EfficientViT: Lightweight Multi-Scale ttention for On-Device Semantic Segmentation

Han Cai , Junyan Li[2], Muyan Hu[3], Chuang Gan[4], Song Han
MIT, [2]Zhejiang University, [3]Tsinghua University, [4]MIT-IBM Watson I Lab
https://github.com/mit-h n-l b/efficientvit

bstract

Semantic segmentation enables many appealing real-world applications, such as computational photography, autonomous driving, etc. However, the vast computational cost makes deploying state-of-the-art semantic segmentation models on edge devices with limited hardware resources difficult. This work presents EfficientViT, a new family of semantic segmentation models with a novel lightweight multi-scale attention for on-device semantic segmentation. Unlike prior semantic segmentation models that rely on heavy self-attention, hardware-inefficient large-kernel convolution, or complicated topology structure to obtain good performances, our lightweight multi-scale attention achieves a global receptive field and multi-scale learning (two critical features for semantic segmentation models) with only lightweight and hardware-efficient operations. s such, EfficientViT delivers remarkable performance gains over previous state-of-the-art semantic segmentation models across popular benchmark datasets with significant speedup on the mobile platform. Without performance loss on Cityscapes, our EfficientViT provides up to 15× and 9.3× mobile latency reduction over SegFormer and SegNeXt, respectively. Maintaining the same mobile latency, EfficientViT provides +7.4 mIoU gain on DE20K over SegNeXt.

1. Introduction

Semantic segmentation is a fundamental task in computer vision, which aims to assign a class label to each pixel in the input image. Semantic segmentation has broad applications in real-world scenarios, including autonomous driving, medical image processing, computational photography, etc. Therefore, deploying state-of-the-art (SOT) semantic segmentation models on edge devices is in great demand to benefit a wide range of users.

However, there is a large gap between the computational cost required by SOT semantic segmentation models and the limited resources of edge devices. It makes deploying these models on edge devices impractical. In particular, semantic segmentation is a dense prediction task requiring high-resolution images and strong context information extraction ability to work well [1, 36, 47, 52, 48, 42]. Therefore, directly porting efficient model architecture from image classification is unsuitable for semantic segmentation.

This work introduces **EfficientViT**, a new family of models for on-device semantic segmentation. The core of EfficientViT is a novel lightweight multi-scale attention module that enables a global receptive field and multi-scale learning with hardware-efficient operations. Our module is motivated by prior SOT semantic segmentation models. They demonstrate that the multi-scale learning [47, 52], and global receptive field [45] play a critical role in improving the performances for semantic segmentation. However, they do not consider hardware efficiency when designing their models, which is essential for on-device semantic segmentation. For example, SegFormer [45] introduces self-attention into the backbone to have a global receptive field. But its computational complexity is quadratic to the input resolution, making it unable to handle high-resolution images efficiently. SegNeXt [17] proposes a multi-branch module with large-kernel convolutions (kernel size up to 21) to enable a large receptive field and multi-scale learning. However, large-kernel convolution requires exceptional support on hardware to achieve good efficiency [15], which is usually not available on edge devices.

Hence, the design principle of our module is to enable these two critical features while avoiding hardware-inefficient operations. Specifically, to have a global receptive field, we propose substituting the inefficient self-attention with lightweight ReLU-based global attention [26]. By leveraging the associative property of matrix multiplication, ReLU-based global attention can reduce the computational complexity from quadratic to linear while preserving functionality. In addition, it avoids hardware-inefficient operations like softmax, making it more suitable for on-device semantic segmentation (Figure 3).

Furthermore, we propose a novel lightweight multi-scale

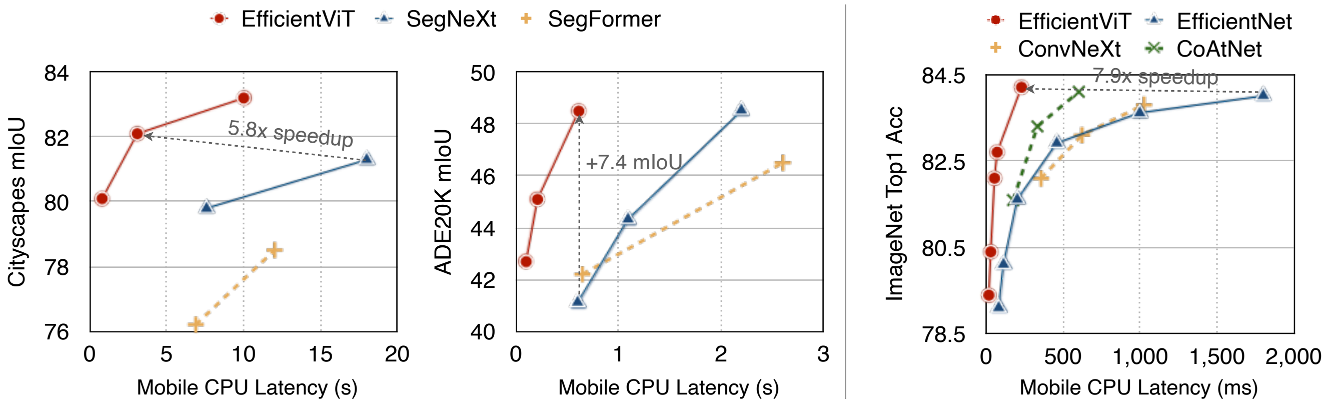

Figure 1: **Latency vs. Performance.** ll performance results are obtained with the single model and single-scale inference. The latency results are obtained on the Qualcomm Snapdragon 8Gen1 CPU using Tensorflow-Lite. Compared with state-of-the-art (SOT) segmentation models, EfficientViT achieves a remarkable boost in speed while providing the same or higher performances on Cityscapes and DE20K. In addition, EfficientViT also shows strong performances in image classification, achieving a 7.9x latency reduction over EfficientNet without accuracy loss on ImageNet.

