MIEP: Channel Pruning with Multi-granular Importance Estimation for Object Detection

Liangwei Jiang
State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
Beijing, China
lw_jiang@buaa.edu.cn

Jiaxin Chen*
State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
Beijing, China
jiaxinchen@buaa.edu.cn

Di Huang
School of Computer Science and Engineering, Beihang University
Beijing, China
dhuang@buaa.edu.cn

Yunhong Wang
State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
Beijing, China
yhwang@buaa.edu.cn

ABSTRACT

This paper investigates compressing a pre-trained deep object detector to a lightweight one by channel pruning, which has proved effective and flexible in promoting efficiency. However, the majority of existing works trim channels based on a monotonous criterion for general purposes, *i.e.*, the importance to the task-specific loss. They are prone to overly prune intermediate layers and simultaneously leave large intra-layer redundancy, severely deteriorating the detection accuracy. To address the issues above, we propose a novel channel pruning approach with multi-granular importance estimation (MIEP), consisting of the **F**eature-level **O**bject-sensitive **I**mportance (FOI) and the **I**ntra-layer **R**edundancy-aware **I**mportance (IRI). The former puts large weights on channels that are critical for object representation through the guidance of object features from the pre-trained model, and mitigates over-pruning when combined with the task-specific loss. The latter groups highly correlated channels based on clustering, which are subsequently pruned with priority to decrease redundancy. Extensive experiments on the COCO and VOC benchmarks demonstrate that MIEP remarkably outperforms the state-of-the-art channel pruning approaches, achieves a better balance between accuracy and efficiency compared to lightweight object detectors, and generalizes well to various detection frameworks (*e.g.*, Faster-RCNN and FSAF) and tasks (*e.g.*, classification).

CCS CONCEPTS

• **Computing methodologies** → **Object detection**; • **Computer systems organization** → **Neural networks**.

*Corresponding author.

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3612563

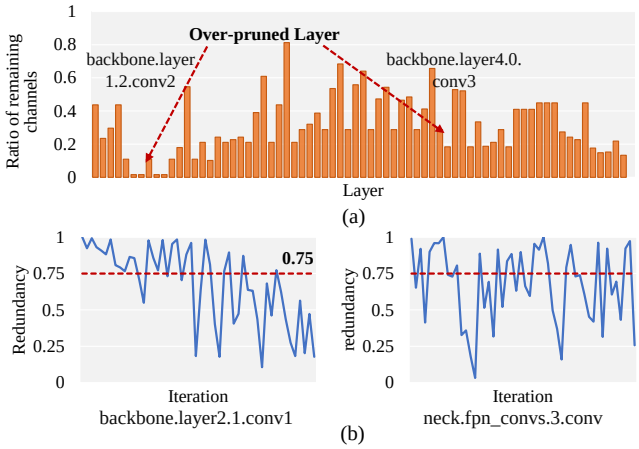


Figure 1: (a) Ratio of remaining channels after pruning. (b) Redundancy (see definition in Eq.(3)) of pruned channels.

KEYWORDS

Model pruning, object detection, inference acceleration

ACM Reference Format:
Liangwei Jiang, Jiaxin Chen, Di Huang, and Yunhong Wang. 2023. MIEP: Channel Pruning with Multi-granular Importance Estimation for Object Detection. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3581783.3612563

1 INTRODUCTION

Object detection [12, 37, 50, 65, 84] is a fundamental task in computer vision and has a wide range of applications in multimedia content understanding such as autonomous driving [3, 46], cross-modal retrieval [53, 61], image caption [1, 74], and optical character recognition [60, 63]. Along with the recent progress in deep learning, the accuracy of object detection has been remarkably promoted [29, 31, 57]. Nevertheless, the high computational cost of deep models significantly influences the efficiency, severely impeding their applicability to various practical scenarios, especially when

deployed on resource-constrained platforms such as unmanned vehicles, mobile devices, and edge devices.

Many efforts have been devoted to balancing detection accuracy and efficiency. The representatives of them, including MobileDets [67], NanoDet [49], YOLO-Tiny [24] and PP-PicoDet [71], concentrate on devising lightweight backbone and head neural networks. However, their capability to reduce computation consumption is restricted by the specific architectures designed by hand, substantially limiting the flexibility of applying them to distinct hardware.

Recently, channel or filter pruning has been extensively studied for compressing deep neural networks [35, 36, 54, 62, 68], showing great potential in boosting the efficiency of deep models. Generally, it estimates the importance of each channel or filter, trims those less important ones by a controllable pruning ratio, and finally fine-tunes the pruned model, making it flexible and friendly for practical deployment. As the most critical procedure, the importance estimation methods can be roughly divided to the task-specific loss based [23, 47, 48, 55], the magnitude-based [18, 22, 30, 69, 75], the redundancy-based [8, 19, 34], the reconstruction-based [20, 25, 44], as well as the learning-based [10, 28, 51, 58] ones. However, most existing approaches are developed for the classification task, failing to handle the prevailing coupling structures (*e.g.* FPN and the shared head), thus not suitable for object detection. A few works attempt to compress the detector by employing the channel-wise importance based on the task-specific (*i.e.* the classification and regression losses for object detection) [23, 39] or the reconstruction loss [21, 66]. Despite the improved performance, they have several limitations: 1) the channel importance *w.r.t.* object detection is determined by a single holistic loss-based criterion (*e.g.* the task-specific loss or the reconstruction loss), making some of the intermediate layers prone to be overly pruned as illustrated in Fig. 1(a), thus degrading the final performance after pruning; 2) the redundancy among channels is not taken into account in importance estimation, incurring poor diversity after pruning as shown in Fig. 1(b).

To address the issues above, we propose a novel deep model compression approach for object detection, namely channel pruning with multi-granular importance estimation (MIEP). Specifically, besides the conventional channel importance measured by their separate influence to the task-specific loss (TLI) , we firstly present a Feature-level Object-sensitive Importance (FOI). It leverages the features from the intermediate layers of the full model as guidance, estimating the channel importance by the cosine similarity between the feature containing that specific channel and the corresponding one from the full model. As restrained by maintaining high feature-level similarity, FOI inclines to avoid over-pruning, considering that excessively trimming a certain layer undoubtedly leads to the sharp drop in similarities. In order to further select the channels that are critical for object localization, we further generate an approximate object mask based on the predicted detection results, and impose larger weights on foreground areas when computing the feature-level similarity. In regards of the redundancy issue, we develop the Intra-layer Redundancy-aware Importance (IRI). It represents each channel by aggregating the filters connected, based on which the overall channels inside a certain layer is divided into distinct groups via agglomerative hierarchical clustering [64]. The channels falling into the majority groups are assigned smaller importance weights, thus favoring pruning redundant ones with priority. Finally, FOI, IRI and TLI are integrated as a multi-granular importance estimation, facilitating attaining a compact and representative pruned model and reaching a better balance in accuracy and efficiency for object detection. Additionally, in order to deal with the coupling structures in complex deep object detectors, we employ a fine-grained layer grouping technique, which greatly strengthens the generalizability of MIEP to different detection frameworks.

The main contribution of this paper lies in three-fold:

1) We propose a novel channel pruning approach based on multi-granular importance estimation, (*i.e.* MIEP), for efficient object detection on resource-constrained devices.

2) We develop a Feature-level Object-sensitive Importance and an Intra-layer Redundancy-aware Importance to mitigate the over-pruning and the intra-layer redundancy respectively, a combination of which facilitates attaining a compact and representative pruned model, thus reaching an improved performance in balancing efficiency and accuracy.

3) We extensively evaluate the performance of the proposed method on public benchmarks for object detection, showing that MIEP archives better performance compared to the state-of-the-art model pruning approaches and the lightweight object detectors; and demonstrate its generalizability to distinct detectors and tasks.

2 RELATED WORK

2.1 General Object Detection

Existing CNN-based object detection frameworks are roughly categorized as the anchor-based one and the anchor-free one, according to whether using pre-defined anchors for generating object proposals. As for the former, representative multi-stage approaches, including R-CNN [14], Faster-RCNN [50] and Mask-RCNN [17], first generate region proposals and subsequently localize target objects. On the contrary, the one-stage methods, such as SSD [40], RetinaNet [37] and GFL [32, 33], densely predict the location and category of objects without proposals. As for the latter such as Centernet [81], FCOS [56] and FSAF [82], the anchors are replaced by efficient alternatives such as centerness constraints or object heatmaps. Despite the reduced computational cost, it is still challenging for deployment on resource-constrained devices.

