

EfficientViT: Lightweight Multi-Scale Attention for High-Resolution Dense Prediction

Han Cai¹, Junyan Li², Muyan Hu³, Chuang Gan⁴, Song Han¹

¹MIT, ²Zhejiang University, ³Tsinghua University, ⁴MIT-IBM Watson AI Lab

<https://github.com/mit-han-lab/efficientvit>

Abstract

High-resolution dense prediction enables many appealing real-world applications, such as computational photography, autonomous driving, etc. However, the vast computational cost makes deploying state-of-the-art high-resolution dense prediction models on hardware devices difficult. This work presents EfficientViT, a new family of high-resolution vision models with novel lightweight multi-scale attention. Unlike prior high-resolution dense prediction 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 high-resolution dense prediction) with only lightweight and hardware-efficient operations. As such, EfficientViT delivers remarkable performance gains over previous state-of-the-art high-resolution dense prediction models with significant speedup on diverse hardware platforms, including mobile CPU, edge GPU, and cloud GPU. Without performance loss on Cityscapes, our EfficientViT provides up to $8.8\times$ and $3.8\times$ GPU latency reduction over SegFormer and SegNeXt, respectively. For super-resolution, EfficientViT provides up to $6.4\times$ speedup over Restormer while providing 0.11dB gain in PSNR.

1. Introduction

High-resolution dense prediction is a fundamental task in computer vision and has broad applications in real-world scenarios, including autonomous driving, medical image processing, computational photography, etc. Therefore, deploying state-of-the-art (SOTA) high-resolution dense prediction models on hardware devices can benefit many use cases.

However, there is a large gap between the computational cost required by SOTA high-resolution dense prediction models and the limited resources of hardware devices. It

makes using these models in real-world applications impractical. In particular, high-resolution dense prediction models require high-resolution images and strong context information extraction ability to work well [2, 41, 53, 58, 54, 47]. Therefore, directly porting efficient model architecture from image classification is unsuitable for high-resolution dense prediction.

This work introduces **EfficientViT**, a new family of models for efficient high-resolution dense prediction. 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 SOTA high-resolution dense prediction models. They demonstrate that the multi-scale learning [53, 58], and global receptive field [51] play a critical role in improving the performances. However, they do not consider hardware efficiency when designing their models, which is essential for real-world applications. For example, SegFormer [51] 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 [18] 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 [16, 50], which is usually not available on hardware 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 [28]. 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 hardware deployment (Figure 3).

Furthermore, we propose a novel lightweight multi-scale

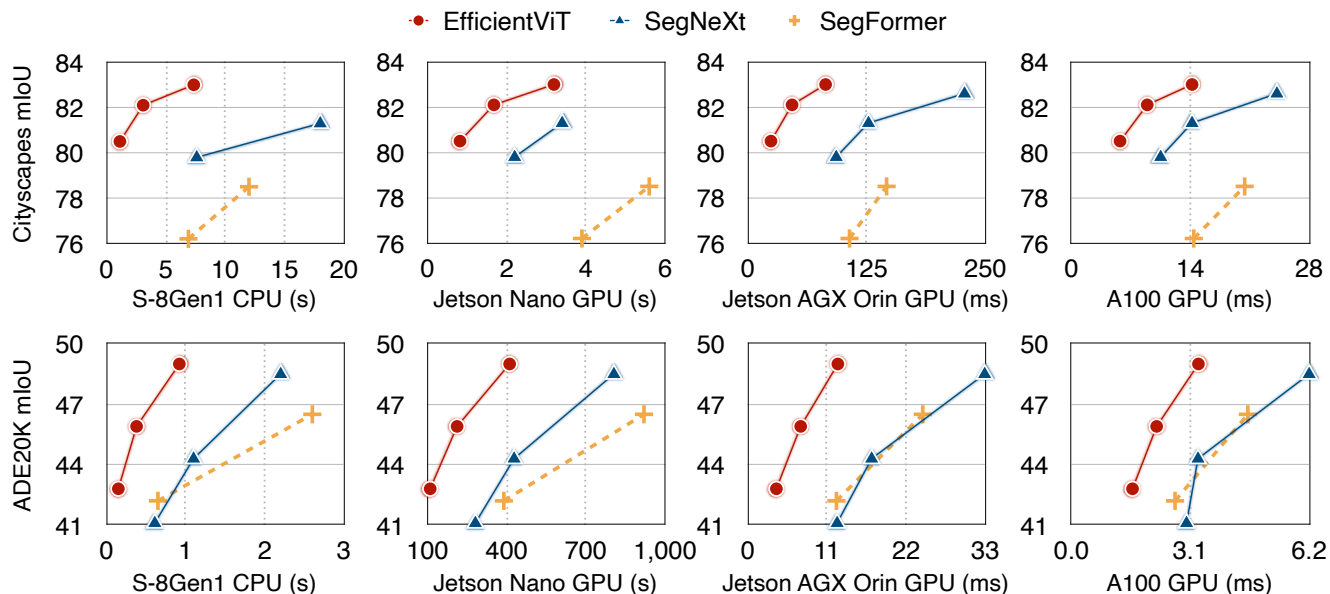


Figure 1: **Latency vs. Performance.** All performance results are obtained with the single model and single-scale inference. The mobile latency results are obtained on the Qualcomm Snapdragon 8Gen1 (S-8Gen1) CPU using Tensorflow-Lite, batch size 1, and fp32. The GPU latency results are obtained on two edge GPUs (Jetson Nano, Jetson AGX Orin) and one cloud GPU (A100) using TensorRT, batch size 1, and fp16. EfficientViT consistently achieves a remarkable boost in speed on diverse hardware platforms while providing the same/higher performances on Cityscapes and ADE20K than state-of-the-art (SOTA) segmentation models.

