IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 2, 2022, accepted February 17, 2022, date of publication February 21, 2022, date of current version March 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3153025

Automated Filter Pruning Based on High-Dimensional Bayesian Optimization

TAEHYEON KIM, (Graduate Student Member, IEEE), HEUNGJUN CHOI, AND YOONSIK CHOE, (Senior Member, IEEE)
Department of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, South Korea

Corresponding author: Yoonsik Choe (yschoe@yonsei.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant by the Korean Government through the Ministry of Science and ICT (MSIT) under Grant 1711108458.

ABSTRACT Filter pruning is necessary to efficiently deploy convolutional neural networks on edge devices that have limited computational resources and power budgets. With conventional filter pruning techniques, the same pruning rate is manually specified for different convolutional layers, which is suboptimal and time-consuming. To extract the features from the coarse level to the fine level, the number of filters in each layer has various distributions. Therefore, it is unsuitable to utilize the same pruning rate for different functional layers. To address this issue, we propose a high-dimensional Bayesian optimization-based filter pruning (HDBOFP) algorithm, which aims to automatically determine the most appropriate pruning rate for each convolutional layer. In addition, the proposed method can automatically identify optimal pruning-rate combinations without a time-consuming retraining phase. Compared with conventional filter pruning methods, this automated filter pruning technique exhibits a higher efficiency, which improves accuracy and reduces the required human labor. The effectiveness of our automated filter pruning algorithm is validated through two major computer vision applications, namely image classification and object detection. Specifically, when used in combination with ResNet-110 to classify the CIFAR-10 dataset, HDBOFP reduces the required number of float point operations (FLOPs) by more than 62% without affecting accuracy. Similarly, when HDBOFP is added to the YOLOv5l framework to run detection experiments on the MS-COCO 2017 dataset, FLOPs decrease by more than 43% with only a 1.2% loss in mean average precision, which has advanced the previous studies.

INDEX TERMS Bayesian optimization, convolutional neural network acceleration, dimensionality reduction, filter pruning, hyper-parameter optimization.

I. INTRODUCTION

Convolutional neural networks (CNNs) with large model sizes and heavy computational costs have achieved remarkable performance in various computer vision research applications; however, it is difficult to deploy these CNNs on edge devices, due to their limited computational resources and power budgets. Even state-of-the-art high-efficiency architectures, such as residual connections or inception modules, have millions of parameters and require billions of float point operations (FLOPs). Therefore, it is necessary to develop CNNs that can deliver high accuracies with a relatively low computational cost. Many recent studies have attempted to enable CNNs to use hardware resources more efficiently, which has resulted in various model compression strategies, including pruning [3], [4], tensor decomposition [14], [18], quantization [19], and knowledge distillation [20]. Among these strategies, pruning has attracted special attention owing to its effective performance and implementation.

Recent developments in pruning can be divided into two categories: weight pruning and filter pruning. Weight pruning aims to remove redundant elements in the weight of filters, which can achieve high model efficiency with no loss of accuracy. However, these algorithms result in an irregular sparsity pattern and require specialized hardware for speeding up. Filter pruning, however, aims to crop all unimportant filters. Thus, the pruned filter weights are regular and can be directly accelerated using off-the-shelf hardware and libraries. In this

The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Alawneh.

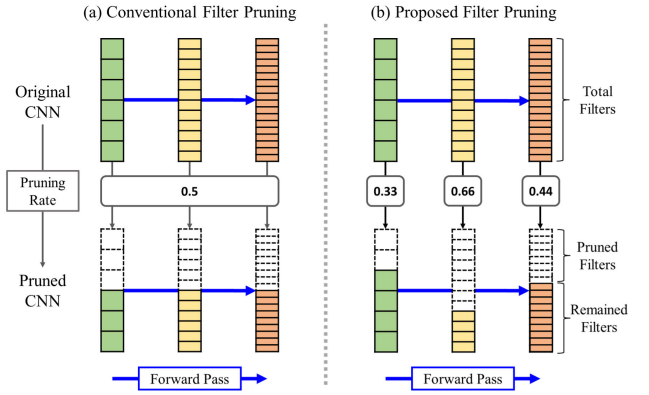


FIGURE 1. Conventional vs proposed filter pruning. (a) Conventional filter pruning methods manually select a pruning rate for all layers. (b) The proposed pruning method provides an appropriate pruning rate for each layer based on high-dimensional Bayesian optimization. In the pruned CNN, the filters without color denote pruned filters, and those with color are the remaining filters.

study, we investigate filter pruning, which is the preferred method for CNN compression. Conventional filter pruning algorithms comprise three steps: (1) training a large CNN on the target dataset; (2) pruning redundant filters from the pre-trained CNN based on a particular pruning rate and pruning criteria, and (3) retraining the pruned CNN to recover its original performance.

The fundamental task in filter pruning is determining the pruning policies, which include the pruning rate and pruning criteria. Determining the appropriate pruning rate for each layer is a crucial research point; however, many recent studies have focused on measuring the importance of each filter. Most existing methods manually specify a single pruning rate for different layers, as illustrated in Figure 1(a). Some previous studies have proposed a rule-based algorithm to determine different pruning rates for each convolutional layer. However, because the convolutional layers in deep CNNs are not independent, these hand-crafted rule-based pruning rate selection algorithms are typically suboptimal and time-consuming, and do not transfer well from one model to another. CNN architectures are evolving rapidly, and an automated compression method is necessary to improve engineering efficiency. Therefore, we propose a novel automated filter pruning method that automatically determines the appropriate pruning rate combination for an arbitrary network, while achieving better performance than conventional human-designed rule-based filter pruning methods. Figure 1(b) illustrates the proposed algorithm.

