Class-Discriminative CNN Compression

Yuchen Liu, David Wentzlaff, S.Y. Kung
Princeton University
{yl16, wentzlaf, kung}@princeton.edu

***Abstract*—Compressing convolutional neural networks (CNNs) by pruning and distillation has received ever-increasing focus. In particular, designing a class-discrimination based approach would be desired as it fits seamlessly with the CNNs training objective. In this paper, we propose class-discriminative compression (CDC), which injects class discrimination in both pruning and distillation to facilitate the CNNs training goal. We first study the effectiveness of a group of discriminant functions for channel pruning, where we include well-known single-variate binary-class statistics like Student's T-Test in our study via an intuitive generalization. We then propose a novel layer-adaptive hierarchical pruning approach, where we use a coarse class discrimination scheme for early layers and a fine one for later layers. This method naturally accords with the fact that CNNs process coarse semantics in the early layers and extract fine concepts at the later. Moreover, we leverage discriminant component analysis (DCA) to distill knowledge of intermediate representations in a subspace with rich discriminative information, which enhances hidden layers' linear separability and classification accuracy of the student. Combining pruning and distillation, CDC is evaluated on CIFAR and ILSVRC-2012, where we consistently outperform the state-of-the-art results.**

I. Introduction

Convolutional neural networks (CNNs) have become a mainstream model for various computer vision tasks, such as image classification [46], [13], object detection [8], [42], and semantic segmentation [36], [44]. To gain better recognition performance, a popular approach is to grow deeper and wider models. However, such CNNs require a larger storage space and higher computational cost, making them unsuitable for edge devices like mobile phones and embedded sensors.

Many methods have been proposed for CNN compression. For example: weight quantization [1], [3], tensor low-rank factorization [20], [25], network pruning [12], [52], and knowledge distillation [19], [43]. Among them all, a combination of channel pruning and knowledge distillation is the preferable method to learn smaller dense models, which can easily leverage Basic Linear Algebra Subprograms (BLAS) libraries [27].

While CNNs are fundamentally trained to differentiate objects from different classes, the study of discrimination based network compression is quite limited. Prior class-discriminative pruning works [39], [52], [30], [24], [48] lack effectiveness study for their pruning metrics, where they propose and evaluate their metrics singly without comparing to other well-known discriminant functions, like Maximum Mean Discrepancy [10]. Besides, these works ignore the hierarchical nature of CNNs' semantic extraction and only use fine class discrimination for both early and later layers, which could be sub-optimal. For knowledge distillation, while the output distillation scheme [19], [49] normally incorporates class discrepancy knowledge, intermediate discriminative distillation has rarely been attempted.

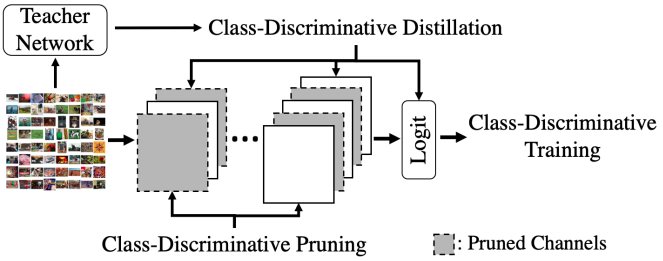


Fig. 1: We propose a novel compression scheme, class-discriminative compression (CDC), which leverages class discrepancy for channel pruning and knowledge distillation, fitting seamlessly with CNN class-discriminative training.

To this end, we propose a novel approach to compress classification CNNs, dubbed class-discriminative compression (CDC), in Fig. 1. We design a unified framework for class-discriminative training, pruning, and distillation, which all aim to improve the final recognition performance.

We first study a group of closed-form functions to find the best metric for class-discriminative channel pruning. This group includes high-dimensional metrics like Maximum Mean Discrepancy (MMD) [10] and single-variate binary-class statistics like Student's T-Test [26], for which we provide an intuitive and lightweight generalization for the high-dimensional multi-class channel scoring. Surprisingly, a generalized metric, generalized Symmetric Divergence (G-SD), achieves the best pruning results. We then propose a novel hierarchical pruning paradigm, which uses a coarse class granularity to evaluate class discrepancy for channels at front layers, and a fine granularity for rear-layer channels. This adaptation is based on the fact that CNNs extract coarse semantics at early layers while understand fine concepts at the later [51], which further improves the pruning results.

Moreover, we make the first attempt to design a subspace distillation approach to allow the class-discriminative knowledge concentratedly distilled to the student model at intermediate layers. To achieve that, we use discriminant component analysis (DCA) [23], which analytically derives linear weights that transform the layer into a subspace with the most class-discriminative power. This scheme improves student hidden layers' linear separability and achieves a better classification accuracy.

Our contributions are summarized as follows:

(1) We propose a novel framework, class-discriminative

compression (CDC), to learn efficient CNNs. This framework incorporates class discrepancy in both channel pruning and knowledge distillation, which is naturally coherent with the discriminative training objective. (2) We study a group of discriminant functions for discriminative pruning, and propose a layer-adaptive hierarchical pruning scheme, which measures the class discrepancy in different label granularities based on layers' positions. (3) We make the first attempt to distill intermediate knowledge in the subspace with the most class-discriminative power, which enhances linear separability of student's hidden layers and achieves better distillation performance. (4) We evaluate the effectiveness of CDC on CIFAR and ILSVRC-2012. On ILSVRC-2012, our compressed ResNet-50 achieves a top-1 accuracy of 76.89% (0.04% accuracy gain from the baseline) with 44.3% FLOPs reduction, outperforming state of the arts.

