EDGEFORMER: IMPROVING LIGHT-WEIGHT CONVNETS BY LEARNING FROM VISION TRANSFORMERS

Haokui Zhang,
Intellifusion

Wenze Hu,
Intellifusion

Xiaoyu Wang
The Chinese University of Hong Kong (Shenzhen)

ABSTRACT

Recently, vision transformers started to show impressive results which outperform large convolution based models significantly. However, in the area of small models for mobile or resource constrained devices, ConvNet still has its own advantages in both performance and model complexity. We propose Edge-Former, a pure ConvNet based backbone model that further strengthens these advantages by fusing the merits of vision transformers into ConvNets. Specifically, we propose global circular convolution (GCC) with position embeddings, a light-weight convolution op which boasts a global receptive field while producing location sensitive features as in local convolutions. We combine the GCCs and squeeze-exictation ops to form a meta-former like model block, which further has the attention mechanism like transformers. The aforementioned block can be used in plug-and-play manner to replace relevant blocks in ConvNets or transformers. Experiment results show that the proposed EdgeFormer achieves better performance than popular light-weight ConvNets and vision transformer based models in common vision tasks and datasets, while having fewer parameters and faster inference speed. For classification on ImageNet-1k, EdgeFormer achieves 78.6% top-1 accuracy with about 5.0 million parameters, saving 11% parameters and 13% computational cost but gaining 0.2% higher accuracy and 23% faster inference speed (on ARM based Rockchip RK3288) compared with MobileViT, and uses only $0.5\times$ parameters but gaining 2.7% accuracy compared with DeIT. On MS-COCO object detection and PASCAL VOC segmentation tasks, EdgeFormer also shows better performance. Code is available at https://github.com/hkzhang91/EdgeFormer

1 INTRODUCTION

Recently, various vision transformers (ViTs) models have achieved remarkable results in many vision tasks, forming strong alternatives to convolutional neural networks (ConvNets) Dosovitskiy et al. (2020) Touvron et al. (2021) Liu et al. (2021).

However, we believe both ViTs and ConvNets are indispensable for the following reasons: 1) From application perspective, both ViTs and ConvNets have their advantages and disadvantages. ViT models generally have better performance but usually suffer from high computational cost and are difficult to train Touvron et al. (2021). Compared with ViTs, ConvNets may show inferior performance, but they still have some unique advantages. For instance, ConvNets have better hardware support and are easy to train. In addition, as is summarized in Guo et al. (2021) and our experiments, ConvNets still dominate in the area of small models for mobile or edge devices. 2) From the information processing perspective, both ViTs and ConvNets have unique features. ViTs are good at extracting global information and use attention mechanism to extract information from different locations driven by input data Chen et al. (2021) Mehta & Rastegari (2022). ConvNets focus on modeling local relationships and have strong prior by inductive bias Dai et al. (2021). The above analysis naturally raise a question: *can we learn from ViTs to improve ConvNets for mobile or edge computing applications?*.

In this paper, we aim to design new light-weight pure ConvNets that further enhance its strength in the area of mobile and edge computing friendly models. Pure convolution is more mobile friendly because convolutions are highly optimized by existing tool chains that are widely used to deploy

model into these resource constrained devices. Even more, because of the huge popularity of ConvNets in the past few years, some existing neural network accelerators are designed mainly around convolution style operations, and the complex non-linear operations such as softmax and data bus bandwidth demanding large matrix multiplications are not efficiently supported. These hardware and software constraints make a pure convolutional light-weight model more preferable even if a ViT based model is equally competitive in other aspects.

To design such a ConvNet, we compare ConvNets with ViTs and summarize three main differences between them: 1) ViTs are good at extracting global features Chen et al. (2021) Mehta & Rastegari (2022) Dai et al. (2021); 2) ViTs adopt Meta-former block Yu et al. (2021); 3) Information aggregations in ViTs are data driven. Corresponding to these three points, we design our EdgeFormer block. 1) We propose the global circular convolution (GCC) to extract global features; 2) Based on the proposed GCC, we build a pure ConvNet Meta-former block as the basic outer structure; 3) We add channel wise attention module to the feature forward network (FFN) part of meta-former, which makes our proposed EdgeFormer adapt kernel weights according inputs. Finally, inspired by CoatNet Dai et al. (2021) and MobileViT Mehta & Rastegari (2022), we use a bifurcate structure (section 3.2) as the outer frame to build a complete network EdgeFormer.

Experiment results show that the proposed EdgeFormer achieves solid performance on three popular vision tasks, including image classification, object detection and semantic segmentation. Taking experiment results of image classification as an example, EdgeFormer achieves 78.6% top-1 accuracy with about 5.0 million parameters, saving 11% parameters and 13% computational cost but gaining 0.2% higher accuracy and 23% inference speed (on Rockchip RK3288) compared with MobileViT Mehta & Rastegari (2022). For experiments of object detection and semantic segmentation, compared with other light-weight models, the proposed EdgeFormer achieves higher mAP and mIOU, while having fewer parameters.

Our main contributions are summarized as follows:

- We propose EdgeFormer, a pure ConvNet for mobile and edge computing applications. The proposed EdgeFormer inherits advantages of ConvNets and ViTs. To our knowledge, this is the first attempt that combines strengths of ConvNets and ViTs to design a light-weight ConvNet.
- To overcome the restriction that traditional convolutions have limited perception fields, we propose global circular convolution (GCC), where base-instance kernel and position embedding strategies are used to handle input size variations and inject location information to output feature maps respectively. We jointly use the proposed GCC and conventional convolution operations to extract local-global features, which brings higher accuracy.
- We apply the proposed EdgeFormer on three vision tasks. Compared with the baseline model, the proposed EdgeFormer achieves better performance on all three tasks, while having fewer parameters, lower computational cost and higher inference speed.