To attain the optimal pruning rate combination during the automated phase, it is necessary to evaluate the accuracy degradation of every pruning rate combination considered. However, as it is typically performed through a time-consuming retraining step, we require a more efficient method to measure accuracy degradation. Therefore, we propose an objective function that utilizes a soft filter pruning algorithm to estimate the expected accuracy degradation for a given pruning rate combination. The proposed objective function is based on the idea that the difference between a soft pruned network and its original structure is proportional to the accuracy degradation of the former. The proposed objective function correctly estimates the expected accuracy degradation of a given pruning rate combination; however, it is a non-differentiable and non-convex optimization problem. Therefore, we utilize Bayesian optimization, which is designed for black-box derivative-free global optimization. In addition, as the CNN deepens, the objective function space becomes a high-dimensional large-scale space, which calls for a trade-off between storage complexity, computational cost, and accuracy. Because standard Bayesian optimization suffers from a curse of dimensionality, the use of standard Bayesian optimization limits the applicability of the proposed algorithm to deep CNNs with many convolutional layers. To overcome this limitation, the proposed algorithm utilizes a low-dimensional embedding-based Bayesian optimization, called high-dimensional Bayesian optimization.

In summary, the main contributions of this study are as follows:

- We propose an effective automated filter pruning algorithm, called high-dimensional Bayesian optimization-based filter pruning (HDBOFP), which automatically determines the most appropriate pruning rate for each convolutional layer in a CNN.
- The HDBOFP can concurrently consider all convolutional layers, owing to the characteristic of the proposed objective function. In addition, it can automatically determine optimal pruning rate combinations without a time-consuming retraining phase.
- The use of a high-dimensional Bayesian optimization enables HDBOFP to theoretically provide a global optimal pruning rate combination and effectively solve the high dimensionality problem of the proposed objective function.
- To demonstrate the wide and general applicability of HDBOFP, we evaluate its performance through four image classification and object detection benchmarks. The results of extensive experiments suggest that HDBOFP performs better than conventional filter pruning algorithms.

The remainder of this paper is organized as follows. Section II briefly introduces filter pruning, Bayesian optimization, and notation. Section III discusses our proposed automated filter pruning algorithm, focusing on the objective function and high-dimensional Bayesian optimization. The experimental results are presented and analyzed in Section IV. Finally, the study is summarized in Section V.

II. PRELIMINARIES

A. FILTER PRUNING

Current pruning methods can be categorized into weight pruning (unstructured pruning) and filter pruning (structured pruning). Weight pruning aims to remove the fine-grained weight of filters, which leads to unstructured sparsity in

pruned CNNs. In contrast, filter pruning can achieve structured sparsity, enabling the pruned network to fully utilize highly efficient basic linear algebra subprogram libraries to achieve improved acceleration.

An active research topic in filter pruning is the quantification of filter importance, that is, defining the pruning criteria. Following [2], pruning criteria can be divided into two categories: weight-based criteria, which evaluate filter importance based on their weights, and activation-based criteria, which utilize training data and filter activations to quantify the importance of filters. Regarding weight-based criteria, [3] and [4] utilized the l1- and l2-norm of filters, respectively; [8] imposed sparsity on the scaling parameters of batch normalization layers to prune the CNN; and [5] proposed that the filter near the geometric median should be pruned. In terms of activation-based criteria, [7] introduced principal component analysis to identify the part of the network that should be pruned; [6] claimed to use information from the next layer to assist with filter pruning; and [9] explored the linear relationship between different activations to eliminate unimportant filters. Most of these studies utilized the same pruning rate for different layers and did not consider that different layers have various peculiarities and different filter distributions.

B. BAYESIAN OPTIMIZATION

Bayesian optimization is a class of machine learning-based optimization methods consisting of two major components: a Bayesian statistical model, which models the objective function, and an acquisition function, which determines the next sampling points. The Bayesian statistical model, which is a Gaussian process, quantifies the uncertainty of the objective function at unobserved data points. The acquisition function measures the predictive enhancement of unobserved data to determine the next sampling point [11], [12]. Therefore, Bayesian optimization is well suited for black-box derivative-free global optimization for the following reasons: (1) it does not require the structural information of the objective function (black box); (2) it does not observe the derivatives of the objective function (derivative-free); and (3) it determines the global optimum by calculating the uncertainty of the objective function at unobserved points (global optimization). Bayesian optimization is extremely versatile owing to its ability to optimize expensive black-box derivative-free functions. Recently, it has been applied to determine optimum hyper-parameters for machine learning algorithms, particularly deep neural networks [13]–[16].

C. NOTATION AND DEFINITION

In this study, the mathematical notation follows the same as used in [2]. For a neural network with L layers, the weight of the lth convolutional layer is denoted as $\mathcal{W}^{(l)} \in \mathbb{R}^{K\times K\times C_I^{(l)}\times C_O^{(l)}}$, where K denotes the kernel size, and $C_I^{(l)}$ and $C_O^{(l)}$ are the number of input and output channels, respectively. The ith filter of the lth convolutional layer is represented by $\mathbf{W}^{(l)} \in \mathbb{R}^{K\times K\times C_I^{(l)}}$. The input and output feature maps are correspondingly denoted by $\mathcal{I} \in \mathbb{R}^{C_I^{(l)}\times H_I^{(l)}\times W_I^{(l)}}$, and $\mathcal{O} \in \mathbb{R}^{C_O^{(l)}\times H_O^{(l)}\times W_O^{(l)}}$, where $W_\cdot^{(\cdot)}$ and $H_\cdot^{(\cdot)}$ are the width and height of the feature maps, respectively. $\mathbf{p} \in \mathbb{R}^L$ is the pruning rate combination, and $\mathbf{p}^{(l)} \in [0, 1-\epsilon]$ is the pruning rate of the lth convolutional layer, where ϵ is a very small number. $\check{\mathcal{W}}^{(l)} \in \mathbb{R}^{K\times K\times(1-\mathbf{p}^{(l-1)})C_I^{(l)}\times(1-\mathbf{p}^{(l)})C_O^{(l)}}$ represents the weight of the lth hard-pruned convolutional layer. The weight of the lth soft pruned convolutional layer is given by the sparse tensor $\bar{\mathcal{W}}^{(l)} \in \mathbb{R}^{K\times K\times C_I^{(l)}\times C_O^{(l)}}$, which indicates that the soft filter pruning zeroizes the filters that are selected to be pruned. $\mathcal{F} = \{\mathbf{W}_i^{(l)}, i \in [1, C_O^{(l)}], l \in [1, L]\}$ is the filter set comprising all the filters in the CNN. $\check{\mathcal{F}}$ and $\bar{\mathcal{F}}$ are the filter sets of the hard-pruned and soft-pruned CNNs, respectively.

