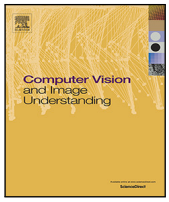
Computer Vision and Image Understanding 240 (2024) 103942

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu



CPRNC: Channels pruning via reverse neuron crowding for model compression

Pingfan Wu [a,b,c], Hengyi Huang [a,b,c], Han Sun [a], Dong Liang [a], Ningzhong Liu [a,b,c,*]

[a] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
[b] MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, Nanjing 211106, China
[c] Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 211106, China

ARTICLE INFO

Communicated by Xuelong Li

MSC:
41A05
41A10
65D05
65D17

Keywords:
Neuron crowding
Convolutional neural networks
Channel pruning
Model compression

ABSTRACT

Channel pruning is an efficient technique for model compression, removing redundant parts of a convolutional neural network with minor degradation in classification accuracy. Previous criteria of channel pruning ignore neurons' intrinsic relationship and the high correlation with input samples. Inspired by the visual crowding phenomenon in neuroscience, this paper presents a novel channel pruning method via reverse neuron crowding, dubbed CPRNC, to address this issue. First, CPRNC involves a neuron crowding degree measure (NCDM) module, which builds the relationship model among all artificial neurons by observing their crowding behaviors. Subsequently, each channel's importance is evaluated by the crowding degree of corresponding channels. Considering that the channel importance is affected by the characteristic of input samples, CPRNC designs a neuron crowding degree recalibrate (NCDR) module. NCDR emphasizes discriminative samples to recalibrate the channel priority list generated by NCDM, further enhancing the precision of the pruning criterion. Experimental results show that CPRNC achieves performance that competes with state-of-the-art pruning methods, including dynamic channel pruning and learning-based pruning. For example, we prune ResNet-50 with 56.7% FLOPs on the large-scale dataset ImageNet1K with only a 0.19% decrease in accuracy. At low pruning rates, CPRNC achieves lossless compression, *e.g.*, the pruned ResNet-56 on CIFAR-10 increases accuracy by 0.13% over the baseline model at 56.3% FLOPs reduction.

1. Introduction

Convolutional neural networks (CNNs) are widely deployed in a large variety of vision-related tasks, *e.g.*, image classification (He et al., 2016), object detection (Ren et al., 2015) and human–computer interaction (Khan et al., 2023). Since CNNs are constrained by run-time latency and model size while requiring to preserve prominent performance, many researchers focus on developing model compression methods such as quantization (Han et al., 2016), distillation (Hinton et al., 2015), pruning (Han et al., 2015). Among them, model pruning (Alqahtani et al., 2021; Bonnaerens et al., 2022; Mondal et al., 2022; Tian et al., 2023) is the most straightforward way and is widely adopted in academia and industry.

Model pruning methods are based on the over-parameterization (LeCun et al., 1989; Hassibi and Stork, 1992) assumption, which can be divided into **unstructured** pruning and **structured** pruning according to whether additional hardware and libraries design is needed to achieve practical acceleration. Unstructured pruning (Han et al., 2016) aims to remove the weight values of neurons, which produces sparse weight matrices leading to unstructured sparsity in the network. Sparse weight matrices cannot lead to speedup without dedicated hardware or libraries (Shen et al., 2022). Structured pruning (Mondal et al., 2022) removes entire channels or layers to obtain a compact sub-network without specific hardware or libraries to reduce computation on GPU/CPU devices. This paper aims to develop a novel channel pruning method to obtain a compact sub-network.

Channel pruning belongs to structured pruning, whose core problem is estimating channel importance to obtain a superior sub-network. Previous channel pruning methods (Liu et al., 2017; Li et al., 2017; Ding et al., 2021) utilize the norms of filters to evaluate their importance based on the hypothesis that small norms of filters are less important. However, these pruning methods lack discrimination between neurons as it forces all neurons in a single channel to obey the synchronous responses. To address this challenge, this paper proposes a novel and precise metric for the pruning criterion. Furthermore, dynamic pruning methods (Gao et al., 2018, 2021; Li et al., 2021) preserve the original network structure to dynamically route sub-networks during inference. For these data-driven methods, the ranking of channel importance is

* Corresponding author at: College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China.
E-mail addresses: wknh006163@163.com (P. Wu), hhyhhy@nuaa.edu.cn (H. Huang), sunhan@nuaa.edu.cn (H. Sun), liangdong@nuaa.edu.cn (D. Liang), lnz_nuaa@163.com (N. Liu).

https://doi.org/10.1016/j.cviu.2024.103942
Received 13 May 2023; Received in revised form 11 November 2023; Accepted 21 January 2024
Available online 23 January 2024

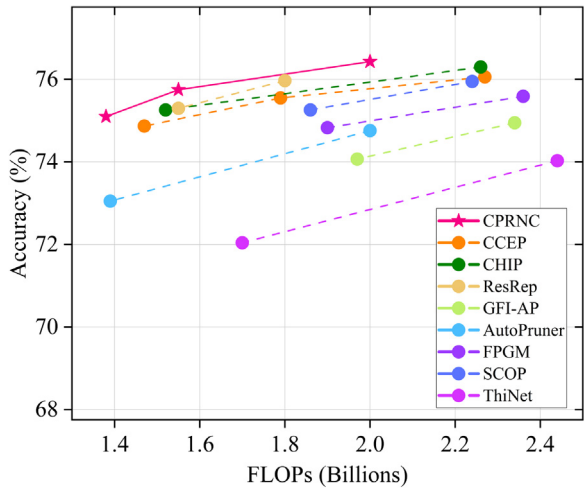


Fig. 1. Comparison of accuracy-FOLPs Pareto curves of compressed ResNet-50 models on ImageNet, CPRNC provides the highest Top-1 accuracy with the least FLOPs compared to competing pruning methods.

sensitive to the input data triggering unstable results (Tang et al., 2020) as the importance of filters is highly input-dependent (Tang et al., 2021). Inspired by this, this paper uses the feature maps generated by the filter to estimate importance, which contains a rich amount of input-related information (Lin et al., 2020). Furthermore, CPRNC identifies discriminative samples to improve the precision of the new metric. As shown in Fig. 1, our CPRNC achieves a fast sub-network search with few training costs while achieving superior performance compared to competitive pruning methods.

This paper presents a channel pruning method relying on a novel metric inspired by crowding phenomenon in neuroscience, as shown in Fig. 2. Flom et al. (1963) indicates that the human brain neurons perceive targets with interference from similar features surrounding the target. Based on this neuroscience discovery, we design a neuron crowding degree measure (NCDM) module to model the relationship of neurons. We point out that the representation capability of a neuron can be inhibited by neighboring similar neurons, which is dubbed as neuron crowding. The more neuron crowding exists within a channel, the weaker the representation capability of that channel, which will be assigned a lower importance score. The more informative neurons with reverse neuron crowding in the preserved channels, the stronger the model representation capability. NCDM calibrates the relationship among all neuron crowding behaviors to excavate informative neurons for guiding channel pruning criterion. Gao et al. (2018) indicates channel importance is highly input-dependent, so we design a neuron crowding degree recalibrate (NCDR) module to find discriminative samples with more reverse neuron crowding. NCDR recalibrates the neuron crowding behavior under different input samples and further improves the precision of the pruning criterion. Samples with low model confidence that cause more informative neurons should be assigned a higher value in the channel priority list, and the model confidence for each sample is defined as the top-2 difference of the output logits as a criterion to distinguish the neuron crowding activity of the sample. Overall, our contributions are three-fold as follows:

- We introduce neuron crowding as a metric that measures the correlation of neurons in channels for guiding the channel pruning criterion. To the best of our knowledge, this is the first paper to model reverse neuron crowding to design a pruning criterion. In contrast to the previous channel pruning criteria, reverse neuron crowding estimates the channel importance from a more fine-grained and precise perspective.
- We propose two lightweight modules, neuron crowding degree measure (NCDM) and neuron crowding degree recalibrate (NCDR). NCDM excavates more informative neurons and important channels for pruning, which boosts the accuracy of the pruned model. NCDR identifies discriminative samples to recalibrate channel importance scores, further improving the pruning criterion's precision. For example, NCDM and NCDR improve the accuracy of pruned ResNet-56 by 0.61% with 56.3% FLOPs reduction on CIFAR-10 dataset and pruned ResNet-34 by 1.53% with 49.5% FLOPs on ImageNet dataset.
- Extensive experiments have shown that our CPRNC outperforms most channel pruning methods. At low pruning rates, CPRNC can achieve lossless compression. For CIFAR-10, CPRNC reduces FLOPs by 56.3% and 52.6%, and meanwhile brings 0.13% and 0.66% accuracy increase over baseline ResNet-56 and ResNet-110, respectively. On large-scale datasets, CPRNC also performs excellently. For ImageNet, our compressed ResNet-50 model yields 2× FLOPs reduction with 76.43% accuracy outperforming the competing methods in the paper.

2. Related work

Model compression techniques include network pruning, knowledge distillation, quantization, etc. In comparison, sub-networks structures are more flexible from pruning and adaptable to various application scenarios. Additionally, it can be combined with other compression methods (*e.g.*, distillation) to compress the model further. Pruning is a more versatile approach, offering advantages in model storage, memory, and computational efficiency. Existing research on channel pruning can be broadly categorized into three forms: static, dynamic, and learning-based pruning.

2.1. Static channel pruning

Static channel pruning methods use a manually elaborate criterion as a unit importance metric to remove redundant channels based on the importance of channels defined artificially. Several studies (Liu et al., 2017; He et al., 2019) focus on designing different channel pruning criteria to identify channel importance. ResRep (Ding et al., 2021) successfully applies structural re-parameterization to channel pruning. These criteria lack discrimination between neurons because they push the entire channel to converge in the same direction, limiting the performance of the sub-network. Furthermore, recent works explore novel pruning methods that lie between weight pruning and channel pruning, such as N:M (Zhou et al., 2021) and 1 × N Pattern (Lin et al., 2023), resulting in improved performance of the pruned models. Recently, some pruning methods (Lin et al., 2020; Tang et al., 2020; Sui et al., 2021) measure feature map importance to provide better guidance for pruning because feature maps capture rich information from input samples and filters. These methods focus on the feature norm as a pruning criterion different from the filter norm, and we also measure channel importance by feature maps. In contrast, our method enhances the discrimination between neurons, so our pruning criterion contains more fine-grained and precise information.

2.2. Dynamic Channel pruning

Dynamic Channel pruning methods (Gao et al., 2018; Tang et al., 2021; Li et al., 2021) propose that each channel responds variously to different input instances owing to a plausible hypothesis that the importance of filters is highly input-dependent. Dynamic pruning methods use a variable subset of convolutional filters to achieve inference acceleration instead of a compact neural network after pruning. FBS (Gao et al., 2018) proposes predictively amplifying salient convolutional channels and skipping unimportant input channels at run-time. ManiDP (Tang et al., 2021) investigates the recognition complexity and

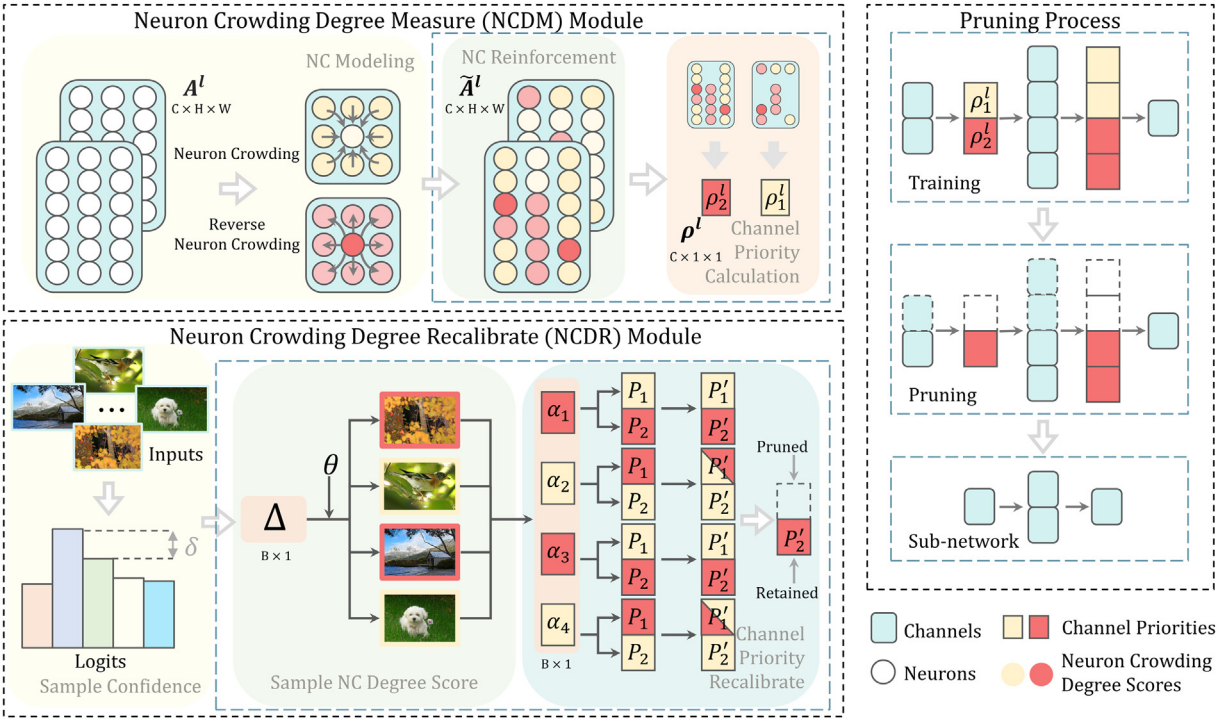


Fig. 2. Illustration of channel pruning via reverse neuron crowding (CPRNC). Neuron crowding reveals that neighboring similar neurons interfere with the perception of the target neuron. Neuron crowding degree measure (NCDM) module captures active reverse neuron crowding behavior in channels after modeling neuron crowding (NC). Then NCDM preserves high priority channels with more informative neurons in the model after reinforcing NC to mitigate performance degradation. Neuron crowding degree recalibrate (NCDR) module implements sample-oriented neuron crowding recalibration, further improving the pruning criterion's precision.

feature similarity between input images and determines the redundant filter variant for each input instance by aligning the manifold relationship between the instances and the pruned sub-networks. DSNet (Li et al., 2021) proposes a two-stage training scheme that achieves good hardware efficiency via dynamically adjusting filter numbers of networks at test time for different inputs. Despite the efficient inference efficiency, most dynamic pruning methods have exorbitant training costs and are difficult to deploy on resource-constrained edge devices.