2 RELATED WORK

2.1 VISION TRANSFORMERS

Vaswani et al. firstly proposed transformer Vaswani et al. (2017) for natural language processing (NLP) tasks. Compared with recurrent neural network (RNN) models, transformer has much higher computational efficiency and it is good at capturing relationship from any pair of elements in the input sequence. As a result, transformers replaced RNNs and dominate the NLP field.

In 2020, Dosovitskiy et al. introduced transformer into vision tasks and proposed vision transformer (ViT) Dosovitskiy et al. (2020), where each image is cropped into a sequence of patches to meet the input requirement of transformer and position embedding is adopted to ensure the model is sensitive to position information of the input patches. With pre-training on huge datasets such as JFT-300M Sun et al. (2017), ViT achieves impressive performance on various vision tasks. However, the original ViT model has some restrictions, for instance, it is heavy-weight, having low computational efficiency and hard to train. Subsequent variants of ViTs are proposed to overcome these problems. From the point of improving training strategy, Touvron et al. Touvron et al. (2021) proposed to use knowledge distillation to train ViT models, and achieved competitive accuracy with less pre-training

data. To further improve the model architecture, some researchers attempted to optimize ViTs by learning from ConvNets. Among them, PVT Wang et al. (2021) and CVT Wu et al. (2021) insert convolutional operations into each stage of ViT model to reduce the number of tokens, and build hierarchical multi-stage structures. Swin transformer Liu et al. (2021) computes self attention within shifted local windows. PiT Heo et al. (2021) jointly use pooling layer and depth wise convolution layer to achieve channel multiplication and spatial reduction. These papers clearly show that some techniques of ConvNets can be applied on vision transformers to design better vision transformer models.

2.2 HYBRID STRUCTURES COMBINING CONVNET AND VISION TRANSFORMERS

The models reviewed in section 2.1 focus on integrating elements of ConvNets into ViTs. Another popular line of research is combining elements of ViTs and ConvNets to design new backbones. Graham et al. mixed ConvNet and transformer in their LeVit model, which significantly outperforms previous ConvNet and ViT models with respect to the speed/accuracy tradeoff Graham et al. (2021). BoTNet Srinivas et al. (2021) replaces the standard convolution with multi-head attention in the last several blocks of ResNet. ViT-C Xiao et al. (2021) adds early convolutional stem to vanilla ViT. ConViT d'Ascoli et al. (2021) incorporates soft convolutional inductive biases via a gated positional self-attention. The CMT Guo et al. (2021) block consists of depth wise convolution based local perception unit and a light-weight transformer module. CoatNet Dai et al. (2021) merges convolution and self-attention to design a new transformer module, which focuses on both local and global information. After comprehensive comparison, we find that these hybrid models simultaneously employed similar structure, that is using convolutional stem to extract local features in the beginning stages and transformer style models later to extract global or local-global features. We choose a similar structure when designing our pure convolutional model.

2.3 LIGHT-WEIGHT CONVNETS AND VITS

Since 2017, light-weight ConvNets attract much attentions as more and more applications needs to run ConvNet models on mobile devices. Now, there are a lot of light-weight ConvNets, such as ShuffleNets Ma et al. (2018) Ma et al. (2018), MobileNets Howard et al. (2017) Sandler et al. (2018) Howard et al. (2019), MicroNet Li et al. (2021), GhostNet Han et al. (2020), EfficientNet Tan & Le (2019), TinyNet Chen et al. (2019) and MnasNet Tan et al. (2019). Compared with standard ConvNets, light-weight ConvNets have fewer parameters, lower computational cost and faster inference speed. In addition, light-weight ConvNets can be applied on a wide range of devices. Despite these benefits, these light-weight models have inferior performance compared with heavy-weight models.

Very recently, following the research line of combining strengths of ConvNet and ViT, some researcher attempted to build light-weight hybrid models for mobile vision tasks. Mobile-Former presents a parallel design of MobileNet and transformer, which leverages the advantages of MobileNet at extracting local features and transformer at capturing global information Chen et al. (2021). Mehta and Rastegari proposed MobileViT, where the upper stages of MobileNetv2 Sandler et al. (2018) are replaced with MobileViT block Mehta & Rastegari (2022). In MobileViT block, local representations extracted by convolution and global representations are concatenated to generate local-global representations.