Filter pruning minimizes the value of the loss function under sparsity constraints on the filters. Given a dataset $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$, where $\mathbf{x}_n$ is the nth input and $\mathbf{y}_n$ is the corresponding label, the constrained optimization problem can be formulated as follows:

$$\begin{aligned} &\min_{\check{\mathcal{F}}} \frac{1}{N}\sum_{n=1}^{N} \hat{\mathcal{L}}(\check{\mathcal{F}}; (\mathbf{x}_n, \mathbf{y}_n)) \\ &\text{subject to } \frac{\mathcal{C}_M(\check{\mathcal{F}})}{\mathcal{C}_M(\mathcal{F})} \le \tau_M \text{ and } \frac{\mathcal{C}_F(\check{\mathcal{F}})}{\mathcal{C}_F(\mathcal{F})} \le \tau_F, \end{aligned} \tag{1}$$

where $\hat{\mathcal{L}}(\cdot)$ is a standard loss function (e.g., cross-entropy loss, mean squared error), $\mathcal{C}_M(\cdot)$ and $\mathcal{C}_F(\cdot)$ are the storage and computational costs of the network, respectively, and τ_M and τ_F are the ratios between the storage and computation costs of the pruned CNN and those of the original CNN, respectively.

III. PROPOSED ALGORITHM

The aim of our proposed HDBOFP is to automatically determine an effective pruning rate for each of the layers in a convolutional network, rather than manually determining them, as is the case in previous filter pruning studies. The HDBOFP consists of two major components: (1) a novel objective function, which estimates the accuracy degradation given a pruning rate combination $\mathbf{p}$ without retraining the pruned CNN; and (2) a high-dimensional Bayesian optimization, which provides the optimal pruning rate combination $\mathbf{p}^*$ by minimizing the proposed high-dimensional non-convex non-differentiable loss function $\mathcal{L}$.

A. PROPOSED OBJECTIVE FUNCTION

Calculating the accuracy degradation given the pruning rate combination $\mathbf{p}$ typically requires retraining the pruned CNN, which is a time-consuming process that produces a non-identical output. To address this problem, we propose a novel objective function that can estimate the relative accuracy loss for a given pruning rate combination $\mathbf{p}$ without the need for a retraining phase. The proposed objective function is defined as follows:

$$\min_{\mathbf{p}} \mathcal{L}(\mathbf{p}; \mathcal{F}, \tau_M, \tau_F) = \min_{\mathbf{p}} \frac{1}{L}\sum_{l=1}^{L} \frac{\|\mathcal{W}^{(l)} - \bar{\mathcal{W}}^{(l)}\|_F^2}{\|\mathcal{W}^{(l)}\|_F^2}$$

$$\text{subject to } \frac{\mathcal{C}_M(\check{\mathcal{F}})}{\mathcal{C}_M(\mathcal{F})} \leq \tau_M \text{ and } \frac{\mathcal{C}_F(\check{\mathcal{F}})}{\mathcal{C}_F(\mathcal{F})} \leq \tau_F, \tag{2}$$

where $\|\cdot\|_F$ is the Frobenius norm, which is defined as the square root of the sum of the squares of the elements. The main idea behind this objective function is that the relative accuracy loss is proportional to the reconstruction loss between the original and the soft pruned filters. Therefore, our algorithm can approximate the expected accuracy loss given the pruning rate combination, while not being affected by the retraining process when searching for an optimal pruning rate combination $\mathbf{p}^*$. The proposed objective function has the form of a constrained high-dimensional combinatorial optimization, resulting in a non-differentiable non-convex optimization problem that can be addressed via Bayesian optimization.

B. HIGH-DIMENSIONAL BAYESIAN OPTIMIZATION

In the proposed algorithm, the Gaussian process provides a distribution that represents the potential values of the proposed objective function $\mathcal{L}(\mathbf{p})$ at an undiscovered pruning rate combination $\mathbf{p}$, based on the distribution obtained from k previously observed pruning rate combinations $\mathbf{p}_{1:k} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_k\}$. Following the Bayesian rule, we must define the prior distribution $\mathcal{L}(\mathbf{p}_{1:k})$ to obtain the conditional distribution $\mathcal{L}(\mathbf{p})|\mathcal{L}(\mathbf{p}_{1:k})$. Therefore, we assume that the prior distribution $\mathcal{L}(\mathbf{p}_{1:k})$ is a multivariate normal distribution as follows:

$$\mathcal{L}(\mathbf{p}_{1:k}) \sim \mathcal{N}(\mu_0(\mathbf{p}_{1:k}), \Sigma_0(\mathbf{p}_{1:k}, \mathbf{p}_{1:k})), \tag{3}$$

where $\mu_0(\mathbf{p}_{1:k}) \in \mathbb{R}^k$ is the mean vector, $\Sigma_0(\mathbf{p}_{1:k}, \mathbf{p}_{1:k}) \in \mathbb{R}^{k\times k}$ denotes the Mahalanobis covariance matrix, and $\mathcal{N}(\cdot,\cdot)$ is a multivariate normal distribution function. We can calculate the conditional distribution $\mathcal{L}(\mathbf{p})$ as:

$$\begin{aligned}
\mathcal{L}(\mathbf{p})|\mathcal{L}(\mathbf{p}_{1:k}) &\sim \mathcal{N}(\mu_k(\mathbf{p}), \sigma_k^2(\mathbf{p})),\\
\mu_k(\mathbf{p}) &= \Sigma_0(\mathbf{p}, \mathbf{p}_{1:k})\Sigma_0(\mathbf{p}, \mathbf{p}_{1:k})^{-1}\\
&\quad \times (\mathcal{L}(\mathbf{p}_{1:k}) - \mu_0(\mathbf{p}_{1:k})) + \mu_0(\mathbf{p}_{1:k}),\\
\sigma_k^2(\mathbf{p}) &= \Sigma_0(\mathbf{p}, \mathbf{p}) - \Sigma_0(\mathbf{p}, \mathbf{p}_{1:k})\Sigma_0\\
&\quad \times (\mathbf{p}_{1:k}, \mathbf{p}_{1:k})^{-1}\Sigma_0(\mathbf{p}_{1:k}, \mathbf{p}).
\end{aligned} \tag{4}$$

