

# FalconNet: Factorization for the Light-weight ConvNets

Zhicheng Cai  
Nanjing University

caizc@smail.nju.edu.cn

Qiu Shen  
Nanjing University

shenqiu@nju.edu.cn

## Abstract

Designing light-weight CNN models with little parameters and Flops is a prominent research concern. However, three significant issues persist in the current light-weight CNNs: i) the lack of architectural consistency leads to redundancy and hindered capacity comparison, as well as the ambiguity in causation between architectural choices and performance enhancement; ii) the utilization of a single-branch depth-wise convolution compromises the model representational capacity; iii) the depth-wise convolutions account for large proportions of parameters and Flops, while lacking efficient method to make them light-weight. To address these issues, we factorize the four vital components of light-weight CNNs from coarse to fine and redesign them: i) we design a light-weight overall architecture termed Light-Net, which obtains better performance by simply implementing the basic blocks of other light-weight CNNs; ii) we abstract a Meta Light Block, which consists of spatial operator and channel operator and uniformly describes current basic blocks; iii) we raise RepSO which constructs multiple spatial operator branches to enhance the representational ability; iv) we raise the concept of receptive range, guided by which we raise RefCO to sparsely factorize the channel operator. Based on above four vital components, we raise a novel light-weight CNN model termed as FalconNet. Experimental results validate that FalconNet can achieve higher accuracy with lower number of parameters and Flops compared to existing light-weight CNNs.

## 1. Introduction

Convolutional Neural Networks (CNNs) possess the capability of representing high-dimensional complex functions and have been successfully applied to various real visual scenarios [20, 24, 29, 46]. With lower computational complexity and higher efficiency than ViTs, CNNs remain the dominance in computer vision applications [12, 40, 61]. For the implementation on mobile devices for real-world applications, the computational and storage resources are always limited, requiring light-weight CNN models with re-

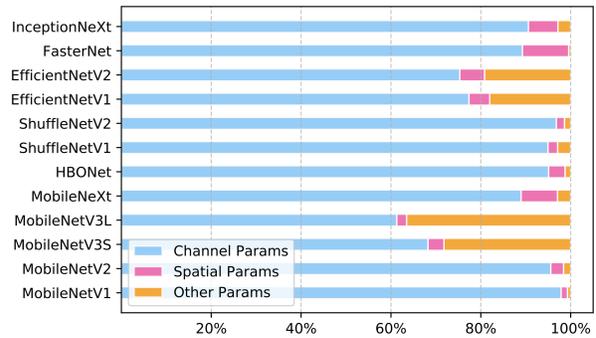


Figure 1. The parameter proportions of channel parameters, spatial parameters and other parameters (ignore the classifier head).

duced parameters and low Flops while maintaining competitive performance. Depth-wise separable convolution (DS-Conv), proposed in MobileNetV1 [22], factorizes the regular convolution into depth-wise convolution (DW-Conv) and point-wise convolution (PW-Conv), which extracts the spatial and channel features individually. Consequently, DS-Conv decreases a large amount of computation and parameters and has been a fundamental design component for subsequent light-weight CNNs [21, 31, 45, 49–51, 64, 69] and modern large CNNs [12, 40, 61]. MobileNetV2 [45] proposes the inverted residual block (IRB) to alleviate the destruction to the features and enhance representational capacity. Utilizing the paradigm of IRB as the basic block, many light-weight models with different overall architectures (stages, width, depth) [17, 21, 49–51, 57] are raised. In addition to IRB, many efficient basic blocks [2, 31, 52, 64, 69] with different structures are designed to improve the representational ability of light-weight CNNs. However, the architectural inconsistencies cause redundant structures that could be consolidated through unification, and the use of varying basic blocks along with varying architectures leading to unfair capacity comparisons and obscuring the causal relationship between architectural choices and performance enhancements.

Moreover, while some works [35, 37, 48, 53, 61] factorize the DW Conv into parallel low-rank branches to save computational cost, the main stem for light-weight mod-

els chasing for even lower Flops and parameters lies in the PW Conv. As Fig. 1 exhibited, the PW Conv (broadly, the densely-connected linear layers processing the channel information) accounts for the majority of the light-weight model parameters, while the DW Conv (broadly, conv layers processing the spatial information) only makes up a small proportion. ShuffleNet splits channels into groups and shuffles channels [35, 41, 67], however, the information communication between channels in different groups is insufficient. IGC [47, 58, 66] raises interleaved group convolutions to replace the regular convolution with multiple permuted group convolution layers, while the structure becomes complicated. ChannelNet [6] introduces sparse connections to PW-Conv, while one output channel can only attends to a small fraction of the input channels, thus is only applied to the last PW-Conv layer due to the inadequate information communication.

This paper want to raise a novel light-weight CNN model with small amount of parameters while maintaining competitive performance. In light of the aforementioned issues, we factorize the four vital components of constructing light-weight CNNs, namely, overall architecture, meta basic block, spatial operator and channel operator. To be specific:

- We first design a light-weight overall architecture termed as *LightNet*, which refers to the structural designs of modern CNNs and has four stages, each stage is stacked with basic blocks. Better performance can be obtained by simply implementing the basic blocks of other light-weight CNNs on LightNet.
- We abstract and analyze a *Meta Basic Block*, consisting of spatial operator and channel operator (specifically, PW-Conv), for light-weight CNN model design. The paradigm of meta basic block uniformly describes the current basic blocks, e.g., IRB in MobileNets [21, 45] and EfficientNets [17, 49–51, 57], sandglass block [69] and FasterNet block [2], inferring that the framework of the meta block provides the basic ability to the model, while the differences of model performances essentially come from different structural instantiations [64]. Through extensive experiments we further simplify meta basic block into *Meta Light Block*, which obtains better performance.
- We introduce *RepSO* as the spatial operator for the Meta Light block. According to the guidance of weight magnitude, RepSO constructs multiple extra branches to compensate for the reduce of learnable parameters and enhance the model representational capacity. RepSO further utilizes the structural reparameterization methodology to equivalently convert these diverse branches into a single branch in inference.
- We introduce the concept of *Receptive Range* for

channel dimension correspondence to the concept of receptive field for spatial dimension. Receptive range elaborates the way of connection between the output and input neurons in the PW-Conv, which claims that one output neuron should attend to all of the input neurons directly or indirectly (first attend to a set of hidden neurons which attend to all the input neurons) to obtain the full receptive range. Based on the concept of receptive range we further raise *RefCO* which factorizes the PW-Conv in the Meta Light block through introducing sparsity to the dense channel connection correspondence to the spatial convolution. Moreover, RefCO utilizes structural reparameterization while construct multiple *sparsely factorized PW-Convs* to compensate for the reduction of channel connections.