Table 1: **Desirable Features for Efficient High-Resolution Dense Prediction.** ‘Linear computational complexity’ means the computational cost grows linearly as the input resolution increases.

| Features | SegFormer [51] | HRFormer [55] | SegNeXt [18] | 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 SOTA high-resolution dense prediction models in Table 1.

We extensively evaluate EfficientViT on two popular high-resolution dense prediction tasks: semantic segmentation and super-resolution. EfficientViT provides significant performance boosts over prior SOTA high-resolution dense prediction models. More importantly, EfficientViT does not involve hardware-inefficient operations, so our FLOPs reduction can easily translate to latency reduction on hardware devices (Figure 1). We summarize our contributions

as follows:

- We introduce a novel lightweight multi-scale attention for efficient high-resolution dense prediction. It achieves a global receptive field and multi-scale learning while maintaining good efficiency on hardware.
- We design EfficientViT, a new family of high-resolution vision models, based on the proposed lightweight multi-scale attention module.
- On semantic segmentation, super-resolution and ImageNet classification, our model demonstrates remarkable speedup on diverse hardware platforms (mobile CPU, edge GPU, and cloud GPU) over prior SOTA models.

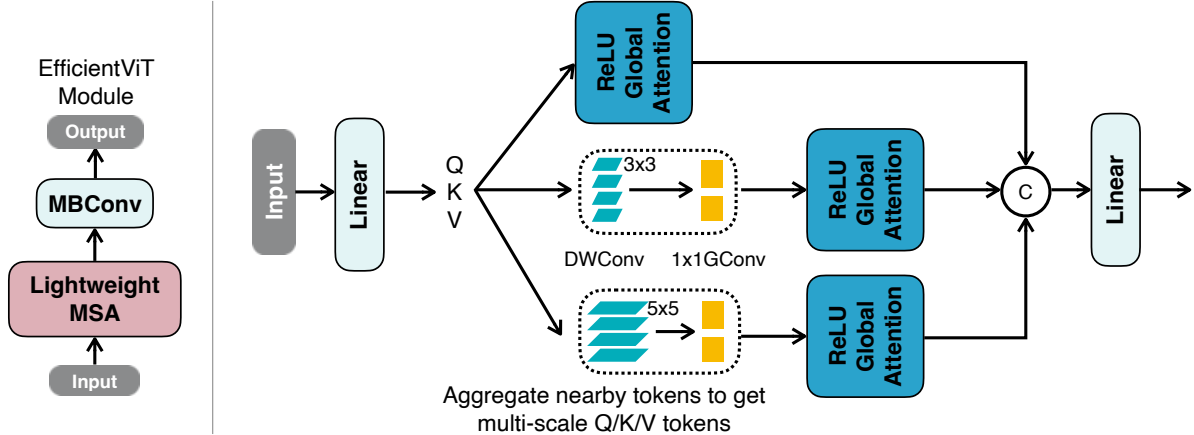


Figure 2: **Illustration of EfficientViT’s Building Block (left) and the Lightweight Multi-Scale Attention (right).** *Left:* A building block of EfficientViT consists of a lightweight MSA module and an MBConv. The lightweight MSA module is responsible for capturing context information, while the MBConv is for capturing local information. *Right:* After getting Q/K/V tokens via the linear projection layer, we 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.

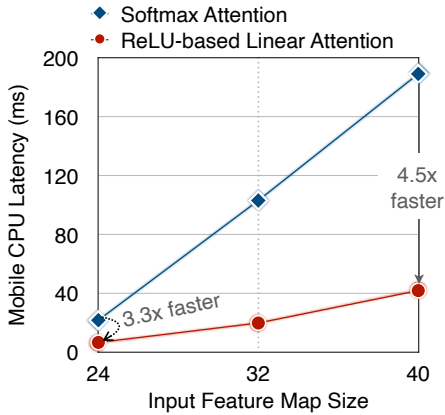


Figure 3: **Latency Comparison Between Softmax Attention and ReLU-Based Linear Attention.** ReLU-based linear attention is 3.3-4.5× faster than softmax attention with similar computation, thanks to removing hardware-unfriendly operations (e.g., softmax). Latency is measured on Qualcomm Snapdragon 855 CPU with TensorFlow-Lite, batch size 1, and fp32.

2. Method

This section first introduces lightweight Multi-Scale Attention (MSA). Unlike prior works, our lightweight MSA module simultaneously achieves a global receptive field and multi-scale learning with only hardware-efficient operations. Then, based on the proposed MSA module, we present a new family of vision models named EfficientViT for high-resolution dense prediction.

