution Block. Lin. MLP stands for linear MLP.

| Stage | Output Size | RapidNet-Ti | RapidNet-S | RapidNet-M | RapidNet-B |
|---|---|---|---|---|---|
| Stem | $\frac{H}{4} \times \frac{W}{4}$ | Conv ×2, Stride=2 | Conv ×2, Stride=2 | Conv ×2, Stride=2 | Conv ×2, Stride=2 |
| Stage 1 | $\frac{H}{4} \times \frac{W}{4}$ | $IRB \times 2$<br>$Channels = 32$ | $IRB \times 3$<br>$Channels = 32$ | $IRB \times 3$<br>$Channels = 32$ | $IRB \times 3$<br>$Channels = 64$ |
| Downsample | $\frac{H}{8} \times \frac{W}{8}$ | Conv, Stride=2 | Conv, Stride=2 | Conv, Stride=2 | Conv, Stride=2 |
| Stage 2 | $\frac{H}{8} \times \frac{W}{8}$ | $IRB \times 2$<br>$Channels = 64$ | $IRB \times 3$<br>$Channels = 64$ | $IRB \times 3$<br>$Channels = 64$ | $IRB \times 3$<br>$Channels = 128$ |
| Downsample | $\frac{H}{16} \times \frac{W}{16}$ | Conv, Stride=2 | Conv, Stride=2 | Conv, Stride=2 | Conv, Stride=2 |
| Stage 3 | $\frac{H}{16} \times \frac{W}{16}$ | $IRB \times 6$<br>$DCB \times 2$<br>$Channels = 112$ | $IRB \times 9$<br>$DCB \times 3$<br>$Channels = 112$ | $IRB \times 9$<br>$DCB \times 3$<br>$Channels = 160$ | $IRB \times 9$<br>$DCB \times 3$<br>$Channels = 224$ |
| Downsample | $\frac{H}{32} \times \frac{W}{32}$ | Conv, Stride=2 | Conv, Stride=2 | Conv, Stride=2 | Conv, Stride=2 |
| Stage 4 | $\frac{H}{32} \times \frac{W}{32}$ | $IRB \times 2$<br>$DCB \times 2$<br>$Channels = 224$ | $IRB \times 3$<br>$DCB \times 3$<br>$Channels = 224$ | $IRB \times 3$<br>$DCB \times 3$<br>$Channels = 320$ | $IRB \times 3$<br>$DCB \times 3$<br>$Channels = 416$ |
| Head | 1 × 1 | Pooling & Lin. MLP | Pooling & Lin. MLP | Pooling & Lin. MLP | Pooling & Lin. MLP |

### 3.5. Network Configurations

The detailed network architectures for RapidNet-Ti, S, M, and B are provided in Table 1. We report the output size of each stage as well as the configuration of the stem, stages, and classification head. In each stage, the number of IRB and Dilated Convolution Blocks (DCB) repeated, as well as their channel dimensions are reported. As we scale up our architecture from RapidNet-Ti to RapidNet-B, we increase the number of IRB and DCB blocks (network depth) as well as the channel dimensions (network width).

## 4. Experimental Results

In this section, we describe our experimental setup and perform a thorough comparison between RapidNet and other mobile vision architectures. Our evaluations show that for similar or fewer parameters, GMACs, and/or latency, RapidNet has a superior performance in terms of top-1 accuracy on ImageNet-1k [9] image classification, average precision (AP) on COCO [36] object detection and instance segmentation, and mean intersection over union (mIoU) on ADE20K [70] semantic segmentation.

### 4.1. Image Classification

We conduct image classification experiments on the widely used ImageNet-1K [9] dataset. The dataset contains training and validation sets of approximately 1.3M images and 50K images, respectively. We train from scratch for 300 epochs with a standard resolution of 224 × 224. We implement our RapidNet model using PyTorch 1.12.1 [50] and Timm library [63]. Like other mobile architectures [34, 46, 60], we use RegNetY-16GF [51] with a top-1 accuracy of 82.9% as the teacher model for knowledge distillation. Our data augmentation pipeline includes RandAugment [7], Mixup [67], Cutmix [66], random erasing [69], and repeated augment [22]. We use the AdamW [40] optimizer and a learning rate of $2e^{-3}$ with a cosine annealing schedule. To measure inference latency, all models are packaged as MLModels using CoreML and profiled on an iPhone 13 Mini (iOS 16) using ModelBench [57]. We use the following ModelBench settings to profile each model: 50 inference rounds, 50 inferences per round, and a low/high trim of 10. Table 2 shows ImageNet-1K classification results for RapidNet and other mobile architectures.

