An Once-for-All Budgeted Pruning Framework for ConvNets Considering Input Resolution

Wenyu Sun[1*] Jian Cao[1*†] Pengtao Xu[1] Xiangcheng Liu[1] Yuan Zhang[1] Yuan Wang[2]
[1]School of Software and Microelectronics, Peking University
[2]School of Integrated Circuits, Peking University
Beijing, China
{sunwenyu,xupengtao,wangyuan}@pku.edu.cn
{liuxiangcheng,zhangyuan}@stu.pku.edu.cn caojian@ss.pku.edu.cn

Abstract

We propose an efficient once-for-all budgeted pruning framework (OFARPruning) to find many compact network architectures close to winner tickets in the early training stage considering the effect of input resolution during the pruning process. In architecture searching stage, we measure the similarity of the pruning mask to get high-quality network architecture with low energy and time consumption. After searching stage, our proposed method randomly sample the compact architectures with different pruning rates and input resolution to achieve joint optimization. Ultimately, we can obtain a cohort of compact networks adaptive to various resolution to meet dynamic FLOPs constraints on different edge devices with only once training. The experiments based on image classification and object detection show that OFARPruning has a higher accuracy than the once-for-all compression methods such as US-Net and MutualNet (1-2% better with less FLOPs), and achieve competitive performance as the conventional pruning methods (72.6% vs. 70.5% on MobileNetv2 under 170 MFLOPs) with much higher efficiency.

1. Introduction

Convolutional neural networks have been widely used in image classification, object detection, semantic segmentation and many other fields. In addition to model accuracy, the security of data during transmission has also attracted increasing attention. Therefore, more models should be deployed to edge devices in practical application. However, with the improvement of network accuracy, the size of the network becomes larger, requiring high amounts of storage space and computing resources, making it difficult to deploy the network to edge hardware with data security but resource constraints. While ensuring the accuracy of the model and reducing the amount of model parameters and computational consumption as much as possible, researchers have proposed several model compression methods, such as model pruning [9, 18, 21, 22, 28], quantization [4, 15, 25, 39], network architecture search [7, 14, 17, 20, 29, 31, 36] and lightweight network design [8, 16, 24, 26]. Pruning has been proven to be an effective way to reduce model parameters for accelerating inference which can be divided into unstructured pruning and structured pruning. The former lacks versatility due to the requirement of a corresponding sparse matrix calculation acceleration library when deploying to the edge devices. The latter directly prunes the kernels, which can achieve acceleration on com-

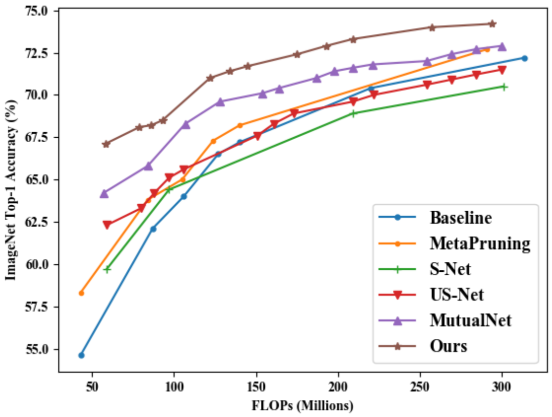


Figure 1. Overall comparison of various compression methods based on MobileNetv2. Among them, MetaPruning is once training for one network, the rest is once-for-all compression like ours.

[*] The first two authors contribute equally to this work.
[†] Corresponding author