II. Related Work

Channel Pruning. Channel pruning is promising to enhance network efficiency [17], [24], [33], [14], [29], [2], [31], [32]. Some works leverage norm statistics of weight parameters [27], [34], [17], feature maps reconstruction losses [18], [37], and ranks of the feature maps [29] to evaluate channel redundancy, without the use of discriminative information.

In line with our work, several discrimination-based pruning methods are proposed [39], [52], [30], [48], [24]. While [39], [30], [48] use Taylor expansion to estimate the accuracy/entropy loss of dropping a channel, which require approximation and back-propagation, our metric doesn't need either of them, making it more efficient in evaluation. Zhuang et al. [52] and Kung et al. [24] use entropy loss and closed-form functions that with time-consuming optimization and heavy matrix operations for channel scoring. In contrast, our metric doesn't need them, speeding up the evaluation.

Moreover, while prior works solely use single label granularity for discriminative pruning, we provide a novel hierarchical pruning paradigm where we adapt the label granularity based on the layer's position, which fits seamlessly with the nature of coarse to fine semantic understanding in CNNs.

Knowledge Distillation. Knowledge distillation is pioneered by Hinton et al. [19] to allow a student classifier to mimic the output of its teacher. While Romero et al. [43] propose a hint-layer method and Zagoruyko [50] propose an attention transfer scheme, no class information is distilled in either of their intermediate layers. Different from that, we propose to distill classification information in the subsapce of hidden layers by discriminant component analysis (DCA) [23], where we achieve better distillation results.

Class-Discriminative Analysis. Our work is closely related to techniques for class-discriminative functions like Discriminant Information (DI) [24] and Maximum Mean Discrepancy (MMD) [10]. We also include a set of single-variate binary-class discriminant metrics, Student's T-Test (Ttest) [26], Absolute SNR (AbsSNR) [9], Symmetric Divergence (SD) [38], and Fisher Discriminant Ratio (FDR) [41] in our study. These metrics are originally defined to measure the significance of two class's difference on univariate datasets' for machine learning tasks. For example, SD is used to select discriminative individual features in bioinformatics feature vectors for dimension reduction and efficient classification [38]. However, no prior work has applied them for effective channel pruning.

Discriminant component analysis (DCA) [23] also plays a key role. It can be seen as a multi-class linear discriminant analysis (LDA) and a supervised version of principle component analysis (PCA). It finds components that represent the subspace with the best class linear separability, and we apply it for class-discriminative distillation at intermediate layers.

III. Methodology

A. Discriminant Functions

We formulate channel's class discrepancy evaluation as $\mathcal{M}(\mathcal{F}, \mathcal{Y})$, where $\mathcal{M}$ is the discriminant metric, $\mathcal{F}$ is the set of feature maps obtained at a channel, and $\mathcal{Y}$ is the label numbering scheme. Although discriminant metrics are mathematically well-defined, their empirical pruning effectiveness remains understudied. As shown in Fig. 2, the very first thing we want to know is which discriminant metric works best.

We study a group of closed-form metrics including MMD and DI, and we also generalize four univariate binary-class metrics, T-test, AbsSNR, FDR, and SD, for channel pruning. We use SD [22] as an example to illustrate our generalization method. Let us denote an n-sample 2-class single-variate dataset as $\mathcal{D} = \{(x_i, b_i)\}_{i=1}^{n}$, where $b_i \in \{0, 1\}$ is the binary labeling scheme $\mathcal{B}$. Let $\mathcal{D}^+ = \{x_i \mid (x_i, y_i) \in \mathcal{D},\ y_i = 1\}$ and $\mathcal{D}^- = \{x_i \mid (x_i, y_i) \in \mathcal{D},\ y_i = 0\}$ denote two partitions of $\mathcal{D}$ based on $\mathcal{B}$, SD of $\mathcal{D}$ is defined as:

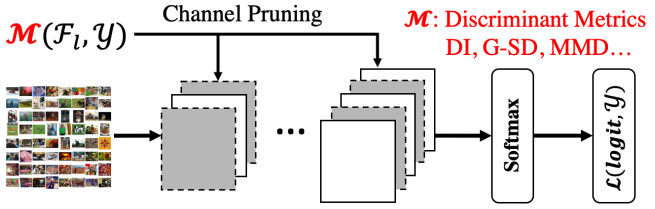
$$\mathrm{SD}(\mathcal{D}, \mathcal{B}) = \frac{1}{2}\left(\frac{\sigma^2_{\mathcal{D}^+}}{\sigma^2_{\mathcal{D}^-}} + \frac{\sigma^2_{\mathcal{D}^-}}{\sigma^2_{\mathcal{D}^+}}\right) + \frac{1}{2}\left(\frac{(\mu_{\mathcal{D}^+} - \mu_{\mathcal{D}^-})^2}{\sigma^2_{\mathcal{D}^+} + \sigma^2_{\mathcal{D}^-}}\right) - 1 \tag{1}$$

where $\mu_{\mathcal{D}^+}$ and $\sigma^2_{\mathcal{D}^+}$ are the sample mean and variance of $\mathcal{D}^+$, and $\mu_{\mathcal{D}^-}$ and $\sigma^2_{\mathcal{D}^-}$ are the statistics for $\mathcal{D}^-$.

