Carrying Out CNN Channel Pruning in a White Box

Yuxin Zhang, Mingbao Lin, Chia-Wen Lin, Fellow, IEEE, Jie Chen, Member, IEEE, Yongjian Wu, Yonghong Tian, Fellow, IEEE, and Rongrong Ji, Senior Member, IEEE

Abstract—Channel pruning has been long studied to compress convolutional neural networks (CNNs), which significantly reduces the overall computation. Prior works implement channel pruning in an unexplainable manner, which tends to reduce the final classification errors while failing to consider the internal influence of each channel. In this article, we conduct channel pruning in a white box. Through deep visualization of feature maps activated by different channels, we observe that different channels have a varying contribution to different categories in image classification. Inspired by this, we choose to preserve channels contributing to most categories. Specifically, to model the contribution of each channel to differentiating categories, we develop a class-wise mask for each channel, implemented in a dynamic training manner with respect to the input image’s category. On the basis of the learned class-wise mask, we perform a global voting mechanism to remove channels with less category discrimination. Lastly, a fine-tuning process is conducted to recover the performance of the pruned model. To our best knowledge, it is the first time that CNN interpretability theory is considered to guide channel pruning. Extensive experiments on representative image classification tasks demonstrate the superiority of our White-Box over many state-of-the-arts (SOTAs). For instance, on CIFAR-10, it reduces 65.23% floating point operations per seconds (FLOPs) with even 0.62% accuracy improvement for ResNet-110. On ILSVRC-2012, White-Box achieves a 45.6% FLOP reduction with only a small loss of 0.83% in the top-1 accuracy for ResNet-50. Code is available at https://github.com/zyxmu/White-Box.

Index Terms—Channel pruning, efficient inference, image classification, network structure.

I. INTRODUCTION

THOUGH convolutional neural networks (CNNs) have shown predominant performance in image classification tasks [1], [2], the vast demand on computation cost has prohibited them from being deployed on edge devices such as smartphones and embedded sensors. To address this, the researchers have developed several techniques for CNN compression, such as network pruning [3], [4], parameter quantization [5], [6], tensor decomposition [7], [8], and knowledge distillation [9], [10]. Among them, channel pruning has attracted ever increasing attention for its easy combination with general hardware and Basic Linear Algebra Subprograms (BLAS) libraries, which is thus the focus of this article.

Channel pruning removes the entire channels to generate a sub-network of the original CNN with less computation cost. Existing studies could roughly be categorized into three categories. The first group abides a three-step pruning pipeline including pre-training a dense model, selection of “important” filters, and fine-tuning the sub-net. Typically, most works of this category focus on the second step by either figuring out a filter importance estimation, such as $\ell_1$-norm [11], geometric information [12], and activation sparsity [13], or regarding channel pruning as an optimization problem [14], [15]. The second category implements channel pruning with additional sparsity constraints [16]–[19], after which, the pruned model can be available by removing zeroed channels or channels below a given threshold. The last group applies AutoML techniques to directly search the channel number of each layer under a given computation budget, thus achieve pruning in an automatic manner [20]–[23].

Despite the progress, existing methods build channel pruning by observing the CNN output, that is, the final classification performance, while leaving the internal influence of a CNN model hardly touched. For example, Li et al. [11] removed filters with smaller $\ell_1$-norm, which can indeed be viewed as to minimize the output difference between the original model and the pruned model. To take a more in-depth analysis, the massive nonlinear operations inside CNNs make them hardly understandable. Thus, existing methods [11], [24] choose to regard the CNNs as a black box and observe the final output for network pruning. For instance, Ding et al. [24] leveraged binary search to remove filters with the least accumulated errors calculated by the final output. From this
Nevertheless, understanding the internal explanation of deep CNNs has attracted increasing attention [25]–[29], which also advances various vision tasks. For instance, Zeiler and Fergus [27] won the championship of the ILSVRC-2013 by adjusting architecture through visualization of internal feature maps. Inspired by this, we believe that exploring the internal logic in CNNs could be a promising prospect to guide channel pruning.

As exploited in [26], the feature maps of each channel have the locality that a particular area in one feature map is activated. Inspired by this, we visualize the feature maps generated by VGG16-Net [1] trained on ImageNet to explore the local information in the internal layers of CNNs. As can be seen from Fig. 1, the fifth channel at the 12th convolutional layer always generates feature maps that contain head information, while the 144th channel attempts to activate textual information. Even though there is no explicitly labeled head or text, this CNN model automatically learns to extract partial information to make better decisions, which exactly meets human intuition when classifying an image. That is, head information extracted by the fifth channel helps the network to identify animals, and textual information extracted by the 144th channel contributes to classify categories with texts such as digital watches. However, some local features may not be beneficial to identifying all categories. For example, the 144th channel always chooses to deactivate most of the pixels when processing images with no textual semantics like dogs and pandas (see the third and fifth columns in Fig. 1). Such local representation on the internal layers of a CNN shows that channels have varying contribution to different categories in image classification, which motivates us to rethink the importance criterion of channel pruning.

Instead of simply considering the CNN output after removing a channel as prior arts do, we target at finding each channel’s contribution to identifying different kinds of images. It is intuitive that if feature maps activated by one channel can benefit most categories’ classification, this channel is essential and should be preserved; otherwise, it can be safely removed.

To this end, we assign each channel a class-wise mask, the length of which is basically the same as the category number in the training set. Specifically, we utilize the ground-truth labels as auxiliary information in pruning the network. For each category of the input images, the corresponding mask is activated to multiply on the output feature map for model inference. By exerting a sparsity constraint that pulls the class-wise mask toward zero to counteract such gradients, these masks will maintain relatively large absolute values. On the contrary, if this channel contributes little to most categories, then the corresponding masks will be punished close to zero. Thus, after a few training epochs, each channel’s importance score can be measured by the absolute sum of its class-wise mask, reflecting its overall contribution to identifying all categories. In this way, we can carry out the pruning in an explainable manner, for which we term our pruning as “White-Box.”

We further propose an iteratively global voting, which is performed using the above importance score to remove unimportant channels until the floating point operations per seconds (FLOPs) of the pruned model meet the pre-given computation budget. It is worth mentioning that the layer-wise pruning rate can be decided in an automatic manner, which demonstrates the efficiency of White-Box compared with the previous works using hand-crafted designs [11], [14] or conducting time-consuming search [20], [22]. Last, a fine-tuning process is conducted to recover the performance of the pruned network.