In terms of purpose, our proposed EdgeFormer is related to Mobile-Former and MobileViT. Different from these two models which still keep transformer blocks, our proposed EdgeFormer is pure ConvNet, which makes our proposed EdgeFormer more mobile friendly. Our experiments of deploying models on low power platform confirm this point. In terms of designing a pure ConvNet via learning from ViTs, our work is most closely related to a parallel work ConvNext Liu et al. (2022). The two major differences are: 1) Ideas and architectures are different. The ConvNext modernizes a standard ResNet toward the design of a vision transformer by introducing a series (more than ten) of incremented but effective designs. Our proposed EdgeFormer starts from three main differences between ConvNets and ViTs and fills the gaps from macro level. As the ideas are different, the corresponding structures are also different; 2) They are proposed for different purposes. Our EdgeFormer is proposed for mobile devices. Compared with ConvNext, the proposed EdgeFormer shows
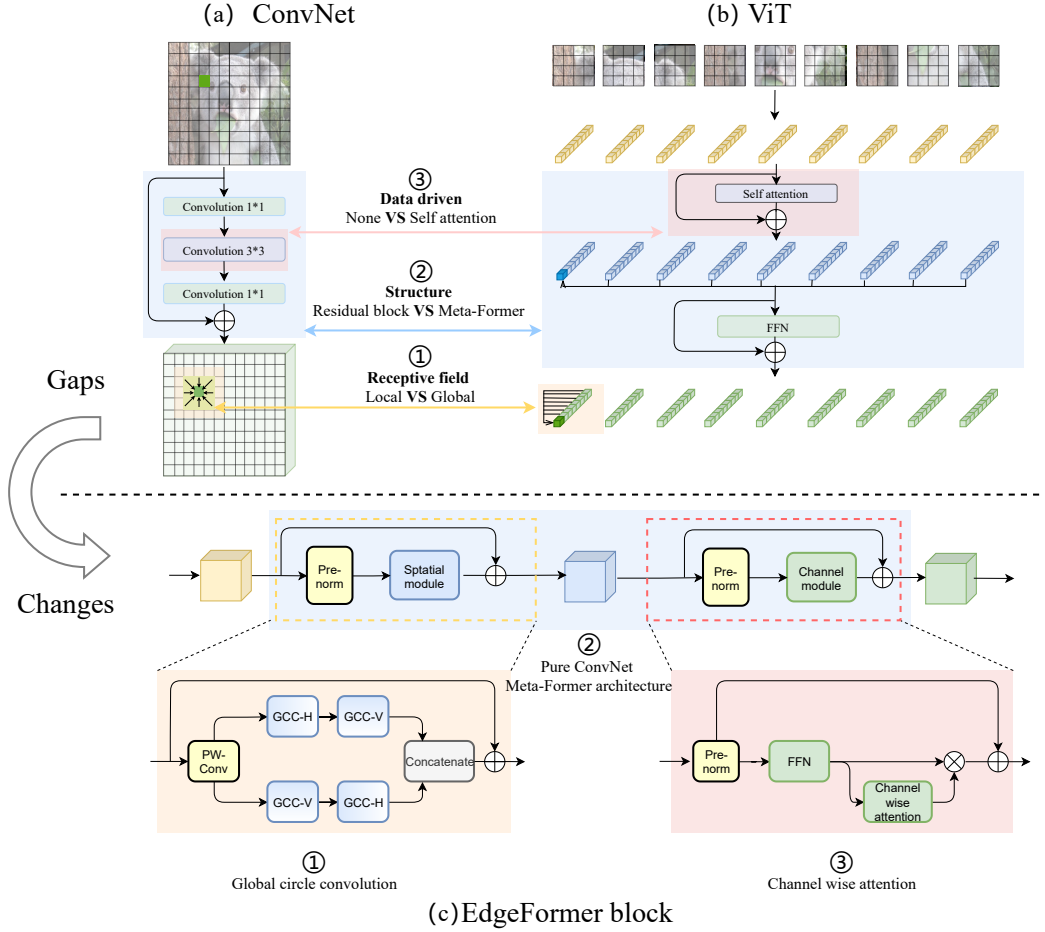
Figure 1: EdgeFormer block. (a) A residual block that is widely used in ConvNets; (b) A ViT block; (c) An EdgeFormer block

advantages when constraining models as light-weight models. Corresponding experiment results are listed in 4.5.

3 THE PROPOSED METHOD

In this section, we will introduce our EdgeFormer network in two parts, the details of the building block (EdgeFormer block) and the overall model structure (EdgeFormer).

3.1 EDGEFORMER BLOCK

The upper part of Figure 1 shows three major differences between common ConvNets and ViTs. The bottom half of Figure 1 illustrates the architecture of our proposed EdgeFormer block. In the following, we will explain the motivation and the specific structure of each component of the proposed EdgeFormer block.

Extracting global features with global circular convolution. In ConvNets, feature is calculated as $y_i = \sum_{j \in \mathcal{L}(i)} w_{i-j} x_j$, where x_i, y_i are the input and output at position i respectively, and $\mathcal{L}(i)$ denotes a local neighborhood of i. In ViTs, self-attention modules extracts features based on formula $y_i = \sum_{j \in \mathcal{G}} \frac{e^{(x_i^T x_j)}}{\sum_{k \in \mathcal{G}} e^{(x_i^T x_k)}} x_j$, where $\mathcal{G}$ means the global spatial space. Comparing these two
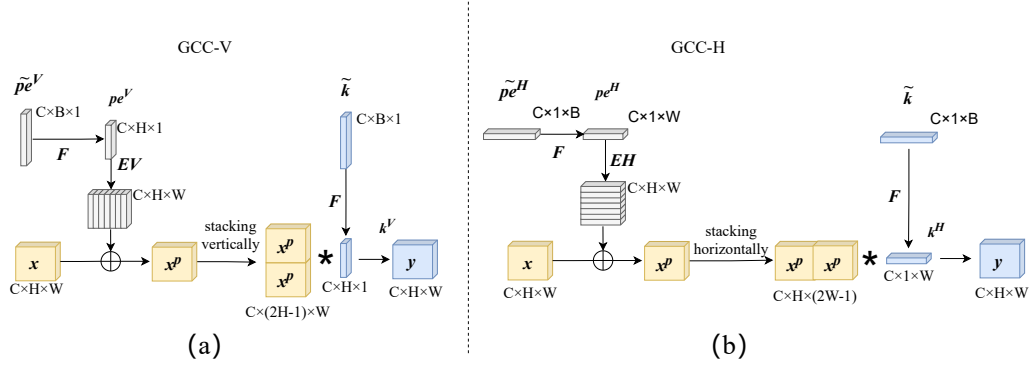
Figure 2: Illustration of the global circular convolution. (a) GCC-V; (b) GCC-H. F, EV and EH are explained in equations 1 and 2

formulas, we can see that self attention learns global features from the entire spatial locations but convolution gathers information from a local receptive field.