For an N-sample Y-class dataset, we denote the feature maps of a channel as $\mathcal{F} = \{(f_i, y_i)\}_{i=1}^{N}$, where $f_i \in \mathbb{R}^{W \times H}$ is the feature map of the i-th input image, $y_i \in [1{:}Y]$ is the i-th class label, and W/H are the spatial sizes. We first partition $\mathcal{F}$ as $\mathcal{F}^c$ and $\mathcal{F}^{-c}$ where $\mathcal{F}^c = \{f_i \mid (f_i, y_i) \in \mathcal{F},\ y_i = c\}$ and $\mathcal{F}^{-c} = \{f_i \mid (f_i, y_i) \in \mathcal{F},\ y_i \neq c\}$, $\forall\ c \in [1{:}Y]$. By this partition, denoted as $\mathcal{B}_c$, we can find the statistics in Eqn. 1 in a two-class manner. Note each f_i in $\mathcal{F}^c$ is a 2D feature map with $W \times H$ activations, and thus there are $|\mathcal{F}^c| \times W \times H$ activations in $\mathcal{F}^c$ in total. We then define two statistics operators g_{mean} and g_{var} on $\mathcal{F}^c$, which return the mean and variance over these $|\mathcal{F}^c| \times W \times H$ activations. We thus get $\mu_c = g_{mean}(\mathcal{F}^c)$ and $\sigma^2_c = g_{var}(\mathcal{F}^c)$ for $\mathcal{F}^c$, and their counterparts μ_{-c} and σ^2_{-c}. The SD score for $\mathcal{B}_c$ is thus:

$$\mathrm{SD}(\mathcal{F}, \mathcal{B}_c) = \frac{1}{2}\left(\frac{\sigma^2_c}{\sigma^2_{-c}} + \frac{\sigma^2_{-c}}{\sigma^2_c}\right) + \frac{1}{2}\left(\frac{(\mu_c - \mu_{-c})^2}{\sigma^2_c + \sigma^2_{-c}}\right) - 1 \tag{2}$$

$\mathrm{SD}(\mathcal{F}, \mathcal{B}_c)$ captures the discriminativenesss of class c relative to the rest of the dataset. In general, we want to select channels that distinguish all classes well on average. Thus, the generalized Symmetric Divergence (G-SD) of $\mathcal{F}$ is:



$\mathcal{F}_l$: Feature Maps at Layer l $\mathcal{Y}$: Class Labels : Pruned Channels

Fig. 2: Find the best discriminant metrics for channel pruning.

$$\text{G-SD}(\mathcal{F}, \mathcal{Y}) = \frac{1}{Y}\sum_{c=1}^{Y}\text{SD}(\mathcal{F}, \mathcal{B}_c) \tag{3}$$

Such generalization method is applicable to other single-variate binary-class metrics, and incurs no expensive operations (e.g., matrix inversion, SVD), which makes the generalized metrics scalable to large networks and datasets.

B. Hierarchical Pruning

While $\mathcal{F}$ is determined by input images and network's weights, $\mathcal{Y}$ can be calibrated for different layers. After finding out the best metric $\mathcal{M}$, we investigate the settings of $\mathcal{Y}$ for more effective discriminative pruning. It is widely recognized that CNN learns coarse semantics (fruit, vehicle) in early layers while extracting finer class (apple, truck) concepts in later layers [51]. Inspired by this nature of hierarchical semantic separation, we propose to adapt the granularity of $\mathcal{Y}$ based on layer positions as shown in Fig. 3. Specifically, we define a watershed layer l_{WS}, where we evaluate the channel discrepancy by a coarse class label $\mathcal{M}(\mathcal{F}, \mathcal{Y}_c)$ when $l \leq l_{WS}$, and use fine label pruning $\mathcal{M}(\mathcal{F}, \mathcal{Y}_f)$ when $l > l_{WS}$.

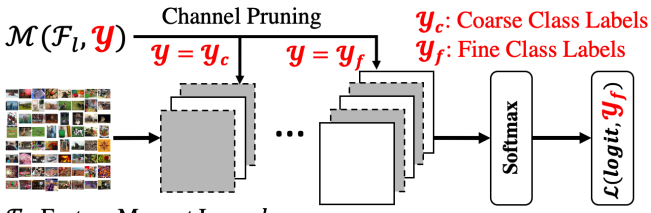
While most image datasets only provide fine class labels $\mathcal{Y}_f \in [1{:}F]$, the coarser labeling scheme $\mathcal{Y}_c \in [1{:}C]$ need to be derived on our own. To tackle this issue, we use a pretrained CNN Net to group similar fine categories into the same coarse category, i.e., to learn a disjoint many-to-one mapping $Q : [1{:}F] \rightarrow [1{:}C]$. We investigate two methods.

Clustering on Class Centroids. We randomly sample a held-out set of images from the training set. We feed these images through Net and get the last hidden activations. We calculate the activations' class centroid for each fine label, denoted as $\mathcal{H} = \{h_1, h_2, ..., h_F\}$, and run a K-means clustering on $\mathcal{H}$ with C clusters to get the mapping Q.

Clustering on Confusion Matrix. We feed the held-out images through Net to get their predicted labels. Based on the predicted and true labels, we construct a confusion matrix $M \in \mathbb{R}^{F\times F}$, where $M_{i,j}$ denotes the number of images with true label i but predicted as label j. We then run a spectral clustering on M with C clusters which gives us Q.

C. Intermediate Class Discrepancy Distillation

We then retrain pruned nets with a combined loss of cross entropy $\mathcal{L}_{CE}$ and knowledge distillation to recover their accuracies. In particular, we propose to distill classification information at intermediate layers' subspaces found by discriminant component analysis (DCA) [23] in Fig. 4. We experimentally compare the DCA-based distillation with the hint layer [43] in Sec. IV-C, and find that distillation at the subspace with rich class information results in better performance.



$\mathcal{F}_l$: Feature Maps at Layer l
$\mathcal{M}$: Discriminant Metrics
: Pruned Channels

Fig. 3: Adapting the label granularity for channels' class discrepancy measurement based on layer positions.