RapidNet-Ti achieves a top-1 accuracy of 76.3%, which is higher than MobileViG-Ti [46] by 0.6% while achieving the same inference latency of 0.9 ms with 0.1 less GMACs. RapidNet-S also outperforms MobileNetV2x1.4 by 3.9% in terms of top-1 accuracy with the same inference latency of 1.1 ms. RapidNet-M and RapidNet-B similarly, outperform competing models for similar inference latency and/or GMACs. The success of RapidNet shows the usefulness of

Table 2. Results of RapidNet and other mobile architectures on ImageNet-1K classification task grouped by NPU latency of an iPhone13 Mini using the ModelBench application. *Type* indicates whether the model is CNN-based, CNN-ViT-based, CNN-Pooling-based, or CNN-GNN-based. *Params* lists the number of model parameters in millions. *GMACs* lists the number of MACs in billions. Shaded entries indicate results obtained using RapidNet. A (-) denotes that the model could not be profiled on the iPhone 13 Mini. * indicates methods that use knowledge distillation from RegNetY-16GF [51]

| Model | Type | Resolution | Params (M) | GMACs | NPU Latency (ms) | Top-1 (%) |
|---|---|---|---|---|---|---|
| MobileNetV2x1.0 [52] | CNN | $224^2$ | 3.5 | 0.3 | 0.8 | 71.8 |
| EdgeViT-XXS [49] | CNN-ViT | $224^2$ | 4.1 | 0.6 | - | 74.4 |
| MobileViG-Ti* [46] | CNN-GNN | $224^2$ | 5.2 | 0.7 | 0.9 | 75.7 |
| **RapidNet-Ti*** | **CNN** | $224^2$ | **6.6** | **0.6** | **0.9** | **76.3** |
| MobileNetV2x1.4 [52] | CNN | $224^2$ | 6.1 | 0.6 | 1.1 | 74.7 |
| EdgeViT-XS [49] | CNN-ViT | $224^2$ | 6.7 | 1.1 | - | 77.5 |
| PoolFormer-S12 [65] | CNN-Pooling | $224^2$ | 12.0 | 1.8 | 1.5 | 77.2 |
| MobileViG-S* [46] | CNN-GNN | $224^2$ | 7.2 | 1.0 | 1.1 | 78.2 |
| **RapidNet-S*** | **CNN** | $224^2$ | **9.2** | **0.9** | **1.1** | **78.6** |
| EfficientNet-B0 [53] | CNN | $224^2$ | 5.3 | 0.4 | 1.5 | 77.7 |
| EfficientFormer-L1* [35] | CNN-ViT | $224^2$ | 12.3 | 1.3 | 1.3 | 79.2 |
| PoolFormer-S24 [65] | CNN-Pooling | $224^2$ | 21.0 | 3.4 | 2.4 | 80.3 |
| MobileViG-M* [46] | CNN-GNN | $224^2$ | 14.0 | 1.5 | 1.5 | 80.6 |
| **RapidNet-M*** | **CNN** | $224^2$ | **17.3** | **1.6** | **1.6** | **81.0** |
| EfficientNet-B3 [53] | CNN | $224^2$ | 12.2 | 2.0 | 4.8 | 81.6 |
| MobileViTv2-1.0 [44] | CNN-ViT | $224^2$ | 4.9 | 1.8 | 3.0 | 78.1 |
| MobileViTv2-2.0 [44] | CNN-ViT | $224^2$ | 18.5 | 7.5 | 6.3 | 82.4 |
| EfficientFormer-L3* [35] | CNN-ViT | $224^2$ | 31.3 | 3.9 | 2.6 | 82.4 |
| EfficientFormer-L7* [35] | CNN-ViT | $224^2$ | 82.1 | 10.2 | 6.5 | 83.3 |
| PoolFormer-S36 [65] | CNN-Pooling | $224^2$ | 31.0 | 5.0 | 3.3 | 81.4 |
| PoolFormer-M36 [65] | CNN-Pooling | $224^2$ | 56.0 | 8.8 | 5.7 | 82.1 |
| **RapidNet-B*** | **CNN** | $224^2$ | **30.5** | **3.4** | **2.7** | **82.8** |

Multi-Level Dilated Convolutions in mobile vision architectures as they can enable theoretical receptive field expansion with lower costs than ViGs and ViTs creating high accuracy and low latency CNN-based models.