Therefore, we can define the potential values of the proposed objective function at a pruning rate combination $\mathbf{p}$ that has not been evaluated using a conditional distribution, which is typically associated with the distribution of k observed pruning rate combinations $\mathbf{p}_{1:k}$. After the Gaussian process, we utilize the expected improvement acquisition function to determine the next pruning rate combination. The expected improvement returns the best pruning rate combination, i.e., that which yields the lowest loss among the already observed pruning rate combinations. Because the proposed objective function computes the loss of the observed pruning rate combination without noise, it is clear that the best current pruning rate combination is $\mathbf{p}_{best} = \arg\min_{\mathbf{p}_{1:k}} \mathcal{L}(\mathbf{p}_{1:k})$.

The expected improvement acquisition function compares the loss of the current best pruning rate combination $\mathcal{L}(\mathbf{p}_{best})$ with the approximate objective error of the candidate pruning rate combinations $\mathcal{L}(\mathbf{p})$ to quantify the improvement yielded by the candidate pruning rate combination $\mathbf{p}$. Thus, the expected improvement can be defined as follows:

$$\mathcal{E}_k(\mathbf{p}) := \mathbb{E}_n(\max(\mathcal{L}(\mathbf{p}_{best}) - \mathcal{L}(\mathbf{p}), 0)), \tag{5}$$

where $\mathbb{E}_k(\cdot) = \mathbb{E}_k(\cdot|\mathbf{p}_{1:k}, \mathcal{L}(\mathbf{p}_{1:k}))$ denotes the expectation of the posterior distribution given evaluations of $\mathcal{L}$ at $\mathbf{p}_{1:k}$. The expected improvement acquisition function is primarily used because of its closed form under a Gaussian process. To determine the next pruning rate combination $\mathbf{p}_{k+1}$, which is expected to be closer to the global optimal $\mathbf{p}^*$ than the current best $\mathbf{p}_{best}$, the expected improvement acquisition function evaluates the candidate pruning rate combination $\mathbf{p}$ with the largest expected improvement:

$$\mathbf{p}_{k+1} = \arg\max \mathcal{E}_k(\mathbf{p}). \tag{6}$$

After the expected improvement, if the Bayesian optimization termination condition is satisfied, then the current pruning rate combination $\mathbf{p}_{k+1}$ is considered as the solution of the HDBOFP; otherwise, it is used to construct a posterior distribution through the Gaussian process.

Despite its remarkable effectiveness, Bayesian optimization is limited by a curse of dimensionality which consists in the Gaussian process producing a poor prediction for dimensions larger than 15–20. In HDBOFP, the dimension of the objective function corresponds to the number of convolutional layers L; therefore, the applicability of the proposed filter pruning algorithm is restricted to deep CNNs with no more than 15–20 layers. A common framework used to mitigate this problem is to consider a high-dimensional Bayesian optimization task as a standard Bayesian optimization in a low-dimensional embedding, where the embedding can be either linear [1], [28] or nonlinear [29], [30]. For HDBOFP, we utilized linear embedding for high-dimensional Bayesian optimization, as in [1], which exhibited superior performance compared with other algorithms in various applications.

When using linear embedding for high-dimensional Bayesian optimization, we assume the existence of a low-dimensional linear subspace that includes all the variations in $\mathcal{L} : \mathbb{R}^L \to \mathbb{R}$. Specifically, let $\mathcal{L}_e : \mathbb{R}^e \to \mathbb{R}$, $e \ll L$, and let $\mathbf{T} \in \mathbb{R}^{e\times L}$ be a projection from the L to the e dimensional space, then the linear embedding assumption is as follows: $\mathcal{L}(\mathbf{p}) = \mathcal{L}_e(\mathbf{T}\mathbf{p})$. Following [1], we generate a linear projection matrix $\mathbf{T}$ by sampling L points from the hypersphere $\mathbb{S}^{e-1}$ because this sampling method empirically has a better high-dimensional Bayesian optimization performance than randomly sampling L points. To prevent distortion in the linear embedding, HDBOFP constrains the optimization to points that do not project outside the bounds, by converting Equation 6 as follows:

$$\max_{\mathbf{p}_e \in \mathbb{R}^e} \mathcal{E}_k(\mathbf{p}_e) \text{ subject to } -1 \leq \mathbf{T}^\dagger \mathbf{p}_e \leq 1, \tag{7}$$

where $\mathbf{p}_e = \mathbf{T}\mathbf{p}$ and $\mathbf{T}^\dagger$ denotes the matrix pseudo-inverse. The constraints in the above equation are all linear; thus,

they form a polytope and can be addressed using off-the-shelf optimization tools. In addition, because the HDBOFP utilizes the Mahalanobis kernel in the Gaussian process, the projection is entirely linear and can be effectively modeled within this constrained space. Consequently, HDBOFP can be applied to deep CNNs using high-dimensional Bayesian optimization.

IV. EXPERIMENTAL RESULTS

To demonstrate the wide and general applicability of HDBOFP, we evaluate its performance using two image classification and one object detection benchmark datasets. Results from extensive experiments verify the effectiveness of the proposed filter pruning method.