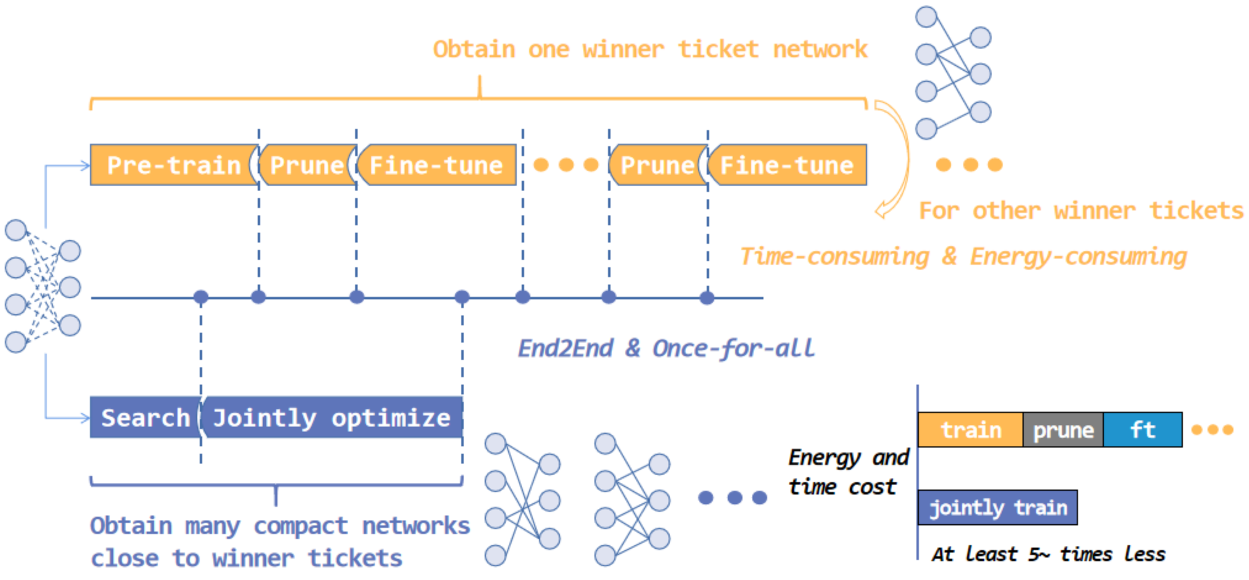


Figure 2. Comparison between conventional pruning methods based on lottery hypothesis and our proposed method. Conventional methods need pruning, fine-tuning and iterate this process to get a compact network while our method can obtain multiple compact networks close to winner tickets after once search and once joint training, saving a lot of energy and time.

mon hardware.

Since the lottery hypothesis [5] was proposed, iterative pruning has become the mainstream method for obtaining networks with high compression rate [6, 28, 38]. It requires time-consuming iterative training, pruning and fine-tuning to get a network that satisfies the FLOPs constraints of edge devices. And it is often necessary to achieve accuracy-latency tradeoff in actual applications resulting in a quite long development cycle. Besides, the large feature map of the first few layers of the network occupies a high proportion of FLOPs, and for some special hardware such as spiking neural network chips [1], a large feature map will cause network deployment difficulties. In addition, we find that knowledge distillation is necessary for compact network with a large pruning rate to improve accuracy. Based on the above findings, we propose a simple and efficient budgeted pruning framework, which randomly sample networks with different pruning rates to form a teacher-teaching assistant-student structure based on sandwich rule and utilize teaching assistant inplace distillation to search the networks corresponding to different pruning rates close to winner tickets in above framework. After searching, we adopt different resolution input for the aforementioned structures to perform joint optimization, so as to achieve multiple networks that are adaptive to different resolution to meet the dynamic constraints of FLOPs. In summary, our main contributions are as follows:

1. Propose a simple and effective neural network pruning framework which considers the effect of input resolution during the pruning process, and solve the problem that large feature maps hold numerous computation to make it feasible to deploy such networks to the specific hardware.

2. In above framework, we use an efficient method to search the networks with different pruning rates close to winner tickets. After searching, the aforementioned networks are jointly optimized through random sampling and teaching assistant inplace distillation to obtain a cohort of compact networks to meet the dynamic constraints of FLOPs which greatly reduces the time and energy consumed. From the perspective of pruning, our framework achieves once training for many compact networks.

3. Analyze the optimal configuration of the above pruning framework through ablation experiments, and prove the effectiveness of the pruning framework through image classification and object detection experiments. As shown in Fig. 1, our framework achieves a higher accuracy than the once-for-all compression methods and achieves competitive performance as the conventional pruning methods with higher efficiency.

2. Related Work

Due to acceleration on common hardware platforms, structured compression methods have attracted more and more attention. The mainstream structured compression methods can be divided into changing the network width through scaling factors and structured pruning. The former has higher development efficiency and simplicity, but the compact model is not necessarily optimal. The latter ranks the importance of the convolutional kernels in the pretrained network according to a certain standard, and then prunes the unimportant kernels so as to reduce the parameters and speed up the network inference. The conventional pruning methods can only obtain single compact network through once training while recent work focuses on how to obtain many compact models through once training to minimize time and energy consumption.

Once training for one network. NetworkSlimming [21] sparse the gamma of the BN layers by introducing a sparse factor in training process and then prune the kernels corresponding to the BN layers with small gamma values. ThiNet [23] minimize the reconstruction error to determine which channel to prune in the current layer through the statistics of the next layer based on the greedy strategy. FPGM [12] regards the convolutional kernels as points in Euclidean space and calculate the geometric median of each convolution layer to prune the kernels close to the geometric median for model compression. HRank [19] achieves pruning by utilizing the average rank of the output feature map. MetaPruning [22] generates weights for encoding vectors of different structures by training a PruningNet to find an optimal network structure under the certain constraint of FLOPs. Such pruning methods obtain only one compact network through once pruning and fine-tuning which only focus on pruning network parameter, while our proposed method not only considers the impact of input resolution on this basis, but also can obtain multiple compact networks through once training.

Once training for many networks. S-Net [37] and US-Net [35] achieve joint training of networks with different widths to obtain multiple sub-networks at one time through weight sharing between networks and outstanding statistical methods for BN layers. MutualNet [30] considers the input resolution as an influencing factor to reduce FLOPs on the basis of the network width, and achieve the joint training of network width and input resolution. OFA [2] pretrains a network then obtains sub-networks through progressive shrinking and fine-tuning to achieve once training for all networks. Such methods roughly focus on scaling networks, while our proposed method is based on structure pruning, which is more fine-grained, and obtains higher accuracy networks through more efficient searching method and distillation method.