## 4.2. Object Detection and Instance Segmentation

We evaluate RapidNet on MS COCO object detection and instance segmentation tasks to verify it generalizes to downstream tasks. Following [34,35,46,60], we use RapidNet as the backbone in the Mask-RCNN framework [18] to conduct experiments on MS COCO 2017 [36]. The dataset contains training and validations sets of 118K and 5K images, respectively. We implement the backbone using PyTorch 1.12.1 [50] and Timm library [63]. The model is initialized with ImageNet-1k pretrained weights from 300 epochs of training. We use AdamW [29,40] optimizer with an initial learning rate of 2e$^{-4}$ and train the model for 12 epochs on 8 NVIDIA RTX 6000 Ada generation GPUs with a 1333 × 800 resolution following prior work [33–35].

As seen in Table 3, with similar model size, RapidNet outperforms ResNet [19], PoolFormer [65], EfficientFormer [35], MobileViG [46], Swin Transformer [38], and PVT [61] in terms of either parameters or improved average precision (AP) on object detection and instance segmentation. Our RapidNet-M model gets 42.0 $AP^{box}$ and 38.3 $AP^{mask}$ on the object detection and instance segmentation tasks outperforming PoolFormer-s12 [65] by 4.7 $AP^{box}$ and 3.7 $AP^{mask}$ and FastViT-SA12 [58] by 3.1 $AP^{box}$ and 2.4 $AP^{mask}$. Our RapidNet-B model achieves 43.1 $AP^{box}$ and 39.3 $AP^{mask}$ outperforming EfficientFormer-L3 [35] by 1.7 $AP^{box}$ and 1.2 $AP^{mask}$ and FastViT-SA24 [58] by 1.1 $AP^{box}$ and 1.3 $AP^{mask}$. The strong performance of RapidNet on object detection and instance segmentation shows the capability of Multi-Level Dilated Convolutions to help RapidNet generalize well to different vision tasks.

Table 3. Results of RapidNet and other backbones on COCO object detection, COCO instance segmentation, and ADE20K semantic segmentation grouped by parameters of the backbone. $AP^{box}$ and $AP^{mask}$ scores are for object detection and instance segmentation on MS COCO 2017 [36]. $mIoU$ scores are for semantic segmentation on ADE20K [70]. Shaded entries indicate results obtained using RapidNet. A (-) denotes a model that did not report these results.

| **Backbone** | **Params (M)** | $AP^{box}$ | $AP^{box}_{50}$ | $AP^{box}_{75}$ | $AP^{mask}$ | $AP^{mask}_{50}$ | $AP^{mask}_{75}$ | $mIoU$ |
|---|---|---|---|---|---|---|---|---|
| ResNet18 [19] | 11.7 | 34.0 | 54.0 | 36.7 | 31.2 | 51.0 | 32.7 | 32.9 |
| FastViT-SA12 [58] | 10.9 | 38.9 | 60.5 | 42.2 | 35.9 | 57.6 | 38.1 | 38.0 |
| MobileViG-M [46] | 14.0 | 41.3 | 62.8 | 45.1 | 38.1 | 60.1 | 40.8 | - |
| EfficientFormer-L1 [35] | 12.3 | 37.9 | 60.3 | 41.0 | 35.4 | 57.3 | 37.3 | 38.9 |
| PoolFormer-S12 [65] | 12.0 | 37.3 | 59.0 | 40.1 | 34.6 | 55.8 | 36.9 | 37.2 |
| RapidNet-M | **17.3** | **42.0** | **63.0** | **46.1** | **38.3** | **60.3** | **41.1** | **41.5** |
| ResNet50 [19] | 25.5 | 38.0 | 58.6 | 41.4 | 34.4 | 55.1 | 36.7 | 36.7 |
| Swin-T [38] | 29.0 | 42.2 | 64.4 | 46.2 | 39.1 | 61.6 | 42.0 | 41.5 |
| PVT-Small [61] | 24.5 | 40.4 | 62.9 | 43.8 | 37.8 | 60.1 | 40.3 | 39.8 |
| FastViT-SA24 [58] | 20.6 | 42.0 | 63.5 | 45.8 | 38.0 | 60.5 | 40.5 | 41.0 |
| EfficientFormer-L3 [35] | 31.3 | 41.4 | 63.9 | 44.7 | 38.1 | 61.0 | 40.4 | 43.5 |
| EfficientFormer-L7 [35] | 82.1 | 42.6 | 65.1 | 46.1 | 39.0 | 62.2 | 41.7 | 45.1 |
| PoolFormer-S24 [65] | 21.0 | 40.1 | 62.2 | 43.4 | 37.0 | 59.1 | 39.6 | 40.3 |
| RapidNet-B | **30.5** | **43.1** | **64.6** | **47.2** | **39.3** | **61.5** | **42.2** | **43.8** |