2.1. Lightweight Multi-Scale Attention

Our lightweight MSA module balances two crucial aspects of efficient high-resolution dense prediction, i.e., performance and efficiency. Specifically, a global receptive field and multi-scale learning are essential from the performance perspective. Previous SOTA high-resolution dense prediction 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.

An illustration of the proposed lightweight MSA module is provided in Figure 2 (right). In particular, we propose to use lightweight ReLU-based attention [28] to enable the global receptive field instead of the heavy self-attention [46]. While ReLU-based attention [28] and other linear attention modules [3, 12, 43, 48] has been explored in other domains, it has never been successfully applied to high-resolution dense prediction. To the best of our knowledge, EfficientViT is the first work demonstrating ReLU-based attention’s effectiveness in high-resolution dense prediction. In addition, our work introduces novel designs (lightweight MSA module) to enhance the capacity, making it more powerful in high-resolution dense prediction.

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

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

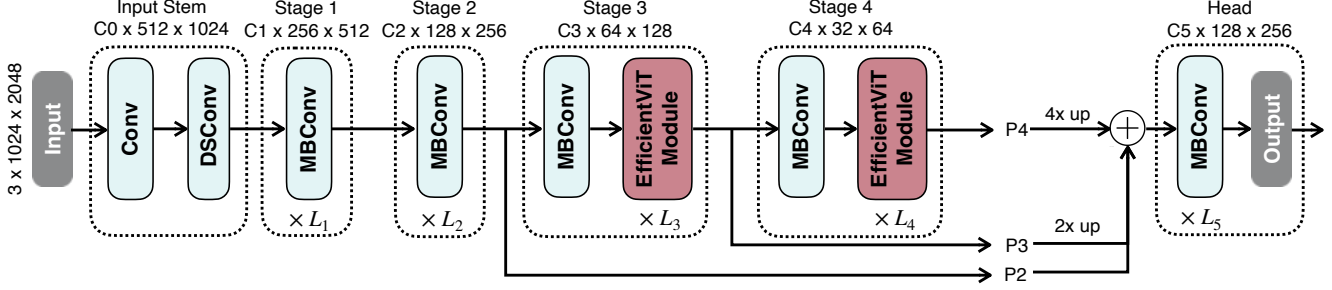


Figure 4: **Macro Architecture of EfficientViT.** We adopt the standard backbone-head/encoder-decoder design. In the backbone, we insert our EfficientViT modules in Stages 3 & 4. Following the common practice, we feed the features from the last three stages (P2, P3, and P4) to the head. We use addition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

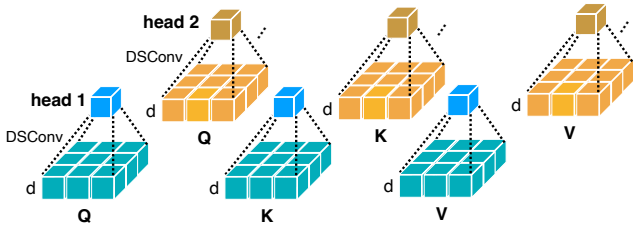


Figure 5: **Illustration of the Aggregation Process for Generating Multi-Scale Tokens.** The information aggregation is done independently for each Q, K, and V in each head. ‘d’ denotes the dimension of each token. The typical value of d is 32.

where $Q = xW_Q$, $K = xW_K$, $V = xW_V$ and $W_Q/W_K/W_V \in \mathbb{R}^{f \times d}$ is the learnable linear projection matrix. O_i represents the i -th row of matrix O . $Sim(\cdot, \cdot)$ is the similarity function. When using the similarity function $Sim(Q, K) = \exp(\frac{QK^T}{\sqrt{d}})$, Eq. (1) becomes the original self-attention [46].

Apart from $\exp(\frac{QK^T}{\sqrt{d}})$, we can use other similarity functions. In this work, we use ReLU-based global attention [28] to achieve both the global receptive field and linear computational complexity. In ReLU-based global attention, the similarity function is defined as

$$Sim(Q, K) = \text{ReLU}(Q)\text{ReLU}(K)^T. \quad (2)$$

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

$$\begin{aligned} O_i &= \sum_{j=1}^N \frac{\text{ReLU}(Q_i)\text{ReLU}(K_j)^T}{\sum_{j=1}^N \text{ReLU}(Q_i)\text{ReLU}(K_j)^T} V_j \\ &= \frac{\sum_{j=1}^N (\text{ReLU}(Q_i)\text{ReLU}(K_j)^T) V_j}{\text{ReLU}(Q_i) \sum_{j=1}^N \text{ReLU}(K_j)^T}. \end{aligned}$$