Our contributions are summarized as follows.

1) Based on an in-depth analysis of CNN interpretation, we propose a novel explainable importance criterion for channel pruning that we should preserve channels beneficial to identifying most categories. To our best knowledge, this is the first time that CNN interpretability theory is considered to guide channel pruning.

2) We carry out channel pruning in a white box by jointly training a class-wise mask along with the original network to find each channel’s contribution for
classifying different categories. Then a global voting and a fine-tuning are conducted to obtain the final pruned model.

3) Extensive experiments on CIFAR-10 and ILSVRC-2012 demonstrate the advantages of the proposed White-Box over several state-of-the-art (SOTA) advances in accelerating the CNNs for image classification.

II. RELATED WORK

A. Channel Pruning

Channel pruning targets at snipping away entire channels in convolution kernel to obtain a pruned model, which not only saves computation cost, but is also compatible with off-the-shelf hardware. As discussed in Section I, previous channel pruning works can be approximately divided into three groups.

Starting from a pre-trained model, the first category designs various importance criteria to remove unimportant channels. For example, Li et al. [11] chose to prune filters with smaller $\ell_1$-norm. SASL [30] proposed to measure the importance of channels using both the prediction performance and computation consumption. He et al. [31] considered the construction error of the next layer as an importance criterion and conducted pruning in a layer-by-layer fashion. Guo et al. [32] further proposed to iteratively remove a group of channels from several selected layers instead of a single layer. Nevertheless, Guo et al. [15] observed the “next-layer feature map removal” problem that if a feature from the next layer will be removed at the next pruning stage, minimizing the reconstruction error of such features is unnecessary. To solve this, they considered both classification loss and feature importance as a pruning criterion to deal with the influence of removing next-layer feature maps. There are also some approaches that judge the importance of channels using attention modules [33]–[35]. For example, Wang et al. [35] leveraged the attention module to obtain the scaling factors of all channels, which are then served as the importance criterion. Our approach differs from attention-based pruning methods in that it considers the class-wise contribution of a single channel on the basis of CNN interpretability, while attention-based methods focus on the channel-wise attention scores.

The second group implements channel pruning in a training-adaptive manner by introducing extra sparsity regularization. For example, Huang and Wang [36] introduced a scaling factor to scale the outputs of specific structures and added sparsity on these factors. They then trained the sparsity-regularized mask for network pruning through data-driven selection. Luo and Wu [17] employed an “autopruner” layer appended in the convolutional layer to prune filters automatically. By regularizing auxiliary parameters instead of original weights values, Xiao et al. [37] pruned the CNN model via a gradient-based updating rule. Chen et al. [38] designed a channel-wise gate to dynamically estimate the conditional accuracy change and gradually prune channels during training process. Tang et al. [39] further explored the manifold information to dynamically excavate the channel redundancy of CNNs.

The last category of channel pruning approaches applies AutoML techniques to directly search the channel number of each layer under a given computation budget [20], [22]. For instance, He et al. [20] proposed to automate the searching process by reinforcement learning. Liu et al. [22] trained a PruningNet in advance to predict the weights of candidate networks and leverage evolutionary algorithm to search for the best candidate. Unfortunately, all of these methods conduct channel pruning with respect to the CNN output, failing to consider the internal mechanism of a CNN model. Though there are considerable improvements, interpretation for channel pruning remains an open problem.

B. CNN Interpretation

Despite an impressive performance in various tasks, CNNs have long been known as “Black-Box” for its end-to-end learning strategy. As an Achilles’ heel, CNN interpretability has attracted increasing attention in recent years [25]–[29]. Most studies of understanding CNN representations fall into visualization. Zeiler and Fergus [27] proposed a visualizing technique by projecting the feature activations back to the input pixel space to observe the function of intermediate visualization. Zeiler and Fergus [27] proposed a visualizing technique by projecting the feature activations back to the input pixel space to observe the function of intermediate visualization. Yosinkin et al. [26] further developed a tool that visualizes the activations produced on each layer of a trained CNN model. They then gleaned several surprising intuitions.
using this tool, including that some channels always represent useful partial information for classification decisions like faces or text, although there are no explicit labels for these items. Such phenomenon is also found in [29], which motivates us to consider what kind of channels should be pruned from an interpretable perspective. Recently, CNN interpretation theory also demonstrates its effectiveness by achieving SOTA results on various tasks such as image classification [28] and object detection [25]. Hence, we believe CNN interpretability could be a superior foreground for guiding channel pruning.

III. PROPOSED METHOD

A. Background

Considering an $L$-layer CNN model, its kernel weights can be represented as $\mathcal{W} = \{W^1, W^2, \ldots, W^L\}$. The kernel in the $l$th layer is denoted as $W^l = \mathbb{R}^{C_{in}^l \times C_{out}^l \times K^l \times K^l}$, where $C_{out}^l$, $C_{in}^l$, and $K$ denote the numbers of output channels and input channels, and the kernel size, respectively. Let $\mathcal{I} = \mathbb{R}^{N \times H \times W}$ be the input of the $l$th layer where $N$ is the batch size of input images, and $H$ and $W$, respectively, stand for the height and width of the input. Given a batch of training image set $X$, associated with a set of class labels $Y \times D$, where $D$ represents the total number of categories in the training set, we denote $X = \mathcal{I}$. For the $i$th input image $X_{i,:,:,;}$, we treat its label $Y_{i,:} = j$ as a one-hot vector. Sometimes, we simply denote $Y_{i,:} = j$ to indicate that the $j$th entry is set to 1, while the others are zero. With a conventional CNN, the output of the $l$th layer can be calculated as

$$O^l = \mathcal{I} \odot W^l$$ (1)

where $\odot$ denotes the convolutional operation. We have $O^l = \mathcal{I}^{l+1} \in \mathbb{R}^{N \times C_{in}^{l+1} \times H^{l+1} \times W^{l+1}}$ and $C_{out}^{l+1} = C_{in}^{l+1}$. Then, its training loss can be expressed as

$$\min_{\mathcal{W}} L(X, W; Y).$$ (2)

For channel pruning, a subgroup of output channels in $\mathcal{W}^l$ will be removed to obtain pruned kernel $\mathcal{W}^l \in \mathbb{R}^{C_{in}^l \times C_{out}^l \times K^l \times K^l}$ under the constraints of $C_{out}^{l} \leq C_{out}^{l+1}$ and $C_{in}^{l} \leq C_{in}^{l+1}$, thus it can reach a better trade-off between computation cost and accuracy performance. It is worth mentioning that the corresponding input channels of $\mathcal{W}^{l+1}$ are also removed. Accordingly, we can reformulate (1) and (2) in the pruned network as follows:

$$\tilde{O}^l = \tilde{\mathcal{I}} \odot \tilde{W}^l$$ (3)

$$\min_{\tilde{\mathcal{W}}} L(X, \tilde{W}; Y).$$ (4)

As discussed in Section I, most prior works perform channel pruning by directly judging channels’ redundancy using an importance estimation or imposing sparsity penalty to dynamically conduct channel pruning. Essentially, these methods complete network pruning based on the final network outputs in an unexplainable way, which neglects the internal influence of channels, thus we term these methods “Black-Box pruning.”

