Dynamic Patch Convolution (DPConv)

Shuchao Deng
School of computer science, Sichuan University, Chengdu, China
2018141461042@stu.scu.edu.cn

Abstract. Lightweight Convolutional Neural Networks (CNNs) due to the small amount of calculation and performance degradation, budget constraints depth (number of convolution layers), and width of CNN (number of channels), resulting in limited representation capabilities. To solve this problem, this paper proposes a new dynamic block convolution design to increase the depth or width of the model complexity without increasing the network. Instead of using a single convolution kernel in each layer, dynamic block convolution dynamically aggregates multiple parallel convolution kernels based on input-related attention. Assembling multiple kernels is not only computationally efficient, but also due to the small size of the kernel, but also has more representation capabilities. These kernels are aggregated in a non-linear manner through attention. The two methods of spatial dimension block and multi-head are used in the traditional CNN attention design. The image is divided into small images and then multiple attention mechanism weights are used for each small image. Compared with the traditional CNN attention, the experimental effect is better.

1. Introduction
Recently, people have a great interest in building lightweight and efficient neural networks. It not only realizes a new experience on mobile devices but also protects the privacy of users from sending personal information to the cloud. The results of recent projects (such as MobileNet[1,2,3] and ShuffleNet[4,5]) show that two effective operator designs (such as deep convolution, channel shuffling, compression, and excitation [6]) are both effective The asymmetric convolution [7]) and architecture search [8,9,10] is very important for designing efficient convolutional neural networks.

However, even the most advanced efficient CNN (for example, when the computational constraints become extreme, MobileNetV3 [11]) will significantly degrade the performance. For example, when the calculation amount of MobileNetV3 is reduced from 219M to 66M Multi Adds, the accuracy of Top-1 ImageNet classification drops from 75.2% to 67.4%. This is because the extremely low cost of calculation seriously restricts the depth (quantity) layer of the network And width (number of channels), which is crucial but proportional to the computational cost.
This paper proposes a new operator design, called dynamic block convolution, to improve the representation ability, and the additional triggers can be ignored. Dynamic block convolution uses a set of $K$ parallel convolution kernels, $\{W_k, b_k\}$. Each layer uses a convolution kernel (see Figure 2). These convolution kernels are dynamically aggregated for each input $x$ (such as an image), $\tilde{W} = \sum_k \pi_k(x)\tilde{W}_k$. Depend on attention by typing $\pi_k(x)$. Bias is to use the same attention to gather $\tilde{b} = \sum_k \pi_k(x)\tilde{b}_k$. Dynamic block convolution is a non-linear function, which has greater expressive power than static convolution. At the same time, the dynamic block convolution algorithm has higher computational efficiency. It does not increase the depth or width of the network, as the convolution kernel shares the output channel through aggregation. It will only bring additional computational overhead to the calculation of attention points $\{\pi_k(x)\}$ and the aggregation core, which is negligible compared with convolution. The key is to be within a reasonable cost range of the model size (as a convolution kernel dynamic kernel aggregation provides an efficient way to improve the presentation ability (low additional triggers)).

The dynamic block convolutional neural network is difficult to train, and it is necessary to jointly train the optimization of all convolution kernels and their considerations across multiple layers. We found two key joint optimizations: (a) limiting the attention $\sum_k \pi_k(x) = 1$ output to promote the learning of the attention model $\pi_k(x)$ and (b) early attention flattening (nearly uniform) to promote the training period of convolutional learning. We only need to integrate these two keys, and then we can use the high-temperature softmax to attract the kernel's attention. We proved the effectiveness of dynamic block convolution in image classification (ImageNet) and key point detection (COCO). No need to pick faults, just simply relocate the static convolution and dynamic block convolution in mobilenetv2 and V3, and you can achieve a slight increase in computational cost (4%) (see Figure 1). For example, the image classification accuracy of our methods MobileNetV2 and MobileNetV3 are 4.5% and 2.9%, respectively.

