基于滤波器参数重构的稀疏剪枝方法

杨火祥[1]，仪双燕[2]，孟凡阳[3]，柳伟[2]，李宗鹏[4]，梁永生[1,5]
（1.深圳大学电子与信息工程学院，广东 深圳 518060；2.深圳信息职业技术大学计算机与软件学院，广东 深圳 518172；
3.鹏城实验室通信部，广东 深圳 518055；4. 清华大学网络科学与网络空间研究院，北京 100084；
5.深圳技术大学大数据与互联网学院，广东 深圳 518118）

摘　要：为提升稀疏重构剪枝方法的效率和性能，提出一种基于融合特征稀疏重构的剪枝方法。首先，设计一种上下层特征融合策略，通过建模上层滤波器与下层对应滤波器通道之间的依赖关系，提取融合层内与层间交互信息的层间依赖特征，提升冗余滤波器选择的准确性；其次，基于提取的层间依赖特征构建 $\ell_{2,1}$ 范数约束的特征重构模型，通过模型优化实现上层滤波器与下层对应滤波器通道的联合结构化稀疏选择，提升冗余滤波器选择的鲁棒性；最后，在完成所有层的滤波器剪枝后，对剪枝后的轻量化模型进行一次性微调，提升模型的剪枝效率。实验结果表明，在常用的CIFAR-10数据集上，所提方法实现了在精度损失0.34%的情况下，VGG-16模型计算量降低62.0%以及参数量降低89.7%。这不仅从理论上证明了所提方法的可行性和有效性，而且在多个数据集上的实验结果也验证了该方法能够有效地实现深度神经网络压缩。

关键词：深度神经网络；滤波器剪枝；特征重构；$\ell_{2,1}$ 范数
中图分类号：TP183
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025174

Sparse pruning method based on filter parameters reconstruction

YANG Huoxiang[1], YI Shuangyan[2], MENG Fanyang[3], LIU Wei[2], LI Zongpeng[4], LIANG Yongsheng[1,5]
1. School of Electronic and Information Engineering, Shenzhen University, Shenzhen 518060, China
2. School of Computer and Software, Shenzhen University of Information Technology, Shenzhen 518172, China
3. Communication Department, Pengcheng Laboratory, Shenzhen 518055, China
4. Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China
5. College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China

Abstract: To enhance the efficiency and performance of sparse reconstruction-based pruning methods, a pruning method based on fused feature sparse reconstruction was proposed. Firstly, a cross-layer feature fusion strategy was developed by modeling the dependency relationships between upper-layer filters and corresponding lower-layer filter channels. This strategy extracted inter-layer dependency features, enhancing the accuracy of redundant filter selection. Subsequently, a $\ell_{2,1}$ norm-constrained feature reconstruction model was constructed based on these features. Through model optimization, joint structured sparse selection was performed on the upper-layer filters and corresponding lower-layer filter channels, enhancing the robustness of redundant filter selection. Finally, after pruning the filters in all layers, the compact structure was fine-tuned in a single step to optimize pruning efficiency. Experimental results on CIFAR-10 show that the proposed method achieves a 62.0% reduction in computational complexity and an 89.7% compression in storage space, with a 0.34% accuracy drop for the VGG-16 model. The proposed method proves effective both theoretically and experimentally, with results across multiple datasets demonstrating its efficacy in deep neural network compression.

Keywords: deep neural network, filter pruning, feature reconstruction, $\ell_{2,1}$ norm

收稿日期：2025–05–27；修回日期：2025–09–15
通信作者：梁永生，liangys@szu.edu.cn
基金项目：国家自然科学基金资助项目（No.61871154, No.62031013）；广东省重点建设学科科研能力提升项目（No.2022ZDJS117）
Foundation Items：The National Natural Science Foundation of China (No. 61871154, No. 62031013), The Guangdong Province Key Construction Discipline Scientific Research Capacity Improvement Project (No.2022ZDJS117)

0 引言

随着5G/6G通信系统与智能无线技术的快速发展，深度神经网络（DNN, deep neural network）被广泛引入通信系统，用于信道估计、信号检测、资源分配等关键任务，显著提升了系统的智能化水平与通信效率[1-3]。近年来，DNN同样成为语义通信系统中的核心支撑技术，尤其是在图像、语音等模态下的端到端语义编码与解码过程中，卷积神经网络（CNN, convolutional neural network）被广泛用于提取信源的深层语义表征，构建非线性特征映射与联合信源信道编码模块[4-5]。然而，通信系统中所采用的DNN模型通常结构复杂、参数庞大，将其部署于边缘设备或物联网终端等资源受限环境时，其高计算负载与存储开销严重制约了模型的实时推理与高效部署。

为应对这一挑战，学术界与工业界共同推进了模型压缩技术的发展，包括低秩分解[6]、参数量化[7]、知识蒸馏[8]和模型剪枝[9-38]等方法，旨在保持模型性能的同时减小模型大小和提高计算效率。其中，模型剪枝通过识别并移除冗余参数或结构，构建更加稀疏、轻量的神经网络，不仅具备坚实的理论基础和良好的压缩效果，还能够有效降低通信系统中的推理延迟，因而成为边缘智能通信与语义通信场景中的关键技术之一。