### 4.3. Semantic Segmentation

To verify the performance of RapidNet on the semantic segmentation task, we conduct experiments on the scene parsing dataset, ADE20k [70]. The dataset contains 20K training images and 2K validation images with 150 semantic categories. Following prior work [34, 35, 58, 60, 65], we integrate RapidNet as the backbone in the Semantic FPN [30] framework. The backbone is initialized with pretrained weights on ImageNet-1K and the model is trained for 40K iterations on 8 NVIDIA RTX 6000 Ada generation GPUs. We follow the process of existing works in segmentation, using AdamW [29, 40] optimizer, set the learning rate as 2 $\times$ $10^{-4}$ with a poly decay by the power of 0.9, and image resolution of 512 $\times$ 512.

As shown in Table 3, RapidNet-M outperforms PoolFormer-S12 [65], FastViT-SA12 [58], and EfficientFormer-L1 [35] by 4.3, 3.5, and 2.6 mIoU. Additionally, RapidNet-B outperforms PoolFormer-S24 [65], FastViT-SA24 [58], and PVT-Small [61] by 3.5, 2.8, and 4.0 mIoU. Through these results we show that with MLDC, RapidNet is better able to learn long-range object interactions compared to other mobile vision architectures.

### 4.4. Ablation Studies

Ablation studies on SVGA, pointwise convolution, and 3 $\times$ 3 convolution are included in Section A.1 of the supplementary material. Ablation studies on CPE, LK FFN, single-level dilated convolution, and MLDC are included in Section A.2 of the supplementary material. Ablation studies on dilation factors and kernel sizes are included in Section A.3 of the supplementary material.

## 5. Conclusion

We have proposed Multi-Level Dilated Convolutions (MLDC) as a method to design mobile CNN models with larger theoretical receptive fields while maintaining low latency. MLDC is able to process features at multiple dilation levels, in parallel, thereby allowing for processing with a larger and smaller theoretical receptive field, then combining that information to enhance feature extraction. Additionally, we have also proposed a novel CNN-based architecture, RapidNet, which uses a combination of MLDC, inverted residual blocks, a large kernel feedforward network, and reparameterizable large kernel depthwise convolutions.

RapidNet outperforms existing CNN, ViG, ViT, and hybrid models on image classification, object detection, instance segmentation, and semantic segmentation. RapidNet performs particularly well on the downstream tasks of object detection, instance segmentation, and semantic segmentation due to the ability of dilated convolutions to better learn long-range object interactions compared to standard convolutions. The effectiveness of RapidNet shows the ability of CNN-based networks to compete with state-of-the-art ViT and hybrid CNN-ViT models.

## 6. Acknowledgements

This work is supported in part by the NSF grant CNS 2007284, and in part by the iMAGiNE Consortium (https://imagine.utexas.edu/).