Combining the recent self-attention results such as VIT to achieve good performance on various tasks, compared with traditional attention structures such as SE, it can be found that the two designs with spatial dimension block and multi-head are relatively large, so we can compare The two points of the VIT are used in the attention design of the traditional CNN. First, the image is divided into small images, and then multiple attention mechanism weights are used for each small image, as shown in Figure 2. Compared with the traditional realization effect, the recognition, classification, and detection effect is better.
2. Related Work

2.1. Attention

**Efficient CNN:** Recently, designing an efficient CNN architecture has been an active research area [12,1,2,3,4,5]. SqueezeNet [12] reduces the number of parameters and widely uses 1×1 convolution in the fire module. MobileNetV1 resolves 3×3 convolution into deep convolution and point-wise convolution. Based on this, MobileNetV2[2] introduces reverse residual and linear bottleneck. MobileNetV3 [11] applies compression and excitation to the remaining layers and uses a platform-aware neural architecture method to find the best network structure [8]. ShuffleNet further reduces the MAdds channel shuffling operation of 1×1 convolution. ShiftNet [13] replaces spatial convolution with shift operations and point-wise convolution. Compared with the existing methods, our method dynamic block convolution can be used to replace any static convolution kernel (for example, 1×1, 3×3, deep convolution, group convolution), and it is suitable for other advanced algorithms Supplementary operators like to squeeze and motivate.

**Model compression and quantification:** The methods of model compression [14, 15, 16] and quantization [17, 18, 19, 20, 21] are also important for learning effective neural networks. They are complementary to our work and help reduce the model size of dynamic convolution methods.

**Dynamic deep neural network:** Our method is related to dynamic neural network [22,23,24]. Part of the latest research results are based on the input image, D2NN2 [23], SkipNet[24], and BlockDrop[25] through the use of reinforcement learning Learn the additional controller that skips the decision. MSDNet [26] withdrew early based on current forecast confidence. A thin network [27] learns a neural network that can be cut in different widths. Once and for all [28] proposed a progressive shrinking algorithm to train a network that supports multiple subnets. The accuracy of these sub-networks is the same as that of independently trained networks. Compared with these works, our method has two main differences. First of all, our method is dynamic block convolution, but the network structure is static, and the existing work is a static convolution kernel, dynamic network structure. Second, our method does not require an additional controller. Attention is embedded in each layer, so that end-to-end training is used in the traditional CNN attention design through spatial dimension block and multi-head. The image is divided into small images, and then more is used for each small image. Times the weight of the attention mechanism. Compared with parallel work, our method is more efficient and performs better.
2.2. Dynamic Deep Neural Networks

We describe the dynamic block convolutional neural network in this section. The goal is to provide a better trade-off between network performance and computational burden in the context of efficient neural networks. The two most popular strategies for improving performance make neural networks "deeper" or "wider." However, they will incur a lot of computational costs, so they are not friendly to efficient neural networks. We propose dynamic block convolution, which neither increases the depth nor the width of the network, but improves the performance of the model by paying attention to aggregating multiple convolution kernels. Please note that for different input images, the combination of these kernels is different. Dynamic block convolution is its name. In this section, we first define a general dynamic perceptron and then apply it to convolution.

2.2.1. Preliminary: Dynamic Perceptron. Definition: Let us represent the traditional or static perceptron \( y = g(W^T x + b) \), where \( W \) and \( b \) are the weight matrix and bias vector, and \( g \) is an activation function (such as ReLU). We define the dynamic perceptron by aggregation and have multiple (K) linear functions \( \{ \tilde{W}_k^T x + \tilde{b}_k \} \) as follows:

\[
y = g(\tilde{W}^T (x) x + \tilde{b}(x))
\]

\[
\tilde{W} (x) = \sum_{k=1}^{K} \pi_k(x)\tilde{W}_k, \tilde{b}(x) = \sum_{k=1}^{K} \pi_k(x)\tilde{b}_k
\]

\[s.t.0 \leq \pi_k(x) \leq 1, \sum_{k=1}^{K} \pi_k(x) = 1\]  

Where \( \pi_k \) is the attention weight for the \( k^{th} \) linear function \( \tilde{W}_k^T x + \tilde{b}_k \). Note that the total weight \( \tilde{W}(x) \) and bias \( \tilde{b}(x) \) are functions of input and share the same attention.