模型剪枝主要分为非结构化剪枝和结构化剪枝[10-12]。非结构化剪枝通过将不重要的参数置零实现稀疏化，但这些参数仍保留在模型中，因此无法有效减少内存占用，也难以直接应用于通用硬件加速。结构化剪枝则通过移除滤波器或神经元等模型结构单元实现压缩，由于剪枝后的模型保留了规则的运算结构，更易于编译优化，因此支持硬件层面的高效推理。本文聚焦于结构化剪枝方法，根据剪枝依据和分析对象的不同，可进一步划分为基于特征图的剪枝方法和基于滤波器参数的剪枝方法。

基于特征图的剪枝方法利用网络前向传播过程中的激活信息设计剪枝策略。文献[11-13]分别通过量化特征图的零激活比例、秩和熵值筛选冗余通道。文献[14]利用离散余弦变换分析特征图频域变化。文献[15]通过计算剪枝前后特征图的Frobenius范数差异，用于评估特征通道的重要性。文献[16]提出基于余弦空间相关性的两阶段剪枝方法，通过保留方向相近通道缓解误剪。文献[17]从通道间几何关系出发，设计了解耦结构下的跨通道剪枝策略。另一类方法通过特征图稀疏重构指导剪枝，文献[18-19]通过ℓ_0和ℓ_1范数稀疏约束优化特征图的稀疏重构，获得稀疏表示系数。文献[20]通过联合求解$\ell_{2,1}$范数约束重构误差项和稀疏正则化项，获得更鲁棒的稀疏表示系数。然而，上述剪枝方法高度依赖网络前向传播的特征采样。采样特征无法准确反映整个数据集的统计特性，直接将其作为剪枝算法的输入可能导致信息偏差，从而影响剪枝效果。尽管部分研究提出结合训练过程交替优化剪枝与网络权重[21-25]，但高计算开销仍是瓶颈。

基于滤波器参数的剪枝方法通过分析滤波器参数的分布特性来制定剪枝规则。文献[26]和文献[27]分别计算滤波器卷积核的ℓ_1范数和ℓ_2范数总和来评估其重要性。文献[28]通过统一阈值移除幅度较小参数实现压缩。文献[29]基于一阶泰勒展开，利用权重与梯度乘积平方和近似重要性进行剪枝。文献[30]提出基于几何中值的参数独立剪枝思想。文献[31]将剪枝形式化为协方差信息最小化问题。此外，文献[32-35]从滤波器相似性出发，分别采用欧氏距离、高斯统计、皮尔逊系数和K近邻图进行剪枝决策。进一步地，文献[36-38]指出单层剪枝忽略了层间依赖，提出跨层重要性建模方法，如文献[36-37]基于跨层卷积相关性构建度量标准，文献[38]通过融合当前层与下一层的相关参数，量化滤波器间的跨层相似性。通过上下层相关参数的融合量化跨层相似性。然而，此类方法对滤波器跨层依赖关系的建模能力不足，导致其在复杂网络结构中的剪枝性能存在瓶颈。

综上所述，基于特征图的剪枝方法和基于滤波器参数的剪枝方法均从早期层内独立剪枝发展到层内依赖建模剪枝，体现了模型内部依赖关系对剪枝任务的重要性。然而，现有研究主要集中于对层内依赖关系的探讨，或者对相邻层依赖关系的浅层建模。本文针对基于特征图的剪枝方法存在的计算开销高和剪枝性能不足的问题，提出一种基于融合特征稀疏重构的剪枝方法。本文认为这不仅能够加速获得轻量化网络结构的过程，而且能够更加准确地衡量滤波器的重要程度。本文创新点总结如下。

1) 利用滤波器参数替代特征图进行稀疏重构。一方面避免特征图抽样带来的信息偏差和前向推理开销；另一方面，滤波器参数数据量显著低于特征

图的数据量，降低重构算法的计算复杂度。

2) 设计基于连续两层滤波器参数的层间依赖特征提取方法。通过建模上层滤波器与下层对应滤波器通道的依赖关系，提取层间依赖特征，突破现有重构方法仅建模层内依赖的局限性。

3) 基于层间依赖特征，提出一种 $\ell_{2,1}$ 范数约束的特征重构模型。通过模型优化实现上层滤波器与下层对应滤波器通道的联合结构化稀疏选择，增强冗余滤波器选择的鲁棒性。