Table 1: **Desirable Features for On-device Semantic Segmentation.** 'Linear computational complexity' means the computational cost grows linearly as the input resolution increases.

Features	SegFormer [45]	HRFormer [49]	SegNeXt [17]	**EfficientViT**
Global receptive field				
Multi-scale learning				
Linear computational complexity				
Hardware efficiency				

attention module based on the ReLU-based global attention. Specifically, we aggregate nearby tokens with small-kernel convolutions to generate multi-scale tokens and perform ReLU-based global attention on multi-scale tokens (Figure 2) to combine the global receptive field with multi-scale learning. We summarize the comparison between our work and prior SOT semantic segmentation models in Table 1. We can see that our model is more suitable for on-device semantic segmentation than previous models.

We extensively evaluate EfficientViT on popular semantic segmentation benchmark datasets, including Cityscapes [12] and DE20K [53]. EfficientViT provides significant performance boosts over prior SOT semantic segmentation models. More importantly, EfficientViT does not involve hardware-inefficient operations, so our FLOPs reduction can easily translate to latency reduction on mobile devices (Figure 1). On Qualcomm Snapdragon 8Gen1 CPU, EfficientViT executes 5.8× faster than SegNeXt [17] while reaching higher mIoU on Cityscapes and 7.9× faster than EfficientNet [39] without accuracy loss on ImageNet. We summarize our contributions as follows:

- We introduce a novel lightweight multi-scale attention for on-device semantic segmentation. It achieves a global receptive field and multi-scale learning while maintaining good efficiency on edge devices.
- We design EfficientViT, a new family of models, based on the proposed lightweight multi-scale attention module.
- On popular semantic segmentation benchmark datasets and ImageNet, our model demonstrates remarkable speedup on mobile over prior SOT semantic segmentation models.

2. Method

This section first introduces lightweight Multi-Scale ttention (MS). Unlike prior works, our lightweight MS module simultaneously achieves a global receptive field and multi-scale learning with only hardware-efficient operations. Then we present a new family of models named EfficientViT based on the proposed MS module for on-device semantic segmentation.

2.1. Lightweight Multi-Scale ttention

Our lightweight MS module balances two crucial aspects for on-device semantic segmentation, i.e., performance and efficiency. Specifically, a global receptive field

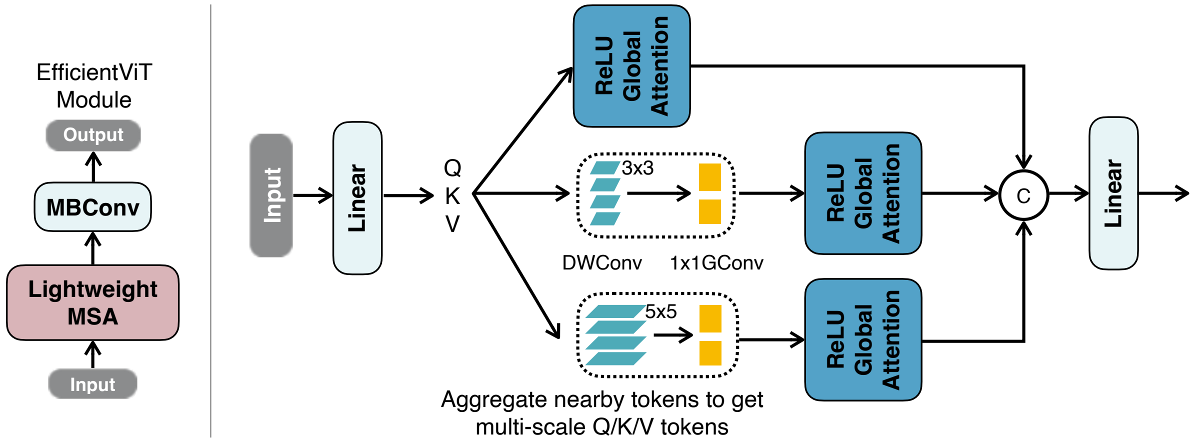

Figure 2: **Illustration of EfficientViT's Building Block (left) and the Proposed Lightweight Multi-Scale ttention (right).** *Left*: building block of EfficientViT consists of a lightweight MS module and an MBConv. The lightweight MS module is responsible for capturing context information, while the MBConv is for capturing local information. *Right*: fter getting Q/K/V tokens via the linear projection layer, we propose to generate multi-scale tokens by aggregating nearby tokens via lightweight small-kernel convolutions. ReLU-based global attention is applied to multi-scale tokens, and the outputs are concatenated and fed to the final linear projection layer for feature fusing.