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:

$$\begin{aligned} O_i &= \frac{\sum_{j=1}^N [\text{ReLU}(Q_i)\text{ReLU}(K_j)^T] V_j}{\text{ReLU}(Q_i) \sum_{j=1}^N \text{ReLU}(K_j)^T} \\ &= \frac{\sum_{j=1}^N \text{ReLU}(Q_i) [\text{ReLU}(K_j)^T V_j]}{\text{ReLU}(Q_i) \sum_{j=1}^N \text{ReLU}(K_j)^T} \\ &= \frac{\text{ReLU}(Q_i) (\sum_{j=1}^N \text{ReLU}(K_j)^T V_j)}{\text{ReLU}(Q_i) (\sum_{j=1}^N \text{ReLU}(K_j)^T)}. \end{aligned} \quad (3)$$

As shown in Eq. (3), we only need to compute $(\sum_{j=1}^N \text{ReLU}(K_j)^T V_j) \in \mathbb{R}^{d \times d}$ and $(\sum_{j=1}^N \text{ReLU}(K_j)^T) \in \mathbb{R}^{d \times 1}$ once, then can reuse them for each query, thereby only requires $\mathcal{O}(N)$ computational cost and $\mathcal{O}(N)$ memory.

Another 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 mitigate this limitation, we first enhance it with the depthwise convolution to improve its local information extraction ability (Figure 2 left). In addition, to enhance its 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

Table 2: **Detailed Architecture 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 | $C \times \frac{H}{2} \times \frac{W}{2}$ | C = 8, L = 1 | C = 16, L = 1 | C = 24, L = 1 | C = 32, L = 1 |
| Stage1 | $C \times \frac{H}{4} \times \frac{W}{4}$ | C = 16, L = 2 | C = 32, L = 2 | C = 48, L = 3 | C = 64, L = 4 |
| Stage2 | $C \times \frac{H}{8} \times \frac{W}{8}$ | C = 32, L = 2 | C = 64, L = 3 | C = 96, L = 4 | C = 128, L = 6 |
| Stage3 | $C \times \frac{H}{16} \times \frac{W}{16}$ | C = 64, L = 2 | C = 128, L = 3 | C = 192, L = 4 | C = 256, L = 6 |
| Stage4 | $C \times \frac{H}{32} \times \frac{W}{32}$ | C = 128, L = 2 | C = 256, L = 4 | C = 384, L = 6 | C = 512, L = 9 |
| Head | $C \times \frac{H}{8} \times \frac{W}{8}$ | C = 32, L = 1 | C = 64, L = 3 | C = 96, L = 3 | C = 128, L = 3 |

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 $3 \times \text{\#heads}$ and the number of channels in each group is d .

After 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 Architecture

We build a new family of vision models based on the proposed lightweight MSA module. The core building block (denoted as ‘EfficientViT Module’) is illustrated in Figure 2 (left). Specifically, an EfficientViT module comprises a lightweight MSA module and an MBCConv [42]. The lightweight MSA module is for context information extraction, while the MBCConv 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 MBCConv 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 MBCConv blocks and the output layers (i.e., prediction and upsample). In the

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

| Components | | mIoU ↑ | Params ↓ | MACs ↓ |
|-------------|-------------|-------------|----------|--------|
| 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 |

experiments, we empirically find this simple head design is sufficient for achieving SOTA performances thanks to our lightweight MSA module.

In addition to dense prediction, 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.

3. Experiments

3.1. Setups

Datasets. We evaluate the effectiveness of EfficientViT on two representative high-resolution dense prediction tasks, including semantic segmentation and super-resolution.

For semantic segmentation, we use two popular benchmark datasets: Cityscapes [13] and ADE20K [59].

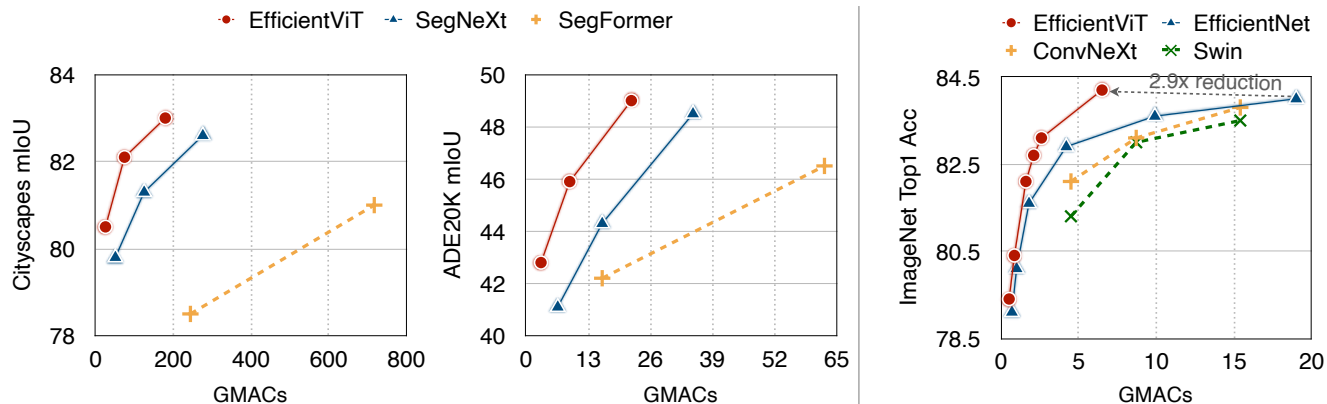


Figure 6: **MACs vs. Performance.** EfficientViT provides a better trade-off between MACs and performance than SOTA semantic segmentation and image classification models.