2.3. Learning-based pruning

The learning-based approach (Liu et al., 2019; Shen et al., 2022; Shang et al., 2022) automatically searches for the optimal sub-network from the original CNN based on the manually formulated constraints. MetaPruning (Liu et al., 2019) trains a hyper-network and adopts evolutionary search to obtain an optimal candidate network. HALP (Shen et al., 2022) leverages latency lookup table and global saliency score to guide pruning. CCEP (Shang et al., 2022) proposes a cooperative coevolution method to reduce pruning space through a divide-and-conquer strategy. A popular scheme is incorporating Neural Architecture Search (NAS) (Cai et al., 2020) and channel pruning to learning and searching for network structures. Despite the remarkable performance, learning-based methods have a huge computational cost overall. In contrast, CPRNC is more efficient and easy to train than learning-based methods.

3. Approach

3.1. Preliminaries

Denote the dataset with N samples as $\mathcal{X} = \{x_i\}_{i=1}^{N}$, and $\mathcal{Y} = \{y_i\}_{i=1}^{N}$ are the ground-truth labels. $W^l \in \mathbb{R}^{c^l \times c^{l-1} \times k^l \times k^l}$ denotes weight parameters of the convolution filters in the lth layer. For a CNN model with L layers, the lth convolutional layer $W^l = \{F_1^l, F_2^l, \ldots, F_{c^l}^l\}$ contains c^l filters $F_i^l \in \mathbb{R}^{c^{l-1} \times k^l \times k^l}$ where c^l, c^{l-1} and k^l denote the number of output channels, the number of input channels and the kernel size, respectively.

Channel pruning discovers and estimates the importance of channels in training to prune redundant channels of the network while recovering an approximate original accuracy by fine-tuning. In general, network pruning can be formulated as the following optimization problem:

$$\min_{\{W^l\}_{l=1}^{L}} \sum_{i=1}^{N} \mathcal{L}(y_i, f(x_i, W^l)), s.t. \|W^l\|_0 \le \kappa^l \tag{1}$$

where $\mathcal{L}(\cdot,\cdot)$ is the loss function and $f(\cdot,\cdot)$ is the output function of CNN model $\{W^l\}_{l=1}^{L}$. Besides, $\|\cdot\|_0$ is the l_0-norm that measures the number of non-zero channels in the set, and κ^l is the number of channels to be preserved in the lth layer.

Considering the input instances, the feature maps generated by channels contain rich correlations distinct from the channels themselves which are not explicitly related to the input data (Lin et al., 2020). Unlike investigating the importance of channels in Eq. (1), feature-guided pruning methods (Lin et al., 2020; Tang et al., 2020; Sui et al., 2021) mainly identify the importance of feature maps produced by channels, which can be formulated as follows:

$$\min_{\{A^l\}_{l=1}^{L}} \sum_{i=1}^{N} \mathcal{L}(y_i, A^l), s.t. \|A^l\|_0 \le \kappa^l \tag{2}$$

where $A^l = \{A_1^l, A_2^l, \ldots, A_{c^l}^l\} \in \mathbb{R}^{b \times c^l \times h \times w}$ denotes a set of feature maps in the lth layer, with mini-batch size b, channels c, rows h, columns w. $A_i^l \in \mathbb{R}^{h \times w}$ denotes the feature maps corresponding to the ith channel. The feature selection aims to eliminate the redundant feature maps generated by channels in the original network and retain the κ^l feature maps by an elaborate pruning criterion. As shown in Eq. (2), feature-guided pruning extracts and estimates the feature maps generated by the channels for designing pruning criteria, which relate to the input data. In the previous feature-guided pruning methods, pruning criteria ignore the interdependence between features because

deriving the features that are genuinely relevant to the input samples through Eq. (2) alone is complex. Conversely, we design schemes to build the relationship model among neurons. We regard feature maps as the responses of neurons within a filter and calculate relationships between features during training to design pruning criteria related to input data. Therefore, our approach is feature-guided pruning.

3.2. Neuron crowding modeling

In visual neuroscience, the visual crowding effect is interpreted as neurons' particular mechanism that neighboring elements compromise a target's perception (Whitney and Levi, 2011). Since consciousness may not necessarily be continuous in the human brain when perceiving objectives, similar features surrounding the target lead to enhanced crowding effects.

Inspired by visual crowding, we introduce visual crowding into the relational modeling of artificial neurons in neural networks, which can be interpreted as neuron crowding, where the perception of a target neuron is compromised by neighboring neurons in the neural network. Nearby similar neurons amplify the target neuron's crowding effect, so a neuron with neuron crowding is a replaceable neuron because the information it perceives can be imitated by neighboring similar neurons. Conversely, a neuron with reverse neuron crowding is an informative neuron because the information it perceives is unique. For pruning, the more informative neurons are retained, the higher the priority of the corresponding channels and the stronger the representational capability of the pruned model.

A practicable implementation of finding neuron crowding is to appraise the linear separability of the target neuron and the others. We measure similarity with a score $s(t) = w_t^T t + b_t$, *i.e.*, a higher score indicates a higher similarity between t and t_i. Therefore, we define for each neuron a cost function of the following form for neuron crowding:

$$E(w_t, b_t) = L(1, s(t)) + \frac{1}{M}\sum_{i=1}^{M} L(-1, s(t_i)) \tag{3}$$

where the first term measures the loss L on the target neuron t and the second term measures the loss L on the other neurons t_i in a single channel of the input feature $A^l \in \mathbb{R}^{b\times c^l\times w^l\times h^l}$. $L(y, s(t)) = (y - s(t))^2$ is a square loss, so the minimum value obtained when the two terms in $L(y, s(t))$ are equal. The label of the target neuron t is $y = 1$, distinguished from the label of others $y = -1$. M means the size of the receptive field near the target neuron. In Bouma's law (Bouma, 1970), crowding is presented within a restricted window around the target, in which the window size is rough to 6 mm on the primary visual cortex (Tripathy and Levi, 1994). Precisely measuring neurons' receptive field during training is complex. Setting it to the entire channel is a reasonable and intuitive assumption to reduce computational costs. If the receptive field is smaller than the whole channel, it would be challenging to compute an approximate solution for the cost function of neuron crowding. To reduce computing cost, we set the receptive field range $M = H \times W - 1$ to all neurons in the entire channel except the target neuron rather than experimentally measuring the exact receptive field size.

3.3. Neuron crowding degree measure module

This section introduces the NCDM module we designed to capture neuron crowding behaviors and then introduces the channel priority calculation to map informative neurons to the channel dimension for guiding pruning. Given that feature maps contain rich input-related information, we treat feature maps generated by filters as mappings of the filter kernel space and apply NCDM to feature maps rather than the kernel space. During training, we capture neuron crowding behaviors across different samples to compute more precise channel importance scores. NCDM maps neuron crowding degree within a channel to a priority score, providing a new metric for the pruning criterion, as shown in Fig. 3.

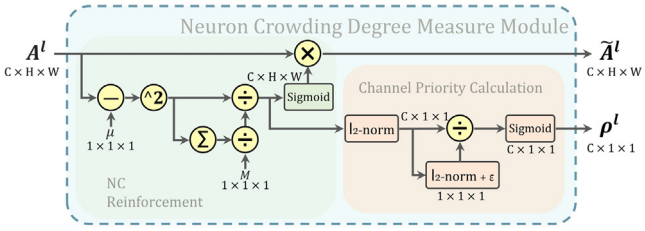


Fig. 3. Illustration of neuron crowding degree measure module (NCDM). After modeling neuron crowding, NCDM generates a candidate channel list through NC (Neuron Crowding) reinforcement and channel prioritization calculation. These operations are only applied in the training phase.