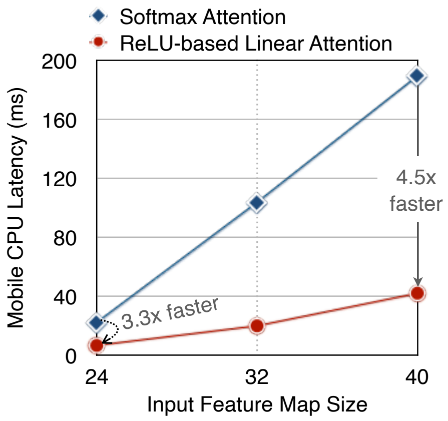

Figure 3: ReLU-based linear attention is 3.3-4.5× faster than softmax attention with similar computation, thanks to removing hardware-unfriendly operations (e.g., softmax). Latency is measured on Qualcomm Snapdragon 855 CPU with TensorFlow-Lite.

and multi-scale learning are essential from the performance perspective. Previous SOT segmentation models provide strong performances by enabling these features but fail to provide good efficiency. Our module tackles this issue by trading slight capacity loss for significant efficiency improvements.

n illustration of the proposed lightweight MS module is provided in Figure 2 (right). In particular, we propose to use lightweight ReLU-based attention [26] to enable the global receptive field instead of the heavy self-attention [41]. While ReLU-based attention [26] and other linear attention modules [2, 11, 38, 43] has been explored in other domains, it has never been applied to the semantic segmentation community. To the best of our knowledge, we are the first work demonstrating ReLU-based attention's effectiveness in semantic segmentation. In addition, our work introduces novel designs (lightweight MS module) to enhance the capacity, making it much more powerful in semantic segmentation.

Enabling Global Receptive Field with Lightweight ReLU-based ttention. Given input $x \in {}^{N \quad f}$, the generalized form of self-attention can be written as:

$$O_i = \sum_{j=}^{N} \frac{Sim\ Q_i, K_j)}{{}^{N}_{j=}\ Sim\ Q_i, K_j)} V_j, \tag{1}$$

where $Q = xW_Q$, $K = xW$, $V = xW_V$ and $W_Q/W\ \ /W_V \in {}^{f \quad d}$ is the learnable linear projection matrix. O_i represents the i-th row of matrix O. $Sim\quad)$ is the similarity function. When using the similarity function $Sim\ Q\ \ K) = \exp\ \frac{Q}{\sqrt{d}})$, Eq. (1) becomes the original self-attention [41].

part from $\exp\ \frac{Q}{\sqrt{d}})$, we can use other similarity functions. In this work, we use ReLU-based global attention [26] to achieve both the global receptive field and linear computational complexity. In ReLU-based global attention, the similarity function is defined as

$$Sim\ Q, K) = \text{ReLU}\ \ Q)\text{ReLU}\ \ K)^T \tag{2}$$

With $Sim\ Q\ \ K) = \text{ReLU}\ \ Q)\text{ReLU}\ \ K)^T$, Eq. (1) can

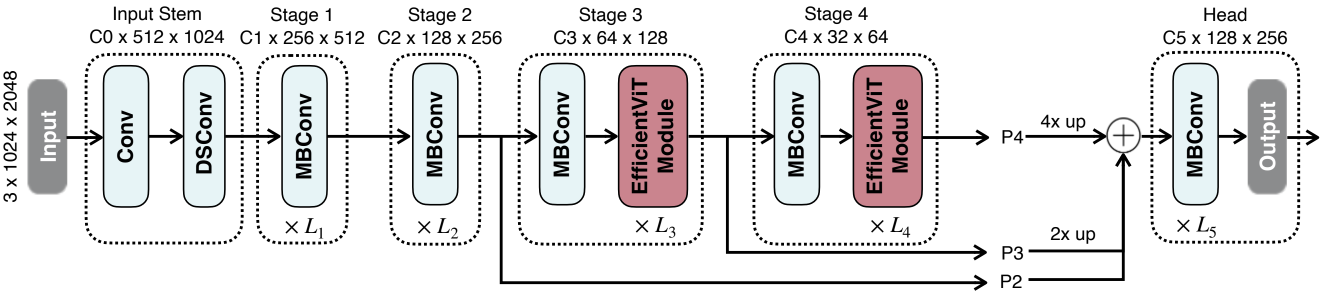

Figure 4: **Macro rchitecture of EfficientViT.** We adopt the standard backbone-head/encoder-decoder design. In the backbone, we insert our lightweight MS modules in Stages 3 and 4. Following the common practice, we feed the features from the last three stages (P2, P3, and P4) to the head. We use addition to fuse these features for simplicity and efficiency. s we already have lightweight MS modules in the backbone, we adopt a simple head design that consists of several MBConv blocks and output layers.