Table 4: **Backbone Performance of EfficientViT on ImageNet Classification.** ‘r224’ means the input resolution is 224x224. With 6.5G MACs, EfficientViT-B3 achieves 84.2 ImageNet top1 accuracy, surpassing EfficientNet-B6 while reducing the MACs by 2.9x and being 3.4x faster on Jetson Nano GPU and 3.7x faster on A100 GPU. ‘bs1’ represents that the latency is measured with batch size 1.

| Models | Top1 Acc \uparrow | Top5 Acc \uparrow | Params \downarrow | MACs \downarrow | Nano(bs1) | Latency \downarrow Orin(bs1) | A100(bs16) |
|-------------------------------|---------------------|---------------------|---------------------|-------------------|---------------|-----------------------------------|--------------|
| EfficientNet-B1 [44] | 79.1 | 94.4 | 7.8M | 0.70G | 36.2ms | 2.0ms | 2.9ms |
| EfficientViT-B1 (r224) | 79.4 | 94.3 | 9.1M | 0.52G | 24.8ms | 1.5ms | 1.9ms |
| EfficientNet-B2 [44] | 80.1 | 94.9 | 9.2M | 1.0G | 45.1ms | 2.3ms | 3.5ms |
| EfficientViT-B1 (r288) | 80.4 | 95.0 | 9.1M | 0.86G | 34.5ms | 1.8ms | 2.9ms |
| CoAtNet-0 [14] | 81.6 | - | 25M | 4.2G | 95.8ms | 4.5ms | 6.9ms |
| ConvNeXt-T [34] | 82.1 | - | 29M | 4.5G | 87.9ms | 3.8ms | 6.1ms |
| EfficientNet-B3 [44] | 81.6 | 95.7 | 12M | 1.8G | 67.8ms | 3.3ms | 5.1ms |
| EfficientViT-B2 (r256) | 82.7 | 96.1 | 24M | 2.1G | 58.5ms | 2.8ms | 4.0ms |
| Swin-B [33] | 83.5 | - | 88M | 15G | 240ms | 6.0ms | 6.9ms |
| CoAtNet-1 [14] | 83.3 | - | 42M | 8.4G | 171ms | 8.3ms | 13.0ms |
| ConvNeXt-S [34] | 83.1 | - | 50M | 8.7G | 146ms | 6.5ms | 10.0ms |
| EfficientNet-B4 [44] | 82.9 | 96.4 | 19M | 4.2G | 143ms | 5.7ms | 9.7ms |
| EfficientViT-B3 (r224) | 83.5 | 96.4 | 49M | 4.0G | 101ms | 4.4ms | 5.7ms |
| EfficientNet-B6 [44] | 84.0 | 96.8 | 43M | 19G | 475ms | 16.2ms | 30.2ms |
| CoAtNet-2 [14] | 84.1 | - | 75M | 16G | 254ms | 10.3ms | 16.8ms |
| ConvNeXt-B [34] | 83.8 | - | 89M | 15G | 211ms | 7.8ms | 12.3ms |
| EfficientViT-B3 (r288) | 84.2 | 96.7 | 49M | 6.5G | 141ms | 5.6ms | 8.2ms |
| CoAtNet-3 [14] | 84.5 | - | 168M | 35G | - | - | 24.9ms |
| ConvNeXt-L [34] | 84.3 | - | 198M | 34G | - | - | 18.5ms |
| EfficientNetV2-S [45] | 83.9 | - | 22M | 8.8G | - | - | 7.0ms |
| EfficientViT-L1 (r224) | 84.5 | 96.9 | 53M | 5.3G | - | - | 3.2ms |

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. ADE20K is a scene-parsing dataset with 150 classes. It contains 20,210/2,000/3,352 images for

training, validation, and testing, respectively.

For super-resolution, we evaluate EfficientViT under two settings: lightweight super-resolution (SR) and high-resolution SR. For lightweight SR, we train models on DIV2K [1] and test on BSD100 [37]. For high-resolution SR, we train models on the first 3000 training images of

Table 5: **Comparison with SOTA Semantic Segmentation Models on Cityscapes.** The input resolution is 1024x2048 for all models. Models with similar mIoU are grouped for efficiency comparison. Compared with SegNeXt-T, EfficientViT-B1 achieves 2.0x MACs reduction, 3.8x latency reduction on Jetson AGX Orin GPU, and 0.7 higher mIoU. Compared with SegFormer-B1, EfficientViT-B1 obtains 9.8x MACs saving and 2.0 higher mIoU.