3. Methodology

3.1. Preliminary

Sandwich Rule for OFARPruning Frame. In US-Net [35], the joint optimization of networks with arbitrary width is achieved by accumulating loss from sub-networks during training. In our proposed framework, we sample the minimum pruning rate network, the maximum pruning rate network and two networks with random pruning rates in each iteration. As shown in Equation 1, y^{ρ} means output of network with pruning rate ρ,and ρ_{min}, ρ_{max} mean minimum and maximum pruning rate,all randomly sampled pruned sub-networks can be affected by optimizing the minimum pruning ratio network (upper bound network) and the maximum pruning ratio network (lower bound network).

$$\left|y^{\rho=0} - y^{\rho_{min}}\right| \leq \left|y^{\rho=0} - y^{\rho_i}\right| \leq \left|y^{\rho=0} - y^{\rho_{\max}}\right| \quad (1)$$

Algorithm 1 OFARPruning

Input: Full model $f(x; W_0)$, Various edge devices FLOPs constraints after sorting in descending order $(c_1, c_2, c_3, \cdots c_n)$, similarity threshold τ

Output: many compact networks adaptive to different resolution

```
Calculate the pruning rate that satisfies the FLOPs constraints of different edge devices at different resolution $\{\rho_1, \rho_2, \cdots, \rho_n\}$
Build the pruning frame as shown in Figure 3 epoch= $0, 1, \ldots, N_{search}$
Prune full model with different pruning rate based on Network Slimming [21]
Calculate mask similarity between networks in adjacent epoch. mask similarity $> \tau$
Add the corresponding structure to PNP
End searching iteration in every epoch
Fix student and teacher network, the rest are randomly sampled from PNP. Choose maximum resolution input for teacher network and randomly choose resolution for other networks.
End training
```

Inplace Distillation with Teaching Assistant. Knowledge distillation is regarded as an effective method for improving the accuracy of compact network by transferring teacher network information to the student network which is widely used in practice. However, it is found that when the teacher network and the student network have a large gap in information extraction ability, the student network cannot effectively learn the large amount of detailed knowledge of the teacher network. Therefore, a medium size network is used as a teaching assistant network between the teacher network and the student network to process the information. Therefore, in our proposed framework, we form a teacher-teaching assistant-student structure to make up for the gap. According to the ablation experiment, such distillation structure is better than ordinary distillation methods.

Batch Normalization Statics. Since OFARPruning can customize the number of compact networks, we can use the

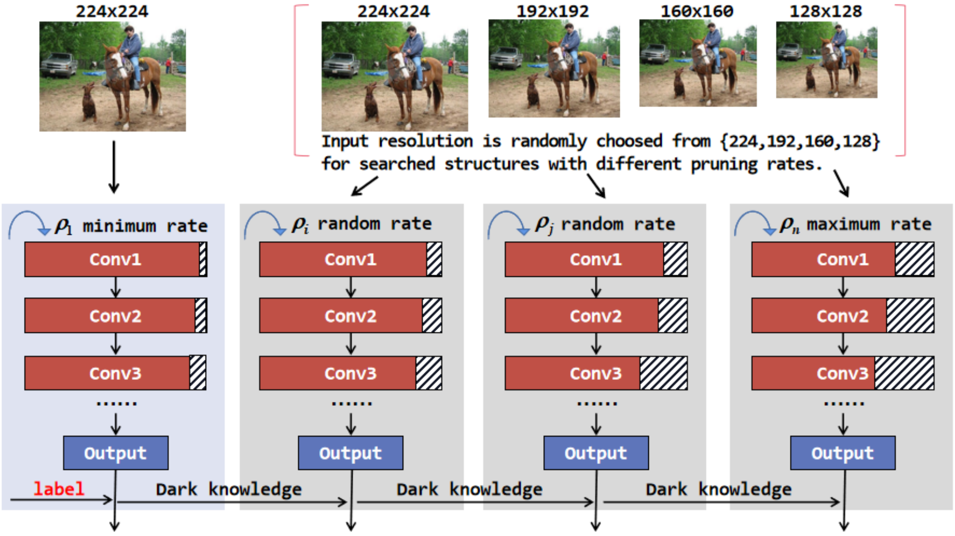


Figure 3. The frame of our proposed OFARPruning. The pruning rate is sampled from $\{\rho_1, \rho_2, ..., \rho_n\}$ to meet the dynamic resource constraints of different edge devices and input resolution is sampled from $\{224, 192, 160, 128\}$ in each iteration. We follow the sandwich rule to sample four networks with different pruning rates. For the first network, we adopt minimum pruning rate and feed 224 resolution to it due to its strong representation ability. For the other networks, we randomly sample the resolution for them. The first network is optimized by the ground truth, the other networks are optimized by the output of previous network to eliminate the representation gap between networks with different pruning rates. When the pruning mask similarity of a compact structure exceeds the threshold, the structure will be stored in the pruned network pool (PNP) for the following training.

BN storage method in S-Net or the post-statistics method of BN in US-Net. When the edge devices resources do not change much dynamically, in other words, once training does not require many compact networks, the former statistical method can be used. The latter is more suitable when the resource range of the edge devices is large and many compact networks are required for once training.