1It also indicates that the image $X_{i,:,...}$ belongs to the $j$th class.
2The convolutional operation usually involves a bias term and is followed by a nonlinear operation. For ease of representation, we omit them in this article.

Algorithm 1 Algorithm Description of White-Box

| Input: An L-layer CNN, global pruning rate $\alpha$, mask training epochs $T$ |
| Output: The pruned model and its parameters $\mathcal{W}$ |
| 1 Initialize model parameters $\mathcal{W}$, pruning rate $\alpha = 0$, class-wise mask $\mathcal{M} = \{1\}$; |
| 2 for $h = 1 \rightarrow T$ do |
| 3 | Train model with $\mathcal{W}$ and $\mathcal{M}$ via Eq. 10; |
| 4 end |
| 5 Get criterion score $S$ via Eq. 11; |
| 6 Sort $S$ in ascending order; |
| 7 while $\alpha \leq \alpha$ do |
| 8 | Remove the first element in $S$ and its corresponding channel in $\mathcal{W}$ and mask in $\mathcal{M}$; |
| 9 | Update current FLOPs pruning rate $\alpha$; |
| 10 end |
| 11 Integrate $\mathcal{M}$ into $\mathcal{W}$ via Eq. 12; |
| 12 Fine-tune pruned model to recover performance. |

In contrast, for the first time, we conduct channel pruning by exploring the internal influence of CNNs. Our motivation is based on the observation in [26], which reveals that the representations in the internal CNN layers are surprisingly local, implying that many channels are only responsible for extracting partial information. We argue that some of this partial information is redundant and may not be beneficial to classify all categories, as illustrated in Fig. 1. Hence, it is crucial to identify each channel’s ability to derive recognizable local features that contribute to recognizing categories.

B. Pipeline of White-Box

In order to address the above issues, we propose a novel explainable channel pruning method, termed “White-Box.” The purpose of White-Box is to find channels that can generate feature maps containing discriminative category information as much as possible, so as to retain those that contribute to the recognition of most categories and prune those channels that only benefit few categories. Specifically, we first design a class-wise mask to multiply on each channel to guide a class-wise training. When training a particular class of images, the corresponding mask will be activated for model inference and backpropagation, which will be described in detail in Section III-C. Fig. 2 shows the framework of White-Box for class-wise mask training. Subsequently, as explained in Section III-D, we propose a global voting mechanism to preserve those channels that make contributions to the recognition of most classes, as well as to automatically determine the layer-wise pruning rate without manual involvement. Finally, a fine-tuning process is conducted to boost the performance of the pruned model. Our White-Box is summarized in Algorithm 1.

C. Class-Wise Mask

The core of our White-Box is to assign per-layer kernel $\mathcal{W}^l$ a class-wise mask, which is formatted in the form of
\( M^l \in \mathbb{R}^{D \times C_{\text{out}}} \). Specifically, the mask value \( M_{j,c}^l \) is built to measure the contributions of individual channels \( W_{c_{\text{out}}}^j \) to the network for recognizing the \( j \)th category.

Then, for the \( i \)th input image \( X_{i,:,:} \) with label \( Y_{i,:} \), the convolution using (1) in the forward propagation under our mask framework can be rewritten as

\[
\mathcal{O}_{i,:,:}^l = \mathcal{O}_{i,:,:}^l \oplus (M_{Y_{i,:}}^l \ast W^j), \quad i = 1, 2, \ldots, N \tag{5}
\]

where * denotes the channel-wise multiplication, that is, channel \( W_{c_{\text{out}}}^j \) is multiplied with the scalar mask \( M_{j,c}^l \).

Subsequently, our training loss can be obtained as

\[
\min_{\mathcal{W}, \mathcal{M}} \mathcal{L}(X, \mathcal{W}, \mathcal{M}; Y) \tag{6}
\]

The rationale of our mask design lies in that, during back-propagation, the mask \( M_{j,c}^l \) will receive the gradient signals regarding the input images of the \( j \)th category. On the premise of this principle, if channel \( W_{c_{\text{out}}}^j \) benefits the network to recognize input images from the \( j \)th category, \( M_{j,c}^l \) will be positively activated, and deactivated, otherwise. Therefore, our class-wise mask design can well reflect the internal logic in CNNs, which seamlessly follows our motivation behind the channel pruning in our white box.

In comparison with typical CNNs where the label information is utilized in the loss layer, our class-wise mask-based convolutional operations are more label-guided since it requires label information in every convolutional layer as shown in (5). This poses a critical challenge of the over-fitting problem since the label information is in the format of one-hot vector, meaning that we need to provide ground-truth labels for each convolutional layer’s forward propagation. Such data flow during training varies largely from the real testing part, thus may cause the over-fitting problem.

Inspired by the label-smoothing regularization [40], we propose to solve this problem by softening the one-hot vector, denoted as \( \hat{Y} \in \mathbb{R}^{N \times D} \), element of which is defined as

\[
\hat{Y}_{i,d} = \begin{cases} 
Y_{i,d}, & \text{if } Y_{i,d} = 1 \\
\mathcal{N}(0, 1), & \text{otherwise}
\end{cases} \tag{7}
\]

where \( \mathcal{N}(\cdot, \cdot) \) denotes the normal distribution.