Neuron crowding reinforcement. The stronger the neuron crowding effect, the more similar neurons surround the target. After capturing the neuron crowding of each neuron, we reinforce the neuron crowding to find informative neurons. At this point, the neurons with reverse neuron crowding are considered informative neurons to guide the pruning criterion. Crowding reinforcement is used to amplify the variability among neurons and retain neurons with reverse neuron crowding, significantly affecting the model representation capabilities.

Due to the high computational cost of optimizing Eq. (3) for each neuron to capture neuron crowding, the aim here is to select a subset of discriminative and informative neurons automatically. Note that the optimal value of Eq. (3) characterizes the separability between the target neuron t and other neurons t_i, and thus can be used to measure the degree of discriminability of t. Optimizing Eq. (3) for thousands of neurons in each channel would be prohibitive, so we adopt a similar practice to Aubry et al. (2014) where the squared loss L can be used to obtain the optimal solution E^* of Eq. (3) in closed form:

$$E^* = \frac{4}{2 + (t-\mu)^T \phi (t-\mu)} \tag{4}$$

where $\mu = \frac{1}{M}\sum_{i=1}^{M} t_i$ and $\phi = \frac{1}{M}\sum_{i=1}^{M}(t_i - \mu)(t_i - \mu)^T$ denote the mean and covariance of the all neurons except t in single channel. Under a reasonable assumption that all neurons within a single channel follow an approximate distribution, then the mean and covariance can be reused by all neurons in the corresponding channel.

For pruning in practice, we evaluate $\Phi(t)$ of each candidate neuron t, which is inversely proportional to the solution E^*. The formula is as follows:

$$\Phi(t) = (t-\mu)^T \phi (t-\mu) \tag{5}$$

If the $\Phi(t)$ for target neuron t is high, the neuron is more distinguishable than other neurons in the receptive field range M. If the $\Phi(t)$ is small, the target t is not discriminative. A sigmoid function is subsequently applied to scale all neuron responses generated via the input samples as follows:

$$\tilde{A}^l = sigmoid(\Phi^l) \odot A^l \tag{6}$$

where Φ^l denotes the reverse neuron crowding scores corresponding to all neurons of all channels in the lth layer, and sigmoid is used to constrain the value Φ that is too large. The captured neuron responses are fed back to the corresponding neurons as Eq. (6) to automatically obtain reinforcement or inhibition, making the neurons with reverse neuron crowding distinctive from others. Unlike L1-norm or L2-norm sparsity, neuron crowding reinforcement amplifies inter-neuron variability more smoothly. Except for calculating the mean and covariance, all operations in the implementation are matrix element-wise operations to ensure parallel acceleration on GPU.

Channel priority calculation. A dimensional mapping method is needed for structured channel pruning to map the various neuron responses automatically captured by each channel to vector as the corresponding channel priority. The list of channel priorities is then sorted according

to the calculated priorities, and pruning operations are performed for channels with low importance scores. The Φ^l that measures the degree of crowding of all neurons in a layer is used to calculate the channel priorities. For aggregating entire channel's contextual information to calculate the channel priority, we apply l_2-norms to operate across all neurons in each channel as follows:

$$\bar{\Phi}^l = \{\sum_{i=1}^{H}\sum_{j=1}^{W}(\Phi^l_{:,:,i,j})^2 + \epsilon\}^{\frac{1}{2}} \tag{7}$$

where ϵ is a small constant used to avoid problems caused by extreme cases such as zero values. Then, the corresponding priority for each channel can be derived by the normalized $\tilde{\Phi}^l$, defined as follows:

$$\tilde{\Phi}^l = \frac{\bar{\Phi}^l}{\|\bar{\Phi}^l\|^2} = \frac{\bar{\Phi}^l}{\{(\sum_{c=1}^{C}(\bar{\Phi}^l_{:,c,:,:})^2) + \epsilon\}^{\frac{1}{2}}} \tag{8}$$

After adopting a sigmoid, $\rho^l = sigmoid(\tilde{\Phi}^l)$ is used to measure the channel importance for subsequent pruning after training. The computational complexity of NCDM is $(O(C))$. The proposed NCDM module is applied after the second convolution layer in each block for networks with residual blocks.

3.4. Neuron crowding degree recalibrate module

This section proposes a neuron crowding degree recalibrate module to nonlinearly process the channel priority list generated by each input sample. Discriminative samples are distinguished by calculating the sample confidence to recalibrate the channel priority list generated by NCDM. In contrast to the common global averaging, we emphasize the results of discriminative samples that capture more informative neurons with reverse neuron crowding. Discriminative samples further improve the precision of the pruning criterion, and the validity is demonstrated experimentally in Section 4.3.

Inspired by knowledge distillation (Zhao et al., 2022), the semantic information of logits related to the input samples is explicit and abstract, so we use the model logits of each sample as a criterion for measuring discriminative samples. When a model is confident in its classification of an input image, its output logits tend to be relatively large. Conversely, when the classification confidence is low, the output logits are usually close across different classes. This phenomenon suggests that the model can learn richer semantic information from different classes when it has low classification confidence. In addition, a well-trained model may gradually approach the overfitting state in the later stage of training, where it usually has high classification confidence and may be less likely to learn new knowledge. Pretrained models are more prone to this state during pruning training, where the channel importance obtained from feature map crowding is of low value. Some dynamic pruning methods (Gao et al., 2018; Li et al., 2021) propose dynamic routing sub-networks in inference based on the assumption that channel importance is highly correlated with input. However, as these methods preserve the original model structure, they are still difficult to deploy on resource-limited devices, limiting their practical applications in industry. To address the issue of highly correlated channel priority lists, we perform different treatments by assigning more significant scores to discriminative samples.

We calculate the difference $\delta \in \mathbb{R}^{b\times 1}$ between the top-2 logits of outputs as model confidence of each sample. Instead of directly comparing the top logits of model outputs across different samples, a more efficient method of obtaining a sample's logit distribution is to compare the first two logits. Although more intricate algorithms can better explore the variation in logit distribution among inputs, we aim to recalibrate the channel priority list generated by NCDM to achieve a more precise pruning criterion. Hence, NCDR employs δ as a simple metric to differentiate discriminative samples. Experimentally, the low confidence samples make the logits of the model contain rich semantic information, and the model learns richer knowledge from this sample when δ is less than a manually set threshold θ. The channel priority list obtained at this point should be more significant than that under the high confidence samples. The key to distinguishing discriminative samples is the setting of θ. Therefore we use the average confidence of all samples in a batch as the threshold $\theta = \frac{1}{B}\sum_{i=1}^{B}\delta_i$. Manually tuning θ can further improve the effect of NCDR. For discriminative samples, we apply a multiplication factor of $\alpha \in (0.5, 1]$, while we apply a factor of $(1-\alpha)$ for other samples to reduce the significance of this component. More experiments exploring the impact of different values of θ and α are shown in the supplementary material. Furthermore, the model may classify discriminative samples incorrectly but can still learn informative knowledge. Conversely, the model may be overly confident in a particular category in the event of misclassification, but this issue can be effectively mitigated by NCDR. The ultimate aim of NCDR is to recalibrate the channel priority list $P = \{\rho^1, \rho^2, \ldots, \rho^l\}$ of the sub-network obtained under high confidence misclassification. Therefore, the formula for NCDR is as follows:

$$P' = \begin{cases} \alpha * P, & \delta < \theta \\ (1-\alpha) * P, & otherwise \end{cases} \tag{9}$$

Finally, averaging P' under all batches to obtain the final channel priority list to prune the model. For a given compression rate, filter F_i^l with smaller $\{P'\}_i^l$ will be pruned to get a compact sub-network. The preserved filters retain the ability of the original network due to the retention of more informative neurons by NCDM and NCDR. Eventually, the network is fine-tuned to recover performance after pruning. The procedure of the overall CPRNC algorithm is shown in Algorithm 1.