- Finally, we raise a novel light-weight CNN model termed as *FalconNet* based on above four vital components. Experimental results show that FalconNet can achieve higher accuracy with less parameters and Flops compared to existing light-weight CNNs.

## 2. Related Works

### 2.1. Light-weight CNN Models

In order to deploy on mobile devices for real-world applications, many light-weight CNN models with reduced parameter amounts and limited computational burdens are proposed. SqueezeNet [27] raises fire module which partially replaces the  $3 \times 3$  convolution kernels with  $1 \times 1$  kernels. InceptionV3 [48] factorizes the standard convolution into two asymmetric convolutions. IGC [47, 58, 66] raises interleaved group convolutions to decompose the regular convolution into multi-layer group convolutions. ShuffleNetV1 [67] and MicroNet [35] utilizes  $1 \times 1$  group convolutions and then shuffles the grouped channels, while ShuffleNetV2 [41] firstly splits the channels then shuffles the channels. ChannelNet [6] introduces sparse connections to the dense layers. MobileNetV1 [22] and Xception [5] propose the depth-wise separable convolution to decouple the regular convolution into depth-wise convolution and point-wise convolution, which alleviates a large amount of computation and parameters and has been a widely-adopted design element for modern efficient CNN models [12, 40, 61]. MobileNetV2 [45] introduces the inverted residual block. MobileNetV3 [21] enhances MobileNetV2 with squeeze-and-excitation module [23] and neural architecture search [36, 71, 72]. MobileNeXt [69] introduces sandglass block to alleviate information loss by flipping the structure of inverted residual block. HBONet [31] raises harmonious bottleneck on two orthogonal dimensions to improve representation. EfficientNet [49–51, 57] proposes a compound scaling method to scale depth, width and resolu-

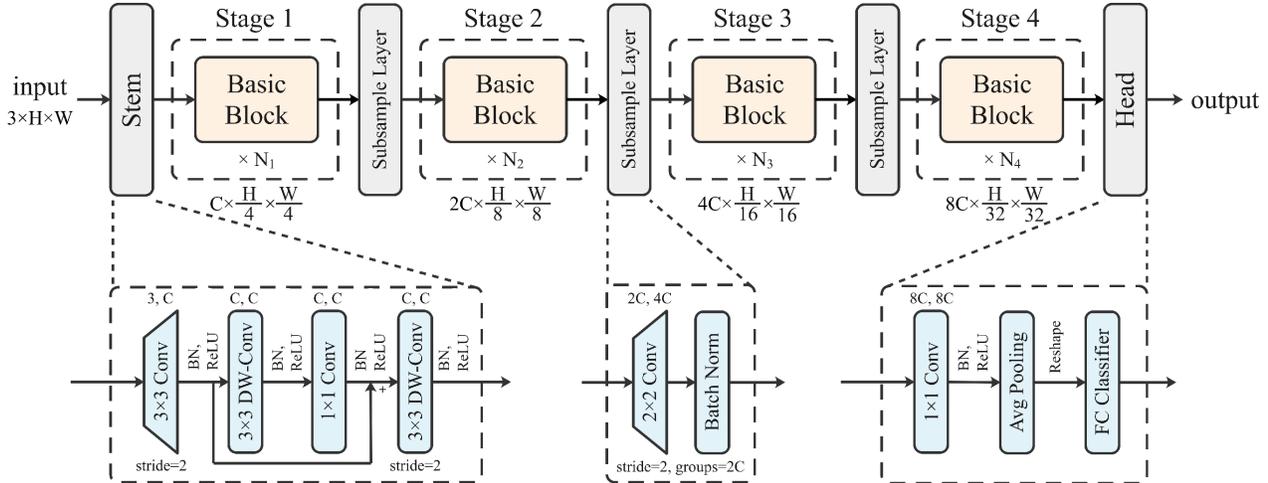


Figure 2. LightNet overall architecture.

tion uniformly. EMO [64] additionally introduces the self-attention to the inverted residual block. FasterNet [2] raises partial convolution to conduct regular convolution on part of the channels to reduce Flops while increasing FLOPS.

## 2.2. Structural Reparameterization

Structural Reparameterization [7, 9–13] is a representative reparameterization methodology to parameterize a structure with the parameters transformed from another structure. Typically, it adds extra branches to the model in training to enhance the representational capacity and improve the performance, then equivalently simplifies the training structure into the same as the original model for inference, without any extra computational or memory cost. ACNet [9] asymmetrically constructs two extra vertical and horizontal convolution branches in training and converts them into the original branch in inference. RepVGG [13] constructs identity shortcuts parallel to the  $3 \times 3$  convolution during training and converts the shortcuts into the  $3 \times 3$  branches. DBB [11] constructs inception-like diverse branches of different scales and complexities to enrich the feature space. RepLKNet [12] adds a relatively small kernel into the large kernel to capture small-scale patterns. MobileOne [54] also multiplies the convolution branch in training and reparameterizes them in inference for performance improvement.

## 3. Method

### 3.1. Design the Overall Architecture

Firstly, although certain light-weight CNNs utilize similar basic building blocks, their overall architectures differ, resulting in unnecessary redundancy that could be consolidated through unification. Secondly, certain light-weight CNNs incorporate distinct basic building blocks, yet they

still exhibit varying overall architectures. This leads to an unfair comparison of capacity between the basic blocks and obscures the causal relationship between architectural choices and performance improvements. Hence we first intend to construct an overall architecture especially for light-weight CNN models. Referring to the modern architectures of powerful CNN [2, 12, 40, 61] and ViT [14, 38, 39] models, we raise *LightNet* of which architecture is sketched in Fig. 2.

**Stem.** Stem refers to the beginning part of the model. Instead of utilizing single conv layer with relatively large stride, we desire to capture more details by several conv layers at the beginning. After the first  $3 \times 3$  regular conv layer with a stride of 2, we employ a  $3 \times 3$  DW-Conv layer to capture low-level patterns, followed by a  $1 \times 1$  PW-Conv layer and another  $3 \times 3$  DW-Conv layer for subsampling [12]. There also exists a shortcut as shown in Fig. 2.

**Stages.** Stages 1-4 are composed of several repeated *basic blocks*, such as IRB. According to the stage compute ratio of 1:1:3:1 [39, 40], the number of blocks in each stage is set as [3, 3, 9, 3]. Following the pyramid principle [20] and considering reducing the parameters, the channel dimension in each stage is set as [32, 64, 128, 256].