2.2 Lightweight Object Detection

Great efforts have been devoted to efficient object detection by designing lightweight network structures. YOLObile [4] accomplishes real-time object detection on mobile devices based on YOLO-v4 [2]. NanoDet and NanoDet-Plus [49] adopt ShuffleNetV2 [45] as the backbone, and promote the accuracy by combining ATSS [77] and GFL [32]. MobileDets [67] boosts the latency-accuracy trade-off via neural architecture search (NAS). PP-PicoDet [71] hybridly leverages the hand-crafted structures and NAS, further improving the performance. Besides, the YOLO based detectors [13, 24, 57] also deliver computationally efficient variants (*e.g.* tiny or nano). However, they are not flexible for various kinds of hardware.

2.3 Model Pruning for Efficient Object Detection

Channel or filter pruning aims to remove unnecessary neural network structures whilst maintaining accuracy under constraints of low memory, computation, and latency. According to how the

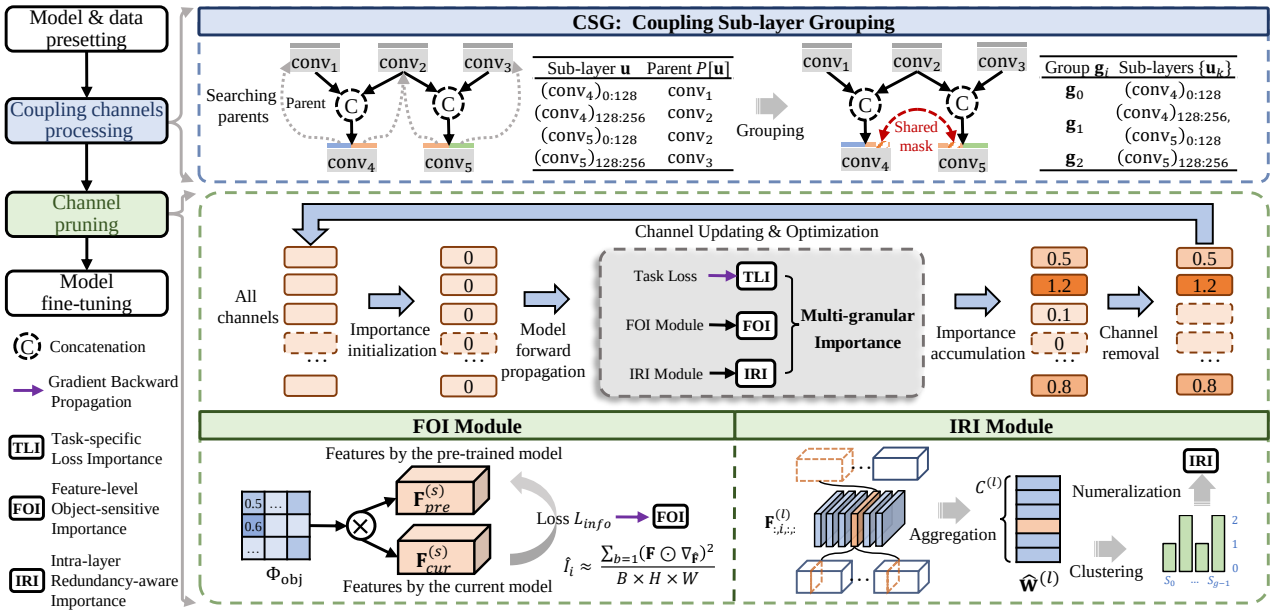


Figure 2: Framework overview of MIEP. The Coupling Sub-layer Grouping is firstly applied to deal with the coupling channels. In the channel pruning stage, the channel-wise importance is estimated by jointly computing the conventional task-specific loss based importance (TLI), the proposed Feature-level Object-sensitive Importance (FOI) and Intra-layer Redundancy-aware Importance (IRI), based on which the channel with the least importance is trimmed. The above process is repeated until reaching the preset FLOPs. Finally, the pruned model is fine-tuned for deployment.

importance is estimated, current channel pruning approaches are roughly divided into the following five categories: (i) based on task-specific loss [39, 47, 48, 55]; (ii) based on magnitude magnitude [18, 22, 30, 75]; (iii) based on redundancy [8, 19, 34]; (iv) based on reconstruction-based objectives [20, 25, 44]; and (v) based on network learning [10, 28, 51, 58]. Besides, there are several special heuristic-based importance estimation methods, such as Bayesian [43], Variational [79], and DMCP [16]. The above methods are devised for general purposes, thus not being specifically optimized for object detection. Recently, a few works [39, 66, 73, 76] have investigated pruning object detectors. Typically, SlimYolov3 [76] reuses Bacth Normalization (BN) layer scaling factors as importance and retains channels with large factors. Slim [73] trains a shared sub-network with switchable BN, adjusting the network width on the fly. GroupFisher [39] deals with coupled channels prevailing in object detectors, and boosts the accuracy-efficiency trade-off.

Despite achieving promising results, most current approaches employ a monotonous criterion for channel-wise importance estimation, and have limitations in tackling with coupling channels, thus leaving much room for improvement.

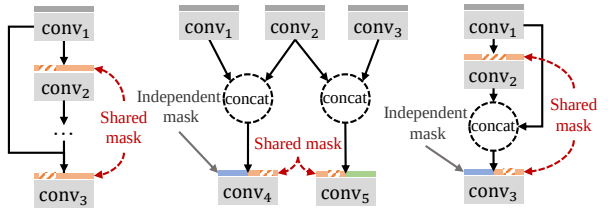
3 METHODOLOGY

3.1 Framework Overview

Given a deep object detector with L layers trained by a task-specific loss $\mathcal{L}_{task}$, the parameterized weight, bias and feature map in the l-th layer are denoted by $\mathbf{W}^{(l)} \in \mathbb{R}^{C^{(l+1)} \times C^{(l)} \times K_h^{(l)} \times K_w^{(l)}}$, $\mathbf{B}^{(l)} \in \mathbb{R}^{C^{(l)}}$ and $\mathbf{F}^{(l)} \in \mathbb{R}^{B \times C^{(l)} \times H^{(l)} \times W^{(l)}}$ respectively, where B, $C^{(l)}$, $H^{(l)}$, $W^{(l)}$, $K_h^{(l)}$ and $K_w^{(l)}$ indicate the bathc size, channel size, the feature map height and width, the kernel height and width of the l-th layer, respectively. K_h and K_w equal 1 for the fully-connected (FC) layer.

Similar to [39], as shown in Fig. 2, our approach mainly consists of three steps: 1) handling the coupling channels such as those connected by skip connections in the feature pyramid network (FPN), concatenation or their combination as displayed in Fig. 3, 2) pruning channels to a fixed ratio based on the channel importance, and 3) finetuning the pruned model on the training data.

In the first step, as illustrated in Fig. 3, channels in different layers of a deep object detector may correlate due to inter-layer connections including the skip connection in FPN, the concatenation operation such as those in ShuffleNet [45] and PAN [24], or their combination, which should be synchronously processed in channel pruning [70, 78]. To that end, we present a Coupling Sub-layer Grouping (CSG) algorithm to explore the correlation among channels. Specifically, CSG firstly builds the computational graph $\mathcal{G}$ of an object detector [11, 23], where each node represents operations such as convolution, batch normalization and concatenation. Subsequently, CSG divides the channels in layer l into different sub-layers $\{\mathbf{u}_k\}$, where channels in $\mathbf{u}_k$ have the same source. As shown in Fig. 3(b), the layer that contains a concatenation operation will be split into two sub-layers (*i.e.* the blue bar and orange bar before $conv_4$). Afterwards, CSG utilizes the depth-first search (DFS) to further bind its sources $P[\mathbf{u}]$ for each sub-layer, and the sub-layers with the same source form a coupled group g. For instance, in Fig. 3(a), $conv_2$ is coupled with $conv_3$ since they have the same parent $conv_1$. The overall procedure is summarized in Algorithm 1.