Attention: Note that the weight \( \{ \pi_k(x) \} \) is not fixed, but varies for each input \( x \). They represent the optimal aggregation of linear models \( \{ \tilde{W}_k^T x + \tilde{b}_k \} \) for a given input. The aggregation model \( \tilde{W}^T (x)x + \tilde{b}(x) \) is a non-linear function. Therefore, the dynamic perceptron has greater expressive power than its static counterpart.

Computational limitations: Compared with static sensors, dynamic sensors have the same number of outputs but a larger model. This article also introduces two additional calculation methods: (a) calculation of attention weights \( \{ \pi_k(x) \} \) and (b) aggregation parameters based upon attention \( \sum_k \pi_k\tilde{W}_k \) and \( \sum_k \pi_k\tilde{b}_k \). The additional calculation cost should be significantly less than the cost of computing \( \tilde{W}^T x + \tilde{b} \). Mathematically, the calculational constraint can be expressed as follows:

\[
O(\tilde{W}^T x + \tilde{b}) \gg O(\sum \pi_k\tilde{W}_k) + O(\sum \pi_k\tilde{b}_k) + O(\sum \pi(x))
\]  

Among them, \( O(\cdot) \) are the measurement costs (for example, FLOPs). Please note that fully connected layers do not meet this requirement, and convolution is a suitable constraint.
2.2.2. Dynamic Convolution. In this section, we show a specific dynamic perceptron, that is, dynamic block convolution that satisfies computational constraints (Equation 4). Similar to the dynamic perceptron, dynamic block convolution (Figure 3) has two convolution kernels that share the same kernel size and input/output dimensions. They are aggregated by using attention weights \( \{\pi_k\} \). We follow the classic design of CNN and use batch normalization and activation functions (such as ReLU) to build a dynamic block convolution layer after aggregated convolution.

Attention: We apply compression and excitation to the computing kernel attentions \( \{\pi_k(x)\} \) (see Figure 3). The global spatial information is first compressed by the global average. Then we use two fully connected layers (with ReLU between them) and softmax to generate the attention weights of the standardized K convolution kernel. The first fully connected layer reduces the dimension by 4. Different from SEnet calculating the attention on the output channel, we calculate the attention on the convolution kernel. The computational cost of this attention is very low. For the feature map with dimension \( H \times W \times C_{in} \), pay attention to the requirement \( O(\sum \pi_k(x)) = HWC_{in} + C_{in}^2 / 4 + C_{in}K / 4 \) Mult-Adds. This is much smaller than the computational cost of convolution, that is \( O(\tilde{W}^T x + \tilde{b}) = HWC_{in}C_{out}D_k^2 \) Mult-Adds, where \( D_k \) is the kernel size, and \( C_{out} \) is the number of output channels.

Kernel aggregation: The aggregation convolution kernel has higher computational efficiency due to the smaller kernel. Introduced \( D_k \times D_k \) the aggregate \( K \) convolution kernel, \( C_{in} \) the input channel, and \( C_{out} \) output channels introduces \( KC_{in}C_{out}D_k^2 + KC_{out} \) extra Multi-Adds. Compared with the amount of calculation of convolution (\( KC_{in}C_{out}D_k^2 \)), if \( K = HW \) there is, additional costs are impossible. Table 1 shows the computational cost of using dynamic block convolution in MobileNetV2. For example, when using MobileNetV2 (\( \times 1.0 \)), the dynamic block convolution of the kernel will only increase the computational cost and reduce it by 4%. Note that even dynamic convolution will increase the size of the model, but it will not increase.
### Table 1. Multi-addition of static convolution and dynamic convolution in MobileNetV2 with four different width multipliers (×1.0, ×0.75, ×0.5, and ×0.35)