Algorithm 1 Procedure description of CPRNC

Input: Pre-trained model, preserved filters κ^l.
Output: Pruned model
for i in batches **do**
  for each input sample **do**
    Capture and reinforce neuron crowding as Equation (6);
    Calculate channel priority list ρ via Equation (8);
  end for
  Model.backward();
  Calculate difference δ_i between the top-2 logits;
  if $\delta_i < \theta$ **then**
    $P'_i = \alpha * P_i$;
  else
    $P'_i = (1-\alpha) * P_i$;
  end if
end for
Averaging channel priority list P'_i under all batches;
Sorting $\{P'_j\}_{j=1}^{c^l}$ in ascending order and prune $c^l - \kappa^l$ filters with the $c^l - \kappa^l$ lowest P'_j;
return Pruned model.

4. Experiment

In this section, the proposed pruning method of channel pruning via reverse neuron crowding (CPRNC) is investigated by extensive experiments on image classification datasets CIFAR-10 (Krizhevsky and Hinton, 2009) and ImageNet1K (LSVRC-2012) (Deng et al., 2009). CIFAR-10 dataset contains 60K RGB images from 10 classes, 50K images for training, and 10K for testing. ImageNet1K dataset composes of 1.28M training images and 50k validation images from 1000 classes. We follow the data augmentation of PyTorch official example including random cropping and flipping. ResNet (He et al., 2016) with different depths are pruned to verify the effectiveness of the proposed method. For ResNet, We use the official torchvision base model for a fair comparison.

Table 1
Comparison in terms of accuracy drop and pruning ratio on CIFAR-10. The algorithms are listed in ascending order of the pruning ratio.

Model	Method	Base Acc (%)	Pruned Acc (%)	Acc↓ (%)	Params. (↓%)	FLOPs (↓%)
ResNet-20	FPGM	92.20	90.44	−1.76	133.41K(51.0)	18.77M(54.0)
	SCOP	92.22	90.75	−1.47	118.99K(56.3)	18.07M(55.7)
	CPRNC(Ours)	92.22	**90.98 ± 0.05**	**−1.24**	**117.62K(56.8)**	**17.95M(56.0)**
ResNet-32	FPGM	92.63	91.93	−0.70	229.62K(50.8)	32.35M(53.2)
	SCOP	92.66	92.13	−0.53	204.42K(56.2)	30.56M(55.8)
	CPRNC(Ours)	92.66	**92.37 ± 0.06**	**−0.29**	**204.42K(56.2)**	**30.49M(55.9)**
ResNet-56	DTP	93.36	93.46	+0.10	–	63.19M(49.7)
	HRank	93.26	93.17	−0.09	492.81K(42.4)	62.88M(50.0)
	FPGM	93.59	93.49	−0.10	422.66K(50.6)	59.60M(52.6)
	SCOP	93.70	93.64	−0.06	373.89K(56.3)	55.33M(56.0)
	CPRNC(Ours)	93.70	**93.83 ± 0.04**	**+0.13**	**373.03K(56.4)**	**54.95M(56.3)**
	GAL	93.26	91.58	−1.68	290.00K(65.9)	49.99M(60.2)
	DTP	93.36	92.46	−0.90	–	35.03M(72.1)
	CHIP	93.26	92.05	−1.21	241.27K(71.8)	34.83M(72.3)
	CPRNC(Ours)	93.70	**92.87 ± 0.06**	**−0.83**	**228.44K(73.3)**	**33.83M(73.1)**
ResNet-110	GAL	93.50	92.74	−0.76	0.95M(44.8)	130.37M(48.5)
	FPGM	93.68	93.74	+0.16	–	120.75M(52.3)
	CPRNC(Ours)	93.50	**94.16 ± 0.03**	**+0.66**	**0.82M(52.6)**	**120.00M(52.6)**
	HRank	93.50	92.65	−0.85	0.53M(68.7)	79.30M(68.6)
	CHIP	93.50	93.63	+0.13	0.55M(68.3)	71.89M(71.6)
	CPRNC(Ours)	93.50	**93.74 ± 0.03**	**+0.24**	**0.48M(72.1)**	**71.89M(71.6)**

The methods for comparison include eight static channel pruning methods: DTP (Li et al., 2023), GAL (Lin et al., 2019), FPGM (He et al., 2019), Autopruner (Luo and Wu, 2020), DepGraph (Fang et al., 2023), HRank (Lin et al., 2020), SCOP (Tang et al., 2020), ResRep (Ding et al., 2021), CHIP (Sui et al., 2021), PNNCCG (Tukan et al., 2022); three dynamic pruning methods: ManiDP (Tang et al., 2021), DSNet (Li et al., 2021), FTWT (Elkerdawy et al., 2022); five learning-based methods: MetaPruning (Liu et al., 2019), EagleEye (Li et al., 2020), NPPM (Gao et al., 2021), HALP (Shen et al., 2022), CCEP (Shang et al., 2022); All the results of the pruning ratio of FLOPs and accuracy are obtained directly from their original reports. The extent to which the model's accuracy declines after pruning is strongly associated with the pruning ratio. Specifically, as the pruning ratio (measured in terms of FLOPs reduction) increases, the loss of accuracy in the model also tends to become more pronounced. Our method CPRNC shows superior results at multiple pruning ratios, which proves its effectiveness. The commonly used ResNet architectures on CIFAR-10 and ImageNet include ResNet-20/32/56/110 and ResNet-34/50.

Implementation details. We conduct our empirical evaluations on two NVIDIA 3090Ti GPUs with PyTorch 1.10 framework in each experiment using a uniform random seed. Although training the model on ImageNet large-scale datasets is time-consuming, we perform at least three experiments and average the results. All layers are pruned with the same pruning rate following Sui et al. (2021) for a fair comparison. In the training phase, we train 8 epochs for pruning to capture and reinforce neurons with neuron crowding with a standard pre-trained model from torchvision. The experiments examining the impact of varying training epochs on the accuracy of the resulting submodel are presented in Appendix. On CIFAR-10 datasets, the learning rate, batch size, and optimizer are set to 0.01, 256, SGD, while those on ImageNet1K are 0.0001, 128, and SGD. After a quick pruning phase, the pruned network is fine-tuned for 300 epochs on CIFAR-10 with momentum, weight decay, and initial learning rate of 0.9, 0.0005, and 0.1, respectively. On the ImageNet1K dataset, fine-tuning is performed for 180 epochs with the batch size, momentum, weight decay, and initial learning rate as 256, 0.99, 0.0001, and 0.01.

4.1. Comparison with different methods on CIFAR-10

Table 1 shows the comparison of different methods on CIFAR-10. CHIP (Sui et al., 2021) and DTP (Li et al., 2023) are state-of-the-art static channel pruning method. Compared to it, our method achieves higher test accuracy with lower FLOPs. FPGM, HRank, and SCOP are all classical static pruning methods that adjust the pruning criteria from four perspectives: filter similarity, feature map rank, and scientific control. Our method designs a more precise pruning criterion through the reverse neuron crowding metric, demonstrating superior performance on different ResNet structures. The pruning ratio for each layer is the same in these methods, and we follow the same strategy to make a fair comparison. CPRNC shows superior performance even with a simple pruning strategy, while our method is flexible enough to combine with more complex pruning strategies to achieve better performance.