1　预备知识

定义一个卷积层数为 L 的 CNN 模型，忽略卷积层偏置的影响，模型卷积层参数可以表示为 $\boldsymbol{W}=\left\{\boldsymbol{W}^l \in \mathbb{R}^{N_l \times N_{l-1} \times k_l \times k_l}, 1 \leqslant l \leqslant L\right\}$。其中，$N_{l-1}$ 和 N_l 表示第 $l-1$ 和第 l 层的输出特征图的通道个数，$\boldsymbol{W}_i^l \in \mathbb{R}^{N_{l-1} \times k_l \times k_l}$ 表示 l 层的第 i 个滤波器。具体到第 l 层的卷积操作时，可以进一步细化该层在整个网络中的功能和作用，如图 1 所示。

第 l 层的滤波器 $\boldsymbol{W}_i^l$ 分别与从第 $l-1$ 层传递过来的特征图 $\boldsymbol{F}^{l-1}$ 进行卷积操作，每个滤波器对输入特征图的卷积生成了输出特征图的一个独立通道。在下一层中，每个通道的输出特征图与下一层滤波器对应的通道执行单通道卷积操作，后续经过卷积层的其他步骤处理并产生新的特征图。本文符号及其含义如表 1 所示。

表 1　本文符号及其含义

符号	含义
N_l	第 l 层输出特征图的通道个数
$\boldsymbol{W}^l$	第 l 层的所有滤波器集合
$\boldsymbol{W}_i^l$	第 l 层卷积层的第 i 个滤波器
$\boldsymbol{W}_j^l(i)$	滤波器 $\boldsymbol{W}_j^l$ 的第 i 个卷积核通道
$\boldsymbol{W}^l(i)$	第 l 层所有滤波器的第 i 个通道集合
$\boldsymbol{F}^l$	第 l 层的输出特征图
$\boldsymbol{F}_i^l$	$\boldsymbol{F}^l$ 的第 i 个特征通道
$\boldsymbol{Y}_i^l$	与 $\boldsymbol{F}_i^l$ 相关的特征
$\boldsymbol{Y}^l$	所有 $\boldsymbol{Y}_i^l$ 的集合

1.1　基于特征图稀疏重构的剪枝方法概述

基于稀疏约束的滤波器的剪枝方法可以视为如式(1)所示的在 ℓ_0 范数约束下的优化问题。

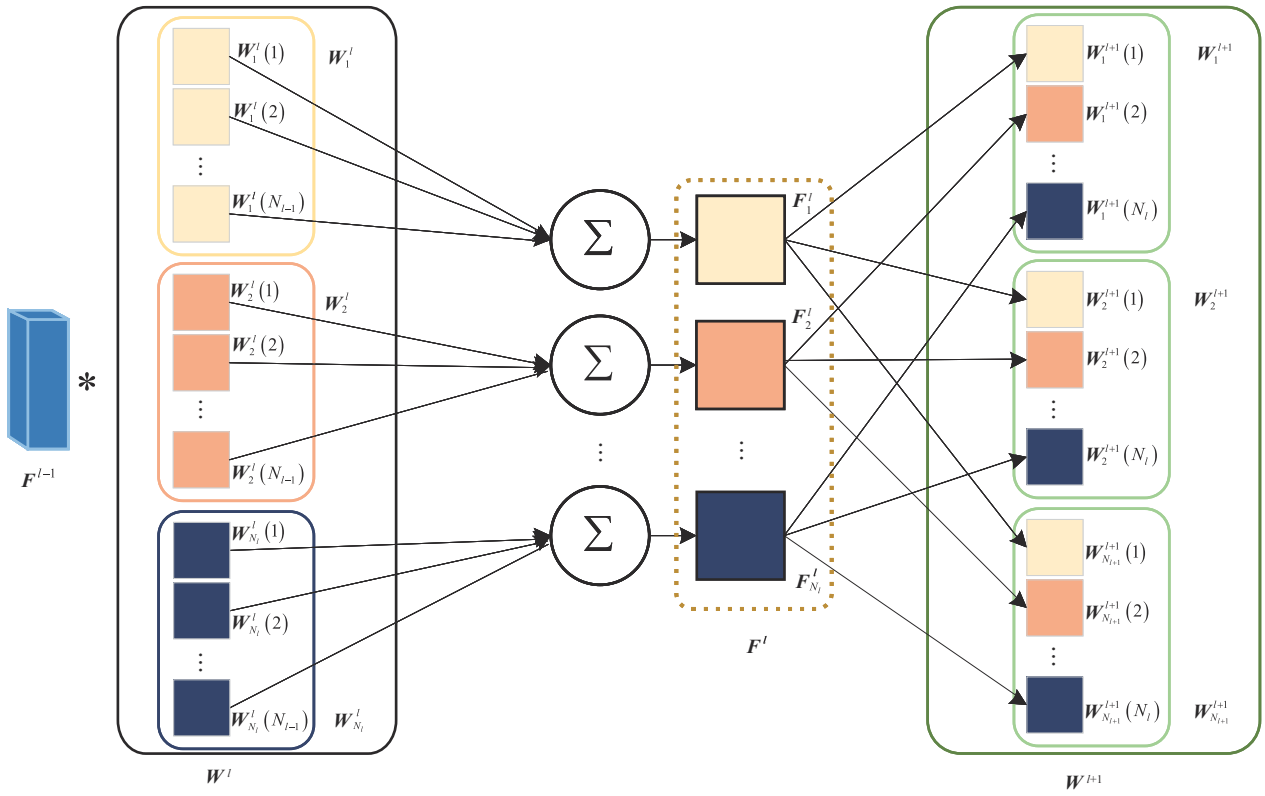


图 1　特征图与滤波器相互作用示意

$$\min_{\boldsymbol{\beta}}\left(\mathcal{L}\left(\tilde{\boldsymbol{W}}\right)+\lambda\left\|\boldsymbol{\beta}\right\|_0\right) \tag{1}$$