| Models | mIoU \uparrow | Params \downarrow | MACs \downarrow | Latency \downarrow | | |
|------------------------|-----------------|---------------------|-------------------|----------------------|---------------|---------------|
| | | | | Nano(bs1) | Orin(bs1) | A100(bs1) |
| PSPNet-Mbv2 [58] | 70.2 | 14M | 423G | - | - | - |
| DeepLabV3plus-Mbv2 [8] | 75.2 | 15M | 555G | - | 83.5ms | 9.8ms |
| FCN-Mbv2 [35] | 61.5 | 9.8M | 317G | 2.1s | 52.0ms | 6.4ms |
| EfficientViT-B0 | 75.7 | 0.7M | 4.4G | 0.28s | 9.9ms | 3.8ms |
| HRFormer-S [55] | 80.0 | 14M | 836G | - | - | - |
| SegFormer-B1 [51] | 78.5 | 14M | 244G | 5.6s | 146ms | 20.4ms |
| SegNeXt-T [18] | 79.8 | 4.3M | 51G | 2.2s | 93.2ms | 10.5ms |
| EfficientViT-B1 | 80.5 | 4.8M | 25G | 0.82s | 24.3ms | 5.7ms |
| HRFormer-B [55] | 81.9 | 56M | 2224G | - | - | - |
| SegFormer-B3 [51] | 81.7 | 47M | 963G | - | 407ms | 54.3ms |
| SegNeXt-S [18] | 81.3 | 14M | 125G | 3.4s | 127ms | 14.2ms |
| EfficientViT-B2 | 82.1 | 15M | 74G | 1.7s | 46.5ms | 8.9ms |
| SegFormer-B5 [51] | 82.4 | 85M | 1460G | - | 638ms | 82.1ms |
| SegNeXt-B [18] | 82.6 | 28M | 276G | - | 228ms | 24.2ms |
| EfficientViT-B3 | 83.0 | 40M | 179G | 3.2s | 81.8ms | 14.2ms |

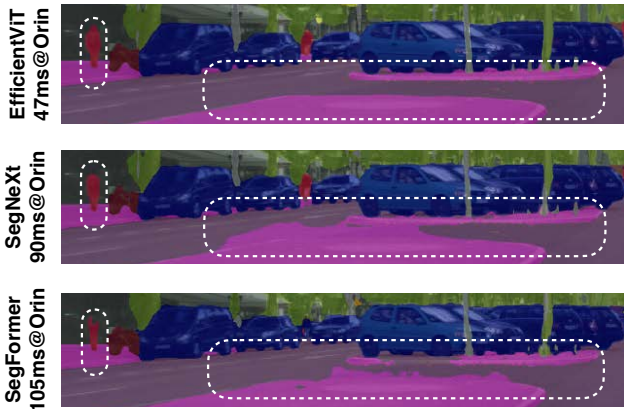


Figure 7: **Qualitative results on Cityscapes.**

FFHQ [27] and test on the first 500 validation images of FFHQ.

Apart from dense prediction, we also study the effectiveness of EfficientViT for image classification using the ImageNet dataset [15].

Latency Measurement. We measure the mobile latency on Qualcomm Snapdragon 8Gen1 CPU with Tensorflow-Lite¹, batch size 1 and fp32. We use TensorRT² and fp16 to measure the latency on edge GPU and cloud GPU. The data

¹<https://www.tensorflow.org/lite>

²<https://docs.nvidia.com/deeplearning/tensorrt/>

transfer time is included in the reported latency.

Implementation Details. We implement our models using Pytorch [39] and train them on GPUs. We use the AdamW 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 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.

For super-resolution, we use PSNR and SSIM on the Y channel as the evaluation metrics, same as previous work [31]. The models are trained with random initialization.

3.2. Ablation Study

Effectiveness of Our Lightweight MSA Module. We conduct ablation study experiments on Cityscapes to study the effectiveness of two key design components of our EfficientViT 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 #MACs. 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

Table 6: **Comparison with SOTA Semantic Segmentation Models on ADE20K.** Compared with SegNeXt-S, EfficientViT-B2 provides a 2.0x speedup on Jetson Nano GPU and 1.6 mIoU gain. Compared with SegFormer-B1, EfficientViT-B1 achieves 0.6 higher mIoU with a 3.1x speedup on Jetson AGX Orin GPU.

| Models | mIoU \uparrow | Params \downarrow | MACs \downarrow | Latency \downarrow | | |
|------------------------|-----------------|---------------------|-------------------|----------------------|---------------|--------------|
| | | | | Nano(bs1) | Orin(bs1) | A100(bs1) |
| SegFormer-B1 [51] | 42.2 | 14M | 16G | 389ms | 12.3ms | 2.7ms |
| SegNeXt-T [18] | 41.1 | 4.3M | 6.6G | 281ms | 12.4ms | 3.0ms |
| EfficientViT-B1 | 42.8 | 4.8M | 3.1G | 110ms | 4.0ms | 1.6ms |
| HRFormer-S [55] | 44.0 | 14M | 110G | - | - | - |
| SegNeXt-S [18] | 44.3 | 14M | 16G | 428ms | 17.2ms | 3.3ms |
| EfficientViT-B2 | 45.9 | 15M | 9.1G | 212ms | 7.3ms | 2.2ms |
| HRFormer-B [55] | 48.7 | 56M | 280G | - | - | - |
| Mask2Former [10] | 47.7 | 47M | 74G | - | - | - |
| MaskFormer [11] | 46.7 | 42M | 55G | - | - | - |
| SegFormer-B2 [51] | 46.5 | 28M | 62G | 920ms | 24.3ms | 4.6ms |
| SegNeXt-B [18] | 48.5 | 28M | 35G | 806ms | 32.9ms | 6.2ms |
| EfficientViT-B3 | 49.0 | 39M | 22G | 411ms | 12.5ms | 3.3ms |

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. We summarize the results and compare our models with SOTA image classification models in Table 4.