**Subsampling Layers.** We add separate subsampling layers between stages. We use  $2 \times 2$  conv layer with groups of input channel dimension and a stride of 2 for halving the spatial resolution. The  $2 \times 2$  conv layer also doubles the channel dimension. A batch normalization layer is arranged subsequently to stabilize training.

**Head.** Head, following the 4-th stage, is the last part of the model. We first use a  $1 \times 1$  conv layer to further mix the information, then utilize the global average pooling to obtain the feature vectors, which is subsequently input into the last fully-connected classifier and obtain the final output.

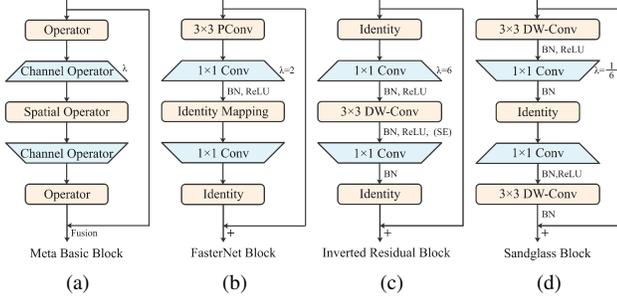


Figure 3. Abstracted unified Meta Basic Block for light-weight models. Some correspondingly instantiated basic blocks (e.g., IRB and sandglass block) are also selected to exhibit.

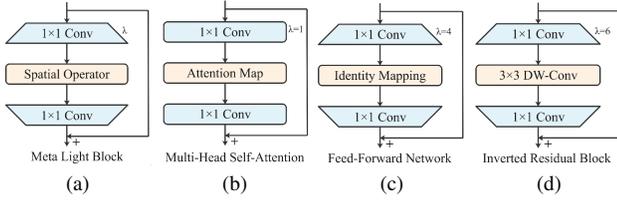
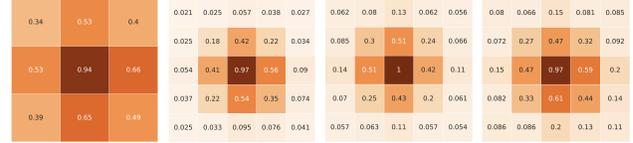


Figure 4. Simplified abstracted Meta Light Block for efficient light-weight models. Some correspondingly instantiated basic blocks (e.g., IRB and FFN) are also selected to exhibit.

Experimental results show that simply implementing the basic blocks of other light-weight CNNs on LightNet achieves better performance than the original models.

### 3.2. Explore the Meta Basic Block

Basic blocks are the pivotal component for light-weight CNNs. As exhibited in Fig. 3, different basic blocks can be generally abstracted into the *Meta Basic Block* (Fig. 3a), which is alternately composed of spatial and channel operators (i.e., PW-Conv). The framework of the meta basic block provides the basic ability to the model, which means instantiating these spatial operators as non-learnable identity mappings can still achieve effective performance, while the differences of model performances essentially come from different structural instantiations of the meta basic block, e.g., FasterNet block instantiates the first spatial operator as PConv and instantiates the other two spatial operators as identity mapping, as shown in Fig. 3a. Through conducting extensive experiments (Sec. 4.3.1), it is observed that the first and the last operator layers make no benefit to the enhancement of model performance, while the second spatial operator layer between two channel operator layers is significant. Thus we further simplify meta basic block into *Meta Light Block* (Fig. 4a), which consists of two PW-Conv layers (with an expansion ratio  $\lambda$ ) and a single spatial operator layer in between.



(a) MobileNetV2 (b) MobileNetV3 (c) EfficientNet (d) MNASNet

Figure 5. The average kernel magnitude matrices of MobileNetV2, MobileNetV3, EfficientNet and MNASNet trained on ImageNet.

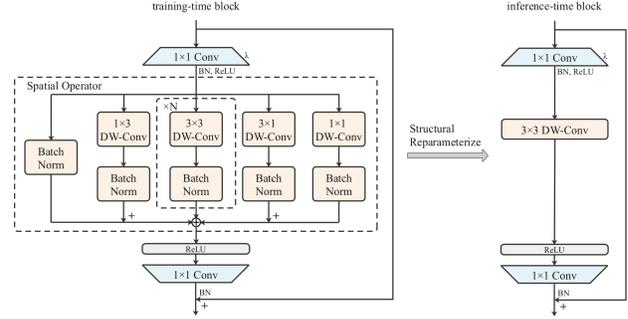


Figure 6. Meta Light Block with RepSO

### 3.3. Strengthen the Spatial Operator

Though the Meta Light Block guarantees the fundamental performance of the model with certain overall architecture, powerful spatial operator can significantly enhance the model representational capacity. Thus we construct multiple branches of versatile spatial operators to enrich the representation space. It is assumed that a position of the kernel tends to be more significant if it has a larger average kernel magnitude [8, 9, 16, 18, 19]. We first calculate and visualize the average kernel magnitude matrices of four popular light-weight CNNs as shown in Fig. 5. It is observed that the positions of the outermost circle in the  $5 \times 5$  kernel have negligible importance compared to the central  $3 \times 3$  positions, thus we utilize  $3 \times 3$  convolution kernels which also have fewer parameters and Flops. To compensate for the reduced feature channels, we construct  $N$  ( $N=3$  by default) parallel  $3 \times 3$  DW-Conv branches. Moreover, Fig. 5 shows that the positions in the skeleton pattern of the  $3 \times 3$  kernel account for much importance compared to these corner positions, thus we additionally construct horizontal  $1 \times 3$  and vertical  $3 \times 1$  DW-Conv branches. In addition, it is observed that the central position always possesses the highest importance score (almost 1), thus we construct an extra  $1 \times 1$  DW-Conv branch to further enhance the central position. Last but not least, as the meta block provides the fundamental capacity, we also add an identity mapping branch to the spatial operator layer. The obtained multi-branch operator is termed as *RepSO (Reparameterized Spatial Operator)*, as sketched in Fig. 6 left, all these seven branches are individually equipped with a batch normalization layer, then the normalized outputs of each branch are element-wise added.