其中，$\boldsymbol{\beta}=\left\{\beta_i^l,1\leqslant i\leqslant N_l,1\leqslant l\leqslant L\right\}$；$\|\boldsymbol{v}\|_0$表示向量$\boldsymbol{v}$的0范数，即$\boldsymbol{v}$中非零元素的个数；$\beta_i^l\in\{0,1\}$对应$\boldsymbol{W}_i^l$是否被剪枝；$\tilde{\boldsymbol{W}}=\left\{\beta_i^l\boldsymbol{W}_i^l\in\mathbb{R}^{N_{l-1}\times k_l\times k_l},1\leqslant i\leqslant N_l,1\leqslant l\leqslant L\right\}$；$\mathcal{L}(\cdot)$用来量化剪枝后模型与原模型在性能上的差异。受范数约束在稀疏领域有效性的启发，许多研究将其引入模型剪枝。现有工作主要将模型稀疏化问题转化为联合特征重构误差与稀疏范数正则约束的最小化问题。Thinet[20]的优化问题可描述为

$$\min_{\boldsymbol{\beta}^l}\left(\sum_{i=1}^{N_l}\left(\left\|\boldsymbol{Y}_i^l-\sum_{n=1}^{N_{l-1}}\left(\left(\beta_n^l\boldsymbol{W}_i^l(n)\right)*\boldsymbol{X}_i^l\right)\right\|_2^2\right)+\lambda\left\|\boldsymbol{\beta}^l\right\|_0\right) \tag{2}$$

其中，$\boldsymbol{Y}_i^l$为$\boldsymbol{F}_i^l$上抽样特征点形成的向量；$\boldsymbol{X}_i^l$为卷积过程中$\boldsymbol{Y}_i^l$对应的输入；$\boldsymbol{\beta}=\left\{\beta_n^l,1\leqslant n\leqslant N_{l-1}\right\}$，$\beta_n^l\in\{0,1\}$用于判断是否删除第$l$层滤波器的第$n$个通道$\boldsymbol{W}^l(n)$，即是否剪枝$l-1$层相应的滤波器$\boldsymbol{W}_n^{l-1}$。

此优化问题通过利用向量ℓ_2范数来评估特征的重构误差，并通过向量ℓ_0范数对参数滤波器的通道数进行正则化约束，从而控制第$l-1$层所需保留的滤波器数量。区别于Thinet，在CP[21]方法中，当前层抽样特征形成特征矩阵，用于指导当前层滤波器剪枝，并采用Lasso回归方法构造如式(3)所示的优化问题。

$$\begin{aligned}&\arg\min_{\boldsymbol{\beta}^l,\boldsymbol{W}^l}\left(\frac{1}{2N}\left\|\boldsymbol{Y}^l-\sum_{i=1}^{N_l}\left(\beta_i^l\boldsymbol{W}_i^l\right)*\boldsymbol{X}^l\right\|_F^2+\lambda\left\|\boldsymbol{\beta}^l\right\|_1\right)\\&\text{s.t.}\left\|\boldsymbol{\beta}^l\right\|_0\leqslant c',\forall i\left\|\boldsymbol{W}_i^l\right\|_F=1\end{aligned} \tag{3}$$

其中，$\|\boldsymbol{v}\|_1$表示向量$\boldsymbol{v}$的1范数，即该向量中所有元素绝对值的总和；$\|\boldsymbol{A}\|_F$表示矩阵的Frobenius范数，即矩阵中$\boldsymbol{A}$所有元素平方和的平方根；$\boldsymbol{Y}^l$为抽样特征形成的矩阵；$\boldsymbol{X}^l$为对应于生成$\boldsymbol{Y}^l$的输入。该优化问题利用矩阵F范数来评估特征重构误差。$\boldsymbol{\beta}^l$向量的ℓ_1范数对滤波器参数进行正则约束，并通过该约束和$\|\boldsymbol{\beta}^l\|_0\leqslant c'$控制每层的剪枝率。TSFR[22]则将剪枝问题转换成$\ell_{2,1}$范数约束下的单层抽样特征的重构问题，具体优化问题为

$$\arg\min_{\boldsymbol{A}}\left\|\boldsymbol{Y}^l-\boldsymbol{A}\boldsymbol{Y}^l\right\|_{2,1}+\lambda\left\|\boldsymbol{A}\right\|_{2,1} \tag{4}$$