|       | ×1.0     | ×0.75    | ×0.5     | ×0.35    |
|-------|----------|----------|----------|----------|
| static| 300.0M   | 209.0M   | 97.0M    | 59.2M    |
| K=2   | 309.5M   | 215.6M   | 100.5M   | 61.5M    |
| K=4   | 312.9M   | 217.5M   | 101.4M   | 62.0M    |
| K=6   | 316.3M   | 219.5M   | 102.3M   | 62.5M    |
| K=8   | 319.8M   | 221.4M   | 103.2M   | 62.9M    |

The output dimension of each layer. Since the convolution kernel is small, the amount of increase is acceptable. From CNNs to DY-CNNs: Dynamic block convolution can be implemented easily as a substitute for any convolution (for example, 1×1 conv, 3×3 conv, group convolution, deep convolution) in any CNN architecture. It can also supplement other operators (such as squeeze and excitation) and activation functions (such as ReLU6, h-swish). In the remainder of this article, we will use prefixes for networks that use dynamic block convolution. For example, DY MobileNetV2 refers to the use of dynamic block convolution in MobileNetV2. We also used weights $\tilde{W}$ to represent the kernel and ignore bias for convolution for brevity $\tilde{b}_k$.

#### 2.3. Methods of Dynamic Patch Convolution

Through the above introduction to Dynamic Convolution and its two more effective joint optimization methods. The method of face recognition, classification, and detection used in this paper is: combining scc with localshare conv, and the weight of each block in localshare is obtained by weighting by scc. The routing network generates the weights of $n, m^p, num$, $num$ is the number of shares, and $m^p$ is the number of blocks localshare, which are obtained by combining $n, m^p, o^c^k^k$. (As shown in equation 5)

$$
\begin{align*}
\begin{bmatrix} X1 & X2 \\ X3 & X4 \end{bmatrix} & \rightarrow \begin{bmatrix} \sum \text{concat}[\text{soft}(\text{MLP}(\text{avg}(X1)))] \cdot X1 \\ \sum \text{concat}[\text{soft}(\text{MLP}(\text{avg}(X2)))] \cdot X2 \\
\sum \text{concat}[\text{soft}(\text{MLP}(\text{avg}(X3)))] \cdot X3 \\ \sum \text{concat}[\text{soft}(\text{MLP}(\text{avg}(X4)))] \cdot X4 \end{bmatrix}
\end{align*}
$$

#### 3. Experiment

In this section, we will prove its effectiveness and the MobileNet series [30,31] by embedding our proposed DPConv into existing popular neural networks, including ShuffleNetV2 [29]. We compared DPConv with the latest existing technologies on ImageNet [32], MS1M-V2 [33] and COCO, in terms of image classification, face recognition, target detection, and segmentation. Unless otherwise specified, all experiments of DPConv are based on 8-learnable regions.

#### 3.1. Classification

ImageNet 2012 data set [32] is a widely accepted authoritative image classification data set, including 1.28 million training images and 50k valid images from 1000 classes. Following mainstream works, all models are trained on the entire training data set, and the accuracy of the verification set is evaluated by a single crop Top-1. For training and evaluation, the input image resolution is 224×224. The training settings are as follows [29]. All models in our experiment have been trained for 240 periods, and the learning rate decreases linearly from 0.5 to 0.
Table 2. Comparison of Top-1 classification accuracy with the latest technology (%) on ImageNet.