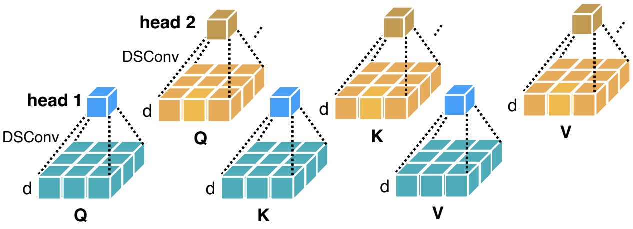

Figure 5: **Illustration of the ggregation Process for Generating Multi-Scale Tokens.** The information aggregation is done independently for each Q, K, and V in each head. 'd' denotes the dimension of each token. The typical value of d is 32.

be rewritten as:

$$O_i = \sum_{j=}^{N} \frac{\text{ReLU } Q_i)\text{ReLU } K_j)^T}{{}^{N}_{j=} \ \text{ReLU } Q_i)\text{ReLU } K_j)^T} V_j$$

$$= \frac{{}^{N}_{j=} \ \text{ReLU } Q_i)\text{ReLU } K_j)^T)V_j}{\text{ReLU } Q_i) \quad {}^{N}_{j=} \ \text{ReLU } K_j)^T}$$

Then, we can leverage the associative property of matrix multiplication to reduce the computational complexity and memory footprint from quadratic to linear without changing the functionality:

$$O_i = \frac{{}^{N}_{j=} \ [\text{ReLU } Q_i)\text{ReLU } K_j)^T]V_j}{\text{ReLU } Q_i) \quad {}^{N}_{j=} \ \text{ReLU } K_j)^T}$$

$$= \frac{{}^{N}_{j=} \ \text{ReLU } Q_i)[\ \text{ReLU } K_j)^T V_j)]}{\text{ReLU } Q_i) \quad {}^{N}_{j=} \ \text{ReLU } K_j)^T}$$

$$= \frac{\text{ReLU } Q_i) \quad {}^{N}_{j=} \ \text{ReLU } K_j)^T V_j)}{\text{ReLU } Q_i) \quad {}^{N}_{j=} \ \text{ReLU } K_j)^T)} \tag{3}$$

s shown in Eq. (3), we only need to compute ${}^{N}_{j=}$ ReLU $K_j)^T V_j) \in {}^{d \ d}$ and ${}^{N}_{j=}$ ReLU $K_j)^T)$ $\in {}^{d}$ once, then can reuse them for each query, thereby only requires $\mathcal{O} \ N)$ computational cost and $\mathcal{O} \ N)$ memory.

nother key merit of ReLU-based global attention is that it does not involve hardware-unfriendly operations like softmax, making it more efficient on hardware. For example, Figure 3 shows the latency comparison between softmax attention and ReLU-based linear attention. With similar computation, ReLU-based linear attention is significantly faster than softmax attention on mobile.

Generate Multi-Scale Tokens. ReLU-based attention alone has limited model capacity. To enhance ReLU-based global attention with multi-scale learning ability, we propose to aggregate the information from nearby Q/K/V tokens to get multi-scale tokens. The aggregation process is illustrated in Figure 5. This information aggregation process is independent for each Q, K, and V in each head. We only use small-kernel convolutions for information aggregation to avoid hurting hardware efficiency.

In the practical implementation, independently executing these aggregation operations is inefficient on GPU. Therefore, we take advantage of the infrastructure of group convolution in modern deep learning frameworks to reduce the number of total operations. Specifically, all DWConvs are fused into a single DWConv while all 1x1 Convs are combined into a single 1x1 group convolution (Figure 2 right) where the number of groups is 3 × #heads and the number of channels in each group is d.

fter getting multi-scale tokens, we perform global attention upon them to extract multi-scale global features. Finally, we concatenate the features from different scales along the head dimension and feed them to the final linear projection layer to fuse the features.

2.2. EfficientViT rchitecture

We build a new family of models based on the proposed lightweight MS module. The core building block (denoted as 'EfficientViT Module') is illustrated in Fig-

Table 2: **Detailed rchitecture Configurations of Different EfficientViT Variants.** We build a series of models to fit different efficiency constraints. 'C' denotes the number of channels. 'L' denotes the number of blocks. 'H' is the height of the feature map, and 'W' is the width of the feature map.

Variants	Feature Map Shape	EfficientViT-B0	EfficientViT-B1	EfficientViT-B2	EfficientViT-B3
Input Stem	C $\frac{}{2}$ $\frac{W}{2}$	C = 8, L = 1	C = 16, L = 1	C = 24, L = 1	C = 32, L = 1
Stage1	C $\frac{}{4}$ $\frac{W}{4}$	C = 16, L = 2	C = 32, L = 2	C = 48, L = 3	C = 64, L = 4
Stage2	C $\frac{}{8}$ $\frac{W}{8}$	C = 32, L = 2	C = 64, L = 3	C = 96, L = 4	C = 128, L = 6
Stage3	C $\frac{}{6}$ $\frac{W}{6}$	C = 64, L = 2	C = 128, L = 3	C = 192, L = 4	C = 256, L = 6
Stage4	C $\frac{}{32}$ $\frac{W}{32}$	C = 128, L = 2	C = 256, L = 4	C = 384, L = 6	C = 512, L = 9
Head	C $\frac{}{8}$ $\frac{W}{8}$	C = 32, L = 1	C = 64, L = 3	C = 96, L = 3	C = 128, L = 3

ure 2 (left). Specifically, an EfficientViT module comprises a lightweight MS module and an MBConv [37]. The lightweight MS module is for context information extraction, while the MBConv is for local information extraction.

The macro architecture of EfficientViT is demonstrated in Figure 4. We use the standard backbone-head/encoder-decoder architecture design.