3.2. Rethink Network Budgeted Pruning

Budgeted pruning is defined as pruning networks under resource constraints. Since the lottery hypothesis was proposed, iterative pruning has become the mainstream method used in budgeted pruning to ensure high compression rate and small accuracy loss. As shown in Fig. 2, the lottery hypothesis based pruning method requires pretraining, pruning, fine-tuning and then iterate, which is time-consuming and energy-consuming. Early-Bird [34] found that the pruning mask is basically unchanged after a few epochs training and this phenomenon is common in each task and network which greatly reduces the time and energy cost of the pruning method based on the lottery hypothesis. However, such method still requires fine-tuning for each searched network structure. When there are multiple edge devices constrained by different FLOPs, it will still consume a lot of time and energy.

Besides, most of the pruning methods only focus on the structure of the network, and neglect the effect of input resolution on FLOPs. For a certain convolutional layer, as shown in Equation 2, where K represents kernel size, W and H represent width and height of output feature map, C_{in} and C_{out} represent input channel and output channel number, when the structure is close to the collapsed node, in other words, keeping pruning the structure will cause a lot of accuracy drop which is not the optimal choice. Instead, the input resolution can be changed to reduce FLOPs and maintain ordinary accuracy.

$$FLOPs = 2K^2 \times W \times H \times C_{in} \times C_{out} \tag{2}$$

3.3. OFARPruning Frame

Aiming at the shortcomings of above iterative pruning methods, we propose a simple and effective neural network pruning framework shown in Fig. 3. The specific process of our proposed method is shown in Algorithm 1. In the search stage, our framework searches for the network structures close to winner tickets corresponding to different pruning rates. For the edge device $E_i (i = 1, 2, \cdots n)$, the network computation which meets the real-time requirements is c_i MFLOPs, and the computation of the unpruned model is C MFLOPs, then the rough pruning rate of the network adaptive to the certain edge device is ρ_i. For each epoch in training process of the framework, we calculate the similarity of pruning mask in adjacent training epochs according to

Table 1. ImageNet performance comparison of different conventional pruning methods based on MobileNetv2 under same FLOPs constrain. All the results of our method are achieved through once training according to Fig. 3, in other words, OFARPruning is once-for-all.

Network	FLOPs	Top-1 Accuracy
Full Model [26]	314M	72.2%
GFS [32]	258M	71.9%
GOPWL [33]	245M	72.2%
OFARPruning(Ours)	253M	**74.0%**
Uniform [26]	220M	69.8%
AMC [11]	220M	70.8%
LeGR [3]	224M	71.4%
GFS [32]	220M	71.6%
MetaPruning [22]	217M	71.2%
GOPWL [33]	218M	71.7%
OFARPruning(Ours)	**209M**	**73.3%**
ThiNet [23]	175M	68.6%
DCP [40]	175M	68.9%
GFS [32]	170M	70.4%
GOPWL [33]	170M	70.5%
OFARPruning(Ours)	179M	**72.6%**
Uniform [26]	106M	64.0%
MetaPruning [22]	105M	65.0%
GFS [32]	107M	66.9%
OFARPruning(Ours)	**107M**	**69.2%**

Table 2. ImageNet performance comparison of different conventional pruning method based on ResNet50 under same FLOPs constrain. All the results of our method are achieved through once training, in other words, OFARPruning is once-for-all.

Network	FLOPs	Top-1 Accuracy
ThiNet [23]	2.9G	75.8%
MetaPruning [22]	3.0G	76.2%
OFARPruning(Ours)	**2.8G**	**76.9%**
Uniform [10]	2.3G	74.8%
ThiNet [23]	2.1G	74.7%
CP [13]	2.0G	73.3%
FPGM [12]	2.3G	75.6%
HRank [19]	2.3G	75.0%
MetaPruning [22]	2.0G	75.4%
DCP [40]	1.8G	75.0%
OFARPruning(Ours)	**1.7G**	**76.1%**
Uniform [10]	1.1G	72.0%
ThiNet [23]	1.2G	72.1%
MetaPruning [22]	1.0G	73.4%
OFARPruning(Ours)	1.0G	**74.1%**

Equation 3, where a,b mean the pruning mask of adjacent training epochs, $\oplus$ means XOR, n means element number of pruning mask. When the consequent similarity exceed the threshold, we add the structure to the Pruned Network Pool (PNP), and then the same method is used until all the compact networks are found. Figure 4 shows that in our proposed method, the pruning mask will effectively converge under different pruning rates after a few epochs of training.

$$similarity = 1 - \frac{\sum_{i=0}^{n-1} a[i] \oplus b[i]}{n} \tag{3}$$

After the architecture search, the framework performs joint optimization on all structures by randomly sampling and makes each compact network consider the effect of input resolution during the training process. With the advantage that teacher network is able to extract detailed information of high-resolution images and the student network is more suitable for extracting general information of low-resolution images, we select 224 input resolution for the teacher network and randomly sample resolution from $\{224, 192, 160, 128\}$ for the rest networks. Among them, the teacher network is trained by ground truth, and the output of each subsequent network is used as the label of the next network which constitutes the teacher-teaching assistant-student architecture to solve the problem that the expression ability gap of searched networks corresponding to different pruning rates is too large. The loss calculation in the framework is as follows where GT means ground truth, $f\left(.\right)$ means output of certain network, ρ means pruning rate while ρ_1 means minimum pruning rate and ρ_n means maximum pruning rate, I_R means input resolution, CE means cross entropy, KL means Kullback-Leibler divergence.