A. EXPERIMENTAL SETTINGS

HDBOFP aims to automatically provide a pruning rate combination, given a CNN and filter pruning criteria. In the experiments herein discussed, the pruning criteria are set to the L2-norm [4], which is based on the ''less norm, less information'' assumption and has exhibited superior performance in two major computer vision applications, namely image classification and object detection. In the proposed algorithm, we set the desired FLOPs reducing ratio τ_F and the desired memory compression ratio τ_M according to the hardware efficiency of state-of-the-art pruning algorithms. In the tables in this section, HDBOFP-A and HDBOFP-B denote two pruned networks with different values of τ_F and τ_M with HDBOFP-B having lower threshold parameters than HDBOFP-A. The embedding dimension in high-dimensional Bayesian optimization e is initialized as half the number of convolutional layers L in the baseline model. We compare HDBOFP with previous CNN acceleration algorithms [3]–[5], [31]–[33], [33]–[42]. For a fair comparison of pruning from-scratch and pre-trained models, we use the same training epochs to train/fine-tune the network following [5], [31]. All the experiments were conducted with PyTorch [44] on NVIDIA RTX 3090 GPUs.

B. IMAGE CLASSIFICATION

The proposed automatic filter pruning algorithm based on high-dimensional Bayesian optimization is tested on two image classification benchmark datasets, specifically CIFAR-10 [21] and ImageNet (LSVRC-2012) [22]. The CIFAR-10 dataset contains 50k training images and 10k testing images, providing a total of 60k 32×32 colored images belonging to 10 different classes. The large-scale ImageNet dataset (LSVRC-2012) comprises 1.28M training images and 50k validation images, pertaining to 1k categories. In these image classification experiments, the effectiveness of the proposed algorithm is verified using ResNet [23] with different depths. ResNet consists of several residual modules which implement skip-connection; owing to this, previous studies claim that ResNet has less redundancy and is hence more difficult to compress than VGGNet [24]. Thus, we follow [10] to focus on filter pruning the challenging ResNet.

1) COMPARISON ON CIFAR-10

The proposed method is compared with state-of-the-art network pruning algorithms using ResNet on the CIFAR-10 dataset. Table 1 reports the accuracy degradation and reduction of FLOPs of the pruned models compared to the corresponding baseline network. As can be seen, our method can substantially reduce the computational cost for a given network without significant accuracy degradation. For example, the proposed HDBOFP-B can reduce FLOPs in ResNet-110 by 62.8% with an accuracy degradation of merely 0.52%. Compared with previous pruning algorithms, our method achieves pruned networks with less computational cost but higher test accuracy. Static methods, which utilize the same pruning rate on every convolutional layer, are clearly inferior to HDBOFP, e.g., the DSA [37] applied to ResNet-56 only decreases FLOPs by 52.2% and obtains a pruned network with 92.91% accuracy, while the proposed HDBOFP-A can deliver a network with an accuracy degradation of only 0.02% and a greater reduction of FLOPs of 64.5%. Our method also proves superior to existing dynamic pruning methods, which utilize a different pruning rate for each convolutional layer. For instance, ManiDP [31] implemented with ResNet-56 achieves 0.06% accuracy loss with 62.4% FLOPs pruned, which is less satisfactory than our method. We can infer that the proposed HDBOFP algorithm can adequately glean the redundancy of networks to get compact but powerful high-performing structures.

2) COMPARISON ON ImageNet (ILSVRC-2012)

For the ImageNet dataset, we test our HDBOFP on ResNet-34 and ResNet-56 with different pruning rates and without pruning the projection shortcuts for simplification, similar to [4]. As shown in Table 2, our HDBOFP achieves the best performance among the compared methods. For example, in the case of ResNet-34, FBS [32] results in a 51.2% speed-up ratio and a 1.65% top1 accuracy drop, while our HDBOFP-A achieves 56.1% speed-up ratio with just a 0.13% top1 accuracy degradation. Similarly, considering ResNet-56, compared to FPGM [5], which prunes 42.2% of FLOPs with a top5 accuracy loss, our HDBOFP-A has a lower 0.19% top-5 accuracy loss and a higher 53.4% FLOP reduction. These experimental results demonstrate that HDBOFP can produce a more compressed model with comparable or even better performance.

3) LAYER-WISE ANALYSIS

The remaining filter density of each layer of ResNet-34 on CIFAR-10 and ImageNet is illustrated in Figure 2. The peaks and crests indicate that the HDBOFP automatically prunes a 3×3 convolutional layer with a large pruning rate because it generally has significant redundancy; however, it prunes a more compact 1×1 convolutional network with lower sparsity. We can observe that the remaining filter distribution of HDBOFP is notably different from human experts' results shown in Table 3.8 of [17]. This suggests that the proposed

TABLE 1. Comparison of the performance of ResNet on CIFAR-10 after pruning with different methods. The "Acc. ↓" is the accuracy degradation between the pruned and baseline models; the smaller, the better. "FLOPs ↓" is the reduction of FLOPs from the baseline to the pruned model; the larger, the better.

Depth	Method	Baseline Acc. (%)	Pruned Acc. (%)	Acc. ↓ (%)	FLOPs	FLOPS ↓ (%)
34	MIL [33]	92.66	90.74	1.59	4.70E7	31.2
	SFP [4]	92.66	92.08	0.55	4.03E7	41.5
	FPGM [5]	92.66	91.93	0.73	3.23E7	53.2
	FBS [32]	92.66	91.98	0.68	3.06E7	55.7
	ManiDP [31]	92.66	92.15	0.51	2.54E7	63.2
	HDBOFP-A	92.66	**92.21**	**0.45**	2.35E7	65.9
	HDBOFP-B	92.66	91.74	0.92	**1.88E7**	**72.8**
56	SFP [4]	93.7	92.26	1.44	5.94E7	52.6
	FPGM [5]	93.7	93.49	0.21	5.94E7	52.6
	HRank [34]	93.7	93.17	0.53	6.27E7	50
	DSA [37]	93.7	92.91	0.79	5.99E7	52.2
	Hinge [36]	93.7	93.69	0.01	6.27E7	50
	DHP [35]	93.7	93.58	0.12	6.15E7	50.9
	FBS [32]	93.7	93.52	0.18	5.81E7	53.6
	ManiDP [31]	93.7	93.64	0.06	4.71E7	62.4
	HDBOFP-A	93.7	**93.68**	**0.02**	4.45E7	64.5
	HDBOFP-B	93.7	92.87	0.83	**3.80E7**	**69.7**
110	PFEC [3]	93.53	92.94	0.61	1.55E8	38.6
	SFP [4]	93.68	93.38	0.3	1.50E8	40.8
	FPGM [5]	93.68	93.85	-0.17	1.21E8	52.3
	HDBOFP-A	93.68	**94.07**	**-0.39**	1.05E8	58.7
	HDBOFP-B	93.68	93.16	0.52	**9.43E7**	**62.8**