DCA. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ denote an N-sample Y-class dataset, the within-/between-class scatter matrix $\mathbf{S}_W$/$\mathbf{S}_B$:

$$\mathbf{S}_W = \sum_{y=1}^{Y}\sum_{j=1}^{N_y}(x_j^{(y)} - \bar{x}_y)(x_j^{(y)} - \bar{x}_y)^T \tag{4}$$

$$\mathbf{S}_B = \sum_{y=1}^{Y} N_y(\bar{x}_y - \bar{x})(\bar{x}_y - \bar{x})^T \tag{5}$$

where N_y, $x_j^{(y)}$ are the number of samples and the j-th sample of class y. $\bar{x}_y$ and $\bar{x}$ are the y-class centroid and overall data centroid. Note the center adjusted scatter matrix $\bar{\mathbf{S}} = \mathbf{S}_W + \mathbf{S}_B$. The discriminant components of $\mathcal{D}$ are:

$$W^{DCA} = \underset{W:W^T\mathbf{S}_W W=I}{\arg\max}\ tr(W^T\bar{\mathbf{S}}W) \tag{6}$$

While LDA only finds one component to separate two classes, DCA can be seen as a multi-class version of it where $\mathbf{S}_W$ and $\mathbf{S}_B$ owns information for all classes. DCA's objective is also the same as PCA while it includes an extra within-class matrix $\mathbf{S}_W$ in its constraint, making it a supervised version of PCA. As evidenced in the original paper [23], the DCA subspace is more linearly separable than the one found by PCA.

To learn DCA weights for intermediate layers, we feed the held-out set of images through the network and get its intermediate activation $A \in \mathbb{R}^{N\times C\times H\times W}$, which is reshaped as $A \in \mathbb{R}^{N\times D}$, $D = C \times H \times W$. With a labeling scheme $\mathcal{Y}$, we apply Eqn. 6 to get its top Y components, $W^{DCA} \in \mathbb{R}^{D\times Y}$, as $\mathbf{S}_B$ has a rank of Y.

Distillation. We learn DCA for the teacher, W_T^{DCA}, at the start of training, and learn the student's DCA, W_S^{DCA}, every d epochs. These weights are not updated in the back-propagation, and the loss is constructed as:

$$\mathcal{L}_{KD}^{Inter} = \sum_{l=1}^{L}||A_T(l)W_T^{DCA}(l) - A_S(l)W_S^{DCA}(l)||_1 \tag{7}$$

where $A_T(l), W_T^{DCA}(l)$ denote the activation and DCA weights for layer l of the teacher, and $A_S(l), W_S^{DCA}(l)$ are those for the student. Mathematically speaking, $\mathcal{L}_{KD}^{Inter}$ imposes regularization on student's hidden layers to push the transformed subspace as linearly separable as its teacher.

We adopt the output distillation loss $\mathcal{L}_{KD}^{Out}$ in [19], and our training loss for the student is formally defined as:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda\mathcal{L}_{KD}^{Inter} + \gamma\mathcal{L}_{KD}^{Out} \tag{8}$$

where λ, γ are the weights for distillation losses.

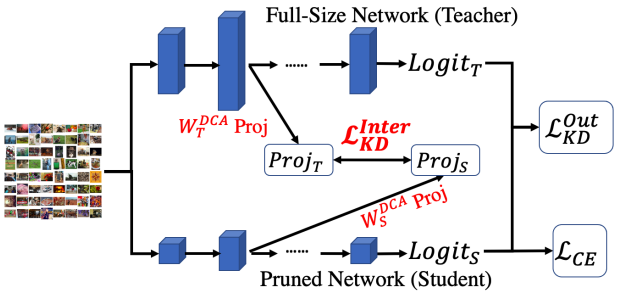


Fig. 4: DCA-based intermediate distillation, which allows class discrepancy to distill at hidden layers.

IV. Experimental Results

We study each part of CDC, i.e., G-SD, hierarchical pruning, and DCA distillation on CIFAR [21]. We then compare CDC with the state-of-the-art approaches on ILSVRC-2012 [4]. More results and studies are in Supplementary.

A. Function Effectiveness Study

We conduct one-shot pruning tests with metrics in Sec. III-A for VGG-16 on CIFAR-10 and ResNet-38 on CIFAR-100, with results shown in Fig. 5. For each metric, we uniformly prune 10%, 20%, 30%, and 40% of the least discriminative channels (measured by fine labels) in each layer. These pruned models are fine-tuned by $\mathcal{L}_{CE}$ only with the same training parameters. We include random pruning as the baseline.

The discriminant metrics outperform random pruning in all ratios, clearly indicating their effectiveness, where G-SD consistently achieves the best results. Without retraining, G-SD outperforms DI by 5.5% accuracy on CIFAR-10 with 40% of channels removed and has an 8% accuracy gain over MMD on CIFAR-100 with 30% of channels removed. With retraining, G-SD gains 0.5% accuracy over MMD on CIFAR-100 when 30% channels removed. Based on such consistent winning results, we adopt G-SD as the pruning metric in our CDC pipeline. We also visually demonstrate the advantage of G-SD over other state-of-the-art pruning criteria in Fig. 7.

B. Hierarchical Pruning Study

To study the hierarchical pruning (HP) scheme, we use G-SD to uniformly remove 45% channels from a ResNet-164 on CIFAR-100 (its ground truth coarse label well fits the study) and retrain it by $\mathcal{L}_{CE}$ only.