To prove the effectiveness of DPConv, we compared DPConv with the latest technologies included [34, 35]. The results are shown in Table 2. In the first column, for example, CondConv-ShuffleNetV2 represents the standard convolution in Shuf-ShuffleteV2 is replaced by CondConv[35]. It can be seen that DPConv-ShuffleteV2 obtains 6.3% and 3.6% gains, respectively, compared with Shuf when the calculation cost is equivalent-fleNetV2 is suitable for 0.5× and 1× scales. DPConv-MobileNetV2 achieved a 3.7% increase over MobileNetV2, and DPConv-MobileNetV1 increased 4.9%. Exceed the baseline MOBILENETEV1. These experimental results show that the DPConv-based network not only has a great improvement over the strong baseline but also has a great improvement over the existing technology, which illustrates the effectiveness of the DPConv network and the effectiveness of our method.

As the basis of some other tasks, classification needs to extract as much information as possible to predict the label of the image, because of the large amount of data in the ImageNet dataset. Traditional large-scale networks can achieve the most advanced level due to their huge depth and breadth. As for effective networks in practical applications, it is necessary to improve the extraction efficiency of useful information under the constraints of limited depth and width. Therefore, we designed DPConv without additional computational cost by making full use of the diversity of spatial information. The multi-filter spatial information strategy means that it can match more information patterns.

3.2. Face Recognition
We use MobileFaceNet[36] as the backbone network, which has only 1M parameters and 189M MADDs, and the input size is 112×96. To maintain the stability of training, we re-placed Arcface loss [37] and AM Softmax loss [38] in our implementation. The data set we used for training is MS1M-V2, which is a large-scale face data set with 5.8 million photos of 85k celebrities. This is a semi-automatic reinstallation machine-a refined version of the MS-Celeb-1M data set [33], including identities from 100k, with a lot of noisy images or wrong ID tags. The data set we used for verification is MegaFace [39], which includes 1 million 60k images as the identity of the gallery set and facial scrub as the probe set. For the same reason, it is also an exquisite version that is cleared manually.

Training and evaluation: We use SGD with a momentum of 0.9 to optimize the model and batch size of 512. We train all models for 420k iterations. The learning rate starts at 0.1 and divides by 10 iterations at 252k, 364k, and 406k. The weight attenuation is set as follows [39]. For evaluation, we use the face recognition metric which refers to the level 1 accuracy on MegaFace as the evaluation index.
To verify the effectiveness of our DPConv, we compared several related methods used by DPConv. Based on the MobileFaceNet backbone network, we use our DPConv to replace the 1×1 standard convolution in all bottleneck blocks.

Table 3. DPConv results on Megaface."ACC." Refers to the level 1 facial recognition accuracy of the 1M Distractor

| Model                      | MADDs (×106) | ACC.(%) |
|----------------------------|--------------|---------|
| MobileFaceNet              | 189          | 91.3    |
| Local-MobileFaceNet        | 189          | 94.9    |
| CondConv-MobileFaceNet     | 195          | 94.8    |
| DPConv-MobileFaceNet       | 201          | 96.2    |

Table 3 shows that the performance of DPConv MobileFaceNet is a bit better than the baseline compared to CondConv, achieving a gain of 1.4%. For further comparison, we choose local convolution for face recognition, but it requires a lot of data parameters. In the case of limited device memory, we use local convolution in the last three layers. The accuracy of DPConv MobileFaceNet is improved by 1.3% compared to native MobileFaceNet (using local convolution in MobileFaceNet), which further illustrates the superiority of our proposed DPConv. Due to spatial stability, DPConv's guided mask module can learn the semantic patterns of local statistical information in the face data set.

3.3. COCO Object Detection and Segmentation

We further evaluated the effectiveness of DPConv in target detection and recognition segmentation. We use the COCO dataset, which consists of 80k train images and 40k val images. As with many previous works, we evaluate our DPConv with training images and 35k val image subsets on the consortium of 80k trains, excluding 5k minimal images.