其中，$\|\boldsymbol{A}\|_{2,1}$表示矩阵$\boldsymbol{A}$的$\ell_{2,1}$范数，且$\|\boldsymbol{A}\|_{2,1}=\sum_{i=1}^{n}\left\|\boldsymbol{A}^i\right\|_2$，$\boldsymbol{A}^i$表示矩阵的第$i$列，$\|\boldsymbol{A}^i\|_2$表示向量$\boldsymbol{A}^i$的2范数。区别于Thinet和CP，求解该优化问题过程中不需要复杂的卷积操作过程，优化求解后得到呈现列一致性稀疏的系数矩阵$\boldsymbol{A}$，并用$\boldsymbol{A}$中的第i列稀疏程度度量第i个滤波器的重要性。

然而，上述方法通过欧氏距离平方和来计算重构误差，容易受到异常值的干扰，这一局限性影响了特征重构过程的准确性[39]。为了克服特征抽样对算法带来的不稳定性，本文采用一种不需要抽样的基于滤波器参数重构的滤波器剪枝方法；同时为了提高重构模型对异常值的鲁棒性，在特征重构的过程中引入自动化均值学习策略，降低异常值的影响。

1.2　基于滤波器参数的剪枝方法概述

基于滤波器参数的剪枝方法在选择重要滤波器的过程中不需要输入数据的参与，仅通过探索滤波器参数分布来设计剪枝方案。通过保留对模型重要的滤波器来缩小剪枝后模型与原始模型之间的性能差异。通过引入二值变量$\boldsymbol{m}_i^l\in\{0,1\}$来判断滤波器$\boldsymbol{W}_i^l$是否应被剪枝，当$\boldsymbol{m}_i^l=0$时意味着$\boldsymbol{W}_i^l$应该被剪枝。针对逐层剪枝策略的第$l$层，定义其剪枝率为$P_l$，剪枝问题可被描述为

$$\begin{aligned}&\min_{\boldsymbol{m}_i^l}\sum_{l=1}^{L}\sum_{i=1}^{N_l}\boldsymbol{m}_i^lE\left(\boldsymbol{W}_i^l\right)\\&\text{s.t.}\sum_{i=1}^{N_l}\boldsymbol{m}_i^l=N_l\left(1-P_l\right)\end{aligned} \tag{5}$$

其中，$E\left(\boldsymbol{W}_i^l\right)$用于对滤波器$\boldsymbol{W}_i^l$重要性评价。然而，现有的基于滤波器参数的剪枝方法聚焦于分析当前层滤波器之间的相互关系或独立性，却常常忽视了相邻层滤波器之间的依赖关系。从图1可以看出，特征通道$\boldsymbol{F}_i^l$是模型前向传播过程中滤波器$\boldsymbol{W}_i^l$作用后的产物，该特征通道不仅是当前层的输出，同时$\boldsymbol{F}_i^l$也作为下一层卷积的输入，并与下一层所有滤波器的第i个通道$\boldsymbol{W}^{l+1}(i)$相互作用，共同参与下一层的卷积运算。在剪枝的过程中，为了不破坏深度神经网络的连续性，$\boldsymbol{W}_i^l$和$\boldsymbol{W}^{l+1}(i)$只能同时剪枝或者保留。这一现象表明，仅通过单层滤波器指导剪枝难以准确度量滤波器的重要程度。

鉴于此，本文提出一种基于融合特征稀疏重构的剪枝方法。该方法不仅探讨了单层滤波器之间的

内在联系，同时也延伸到对相邻两层滤波器之间的依赖关系研究。通过挖掘层间依赖关系，本文方法能够更精确地评估滤波器通道和滤波器的重要性，从而提升剪枝模型的性能表现。

2　剪枝框架

根据前面的分析，上层滤波器与下层对应滤波器通道通过特征通道建立依赖关系，因此只能同时剪枝或保留，删除上一层滤波器和下一层滤波器通道，均是对滤波器参数进行结构化稀疏处理。鉴于 $\ell_{2,1}$ 范数在稀疏结构上的优势，本文提出在 $\ell_{2,1}$ 范数约束下，同时对上下两层滤波器参数进行稀疏重构，具体模型为

$$\left\|\boldsymbol{W}^{l}-\boldsymbol{A}\boldsymbol{W}^{l}\right\|_{2,1}+\left\|\left(\boldsymbol{W}^{l+1}\right)^{\mathrm{T}}-\boldsymbol{A}\left(\boldsymbol{W}^{l+1}\right)^{\mathrm{T}}\right\|_{2,1}+\lambda\|\boldsymbol{A}\|_{2,1} \tag{6}$$