Effectiveness. We compare HP with three other labeling schemes for class discrepancy measurement. All four pruning modes are: (1) All layer fine label $\mathcal{Y}_f$ pruning. (2) All layer coarse label $\mathcal{Y}_c$ pruning. (3) Front layer $\mathcal{Y}_f$ + rear layer $\mathcal{Y}_c$ pruning. (4) Front layer $\mathcal{Y}_c$ + rear layer $\mathcal{Y}_f$ pruning (**HP**). We use the ground truth coarse label for $\mathcal{Y}_c$, and set the watershed layer $l_{WS} = 0.5L$, where L is the total number of layers in the network. As shown in Fig. 6a, HP (**Mode 4**) achieves the best accuracy among all schemes. This suggests that the channels' class discrimination shall be measured based on its semantic granularity for better pruning performance.

Watershed Layer. We varies the placement of the watershed layer, parameterized by $l_{WS} = \alpha L, \alpha \in (0, 1)$. As shown in Fig. 6b, we find $\alpha = 0.5$ gives the best result, suggesting that the CNN processes coarse semantics in the first half of the layers, and extracts finer concepts in the second half.

TABLE I: G-SD hierarchical pruning (HP) outperforms state-of-the-art pruning methods.

Network	Method	Test Acc. (%)	Acc. ↓	FLOPs (M) Pruned (%)
ResNet 164	LCCL [6]	75.67 → 75.26	0.41	197 (21.3)
	SLIM [34]	76.63 → 76.09	0.54	124 (50.6)
	DI [24]	77.63 → 76.11	1.52	105 (58.0)
	HP	**78.05 → 77.77**	**0.28**	**92 (63.2)**

Class Hierarchies. While most image datasets don't have ground truth coarse labels, we further evaluate proposed HP using the coarse labels learned by the clustering algorithms in Sec. III-B, and set $l_{WS} = 0.5L$. We set the number of learned coarse classes to be 20 (same as the ground truth scheme). As shown in Fig. 6c, the coarse class labels learned by spectral clustering on the accuracy confusion matrix could even outperform the ground truth scheme. This indicates that HP is effective even without the ground truth coarse label. We study multiple coarse levels HP in Supplementary.

Compared to State of the Arts. We compare G-SD HP scheme with state-of-the-art pruning methods in Tab. I, where it outperforms all of them. We achieve a 2.51% accuracy gain over LCCL [6] with 41.9% less FLOPs. Compared to DI [24], we achieve 1.66% higher accuracy and 5.2% less FLOPs.

C. Intermediate Distillation Study

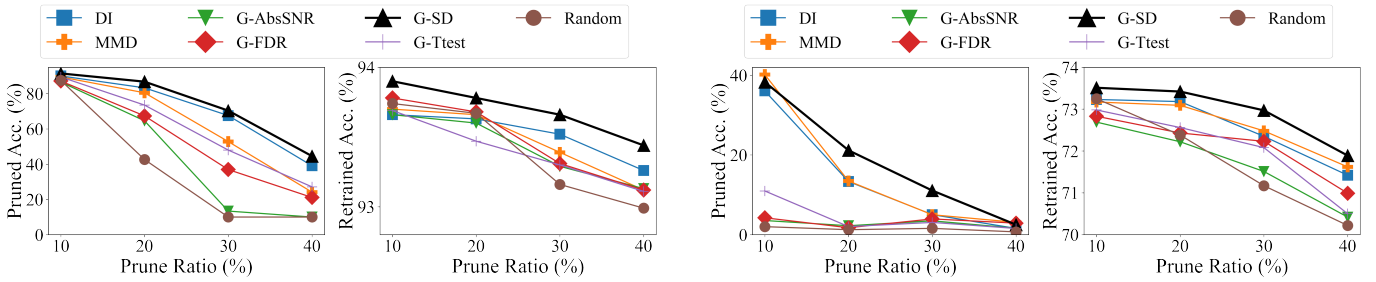
We then combine $\mathcal{L}_{CE}$ with different distillation losses to retrain HP-pruned ResNet-164 with results in Fig. 6d. We investigate the following modes with similar computational budgets: (1) No distillation. (2) Only output distillation [19]. (3) Output + hint-layer intermediate distillation [43]. (4) Output + $\mathcal{Y}_f$ DCA intermediate distillation. (5) Output + $\mathcal{Y}_c$ DCA intermediate distillation.

We set $\lambda = 10.0$ and $\gamma = 1.0$ in Eqn. 8 and we only insert intermediate loss at the watershed layer for all intermediate distillation schemes. We include study on inserting losses at multiple intermediate layers in Supplementary Material. We find adding hint-layer distillation (**Mode 3**) does not improve over output only distillation (**Mode 2**). On the contrary, DCA-based distillation (**Mode 4-5**) improves the distillation quality, where using the coarse label $\mathcal{Y}_c$ to learn DCA weights (**Mode 5**) achieves the best results, which again emphasizes that we should adopt the class granularity of the layer for class discrepancy analysis. These results demonstrate the advantage of DCA-based intermediate distillation over the hint layer [43].

Combining HP with DCA-based distillation, the 2.7×-accelerated ResNet-164 derived by our CDC pipeline achieves an accuracy of 78.05% on CIFAR-100, further advancing the state-of-the-art compression results.