Then, the convolution in (5) is reformulated as

\[
\mathcal{O}_{i,:,:}^l = \sum_{d=1}^{D} T_{i,:,:}^l \oplus ( (\hat{Y}_{i,d} \cdot M_{d,c}^l) \ast W^j) \tag{8}
\]

Thus, channel pruning can be realized by removing those channels with poor masks. It is natural to impose sparsity constraint on per-channel mask as

\[
\min_{\mathcal{M}} \sum_{l=1}^{L} \sum_{c=1}^{C_{\text{out}}} \| M_{l,c}^l \|_2. \tag{9}
\]

Note that we choose \( \ell_2 \)-norm instead of \( \ell_1 \)-norm as previous sparsity regularization works do [17], [37]. The rationale falls in that our object is not to regularize the masks to exactly 0s that \( \ell_1 \) norm leads to, but to measure the class-wise contribution of each channel. After training, we leverage global voting to directly obtain the pruned model, which will be introduced in the following section. Thus, we choose to leverage \( \ell_2 \)-norm for its smoothness and rotation invariance.

Equations (6) and (9) lead to our final training loss

\[
\min_{\mathcal{W}, \mathcal{M}} \mathcal{L}(X, \mathcal{W}, \mathcal{M}; Y) + \lambda \sum_{l=1}^{L} \sum_{c=1}^{C_{\text{out}}} \| M_{l,c}^l \|_2. \tag{10}
\]

Noticeably, the objective of (10) targets at locating channels that contribute more to recognizing the input images, which then make up of the pruned kernel \( \hat{W} \) as described in Section III-A, followed by a series of fine-tuning procedures using loss objective of (6). Therefore, only a few epochs are needed to train our class-wise mask so as to derive \( \hat{W} \) in our empirical observation.  

### D. Global Voting for Cross-Layer Pruning

Given a global pruning rate \( \alpha \), how to appropriately distribute it to each layer to preserve \( C_{\text{out}}^l \) channels would significantly affect the performance of the pruned model [41]. Prevalent methods resort to rule-of-thumb designs [11], [14] or complex structure search [22], [42].

Fortunately, our White-Box can tacitly obtain a global important criterion for all channels in the network and conduct layer-wise pruning rate decision in an iteratively voting manner. In detail, considering a trained class-wise mask \( M_{l,c}^l \in \mathbb{R}^D \) of the \( c \)th channel in the \( l \)th layer, each item in this tensor represents this channel’s ability for classifying one corresponding category of the dataset, thus we can measure this channel’s contribution to overall classification performance by simply summing up these class-wise mask scores. We denote all scores of \( M^l \) as \( S_c^l \in \mathbb{R}^{C_{\text{out}}} \)

\[
S_c^l = \sum_{d=1}^{D} M_{d,c}^l, \quad c = 1, \ldots, C_{\text{out}} \tag{11}
\]

which then will serve as an importance criterion for this channel.

Given a global pruning rate \( \alpha \), after obtaining all channels’ importance scores \( S \) in the whole network, we iteratively remove the least-impact channels and calculate FLOPs pruning rate \( \hat{\alpha} \) of the current model until \( \hat{\alpha} \geq \alpha \). After voting, we integrate the left class-wise mask \( \hat{M} \) into \( \hat{W} \) to conduct fine-tuning for performance recovery. Particularly, as we soften the label obeying a standard normal distribution \( \mathcal{N}(\mu = 0.5, \sigma = 1) \) during training except for the ground-truth-related current input, the overall pruned \( \hat{M} \) can be mixed into \( \hat{W} \) by

\[
\hat{W} = \hat{W} \ast \sum_{c=1}^{C_{\text{out}}} \mu \cdot \hat{M}_{d,c}. \tag{12}
\]

Finally, more epochs are used to fine-tune the pruned model.

### IV. EXPERIMENTS

#### A. Implementation Details

1) Datasets and Backbones: We conduct extensive experiments on two representative datasets including CIFAR-10 [43]

\[\text{We consider 10% of the total fine-tuning epochs for training the class-wise mask.}\]
TABLE I
RESULTS FOR PRUNING VGGNET-16 ON CIFAR-10

| Model     | Top-1 Acc. | Acc. ↓ | FLOPs ↓ |
|-----------|------------|--------|---------|
| $\ell_1$ [11] | 93.25% → 93.40% | -0.25% | 34.2%   |
| GAL [47]  | 93.02% → 92.03% | 1.93%  | 39.6%   |
| SSS [36]  | 93.02% → 93.02% | 0.00%  | 41.6%   |
| Slimming [16] | 93.66% → 93.80% | -0.14% | 51.0%   |
| HRank [14] | 93.02% → 91.23% | 1.79%  | 76.5%   |
| White-Box | 93.02% → 93.47% | -0.45% | 76.4%   |

TABLE II
RESULTS FOR PRUNING RESNET-56 ON CIFAR-10

| Model     | Top-1 Acc. | Acc. ↓ | FLOPs ↓ |
|-----------|------------|--------|---------|
| HRank [14] | 93.26% → 93.17% | 0.09%  | 50.0%   |
| AMC [20]  | 92.80% → 91.90% | 0.90%  | 50.0%   |
| SCP [48]  | 93.69% → 93.23% | 0.46%  | 51.5%   |
| SFP [49]  | 93.59% → 92.26% | 1.33%  | 52.6%   |
| LFPC [50] | 93.26% → 93.24% | 0.02%  | 52.9%   |
| DSA [51]  | 93.12% → 92.91% | 0.21%  | 53.2%   |
| FPGM [12] | 93.59% → 92.93% | 0.66%  | 53.6%   |
| White-Box | 93.26% → 93.54% | -0.28% | 55.6%   |

TABLE III
RESULTS FOR PRUNING RESNET-110 ON CIFAR-10

| Model     | Top-1 Acc. | Acc. ↓ | FLOPs ↓ |
|-----------|------------|--------|---------|
| $\ell_1$ [11] | 93.55% → 93.30% | 0.25%  | 38.7%   |
| Rethink [41] | 93.77% → 93.70% | 0.07%  | 40.8%   |
| SFP [49]  | 93.68% → 93.38% | 0.30%  | 40.8%   |
| GAL [47]  | 93.39% → 92.74% | 0.65%  | 48.5%   |
| HRank [14] | 93.50% → 93.36% | 0.14%  | 58.2%   |
| LFPC [50] | 93.50% → 93.07% | 0.43%  | 60.3%   |
| ABC [52]  | 93.57% → 93.58% | -0.01% | 65.0%   |
| White-Box | 93.50% → 94.12% | -0.62% | 66.0%   |

TABLE IV
RESULTS FOR PRUNING MOBILENET-V2 ON CIFAR-10

| Model     | Top-1 Acc. | Acc. ↓ | FLOPs ↓ |
|-----------|------------|--------|---------|
| WM [53]   | 94.47% → 94.02% | 0.45%  | 27.0%   |
| DCP [54]  | 94.47% → 94.69% | 0.22%  | 27.0%   |
| MDP [55]  | 95.02% → 95.14% | -0.12% | 28.7%   |
| White-Box | 95.02% → 95.28% | -0.26% | 29.2%   |

and ILSVRC-2012 [44] to demonstrate the efficacy of the proposed White-Box. We prune prevailing CNN models including VGG-16 [1], ResNet-56/110 [2], MobileNet-v2 [45] on CIFAR-10, and ResNet-50 [2] on ILSVRC-2012.