To overcome this issue, we propose the global circular convolution (GCC). As shown in Figure 2, our proposed GCC has two types, one is GCC of vertical direction (GCC-V) and the other one is GCC of horizontal direction (GCC-H). The receptive field of the GCC-V and GCC-H covers all pixel in the same column and the same row, respectively. Jointly using GCC-V and GCC-H can extract global features from all input pixels. For notational simplicity, we assume the input x has only one channel and the corresponding shape is $1 \times h \times w$. The output of GCC-V at location (i, j) is computed with:

$$
\begin{aligned}
pe^V &= F(\widetilde{pe}^V) = [pe_0^V, pe_1^V, \cdots, pe_{h-1}^V]^T \\
pe_e^V &= EV(pe^V, w) \\
k^V &= F(\widetilde{k}) = [k_0^V, k_1^V, \cdots, k_{h-1}^V] \\
x^p &= x + pe_e^V \\
y_{i,j} &= \sum_{t \in (0, h-1)} k_t^V x_{((i+t) \bmod h, j)}^p
\end{aligned}
\tag{1}
$$

where, pe^V is instance position embedding and it is generated from a base embedding $\widetilde{pe}^V$ via bilinear interpolation function $F()$. Here $F()$ is used to adapt the size of position embedding to the size of input features. pe_e^V is expanded position embedding. k^V is instance kernel. $EV()$ is an expand function of vertical direction. After copying the input vector w times, $EV()$ concatenates these copied vectors along horizontal direction to generate a $h \times w$-sized position embedding matrix. Similarly, the output of GCC-H at location (i, j) can be expressed as:

$$
\begin{aligned}
x^p &= x + pe_e^H \\
y_{i,j} &= \sum_{t \in (0, w-1)} k_t^H x_{(i, (j+t) \bmod w)}^p
\end{aligned}
\tag{2}
$$

where $pe_e^H = EH(pe^H, h)$ and $EH()$ is an expand function. $EH()$ expands input vector along the vertical direction. Implementing the GCC in modern deep learning libraries is straightforward. Taking the most complicated part $y_{i,j} = \sum_{t \in (0, w-1)} k_t^H x_{(i, (j+t) \bmod w)}^p$ as an example, it can be implemented with one line of code: $y = F.conv2D(torch.cat(x^p, x^p, dim = 3), k^H)$. Figure 3 illustrates the computational process in the case that the input is an one dimensional vector. From Figure 3, we can see that GCC-H perform convolutions along a circle generated by connecting the start and the end of the input. So, we name the proposed convolution as the global circular convolution.

Compared with conventional convolutions, the proposed global circular convolution introduces three modifications:
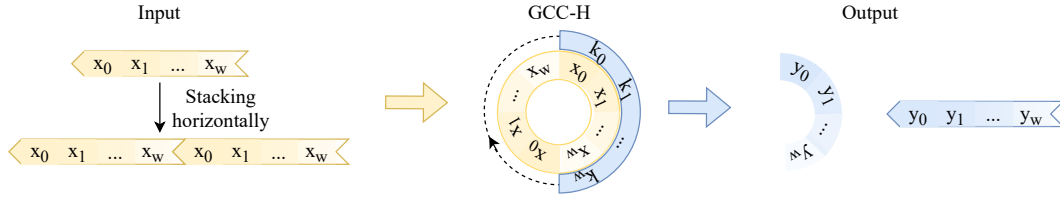
Figure 3: Illustration of global circular convolution on horizontal direction.
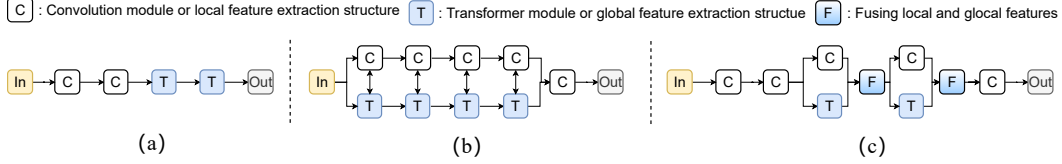

Figure 4: Three main hybrid structures. (a) serial structure; (b) parallel structure; (c) bifurcate structure

- The receptive field is increased to global spatial space. Note that, increasing the kernel size of tradition local convolution to full input size does not extract global features. In local convolution, zero padding is usually used to keep the size of convolutional feature the same with that of the input. Even if we increase the kernel size to global size, the global kernel only covers part pixels coming from input. Especially for extracting feature in edge portion, only about half of pixels that covered by global kernel are from input actual input, while others are simply zeros.

- The position embedding is used to keep the output feature sensitive to spatial location. Global circular convolution can extract global features, but it disturbed the spatial structure of the original input. For classification, keeping spatial structure may not be a big issue. But, as is shown in ablation study, for location sensitive tasks such as segmentation and detection, keeping spatial structure does matter. Here, following the design in ViTs, we introduce position embedding to keep spatial structure. Experiment results in ablation study show that position embedding is useful in segmentation and detection tasks

- The kernel and position embedding are dynamically generated according to the input size. In GCC, the sizes of kernels and position embedding codes must be consistent with that of instance inputs. To handle the case that inputs have different spatial resolution, we generate instance kernels and position embedding codes via interpolation functions.

Designing EdgeFormer block with GCC. From ConvNets to ViTs, a considerable modification is meta-former block replaced residual block (the blue two-way arrow). A Meta-former block generally consists of a sequence of two components: a token mixer and a channel mixer. The token mixer is for exchanging information among tokens in different spatial locations. The channel mixer is for mixing information among different channels. Both two components use residual learning structure.

Inspired by this, we insert GCC into Meta-former like block to build our EdgeFormer block. Specifically, we replace self-attention module with the proposed GCC to build an new spatial module to replace token mixer part. Here, we do this for two main reasons: 1) GCC can extract global features and interacts information among pixels from global space, which meets the requirement of token mixer module; 2) the computation complexity of self attention module is quadratic. Replacing this part with GCC can reduce computational cost significantly, which is helps achieving our goal of designing a light-weight ConvNet. Based on the proposed GCC, we build a pure ConvNet meta-former block.