For the ResNet-56 model, CPRNC can bring a 0.13% accuracy increase over the baseline model with a 56.3% FLOPs reduction. On small-scale datasets, a pruned model can outperform the original model due to the sufficient training cost. Additionally, different fine-tuning settings and the new search space after pruning are also factors that affect the final performance of the model. Therefore, ResNet-56 can be fine-tuned to better accuracy when compressed by more than 2x FLOPs reduction. Compared with FPGM and SCOP, CPRNC can achieve both a larger pruning ratio and a smaller accuracy drop. High pruning rates can result in severe accuracy loss due to the inevitable permanent damage to the model structure. Therefore, static pruning methods under aggressive pruning experience even more drastic accuracy degradation. At a high FLOPs reduction of 73.1%, CPRNC still achieves fewer accuracy drops than SOTA methods CHIP and DTP with smaller model sizes.

4.2. Comparison with different methods on ImageNet

We evaluate the performance of ResNet-34/50, the most popular CNN in compression research. The ResNet structure is more compact and less redundant, making it more challenging to prune than VGG. Regarding model complexity, the pre-trained ResNet-34/50 on ImageNet consisted of 21.80/25.56 million parameters and 3.66/4.09 giga multiply-add (as a measure of FLOPs). Tables 2 and 3 summarize the performance of different approaches on the ImageNet1K dataset. We prune ResNet-50 at four different compression rates (50%, 57%, 62%, 77%) to compare with different methods. Compared to the previous methods (*e.g.*, CHIP, DSNet, CCEP), our method achieves higher accuracy (*e.g.* 76.43% with CPRNC *v.s.* 76.30% with CHIP *v.s.* 76.10% with DSNet *v.s.* 76.06% with CCEP on ResNet-50), while more FLOPs are pruned. These three methods are static pruning, dynamic pruning,

Table 2
Comparison of different methods based on ResNet-50 in terms of accuracy drop and pruning ratio on ImageNet1K. 'Baseline' and 'Pruned' denote the test accuracy of the pre-trained and pruned networks. The algorithms are listed in ascending order of the FLOPs reduction. The '–' means that the corresponding result is not provided in its original paper.

Model	Method	Static	Top-1 accuracy (%)			Top-5 accuracy (%)			FLOPs (↓ %)
			Baseline	Pruned	Acc↓	Baseline	Pruned	Acc↓	
ResNet-50	CCEP	✗	76.13	76.06	−0.07	92.86	92.81	−0.05	2.27G(44.6)
	DSNet	✗	–	76.10	–	–	–	–	2.20G(46.2)
	MetaPruning	✗	76.60	75.40	−1.20	–	–	–	2.00G(51.1)
	EagleEye	✗	–	76.40	–	–	–	–	2.00G(51.1)
	GAL	✓	76.15	71.95	−4.20	92.87	90.94	−1.93	2.33G(43.0)
	CHIP	✓	76.15	76.30	+0.15	92.87	93.02	+0.15	2.26G(44.8)
	Autopruner	✓	76.15	74.76	−1.39	92.87	92.15	−0.72	2.10G(48.7)
	CPRNC(Ours)	✓	76.15	**76.43 ± 0.07**	**+0.28**	92.87	**93.20 ± 0.08**	**+0.33**	**2.04G(50.1)**
	NPPM	✗	76.15	75.96	−0.19	92.87	92.75	−0.12	1.81G(56.0)
	CCEP	✗	76.13	75.55	−0.58	92.86	92.63	−0.23	1.79G(56.4)
	DepGraph	✓	76.15	75.83	−0.32	–	–	–	1.99G(51.8)
	FPGM	✓	76.15	74.83	−1.32	92.87	92.32	−0.55	1.90G(53.5)
	SCOP	✓	76.15	75.26	−0.89	92.87	92.53	−0.34	1.86G(54.6)
	DTP	✓	76.13	75.55	−0.58	–	–	–	1.77G(56.7)
	CPRNC(Ours)	✓	76.15	**75.96 ± 0.05**	**−0.19**	92.87	**92.86 ± 0.07**	**−0.01**	**1.77G(56.7)**
	HALP	✗	76.15	74.30	−1.85	–	–	–	1.51G(63.0)
	CCEP	✗	76.13	74.87	−1.26	92.86	92.35	−0.51	1.47G(64.1)
	DTP	✓	76.13	75.24	−0.89	–	–	–	1.60G(60.9)
	PNNCCG	✓	76.22	75.13	−1.09	–	–	–	1.57G(61.5)
	HRank	✓	76.15	71.98	−4.17	92.87	91.01	−1.86	1.55G(62.1)
	ResRep	✓	76.15	75.30	−0.85	92.87	92.47	−0.40	1.55G(62.1)
	CHIP	✓	76.15	75.26	−0.89	92.87	92.53	−0.34	1.52G(62.8)
	CPRNC(Ours)	✓	76.15	**75.45 ± 0.02**	**−0.70**	92.87	**92.62 ± 0.08**	**−0.25**	**1.55G(62.1)**
	Autopruner	✓	76.15	73.05	−3.10	92.87	91.25	−1.62	1.39G(66.0)
	HRank	✓	76.15	69.10	−7.05	92.87	89.58	−3.29	0.98G(76.0)
	CPRNC(Ours)	✓	76.15	**73.05 ± 0.07**	**−3.10**	92.87	**91.27 ± 0.10**	**−1.60**	**0.95G(76.7)**

and learning-based pruning methods. This result confirms the superior performance of our method on large-scale datasets.

ResNet-50. For a fair comparison with other static pruning methods, all layers in CPRNC are pruned with the same pruning rate. 50% and 56% are common pruning ratios in static pruning methods. With FLOPs reduction around 56%, DepGraph and DTP reach 75.83% and 75.55% top-1 accuracy at 51.8% and 56.7% compression, respectively, while CPRNC reaches a better accuracy 75.96% at a higher compression rate. With FLOPs reduction around 50%, CPRNC achieves a top-1 accuracy of 76.43%. While CHIP reaches 76.30% top-1 accuracy at 44.8% compression, CPRNC achieves 0.13% higher accuracy than CHIP at a higher compression of 50.1%. At high pruning rates, CPRNC also achieves higher performance at lower FLOPs than competing methods. HRank and ResRep accelerate ResNet-50 by a 62.1% speedup rate with 71.98% and 75.30% top-1 accuracy, respectively, while CPRNC reaches a higher accuracy by 75.45% at the same compression. With FLOPs reduction around 76%, CPRNC achieves a lower accuracy drop of 3.10%, which is better than HRank of 7.05%. This result demonstrates that CPRNC mitigates the performance degradation of sub-networks at high pruning rates.

Dynamic pruning methods retain the complete original network and achieve acceleration by dynamically routing sub-networks during inference. Although dynamic pruning methods are limited in practical deployment, some methods (Tang et al., 2021; Li et al., 2021) achieve superior results on ImageNet. CPRNC, a static pruning method, achieves competitive performance with dynamic pruning methods on ImageNet. DSNet is a dynamic channel pruning method that achieves 76.10% accuracy at 46.2% compression. CPRNC performs better at 50.1% compression with a top-1 accuracy of 0.48% higher. EagleEye is a learning-based pruning method that achieves 76.40% accuracy at 51.1% compression. CPRNC achieves higher performance by 76.43% at 50.1% compression. Learning-based pruning methods are like algorithms of neural architecture search rather than pruning methods. Such as, CCEP relies on an evolutionary algorithm to search for an optimal sub-network. The training budget for such algorithms is hefty, while that of CPRNC for pruning is almost negligible. Compared to CCEP, CPRNC has a lower accuracy drop of 0.19% at a similar FLOPs reduction of 56.7%, which is better than CCEP 0.58%.