(a) inter-layer connection (b) concatenation (c) hybrid coupling

Figure 3: Different coupling structures in object detectors: (a) the inter-layer connection such as the skip connection; (b) the concatenation; (c) combination of the inter-layer connection and concatenation. In (a), the pruning mask is completely shared for the layer with coupling channels. In (b), the pruning mask is partially shared, where the shared part locates at the concatenated channels derived from the same source (*e.g.* conv2). In (c), the mask is processed by combining (a) and (b). Note that the bars with the same color indicate a shared pruning mask, and the red stripes correspond to pruned channels.

In the second step, MIEP aims to compute a binary mask $\mathbf{m}^{(l)} \in \{0,1\}^{C^{(l)}}$ for each $\mathbf{F}^{(l)}$, where the masked feature $\hat{\mathbf{F}}^{(l)} = \mathbf{F}^{(l)} \odot \mathbf{m}^{(l)}$ are adopted for successive computation during pruning, and the k-th channel in $\mathbf{F}^{(l)}$ with the corresponding mask value $\mathbf{m}_k^{(l)} = 0$ is directly discarded during inference, thus saving the computation cost. $\mathbf{m}^{(l)}$ is usually initialized with all ones, iteratively checks and sets $\mathbf{m}_k^{(l)} = 0$ if the k-th channel has the lowest importance, until reaching the pruning ratio. As in [39], the importance $\hat{I}_i^{(l)}$ of the i-th channel in the l-th layer is typically estimated by its influence on the task-specific loss $\mathcal{L}_{task}$, formulated as below according to Taylor expansion [26] and Fisher information [55]:

$$\begin{aligned}\hat{I}_i^{(l)} &= \frac{\mathcal{L}_{task}(\mathbf{1}-\mathbf{e}_i) - \mathcal{L}_{task}(\mathbf{1})}{H^{(l)} \times W^{(l)}} \\ &\approx \frac{1}{B \times H^{(l)} \times W^{(l)}} \sum_{b=1}^{B} \|\mathbf{F}^{(l)} \odot \nabla_{\hat{\mathbf{F}}^{(l)}}\|^2,\end{aligned} \tag{1}$$

where $\mathcal{L}_{task}$ is the task-specific loss, $\mathbf{1} \in \mathbb{R}^{C^{(l)}}$ is the all-one vector, $\mathbf{e}_i \in \mathbb{R}^{C^{(l)}}$ denotes the one-hot vector of which the i-th element equals 1, $\odot$ indicates the broadcasting point-wise multiplication, and $\nabla_{\hat{\mathbf{F}}^{(l)}}$ is the gradient of $\hat{\mathbf{F}}^{(l)}$ w.r.t. $\mathcal{L}_{task}$. To deal with the coupling channels shown in Fig. 3, based on the output $\mathbb{G} = \{\mathbf{g}_i\}$ of CSG in Algorithm 1, channels from the sub-layers $\{\mathbf{u}_k\}$ belonging to the same $\mathbf{g}_i$ share the same pruning mask. In the meantime, the importance of a particular channel is reformulated by summing up the importance of all coupling channels as follows:

$$I_{TLI,i}^{(l)} = \hat{I}_i^{(l)} + \sum_{t \neq l} \hat{I}_j^{(t)}, \tag{2}$$

where the i-th channel in layer l and the j-th channel in layer t are coupling ones, *i.e.* from the same index position in sub-layers $\{\mathbf{u}_k\}$ belonging to the same $\mathbf{g}_i$.

However, using $\{I_{TLI,i}^{(l)}\}$ as the single importance criterion always incurs the over-pruning and intra-layer redundancy issues.

Algorithm 1: Coupling Sub-layer Grouping

Input: The computational graph $\mathcal{G}$ with L layers.
Output: Groups of sub-layers $\mathbb{G} = \{\mathbf{g}_i\}$ where $\mathbf{g}_i = \{\mathbf{u}_k\}$.

Initialize $\mathbb{G} = \emptyset$.
for $l = 1, \cdots, L$ **do**
  Divide the channels in layer l to disjoint sub-layers $\{\mathbf{u}\}$.
  for $\mathbf{u}$ *in layer* l **do**
    Find the parents $P[\mathbf{u}]$ of $\mathbf{u}$ in $\mathcal{G}$ via DFS.
  end
end
for $l = 1, \cdots, L$ **do**
  for $\mathbf{u}$ *in layer* l **do**
    Assign $\mathbf{g}_{new} := \{\mathbf{u}\}$, $P_{new} := P[\mathbf{u}]$.
    for $\mathbf{g} \in \mathbb{G}$ **do**
      if $P[\mathbf{u}] \cap P[\mathbf{g}] \neq \emptyset$ **then**
        $\mathbf{g}_{new} := \mathbf{g}_{new} \cup \mathbf{g}$, $P_{new} := P_{new} \cup P[\mathbf{g}]$.
        Delete $\mathbf{g}$.
      end
    end
    Update $\mathbb{G} := \mathbb{G} \cup \{\mathbf{g}_{new}\}$, $P[\mathbf{g}_{new}] := P_{new}$.
  end
end

For instance, Fig. 1(a) displays the percentage of remaining channels in each layer of the GFL detector [33] after pruning by $I_{TLI,i}^{(l)}$ to 10% GLOPs, indicating that some intermediate layers such as *backbone.layer1.2.conv2* and *backbone.layer4.0.conv3* are overly trimmed. As for the redundancy, by following [19, 34], we use the definition of redundancy $R_i^{(l)}$ for the i-th channel in the l-th layer as below:

$$R_i^{(l)} = 1 - \frac{1}{|\mathbf{m}^{(l)}|} \sum_{j \in r^{(l)}} \mathbb{1}[d_j^{(l)} < d_i^{(l)}], \tag{3}$$

where $\mathbb{1}[\cdot]$ is the indicator function, $\hat{\mathbf{W}}_i^{(l)}$ is the aggregation weight for the i-th channel, and $r^{(l)} = \{i | 0 \leq i < C^{(l)}, \text{ and } \mathbf{m}_i^{(l)} = 1\}$ indicates the set of untrimmed channel indices. $d_i^{(l)} = \min_{j \in r^{(l)}, j \neq i} ||\hat{\mathbf{W}}_i^{(l)} - \hat{\mathbf{W}}_j^{(l)}||_2$ is the distance between the i-th channel and the remaining channels. A small (large) $d_i^{(l)}$ implies that the i-th channel is similar (dissimilar) to the other channels. Fig. 1(b) visualizes the redundancy of pruned channels, indicating that a large portion of channels with low redundancy are trimmed, leaving many redundant ones. The above two issues deteriorate the representative capability of the pruned network. To address these two issues, we propose the MIEP method based on multi-granular importance estimation, by developing Feature-level Object-sensitive Importance (FOI) and Intra-layer Redundancy-aware Importance (IRI), which are elaborated in Sec. 3.2 and Sec. 3.3, respectively.

3.2 Feature-level Object-sensitive Importance

The basic idea of FOI is to leverage the guidance of the pre-trained model denoted by Φ_{pre} and the object-sensitive information denoted as Φ_{obj}, to avoid over-pruning and boost channel pruning for object detection.

Specifically, we consider the features in certain intermediate layers S (usually the neck layers) from both the pre-trained model and the currently pruned model, denoted by $\{\mathbf{F}_{pre}^{(s)} : s \in S\}$ and $\{\mathbf{F}_{cur}^{(s)} : s \in S\}$, respectively. The guidance $\Phi_{pre}^{(s)} \in \mathbb{R}^{B\times H^{(s)}\times W^{(s)}}$ is formulated as their cosine similarity at each spatial position:

$$\left(\Phi_{pre}^{(s)}\right)_{b,h,w} = \frac{(\mathbf{F}_{pre}^{(s)})_{b,:,h,w}}{\parallel (\mathbf{F}_{pre}^{(s)})_{b,:,h,w} \parallel} \cdot \frac{f(\mathbf{F}_{cur}^{(s)})_{b,:,h,w}}{\parallel f(\mathbf{F}_{cur}^{(s)})_{b,:,h,w} \parallel}, \tag{4}$$

where f is a convolutional layer with stride 1 and 1×1 kernel size, aiming to transform features of the pruned model into the feature space of the pre-trained ones.