Adding channel wise attention in channel mixer part. In ViTs, self attention module can adapt weights according input, which makes ViTs data driven models. By adopting attention mechanism, data driven models can focus on important features and suppress unnecessary ones, which brings

6

better performance. Previous literature Hu et al. (2018)Woo et al. (2018)Jaderberg et al. (2015) already explained the importance of keep model data driven.

By replacing the self-attention with the proposed global circular convolution, we get a pure ConvNet which can extract global features. But the replaced model is no longer a data driven model. To compensate, we insert channel wise attention module into channel mixer part, as shown in Figure 1(c). Following SENet Hu et al. (2018), we first aggregate spatial information of input features $x \in \mathbb{R}^{c \times h \times w}$ via global average pooling and get aggregated feature $x_a \in \mathbb{R}^{c \times 1 \times 1}$; Then we feed x_a into a multi-layer perceptron to generate channel wise weight $a \in \mathbb{R}^{c \times 1 \times 1}$. The a is multiplied with x channel wise to generate the final output.

3.2 EDGEFORMER

In section 3.1, we have presented the EdgeFormer block, which is a basic block and can be inserted into most of the current existing models. In this section, we select an outer frame for it and build the complete network EdgeFormer.

Currently, as shown in Figure 4, existing hybrid structures can be basically divided into three main structures, including serial structure (Figure 4(a)) Graham et al. (2021)Xiao et al. (2021), parallel structure (Figure 4(b)) Chen et al. (2021) and bifurcate structure (Figure 4(c)) Mehta & Rastegari (2022)Dai et al. (2021). Among all three structures, the third one achieves best performance for now. At present, bifurcate model CoatNet achieves the highest classification accuracy on Imagenet-1k. Aiming to deploy model in mobile devices, Mehta and Rastegari proposed MobileViT, which also adopts the third structure.

Inspired by this, we adopt bifurcate structure as our outer frame and build our final outer frame based on MobileViT. Specifically, taking the outer frame adopted in MobileViT as baseline, we further make some improvements:

- MobileViT consists of two major types of modules, ViT and MobileNetV2 blocks. We keep all MobileNetV2 blocks and replacing ViT blocks with corresponding EdgeFormer blocks. This replacement converts the model from hybrid structure to pure ConvNet.

- We appropriately increase the widths of EdgeFormer blocks. Even so, the replaced model still has fewer parameters and less computational cost.

- As show in Figure 4(c), the bifurcate structure contains some interaction modules, which are in charge of interacting information between local and global feature modules. In the original MobileViT, ViT blocks are the most heavy modules. After replacing ViT blocks with EdgeFormer blocks, the cost of these interaction modules becomes prominent. So, we introduce group convolution and point wise convolution into these modules, which decreases number of parameters without hurting performance.

4 EXPERIMENT RESULTS

In experiments, we show the overall advantages of the proposed EdgeFormer on three typical vision tasks, and then conduct detailed study to show the value of our design choices, the model scaling characteristics, and its speed advantage on low power devices.

4.1 IMAGE CLASSIFICATION

We conduct image classification experiments on ImageNet-1k, the most widely used benchmark dataset for this task. We train the proposed EdgeFormer models on the training set of ImageNet-1K, and report top-1 accuracy on the validation set.

Training setting. As we adopt MobileViT like structure as our outer framework, we train our model using a very similar training strategy as well. To be specific, we train each model for 300 epochs on 8 V100 or A100 GPUs with AdamW optimizer Loshchilov & Hutter (2019), where the maximum learning rate, minimum learning rate, weight decay and batchsize are set to 0.004, 0.0004, 0.025 and 1024 respectively. Optimizer momentum β_1 and β_2 of the AdamW optimizer are set to 0.9 and 0.999 respectively. We use the first 3000 iterations as warm up stage. We adjust learning rate

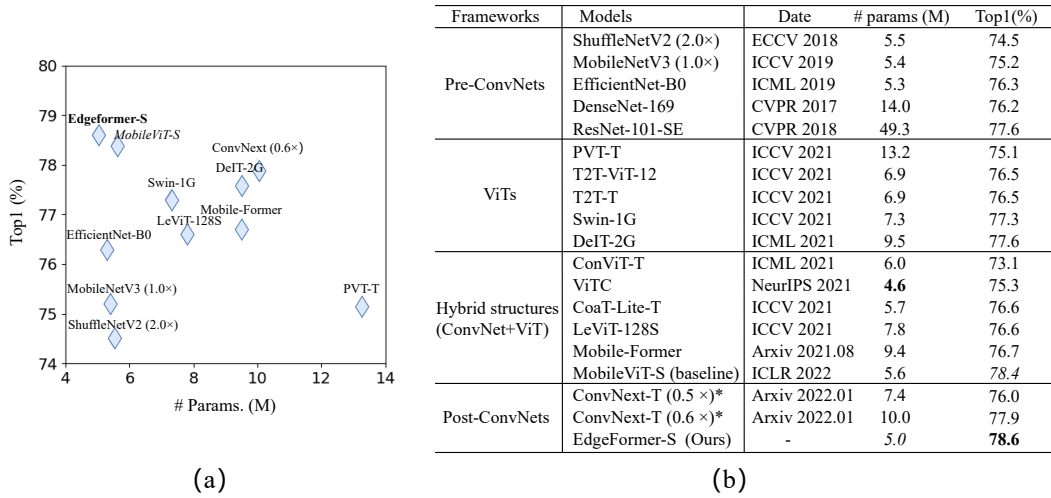| Frameworks	Models	Date	# params (M)	Top1(%)
Pre-ConvNets	ShuffleNetV2 (2.0×)	ECCV 2018	5.5	74.5
	MobileNetV3 (1.0×)	ICCV 2019	5.4	75.2
	EfficientNet-B0	ICML 2019	5.3	76.3
	DenseNet-169	CVPR 2017	14.0	76.2
	ResNet-101-SE	CVPR 2018	49.3	77.6
ViTs	PVT-T	ICCV 2021	13.2	75.1
	T2T-ViT-12	ICCV 2021	6.9	76.5
	T2T-T	ICCV 2021	6.9	76.5
	Swin-1G	ICCV 2021	7.3	77.3
	DeIT-2G	ICML 2021	9.5	77.6
Hybrid structures (ConvNet+ViT)	ConViT-T	ICML 2021	6.0	73.1
	ViTC	NeurIPS 2021	4.6	75.3
	CoaT-Lite-T	ICCV 2021	5.7	76.6
	LeViT-128S	ICCV 2021	7.8	76.6
	Mobile-Former	Arxiv 2021.08	9.4	76.7
	MobileViT-S (baseline)	ICLR 2022	5.6	*78.4*
Post-ConvNets	ConvNext-T (0.5 ×)*	Arxiv 2022.01	7.4	76.0
	ConvNext-T (0.6 ×)*	Arxiv 2022.01	10.0	77.9
	EdgeFormer-S (Ours)	-	*5.0*	**78.6**