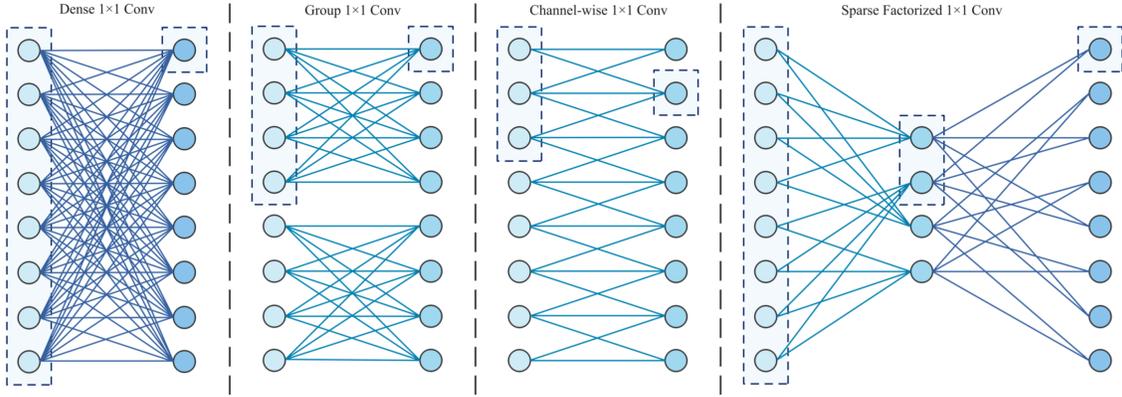


Figure 7. Connections and corresponding receptive ranges of different  $1 \times 1$  Convs

According to the additivity and homogeneity of convolution and the methodology of structural reparameterization [9], these seven branches can be equivalently converted to a single  $3 \times 3$  DW-Conv branch in inference as sketched in Fig. 6 right, which produces no extra inference cost.

### 3.4. Factorize the Channel Operator

#### 3.4.1 Receptive Range

For that  $1 \times 1$  Conv accounts for large proportion of model parameters and Flops, we want to make it light-weight. Essentially, we can change the connections between input and output neurons to change the amount of parameters. Hence we propose the concept of **Receptive Range** as the guideline for establishing these connections. The concept of receptive range is introduced specifically for channel dimension and analogous to the receptive field for spatial dimension. The value of the receptive range of a certain output neuron is the number of input neurons that it attend to directly (by a weight) or indirectly (attend to a set of hidden neurons which attend to the input neurons). Fig. 7 exhibits the connections and corresponding receptive ranges of different  $1 \times 1$  Convs. Suppose  $C$  is the number of input neurons (channel numbers), as observed, each output neuron of dense  $1 \times 1$  Conv can attend to all the input neurons directly, thus the receptive range is  $C$ . For group  $1 \times 1$  Conv with  $g$  groups, the receptive range is  $\frac{C}{g}$ , thus different groups can establish information connections, causing significant reduce of representation capacity. For channel-wise  $1 \times 1$  Conv [6] with window size of  $k$ , the receptive range is  $k$ , thus each output neuron can only attend to small amount of input neurons, leading to insufficient channel information aggregating. Consequently, to make the channel information aggregated sufficiently thus guarantee the model representation capacity, each output neuron should have a full receptive range, namely, each output neuron should be connected to all of the input neurons directly or indirectly.

#### 3.4.2 Sparsely Factorized $1 \times 1$ Conv

With the guidance of full receptive range, we try to introduce sparsity to the densely connected  $1 \times 1$  conv emulating the spatial convolution. We raise the **Sparsely Factorized  $1 \times 1$  Conv (SF-Conv for short)**, which is proposed to be a new paradigm of channel sparse connectivity. As Fig. 8 exhibits, SF-Conv factorizes a densely connected  $1 \times 1$  conv into two sparsely connected  $1 \times 1$  convs, namely, 1st and 2nd SF-Convs, to guarantee the full receptive range. Given input feature map of tensor  $X \in \mathbb{R}^{H \times W \times C_{in}}$ , where  $H, W$  is the spatial size, and  $C_{in}$  is the input channel numbers. The output feature map is  $\hat{X} \in \mathbb{R}^{H \times W \times C_{out}}$ , where  $C_{out} = \lambda C_{in} = \lambda C$ . For certain operation, we only consider the number of parameters since  $Flops = H \times W \times Params$  where  $H, W$  are fixed. For a SF-Conv, we can consider  $X_{i,j,:} \in \mathbb{R}^{C \times 1 \times 1}$  for each pixel  $(i, j)$  in  $X$  as an actual input feature map with a spatial size of  $C_{in} \times 1$  and 1 channel, thus we can conduct standard convolution on it, which introduces sparsity connection. The 1st SF-Conv has a channel reduction coefficient  $R$  (hyper-parameter,  $R = 2$  by default) to control the neuron number of the hidden feature map  $X_h$  thus control the parameters. Thus there are  $\frac{C}{R}$  neurons in  $X_h$ . Suppose a convolution kernel  $\hat{W}_1$  with spatial size of  $K \times 1$ . Consider the case of single channel, sliding  $K \times 1$  on the input  $C \times 1$  with stride  $S$  generates  $\frac{C-K+S}{S}$  output neurons. Thus to generate  $\frac{C}{R}$  hidden neurons, the convolution kernel should have  $\frac{C}{R} / \frac{C-K+S}{S}$  channels, that is  $W_1 \in \mathbb{R}^{(\frac{C}{R} / \frac{C-K+S}{S}) \times 1 \times K \times 1}$ , and the feature map of the hidden neurons is  $X_h \in \mathbb{R}^{\frac{C-K+S}{S} \times 1 \times (\frac{C}{R} / \frac{C-K+S}{S})}$ . To achieve the minimal connections while obtain the maximum receptive range, the stride  $S$  it to be  $S = K$ . Moreover, to enhance the representational capacity and increase the degree of freedom, we make the kernel weights unshared in the spatial dimension, that is, each set of input neurons is operated by a individual set of weights. As a consequence, the weights of 1st SF-Conv is  $W_1 \in \mathbb{R}^{\frac{C}{R} \times \frac{C}{K} \times K \times 1}$ , and the output of 1st SF-Conv is  $X_h \in \mathbb{R}^{\frac{C}{K} \times 1 \times \frac{C}{R}}$ . Thus 1st SF-

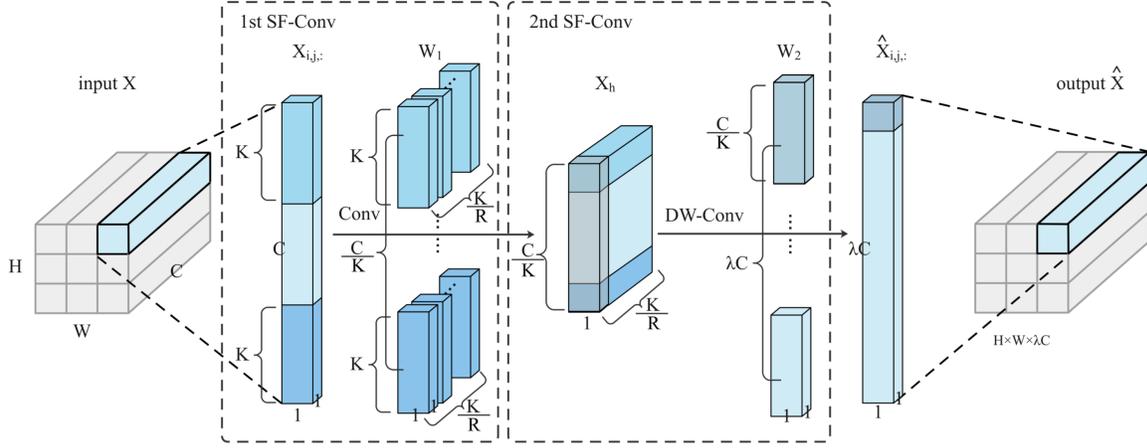


Figure 8. Sparse Factorized  $1 \times 1$  Convolution.