## References

[1] William Avery, Mustafa Munir, and Radu Marculescu. Scaling graph convolutions for mobile vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 5857–5865, June 2024. 1

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 1

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, pages 213–229. Springer, 2020. 1

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 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

[6] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021. 2, 5

[7] 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. 6

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 4, 1

[9] 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, 2009. 2, 6, 1

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 1

[11] 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. 2

[12] 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. 4

[13] Alexey Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[14] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7346–7356, 2023. 1

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014. 1

[16] Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision gnn: An image is worth graph of nodes. *arXiv preprint arXiv:2206.00272*, 2022. 1, 3

[17] Yan Han, Peihao Wang, Souvik Kundu, Ying Ding, and Zhangyang Wang. Vision hgnn: An image is more than a graph of nodes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19878–19888, 2023. 1

[18] 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. 7

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 7, 8

[20] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 5

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1

[22] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8129–8138, 2020. 6

[23] Matthias Holschneider, Richard Kronland-Martinet, Jean Morlet, and Ph Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets: Time-Frequency Methods and Phase Space Proceedings of the International Conference, Marseille, France, December 14–18, 1987*, pages 286–297. Springer, 1990. 2

[24] 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

[25] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017. 1

[26] He Huang, Philip S Yu, and Changhu Wang. An introduction to image synthesis with generative adversarial nets. *arXiv preprint arXiv:1803.04469*, 2018. 1

[27] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. *arXiv:1602.07360*, 2016. 2, 3

[28] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. pmlr, 2015. 5

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7, 8

[30] 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

[31] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[32] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9267–9276, 2019. 4

[33] Jiashi Li, Xin Xia, Wei Li, Huixia Li, Xing Wang, Xuefeng Xiao, Rui Wang, Min Zheng, and Xin Pan. Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios. *arXiv preprint arXiv:2207.05501*, 2022. 7

[34] 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. 3, 6, 7, 8

[35] Yanyu Li, Geng Yuan, Yang Wen, Eric Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *arXiv preprint arXiv:2206.01191*, 2022. 3, 5, 7, 8

[36] 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 *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014. 2, 6, 7, 8

[37] Ming-Yu Liu, Xun Huang, Jiahui Yu, Ting-Chun Wang, and Arun Mallya. Generative adversarial networks for image and video synthesis: Algorithms and applications. *Proceedings of the IEEE*, 109(5):839–862, 2021. 1

[38] 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, 7, 8

[39] 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. 1, 2, 5

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6, 7, 8

[41] 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. 2

[42] Xiaofeng Mao, Gege Qi, Yuefeng Chen, Xiaodan Li, Ranjie Duan, Shaokai Ye, Yuan He, and Hui Xue. Towards robust vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12042–12051, 2022. 5

[43] Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021. 1, 3

[44] Sachin Mehta and Mohammad Rastegari. Separable self-attention for mobile vision transformers. *arXiv preprint arXiv:2206.02680*, 2022. 1, 3, 7

[45] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 552–568, 2018. 2

[46] 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 Workshops*, pages 2211–2219, 2023. 1, 3, 4, 5, 6, 7, 8

[47] Mustafa Munir, William Avery, Md Mostafijur Rahman, and Radu Marculescu. Greedyvig: Dynamic axial graph construction for efficient vision gnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6118–6127, June 2024. 1

[48] Mustafa Munir, Saloni Modi, Geffen Cooper, HunTae Kim, and Radu Marculescu. Three decades of low power: From watts to wisdom. *IEEE Access*, 2024. 1

[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 *Proceedings of the European Conference on Computer Vision*, pages 294–311. Springer, 2022. 7

[50] Adam Paszke et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019. 6, 7

[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. 6, 7

[52] 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, 5, 7

[53] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 2, 7

[54] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021. 2

[55] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung,

Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34:24261–24272, 2021. 1

[56] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1

[57] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. An improved one millisecond mobile backbone. *arXiv preprint arXiv:2206.04040*, 2022. 6

[58] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. Fastvit: A fast hybrid vision transformer using structural reparameterization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 2, 5, 7, 8

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 1

[60] Ao Wang, Hui Chen, Zijia Lin, Hengjun Pu, and Guiguang Ding. Repvit: Revisiting mobile cnn from vit perspective. *arXiv preprint arXiv:2307.09283*, 2023. 4, 6, 7, 8

[61] 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. 7, 8

[62] Zeyu Wang, Yutong Bai, Yuyin Zhou, and Cihang Xie. Can cnns be more robust than transformers? *arXiv preprint arXiv:2206.03452*, 2022. 5

[63] Ross Wightman. PyTorch Image Models. https://github.com/rwightman/pytorch-image-models, 2019. 6, 7

[64] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 1, 2

[65] 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. 1, 7, 8

[66] 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. 6

[67] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 6

[68] 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. 2, 3

[69] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020. 6

[70] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 633–641, 2017. 2, 6, 8