(a)   (b)

Figure 5: Classification experiment results on ImageNet-1K. (a) Accuracy vs model size. Here we only keep part of comparison models for clarity. (b) Comparison of results on image classification. ∗ indicates our implementation. Pre-ConvNets indicate classical ConvNets appeared before ViTs. Post-ConvNets denote ConvNets which integrate merits of ViTs but still keep pure ConvNet structures.

Table 1: Comparisons of light-weight models on ImageNet-1K classification

Frameworks	Models	Date	# params (M)	Top1(%)
Pre-ConvNets	ShuffleNetV2(2.0×)	ECCV 2018	5.5	74.5
	MobileNetV3(1.0×)	ICCV 2019	5.4	75.2
	EfficientNet-B0	ICML 2019	5.3	76.3
ViTs	T2T-ViT-7	ICCV 2021	4.3	71.7
	DeiT-T	ICML 2021	5.7	72.2
Hybrid structures	ViT-C	NeurIPS 2021	4.6	75.3
	CoaT-Lite-T	ICCV 2021	5.7	76.6
	MobileViT-S	ICLR 2022	5.6	78.4
Post-ConvNet	EdgeFormer-S	-	5.0	78.6

following the cosine schedule. For data augmentation, we use random cropping, horizontal flipping and multi-scale sampler. We use label smoothing Szegedy et al. (2016) to regularize the networks and set smoothing factor to 0.1. We use Exponential Moving Average (EMA) Polyak & Juditsky (1992).

Comparison results. The experiment results of image classification are listed in Figure 5. Figure 5 (a) shows that EdgeFormer-S and MobileViT-S beat other model by a clear margin. Figure 5 (b) shows comparison with more models. The proposed EdgeFormer-S achieves highest classification accuracy, and have fewer parameters than most models. Compared with the second best model MobileViT-S, our EdgeFormer-S decreases the number of parameters by 11% and increases the top 1 accuracy by 0.2 percentage points.

Light-weight models. Table 1 shows comparison results among light-weight models, which confirms our ideas and answers the question proposed in introduction.

Firstly, comparing results of light-weight ConvNets with that of ViTs, light-weight ConvNets show much better performance.

Secondly, comparing the popular ConvNets before ViT appears (pre-ConvNets), ViTs and hybrid structures, hybrid structures achieve the best performance. Therefore improving ConvNets by learning from the merits of ViT is feasible.
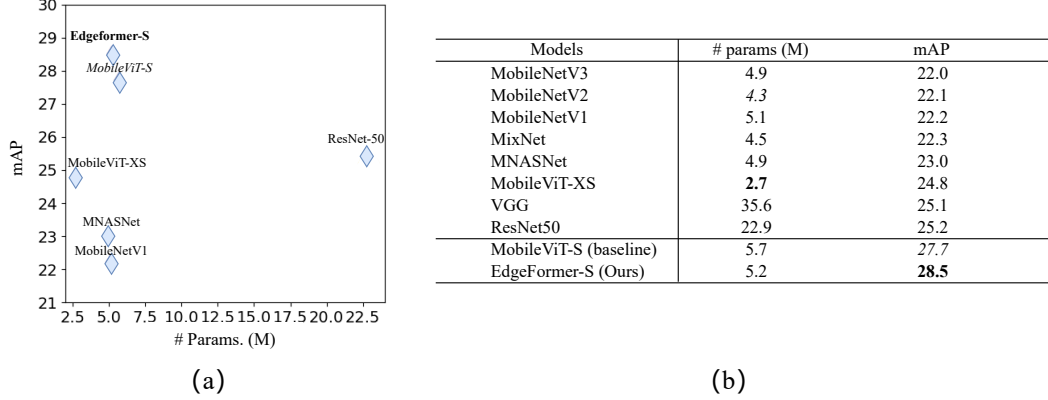
Models	# params (M)	mAP
MobileNetV3	4.9	22.0
MobileNetV2	*4.3*	22.1
MobileNetV1	5.1	22.2
MixNet	4.5	22.3
MNASNet	4.9	23.0
MobileViT-XS	**2.7**	24.8
VGG	35.6	25.1
ResNet50	22.9	25.2
MobileViT-S (baseline)	5.7	*27.7*
EdgeFormer-S (Ours)	5.2	**28.5**

(a) (b)

Figure 6: Object detection results on MS-COCO. (a) mAP vs model size. (b) Comparison results