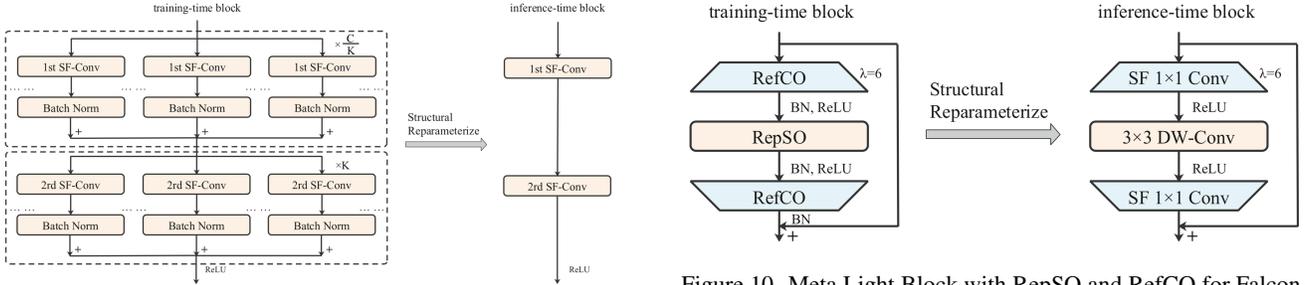


Figure 9. RefCO Channel Operator

Conv has  $\frac{K}{R} \times \frac{C}{K} \times K \times 1 = \frac{CK}{R}$  parameters.

For each output neuron of 1st SF-Conv has a receptive range of  $K$ , while for the set of neurons in a certain channel of  $X_h$  has a total receptive range of  $C$ . Thus through attending to the set of neurons in a certain channel of  $X_h$ , the output neuron of SF-Conv can have a full receptive range with minimal number of parameters. Thus the 2nd SF-Conv takes  $X_h \in \mathbb{R}^{\frac{C}{K} \times 1 \times \frac{K}{R}}$  as the input and conducts a DW-Conv on it, thus the weights of 2nd SF-Conv  $W_2$  has a spatial kernel size of  $\frac{C}{K} \times 1$ . To obtain the  $\lambda C$  output channels,  $W_2$  should have a width multiplier of  $\lambda C / \frac{C}{K}$ , thus  $W_2 \in \mathbb{R}^{\lambda C \times 1 \times \frac{C}{K} \times 1}$  with  $\frac{\lambda C^2}{K}$  parameters. The output feature map of the 2nd SF-Conv is  $\hat{X}_{i,j,:} \in \mathbb{R}^{\lambda C \times 1 \times 1}$  for the pixel  $(i, j)$  in  $\hat{X}$ . Through operating the shared SF-Conv in all spatial positions of  $X$ , we can obtain the  $\hat{X}$ .

SF-Conv has a total parameters of  $\frac{CK}{R} + \frac{\lambda C^2}{K}$ . Given a certain channel reduction ratio  $R$ , to achieve the least parameters, the kernel size  $K$  is set as  $K = \sqrt{\lambda CR}$ , which is dynamically adjusted according to  $C$ ,  $\lambda$  and  $R$ . The total number of parameters of SF-Conv becomes  $2C\sqrt{\frac{\lambda C}{R}}$ . For a dense  $1 \times 1$  conv with weights  $W_d \in \mathbb{R}^{\lambda C \times C \times 1 \times 1}$  has  $\lambda C^2$  parameters. Therefore, SF-Conv exhibits a parameter count that is only  $\frac{2}{\sqrt{\lambda CR}}$  of the parameter count of PW-

Conv. In this way, SF-Conv can introduce sparsity to the channel connections, as well as maintain the full receptive range (as shown in Fig. 7), thus reducing the number of parameters and Flops while obtain a competitive representation capacity.

### 3.4.3 RefCO

We further raise **RefCO** as the **Reparameterized factorized Channel Operator**. RefCO also employs the structural reparameterization methodology to compensate for the reduced parameter count and enhance representations. As Fig. 9 exhibits, in training, RefCO firstly constructs  $\frac{C}{R}$  parallel 1st SF-Conv branches and added the outputs, then constructs  $K$  parallel 2nd SF-Conv branches and added the  $K$  output feature maps to obtain the final output. In inference, these parallel 1st/2nd SF-Conv branches can be equivalently converted into a single SF-Conv branch.

### 3.5. FalconNet

Based on above four vital components, namely, LightNet overall architecture, Meta Light Block, RepSO spatial operator and RefCO channel operator, we obtain a novel light-weight CNN model termed as **FalconNet** (**F**actorization for the **l**ight-weight **convNet**). Fig. 9 exhibits the Meta Light

Figure 10. Meta Light Block with RepSO and RefCO for FalconNet

Table 1. Results of various light-weight CNN models and LightNets with corresponding basic blocks on CIFAR-10, CIFAR-100 and Tiny-ImageNet-200. Width multiplier is used for some LightNets to compensate for the largely reduced channels of certain basic blocks, such as Sandglass and ShuffleNet blocks. The numbers of parameters and Flops in inference are also exhibited.