$$L_T = CE\left(f\left(\rho = \rho_1, I_R = 224\right), GT\right) \tag{4}$$

$$L_{TA1} = KL\left(f\left(\rho = \rho_i, I_R = r_i\right), f\left(\rho = \rho_1, I_R = 224\right)\right) \tag{5}$$

$$L_{TA2} = KL\left(f\left(\rho = \rho_j, I_R = r_j\right), f\left(\rho = \rho_i, I_R = r_i\right)\right) \tag{6}$$

$$L_S = KL\left(f\left(\rho = \rho_n, I_R = r_k\right), f\left(\rho = \rho_j, I_R = r_j\right)\right) \tag{7}$$

$$L_{\text{total}} = L_T + L_{TA1} + L_{TA2} + L_S \tag{8}$$

Since we prune the networks based on global rank in our proposed framework, the method of calculating partial

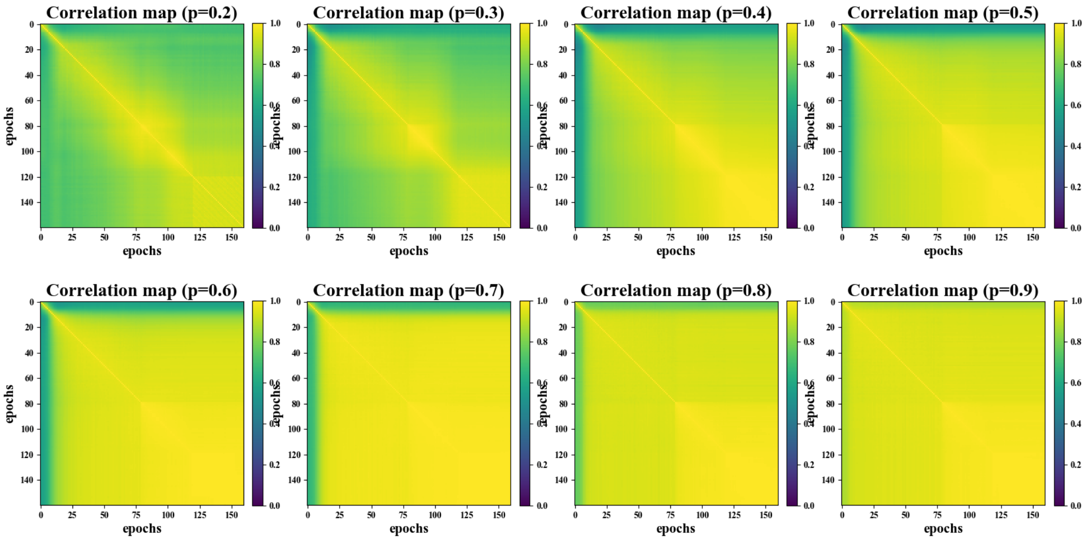


Figure 4. Similarity map with different pruning rates. The pruning mask will become very similar under a few epochs of training no matter what the pruning rate is.

derivatives in MutualNet [30] can be reused to prove effectiveness of different resolutions. Assuming that the pruning rate of a sub-network is ρ_1 and the pruning rate of another sub-network is $\rho_2(\rho_1 < \rho_2)$, the input resolution for the former is 128 while for the latter is 224. Since weights are shared between sub-networks, we can calculate the gradients as follows.

$$\begin{aligned}\frac{\partial L}{\partial W} &= \frac{\partial L_{W_{\rho_1},I_R=224}}{\partial W_{\rho_1}} + \frac{\partial L_{W_{\rho_2},I_R=128}}{\partial W_{\rho_2}} \\ &= \frac{\partial L_{W_{\rho_1},I_R=224}}{\partial W_{\rho_2}} \oplus \frac{\partial L_{W_{\rho_1},I_R=224}}{\partial W_{\rho_1-\rho_2}} + \frac{\partial L_{W_{\rho_2},I_R=128}}{\partial W_{\rho_2}} \\ &= \frac{\partial L_{W_{\rho_1},I_R=224}+\partial L_{W_{\rho_2},I_R=128}}{\partial W_{\rho_2}} \oplus \frac{\partial L_{W_{\rho_1},I_R=224}}{\partial W_{\rho_1-\rho_2}}\end{aligned} \tag{9}$$

where $\oplus$ means vector concentration, $W_{\rho_1-\rho_2}$ means kernels which the former has while the latter doesn't. Since the first item in Equation 9 consist two parts including I_R=224 and I_R=128, it effectively proves that the sub-networks with different pruning rates have the ability to learn different resolution inputs in our proposed framework.