2) Configurations: We set the sparse parameter $\lambda$ as $10^{-2}$ for VGGNet-16 and MobileNet-v2, and $5 \times 10^{-4}$ for ResNets. Then, we train our class-wise masks using the original full network with a learning rate of 0.1 for 30 epochs on CIFAR-10 and nine epochs on ILSVRC-2012. After the global voting, the pruned model is then fine-tuned via the stochastic gradient descent (SGD) optimizer. The momentum and batch size are set to 0.9 and 256, respectively, in all experiments. On CIFAR-10, we iterate 300 epochs to fine-tune the pruned model with an initial learning rate of 0.1, which is divided by 10 at the 150th and 225th epochs. On ILSVRC-2012, ResNet-50 is fine-tuned for 90 epochs with a step scheduler learning rate, which begins at 0.1 and is divided by 10 every 30 epochs. The weight decay rate is set to $5 \times 10^{-4}$ for all models except for MobileNet-v2: $4 \times 10^{-5}$ on CIFAR-10 and $10^{-4}$ on ILSVRC-2012. All experiments are implemented with Pytorch [46] and run on NVIDIA Tesla V100 GPUs. The data argumentation includes crop and horizontal flip.

B. Comparison on CIFAR-10

We first demonstrate the superiority of White-Box on CIFAR-10. The FLOPs pruning rate of the compressed models and their top-1 accuracy performance are reported. The accuracy reported is in the format of “pre-trained model → pruned model.” Several SOTA channel pruning methods are compared, including $\ell_1$ [11], SSS [36], GAL [47], HRank [14], SCP [48], DSA [51], SFP [49], FPGM [12], LFPC [50], Rethink [41], ABC [52], and WM [53].

1) VGGNet-16: Table I shows the results of pruning 16-layer VGGNet [1] model, which consists of 13 sequential convolutional layers and three fully connected layers. As can be seen, White-Box yields significantly better top-1 accuracy of 93.47% compared to the recent SOTA and HRank [14] of 91.23%, under similar FLOPs reductions. Moreover, compared to GAL [47] which simply imposes masks upon the outputs of convolutions, our White-Box that considers the internal influence of each channel to the categories, results in a significant reduction on the FLOPs, that is, 76.64% versus 39.6%, while retaining a better top-1 accuracy of 93.47% versus 92.03%.

2) ResNet: We also evaluate the network pruning performances of various methods on ResNet [2], a predominant deep CNN with residual modules, as shown in Tables II and III. As can be observed, our White-Box increases the performance of original ResNet-56 by 0.28% and removes around 55.60% computation burden, while the other methods suffer the accuracy degradation more or less, even reducing less FLOPs. Besides, our White-Box also shows impressive superiority when pruning ResNet-110. With 66.0% reductions on FLOPs, it still yields 0.55% performance improvement, surpassing the other methods by a large margin.

3) MobileNet-v2: MobileNet-v2 [45] is a prevailing network with a compact design of depth-wise separable convolution. Due to its extremely small computation cost, pruning MobileNet-v2 becomes a particularly challenging task. Nevertheless, compared with the competitors in Table IV, White-Box still retains better top-1 accuracy of 95.23%, while pruning more FLOPs of 29.2%.

Furthermore, we plot the performance comparison under different pruning rates of FLOPs in Fig. 3. To show our advantage, we compare the proposed White-Box with several SOTAs. As can be observed, though the pruning rate changes, our White-Box consistently retains a higher top-1 accuracy, which well demonstrates the correctness of exploring the internal CNNs.
Fig. 3. Top-1 accuracy comparison between existing methods and the proposed White-Box under different pruning rates of FLOPs. The experiments are conducted using ResNet-56 and ResNet-110 on CIFAR-10.

### Table V

**RESULTS FOR PRUNING RESNET-50 ON ILSVRC-2012**

| Method          | Top-1 Acc. | Top-1 Acc. ↓ | Top-5 Acc. | Top-5 Acc. ↓ | FLOPs   | FLOPs ↓ |
|-----------------|------------|---------------|------------|---------------|---------|---------|
| SSS-26 [36]     | 76.15% → 74.18% | 1.97%         | 92.96% → 91.91% | 1.05%          | 2.82G   | 31.9%   |
| CP [31]         | 76.15% → 72.30% | 3.85%         | 92.96% → 90.80% | 2.16%          | 2.73G   | 34.1%   |
| SPP [49]        | 76.15% → 74.61% | 1.54%         | 92.87% → 92.06% | 0.81%          | 2.39G   | 41.8%   |
| GAL [47]        | 76.15% → 71.95% | 4.20%         | 92.96% → 90.79% | 2.17%          | 2.33G   | 43.7%   |
| SSS-32 [36]     | 76.12% → 71.82% | 4.30%         | 92.86% → 90.79% | 2.07%          | 2.33G   | 43.7%   |
| HRank [14]      | 76.15% → 75.01% | 1.14%         | 92.96% → 92.33% | 0.63%          | 2.30G   | 43.9%   |
| **White-Box**   | 76.60% → 75.32% | **0.83%**     | 92.96% → 92.43% | **0.53%**      | **2.22G** | **45.6%** |
| MetaPruning [22] | -          | -             | -          | -             | 2.00G   | 48.7%   |
| FPGM [12]       | 76.15% → 74.13% | 2.02%         | 92.96% → 92.87% | 0.09%          | 1.90G   | 53.5%   |
| RRRP [56]       | 76.15% → 73.00% | 3.15%         | 92.96% → 91.00% | 1.96%          | 1.86G   | 54.5%   |
| GAL [47]        | 76.15% → 71.80% | 4.35%         | 92.96% → 90.82% | 2.14%          | 1.84G   | 55.6%   |
| ThiNet [57]     | 72.88% → 71.01% | 1.87%         | 91.06% → 90.02% | 1.12%          | 1.71G   | 58.7%   |
| LFPC [50]       | 76.15% → 74.18% | 1.97%         | 92.96% → 91.92% | 1.04%          | 1.61G   | 60.8%   |
| HRank [14]      | 76.15% → 71.98% | 4.17%         | 92.96% → 91.01% | 1.95%          | 1.53G   | 62.6%   |
| **White-Box**   | 76.15% → 74.21% | **1.94%**     | 92.96% → 92.01% | **0.95%**      | **1.50G** | **63.5%** |