Basic Block	Model	CIFAR-10	CIFAR-100	Tiny-ImageNet-200	Params	Flops
DSC Block	MobileNetV1	93.18%	72.47%	64.36%	4.23M	588.91M
	<b>LightNet</b> ×3.5	93.22%	74.26%	64.38%	3.59M	536.06M
Residual DSC Block	ResNet-18	93.89%	74.52%	65.02%	11.72M	1844.08M
	ResNet-34	93.95%	74.82%	65.21%	21.84M	3698.78M
	<b>LightNet</b> ×3.5	93.70%	75.01%	64.66%	3.59M	536.06M
ShuffleNetV1 Block	ShuffleNetV1	91.35%	68.51%	56.86%	1.81M	138.75M
	<b>LightNet</b> ×4.0	91.61%	69.42%	58.94%	1.05M	132.86M
ShuffleNetV2 Block	ShuffleNetV2	92.31%	70.08%	60.38%	2.28M	154.87M
	<b>LightNet</b> ×3.5	93.50%	73.56%	64.10%	2.01M	219.65M
Inverted Residual Block	MobileNetV2	93.43%	74.03%	66.30%	3.56M	353.01M
	MobileNetV3S	91.89%	70.50%	60.18%	2.94M	66.89M
	MobileNetV3L	94.07%	73.56%	65.54%	5.48M	238.85M
	EfficientNetV1-B0	94.21%	75.68%	66.62%	4.98M	404.42M
	EfficientNetV2-S	93.85%	75.42%	67.18%	21.14M	2915.26M
	<b>LightNet</b>	94.84%	76.92%	68.18%	3.32M	526.51M
Sandglass Block	MobileNeXt	93.01%	67.42%	58.12%	3.31M	310.04M
	<b>LightNet</b> ×6.0	94.09%	75.43%	65.76%	3.77M	607.85M
HBO Block	HBONet	92.32%	72.39%	64.20%	4.56M	326.99M
	<b>LightNet</b>	92.44%	72.85%	65.92%	3.83M	209.23M
FasterNet Block	FasterNet-T0	92.94%	68.02%	58.32%	3.64M	310.98M
	<b>LightNet</b>	93.45%	74.14%	64.78%	3.36M	509.10M
<b>RepSO Block</b>	<b>LightNet</b>	94.96%	78.24%	69.72%	3.32M	526.51M
<b>RepSO+RefCO Block</b>	<b>FalconNet</b>	94.85%	78.04%	69.46%	2.39M	333.14M

Block with RepSO and RefCO, which is utilized as the basic block for FalconNet. Later experimental results validate that FalconNet can achieve higher accuracy with lower number of parameters and Flops compared to existing light-weight CNNs.

## 4. Experiments

### 4.1. Configurations

We conduct abundant experiments on three challenging benchmark datasets, CIFAR-10, CIFAR-100 and Tiny-ImageNet-200, to validate the effectiveness and superiority of the four vital components illustrated above. CIFAR-10/100 consists of 50K training images and 10K testing images, Tiny-ImageNet-200 contains 100K training images and 10K testing images. For the training configuration, we

use the cross entropy loss function and adopt an SGD optimizer with momentum of 0.9, batch size of 256, and weight decay of  $4 \times 10^{-5}$ , as the common practice [12, 45]. We use a learning rate schedule with a 5-epoch warmup, initial value of 0.1, and cosine annealing for 300 epochs to guarantee the complete convergence. The data augmentation uses random cropping and horizontal flipping. The input resolution is uniformly resized to  $224 \times 224$ . All the models are random initialized with Xavier initialization and trained with the same training configuration from scratch.

### 4.2. Performance Evaluation

We evaluate the performance of various existing light-weight CNN models, including MobileNetV1/V2/V3 [21, 22, 45], MobileNeXt [69], EfficientNetV1/V2 [50, 51], ShuffleNetV1/V2 [41, 67], HBONet [31], and FasterNet [2],

Table 2. Results of different meta basic block instantiations on CIFAR-10, CIFAR-100 and Tiny-ImageNet-200.

Index	1st SO	2nd SO	3rd SO	1st CO	2nd CO	Exp Ratio	CIFAR-10	CIFAR-100
A	Identity	Identity	Identity	PW-Conv	PW-Conv	$\lambda = 6$	90.38%	67.56%
B	DW-Conv	Identity	Identity	PW-Conv	PW-Conv	$\lambda = 4$	91.55%	--
C	DW-Conv	Identity	Identity	PW-Conv	PW-Conv	$\lambda = 6$	92.11%	--
D	Identity	DW-Conv	Identity	PW-Conv	PW-Conv	$\lambda = 4$	94.31%	--
E	<b>Identity</b>	<b>DW-Conv</b>	<b>Identity</b>	<b>PW-Conv</b>	<b>PW-Conv</b>	$\lambda = 6$	<b>94.84%</b>	<b>76.92%</b>
F	DW-Conv	DW-Conv	Identity	PW-Conv	PW-Conv	$\lambda = 6$	93.36%	75.10%
G	DW-Conv	DW-Conv	DW-Conv	PW-Conv	PW-Conv	$\lambda = 6$	91.83%	74.92%
H	P-Conv	Identity	Identity	PW-Conv	PW-Conv	$\lambda = 4$	92.75%	73.58%
I	P-Conv	Identity	Identity	PW-Conv	PW-Conv	$\lambda = 6$	93.45%	74.14%
J	Conv	Identity	Identity	PW-Conv	PW-Conv	$\lambda = 4$	93.58%	73.42%
K	Conv	Identity	Identity	PW-Conv	PW-Conv	$\lambda = 6$	93.67%	73.89%
L	Conv	DW-Conv	Identity	PW-Conv	PW-Conv	$\lambda = 4$	93.71%	74.40%
M	Conv	DW-Conv	Identity	PW-Conv	PW-Conv	$\lambda = 6$	94.10%	75.42%
N	DW-Conv	Identity	Identity	PW-Conv	Identity	$\lambda = 1$	93.70%	74.26%
O	Identity	DW-Conv	Identity	PW-Conv	PW-Conv	$\lambda = 1$	93.16%	75.01%
P	DW-Conv	DW-Conv	Identity	PW-Conv	PW-Conv	$\lambda = 1$	93.29%	74.92%
Q	DW-Conv	Identity	DW-Conv	PW-Conv	PW-Conv	$\lambda = 1/6$	94.09%	75.43%
R	DW-Conv	DW-Conv	DW-Conv	PW-Conv	PW-Conv	$\lambda = 1/6$	92.11%	--
S	PW-Conv	Identity	Identity	DW-Conv	PW-Conv	$\lambda = 6$	92.27%	--
T	PW-Conv	Identity	PW-Conv	DW-Conv	DW-Conv	$\lambda = 6$	91.59%	--

Table 3. Results of Meta Light Block different instantiations of spatial operator on CIFAR-10, CIFAR-100 and Tiny-ImageNet-200. Each row exhibits the branch numbers of DW-Conv with certain kernel size that consist the corresponding spatial operator.