4. Experiments

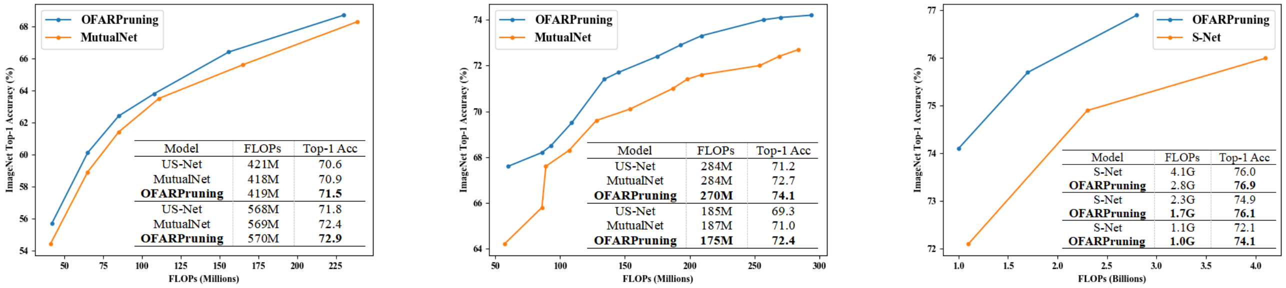
In this section, we prove the high efficiency of our proposed method comparing with conventional pruning methods and other once-for-all compression methods on ImageNet classification and COCO object detection. Further ablation experiments are performed to obtain the best configuration of the proposed framework. And OFARPruning can be migrated to pruning a single network which proves the wide applicability of the framework.

Table 3. COCO performance comparison of different once-for-all methods using VGG-16 backbone in MMDetection.

Network	FLOPs	mean Average Precision
US-Net	110G	34.4
MutualNet	110G	35.4
OFARPruning(Ours)	**103G**	**36.1**
US-Net	45.9G	33.0
MutualNet	44.9G	34.3
OFARPruning(Ours)	**42.0G**	**35.0**
US-Net	2.5G	25.0
MutualNet	2.5G	27.8
OFARPruning(Ours)	**2.0G**	**29.1**

4.1. ImageNet Classification

In this section, we test ImageNet dataset based on MobileNetv1, MobileNetv2, and ResNet50. Among them, Mo-



Model	FLOPs	Top-1 Acc
US-Net	421M	70.6
MutualNet	418M	70.9
OFARPruning	419M	**71.5**
US-Net	568M	71.8
MutualNet	569M	72.4
OFARPruning	570M	**72.9**

(a) MobileNetv1

Model	FLOPs	Top-1 Acc
US-Net	284M	71.2
MutualNet	284M	72.7
OFARPruning	**270M**	**74.1**
US-Net	185M	69.3
MutualNet	187M	71.0
OFARPruning	**175M**	**72.4**

(b) MobileNetv2

Model	FLOPs	Top-1 Acc
S-Net	4.1G	76.0
OFARPruning	2.8G	**76.9**
S-Net	2.3G	74.9
OFARPruning	**1.7G**	**76.1**
S-Net	1.1G	72.1
OFARPruning	**1.0G**	**74.1**

(c) ResNet50

Figure 5. ImageNet performance comparison of different once-for-all compression methods based on MobileNetv1, MobileNetv2 and ResNet50 under similar FLOPs constraint. The table in (a) shows some results not included in the curves.

Table 4. OFARPruning for single network Wider-ResNet based on CIFAR-10 and CIFAR-100. * means our implementation.

Dataset	Network	FLOPS	Accuracy
CIFAR-10	Baseline	5243M	96.1%
	NetworkSlimming*	987M	95.3%
	MetaPruning*	903M	95.6%
	OFARPruning(Ours)	**828M**	**96.9%**
CIFAR-100	Baseline	5243M	81.2%
	NetworkSlimming*	1289M	79.9%
	MetaPruning*	1077M	80.4%
	OFARPruning(Ous)	**1022M**	**81.1%**

bileNetv1 and MobileNetv2 represent lightweight network, while ResNet50 represents large network. Besides, MobileNetv1 is a typical non-residual network, while the rest are residual networks. Under the same range dynamic constraints of FLOPs, we first perform pruning on 1.5x width MobileNetv1 , using the same training configuration as US-Net and Mutual-Net. The Accuracy-FLOPs curve is shown as Fig. 5(a). It shows that OFARPruning has higher accuracy than US-Net (73.1% vs. 71.8%) and MutualNet (73.1% vs. 72.4%) under 570 MFLOPs. Besides, under the dynamic constraints of [50, 570] MFLOPs, OFARPruning also achieves higher accuracy than MutualNet. Besides, we conduct experiments on MobileNetv2 and ResNet50 with residual structure to further show that the network structure obtained by OFARPruning is significantly better than reducing the network width directly, and the resolution considered in the searched network can promote inter-network information flow which boosts all the networks' performance. Similarly, the 1.5x width MobileNetv2 and ResNet are pruned under the same training configuration, and the comparison results are shown in Fig. 5(b) and Fig. 5(c). We can see that OFARPruning outperforms US-Net (74.1% vs. 71.2% on MobileNetv2) and MutualNet (74.1% vs. 72.7% on MobileNetv2) under the same FLOPs constrain. As shown in Table 1 and Table 2, while PFARPruning is highly efficient, its accuracy is even higher than the state-of-the-art conventional pruning methods which are once training for single network.