### Table VI

**FLOPs, LATENCY, AND ACCURACY OF WHITE-BOX FOR PRUNING THE RESNET-50. REPORTED LATENCY IS THE RUN-TIME OF THE CORRESPONDING NETWORK ON ONE NVIDIA TESLA V100 GPU WITH A BATCH-SIZE OF 32**

| Method          | FLOPs   | Latency | Speedup | Top-1 Acc. |
|-----------------|---------|---------|---------|------------|
| Baseline        | 4.11G   | 1.75ms  | 0.00×   | 76.15%     |
| White-Box       | 2.22G   | 1.29ms  | 1.35×   | 75.32%     |
| White-Box       | 1.50G   | 1.08ms  | 1.62×   | 74.21%     |

### C. Comparison on ILSVRC-2012

We further show the results for pruning ResNet-50 [2] on ILSVRC-2012. In Table V, compared with the SOTAs, White-Box shows the best performance under different pruning rates. By setting $\alpha$ to 0.45, White-Box reduces the FLOPs to around 2.22B while obtaining the top-1 accuracy of 75.32% and top-5 accuracy of 92.43%. In contrast, the recent HRank [14] bears more computation of 2.30 FLOPs and poor top-1 accuracy of 75.01% and top-5 accuracy of 92.33%. Furthermore, we increase $\alpha$ to 0.63 and White-Box shows the least accuracy drops of 2.02% in top-1 accuracy and 1.03% in top-5 accuracy. With less FLOP reductions, LFPC [50] shows poor top-1 accuracy of 74.18% and top-5 accuracy of 91.92%. Table VI reports the reduction of inference time. Our White-Box achieves significant speedups while losing marginal performance. For instance, it obtains $1.62 \times$ GPU speedups with only 1.94% top-1 accuracy drop, compared with the baseline.

### D. Ablation Study

1) Class-Wise Mask: In this section, we prune ResNet-56 and test its performance on CIFAR-10 as an example to investigate the influences of individual components in our class-wise mask. We first train each channel with a single mask...
for all categories of images, denoted as w/o Class-wise mask in Table VII. Such a mechanism suffers more accuracy drops as it fails to consider individual channel’s discriminating power to recognize different categories as discussed in Section I. In addition, we conduct experiments without the smooth operation for mask activation, which is referred to as w/o Soft mask. Table VII shows that such an implementation leads to the over-fitting problem that the network will converge in one epoch. Thus, the trained mask cannot well contribute to discriminating different categories, leading to an even worse top-1 accuracy than w/o Class-wise mask under a similar FLOP reduction.

We also train the class-wise mask with $\ell_1$-norm regularization, denoted as “w $\ell_1$-norm” in Table VII. The $\ell_1$-norm brings worse performance than $\ell_2$-norm, thus demonstrating our motivation of choosing $\ell_2$-norm to measure the class-wise contribution of each channel. Lastly, we show the distribution of the mask value after training over different classes. Visualization in Fig. 4 shows that the class-wise mask value changes from channel to channel, which well confirms the motivation of White-Box. Channels with low mask values over all classes will be pruned using the global voting.

2) Sparsity Factor: We further analyze the impact of the sparsity factor $\lambda$. We choose to prune VGGNet-16 on CIFAR-10 with different $\lambda$ under the similar FLOP pruning rate. In Fig. 5, with different $\lambda$, all of the pruned networks perform significantly better than the SOTA HRank [14]. To explain, our motivation for the class-wise mask training merely falls into observing each channel’s contribution to classifying different categories of image, instead of recovering performance.

V. CONCLUSION

Based on visualization and analysis of the deep feature in CNNs, we proposed a new perspective of channel pruning for efficient image classification that one should preserve channels activating discriminative features for more categories in the dataset. We further carry out channel pruning in a white box by devising a class-wise mask for each channel. During training, different sub-masks are activated for model inference, with respect to the current label of input images. A global voting and a fine-tuning are then performed to obtain the compressed model.

REFERENCES

[1] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2015, pp. 1–14.
[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
[3] S. Han, J. Pool, J. Tran, and W. Dally, “Learning both weights and connections for efficient neural network,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2015, pp. 1135–1143.
[4] X. Ding et al., “Global sparse momentum SGD for pruning very deep neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2019, pp. 6382–6394.
[5] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, “Binarized neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2016, pp. 4107–4115.
[6] Z. Liu, W. Luo, B. Wu, X. Yang, W. Liu, and K.-T. Cheng, “Bi-real Net: Binarizing deep network towards real-network performance,” Int. J. Comput. Vis., vol. 128, no. 1, pp. 202–219, Jan. 2020.
[7] B. Peng, W. Tan, Z. Li, S. Zhang, D. Xie, and S. Pu, “Extreme network compression via filter group approximation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 300–316.
[8] K. Hayashi, T. Yamaguchi, Y. Sugawara, and S.-I. Maeda, “Exploring unexplored tensor network decompositions for convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2019, pp. 5552–5562.
[9] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, “FitNets: Hints for thin deep nets,” 2014, arXiv:1412.6550.
[10] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” 2015, arXiv:1503.02531.
[11] H. Li, A. Kadav, I. Durandovic, H. Sameet, and H. P. Graf, “Pruning filters for efficient convnets,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2017, pp. 1–9.
[12] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang, “Filter pruning via geometric median for deep convolutional neural networks acceleration,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4340–4349.
[13] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, “Network trimming: A data-driven neuron pruning approach towards efficient deep architectures,” 2016, arXiv:1607.03250.
[14] M. Lin et al., “HRank: Filter pruning using high-rank feature map,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 1529–1538.
[15] J. Guo, W. Ouyang, and D. Xu, “Channel pruning guided by classification loss and feature importance,” in Proc. AAAI Conf. Artif. Intell. (AAAI), 2020, pp. 10885–10892.
[16] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, “Learning efficient convolutional networks through network slimming,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2736–2744.
[17] J.-H. Luo and J. Wu, “AutoPruner: An end-to-end trainable filter pruning method for efficient deep model inference,” Pattern Recognit., vol. 107, Nov. 2020, Art. no. 107461.
[18] X. Ding, G. Ding, Y. Guo, and J. Han, “Centripetal sgd for pruning very deep convolutional networks with complicated structure,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 4943–4953.
K. Yamamoto and K. Maeno, “PCAS: Pruning channels with attention,” 2020, arXiv:2007.03260.

Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, “AMC: Autonl for model compression and acceleration on mobile devices,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 784–800.

T.-J. Yang et al., “Netadapt: Platform-aware neural network adaptation for mobile applications,” in Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 2818–2826.

Z. Liu et al., “MetaPruning: Meta learning for automatic neural network channel pruning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 3296–3305.

Z. Liu, X. Zhang, Z. Shen, Y. Wei, K.-T. Cheng, and J. Sun, “Joint multi-dimension pruning via numerical gradient update,” IEEE Trans. Image Process., vol. 30, pp. 8034–8045, 2021.

Y. He, G. Ding, J. Guo, J. Han, and C. Yan, “Approximated oracle filter pruning for destructive cnn width optimization,” in Proc. Int. Conf. Mach. Learn. (ICML), 2019, pp. 1607–1616.

T. Wu, W. Sun, X. Li, X. Song, and B. Li, “Towards interpretable R-CNN by unfolding latent structures,” 2017, arXiv:1711.05226.

J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, “Understanding neural networks through deep visualization,” in Proc. Deep Learn. Workshop, Int. Conf. Mach. Learn. (ICML), 2015, pp. 1–12.

M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in Proc. Eur. Conf. Comput. Vis. (ECCV), New York, NY, USA: Springer, 2014, pp. 818–833.

Q. Zhang, Y. N. Wu, and S.-C. Zhu, “Interpretable convolutional neural networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8827–8836.

B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Object detectors emerge in deep scene CNNs,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2015, pp. 1–9.

J. Shi, J. Xu, K. Tasaka, and Z. Chen, “SASLI: Saliency-adaptive sparsity learning for neural network acceleration,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 5, pp. 2008–2019, May 2021.

Y. He, X. Zhang, and J. Sun, “Channel pruning for accelerating very deep neural networks,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 1378–1387.

J. Guo, W. Zhang, W. Ouyang, and D. Xu, “Model compression using progressive channel pruning,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 3, pp. 1114–1124, Mar. 2021.

F. Yu, C. Liu, D. Wang, Y. Wang, and X. Chen, “AntiDote: Attention-based dynamic optimization for neural network runtime efficiency,” in Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE), Mar. 2020, pp. 951–956.

K. Yamamoto and K. Maeno, “PCAS: Pruning channels with attention statistics for deep network compression,” in Proc. Brit. Mach. Vis. Conf. (BMVC), 2018, pp. 106.1–106.13.

X. J. Wang, W. Yao, and H. Fu, “A convolutional neural network pruning method based on attention mechanism,” in Proc. Int. Conf. Softw. Eng. Knowl. Eng. (CS영K), Jul. 2019, pp. 343–452.

Z. Huang and N. Wang, “Data-driven sparse structure selection for deep neural networks,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 304–320.

X. Xiao, Z. Wang, and S. Rajasekar, “AutoPrune: Automatic network pruning by regularizing auxiliary parameters,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2019, pp. 13681–13691.

Z. Chen, T.-B. Xu, C. Du, C.-L. Liu, and H. He, “Dynamical channel pruning by conditional accuracy change for deep neural networks,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 2, pp. 799–813, Feb. 2021.

Y. Tang et al., “Manifold regularized dynamic network pruning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 5018–5028.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2818–2826.

Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, “Rethinking the value of network pruning,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2019, pp. 1–26.

B. Li, B. Wu, J. Su, G. Wang, and L. Lin, “EagleEye: Fast sub-net evaluation for efficient neural network pruning,” 2020, arXiv:2007.02491.

A. Krizhevsky et al., “Learning multiple layers of features from tiny images,” Univ. Toronto, Tech. Rep., 2009.

O. Russakovsky et al., “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, Dec. 2015.

M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4510–4520.

A. Paszke et al., “Pytorch: An imperative style, high-performance deep learning library,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2019, pp. 8026–8037.

S. Lin et al., “Towards optimal structured CNN pruning via generative adversarial learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 2790–2799.

M. Kang and B. Han, “Operation-aware soft channel pruning using differentiable masks,” in Proc. Int. Conf. Mach. Learn. (ICML), 2020, pp. 5122–5131.

Y. He, G. Kang, X. Dong, Y. Fu, and Y. Yang, “Soft filter pruning for accelerating deep convolutional neural networks,” in Proc. 27th Int. Joint Conf. Artif. Intell., Jul. 2018, pp. 2234–2240.

Y. He, Y. Ding, P. Liu, L. Zhu, H. Zhang, and Y. Yang, “Learning filter pruning criteria for deep convolutional neural networks acceleration,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 2009–2018.

X. Ning, T. Zhao, W. Li, P. Lei, Y. Wang, and H. Yang, “DAS: More efficient budgeted pruning via differentiable sparsity allocation,” 2020, arXiv:2004.02164.

M. Lin, R. Ji, Y. Zhang, B. Zhang, Y. Wu, and Y. Tian, “Channel pruning via automatic structure search,” in Proc. 29th Int. Joint Conf. Artif. Intell., Jul. 2020, pp. 673–679.

A. G. Howard et al., “MobileNets: Efficient convolutional neural networks for mobile vision applications,” 2017, arXiv:1704.04861.

Z. Zhang et al., “Discrimination-aware channel pruning for deep neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2018, pp. 875–886.

J. Guo, W. Ouyang, and D. Xu, “Multi-dimensional pruning: A unified framework for model compression,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 1508–1517.

Y. Zhou, Y. Zhang, Y.-F. Wang, and Q. Tian, “Accelerate CNN via recursive Bayesian pruning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 3306–3315.