Though EfficientViT is designed for high-resolution dense prediction, it achieves highly competitive performances on ImageNet classification. In particular, EfficientViT-B3 obtains 84.2 top1 accuracy on ImageNet, providing +0.2 accuracy gain over EfficientNet-B6 and 3.7x speedup on A100 GPU.

3.3. Main Results

Cityscapes. Table 5 reports the comparison between EfficientViT and SOTA semantic segmentation models on Cityscapes. EfficientViT achieves remarkable efficiency improvements over prior SOTA semantic segmentation models without sacrificing performances. Specifically, compared with SegFormer, EfficientViT obtains up to 13x MACs saving and up to 8.8x latency reduction with higher mIoU. Compared with SegNeXt, EfficientViT provides up to 2.0x MACs reduction and 3.8x speedup on GPU while maintaining higher mIoU.

Having similar computational cost, EfficientViT yields significant performance gains over previous SOTA models. For example, EfficientViT-B3 yields +4.5 mIoU gain over SegFormer-B1 with lower MACs.

In addition to the quantitative results, we visualize EfficientViT and the baseline models qualitatively on

Cityscapes. The results are shown in Figure 7. We can find that EfficientViT can better recognize boundaries and small objects than the baseline models while achieving lower latency on GPU.

ADE20K. Table 6 summarizes the comparison between EfficientViT and SOTA semantic segmentation models on ADE20K. Similar to Cityscapes, we can see that EfficientViT also achieves significant efficiency improvements on ADE20K. For example, with +0.6 mIoU gain, EfficientViT-B1 provides 5.2x MACs reduction and up to 3.5x GPU latency reduction than SegFormer-B1. With +1.6 mIoU gain, EfficientViT-B2 requires 1.8x fewer computational cost and runs 2.4x faster on Jetson AGX Orin GPU than SegNeXt-S.

Super-Resolution. Table 7 presents the comparison of EfficientViT with SOTA ViT-based SR methods (SwinIR [31] and Restormer [56]) and SOTA CNN-based SR methods (VapSR [60] and BSRN [30]). EfficientViT provides a better latency-performance trade-off than all compared methods.

On lightweight SR, EfficientViT provides up to 0.09dB gain in PSNR on BSD100 while maintaining the same or lower GPU latency compared with SOTA CNN-based SR methods. Compared with SOTA ViT-based SR methods, EfficientViT provides up to 5.4x speedup on GPU and maintains the same PSNR on BSD100.

On high-resolution SR, the advantage of EfficientViT over previous ViT-based SR methods becomes more significant. Compared with Restormer, EfficientViT achieves up to 6.4x speedup on GPU and provides 0.11dB gain in PSNR on FFHQ.

Table 7: **Comparison with SOTA super-resolution models.** Regarding the latency-performance trade-off, EfficientViT outperforms previous SOTA models by a significant margin, providing up to 6.4× A100 latency reduction while providing higher PSNR and SSIM on FFHQ compared with Restormer.

| Model | FFHQ (512x512 → 1024x1024) | | | | BSD100 (160x240 → 320x480) | | | |
|---------------------------|----------------------------|---------------|---------------|-------------|----------------------------|---------------|--------------|-------------|
| | PSNR ↑ | SSIM ↑ | A100(bs1) ↓ | Speedup ↑ | PSNR ↑ | SSIM ↑ | A100(bs1) ↓ | Speedup ↑ |
| Restormer [56] | 43.43 | 0.9806 | 92.0ms | 1x | 32.31 | 0.9021 | 15.1ms | 1x |
| SwinIR [31] | 43.49 | 0.9807 | 61.2ms | 1.5x | 32.31 | 0.9012 | 9.7ms | 1.6x |
| VapSR [60] | - | - | - | - | 32.27 | 0.9011 | 4.8ms | 3.1x |
| BSRN [30] | - | - | - | - | 32.24 | 0.9006 | 4.5ms | 3.4x |
| EfficientViT w0.75 | 43.54 | 0.9809 | 14.3ms | 6.4x | 32.31 | 0.9016 | 2.8ms | 5.4x |
| EfficientViT | 43.58 | 0.9810 | 17.8ms | 5.2x | 32.33 | 0.9019 | 3.2ms | 4.7x |

4. Related Work

High-Resolution Dense Prediction. Dense prediction targets producing predictions 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. Extensive studies have been done to improve the performance of CNN-based high-resolution dense prediction models [2, 41, 53, 58, 54, 47].

In addition, there are also some works targeting improving the efficiency of high-resolution dense prediction models [57, 40, 29, 52]. While these models provide good efficiency, their performances are far behind SOTA high-resolution dense prediction models.