4.2. COCO Object Detection

To verify the robustness and universal applicability of the OFARPruning method, we migrate it to object detection. The experiment settings in this section are consistent with the MutualNet settings, and the VGG-16 [27] is pretrained by OFARPruning. Different from MutualNet and the above classification experiments, we only perform inplace distillation with teaching assistant on the classification branch for simplicity. In the training process, the input resolution is 1000×$\{600, 480, 360, 240\}$, and mAP@0.50:0.05:0.95 is used as the performance evaluation metric. The comparison results with US-Net and MutualNet are shown in the Table 3. It shows that OFARPruning outperforms the above two methods under the same resource constraints. We also achieve a compact model with higher mAP than the full model which validates that OFARPruning method effectively optimizes each sub-network in the framework.

4.3. Single Network Pruning

OFARPruning can also be implemented to a single network. According to Equation 1, increasing the maximum pruning rate can achieve the pruning of a single network. We conduct experiments on Cifar10 and Cifar100 based on Wider-ResNet. The resolution is selected from $\{32, 28, 24, 20\}$ and the pruning rate is set $[\rho_i - 0.05, \rho_i + 0.05]$ which 0.05 is only to distinguish when sampling. After searching the corresponding structure, we also train it for 200 epochs. The result is compared in Table 4.

Table 5. Ablation experiments based on MobileNetv2. TA means teaching assistant, lr_s means learning rate in the searching stage, ρ_n means the maximum pruning rate. We choose different FLOPs levels to compare the results. - means corresponding networks are not involved in optimization through once training. The results are averaged over multiple random seeds.

Configuration	FLOPs	Top-1 Accuracy
w/ TA, lr_s=0.5, ρ_n=0.8	504M / 253M / 179M / 107M	75.5% / 74.0% / 72.6% / 69.2%
w/o TA, lr_s=0.5, ρ_n=0.8	504M / 253M / 179M / 107M	74.1% / 72.5% / 71.0% / 66.5%
Pure parameter pruning	504M / 253M / 179M / 107M	73.7% / 71.8% / 70.4% / 65.1%
w/ TA, lr_s=0.3, ρ_n=0.8	504M / 253M / 179M / 107M	74.6% / 73.0% / 71.5% / 68.0%
w/ TA, lr_s=0.1, ρ_n=0.8	504M / 253M / 179M / 107M	74.0% / 72.6% / 71.1% / 67.6%
w/ TA, lr_s=0.5, ρ_n=0.6	504M / 253M / 179M / 107M	75.7% / 74.3% / - / -
w/ TA, lr_s=0.5, ρ_n=0.4	504M / 253M / 179M / 107M	76.0% / - / - / -

4.4. Ablation Experiment

In this section, we conduct ablation experiments for each component of OFARPruning to explore the main reasons that OFARPruning can improve the performance of the pruning network, and to determine the best configuration of OFARPruning.

Inplace distillation with teaching assistant or not. US-Net and Mutual-Net adopt normal knowledge distillation to jointly train the networks sampled by sandwich rule. But we find that when we increase the maximum pruning rate, the network performance with large pruning rate will be unsatisfactory. We believe that the information extraction ability gap between the teacher network and the student network is too large, causing the student network unable to effectively obtain the output information of the teacher network. In order to confirm this conjecture, we conduct a comparative experiment based on MobileNetv2 on whether to use distillation with teaching assistant. It can be seen from Table 5 that the distillation with teaching assistant has a great impact on network with large pruning rate, and can improve the accuracy of teacher network to a certain extent.

Learning rate in searching stage. Though OFARPruning uses very low energy and time consumption to find a compact network structure that is close to the winner tickets, we find that the hyperparameters (especially learning rate) used in the searching stage have a great impact on the final result through experiments. With a fixed number of searching epochs and other related configurations, we fairly train the networks searched with different learning rates, and the results are shown in Table 5. It is observed that large learning rate contributes a lot to the emergence of high-quality network structures whose final performances are also better.

Maximum Pruning Rate. It can be seen from Equation 1 that the upper bound and lower bound network will have an impact on OFARPruning, moreover the lower bound network has a greater impact on the performance of each subnetwork. For this problem, we conduct experiments on the maximum pruning rate (lower bound network), and the results are shown in Table 5. When the maximum pruning rate is reduced, the performance of each network is improved, but the dynamic range of the model will be correspondingly narrowed.

Input Resolution. We verify the importance of considering the input resolution in the pruning process by conducting a comparative experiment shown in Table 5. It shows that with similar FLOPs, considering the input resolution in the pruning process outperforms pure parameter pruning. We believe that there are collapse nodes in pure parameter pruning. When parameter pruning rate reaches a certain level, it is more advantageous to compress the network from other dimensions such as input resolution.

In summary, our proposed effective architecture search, knowledge distillation with teaching assistant, and pruning considering resolution all promote model compression.