Index	Identity	$1 \times 1$	$1 \times 3$	$3 \times 1$	$3 \times 3$	$1 \times 5$	$5 \times 1$	$3 \times 5$	$5 \times 3$	$5 \times 5$	$7 \times 7$	CIFAR-10	CIFAR-100
A	0	0	0	0	0	0	0	0	0	0	0	90.38%	68.56%
B	0	0	1	1	0	0	0	0	0	0	0	94.86%	76.85%
C	0	0	0	0	1	0	0	0	0	0	0	94.84%	76.92%
D	0	0	0	0	0	1	1	0	0	0	0	94.16%	76.20%
E	0	0	0	0	0	0	0	0	0	1	0	94.31%	76.27%
F	0	0	0	0	0	0	0	0	0	0	1	93.94%	75.21%
G	1	0	0	0	1	0	0	0	0	0	0	94.64%	76.10%
H	0	1	0	0	1	0	0	0	0	0	0	94.61%	76.88%
I	0	0	1	1	1	0	0	0	0	0	0	94.67%	77.24%
J	0	0	0	0	1	1	1	0	0	0	0	94.59%	76.54%
K	0	0	0	0	1	0	0	1	1	0	0	94.63%	76.42%
L	0	0	0	0	1	0	0	0	0	1	0	94.75%	76.27%
M	0	1	1	1	1	0	0	0	0	0	0	--	77.58%
N	0	0	1	1	1	1	1	0	0	1	0	--	76.98%
O	0	1	1	1	1	1	1	1	1	1	0	--	76.92%
P	0	0	3	3	0	0	0	0	0	0	0	--	76.01%
Q	0	0	0	0	3	0	0	0	0	0	0	--	77.69%
R	0	0	1	1	3	0	0	0	0	0	0	--	77.92%
S	0	1	1	1	3	0	0	0	0	0	0	94.92%	78.23%
T	1	1	1	1	3	0	0	0	0	0	0	94.96%	78.24%

as well as two heavy-weight CNNs, i.e., ResNet-18 and ResNet-34 [20]. Then we implement the basic blocks of

these light-weight CNN models to our LightNet overall architecture and compare the performance with existing light-

weight CNNs. Table 1 shows the experimental results. As can be observed, LightNet can achieve better performance by simply implementing the basic blocks. For example, LightNet with Inverted Residual Block surpasses EfficientNetV1 by 0.63%, 1.24% and 1.56% on CIFAR-10/100 and Tiny-ImageNet-200 respectively with only 66% of the parameters (there exists a trade-off that the Flops is enhanced by 30%). Moreover, LightNet with ShuffleNetV2 Block significantly surpasses ShuffleNetV2 by 3.48% and 3.72% on CIFAR-100 and Tiny-ImageNet-200 with 88% parameters (trade-off of 40% more Flops). Besides, LightNet with Sandglass Block significantly surpasses MobileNet by 8.01% and 9.94% on CIFAR-100 and Tiny-ImageNet-200 with 14% more parameters, and LightNet with FasterNet Block surpasses FasterNet by 6.08% and 6.46% on CIFAR-100 and Tiny-ImageNet-200 with 8% less parameters. In addition, when compared to the heavy-weight CNN models ResNet-18/34, LightNet with Residual DSC Block (obtained by replacing the standard convolution in the bottleneck block of ResNet with DS-Conv) still achieves competitive performance, while the numbers of parameters are reduced by 70% and 84% respectively. The experimental results validate the effectiveness and efficiency of LightNet overall architecture compared to the overall architectures of existing light-weight CNN models.

Then we evaluate the performance of LightNet with RepSO Block (Meta Light Block with RepSO), it is observed in Table 1 that RepSO can enhance the performance of LightNet significantly and achieves the highest accuracy on all of the three datasets, for example, it surpasses LightNet with Inverted Residual Block by 1.32% and 1.54% on CIFAR-100 and Tiny-ImageNet-200 respectively, while maintaining the inference parameters and Flops unchanged. Thus validates the effectiveness of RepSO which boosts the model performance significantly without incurring any extra inference costs.

We then evaluate the performance of FalconNet, which equips LightNet with Meta Light Block of RepSO and RefCO. As Table 1 exhibits, FalconNet can achieve higher accuracy than other existing light-weight CNNs while possessing less parameters and Flops. Moreover, compared to FalconNet without RefCO (namely, LightNet with RepSO Block), FalconNet significantly reduces the number of parameters and Flops by 28% and 37% while still achieving competitive accuracy with negligible decline of 0.11%, 0.20% and 0.26% on CIFAR-10, CIFAR-100 and Tiny-ImageNet-200 respectively. This validates the effectiveness of RefCO which significantly reduce the number of parameters and Flops while maintaining good representation capacity and competitive performance.

### 4.3. Ablation Study

#### 4.3.1 Meta Basic Block

We first conduct various ablation study to find some principles in instantiating the Meta Basic Blocks, Table 2 exhibits the experimental results. We test different combinations of the 1st/2nd/3rd spatial operators and the 1st/2nd channel operators (with the expansion ratio  $\lambda$ ). The *A* experiment validates that Meta Basic Block provides the basic ability for the model since the model when having no learnable parameters for spatial operators (i.e., identity) still achieve satisfying performance. Compare the experiments of *E, F, G, M*, it can be concluded that instantiating the 1st/3rd spatial operators with DW-Conv will undermine the final performance. Moreover, compare the experiments of *C, E, I, K, S, T*, it can be concluded that instantiating the 1st/3rd spatial operators learnable while instantiating the wnd spatial operator identity will undermine the final performance. Besides, experiments *N, S, T* illustrate that these two channel operators of PW-Convs are significant. Thus we further simplify the Meta Basic Block into Meta Light Block as illustrated in Sec. 3.2 and Fig. 4a, which only maintain the 2nd spatial operator and 1st/2nd channel operator while leaving the other two spatial operators.

#### 4.3.2 Spatial Operator

Here we conduct ablation studies to explore the form of the spatial operator in the Meta Light Block, Table 3 exhibits the experimental results. Each row exhibits the branch numbers of DW-Conv with certain kernel size that consist the corresponding spatial operator. From experiments *C, E, F*, it can be observed that employing large convolutional kernels does not contribute to the improvement of model representational capacity. Instead, it adversely impacts the model’s performance. This conclusion is consistent with the phenomenon stated in Sec. 3.3 and Fig. 5. In addition, experiments *B, C, I, S, T* validate that adding horizontal and vertical branches can enhance the representational capacity as stated in Sec. 3.3 and Fig. 5.

## 5. Conclusion

This paper factorizes the structure of light-weight from coarse to fine and obtain four vital components, namely, overall structure, basic block, spatial operator and channel operator. In light of the existing issues of these components, this paper respectively raise LightNet overall architecture, Meta Light Block, RepSO spatial operator, concept of receptive range and RefCO channel operator. Based on these four components, this paper raises FalconNet, which achieves higher accuracy with lower number of parameters and Flops compared to existing light-weight CNNs.