In the experiment, we use the DetNAS-300M [40] and maskr-CNN [41] frameworks to evaluate our method based on FPN [42] and 4conv1fc. The weights are initialized by the parameters of ClsNASNet [40] and ResNet50 [43], respectively, trained on the ImageNet dataset [32], and used as a feature extraction procedure. In DetNAS-300M, the training settings are as follows [40]. In Mask R CNN, the suggested number of possible objects on the head is set to 512. We train the detection and segmentation network in batches on 8gpu for 180k iterations, which is 16. In the beginning, we used factor to warm up the network for 500 iterations to 0.33. In the training process, we use a learning rate of 0.2. In the iterations of 120k, 140k, and 150k, the learning rate decays by 0.1 times.

Table 4. COCO object detection by DPConv and the result of segmentation. The "R" in 8R represents the area code

| Model                      | APbbox | APbbox50 | APbbox75 | APmask | APmask50 | APmask75 |
|----------------------------|--------|----------|----------|--------|----------|----------|
| DetNAS-300M                | 36.6   | 57.4     | 39.3     |        |          |          |
| DPConv-DetNAS-300M 8R      | 38.4   | 59.6     | 41.6     |        |          |          |
| MaskRCNN                   | 39.1   | 59.0     | 42.8     | 34.5   | 55.8     | 36.6     |
| DPConv-MaskRCNN 4R         | 39.8   | 60.3     | 43.3     | 35.3   | 57.1     | 37.4     |
| DPConv-MaskRCNN 8R         | 40.2   | 60.8     | 44.0     | 35.5   | 57.6     | 37.6     |
| DPConv-MaskRCNN 16R        | 40.3   | 61.2     | 44.2     | 35.6   | 58.0     | 37.6     |
Our goal is to evaluate that any improvement in the performance of the DetNAS-300M and FPN backbone of Mask R-CNN with DPConvs can be attributed to our doctors. Besides, we also applied 4-learnable area, 8-learnable areas, and 16 learnable areas to DPConv to analyze the number of different areas. The comparison results between our DPConv and the standard convolution are given in Table 4. It can be seen from the results that DPConv has 8 regions in DetNAS-300M can significantly improve detection performance by 1.8%, and Mask R-CNN with 16 regions in DR Conv can significantly improve detection performance by 1.2% and segmentation by 1.1%. Standard AP measurement based on COCO. Dr. Conv uses oriented masks to divide the space into several groups so that each filter can focus on a particular environment. On the other hand, the noise like background can also be ignored and easily separated from other areas of interest, and most filters can be concentrated on important areas. For a different number of shared areas, the results show that DPConv can achieve better performance to label more areas when performing spatial segmentation. More divided regions make up the background of each group more dedicated, and each filter can be optimized more easily.

4. Conclusion

This paper proposes a new convolution algorithm and dynamic region-aware convolution algorithm convolution (DPConv), which is driven by partial filter sharing in the spatial domain and successfully maintains translation invariance. Therefore, our proposed DPConv can completely replace any existing network of standard convolution. We designed a small learnable module to predict the mask used by the guidance system to indicate the filter assignment, which ensures that the same filter can be matched in the area. On this basis, the filter generation module is designed to make customized filters for each sample, which allows different inputs to use their dedicated filters. Comprehensive experiments on several different tasks, our DPConv has shown its effectiveness, it can greatly outperform the most advanced and other excellent hand-designed classification, face recognition, target detection, and segmentation methods. Our experiments in ablation studies show that the learnable guide mask plays a key role in the filter distribution of each sample, which helps to achieve better performance.