- **Backbone.** The backbone of EfficientViT also follows the standard design, which consists of the input stem and four stages with gradually decreased feature map size and gradually increased channel number. We insert the EfficientViT module in Stages 3 and 4. For downsampling, we use an MBConv with stride 2.

- **Head.** P2, P3, and P4 denote the outputs of Stages 2, 3, and 4, forming a pyramid of feature maps. For simplicity and efficiency, we use 1x1 convolution and standard upsampling operation (e.g., bilinear/bicubic upsampling) to match their spatial and channel size and fuse them via addition. Since our backbone already has a strong context information extraction capacity, we adopt a simple head design that comprises several MBConv blocks and the output layers (i.e., prediction and upsample). In the experiments, we empirically find this simple head design is sufficient for achieving SOT performances thanks to our lightweight MS module.

  In addition to semantic segmentation, our model can be applied to other vision tasks, such as image classification, by combining the backbone with task-specific heads.

Following the same macro architecture, we design a series of models with different sizes to satisfy various efficiency constraints. The detailed configurations are demonstrated in Table 2. We name these models as EfficientViT-B0, EfficientViT-B1, EfficientViT-B2, and EfficientViT-B3, respectively.

Table 3: **blation Study on Two Key Components of Our Lightweight MS Module.** The mIoU and M Cs are measured on Cityscapes with 1024x2048 input resolution. We rescale the width of the models so that they have the same M Cs. Multi-scale learning and the global receptive field are essential for obtaining good semantic segmentation performance.

Components		mIoU ↑	Params ↓	M Cs ↓
Multi-scale	Global att.			
		68.1	0.7M	4.4G
		72.3	0.7M	4.4G
		72.2	0.7M	4.4G
		74.5	0.7M	4.4G

3. Experiments

3.1. Setups

Datasets. We evaluate the effectiveness of EfficientViT on two representative semantic segmentation datasets, including Cityscapes [12] and DE20K [53]. Cityscapes is an autonomous driving dataset that mainly focuses on urban scenes. It contains 5,000 fine-annotated high-resolution (1024x2048) images with 19 classes divided into three subsets of size 2,975/500/1,525 for training/validation/testing. DE20K is a scene-parsing dataset with 150 classes. It contains 20,210/2,000/3,352 images for training, validation, and testing, respectively.

 part from Cityscapes and DE20K, we also study the effectiveness of EfficientViT for image classification using the ImageNet dataset [14].

Latency Measurement. We measure the latency of the models on Qualcomm Snapdragon 8Gen1 CPU with Tensorflow-Lite[1], batch size 1 and fp32.

[1] https://www.tensorflow.org/lite

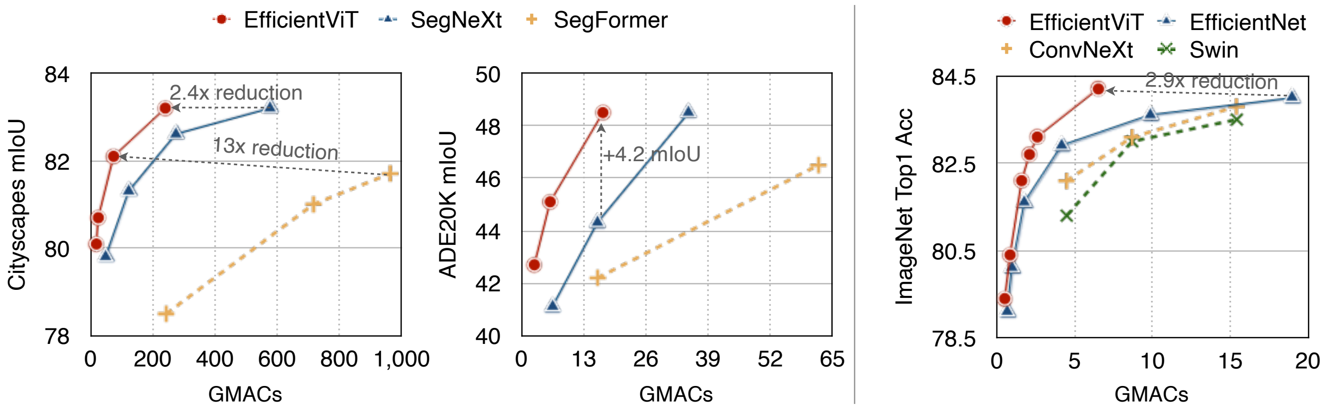

Figure 6: **M Cs vs. Performance.** EfficientViT provides a better trade-off between M Cs and performance than SOT semantic segmentation and image classification models. On Cityscapes, EfficientViT provides 13× and 2.4× M Cs reduction than SegFormer and SegNeXt, respectively, while achieving the same or higher performances. On ImageNet, EfficientViT achieves 2.9x M Cs reduction than EfficientNet without accuracy loss.

Table 4: **Backbone Performance of EfficientViT on ImageNet Classification.** 'r224' means the input resolution is 224x224. While EfficientViT is mainly designed for semantic segmentation, it also works well on ImageNet classification. With 6.5G M Cs, EfficientViT-B3 achieves 84.2 top1 ImageNet accuracy, surpassing EfficientNet-B6 while reducing the M Cs by 2.9x and being 7.9x faster on mobile.