In order to pay more attention to object areas, inspired by [15, 80], we use the classification outputs of the pre-trained model and current pruned model, denoted as $\mathbf{O}_{pre}^{cls}, \mathbf{O}_{cur}^{cls} \in \mathbb{R}^{B\times C\times H\times W}$ respectively, to approximate object masks. The object-sensitive guidance $\Phi_{obj} \in \mathbb{R}^{B\times H\times W}$ is calculated as:

$$\Phi_{obj} = \max\{\hat{\mathbf{O}}_{pre}^{cls}, \hat{\mathbf{O}}_{cur}^{cls}\},\ \hat{\mathbf{O}}^{cls} = \max\{\sigma(\mathbf{O}^{cls})_{:,c,:,:}\}, \tag{5}$$

where σ is the activation function (*e.g.* sigmoid or softmax). Based on Φ_{pre} and Φ_{obj}, the pre-trained model guided object-sensitive information loss is formulated as below:

$$\mathcal{L}_{info} = \sum_{s\in S}\left(1 - |\Phi_{obj} \odot \Phi_{pre}^{(s)}|/|\Phi_{obj}|\right). \tag{6}$$

Similar to I_{TLI} as depicted in Eq. (1) and Eq. (2), we can calculate the FOI importance I_{FOI} via backpropagation of $\mathcal{L}_{info}$.

Despite that the pre-trained model guided object-sensitive information loss $\mathcal{L}_{info}$ shares similar spirit with those of several feature distillation approaches [15, 59, 80], they are distinct in the following two aspects: i) $\mathcal{L}_{info}$ is applied in the pruning stage to facilitate preserving channels that are important for object detection, rather than in the fine-tuning stage for network optimization; ii) $\mathcal{L}_{info}$ utilizes the cosine similarity rather than MSE [27] that is commonly used in feature distillation, and their comparison is summarized in Tab. 6.

3.3 Intra-layer Redundancy-aware Importance

As displayed in Fig. 2, IRI mainly consists of three steps, including weight aggregation, channel clustering, and numeralization.

In the weight aggregation step, inspired by CUP [8], each channel aggregates the weights of the corresponding source in preceding layers and those of the target in successive layers. The aggregated weight $\hat{\mathbf{W}}^{(l)} \in \mathbb{R}^{C^{(l)}\times(C^{(l-1)}+C^{(l+1)})}$ of the i-th channel in layer l has the following form:

$$\begin{aligned} \hat{\mathbf{W}}_i^{(l)} &= concat(h(\mathbf{W}_{i,:,:,:}^{(l-1)}), h(\mathbf{W}_{:,i,:,:}^{(l)})), \\ h(\mathbf{W}_{:,:,:}) &= concat(\parallel \mathbf{W}_{0,:,:} \parallel_F, \cdots, \parallel \mathbf{W}_{C,:,:} \parallel_F), \end{aligned} \tag{7}$$

where $concat(\cdot)$ is the concatenation operation, and $\parallel \cdot \parallel_F$ represents the Frobenius norm. As in Eq. (7), a channel aggregates the weights corresponding to all of its coupled channels.

In the channel clustering step, the agglomerative hierarchical clustering algorithm [64] with thresholding is adopted to cluster the aggregation weights $\{\hat{\mathbf{W}}^{(l)}\}$. Considering that there are many coupled channels and the size of aggregated weights varies significantly, we control the threshold by setting a ratio rather than using the distance between clusters. Specifically, we fix the ratio as 0.1 to terminate the clustering process when 10% of channels are clustered with the other ones. The clustering result of layer l is written as $S_0^{(l)}, S_1^{(l)}, \cdots, S_{g-1}^{(l)}$, where $|S_0^{(l)}| + \cdots + |S_{g-1}^{(l)}| = |\mathbf{m}^{(l)}|$.

In the numeralization step, we assume that the channels in the minority clusters have a lower pruning priority than those in the majority clusters. To that end, we formulate the importance of the i-th channel in layer l belonging to the j_{th} cluster as the following:

$$\tilde{I}_i^{(l)} = \frac{\sum_{c=1}^{C} \mathbf{m}_c^{(l)}}{\sum_{k\in S_j^{(l)})} \mathbf{m}_k^{(l)}}, i \in S_j^{(l)}. \tag{8}$$

The above importance is further re-scaled by the following formula:

$$I_{IRI,i}^{(l)} = \exp\left(\lambda_{scale} \times \left(\frac{\tilde{I}^{(l)} - \min(\tilde{I}^{(l)})}{\max(\tilde{I}^{(l)}) - \min(\tilde{I}^{(l)})}\right)\right), \tag{9}$$

where λ_{scale} is the scaling factor.

Before pruning, all layers will be clustered to generate the initial importance $I_{IRI,i}$. During the iterative pruning process, if the channel of a certain layer is trimed, *i.e.* the mask value $\mathbf{m}_i^{(l)}$ is set to 0, we separately update I_{IRI} of this specific layer.

Compared to existing redundancy-based approaches [8, 19, 34], IRI has the following advantages: i) IRI reuses the channel importance estimated in previous iterations, thus accomplishing efficient iterative pruning; ii) IRI clusters the channels based on coupled structures and controls the number of clusters in a proportional way, thus being generalizable to different detector frameworks.

Finally, the overall multi-granular channel importance $I_{MIEP,i}^{(l)}$ of the i-th channel in layer l is written as below:

$$I_{MIEP,i}^{(l)} = \left(I_{TLI,i}^{(l)} + I_{FOI,i}^{(l)}\right) \times I_{IRI,i}^{(l)}. \tag{10}$$

4 EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we evaluate the effectiveness of MIEP by comparing to state-of-the-art model pruning methods and lightweight object detectors as well as extensive ablation studies. We also display the generalizability of MIEP to distinct detectors and tasks.

4.1 Dataset and Evaluation Metric

We conduct experiments on two widely used benchmarks for object detection, *i.e.* MS COCO 2017 [38] and Pascal VOC [9]. COCO contains 118k training images, 5,000 validation images, and 41k test images. Following previous works [33, 71], we evaluate on the validation set. Pascal VOC includes 5,011 trainval images for VOC 2007, 11,540 trainval images for VOC 2012, and 4,952 test images. We follow the official split setting to train models on both trainval sets, and evaluate on the PASCAL VOC 2007 test set.

We report the mean Average Precision (mAP), AP_{50}, and AP_{75} as the evaluation metrics on accuracy, and GFLOPs on efficiency.

4.2 Implementation Details

By following [5, 6, 39], the hyper-parameters are set by default for fair comparisons. Generally, we utilize the ResNet50 as the backbone. In the channel pruning stage, we follow [39] to update the weights by using a fixed learning rate that is $\frac{1}{5}$ of the initial one used in training the full model. In the fine-tuning stage after

Table 1: Comparison of mAP, AP_{50}, AP_{75} and FLOPs on COCO by various model pruning methods based on two detectors with the ResNet50 backbone. "Full" and "Pruning" refer to unpruned full and pruned models, respectively. "Δ ↓" is the performance drop between the pruned and full models, the smaller, the better. "FLOPs (%)" is the remaining FLOPs after pruning.