ResNet-34. Some pruning methods do not provide results on ResNet-50, such as ManiDP and FTWT, so we follow them to provide results on ResNet-34 for comparison. Both FTWT and ManiDP are recent compression research about dynamic pruning. CPRNC accelerates ResNet-34 by a 49.5% FLOPs reduction with 73.41% top-1 accuracy, while FTWT and ManiDP reach lower accuracy and compression rate. Compared with static pruning methods (*e.g.* 72.63% with FPGM at 41.1% compression), CPRNC has a noticeable accuracy improvement.

4.3. Ablation study

Varying pruning rate. We follow SCOP (Tang et al., 2020) to provide the accuracy variation of ResNet-56 on CIFAR-10 between pruning rates of 10% and 80%. A pruning rate of 0 represents the accuracy of the original model as a baseline. For a fair comparison, we use the same parameter settings in training. The accuracy of the pruned network with different pruning ratios is shown in Fig. 4. Excessive reduction in the number of channels will inevitably impair the original network performance. Our method also performs better than other state-of-the-art methods (*e.g.* SCOP, CCEP) at higher pruning ratios. The accuracy under aggressive pruning does not match our expectations, and this is related to the pruning strategy with the same pruning ratio in each layer. In addition, our method achieves a slight accuracy improvement over the baseline model at a low pruning rate. We attribute this to the effect of relatively sufficient fine-tuning epochs on the small-scale dataset.

Effectiveness of excavating neuron crowding. We perform ablation experiments using ResNet-56 on CIFAR-10 and ResNet-34 on ImageNet to demonstrate the effectiveness of NCDM and NCDR in excavating neuron crowding, respectively. Even on the large-scale dataset ImageNet we still perform at least three experiments and take the average results. The impacts of NCDM or NCDR are empirically investigated in Table 4. Without NCDM and NCDR, the simplest channel pruning (pruning only by CPC (Channel Priority Calculation) in Section 3.3 under inputs A^l) cannot accurately identify the more competitive channels. At this point, the method degrades to a feature map pruning method based on the L2-norm criterion similar to Li et al. (2017), which brings a severe

Table 3
Comparison of different methods based on ResNet-34 in terms of accuracy degradation and pruning rate on ImageNet1K. 'Baseline' and 'Pruned' denote the test accuracy of the pre-trained and pruned networks. The '–' means that the corresponding result is not provided in its original paper.

Model	Method	Static	Top-1 accuracy (%)			Top-5 accuracy (%)			FLOPs (↓ %)
			Baseline	Pruned	Acc↓	Baseline	Pruned	Acc↓	
ResNet-34	FTWT	✗	73.30	72.17	−1.13	–	–	–	1.92G(47.4)
	NPPM	✗	73.30	73.01	−0.29	91.42	91.30	−0.12	2.06G(44.0)
	ManiDP	✗	73.30	73.29	−0.01	91.42	91.42	0.00	1.95G(46.8)
	FPGM	✓	73.92	72.63	−1.29	91.62	91.08	−0.54	2.16G(41.1)
	CPRNC(Ours)	✓	73.30	**73.41 ± 0.05**	**+0.11**	91.42	**91.44 ± 0.02**	**+0.02**	**1.85G(49.5)**

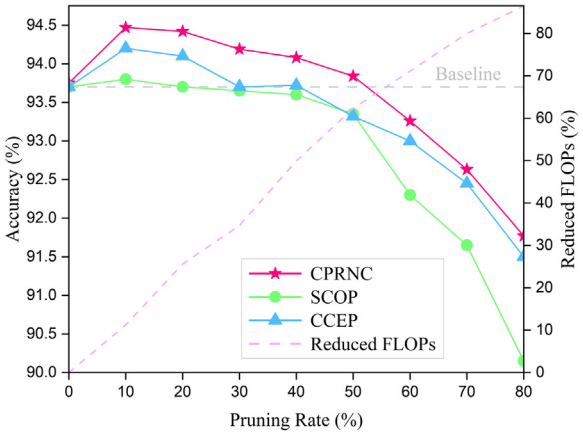


Fig. 4. The accuracy of the pruned ResNet-56 vary *w.r.t* pruning rate on CIFAR-10. The reduction in FLOPs of the pruned model is not strictly linearly related to the set pruning rate. CPRNC outperforms competing methods at different pruning rates.

Table 4
Effectiveness of excavating neuron crowding. The top-1 accuracy of the pruned networks is reported. 'Gap' means the difference in accuracy after applying NCDM or NCDR.

Dataset	Model	NCDM	NCDR	Acc (%)	Gap (%)
CIFAR-10	ResNet-56	✗	✗	93.22 ± 0.12	–
		✓	✗	93.70 ± 0.03	+0.48
		✓	✓	93.83 ± 0.04	+0.61
ImageNet	ResNet-34	✗	✗	71.88 ± 0.10	–
		✓	✗	73.13 ± 0.08	+1.25
		✓	✓	73.41 ± 0.05	+1.53

accuracy drop. The degraded method struggles to precisely estimate the channel importance, which results in a severe drop in accuracy. The accuracy of the pruning methods without NCDM and NCDR is 93.22% and 71.88% on CIFAR-10 and ImageNet, respectively, which is 0.48% and 1.42% lower than the original model. Experiments validate the effectiveness of NCDM and NCDR (*e.g.*+0.61% on CIFAR-10 and +1.53% on ImageNet) that avoid the performance degradation. CPRNC is a data-driven pruning method. On CIFAR-10, we exhibit marginal differences compared to other competitive methods, primarily due to the limited sample numbers of the dataset. Our approach demonstrates more significant performance improvement on the large-scale ImageNet dataset. NCDM effectively aggregates the informative neurons in the channel, and NCDR further screens for discriminative samples to recalibrate reverse neuron crowding. CPRNC provides a more efficient and accurate perspective than the previous pruning criterion.

5. Conclusion

This paper proposes a lightweight CPRNC method to excavate efficient sub-networks and provides a new perspective on neuron crowding for channel pruning criteria. By systematically exploring visual crowding theories, we construct a cost to perform relational modeling of artificial neurons in the neural network for guiding pruning. In NCDM, the assessment of channel importance scores on the extent of reverse neuron crowding within the channel. This approach affords a more fine-grained and precise perspective than prior pruning criteria. In the training phase, the neuron-wise crowding reinforcement applies to capture informative neurons with reverse neuron crowding, which facilitates the acquisition of discriminative channels. Then NCDR further excavates the relationship between input samples and neuron crowding, which improves the probability that the model learns rich knowledge. NCDR enhances the accuracy and rationality of the channels priority list used to obtain sub-networks by recalibrating reverse neuron crowding evaluation scores with the aid of discriminative samples. Compared with the state-of-the-art methods, the pruned networks obtained by CPRNC perform better with less computational cost. Extensive experiments are conducted on several benchmarks to verify the effectiveness of our method.

However, our pruning strategies limit channel pruning only within residual connections to ensure output dimensions remain consistent while applying a fixed pruning ratio for each layer. It is sub-optimal, although we achieve superior performance. On the other hand, there are still several parameters in our method. In the future, we will address these issues better and achieve more efficient model compression.

CRediT authorship contribution statement

Pingfan Wu: Conceptualization, Methodology, Validation, Visualization, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Hengyi Huang:** Writing – review & editing, Supervision. **Han Sun:** Writing – review & editing, Supervision. **Dong Liang:** Writing – original draft, Supervision. **Ningzhong Liu:** Writing – review & editing, Project administration, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors gratefully acknowledge support from the Natural Science Foundation of Jiangsu Province of China (BK20222012), Guangxi Science and Technology Project (AB22080026/2021AB22167), National Natural Science Foundation of China (No. 61375021).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cviu.2024.103942.