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 [44, 25, 6, 19] when targeting the low-computation region. To close the gap, MobileViT [38] proposes to combine the strength of CNN and ViT by replacing local processing in convolutions with global processing using transformers. MobileFormer [9] proposes to parallelize MobileNet and Transformer with a two-way bridge in between for feature fusing. NASViT [17] 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 high-resolution dense prediction.

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 [21, 23, 32], quantization [20], efficient model architecture design [26, 36], and training techniques [24, 5]. In addition to manual designs, many recent works use AutoML techniques [61, 4, 7] to automatically design [6], prune [22] and quantize [49] neural networks.

5. Conclusion

In this work, we studied efficient architecture design for high-resolution dense prediction. 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 diverse hardware devices without performance loss than SOTA high-resolution dense prediction models. For future work, we will explore applying EfficientViT to other vision tasks and further scaling up our EfficientViT models.

Acknowledgments

We thank MIT-IBM Watson AI Lab, MIT AI Hardware Program, Amazon and MIT Science Hub, Qualcomm Innovation Fellowship, National Science Foundation for supporting this research.

References

- [1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017. 6
- [2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 1, 9
- [3] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, and Judy Hoffman. Hydra attention: Efficient at-

- tention with many heads. In *Computer Vision—ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 35–49. Springer, 2023. 3
- [4] Han Cai, Tianyao Chen, Weinan Zhang, Yong Yu, and Jun Wang. Efficient architecture search by network transformation. In *AAAI*, 2018. 9
- [5] Han Cai, Chuang Gan, Ji Lin, and Song Han. Network augmentation for tiny deep learning. *arXiv preprint arXiv:2110.08890*, 2021. 9
- [6] 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. 9
- [7] Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct neural architecture search on target task and hardware. In *ICLR*, 2019. 9
- [8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 7
- [9] 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. 9
- [10] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 8
- [11] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34, 2021. 8
- [12] Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020. 3
- [13] 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. 5
- [14] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977, 2021. 6
- [15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 7
- [16] 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
- [17] Chengyue Gong, Dilin Wang, Meng Li, Xinlei Chen, Zhicheng Yan, Yuandong Tian, qiang liu, and Vikas Chandra. NASVit: Neural architecture search for efficient vision transformers with gradient conflict aware supernet training. In *International Conference on Learning Representations*, 2022. 9
- [18] 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 Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 1, 2, 7, 8
- [19] 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. 9
- [20] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR*, 2016. 9
- [21] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *NeurIPS*, 2015. 9
- [22] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. Amc: Automl for model compression and acceleration on mobile devices. In *ECCV*, 2018. 9
- [23] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *ICCV*, 2017. 9
- [24] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 9
- [25] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019. 9
- [26] 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. 9
- [27] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 7
- [28] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 1, 3, 4
- [29] 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. 9
- [30] Zheyuan Li, Yingqi Liu, Xiangyu Chen, Haoming Cai, Jinjin Gu, Yu Qiao, and Chao Dong. Blueprint separable residual

- network for efficient image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 833–843, 2022. 8, 9
- [31] Jingyun Liang, Jie Zhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 7, 8, 9
- [32] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *ICCV*, 2017. 9
- [33] 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
- [34] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022. 6
- [35] 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
- [36] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, 2018. 9
- [37] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001. 6
- [38] Sachin Mehta and Mohammad Rastegari. Mobilevit: Lightweight, general-purpose, and mobile-friendly vision transformer. In *International Conference on Learning Representations*, 2022. 9
- [39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 7
- [40] Rudra PK Poudel, Stephan Liwicki, and Roberto Cipolla. Fast-scnn: Fast semantic segmentation network. *arXiv preprint arXiv:1902.04502*, 2019. 9
- [41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 1, 9
- [42] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 5
- [43] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3531–3539, 2021. 3
- [44] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 6, 9
- [45] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021. 6
- [46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3, 4
- [47] 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, 9
- [48] 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
- [49] Tianzhe Wang, Kuan Wang, Han Cai, Ji Lin, Zhijian Liu, Hanrui Wang, Yujun Lin, and Song Han. Apq: Joint search for network architecture, pruning and quantization policy. In *CVPR*, 2020. 9
- [50] Yihan Wang, Muyang Li, Han Cai, Wei-Ming Chen, and Song Han. Lite pose: Efficient architecture design for 2d human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022. 1
- [51] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 2, 7, 8
- [52] 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. 9
- [53] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 1, 9
- [54] 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, 9
- [55] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution vision transformer for dense predict. *Advances in Neural Information Processing Systems*, 34, 2021. 2, 7, 8
- [56] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference*

- on *Computer Vision and Pattern Recognition*, pages 5728–5739, 2022. 8, 9
- [57] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnets for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 405–420, 2018. 9
- [58] 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, 9
- [59] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 5
- [60] Lin Zhou, Haoming Cai, Jinjin Gu, Zheyuan Li, Yingqi Liu, Xiangyu Chen, Yu Qiao, and Chao Dong. Efficient image super-resolution using vast-receptive-field attention. *arXiv preprint arXiv:2210.05960*, 2022. 8, 9
- [61] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017. 9