5. Conclusion

The paper introduces a simple and efficient pruning framework which takes input resolution into consideration during network pruning. We prove that structures close to winner tickets can be searched with low time and energy consuming in this pruning framework. After searching, we can obtain a cohort of compact networks adaptive to different input resolution to meet dynamic resource constraints of various edge devices through once joint optimization. Through image classification and object detection experiments, we show that OFARPruning has a significant increase in accuracy and efficiency.

6. Acknowledgment

This work was supported by the National Key Research and Development Program of China (Grant No. 2018YFE0203801).

References

[1] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE transactions on computer-aided design of integrated circuits and systems*, 34(10):1537–1557, 2015. 2

[2] 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. 3

[3] Ting-Wu Chin, Ruizhou Ding, Cha Zhang, and Diana Marculescu. Legr: Filter pruning via learned global ranking. *arXiv*, pages arXiv–1904, 2019. 5

[4] Jun Fang, Ali Shafiee, Hamzah Abdel-Aziz, David Thorsley, Georgios Georgiadis, and J Hassoun. Post-training piecewise linear quantization for deep neural networks. 2020. 1

[5] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018. 2

[6] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. Stabilizing the lottery ticket hypothesis. *arXiv preprint arXiv:1903.01611*, 2019. 2

[7] 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. In *European Conference on Computer Vision*, pages 544–560. Springer, 2020. 1

[8] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1580–1589, 2020. 1

[9] 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. 1

[10] 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. 5

[11] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–800, 2018. 5

[12] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2019. 3, 5

[13] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1389–1397, 2017. 5

[14] 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

[15] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018. 1

[16] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1

[17] Guohao Li, Guocheng Qian, Itzel C Delgadillo, Matthias Muller, Ali Thabet, and Bernard Ghanem. Sgas: Sequential greedy architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1620–1630, 2020. 1

[18] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016. 1

[19] Mingbao Lin, Rongrong Ji, Yan Wang, Yichen Zhang, Baochang Zhang, Yonghong Tian, and Ling Shao. Hrank: Filter pruning using high-rank feature map. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1529–1538, 2020. 3, 5

[20] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018. 1

[21] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2736–2744, 2017. 1, 3

[22] Zechun Liu, Haoyuan Mu, Xiangyu Zhang, Zichao Guo, Xin Yang, Kwang-Ting Cheng, and Jian Sun. Metapruning: Meta learning for automatic neural network channel pruning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3296–3305, 2019. 1, 3, 5

[23] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision*, pages 5058–5066, 2017. 3, 5

[24] 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. 1

[25] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1325–1334, 2019. 1

[26] 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, 5

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[28] Hidenori Tanaka, Daniel Kunin, Daniel LK Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *arXiv preprint arXiv:2006.05467*, 2020. 1, 2

[29] Alvin Wan, Xiaoliang Dai, Peizhao Zhang, Zijian He, Yuandong Tian, Saining Xie, Bichen Wu, Matthew Yu, Tao Xu, Kan Chen, et al. Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12965–12974, 2020. 1

[30] Taojiannan Yang, Sijie Zhu, Chen Chen, Shen Yan, Mi Zhang, and Andrew Willis. Mutualnet: Adaptive convnet via mutual learning from network width and resolution. In *European Conference on Computer Vision (ECCV)*, 2020. 3, 6

[31] Zhaohui Yang, Yunhe Wang, Xinghao Chen, Boxin Shi, Chao Xu, Chunjing Xu, Qi Tian, and Chang Xu. Cars: Continuous evolution for efficient neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1829–1838, 2020. 1

[32] Mao Ye, Chengyue Gong, Lizhen Nie, Denny Zhou, Adam Klivans, and Qiang Liu. Good subnetworks provably exist: Pruning via greedy forward selection. *arXiv preprint arXiv:2003.01794*, 2020. 5

[33] Mao Ye, Lemeng Wu, and Qiang Liu. Greedy optimization provably wins the lottery: Logarithmic number of winning tickets is enough. *Advances in Neural Information Processing Systems*, 33, 2020. 5

[34] Haoran You, Chaojian Li, Pengfei Xu, Yonggan Fu, Yue Wang, Xiaohan Chen, Richard G Baraniuk, Zhangyang Wang, and Yingyan Lin. Drawing early-bird tickets: Towards more efficient training of deep networks. *arXiv preprint arXiv:1909.11957*, 2019. 4

[35] Jiahui Yu and Thomas S Huang. Universally slimmable networks and improved training techniques. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1803–1811, 2019. 3

[36] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas Huang, Xiaodan Song, Ruoming Pang, and Quoc Le. Bignas: Scaling up neural architecture search with big single-stage models. *arXiv preprint arXiv:2003.11142*, 2020. 1

[37] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. *arXiv preprint arXiv:1812.08928*, 2018. 3

[38] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. *arXiv preprint arXiv:1905.01067*, 2019. 2

[39] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016. 1

[40] Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jinhui Zhu. Discrimination-aware channel pruning for deep neural networks. In *Advances in Neural Information Processing Systems*, pages 875–886, 2018. 5