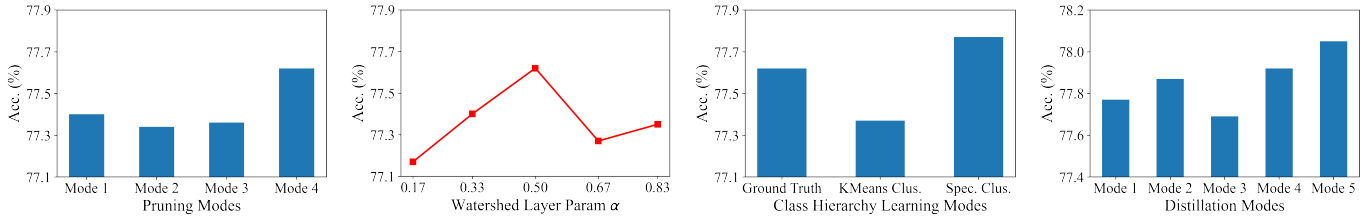
D. Comparing to State of the Arts on ILSVRC-2012

Benchmarks. Combing HP and DCA-based distillation, we evaluate CDC on ILSVRC-2012 [4] (models at different compression levels are suffixed with letters, e.g. "CDC-A"



(a) VGG-16 on CIFAR-10 (b) ResNet-38 on CIFAR-100

Fig. 5: Empirical study on the discriminant functions' effectiveness. The functions are used to prune VGG-16 and ResNet-38, whose pruned acc. (w/o retraining) and retrained acc. (w/ retraining) are shown. We find G-SD outperforms others in all tasks.



(a) Investigation of Label Pruning Modes (b) Choice of Watershed Layer (c) Effects of Different Class Hierarchies (d) Effects of Different Knowledge Distillation.

Fig. 6: Ablation study on hierarchical pruning (HP) (Fig. a,b,c) and DCA distillation (Fig. d). (a) The front-layer $\mathcal{Y}_c$ + rear-layer $\mathcal{Y}_f$ HP scheme outperforms other label placement schemes. Notably, HP largely improves over the all-layer fine label pruning used in all other literatures. (b) Setting $\alpha = 0.5$, i.e., setting the center layer as the watershed layer gives the best performance for HP. (c) The coarse label learned by spectral clustering on the confusion matrix outperforms the ground truth, suggesting HP can be effective even without ground truth coarse labels. (d) DCA-based distillation improves over the hint-layer distillation [43], where using the coarse label for DCA learning achieves the best result.

and "CDC-B"). We compare various state-of-the-art compression methods (including baseline acc.), e.g., TAS [7], LeGR [2], DMCP [11], SSR-GR [47], and CC [28]. While some approaches only use channel pruning for compression, we include output knowledge distillation [19] for them in our own implementation (+KD) for a fairer comparison.

Training Settings. We use Nesterov SGD [40] with a momentum of 0.9, and a weight decay of 0.0001. We use the standard data augmentation scheme [13] with a batch size of 128 to fine-tune 100 epochs. The learning rate is started at 0.025 with a cosine decay learning rate schedule.

CDC Settings. We adopt the same one-shot uniform hierarchical pruning scheme[1] as in Sec. IV-B. 10,000 training images are randomly sampled for G-SD scoring and we learn 100 coarse classes by spectral clustering on the confusion matrix. We set l_{WS} at the middle layer of the net. For distillation, we set $\lambda = 10.0$ and $\gamma = 1.0$ and learn DCA weights by coarse class with the intermediate loss inserted at l_{WS}. The student DCA is updated at 40% and 80% of the total epochs.

Results. As shown in Tab. II, CDC outperforms all prior arts. On ResNet-50, CDC-A has a 0.69% top-1 accuracy gain compared to DMCP [11] and TAS [5] which also leverages knowledge distillation. CDC-B achieves a top-1 accuracy of 76.35% with 53.5% FLOPs reduction, surpassing all prior methods. On ResNet-18, CDC-A achieves a 3.06% accuracy gain with a higher FLOPs reduction compared to LCCL [6], while CDC-B demonstrates top-1 accuracy gains of 1.51% and 1.32% with respect to DCP [52] and SFP (+KD) [15]. On a more compact MobileNet-V2, CDC-A achieves a top-1 accuracy of 71.97% with 26.9% FLOPs reduction, outperforming AMC [16], Meta (+KD) [2], and CC [28]. CDC-B advances DCP [52] and Meta (+KD) [35] by 5% and 0.76% top-1 accuracy, when 53.4% of FLOPs are pruned.

[1]For pruned MobileNet-V2, we round the number of channels to its closest integer that is divisible by 8 in each layer, as suggested in [45].

V. Visualization

In Fig. 7, we provide additional visualization with channels at `Res1_2` in ResNet-50 on ILSVRC-2012 to intuitively show the effectiveness of our approach on selecting informative channels. In **Col. 1-3**, we observe that the channel with low G-SD (**Col. 2**) generates indistinguishable responses across classes, while the high one (**Col. 3**) well preserves the informative patterns for classification. We further compare G-SD with SOTA criteria [27], [34], [17], [24], by computing an average response over the top-10 highest scored channels for in **Col. 4-8**. We observe the average responses of G-SD (**Col. 8**) tends to display more class information than the others (**Col. 4-7**). G-SD clearly separates the ostrich from the background grass in the first row, and it is the only one that preserves both the vertical nail and its long diagonal shadow in the second row.

VI. Conclusion

In this paper, we propose class-discriminative compression (CDC) to learn efficient neural networks. While limited attempts have been made to leverage classification information for network compression, we design a unified framework for

TABLE II: Comparing to state of the arts on ILSVRC-2012. "Use KD" indicates whether the method leverages distillation and "+KD" refers to our own implementation on SOTA methods with distillation. Other numbers are from original papers.