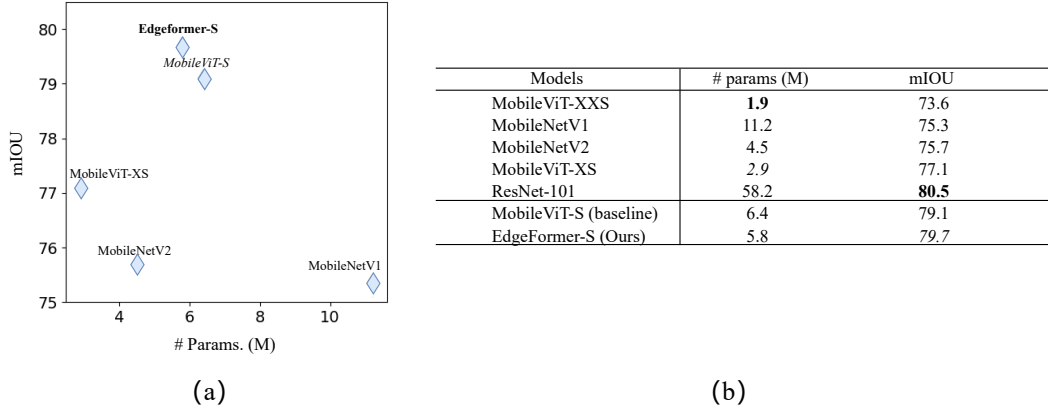| Models	# params (M)	mIOU
MobileViT-XXS	**1.9**	73.6
MobileNetV1	11.2	75.3
MobileNetV2	4.5	75.7
MobileViT-XS	*2.9*	77.1
ResNet-101	58.2	**80.5**
MobileViT-S (baseline)	6.4	79.1
EdgeFormer-S (Ours)	5.8	*79.7*

(a) (b)

Figure 7: Semantic segmentation experiments on PASCAL VOC. (a) mIOU vs model size.(b) Comparison results with more models.

Finally, the proposed EdgeFormer achieves the best performance among all comparison models. So indeed by learning from ViT design, performance of pure light-weight ConvNets can be improved significantly.

4.2 OBJECT DETECTION

We use MS-COCO Lin et al. (2014) datasets and its evaluation protocol for object detection experiments. Following Mehta & Rastegari (2022)Sandler et al. (2018), we take single shot object detection (SSD) Liu et al. (2016) as the detection framework and use separable convolution to replace the standard convolutions in the detection head.

Experiment setting. Taking models pretrained on ImageNet-1K as backbone, we finetune detection models on training set of MS-COCO with AdamW optimizer for 200 epochs. Batchsize and weight decay are set to 128 and 0.01. We use the first 500 iterations as warm up stage, where the learning rate is increased from 0.000001 to 0.0009. Both label smoothing and EMA are used during training.

Comparison results. Figure 6 lists the corresponding results. Similar to results in image classifiction, MobileViT-S and EdgeFormer-S achieve the the second best and the best in terms of mAP. Compared with the second best model, EdgeFormer-S shows advantages in both model size and detection accuracy.

Table 2: Ablation study. BK, MF, CA and PE denote big kernel, meta-former architecture, channel wise attention and position embedding. BK 1/4 and BK 1/2 means the kernel size is set to 1/4 and 1/2 of the input features, respectively.

Row	Task	Kernel	MF	CA	PE	# params (M)	Top1/mAP/mIOU
1	classification	Baseline	-	-	-	5.6	78.35
2	classification	BK 1/4	Y	Y	N	5.0	78.46
3	classification	BK 1/2	Y	Y	N	5.0	78.45
4	classification	GCC	N	Y	Y	5.3	76.00
5	classification	GCC	Y	N	Y	5.0	78.50
6	classification	GCC	Y	Y	N	5.0	78.63
7	classification	GCC	Y	Y	Y	5.0	78.63
8	detection	Baseline	-	-	-	5.7	27.7
9	detection	GCC	Y	Y	N	5.2	27.5
10	detection	GCC	Y	Y	Y	5.2	28.5
11	segmentation	Baseline	-	-	-	6.4	79.1
12	segmentation	GCC	Y	Y	N	5.8	79.2
13	segmentation	GCC	Y	Y	Y	5.8	79.7

4.3 SEMANTIC SEGMENTATION

Experiment settings. DeepLabV3 is adopted as the semantic segmentation framework. We fine tune segmentation models on training set of PASCAL VOC Everingham et al. (2015) and COCO dataset, then evaluate trained models on validation set of PASCAL VOC using mean intersection over union (mIOU) and report the final results for comparison. We fine tune each model for 50 epochs with AdamW.

Comparison results. Results are summarized in Figure 7. We can see that MobileViT-S and EdgeFormer-S have the best trade-off between model scale and mIOU. Compared with ResNet-101, MobileViT-S and EdgeFormer achieve competitive mIOU, while having much fewer parameters.

4.4 ABLATION STUDY

Using the MobileViT as a baseline model, we further conduct ablation analysis on three components proposed in our EdgeFormer network.

- **Global circlular convolution**. The proposed GCC has two major characteristics: 1) GCC brings global receptive field; 2) Position embedding keeps spatial structure information. Experiment results confirm that both characteristics are important. 1) Results in rows 1,2,3 show that, using big kernel can also improve accuracy, but the benefit of big kernel reaches a saturation point when kernel size reaches a certain level. This results are consistent with the statement claimed in Liu et al. (2022). Using GCC can further improve accuracy, as shown in rows 2-3 and 6-7. 2) Introducing position embedding to GCC is necessary. As we explained in section 3.1, using GCC alone can indeed capture global features but it disturbs the original spatial structures. For classification task, position embedding has no impact (rows 6 and 7). However, for detection and segmentation tasks which are sensitive to spatial location, abandoning position embedding hurts performances (rows 9-10 and 12-13).

- **Meta-former architecture**. In experiments of abandoning Meta-former architecture, we integrate GCC with the ResNeXt block Xie et al. (2017) to replace Meta-former architecture. By comparing row 4 and 7, we can see that using the proposed pure ConvNet meta-former architecture is useful.