Models	Top1 cc ↑	Top5 cc ↑	Params ↓	M Cs ↓	Mobile Latency ↓	Speedup ↑
EfficientNet-B1 [39]	79.1	94.4	7.8M	0.70G	87ms	1.0x
EfficientNetV2-B0 [40]	78.7	-	7.1M	0.72G	53ms	1.6x
EfficientViT-B1 (r224)	**79.4**	94.3	9.1M	0.52G	**19ms**	**4.6x**
EfficientNet-B2 [39]	80.1	94.9	9.2M	1.0G	118ms	1.0x
EfficientNetV2-B1 [40]	79.8	-	8.1M	1.2G	85ms	1.4x
EfficientViT-B1 (r288)	**80.4**	95.0	9.1M	0.86G	**31ms**	**3.8x**
Swin-T [29]	81.3	-	29M	4.5G	-	-
ConvNeXt-T [30]	82.1	-	29M	4.5G	356ms	1.0x
EfficientNet-B3 [39]	81.6	95.7	12M	1.8G	208ms	1.7x
EfficientNetV2-B3 [40]	82.1	-	14M	3.0G	201ms	1.8x
Co tNet-0 [13]	81.6	-	25M	4.2G	175ms	2.0x
EfficientViT-B2 (r256)	**82.7**	96.1	24M	2.1G	**72ms**	**4.9x**
Swin-S [29]	83.0	-	50M	8.7G	-	-
ConvNeXt-S [30]	83.1	-	50M	8.7G	622ms	1.0x
EfficientNet-B4 [39]	82.9	96.4	19M	4.2G	464ms	1.3x
Co tNet-1 [13]	83.3	-	42M	8.4G	332ms	1.9x
EfficientViT-B3 (r224)	**83.5**	96.4	49M	4.0G	**140ms**	**4.4x**
Swin-B [29]	83.5	-	88M	15G	-	-
EfficientNet-B6 [39]	84.0	96.8	43M	19G	1804ms	1.0x
ConvNeXt-B [30]	83.8	-	89M	15G	1025ms	1.8x
Co tNet-2 [13]	84.1	-	75M	16G	600ms	3.0x
EfficientNetV2-S [40]	83.9	-	22M	8.8G	509ms	3.5x
EfficientViT-B3 (r288)	**84.2**	96.7	49M	6.5G	**228ms**	**7.9x**

Implementation Details. We implement our models using Pytorch [34] and train them on GPUs. We use the damW optimizer with cosine learning rate decay for training our models. For lightweight multi-scale attention, we use a two-branch design for the best trade-off between performance and efficiency, where 5x5 nearby tokens are aggregated to generate multi-scale tokens.

For semantic segmentation experiments, we use the

Table 5: **Comparison with SOT Semantic Segmentation Models on Cityscapes.** 'r1024x2048' denotes the input resolution is 1024x2048. Models with similar mIoU are grouped for efficiency comparison. Compared with SegNeXt-T, EfficientViT-B1 achieves 2.7x M Cs reduction, 9.3x latency reduction, and 0.3 higher mIoU. Compared with SegFormer-B1, EfficientViT-B1 obtains 13x M Cs saving, 15x measured speedup, and 1.6 higher mIoU.

Models	mIoU ↑	Params ↓	M Cs ↓	Mobile Latency ↓	Speedup ↑
DeepLabV3plus-Mbv2 [7]	75.2	15M	555G	-	-
PSPNet-Mbv2 [52]	70.2	14M	423G	-	-
FCN-Mbv2 [31]	61.5	9.8M	317G	-	-
SegFormer-B0 (r768) [45]	75.3	3.8M	52G	2.8s	1.0x
EfficientViT-B0 (r960x1920)	**75.5**	0.7M	3.9G	**0.20s**	**14x**
HRFormer-S [49]	80.0	14M	836G	-	-
SegFormer-B1 [45]	78.5	14M	244G	12s	1.0x
SegNeXt-T [17]	79.8	4.3M	51G	7.6s	1.6x
EfficientViT-B1 (r896x1792)	**80.1**	4.8M	19G	**0.82s**	**15x**
HRFormer-B [49]	81.9	56M	2224G	-	-
SegFormer-B3 [45]	81.7	47M	963G	-	-
SegNeXt-S [17]	81.3	14M	125G	18s	1.0x
EfficientViT-B2 (r1024x2048)	**82.1**	15M	74G	**3.1s**	**5.8x**
SegFormer-B5 [45]	82.4	85M	1460G	-	-
SegNeXt-L [17]	**83.2**	49M	578G	-	-
EfficientViT-B3 (r1184x2368)	**83.2**	40M	240G	**10s**	-

mean Intersection over Union (mIoU) as our evaluation metric. The backbone is initialized with weights pretrained on ImageNet and the head is initialized randomly, following the common practice. Common data augmentation strategies such as random scaling, random horizontal flip, and random cropping are employed following prior works.

3.2. blation Study

Effectiveness of Our Lightweight MS Module. We conduct ablation study experiments on Cityscapes to study the effectiveness of two key design components of our lightweight MS module, i.e., multi-scale learning and global attention. To eliminate the impact of pre-training, we train all models from random initialization. In addition, we rescale the width of the models so that they have the same #M Cs. The results are summarized in Table 3. We can see that removing either global attention or multi-scale learning will significantly hurt the performances. It shows that all of them are essential for achieving a better trade-off between performance and efficiency.