其中，$\boldsymbol{W}^{l}$、$\boldsymbol{W}^{l+1}$ 被表示成矩阵形式 $\boldsymbol{W}^{l}\in\mathbb{R}^{N_l\times(N_{l-1}\times k_l\times k_l)}$、$\boldsymbol{W}^{l+1}\in\mathbb{R}^{(N_{l+1}\times k_{l+1}\times k_{l+1})\times N_l}$。本文同时对上下两层滤波器参数进行重构。首先，模型式(6)同时利用了上层所有滤波器之间的相关性和下层滤波器通道之间的相关性。其次，模型式(6)还利用了上层滤波器与下层滤波器通道的依赖关系，即系数矩阵 $\boldsymbol{A}$ 中第 i 列的稀疏性同时反映上层第 i 个滤波器和下层所有滤波器第 i 个通道的冗余性。为了便于优化，本文对 $\boldsymbol{W}^{l}$ 和 $\boldsymbol{W}^{l+1}$ 参数进行融合，构建层间依赖特征集 $\boldsymbol{Y}^{l}$。由于采用了特定的层间参数融合策略，模型输入数据量大小进一步减少，在此基础上，优化问题式(6)简化为在 $\ell_{2,1}$ 范数约束下，针对低维特征集 $\boldsymbol{Y}^{l}$ 进行最小化重构误差的优化问题。

本文剪枝流程在图2的顶部虚线框内呈现，主要包括滤波器剪枝和模型微调2个部分。滤波器剪枝部分由构建层间依赖特征和层间依赖特征重构2个关键步骤构成。

构建层间依赖特征：如图2左下角虚线框所示，通过分析和建模相邻两层卷积层的依赖关系提取层间依赖特征，并将其作为特征重构算法的输入。

层间依赖特征重构：如图2右下角虚线框所示，采用一个基于 $\ell_{2,1}$ 范数约束的特征重构模型，通过优化求解该模型来识别层间依赖特征中的冗余特征，同时输出滤波器冗余性判断向量。