Detector	Method	mAP (%)			AP_{50} (%)			AP_{75} (%)			GFLOPs	FLOPs (%)
		Full	Pruning	Δ ↓	Full	Pruning	Δ ↓	Full	Pruning	Δ ↓		
GFL	Slim [73]	38.5	37.2	1.3	55.7	54.3	1.4	41.6	40.1	1.5	119.2	57%
	GroupFisher [39]	39.1	38.9	0.2	56.8	56.1	0.7	42.1	41.9	0.2	104.2	50%
	DepGraph [11]	39.1	38.1	1.0	56.8	55.1	1.7	42.1	41.2	0.9	107.6	52%
	MIEP (Ours)	**39.1**	**39.0**	**0.1**	**56.8**	**56.5**	**0.3**	**42.1**	**42.3**	**-0.2**	104.2	50%
	Slim [73]	38.5	34.2	4.3	55.7	50.4	5.3	41.6	36.8	4.8	54.7	26%
	GroupFisher [39]	39.1	36.3	2.8	56.8	53.0	3.8	42.1	39.1	3.0	52.1	25%
	DepGraph [11]	39.1	33.3	5.8	56.8	48.8	8.0	42.1	35.8	6.3	53.8	26%
	MIEP (Ours)	**39.1**	**37.0**	**2.1**	**56.8**	**54.1**	**2.7**	**42.1**	**39.9**	**2.2**	52.1	25%
	Slim [73]	38.5	27.7	10.8	55.7	41.7	14.0	41.6	29.5	12.1	22.5	11%
	GroupFisher [39]	39.1	30.2	8.9	56.8	45.0	11.8	42.1	32.3	9.8	20.8	10%
	DepGraph [11]	39.1	16.2	22.9	56.8	25.3	31.5	42.1	17.0	25.1	22.7	11%
	MIEP (Ours)	**39.1**	**32.2**	**6.9**	**56.8**	**48.2**	**8.6**	**42.1**	**34.7**	**7.4**	20.8	10%
SSD	Slim [73]	23.8	22.5	1.3	41.9	39.9	2.0	24.3	22.6	1.7	12.5	62%
	FPGM [19]	25.5	24.3	1.2	44.0	42.2	1.8	26.4	24.8	1.6	10.4	52%
	LCP [66]	21.9	20.9	1.0	37.3	35.5	1.8	22.7	21.6	1.1	10.7	53%
	DMCP [16]	25.2	24.1	1.1	42.7	41.2	1.5	25.8	24.7	1.1	13.4	66%
	GroupFisher [39]	25.5	24.8	0.7	44.0	42.7	1.3	26.4	25.3	1.1	10.1	50%
	CHEX [21]	25.2	24.3	0.9	42.7	41.0	1.7	25.8	24.9	0.9	10.3	51%
	DepGraph [10]	25.5	24.3	1.2	44.0	42.1	1.9	26.4	24.6	1.8	10.5	52%
	MIEP (Ours)	**25.5**	**24.9**	**0.6**	**44.0**	**42.9**	**1.1**	**26.4**	**25.5**	**0.9**	10.1	50%
	Slim [73]	23.8	19.2	4.6	41.9	35.0	6.9	24.3	18.9	5.4	6.5	32%
	FPGM [19]	25.5	20.0	5.5	44.0	35.9	8.1	26.4	20.1	6.3	5.2	26%
	GroupFisher [39]	25.5	20.6	4.9	44.0	36.7	7.3	26.4	20.7	5.7	5.1	25%
	DepGraph [10]	25.5	19.8	5.7	44.0	35.3	8.7	26.4	19.9	6.5	5.3	26%
	MIEP (Ours)	**25.5**	**21.0**	**4.5**	**44.0**	**37.3**	**6.7**	**26.4**	**21.1**	**5.3**	5.1	25%
	Slim [73]	23.8	12.7	11.1	41.9	24.9	17.0	24.3	11.9	12.4	2.4	12%
	FPGM [19]	25.5	13.8	11.7	44.0	26.1	17.9	26.4	13.1	13.3	2.4	12%
	GroupFisher [39]	25.5	14.3	11.2	44.0	27.1	16.9	26.4	13.7	12.7	2.0	10%
	DepGraph [10]	25.5	12.7	12.8	44.0	24.4	19.6	26.4	12.0	14.4	2.1	11%
	MIEP (Ours)	**25.5**	**15.0**	**10.5**	**44.0**	**28.3**	**15.7**	**26.4**	**14.3**	**12.1**	2.0	10%

pruning, we use the same configurations as used in training the full model. The scaling factor λ_{scale} is fixed as 2. All the experiments are conducted on 4 NVIDIA 3080Ti GPUs.

4.3 Comparison to the State-of-the-art Methods

We compare with the state-of-the-art approaches for model pruning, including Slim [73], GroupFisher [39], FPGM[19], DepGraph [11], LCP [66], DMCP [83] and CHEX [21], by using the GFL [33] and SSD [40] detectors with ResNet50. As summarized in Tab. 1 and Tab. 2, our method clearly achieves higher accuracy than the compared methods on COCO and VOC, when compressing the full models to smaller ones with similar GFLOPs. Concretely, MIEP promotes the mAP of the second best approach by 2.0% and 0.7% on COCO for GFL and SSD, respectively. The reason lies in that MIEP specifically addresses the over-pruning issue, which tends to become more severe as more channels are trimmed.

We further compare the lightweight object detectors including PP-YOLO [42], MobileDets [67], YOLOv5 [24], YOLOv7 [57], NanoDet [49], NanoDet-Plus [49] and PP-PicoDet [71]. We adopt PP-PicoDet as the full model. As shown in Tab. 3 and Fig. 4, pruning PP-PicoDet by MIEP reaches higher accuracy than the compared lightweight detectors whilst using fewer parameters and fewer GFLOPs. The results imply that our method can effectively compress trained object detectors to a lightweight one, and boost the accuracy and efficiency trade-off.

4.4 Ablation Study

We first evaluate the effect of the main components, *i.e.* FOI and IRI, on the proposed method. As shown in Tab. 4, when using 10% FLOPs, FOI and IRI improve the mAP by 1.2% and 1.3%, respectively, and their combination further boosts the accuracy. Similar results are achieved when using 25% FLOPs.

Table 2: Comparison of mAP and FLOPs on Pascal VOC by using various model pruning methods.

Detector	Method	mAP (%)		FLOPs (%)
		Full	Pruning	
Faster-RCNN [50]	Slim [73]	81.0	71.5	16%
	GroupFisher [39]	80.4	75.1	10%
	MIEP (Ours)	80.4	**76.4**	10%
RetinaNet [37]	Slim [73]	79.8	73.3	11%
	GroupFisher [39]	77.3	72.5	10%
	MIEP (Ours)	77.3	**74.1**	10%

Table 3: Comparison of mAP, parameters, and GFLOPs on COCO by comparing to the lightweight detectors.

Method	Input size	Param. (M)	GFLOPs	mAP (%)
PP-YOLO-Tiny [42]	320	1.08	0.58	20.6
PP-YOLO-Tiny [42]	416	1.08	1.02	22.7
MobileDets-CPU [67]	320	3.85	1.02	24.2
YOLOv5-n [24]	640	1.87	4.5	28.4
YOLOX-Nano [13]	416	0.91	1.08	25.8
YOLOv7-Tiny [57]	320	6.23	3.46	29.1
YOLOv7-Tiny [57]	416	6.23	5.82	35.2
NanoDet-m [49]	320	0.95	0.72	20.6
NanoDet-m-1.5x [49]	320	2.08	1.44	23.5
NanoDet-m-1.5x [49]	416	2.08	2.42	26.8
NanoDet-Plus-m [49]	320	1.17	0.9	27.0
NanoDet-Plus-m-1.5x [49]	320	2.44	1.75	29.9
NanoDet-Plus-m-1.5x [49]	416	2.44	2.97	34.1
PP-PicoDet-S [71]	320	0.99	0.73	27.1
PP-PicoDet-M [71]	320	2.15	1.48	30.9
PP-PicoDet-L [71]	320	3.30	2.23	32.9
PP-PicoDet-L [71]	416	3.30	3.76	36.6
Ours (PicoDet-M-25%)	416	0.97	0.63	28.5 (+1.4)
Ours (PicoDet-L-39%)	416	2.11	1.47	32.4 (+1.5)
Ours (PicoDet-L-66%)	416	2.70	2.22	34.9 (+2.0)
Ours (PicoDet-L-42%)	640	2.35	3.74	37.6 (+1.0)
Ours (PicoDet-L-65%)	640	2.93	5.79	39.8 (+4.6)

Table 4: Effect of FOI and IRI on COCO with GFL.