Backbone Performance on ImageNet. To understand the effectiveness of EfficientViT's backbone in image classification, we train our models on ImageNet following the standard training strategy (300 epochs with random initialization, no knowledge distillation). We summarize the results and compare our models with SOT image classification models in Table 4.

Though EfficientViT is designed for semantic segmentation, it achieves highly competitive performances on ImageNet. In particular, EfficientViT-B3 obtains 84.2 top1 accuracy on ImageNet, providing +0.2 accuracy gain over EfficientNet-B6 and 7.9x speedup.

3.3. Main Results

Cityscapes. Table 5 reports the comparison between EfficientViT and SOT semantic segmentation models on Cityscapes. EfficientViT achieves remarkable efficiency improvements over prior SOT semantic segmentation models without sacrificing performances. Specifically, compared with SegFormer, EfficientViT obtains up to 13x M Cs saving and up to 15x latency reduction with higher mIoU. Compared with SegNeXt, EfficientViT provides up to 2.7x M Cs reduction and 9.3x speedup on mobile while maintaining higher mIoU.

Having similar computational cost, EfficientViT yields significant performance gains over previous SOT models. For example, EfficientViT-B3 yields +4.7 mIoU gain over SegFormer-B1 with similar M Cs.

DE20K. Table 6 summarizes the comparison between EfficientViT and SOT semantic segmentation models on DE20K. Similar to Cityscapes, we can see that EfficientViT also achieves significant efficiency improvements on DE20K. For example, with +0.5 mIoU gain, EfficientViT-B1 provides 5.9x M Cs reduction and 6.5x latency reduction than SegFormer-B1. With +0.8 mIoU gain,

Table 6: **Comparison with SOT Semantic Segmentation Models on DE20K.** Compared with SegNeXt-S, EfficientViT-B2 provides a 5.2x speedup and 0.8 mIoU gain. Compared with SegFormer-B1, EfficientViT-B1 achieves 0.5 higher mIoU with a 6.5x speedup.

Models	mIoU ↑	Params ↓	M Cs ↓	Mobile Latency ↓	Speedup ↑
SegFormer-B1 [45]	42.2	14M	16G	0.65s	1.0x
SegNeXt-T [17]	41.1	4.3M	6.6G	0.61s	1.1x
EfficientViT-B1 (r480)	**42.7**	4.8M	2.7G	**0.10s**	**6.5x**
HRFormer-S [49]	44.0	14M	110G	-	-
SegNeXt-S [17]	44.3	14M	16G	1.1s	1.0x
EfficientViT-B2 (r416)	**45.1**	15M	6.0G	**0.21s**	**5.2x**
Mask2Former [9]	47.7	47M	74G	-	-
MaskFormer [10]	46.7	42M	55G	-	-
SegFormer-B2 [45]	46.5	28M	62G	2.6s	1.0x
EfficientViT-B3 (r384)	**48.0**	39M	12G	**0.45s**	**5.8x**
HRFormer-B [49]	48.7	56M	280G	-	-
SegNeXt-B [17]	48.5	28M	35G	2.2s	1.0x
EfficientViT-B3 (r512)	**49.0**	39M	22G	**0.80s**	**2.8x**

EfficientViT-B2 requires 2.7x fewer M Cs and runs 5.2x faster than SegNeXt-S.

4. Related Work

Semantic Segmentation. Semantic segmentation targets producing a class prediction for each pixel given the input image. It can be viewed as an extension of image classification from per-image prediction to per-pixel predictions. Since the groundbreaking work FCN [31], which designs a fully convolutional neural network for end-to-end pixel-to-pixel prediction, extensive studies have been done to improve the performance for semantic segmentation [1, 36, 47, 52, 48, 42].

In addition, there are also some works targeting improving the efficiency of semantic segmentation models [51, 35, 27, 46, 50]. Representative examples include ICNet [51], DF Net [27], BiSeNet [46], etc. While these models provide good efficiency, their performances are far behind SOT semantic segmentation models, especially on the challenging Cityscapes dataset.

Compared to these works, our models provide a better trade-off between performance and efficiency by enabling a global receptive field and multi-scale learning with lightweight operations.

Efficient Vision Transformer. While ViT provides impressive performances in the high-computation region, it is usually inferior to previous efficient CNNs [39, 24, 5, 18] when targeting the low-computation region. To close the gap, MobileViT [33] proposes to combine the strength of CNN and ViT by replacing local processing in convolutions with global processing using transformers. Mobile-Former [8] proposes to parallelize MobileNet and Transformer with a two-way bridge in between for feature fusing. N SViT [16] proposes to leverage neural architecture search to search for efficient ViT architectures.

However, these models mainly focus on image classification and still rely on self-attention with quadratic computational complexity, thus unsuitable for on-device semantic segmentation.

Efficient Deep Learning. Our work is also related to efficient deep learning, which aims at improving the efficiency of deep neural networks so that we can deploy them on hardware platforms with limited resources, such as mobile phones and IoT devices. Typical technologies in efficient deep learning include network pruning [20, 22, 28], quantization [19], efficient model architecture design [25, 32], and training techniques [23, 4]. In addition to manual designs, many recent works use utoML techniques [54, 3, 6] to automatically design [5], prune [21] and quantize [44] neural networks.