TABLE 2. Comparison of the performance of ResNet on ImageNet after pruning with different methods. The "Acc. ↓" is the accuracy degradation between the pruned and baseline models; the smaller, the better. "FLOPs ↓" is the reduction of FLOPs from the baseline to the pruned model; the larger, the better.

Depth	Method	Baseline Top1 Acc. (%)	Pruned Top1 Acc. (%)	Baseline Top5 Acc. (%)	Pruned Top5 Acc. (%)	Top1 Acc. ↓ (%)	Top5 Acc. ↓ (%)	FLOPS ↓ (%)
34	SFP [4]	73.31	71.83	91.42	90.33	1.48	1.09	41.1
	FPGM [5]	73.31	72.54	91.42	91.13	0.77	0.29	41.1
	DMC [38]	73.31	72.57	91.42	91.11	0.74	0.31	43.4
	LCCN [33]	73.31	72.99	91.42	91.19	0.32	0.23	24.8
	FBS [32]	73.31	71.66	91.42	90.13	1.65	1.29	51.2
	ManiDP [31]	73.31	72.74	91.42	91.04	0.57	0.38	55.3
	HDBOFP-A	73.31	**73.18**	91.42	**91.38**	**0.13**	**0.04**	56.1
	HDBOFP-B	73.31	71.78	91.42	90.48	1.53	0.94	**65.8**
56	ThiNet [39]	72.88	72.04	91.14	90.67	0.84	0.47	41.8
	SFP [4]	76.15	62.14	92.87	84.6	14.01	8.27	41.8
	FPGM [5]	76.15	75.59	92.87	92.63	0.56	0.24	42.2
	HDBOFP-A	76.15	**75.63**	92.87	**92.68**	**0.52**	**0.19**	53.4
	HDBOFP-B	76.15	74.89	92.87	91.98	1.26	0.89	**61.7**

automated filter pruning can fully explore the optimization space and allocate sparsity in an improved way.

C. OBJECT DETECTION

We further validate the effectiveness of our automated filter pruning method using one object-detection benchmark dataset, MS-COCO 2017 [25]. This dataset contains 118k images in the training set, 5k in the validation set, and 20k in the test set, i.e., a total of 143k color images belonging to 80 object classes. To evaluate the object detection performance, we used the average precision (AP) over multiple intersection-over-union (IOU). AP@.5:.95 corresponds to the average AP for IOU from 0.5 to 0.95 with a step size of 0.05. For the MS-COCO 2017, AP is the average over 10 IOU levels on 80 categories. In these object detection experiments, we applied our automated filter pruning method on YOLOv5 [27], which is a state-of-the-art object detection model that has exhibited remarkable performance in various computer vision applications.

1) COMPARISON ON MS-COCO 2017

The experimental results of the pruning of YOLOv5s on MS-COCO 2017 are presented in Table 3, in which the AP on the validation set is reported. Compared with previous

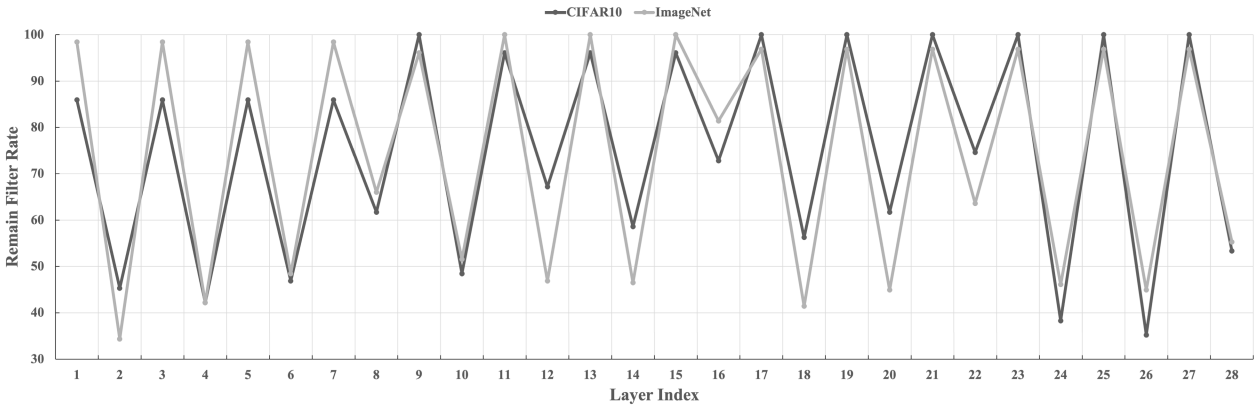


FIGURE 2. Visualization of the remaining filters in ResNet-34 as classification on CIFAR-10 and ImageNet progresses with pruning via HDBOFP.

TABLE 3. Detection performance of different models on the COCO 2017 val set. The models are compared in terms of number of parameters, FLOPs and average precision.