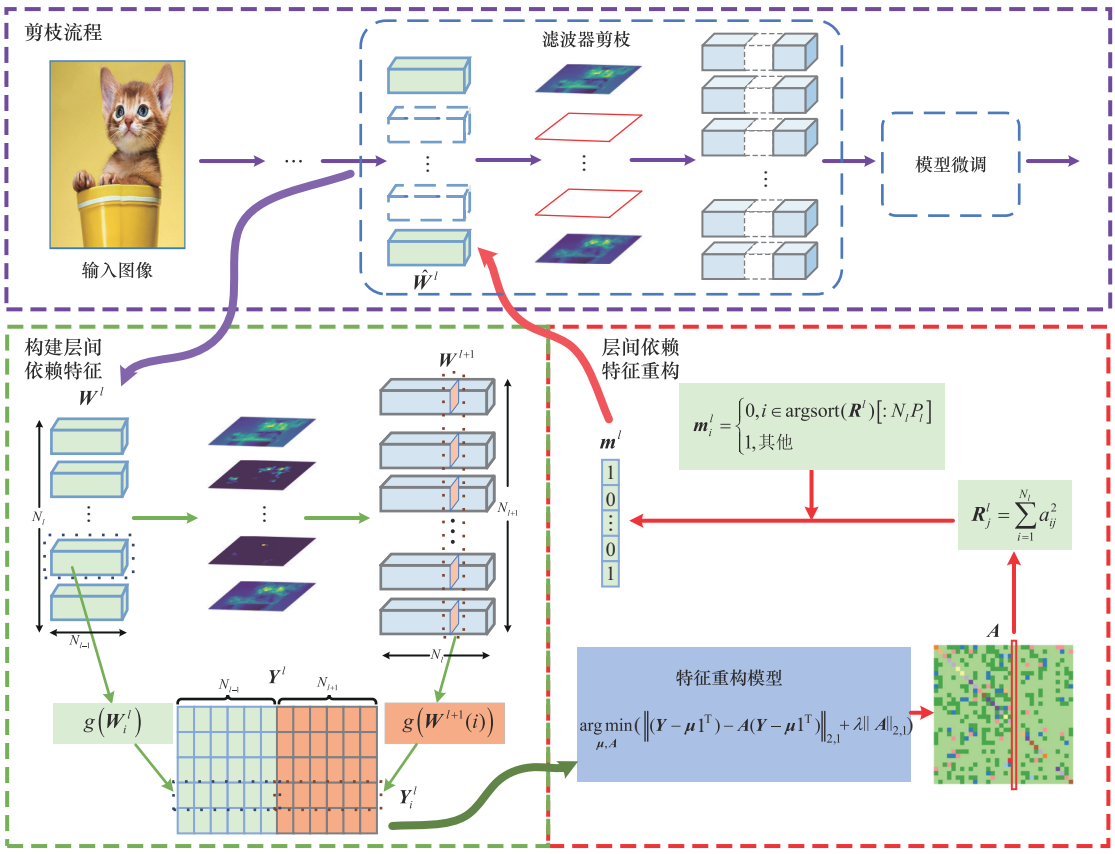


图2　本文所提剪枝框架示意

在模型微调中，当所有层全部剪枝完成后，通过一次性微调所有层的方式恢复剪枝模型的性能。

2.1 构建层间依赖特征

该步骤考虑了相邻层滤波器之间的依赖关系，采用精心设计的特征融合策略提取层间依赖特征，提升滤波器重要性度量的准确性。

滤波器层间依赖分析：对于第l个卷积层，一个滤波器$\boldsymbol{W}_i^l$的作用是实现输入特征图$\boldsymbol{F}^{l-1}$到与输出特征通道$\boldsymbol{F}_i^l$的映射。$\boldsymbol{F}_i^l$在模型的前向传播过程中继续与下层滤波器中的第i个通道$\boldsymbol{W}^{l+1}(i)$相互作用。具体而言，第l层的输出通道$\boldsymbol{F}_i^l$在模型的前向传播中参与如式(7)所示的非线性操作。

$$\begin{aligned}\boldsymbol{F}_i^l &= \sigma\left(\boldsymbol{W}_i^l * \boldsymbol{F}^{l-1}\right)\\ \boldsymbol{F}_j^{l+1} &= \sigma\left(\sum_{i=1}^{N_l}\left(\boldsymbol{W}_j^{l+1}(i) * \boldsymbol{F}_i^l\right)\right)\end{aligned} \tag{7}$$

其中，$*$表示单通道的卷积运算，σ为非线性激活函数。从式(7)可以看出，上层的滤波器输出特征同时也是下层滤波器的输入，两层滤波器之间通过上层的输出特征建立层间依赖关系。

计算层间依赖特征：根据上述的分析，上层滤波器$\boldsymbol{W}_i^l$与下层滤波器通道$\boldsymbol{W}^{l+1}(i)$通过特征通道$\boldsymbol{F}_i^l$建立依赖关系。在剪枝过程中，剪枝一个滤波器，下层滤波器相应的通道也同时被删除。本文结合图1和上述对式(7)的描述，联合提取$\boldsymbol{W}_i^l$及其对应的下层滤波器通道$\boldsymbol{W}^{l+1}(i)$，并通过函数$g(\cdot)$处理后作为层间依赖特征作为剪枝模型的输入，层间依赖特征表示为

$$\boldsymbol{Y}_i^l = \left[g\left(\boldsymbol{W}_i^l\right), g\left(\boldsymbol{W}^{l+1}(i)\right)\right] \tag{8}$$

其中，$\boldsymbol{W}_i^l \in \mathbb{R}^{N_{l-1} \times k_l \times k_l}$表示$\boldsymbol{W}^l$的第$i$个滤波器，$\boldsymbol{W}^{l+1}(i) \in \mathbb{R}^{N_{l+1} \times k_{l+1} \times k_{l+1}}$表示$\boldsymbol{W}^{l+1}$中所有滤波器第$i$个通道的集合。针对$\boldsymbol{X} \in \mathbb{R}^{a \times b \times c}$，$g(\boldsymbol{X})$定义为

$$g(\boldsymbol{X}) = \left[\|\boldsymbol{X}_1\|_{\mathrm{F}}, \|\boldsymbol{X}_2\|_{\mathrm{F}}, \cdots, \|\boldsymbol{X}_a\|_{\mathrm{F}}\right] \tag{9}$$

通过$g(\cdot)$运算，获得以向量形式呈现的层间依赖特征$\boldsymbol{Y}_i^l \in \mathbb{R}^{(N_{l-1}+N_{l+1}) \times 1}$，然后将多个$\boldsymbol{Y}_i^l$进行组合，得到以矩阵形式呈现的层间依赖特征集$\boldsymbol{Y}^l \in \mathbb{R}^{N_l \times (N_{l-1}+N_{l+1})}$

$$\boldsymbol{Y}^l = \left[\boldsymbol{Y}_1^l, \boldsymbol{Y}_2^l, \cdots, \boldsymbol{Y}_{N_l}^l\right] \tag{10}$$

针对3×3的卷积核，相比模型式(6)的输入数据维度$N_l \times \left(9(N_{l-1}+N_{l+1})\right)$，$\boldsymbol{Y}^l$数据总量降低89%。

2.2 层间依赖特征重构

针对构建的层间依赖特征集$\boldsymbol{Y}^l$，优化问题式(5)被重新表示为

$$\begin{aligned}&\min_{\boldsymbol{m}_i^l} \sum_{l=1}^{L}\sum_{i=1}^{N_l} \boldsymbol{m}_i^l \hat{\mathcal{L}}\left(\boldsymbol{Y}_i^l\right)\\ &\text{s.t.} \sum_{i=1}^{N_l} \boldsymbol{m}_i^l = N_l(1-P_l)\end{aligned} \tag{11}$$

其中，$\boldsymbol{m}_i^l$为一个二值变量，当$\boldsymbol{m}_i^l = 1$时代表$\boldsymbol{Y}_i^l$不重要，应该被删除，同时也意味着相应的滤波器$\boldsymbol{W}_i^l$应当被剪枝。