5. Conclusion

In this work, we studied efficient architecture design for on-device semantic segmentation. We introduced a lightweight multi-scale attention module that simultaneously achieves a global receptive field, and multi-scale learning with lightweight and hardware-efficient operations, thus providing significant speedup on edge devices without performance loss than SOT semantic segmentation models. For future work, we will explore applying EfficientViT to other vision tasks and further scaling up our EfficientViT models.

References

[1] Vijay Badrinarayanan, lex Kendall, and Roberto Cipolla. Segnet: deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 1, 8

[2] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, and Judy Hoffman. Hydra attention: Efficient attention with many heads. In *Computer Vision–ECCV 2022 Workshops: Tel viv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 35–49. Springer, 2023. 3

[3] Han Cai, Tianyao Chen, Weinan Zhang, Yong Yu, and Jun Wang. Efficient architecture search by network transformation. In *I*, 2018. 8

[4] Han Cai, Chuang Gan, Ji Lin, and Song Han. Network augmentation for tiny deep learning. *arXiv preprint arXiv:2110.08890*, 2021. 8

[5] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once for all: Train one network and specialize it for efficient deployment. In *ICLR*, 2020. 8

[6] Han Cai, Ligeng Zhu, and Song Han. ProxylessN S: Direct neural architecture search on target task and hardware. In *ICLR*, 2019. 8

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig dam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 7

[8] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobileformer: Bridging mobilenet and transformer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022. 8

[9] Bowen Cheng, Ishan Misra, lexander G Schwing, lexander 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. 8

[10] Bowen Cheng, lex Schwing, and lexander Kirillov. Perpixel classification is not all you need for semantic segmentation. *dvances in Neural Information Processing Systems*, 34, 2021. 8

[11] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, ndreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, froz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020. 3

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2, 5

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

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: large-scale hierarchical image database. In *CVPR*, 2009. 5

[15] Xiaohan Ding, Xiangyu Zhang, Yizhuang Zhou, Jungong Han, Guiguang Ding, and Jian Sun. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022. 1

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

[17] Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zheng-Ning Liu, Ming-Ming Cheng, and Shi min Hu. Segnext: Rethinking convolutional attention design for semantic segmentation. In lice H. Oh, lekh garwal, Danielle Belgrave, and Kyunghyun Cho, editors, *dvances in Neural Information Processing Systems*, 2022. 1, 2, 7, 8

[18] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1580–1589, 2020. 8

[19] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR*, 2016. 8

[20] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *NeurIPS*, 2015. 8

[21] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. mc: utoml for model compression and acceleration on mobile devices. In *ECCV*, 2018. 8

[22] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *ICCV*, 2017. 8

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

[24] ndrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019. 8

[25] ndrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco ndreetto, and Hartwig dam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 8

[26] ngelos Katharopoulos, poorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 1, 3

[27] Hanchao Li, Pengfei Xiong, Haoqiang Fan, and Jian Sun. Dfanet: Deep feature aggregation for real-time semantic segmentation. In *Proceedings of the IEEE/CVF conference on*

computer vision and pattern recognition, pages 9522–9531, 2019. 8

[28] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *ICCV*, 2017. 8

[29] 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. 6

[30] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. convnet for the 2020s. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022. 6

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 7, 8

[32] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, 2018. 8

[33] Sachin Mehta and Mohammad Rastegari. Mobilevit: Lightweight, general-purpose, and mobile-friendly vision transformer. In *International Conference on Learning Representations*, 2022. 8

[34] dam Paszke, Sam Gross, Francisco Massa, dam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca ntiga, et al. Pytorch: n imperative style, high-performance deep learning library. *dvances in neural information processing systems*, 32, 2019. 6

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

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer- ssisted Intervention–MICC I 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 1, 8

[37] Mark Sandler, ndrew Howard, Menglong Zhu, ndrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 5

[38] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: ttention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3531–3539, 2021. 3

[39] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 2, 6, 8

[40] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021. 6

[41] shish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, idan N Gomez, Łukasz Kaiser, and Illia Polosukhin. ttention is all you need. In *NeurIPS*, 2017. 3

[42] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020. 1, 8

[43] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 3

[44] Tianzhe Wang, Kuan Wang, Han Cai, Ji Lin, Zhijian Liu, Hanrui Wang, Yujun Lin, and Song Han. pq: Joint search for network architecture, pruning and quantization policy. In *CVPR*, 2020. 8

[45] Enze Xie, Wenhai Wang, Zhiding Yu, nima nandkumar, Jose M lvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *dvances in Neural Information Processing Systems*, 34, 2021. 1, 2, 7, 8

[46] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018. 8

[47] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 1, 8

[48] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *European conference on computer vision*, pages 173–190. Springer, 2020. 1, 8

[49] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution vision transformer for dense predict. *dvances in Neural Information Processing Systems*, 34, 2021. 2, 7, 8

[50] 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. 8

[51] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 405–420, 2018. 8

[52] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 1, 7, 8

[53] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, dela Barriuso, and ntonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 2, 5

[54] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017. 8