## References

- [1] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.
- [2] Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S-H Gary Chan. Run, don't walk: Chasing higher flops for faster neural networks. *arXiv preprint arXiv:2303.03667*, 2023. **1, 2, 3, 7**
- [3] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11030–11039, 2020.
- [4] Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yan-nis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Ji-ashi Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3435–3444, 2019.
- [5] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. **2**
- [6] Channel-Wise Convolutions. Channelnets: Compact and efficient convolutional neural networks via. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 43(8), 2021. **2, 5**
- [7] Xiaohan Ding, Honghao Chen, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Repmlpnet: Hierarchical vision mlp with re-parameterized locality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 578–587, 2022. **3**
- [8] Xiaohan Ding, Guiguang Ding, Jungong Han, and Sheng Tang. Auto-balanced filter pruning for efficient convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. **4**
- [9] Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungong Han. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1911–1920, 2019. **3, 4, 5**
- [10] Xiaohan Ding, Tianxiang Hao, Jianchao Tan, Ji Liu, Jungong Han, Yuchen Guo, and Guiguang Ding. Resrep: Lossless cnn pruning via decoupling remembering and forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4510–4520, 2021. **3**
- [11] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Diverse branch block: Building a convolution as an inception-like unit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10886–10895, 2021. **3**
- [12] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11963–11975, 2022. **1, 2, 3, 7**
- [13] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13733–13742, 2021. **3**
- [14] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12124–12134, 2022. **3**
- [15] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.
- [16] Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic network surgery for efficient dnns. *Advances in neural information processing systems*, 29, 2016. **4**
- [17] Kai Han, Yunhe Wang, Qiulin Zhang, Wei Zhang, Chunjing Xu, and Tong Zhang. Model rubik's cube: Twisting resolution, depth and width for tinynets. *Advances in Neural Information Processing Systems*, 33:19353–19364, 2020. **1, 2**
- [18] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. **4**
- [19] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015. **4**
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. **1, 3, 8**
- [21] 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 International Conference on Computer Vision*, pages 1314–1324, 2019. **1, 2, 7**
- [22] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. **1, 2, 7**
- [23] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. **2**

- [24] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. [1](#)
- [25] Lei Huang, Xianglong Liu, Bo Lang, Adams Yu, Yongliang Wang, and Bo Li. Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [26] Lei Huang, Xianglong Liu, Yang Liu, Bo Lang, and Dacheng Tao. Centered weight normalization in accelerating training of deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2803–2811, 2017.
- [27] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. [2](#)
- [28] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. [1](#)
- [30] Chao Li, Aojun Zhou, and Anbang Yao. Omni-dimensional dynamic convolution. *arXiv preprint arXiv:2209.07947*, 2022.
- [31] Duo Li, Aojun Zhou, and Anbang Yao. Hbonet: Harmonious bottleneck on two orthogonal dimensions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3316–3325, 2019. [1](#), [2](#), [7](#)
- [32] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.
- [33] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.
- [34] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 510–519, 2019.
- [35] 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*, pages 468–477, 2021. [1](#), [2](#)
- [36] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018. [2](#)
- [37] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *arXiv preprint arXiv:2207.03620*, 2022. [1](#)
- [38] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022. [3](#)
- [39] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. [3](#)
- [40] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. [1](#), [2](#), [3](#)
- [41] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. [2](#), [7](#)
- [42] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020.
- [43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- [44] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016.
- [45] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. [1](#), [2](#), [7](#)
- [46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [1](#)
- [47] Ke Sun, Mingjie Li, Dong Liu, and Jingdong Wang. Igcv3: Interleaved low-rank group convolutions for efficient deep neural networks. *arXiv preprint arXiv:1806.00178*, 2018. [2](#)
- [48] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. [1](#), [2](#)
- [49] 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*, pages 2820–2828, 2019. 1, 2
- [50] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 1, 2, 7
- [51] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021. 1, 2, 7
- [52] Mingxing Tan and Quoc V Le. Mixconv: Mixed depthwise convolutional kernels. *arXiv preprint arXiv:1907.09595*, 2019. 1
- [53] Yehui Tang, Kai Han, Jianyuan Guo, Chang Xu, Chao Xu, and Yunhe Wang. Ghostnetv2: Enhance cheap operation with long-range attention. *arXiv preprint arXiv:2211.12905*, 2022. 1
- [54] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. An improved one millisecond mobile backbone. *arXiv preprint arXiv:2206.04040*, 2022. 3
- [55] Xudong Wang and X Yu Stella. Tied block convolution: Leaner and better cnns with shared thinner filters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10227–10235, 2021.
- [56] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.
- [57] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10734–10742, 2019. 1, 2
- [58] Guotian Xie, Jingdong Wang, Ting Zhang, Jianhuang Lai, Richang Hong, and Guo-Jun Qi. Interleaved structured sparse convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8847–8856, 2018. 2
- [59] 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*, pages 1492–1500, 2017.
- [60] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. *Advances in Neural Information Processing Systems*, 32, 2019.
- [61] Weihao Yu, Pan Zhou, Shuicheng Yan, and Xinchao Wang. Inceptionnext: When inception meets convnext. *arXiv preprint arXiv:2303.16900*, 2023. 1, 2, 3
- [62] Sergey Zagoruyko and Nikos Komodakis. Diracnets: Training very deep neural networks without skip-connections. *arXiv preprint arXiv:1706.00388*, 2017.
- [63] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.
- [64] Jiangning Zhang, Xiangtai Li, Jian Li, Liang Liu, Zhucun Xue, Boshen Zhang, Zhengkai Jiang, Tianxin Huang, Yabiao Wang, and Chengjie Wang. Rethinking mobile block for efficient neural models. *arXiv preprint arXiv:2301.01146*, 2023. 1, 2, 3
- [65] Richard Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pages 7324–7334. PMLR, 2019.
- [66] Ting Zhang, Guo-Jun Qi, Bin Xiao, and Jingdong Wang. Interleaved group convolutions. In *Proceedings of the IEEE international conference on computer vision*, pages 4373–4382, 2017. 2
- [67] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 2, 7
- [68] Yikang Zhang, Jian Zhang, Qiang Wang, and Zhao Zhong. Dynet: Dynamic convolution for accelerating convolutional neural networks. *arXiv preprint arXiv:2004.10694*, 2020.
- [69] Daquan Zhou, Qibin Hou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Rethinking bottleneck structure for efficient mobile network design. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 680–697. Springer, 2020. 1, 2, 7
- [70] Yuefu Zhou, Ya Zhang, Yanfeng Wang, and Qi Tian. Accelerate cnn via recursive bayesian pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3306–3315, 2019.
- [71] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016. 2
- [72] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018. 2