FOI Module	IRI Module	mAP (%)	FLOPs (%)
		36.3	25%
✓		36.6	
	✓	36.6	
✓	✓	37.0	
		30.2	10%
✓		31.4	
	✓	31.5	
✓	✓	32.2	

On FOI. We evaluate the effect of the pre-trained feature guidance Φ_{pre} and the object-sensitive information guidance Φ_{obj}. As shown in Tab. 5, both Φ_{pre} and Φ_{obj} promote the accuracy. As discussed in Sec. 3.2, we investigate the influence of loss usage (*i.e.* $\mathcal{L}_{info}$) in different stages on FOI. As displayed in Tab. 6, when applying $\mathcal{L}_{info}$ for pruning already achieves 0.6% improvement on mAP, and further boosts the accuracy when jointly applying it for parameter optimization. We also compare the cosine loss to MSE that is used by existing feature distillation approaches. The results imply that the cosine loss empirically outperforms MSE, as the latter aims to minimize pixel-wise errors, imposing excessively strong

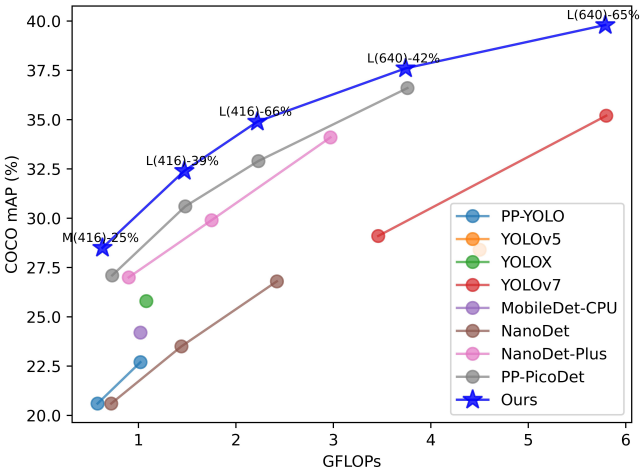


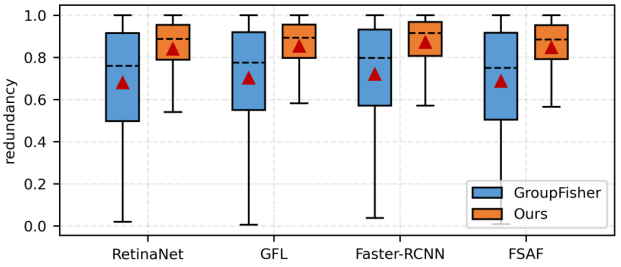
Figure 4: Comparison of mAP and GFLOPs with different lightweight detectors.

Table 5: Ablation study on FOI by GFL on COCO with 10% FLOPs.

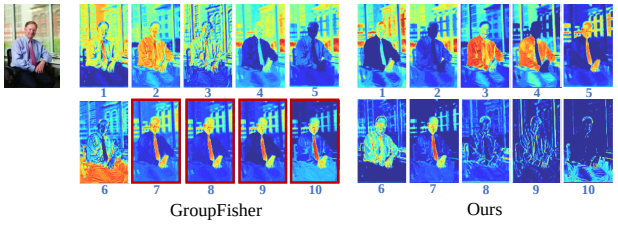
Φ_{pre} in Eq. (4)	Φ_{obj} in Eq. (5)	mAP (%)
		30.2
✓		31.1
✓	✓	31.4

Table 6: Effect of different losses and model on FOI, and distinct clustering methods on IRI by GFL with 10% FLOPs.

Module	Setting	Method	mAP (%)
FOI	baseline	GroupFisher	30.2
	loss usage	only for pruning	30.8
		pruning & optimization	31.4
	loss form	MSE (feature distillation)	30.8
		cosine loss	31.4
	pre-trained model	ResNet50-2x	31.7
		ResNet101	31.9
IRI	baseline	GroupFisher w/ FOI	31.4
	clustering algo.	k-means	31.9
		hierarchical	32.2
	thresholding	distance=0.8	31.7
		ratio=0.1	32.2



(a) Redundancy of removed channels



(b) Visualization of remaining channels

Figure 5: Visualization on redundancy removal by comparing GropuFisher [39] with our method. (a) Redundancy of pruned channels by various detectors. (b) Feature maps in the *backbone.layer*.2.1.*conv*2 layer, where the redundant ones are highlighted by red boxes.

constraints. We also study the influence of different pre-trained models on FOI, including ResNet50-2x trained with double epochs and ResNet101 with deeper structures. The results indicate that better pre-trained models promote the performance of FOI.

On IRI. As shown in Tab. 6, we evaluate the performance of IRI by using different clustering algorithms including k-means and hierarchical clustering, indicating that the latter yields higher accuracy. In regards of the thresholding strategy, we also compare the distance-based and the ratio-based ones, clearly showing the advantage of the ratio-based thresholding.

We also qualitatively verify the effect of IRI on removing channel redundancy. As shown in Fig. 5(a), the redundancy (see definition in Eq. (3)) of pruned channels remarkably increases, implying that fewer redundant channels are reserved when using IRI. Moreover, inspired by [19], we visualize the intermediate features generated by untrimmed channels, where the regions colored red/blue represent the most/least salient areas. As displayed in Fig. 5(b), the feature maps (*e.g.* those indexed by 7, 8, 9, 10) generated by GroupFisher exhibit a similar appearance, thus being redundant. In contrast, our method generates more diverse feature maps.

4.5 Generalizability of the Proposed Approach

On different detectors. We choose the representative two-stage Faster-RCNN [50], the anchor-free FSAF [82], and RetinaNet [37] as base detectors. As shown in Table 7, our method consistently boosts the performance for different detection frameworks, demonstrating the generalizability of MIEP to distinct detectors.

On the classification task. Besides object detection, we also evaluate MIEP on the classification task on ImageNet-1K [7]. The compared approaches include ThiNet [44], MetaPruning [41], AutoSlim [72], GReg [58], HALP [52], SMCP [23], and GroupFisher.

Table 7: Comparison of mAP and FLOPs on COCO with different detectors by using various pruning methods.

Detector	Method	mAP (%)		FLOPs (%)
		Full	Pruning	
Faster-RCNN [50]	Slim [73]	37.5	27.2	15%
	GroupFisher [39]	37.4	32.3	10%
	MIEP (Ours)	37.4	**33.0 (+0.7)**	10%
FASF [82]	Slim [73]	36.8	26.4	10%
	GroupFisher [39]	37.4	28.9	10%
	MIEP (Ours)	37.4	**29.5 (+0.6)**	10%
RetinaNet [37]	Slim [73]	36.7	26.4	10%
	GroupFisher [39]	36.5	27.5	10%
	MIEP (Ours)	36.5	**29.1 (+1.6)**	10%

Table 8: Comparison of the Top1 and Top5 accuracy on ImageNet-1K by using various model pruning methods.

Method	Top1 *Acc.* (%)		Top5 *Acc.* (%)		FLOPs (%)
	Full	Pruning	Full	Pruning	
0.5*ResNet50 [23]	76.15	72.00	92.87	-	26%
ThiNet [44]	76.15	72.10	92.87	88.30	30%
MetaPruning [41]	76.15	73.40	92.87	-	26%
AutoSlim [72]	76.15	74.00	92.87	-	25%
GroupFisher [39]	76.15	74.15	92.87	91.70	25%
GReg-2 [58]	76.15	73.90	92.87	-	33%
HALP [52]	76.15	74.41	92.87	91.85	27%
SMCP [23]	76.15	74.60	92.87	92.00	27%
MIEP (Ours)	76.15	**74.73**	92.87	**92.17**	25%
GroupFisher [39]	76.15	69.68	92.87	89.11	10%
SMCP [23]	76.15	69.86	92.87	89.00	12%
MIEP (Ours)	76.15	**70.32**	92.87	**89.81**	10%

The results in Tab. 8 reveals that our method clearly promotes the accuracy with 25% and 10% FLOPs, indicating the generalizability of our method to different tasks.

5 CONCLUSION

In this paper, we propose a novel channel pruning approach with multi-granular importance estimation, namely MIEP, for efficient object detection. We develop the Feature-level Object-sensitive Importance and the Intra-layer Redundancy-aware Importance to address the over-pruning and intra-layer redundancy issues arised in existing pruning approaches, respectively. We extensively evaluate our method on object detection and additional image classification task by comparing to the state-of-the-art approaches, clearly displaying the effectiveness of the proposed method.