Model	AP@ 0.5:0.95	FLOPs	Params.
YOLOv5l [27]	48.8	1.15E11	4.81E7
YOLOv5m [27]	45.2	5.10E10	4.45E7
DTER-ResNet50 [40]	42	8.60E10	4.10E7
DTER-ResNet101 [40]	43.5	1.52E11	6.00E7
Deformable DETR [41]	43.8	1.73E11	4.38E7
Pruned-YOLOv5-I [42]	46.5	7.20E10	4.65E7
Pruned-YOLOv5-II [42]	44.6	5.40E10	4.46E7
HDBOFP-YOLOv5l	**47.62**	6.53E10	4.26E7
HDBOFP-YOLOv5m	43.28	**4.98E10**	**3.78E7**

light-weight object detectors, a significantly higher reduction of FLOPs with less degradation of performance is achieved by our method. Additionally, HDBOFP offers clear advantages in model volume, which reduce the overhead model storage. In other words, HDBOFP achieves excellent performance in terms of number of parameters, FLOPs, and average precision. For example, using our method, more than 43% FLOPs of the YOLOv5l model are reduced while the validation AP reaches 47.62AP. Compared to the Pruned-YOLO [42], our method shows notable superiority. Therefore, these experimental results verify that selecting a proper pruning rate combination via HDBOFP can make the conventional filter pruning algorithm more powerful, leading to high model efficiency with low performance degradation.

D. ABLATION STUDY

The experimental results of an ablation study analyzing the effect of the embedding method on filter pruning when applied to ResNet-18 on CIFAR-10, are summarized in Table 4.

TABLE 4. Comparison of the performance of ResNet-18 CIFAR-10 with different embedding methods. The "Acc. ↓" is the accuracy degradation between the pruned and baseline models; the smaller, the better. "FLOPs ↓" is the reduction of FLOPs from the baseline to the pruned model; the larger, the better.

Embedding Method	Pruned Acc. (%)	Acc. ↓ (%)	FLOPs	FLOPS ↓ (%)
BayesOpt (w.o. Embedding)	88.97	3.23	1.77E7	57.8
REMBO [28]	89.63	2.57	1.63E7	61.2
HeSBO [43]	90.48	1.72	1.82E7	56.8
ALEBO [1]	**91.94**	**0.26**	**1.56E7**	**62.8**

1) INFLUENCE OF EMBEDDING

To verify the effectiveness of high-dimensional Bayesian optimization, we tested HDBOFP without an embedding method, which is equivalent to implementing the proposed algorithm with standard Bayesian optimization. As shown in Table 4, the pruned ResNet-18 with standard Bayesian optimization has a poor filter pruning performance with an accuracy of only 88.97%. These experimental results imply that high-dimensional Bayesian optimization is inevitably required, the major reason being that the Gaussian process is known to produce poor predictions for dimensions larger than 15-20 [28], [43], [45].

2) VARYING EMBEDDING METHOD

Table 4 also shows high-dimensional Bayesian optimization performance on automated filter pruning. ALEBO [1] delivered the best performance among the tested embedding methods, which also included REMBO [28] and HeSBO [43]. REMBO specifies embedding via a random projection matrix with each element i.i.d. $N(0, 1)$. Bayesian optimization is performed in the embedding to identify a point to be evaluated, which is given an objective value. Without box bounds, REMBO comes with a strong guarantee: if embedding space is larger than ambient space, then the embedding contains an

optimum with a 100% probability [28]. Unfortunately, things become complicated when there are box bounds in the ambient space, therefore, as the proposed automated filter pruning algorithm is based on constrained Bayesian optimization, REMBO has a poorer performance than ALEBO. HesBO avoids the challenges of REMBO related to box bounds. However, the embedding method in HeSBO is not guaranteed to contain an optimum with high probability, and such probability may be low. Relative to REMBO, HeSBO improves the ability to model and optimize the embedding but reduces the chance of the embedding containing an optimum. From this experimental evidence, the effectiveness of ALEBO in the proposed algorithm can be considered proved.

V. CONCLUSION

In this study, we propose a novel automated filter pruning method for deep CNN compression, named HDBOFP. Unlike conventional methods, HDBOFP explicitly considers the difference between layers and adaptively selects a specific pruning rate for each of them. To efficiently and effectively produce a pruning rate combination, HDBOFP approximates an expected accuracy degradation given a pruning rate combination with no need to incur in a time-consuming retraining process. In addition, the proposed method utilizes high-dimensional Bayesian optimization to solve the non-convex non-differentiable minimization problem, which provides the optimal high-dimensional combinatorial distribution of pruning rates. After extensive experimentation with image classification and object detection, it was demonstrated that HDBOFP outperforms conventional filter pruning methods delivering a higher compression ratio, while simultaneously better preserving accuracy and reducing human labor demand. In future work, we may consider utilizing more efficient Bayesian optimization algorithms, such as parallel processing and sparse approximation.