Network	Method	Use KD	Top-1 Acc. (%)	Top-1 ↓ (%)	Top-5 Acc. (%)	Top-5 ↓ (%)	FLOPs (B) Pruned (%)	Params (M) Pruned (%)
ResNet 50	HRank [29]	✗	76.15 → 74.98	1.17	92.87 → 92.33	0.54	2.30 (43.7)	16.2 (36.5)
	HRank [29] + KD	✓	76.15 → 75.30	0.85	92.87 → 92.50	0.37	2.30 (43.7)	16.2 (36.5)
	SSR-GR [47]	✗	76.13 → 75.76	0.37	92.86 → 92.67	0.19	2.29 (44.1)	-
	TAS [7]	✓	77.46 → 76.20	1.26	93.55 → 93.07	0.48	2.31 (43.5)	-
	DMCP [11]	✗	76.60 → 76.20	0.40	-	-	2.20 (46.0)	-
	CDC-A	✓	**76.85 → 76.89**	**-0.04**	**93.17 → 93.33**	**-0.16**	**2.28 (44.3)**	**14.9 (41.6)**
	FPGM [17]	✗	76.15 → 74.83	1.32	92.87 → 92.32	0.55	1.90 (53.5)	-
	GBN [48]	✗	75.85 → 75.18	0.67	92.67 → 92.41	0.26	1.85 (55.0)	11.9 (53.4)
	LeGR [2]	✗	76.10 → 75.30	0.80	92.90 → 92.40	0.50	1.93 (53.0)	-
	LeGR [2] + KD	✓	76.10 → 75.45	0.65	92.90 → 92.52	0.38	1.93 (53.0)	-
	CDC-B	✓	**76.85 → 76.35**	**0.50**	**93.17 → 93.04**	**0.13**	**1.90 (53.5)**	**12.7 (50.3)**
ResNet 18	LCCL [6]	✗	69.98 → 66.33	3.65	89.24 → 86.94	2.30	1.18 (34.6)	11.7 (0.0)
	CDC-A	✓	**70.05 → 69.39**	**0.66**	**89.40 → 88.81**	**0.59**	**1.15 (36.5)**	**7.3 (37.0)**
	SFP [15]	✗	70.28 → 67.10	3.18	89.63 → 87.78	1.85	1.06 (41.8)	-
	SFP [15] + KD	✓	70.28 → 67.58	2.70	89.63 → 88.01	1.62	1.06 (41.8)	-
	DCP [52]	✗	69.64 → 67.35	2.29	88.98 → 87.60	1.38	0.98 (46.0)	6.2 (47.0)
	FPGM [17]	✗	70.28 → 68.41	1.87	89.63 → 88.48	1.15	1.06 (41.8)	-
	CDC-B	✓	**70.05 → 68.86**	**1.19**	**89.40 → 88.61**	**0.79**	**1.05 (41.9)**	**6.7 (42.5)**
MobileNet V2	AMC [16]	✗	71.80 → 70.80	1.00	-	-	0.22 (26.9)	-
	CC [28]	✗	71.88 → 70.91	0.97	-	-	0.22 (28.3)	-
	Meta [35]	✗	72.70 → 71.20	1.50	-	-	0.22 (27.9)	-
	Meta [35] + KD	✓	72.70 → 71.44	1.26	-	-	0.22 (27.9)	-
	CDC-A	✓	**72.18 → 71.97**	**0.21**	**90.49 → 90.39**	**0.10**	**0.22 (26.9)**	**2.8 (20.4)**
	DCP [52]	✗	70.11 → 64.22	5.89	-	3.77	0.17 (44.7)	2.6 (25.9)
	Meta [35]	✗	72.70 → 68.20	4.50	-	-	0.14 (53.4)	-
	Meta [35] + KD	✓	72.70 → 68.48	4.22	-	-	0.14 (53.4)	-
	CDC-B	✓	**72.18 → 69.22**	**1.96**	**90.49 → 88.69**	**1.80**	**0.14 (53.4)**	**2.1 (39.3)**

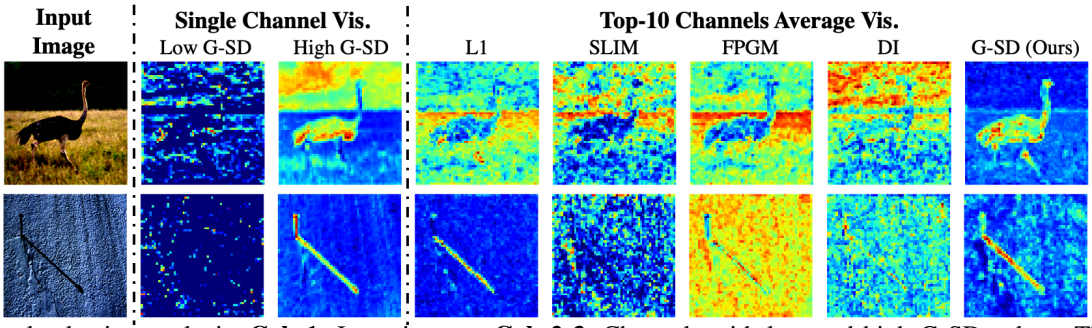


Fig. 7: Channel selection analysis. **Col. 1:** Input images. **Col. 2-3**: Channels with low and high G-SD values. The low G-SD channel generates indistinguishable responses, while the high one produces informative activations. **Col. 4-8**: Average responses of the top-10 channels. From left to right, the metrics are: $\ell 1$-weight [27], batch-norm scaling factor [34], filter's geometric median [17], DI [24], and G-SD. In general, the channels picked by G-SD preserves the most classification information.

discriminative pruning and distillation, fitting seamlessly with the discriminative training objective. To better identify channels' redundancy for class-discriminative pruning, we study the pruning effectiveness of a group of closed-form discriminant functions and propose a hierarchical pruning paradigm. Moreover, we make the first attempt to distill discriminative information in hidden layers' subspace by discriminant component analysis. Combining the pruning and distillation approaches, CDC outperforms state-of-the-art methods by a clear margin on CIFAR and ILSVRC-2012.