ACKNOWLEDGMENTS

This work is supported by the National Key R&D Program of China (2021ZD0110503), the National Natural Science Foundation of China (62202034 and 62022011), the Research Program of State Key Laboratory of Virtual Reality Technology and Systems, and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6077–6086.

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020).

[3] Francesco Bongini, Lorenzo Berlincioni, Marco Bertini, and Alberto Del Bimbo. 2021. Partially fake it till you make it: mixing real and fake thermal images for improved object detection. In *Proceedings of the 29th ACM International Conference on Multimedia*. 5482–5490.

[4] Yuxuan Cai, Hongjia Li, Geng Yuan, Wei Niu, Yanyu Li, Xulong Tang, Bin Ren, and Yanzhi Wang. 2021. Yolobile: Real-time object detection on mobile devices via compression-compilation co-design. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 955–963.

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. 2019. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155* (2019).

[6] MMPreTrain Contributors. 2023. OpenMMLab's Pre-training Toolbox and Benchmark. https://github.com/open-mmlab/mmpretrain.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 248–255.

[8] Rahul Duggal, Cao Xiao, Richard Vuduc, Duen Horng Chau, and Jimeng Sun. 2021. CUP: Cluster Pruning for Compressing Deep Neural Networks. *IEEE International Conference on Big Data* (Dec. 2021), 5102–5106. https://doi.org/10.1109/BigData52589.2021.9671980

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision* 88 (2010), 303–338.

[10] Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. 2023. DepGraph: Towards Any Structural Pruning. *arXiv preprint arXiv:2301.12900* (2023).

[11] Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. 2023. DepGraph: Towards Any Structural Pruning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[12] Yan Gao, Qimeng Wang, Xu Tang, Haochen Wang, Fei Ding, Jing Li, and Yao Hu. 2021. Decoupled iou regression for object detection. In *Proceedings of the 29th ACM International Conference on Multimedia*. 5628–5636.

[13] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. 2021. YOLOX: Exceeding YOLO Series in 2021. *arXiv preprint arXiv:2107.08430* (2021).

[14] Ross Girshick. 2015. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, Santiago, Chile, 1440–1448. https://doi.org/10.1109/ICCV.2015.169

[15] Jianyuan Guo, Kai Han, Yunhe Wang, Han Wu, Xinghao Chen, Chunjing Xu, and Chang Xu. 2021. Distilling object detectors via decoupled features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2154–2164.

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

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*. 2961–2969.

[18] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. 2018. Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, Stockholm, Sweden, 2234–2240. https://doi.org/10.24963/ijcai.2018/309

[19] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. 2019. Filter Pruning via Geometric Median for Deep Convolutional Neural Networks Acceleration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4340–4349.

[20] Yihui He, Xiangyu Zhang, and Jian Sun. 2017. Channel Pruning for Accelerating Very Deep Neural Networks. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, Venice, 1398–1406. https://doi.org/10.1109/ICCV.2017.155

[21] Zejiang Hou, Minghai Qin, Fei Sun, Xiaolong Ma, Kun Yuan, Yi Xu, Yen-Kuang Chen, Rong Jin, Yuan Xie, and Sun-Yuan Kung. 2022. Chex: channel exploration for CNN model compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 12287–12298.

[22] Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. 2016. Network Trimming: A Data-Driven Neuron Pruning Approach towards Efficient Deep Architectures. https://doi.org/10.48550/arXiv.1607.03250 arXiv:arXiv:1607.03250

[23] Ryan Humble, Maying Shen, Jorge Albericio Latorre, Eric Darve, and Jose Alvarez. 2022. Soft Masking for Cost-Constrained Channel Pruning. In *Proceedings of the European Conference on Computer Vision*. Springer, 641–657.

[24] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, Zeng Yifu, Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. 2022. *ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation*. https://doi.org/10.5281/zenodo.7347926

[25] Woojeong Kim, Suhyun Kim, Mincheol Park, and Geunseok Jeon. 2020. Neuron Merging: Compensating for Pruned Neurons. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 585–595.

[26] Yann LeCun, John Denker, and Sara Solla. 1989. Optimal Brain Damage. In *Advances in Neural Information Processing Systems*, Vol. 2. Morgan-Kaufmann.

[27] Erich L Lehmann and George Casella. 2006. *Theory of point estimation*. Springer Science & Business Media.

[28] Baopu Li, Yanwen Fan, Zhihong Pan, Yuchen Bian, and Gang Zhang. 2021. Automatic Channel Pruning with Hyper-parameter Search and Dynamic Masking. In *Proceedings of the 29th ACM International Conference on Multimedia*. 2121–2129.

[29] Chuyi Li, Lulu Li, Yifei Geng, Hongliang Jiang, Meng Cheng, Bo Zhang, Zaidan Ke, Xiaoming Xu, and Xiangxiang Chu. 2023. YOLOv6 v3. 0: A Full-Scale Reloading. *arXiv preprint arXiv:2301.05586* (2023).

[30] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. 2017. Pruning Filters for Efficient ConvNets. https://doi.org/10.48550/arXiv.1608.08710 arXiv:arXiv:1608.08710

[31] Siyuan Li, Zedong Wang, Zicheng Liu, Cheng Tan, Haitao Lin, Di Wu, Zhiyuan Chen, Jiangbin Zheng, and Stan Z Li. 2022. Efficient multi-order gated aggregation network. *arXiv preprint arXiv:2211.03295* (2022).

[32] Xiang Li, Wenhai Wang, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. 2021. Generalized Focal Loss V2: Learning Reliable Localization Quality Estimation for Dense Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11632–11641.

[33] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. 2020. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 21002–21012.

[34] Mingbao Lin, Liujuan Cao, Yuxin Zhang, Ling Shao, Chia-Wen Lin, and Rongrong Ji. 2022. Pruning Networks With Cross-Layer Ranking & k-Reciprocal Nearest Filters. *IEEE Transactions on Neural Networks and Learning Systems* (2022), 1–10. https://doi.org/10.1109/TNNLS.2022.3156047

[35] Mingbao Lin, Yuxin Zhang, Yuchao Li, Bohong Chen, Fei Chao, Mengdi Wang, Shen Li, Yonghong Tian, and Rongrong Ji. 2022. 1xn pattern for pruning convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[36] Shaohui Lin, Rongrong Ji, Chenqian Yan, Baochang Zhang, Liujuan Cao, Qixiang Ye, Feiyue Huang, and David Doermann. 2019. Towards optimal structured cnn pruning via generative adversarial learning. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. 2790–2799.

[37] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 2980–2988.

[38] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*. Springer, 740–755.

[39] Liyang Liu, Shilong Zhang, Zhanghui Kuang, Aojun Zhou, Jing-Hao Xue, Xinjiang Wang, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. 2021. Group Fisher Pruning for Practical Network Compression. In *Proceedings of the International Conference on Machine Learning*. ICML, 7021–7032.

[40] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*. Springer, 21–37.

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

[42] Xiang Long, Kaipeng Deng, Guanzhong Wang, Yang Zhang, Qingqing Dang, Yuan Gao, Hui Shen, Jianguo Ren, Shumin Han, Errui Ding, et al. 2020. PP-YOLO: An effective and efficient implementation of object detector. *arXiv preprint arXiv:2007.12099* (2020).

[43] Xiaotong Lu, Teng Xi, Baopu Li, Gang Zhang, Weisheng Dong, and Guangming Shi. 2022. Bayesian based Re-parameterization for DNN Model Pruning. In *Proceedings of the 30th ACM International Conference on Multimedia*. 1367–1375.

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

[45] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. 2018. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision*. 116–131.

[46] Zeyu Ma, Yang Yang, Guoqing Wang, Xing Xu, Heng Tao Shen, and Mingxing Zhang. 2022. Rethinking Open-World Object Detection in Autonomous Driving Scenarios. In *Proceedings of the 30th ACM International Conference on Multimedia*. 1279–1288.