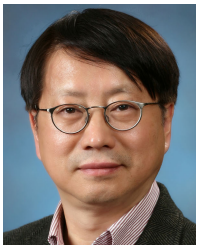
REFERENCES

[1] B. Letham, R. Calandra, A. Rai, and E. Bakshy, "Re-examining linear embeddings for high-dimensional Bayesian optimization," 2020, *arXiv:2001.11659*.
[2] 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.
[3] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. Peter Graf, "Pruning filters for efficient ConvNets," 2016, *arXiv:1608.08710*.
[4] 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.
[5] 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.
[6] J.-H. Luo, J. Wu, and W. Lin, "ThiNet: A filter level pruning method for deep neural network compression," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5058–5066.
[7] X. Suau, L. Zappella, and N. Apostoloff, "Filter distillation for network compression," 2018, *arXiv:1807.10585*.
[8] J. Ye, X. Lu, Z. Lin, and J. Z. Wang, "Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers," 2018, *arXiv:1802.00124*.
[9] D. Wang, L. Zhou, X. Zhang, X. Bai, and J. Zhou, "Exploring linear relationship in feature map subspace for ConvNets compression," 2018, *arXiv:1803.05729*.
[10] X. Dong and Y. Yang, "Network pruning via transformable architecture search," 2019, *arXiv:1905.09717*.
[11] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Global Optim.*, vol. 13, no. 4, pp. 455–492, Dec. 1998.
[12] J. Mockus, "The application of Bayesian methods for seeking the extremum," *Towards Global Optim.*, vol. 2, pp. 117–129, Dec. 1978.
[13] J. Snoek, "Practical Bayesian optimization of machine learning algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012.
[14] T. Kim, J. Lee, and Y. Choe, "Bayesian optimization-based global optimal rank selection for compression of convolutional neural networks," *IEEE Access*, vol. 8, pp. 17605–17618, 2020.
[15] D. Silver, A. Huang, C. J. Maddison, A. Guez, and L. Sifre, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, Jan. 2016.
[16] B. Kim, T. Kim, and Y. Choe, "Bayesian optimization based efficient layer sharing for incremental learning," *Appl. Sci.*, vol. 11, no. 5, p. 2171, Mar. 2021.
[17] S. Han, "Efficient methods and hardware for deep learning," Ph.D. dissertation, Stanford Univ., Stanford, CA, USA, 2017.
[18] M. Yin, Y. Sui, S. Liao, and B. Yuan, "Towards efficient tensor decomposition-based DNN model compression with optimization framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10674–10683.
[19] S. Young, Z. Wang, D. Taubman, and B. Girod, "Transform quantization for CNN compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, May 28, 2021, doi: 10.1109/TPAMI.2021.3084839.
[20] Y. Wang, X. Li, M. Shi, K. Xian, and Z. Cao, "Knowledge distillation for fast and accurate monocular depth estimation on mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 2457–2465.
[21] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.
[22] O. Russakovsky, J. Deng, H. Su, and J. Krause, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[23] 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.
[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[25] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.
[26] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2014.
[27] G. Jocher, "Ultralytics/YOLOV5: V3. 1-bug fixes and performance improvements," Tech. Rep., 2020. [Online]. Available: https://github.com/ultralytics/yolov5
[28] Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. De Feitas, "Bayesian optimization in a billion dimensions via random embeddings," *J. Artif. Intell. Res.*, vol. 55, pp. 361–387, Feb. 2016.
[29] X. Lu, "Structured variationally auto-encoded optimization," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3267–3275.
[30] R. Moriconi, M. P. Deisenroth, and K. S. Sesh Kumar, "High-dimensional Bayesian optimization using low-dimensional feature spaces," *Mach. Learn.*, vol. 109, nos. 9–10, pp. 1925–1943, Sep. 2020.
[31] Y. Tang, Y. Wang, Y. Xu, Y. Deng, C. Xu, D. Tao, and C. Xu, "Manifold regularized dynamic network pruning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5018–5028.
[32] X. Gao, Y. Zhao, L. Dudziak, R. Mullins, and C.-Z. Xu, "Dynamic channel pruning: Feature boosting and suppression," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–14.
[33] X. Dong, J. Huang, Y. Yang, and S. Yan, "More is less: A more complicated network with less inference complexity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5840–5848.
[34] M. Lin, R. Ji, Y. Wang, Y. Zhang, B. Zhang, Y. Tian, and L. Shao, "HRank: Filter pruning using high-rank feature map," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1529–1538.

[35] Y. Li, S. Gu, K. Zhang, L. V. Gool, and R. Timofte, "DHP: Differentiable meta pruning via hypernetworks," in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K.: Springer, Aug. 2020, pp. 608–624.
[36] Y. Li, S. Gu, C. Mayer, L. V. Gool, and R. Timofte, "Group sparsity: The Hinge between filter pruning and decomposition for network compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 8018–8027.
[37] X. Ning, T. Zhao, W. Li, P. Lei, Y. Wang, and H. Yang, "DSA: More efficient budgeted pruning via differentiable sparsity allocation," in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K.: Springer, Aug. 2020, pp. 592–607.
[38] S. Gao, F. Huang, J. Pei, and H. Huang, "Discrete model compression with resource constraint for deep neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1899–1908.
[39] J.-H. Luo, H. Zhang, H.-Y. Zhou, C.-W. Xie, J. Wu, and W. Lin, "ThiNet: Pruning CNN filters for a thinner Net," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2525–2538, Oct. 2019.
[40] N. Carion, F. Massa, G. Synnaeve, and N. Usunier, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.
[41] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," 2020, *arXiv:2010.04159*.
[42] J. Zhang, P. Wang, Z. Zhao, and F. Su, "Pruned-YOLO: Learning efficient object detector using model pruning," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2021, pp. 34–45.
[43] A. Nayebi, A. Munteanu, and M. Poloczek, "A framework for Bayesian optimization in embedded subspaces," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4752–4761.
[44] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. Autodiff Workshop Adv. Neural Inf. Process. Syst.*, 2017.
[45] C. Li, K. Kandasamy, B. Póczos, and J. G. Schneider, "High dimensional Bayesian optimization via restricted projection pursuit models," in *Proc. 19th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2016, pp. 884–892.

TAEHYEON KIM (Graduate Student Member, IEEE) received the B.S. degree in electronics from Kangwon National University, Chuncheon, South Korea, in 2017. He is currently pursuing the Ph.D. degree in electrical and electronic engineering with Yonsei University, Seoul, South Korea. His research interests include computer vision, Bayesian optimization, tensor analysis, and automated machine learning.

HEUNGJUN CHOI is currently pursuing the B.S. degree in electrical and electronic engineering with Yonsei University, Seoul, South Korea. His research interests include signals and systems, wireless communication, computer networking systems, and neural network compression.



YOONSIK CHOE (Senior Member, IEEE) received the B.S. degree in electrical engineering from Yonsei University, Seoul, South Korea, in 1979, the M.S.E.E. degree in systems engineering from Case Western Reserve University, Cleveland, OH, USA, in 1984, the M.S. degree in electrical engineering from Pennsylvania State University, State College, PA, USA, in 1987, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1990. From 1990 to 1993, he was a Principal Research Staff with the Industrial Electronics Research Center, Hyundai Electronics Company Ltd. Since 1993, he has been with the Department of Electrical and Electronic Engineering, Yonsei University, where he is currently a Professor. His research interests include video coding, video communication, statistical signal processing, and digital image processing.