- **Channel wise attention**. Results in rows 5 and 7 show that using channel wise attention can improves performance. Compared with GCC, channel wise attention brings less benefit.

In summary, all three components are useful. Connecting them as a whole achieves best performance.

Table 3: Model scaling experiments.

Tiny			Small		
Models	# params. (M)	Top1	Models	# params. (M)	Top1
T2T-ViT-7	4.3	71.7	ConvNext-T (0.5×)	7.4	76.0
DeiT-T	5.7	72.2	ConvNext-T (0.6×)	10.0	77.9
MobileViT-XS	2.3	74.8	MobileViT-S	5.6	78.4
EdgeFormer-XS	**2.1**	**75.0**	EdgeFormer-S	**5.0**	**78.6**

Table 4: Inference speed comparison with baseline model on RK3288

Models	# params. (M)	Madds (M)	inference speed (ms)	Top1
MobileViT-S	5.6	2010	457	78.4
EdgeFormer-S	5.0 (-11%)	1740 (-13%)	353 (+23%)	78.6

4.5   MODEL SCALING EXPERIMENTS

In this section, we build two EdgeFormers of different sizes (Tiny and Small) to evaluate the scalability. Experiment results are listed in Table 3. It that EdgeFormer-XS and EdgeFormer-S achieves best performance while having fewer parameters than their counterparts. Even compared with the most recent released work ConvNext, the proposed EdgeFormer still show some advantages. Part of such a result is ConvNext and EdgeFormer are proposed for different purposes. The ConvNext is proposed for conventional equipment, but our EdgeFormer is designed for edge computing devices, as we explained in section 2.3.

4.6   INFERENCE SPEED ON LOW POWER DEVICES.

As we mentioned in introduction, the EdgeFormer is proposed for edge computing devices. To verify whether the proposed EdgeFormer meets our requirements, we deploy the proposed EdgeFormer on a widely used low power chip Rockchip RK3288 and compare it with baseline. We use ONNX Bai et al. (2019) and MNNJiang et al. (2020) to port these models to RK3288 and time each model for 100 iterations to measure the average inference speed.

Comparison results are listed in Table 4. Compared with baseline, EdgeFormer is 23% faster than baseline on Rockchip RK3288. Besides less Madds operations, we believe this speed improvement is also brought by two factors: 1) Convolutions are highly optimized by existing tool chains that are widely used to deploy models into these resource constrained devices; 2) Compared with convolutions, transformers are more data bandwith demanding as computing the attention map involves two large matrices K and Q, whereas in convolutions the kernel is a rather small matrix compared with the input feature map. In case the bandwith requirement exceeds that of the chip design, the CPU will be left idle waiting for data, resulting in lower CPU utilization and overall slower inference speed.

5   CONCLUSION

In this paper, for edge computing devices, we present EdgeFormer, a pure ConvNet, which inherits advantages of ConvNet and integrated structure characteristics of ViT. To evaluate the performances, we apply the proposed model on three popular vision tasks, image classification, object detection and semantic segmentation. The proposed model achieves better performance on all three tasks, while having fewer parameters compared with other ConvNet, ViT and hybrid models. In addition, to verify that the proposed model suit edge devices, we deploy the proposed model on Rockchip RK3288. Experiment results show that the proposed EdgeFormer does inherit ConvNets and it is well supported by edge computing devices.

The proposed EdgeFormer block is an plug-and-play block, it can be inserted other models. In our future work, we will integrate the proposed EdgeFormer block with more existing models to evaluate its performance and deploy it on more edge computing devices to test is hardware support.

REFERENCES

Junjie Bai, Fang Lu, Ke Zhang, et al. Onnx: Open neural network exchange. `https://github.com/onnx/onnx`, 2019.

Gonglong Chen, Yihui Wang, Huikang Li, and Wei Dong. Tinynet: a lightweight, modular, and unified network architecture for the internet of things. In *Proceedings of the ACM SIGCOMM 2019 conference posters and demos*, pp. 9–11, 2019.

Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. *arXiv preprint arXiv:2108.05895*, 2021.

Zihang Dai, Hanxiao Liu, Quoc Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34, 2021.

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

Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, pp. 2286–2296. PMLR, 2021.

M. Everingham, Sma Eslami, L Van Gool, Cki Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111 (1):98–136, 2015.

Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12259–12269, 2021.

Jianyuan Guo, Kai Han, Han Wu, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. Cmt: Convolutional neural networks meet vision transformers. *arXiv preprint arXiv:2107.06263*, 2021.

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*, pp. 1580–1589, 2020.

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

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1314–1324, 2019.

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.

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

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

Xiaotang Jiang, Huan Wang, Yiliu Chen, Ziqi Wu, Lichuan Wang, Bin Zou, Yafeng Yang, Zongyang Cui, Yu Cai, Tianhang Yu, et al. Mnn: A universal and efficient inference engine. *Proceedings of Machine Learning and Systems*, 2:1–13, 2020.

Yunsheng Li, Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Lu Yuan, Zicheng Liu, Lei Zhang, and Nuno Vasconcelos. Micronet: Improving image recognition with extremely low flops. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 468–477, 2021.

T. Y. Lin, M. Maire, S. Belongie, J. Hays, and C. L. Zitnick. Microsoft coco: Common objects in context. *Springer International Publishing*, 2014.

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. *Springer, Cham*, 2016.

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*, pp. 10012–10022, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019.

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)*, pp. 116–131, 2018.

Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *ICLR*, 2022.

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

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*, pp. 4510–4520, 2018.

Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16519–16529, 2021.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pp. 843–852, 2017.

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

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

Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2820–2828, 2019.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

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*, pp. 568–578, 2021.

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

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22–31, 2021.

Tete Xiao, Piotr Dollar, Mannat Singh, Eric Mintun, Trevor Darrell, and Ross Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems*, 34, 2021.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

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. *arXiv preprint arXiv:2111.11418*, 2021.