References

Alqahtani, A., Xie, X., Jones, M.W., Essa, E., 2021. Pruning CNN filters via quantifying the importance of deep visual representations. Comput. Vis. Image Underst. 208–209, 103220.

Aubry, M., Russell, B.C., Sivic, J., 2014. Painting-to-3D model alignment via discriminative visual elements. ACM Trans. Graph. 33 (2), 14:1–14:14.

Bonnaerens, M., Freiberger, M., Dambre, J., 2022. Anchor pruning for object detection. Comput. Vis. Image Underst. 221, 103445.

Bouma, H., 1970. Interaction effects in parafoveal letter recognition. Nature 226 (5241), 177–178.

Cai, H., Gan, C., Wang, T., Zhang, Z., Han, S., 2020. Once-for-all: Train one network and specialize it for efficient deployment. In: Proceedings of the 8th International Conference on Learning Representations.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255.

Ding, X., Hao, T., Tan, J., Liu, J., Han, J., Guo, Y., Ding, G., 2021. Resrep: Lossless cnn pruning via decoupling remembering and forgetting. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4490–4500.

Elkerdawy, S., Elhoushi, M., Zhang, H., Ray, N., 2022. Fire together wire together: A dynamic pruning approach with self-supervised mask prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp. 12454–12463.

Fang, G., Ma, X., Song, M., Mi, M.B., Wang, X., 2023. Depgraph: Towards any structural pruning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 16091–16101.

Flom, M.C., Heath, G.G., Takahashi, E., 1963. Contour interaction and visual resolution: Contralateral effects. Science 142 (3594), 979–980.

Gao, S., Huang, F., Cai, W., Huang, H., 2021. Network pruning via performance maximization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9270–9280.

Gao, X., Zhao, Y., Dudziak, L., Mullins, R.D., Xu, C., 2018. Dynamic channel pruning: Feature boosting and suppression. In: Proceedings of the 7th International Conference on Learning Representations.

Han, S., Mao, H., Dally, W.J., 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In: Proceedings of the 4th International Conference on Learning Representations.

Han, S., Pool, J., Tran, J., Dally, W.J., 2015. Learning both weights and connections for efficient neural network. In: Advances in Neural Information Processing Systems. pp. 1135–1143.

Hassibi, B., Stork, D., 1992. Second order derivatives for network pruning: Optimal brain surgeon. Adv. Neural Inf. Process. Syst. 5.

He, Y., Liu, P., Wang, Z., Hu, Z., Yang, Y., 2019. Filter pruning via geometric median for deep convolutional neural networks acceleration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4340–4349.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Hinton, G.E., Vinyals, O., Dean, J., 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531. 2.

Khan, M., El Saddik, A., Alotaibi, F.S., Pham, N.T., 2023. AAD-Net: Advanced end-to-end signal processing system for human emotion detection & recognition using attention-based deep echo state network. Knowl.-Based Syst. 270, 110525.

Krizhevsky, A., Hinton, G., 2009. Learning Multiple Layers of Features from Tiny Images. Technical Report.

LeCun, Y., Denker, J.S., Solla, S.A., 1989. Optimal brain damage. In: Advances in Neural Information Processing Systems. pp. 598–605.

Li, Y., van Gemert, J.C., Hoefler, T., Moons, B., Eleftheriou, E., Verhoef, B., 2023. Differentiable transportation pruning. In: Proceedings of the IEEE International Conference on Computer Vision.

Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P., 2017. Pruning filters for efficient ConvNets. In: Proceedings of the 5th International Conference on Learning Representations.

Li, C., Wang, G., Wang, B., Liang, X., Li, Z., Chang, X., 2021. Dynamic slimmable network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8607–8617.

Li, B., Wu, B., Su, J., Wang, G., 2020. EagleEye: Fast sub-net evaluation for efficient neural network pruning. In: Proceedings of the European Conference on Computer Vision. pp. 639–654.

Lin, M., Ji, R., Wang, Y., Zhang, Y., Zhang, B., Tian, Y., Shao, L., 2020. Hrank: Filter pruning using high-rank feature map. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1529–1538.

Lin, S., Ji, R., Yan, C., Zhang, B., Cao, L., Ye, Q., Huang, F., Doermann, D.S., 2019. Towards optimal structured CNN pruning via generative adversarial learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2790–2799.

Lin, M., Zhang, Y., Li, Y., Chen, B., Chao, F., Wang, M., Li, S., Tian, Y., Ji, R., 2023. 1XN pattern for pruning convolutional neural networks. IEEE Trans. Pattern Anal. Mach. Intell. 45 (4), 3999–4008.

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C., 2017. Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2736–2744.

Liu, Z., Mu, H., Zhang, X., Guo, Z., Yang, X., Cheng, K., Sun, J., 2019. Metapruning: Meta learning for automatic neural network channel pruning. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3296–3305.

Luo, J., Wu, J., 2020. AutoPruner: An end-to-end trainable filter pruning method for efficient deep model inference. Pattern Recognit. 107, 107461.

Mondal, M., Das, B., Roy, S.D., Singh, P., Lall, B., Joshi, S.D., 2022. Adaptive CNN filter pruning using global importance metric. Comput. Vis. Image Underst. 222, 103511.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems. pp. 91–99.

Shang, H., Wu, J., Hong, W., Qian, C., 2022. Neural network pruning by cooperative coevolution. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 4814–4820.

Shen, M., Yin, H., Molchanov, P., Mao, L., Liu, J., Alvarez, J.M., 2022. Structural pruning via latency-saliency knapsack. In: Advances in Neural Information Processing Systems.

Sui, Y., Yin, M., Xie, Y., Phan, H., Zonouz, S.A., Yuan, B., 2021. CHIP: Channel independence-based pruning for compact neural networks. In: Advances in Neural Information Processing Systems. pp. 24604–24616.

Tang, Y., Wang, Y., Xu, Y., Deng, Y., Xu, C., Tao, D., Xu, C., 2021. Manifold regularized dynamic network pruning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5018–5028.

Tang, Y., Wang, Y., Xu, Y., Tao, D., Xu, C., Xu, C., Xu, C., 2020. Scop: Scientific control for reliable neural network pruning. In: Advances in Neural Information Processing Systems. pp. 10936–10947.

Tian, Q., Arbel, T., Clark, J.J., 2023. Grow-push-prune: Aligning deep discriminants for effective structural network compression. Comput. Vis. Image Underst. 231, 103682.

Tripathy, S.P., Levi, D.M., 1994. Long-range dichoptic interactions in the human visual cortex in the region corresponding to the blind spot. Vis. Res. 34 (9), 1127–1138.

Tukan, M., Mualem, L., Maalouf, A., 2022. Pruning neural networks via coresets and convex geometry: Towards no assumptions. In: Advances in Neural Information Processing Systems.

Whitney, D., Levi, D.M., 2011. Visual crowding: a fundamental limit on conscious perception and object recognition. Trends in Cognitive Sciences 15 (4), 160–168.

Zhao, B., Cui, Q., Song, R., Qiu, Y., Liang, J., 2022. Decoupled knowledge distillation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11943–11952.

Zhou, A., Ma, Y., Zhu, J., Liu, J., Zhang, Z., Yuan, K., Sun, W., Li, H., 2021. Learning N: M fine-grained structured sparse neural networks from scratch. In: Proceedings of the 9th International Conference on Learning Representations.