J. Guo, L. Wu, and W. Lin, “ThinNet: A filter level pruning method for deep neural network compression,” in Proc. Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 5058–5066.
Chia-Ven Lin (Fellow, IEEE) received the Ph.D. degree in electrical engineering from National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 2000. He worked for the Information and Communications Research Laboratories, Industrial Technology Research Institute, Hsinchu, from 1992 to 2000. He was with the Department of Computer Science and Information Engineering, National Chung Cheng University, Minxiong, Taiwan, from 2000 to 2007. He is currently a Professor with the Department of Electrical Engineering and the Institute of Communications Engineering, NTHU. He is also the Deputy Director of the AI Research Center, NTHU. His research interests include image and video processing, computer vision, and video networking.

Dr. Lin served as a Distinguished Lecturer for IEEE Circuits and Systems Society from 2018 to 2019, a Steering Committee Member for IEEE TRANSACTIONS ON MULTIMEDIA from 2014 to 2015, and the Chair for the Multimedia Systems and Applications Technical Committee of the IEEE Circuits and Systems Society from 2013 to 2015. His articles received the Best Paper Award of IEEE International Conference on Visual Communication and Image Processing (VCIP) 2015 and the Young Investigator Award of VCIP 2005. He received the Young Investigator Award presented by the Ministry of Science and Technology, Taiwan, in 2006, and the Outstanding Electrical Professor Award presented by the Chinese Institute of Electrical Engineering in 2019. He is also the Chair of the Steering Committee of IEEE International Conference on Multimedia and Expo (ICME). He has served as the Technical Program Co-Chair for IEEE ICME 2010 and 2011, the Technical Program Co-Chair of IEEE ICIP 2011 and 2017, and the General Co-Chair for IEEE VCIP 2018. He has served as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE MULTIMEDIA, and Journal of Visual Communication and Image Representation.

Jie Chen (Member, IEEE) received the M.S. and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 2002 and 2007, respectively. He has been a Senior Researcher with the Center for Machine Vision and Signal Analysis, University of Oulu, Oulu, Finland, since 2007. In 2012 and 2015, he visited the Computer Vision Laboratory, University of Maryland, College Park, MD, USA, and the School of Electrical and Computer Engineering, Duke University, Durham, NC, USA, respectively. He has been with the Peng Cheng Laboratory, Shenzhen, China, since 2018. His research interests include pattern recognition, computer vision, machine learning, dynamic texture, deep learning, and medical image analysis.

Dr. Chen was the Co-Chair of International Workshops at Asian Conference on Computer Vision (ACCV), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), and International Conference on Computer Vision (ICCV). He was a Guest Editor of special issues for IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI) and International Journal of Computer Vision (IJCV).

Yonghong Tian (Fellow, IEEE) is currently a Boya Distinguished Professor with the Department of Computer Science and Technology, Peking University, Beijing, China, and also the Deputy Director of the Artificial Intelligence Research Center, Peng Cheng Laboratory, Shenzhen, China. He has authored or coauthored over 200 technical articles in refereed journals, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), IEEE TRANSACTIONS ON MULTIMEDIA (TMM), IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS (TPDS), the ACM Computing Surveys (CSUR), ACM Transactions on Information Systems (TOIS), ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), and conferences, such as Neural Information Processing Systems (NeurIPS), the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), the International Conference on Computer Vision (ICCV), Association for the Advancement of Artificial Intelligence Conference (AAAI), the ACM International Conference on Multimedia, and the International Worldwide Web Conference. His research interests include neuromorphic vision, brain-inspired computation, and multimedia big data.

Prof. Tian is a Senior Member of the Chinese Institute of Electrical Engineering (CIE) and China Computer Federation (CCF) and a member of the ACM. He was a recipient of the Chinese National Science Foundation for Distinguished Young Scholars in 2018, the two National Science and Technology Awards, and the three ministerial-level awards in China, the 2015 EURASIP Best Paper Award for the Journal on Image and Video Processing, and the Best Paper Award of the IEEE International Conference on Multimedia Big Data (BigMM) in 2018. He has been an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT) since 2018, the IEEE Multimedia Magazine since January 2018, and the IEEE ACCESS since January 2017. He was an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA (TMM) from August 2014 to August 2018. He co-initiated the IEEE BigMM and served as the Technical Program Committee (TPC) Co-Chair for BigMM 2015. He also served as the Technical Program Co-Chair for the IEEE International Conference on Multimedia and Expo (ICME) 2015, the IEEE ISM 2015, and the IEEE International Conference on Multimedia Information Processing and Retrieval (MIPR) 2018 and 2019, and the General Co-Chair of the IEEE MIPR 2020 and ICME 2021. He has been a steering member of the IEEE ICME since 2015 and the IEEE BigMM since 2015, and a TPC Member of more than ten conferences, such as the Conference on Computer Vision and Pattern Recognition (CVPR), the International Conference on Computer Vision (ICCV), the ACM MULTIMEDIA Information Processing and Retrieval (MIPR) 2018 and 2019, and the General Co-Chair of the IEEE Multimedia Conference (MM) 2018 and 2019. He is currently a Senior Editor of ACM/IEEE TRANSACTIONS, including TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI) and International Journal of Computer Vision (IJCV).

Rongrong Ji (Senior Member, IEEE) is currently a Nanqiang Distinguished Professor and the Deputy Director of the Office of Science and Technology, Xiamen University, Xiamen, China, and the Director of Media Analytics and Computing Laboratory. He has published more than 50 papers in ACM/IEEE TRANSACTIONS, including TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI) and International Journal of Computer Vision (IJCV), and more than 100 full papers in top-tier conferences, such as IEEE Conference on Computer Vision and Pattern Recognition (CVPR) and Neural Information Processing Systems (NeurIPS). His publications have got over 10K citations in Google Scholar. His research interests include computer vision, multimedia analysis, and machine learning.

Prof. Ji was awarded as the National Science Foundation for Excellent Young Scholars in 2014, the National Ten Thousand Plan for Young Top Talents in 2017, and the National Science Foundation for Distinguished Young Scholars in 2020. He was the recipient of the Best Paper Award of ACM Multimedia 2011. He has served as the Area Chair in top-tier conferences, such as IEEE Conference on Computer Vision and Pattern Recognition (CVPR) and ACM Multimedia. He is also an Advisory Member for Artificial Intelligence Construction in the Electronic Information Education Committee of the National Ministry of Education.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.