References

[1] 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.
[2] 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.
[3] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. CoRR, abs/1905.02244, 2019.
[4] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflflenet: An extremely efficient convolutional neural network for mobile devices. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
[5] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflflenet v2: Practical guidelines for efficient cnn architecture design. In The European Conference on Computer Vision (ECCV), September 2018.
[6] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
[7] Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungong Han. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In The IEEE International Conference on Computer Vision (ICCV), October 2019.
[8] 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 The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.
[9] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling, 2019.
[10] Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct neural architecture search on target task and hardware. In International Conference on Learning Representations, 2019.
[11] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. CoRR, abs/1905.02244, 2019.
[12] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. CoRR, abs/1602.07360, 2016.
[13] Bichen Wu, Alvin Wan, Xiangyu Yue, Peter Jin, Sicheng Zhao, Noah Golmant, Amir Gholaminejad, Joseph Gonzalez, and Kurt Keutzer. Shift: A zero flipflop, zero parameter alternative to spatial convolutions, 2017.
[14] Song Han, Huizi Mao, and William Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization, and Huffman coding. In International Conference on Learning Representations (ICLR), 2016.
[15] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In The IEEE International Conference on Computer Vision (ICCV), Oct 2017.
[16] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. Amc: Automl for model compression and acceleration on mobile devices. In The European Conference on Computer Vision (ECCV), September 2018.
[17] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28, pages 3123–3131. Curran Associates, Inc., 2015.
[18] Chenzhuo Zhu, Song Han, Huizi Mao, and William J Dally. Trained ternary quantization. In International Conference on Learning Representations (ICLR), 2017.
[19] Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In The European Conference on Computer Vision (ECCV), September 2018.
[20] Jiwei Yang, Xu Shen, Jun Xing, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-sheng Hua. Quantization networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
[21] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.
[22] Ji Lin, Yongming Rao, Jiwen Lu, and Jie Zhou. Runtime neural pruning. In Advances in Neural Information Processing Systems, pages 2181–2191. 2017.
[23] Lanlan Liu and Jia Deng. Dynamic deep neural networks: Optimizing accuracy-efficiency trade-offs by selective execution. In AAAI Conference on Artificial Intelligence (AAAI), 2018.
[24] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E. Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In The European Conference on Computer Vision (ECCV), September 2018.
[25] Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S. Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
[26] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Weinberger. Multi-scale dense networks for resource efficient image classification. In International Conference on Learning Representations, 2018.
[27] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks.
In *International Conference on Learning Representations*, 2019.

[28] Han Cai, Chuang Gan, and Song Han. Once for all: Train one network and specialize it for efficient deployment. *ArXiv*, abs/1908.09791, 2019.

[29] Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shuﬄeﬂlﬀlenet v2: Practical guidelines for eﬃcient cnn architecture design. In: European Conference on Computer Vision (ECCV). pp. 116–131 (2018)

[30] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Eﬃcient convolutional neural networks for mobile vision applications. *arXiv* preprint arXiv:1704.04861 (2017)

[31] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 4510–4520 (2018)

[32] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)* pp. 211–252 (2015)

[33] Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In: European Conference on Computer Vision(ECCV). pp. 87–102 (2016)

[34] Gao, H., Zhu, X., Lin, S., Dai, J.: Deformable kernels: Adapting eﬀective receptive ﬁelds for object deformation. *arXiv* preprint arXiv:1910.02940 (2019)

[35] Yang, B., Bender, G., Le, Q.V., Ngiam, J.: Condconv: Conditionally parameterized convolutions for eﬃcient inference. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 1305–1316 (2019)

[36] Chen, S., Liu, Y., Gao, X., Han, Z.: Mobilefacenets: Eﬃcient cnns for accurate real-time face veriﬁcation on mobile devices. In: Chinese Conference on Biometric Recognition (CCBR). pp. 428–438 (2018)

[37] Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 4690–4699 (2019)

[38] Wang, F., Cheng, J., Liu, W., Liu, H.: Additive margin softmax for face veriﬁcation. *IEEE Signal Processing Letters (SPL)* pp. 926–930 (2018)

[39] Kemelmacher-Shlizerman, I., Seitz, S.M., Miller, D., Brossard, E.: The megaface benchmark: 1 million faces for recognition at scale. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 4873–4882 (2016)

[40] Chen, Y., Yang, T., Zhang, X., MENg, G., Xiao, X., Sun, J.: Detnas: Backbone search for object detection. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 6642–6652 (2019)

[41] He, K., Gkioxari, G., Doll‘ar, P., Girshick, R.: Mask r-cnn. In: *IEEE International Conference on Computer Vision (ICCV)*. pp. 2961–2969 (2017)