ACKNOWLEDGEMENT

This material is based on research sponsored by the National Science Foundation (NSF) under Grant No. CCF-1822949, Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) under agreement No. FA8650-18-2-7862. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA), the National Science Foundation, or the U.S. Government.

REFERENCES

[1] Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *International Conference on Machine Learning*, pages 2285–2294, 2015. 1

[2] Ting-Wu Chin, Ruizhou Ding, Cha Zhang, and Diana Marculescu. Towards efficient model compression via learned global ranking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1518–1528, 2020. 2, 5, 6

[3] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016. 1

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. 2009. 4

[5] Xin Dong, Shangyu Chen, and Sinno Pan. Learning to prune deep neural networks via layer-wise optimal brain surgeon. In *Advances in Neural Information Processing Systems*, pages 4857–4867, 2017. 5

[6] Xuanyi Dong, Junshi Huang, Yi Yang, and Shuicheng Yan. More is less: A more complicated network with less inference complexity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5840–5848, 2017. 4, 5, 6

[7] Xuanyi Dong and Yi Yang. Network pruning via transformable architecture search. In *Advances in Neural Information Processing Systems*, pages 759–770, 2019. 5, 6

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1

[9] Todd R Golub, Donna K Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P Mesirov, Hilary Coller, Mignon L Loh, James R Downing, Mark A Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537, 1999. 2

[10] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012. 1, 2

[11] Shaopeng Guo, Yujie Wang, Quanquan Li, and Junjie Yan. Dmcp: Differentiable markov channel pruning for neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1539–1547, 2020. 5, 6

[12] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, 2015. 1

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 5

[14] Yang He, Yuhang Ding, Ping Liu, Linchao Zhu, Hanwang Zhang, and Yi Yang. Learning filter pruning criteria for deep convolutional neural networks acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2009–2018, 2020. 2

[15] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. Soft filter pruning for accelerating deep convolutional neural networks. *arXiv preprint arXiv:1808.06866*, 2018. 5, 6

[16] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–800, 2018. 5, 6

[17] 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. 2, 5, 6

[18] 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. 2

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 2, 3, 4, 5

[20] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014. 1

[21] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 4

[22] S.Y. Kung. *Kernel methods and machine learning*. Cambridge University Press, 2014. 2

[23] S.Y. Kung. Discriminant component analysis for privacy protection and visualization of big data. *Multimedia Tools and Applications*, 76(3):3999–4034, 2017. 1, 2, 3

[24] S.Y. Kung, Zejiang Hou, and Yuchen Liu. Methodical design and trimming of deep learning networks: Enhancing external bp learning with internal omnipresent-supervision training paradigm. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8058–8062. IEEE, 2019. 1, 2, 4, 5, 6

[25] Vadim Lebedev, Yaroslav Ganin, Maksim Rakhuba, Ivan Oseledets, and Victor Lempitsky. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. *arXiv preprint arXiv:1412.6553*, 2014. 1

[26] Erich L Lehmann and Joseph P Romano. *Testing statistical hypotheses*. Springer Science & Business Media, 2006. 1, 2

[27] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016. 1, 2, 5, 6

[28] Yuchao Li, Shaohui Lin, Jianzhuang Liu, Qixiang Ye, Mengdi Wang, Fei Chao, Fan Yang, Jincheng Ma, Qi Tian, and Rongrong Ji. Towards compact cnns via collaborative compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6438–6447, 2021. 5, 6

[29] 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. 2, 6

[30] Shaohui Lin, Rongrong Ji, Yuchao Li, Yongjian Wu, Feiyue Huang, and Baochang Zhang. Accelerating convolutional networks via global & dynamic filter pruning. In *IJCAI*, pages 2425–2432, 2018. 1, 2

[31] Yuchen Liu, SY Kung, and David Wentzlaff. Evolving transferable pruning functions. *arXiv preprint arXiv:2110.10876*, 2021. 2

[32] Yuchen Liu, Zhixin Shu, Yijun Li, Zhe Lin, Federico Perazzi, and Sun-Yuan Kung. Content-aware gan compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12156–12166, 2021. 2

[33] Yuchen Liu, David Wentzlaff, and Sun-Yuan Kung. Rethinking class-discrimination based cnn channel pruning. *arXiv preprint arXiv:2004.14492*, 2020. 2

[34] 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. 2, 4, 5, 6

[35] 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/CVF International Conference on Computer Vision*, pages 3296–3305, 2019. 5, 6

[36] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[37] 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. 2

[38] Man-Wai Mak and S.Y. Kung. A solution to the curse of dimensionality problem in pairwise scoring techniques. In *International Conference on Neural Information Processing*, pages 314–323. Springer, 2006. 2

[39] P Molchanov, S Tyree, T Karras, T Aila, and J Kautz. Pruning convolutional neural networks for resource efficient inference. In *5th International Conference on Learning Representations, ICLR 2017-Conference Track Proceedings*, 2017. 1, 2

[40] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983. 5

[41] Paul Pavlidis, Jason Weston, Jinsong Cai, and William Noble Grundy. Gene functional classification from heterogeneous data. In *Proceedings of the fifth annual international conference on Computational biology*, pages 249–255, 2001. 2

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 1

[43] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 1, 2, 3, 4, 5

[44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International*

Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015. 1

[45] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 5

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[47] Zi Wang, Chengcheng Li, and Xiangyang Wang. Convolutional neural network pruning with structural redundancy reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14913–14922, 2021. 5, 6

[48] Zhonghui You, Kun Yan, Jinmian Ye, Meng Ma, and Ping Wang. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1, 2, 6

[49] Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13876–13885, 2020. 1

[50] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 2

[51] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 1, 3

[52] 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 883–894, 2018. 1, 2, 5, 6