[47] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. 2019. Importance Estimation for Neural Network Pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Long Beach, CA, USA, 11256–11264. https://doi.org/10.1109/CVPR.2019.01152

[48] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2016. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440* (2016).

[49] RangiLyu. 2021. NanoDet-Plus: Super fast and high accuracy lightweight anchor-free object detection model. https://github.com/RangiLyu/nanodet.

[50] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 6 (June 2017), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

[51] Victor Sanh, Thomas Wolf, and Alexander Rush. 2020. Movement Pruning: Adaptive Sparsity by Fine-Tuning. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 20378–20389.

[52] Maying Shen, Hongxu Yin, Pavlo Molchanov, Lei Mao, Jianna Liu, and Jose Alvarez. 2022. Structural Pruning via Latency-Saliency Knapsack. In *Advances in Neural Information Processing Systems*.

[53] Meng Sun, Ju Ren, Xin Wang, Wenwu Zhu, and Yaoxue Zhang. 2022. FastPR: One-stage Semantic Person Retrieval via Self-supervised Learning. In *Proceedings of the 30th ACM International Conference on Multimedia*. 3628–3636.

[54] Yehui Tang, Yunhe Wang, Yixing Xu, Dacheng Tao, Chunjing XU, Chao Xu, and Chang Xu. 2020. SCOP: Scientific Control for Reliable Neural Network Pruning. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 10936–10947.

[55] Lucas Theis, Iryna Korshunova, Alykhan Tejani, and Ferenc Huszár. 2018. Faster Gaze Prediction with Dense Networks and Fisher Pruning. arXiv:arXiv:1801.05787

[56] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. 2019. FCOS: Fully Convolutional One-Stage Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 9627–9636.

[57] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696* (2022).

[58] Huan Wang, Can Qin, Yulun Zhang, and Yun Fu. 2020. Neural Pruning via Growing Regularization. In *International Conference on Learning Representations*.

[59] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. 2019. Distilling object detectors with fine-grained feature imitation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4933–4942.

[60] Wenhai Wang, Enze Xie, Xiang Li, Xuebo Liu, Ding Liang, Zhibo Yang, Tong Lu, and Chunhua Shen. 2021. Pan++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 9 (2021), 5349–5367.

[61] Zijie Wang, Aichun Zhu, Jingyi Xue, Xili Wan, Chao Liu, Tian Wang, and Yifeng Li. 2022. CAIBC: Capturing All-round Information Beyond Color for Text-based Person Retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*. 5314–5322.

[62] Dulanga Weerakoon, Vigneshwaran Subbaraju, Tuan Tran, and Archan Misra. 2022. SoftSkip: Empowering Multi-Modal Dynamic Pruning for Single-Stage Referring Comprehension. In *Proceedings of the 30th ACM International Conference on Multimedia*. 3608–3616.

[63] Jin Wei, Yuan Zhang, Yu Zhou, Gangyan Zeng, Zhi Qiao, Youhui Guo, Haiying Wu, Hongbin Wang, and Weipinng Wang. 2022. Textblock: Towards scene text spotting without fine-grained detection. In *Proceedings of the 30th ACM International Conference on Multimedia*. 5892–5902.

[64] Daniel S Wilks. 2011. *Statistical methods in the atmospheric sciences*. Vol. 100. Academic press.

[65] Jialian Wu, Liangchen Song, Tiancai Wang, Qian Zhang, and Junsong Yuan. 2020. Forest r-cnn: Large-vocabulary long-tailed object detection and instance segmentation. In *Proceedings of the 28th ACM International Conference on Multimedia*. 1570–1578.

[66] Zihao Xie, Li Zhu, Lin Zhao, Bo Tao, Liman Liu, and Wenbing Tao. 2020. Localization-Aware Channel Pruning for Object Detection. *Neurocomputing* 403 (Aug. 2020), 400–408. https://doi.org/10.1016/j.neucom.2020.03.056

[67] Yunyang Xiong, Hanxiao Liu, Suyog Gupta, Berkin Akin, Gabriel Bender, Yongzhe Wang, Pieter-Jan Kindermans, Mingxing Tan, Vikas Singh, and Bo Chen. 2021. Mobiledets: Searching for object detection architectures for mobile accelerators. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3825–3834.

[68] Tianshuo Xu, Yuhang Wu, Xiawu Zheng, Teng Xi, Gang Zhang, Errui Ding, Fei Chao, and Rongrong Ji. 2021. CDP: Towards Optimal Filter Pruning via Class-wise Discriminative Power. In *Proceedings of the 29th ACM International Conference on Multimedia*. 5491–5500.

[69] Jianing Yang, Yongxin Wang, Ruitao Yi, Yuying Zhu, Azaan Rehman, Amir Zadeh, Soujanya Poria, and Louis-Philippe Morency. 2020. Mtag: Modal-temporal attention graph for unaligned human multimodal language sequences. *arXiv preprint arXiv:2010.11985* (2020).

[70] Zhonghui You, Kun Yan, Jinmian Ye, Meng Ma, and Ping Wang. 2019. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. *Advances in Neural Information Processing Systems* 32 (2019).

[71] Guanghua Yu, Qinyao Chang, Wenyu Lv, Chang Xu, Cheng Cui, Wei Ji, Qingqing Dang, Kaipeng Deng, Guanzhong Wang, Yuning Du, Baohua Lai, Qiwen Liu, Xiaoguang Hu, Dianhai Yu, and Yanjun Ma. 2021. PP-PicoDet: A Better Real-Time Object Detector on Mobile Devices. arXiv:arXiv:2111.00902

[72] Jiahui Yu and Thomas Huang. 2019. AutoSlim: Towards One-Shot Architecture Search for Channel Numbers. arXiv:arXiv:1903.11728

[73] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. 2019. Slimmable neural networks. In *International Conference on Learning Representations*.

[74] Zheng-Jun Zha, Daqing Liu, Hanwang Zhang, Yongdong Zhang, and Feng Wu. 2019. Context-aware visual policy network for fine-grained image captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 2 (2019), 710–722.

[75] Haonan Zhang, Longjun Liu, Hengyi Zhou, Wenxuan Hou, Hongbin Sun, and Nanning Zheng. 2021. Akecp: Adaptive knowledge extraction from feature maps for fast and efficient channel pruning. In *Proceedings of the 29th ACM International Conference on Multimedia*. 648–657.

[76] Pengyi Zhang, Yunxin Zhong, and Xiaoqiong Li. 2019. SlimYOLOv3: Narrower, Faster and Better for Real-Time UAV Applications. In *Proceedings of the IEEE International Conference on Computer Vision Workshop*. IEEE, Seoul, Korea (South), 37–45. https://doi.org/10.1109/ICCVW.2019.00011

[77] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. 2020. Bridging the Gap Between Anchor-Based and Anchor-Free Detection via Adaptive Training Sample Selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9759–9768.

[78] Yulun Zhang, Huan Wang, Can Qin, and Yun Fu. 2021. Aligned structured sparsity learning for efficient image super-resolution. *Advances in Neural Information Processing Systems* 34 (2021), 2695–2706.

[79] Chenglong Zhao, Bingbing Ni, Jian Zhang, Qiwei Zhao, Wenjun Zhang, and Qi Tian. 2019. Variational Convolutional Neural Network Pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Long Beach, CA, USA, 2775–2784. https://doi.org/10.1109/CVPR.2019.00289

[80] Du Zhixing, Rui Zhang, Ming Chang, xishan zhang, Shaoli Liu, Tianshi Chen, and Yunji Chen. 2021. Distilling Object Detectors with Feature Richness. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., 5213–5224.

[81] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. 2019. Objects as points. *arXiv preprint arXiv:1904.07850* (2019).

[82] Chenchen Zhu, Yihui He, and Marios Savvides. 2019. Feature selective anchor-free module for single-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 840–849.

[83] Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jinhui Zhu. 2018. Discrimination-aware channel pruning for deep neural networks. *Advances in Neural Information Processing Systems* 31 (2018).

[84] Zhuofan Zong, Qianggang Cao, and Biao Leng. 2021. RCNet: Reverse feature pyramid and cross-scale shift network for object detection. In *Proceedings of the 29th ACM International Conference on Multimedia*. 5637–5645.