建立特征重构模型：本文引入自动学习的均值向量$\boldsymbol{\mu}^l$，提出一种基于$\ell_{2,1}$范数稀疏约束的鲁棒特征重构方法。对$\boldsymbol{Y}^l$的稀疏重构模型为

$$\arg\min_{\boldsymbol{\mu}^l, \boldsymbol{A}^l} \left\|(\boldsymbol{Y}^l - \boldsymbol{\mu}^l\boldsymbol{1}^{\mathrm{T}}) - \boldsymbol{A}^l(\boldsymbol{Y}^l - \boldsymbol{\mu}^l\boldsymbol{1}^{\mathrm{T}})\right\|_{2,1} + \lambda\left\|\boldsymbol{A}^l\right\|_{2,1} \tag{12}$$

其中，$\boldsymbol{Y}^l \in \mathbb{R}^{N_l \times (N_{l-1}+N_l)}$，$\boldsymbol{1} \in \mathbb{R}^{(N_{l-1}+N_l) \times 1}$为元素全为1的列向量，$\boldsymbol{A} \in \mathbb{R}^{N_l \times N_l}$表示稀疏系数矩阵，$\boldsymbol{\mu}^l$表示第$l$层所有$\boldsymbol{Y}_i^l$的均值向量。区别于现有$\ell_{2,1}$范数约束下的特征重构模型，该模型中的均值向量是自动学习的，即通过优化均值中心得到最优均值。模型中的$\ell_{2,1}$范数用于实现重构误差项和正则约束项的列一致性，λ是平衡重构项和正则项的参数。模型式(12)适用于网络的所有卷积层，为了方便表述，忽略l的影响，模型式(12)简化为

$$\arg\min_{\boldsymbol{\mu}, \boldsymbol{A}} \|(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}}) - \boldsymbol{A}(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}})\|_{2,1} + \lambda\|\boldsymbol{A}\|_{2,1} \tag{13}$$

对于任意矩阵$\boldsymbol{M}$，用$\boldsymbol{M}^i$表示矩阵$\boldsymbol{M}$的第i列，则$\|\boldsymbol{A}\|_{2,1}$可以转化为

$$\begin{aligned}&\|\boldsymbol{A}\|_{2,1} = \|\boldsymbol{A}^1\|_2 + \cdots + \|\boldsymbol{A}^{N_l}\|_2 =\\ &\left\|\frac{\boldsymbol{A}^1}{\sqrt{\|\boldsymbol{A}^1\|_2}}\right\|_2^2 + \cdots + \left\|\frac{\boldsymbol{A}^{N_l}}{\sqrt{\|\boldsymbol{A}^{N_l}\|_2}}\right\|_2^2 =\\ &\left\|\left(\frac{\boldsymbol{A}^1}{\sqrt{\|\boldsymbol{A}^1\|_2}}, \cdots, \frac{\boldsymbol{A}^{N_l}}{\sqrt{\|\boldsymbol{A}^{N_l}\|_2}}\right)\right\|_{\mathrm{F}}^2 =\\ &\left\|\left(\boldsymbol{A}^1, \cdots, \boldsymbol{A}^{N_l}\right)\begin{pmatrix}\frac{1}{\sqrt{\|\boldsymbol{A}^1\|_2}} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{\|\boldsymbol{A}^{N_l}\|_2}}\end{pmatrix}\right\|_{\mathrm{F}}^2\end{aligned} \tag{14}$$

当$\sqrt{\| \boldsymbol{A}^i \|_2} = 0$时，式(14)中的$\frac{1}{\sqrt{\| \boldsymbol{A}^i \|_2}}$应为$\frac{1}{\sqrt{\| \boldsymbol{A}^i \|_2} + \zeta}$，$\zeta$为一个很小的正数。进而模型式(13)可以转换成

$$\arg\min_{\boldsymbol{\mu},\boldsymbol{A}} \left\| \left(\left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right) - \boldsymbol{A}\left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right) \right) \sqrt{\boldsymbol{W}_1} \right\|_{\mathrm{F}}^2 + \lambda \left\| \boldsymbol{A} \sqrt{\boldsymbol{W}_2} \right\|_{\mathrm{F}}^2 \tag{15}$$

其中，$\boldsymbol{W}_1 \in \mathbb{R}^{\left(N_{l-1}+N_l\right) \times \left(N_{l-1}+N_l\right)}$和$\boldsymbol{W}_2 \in \mathbb{R}^{N_l \times N_l}$为两个对角矩阵。$\boldsymbol{W}_2$可表示为

$$\boldsymbol{W}_2 = \begin{pmatrix} \frac{1}{\| \boldsymbol{A}^1 \|_2} & & \\ & \ddots & \end{pmatrix} \tag{16}$$

令$\boldsymbol{W}_2^{ii}$为$\boldsymbol{W}_2$的对角线元素，当$\| \boldsymbol{A}^i \|_2 = 0$时，有

$$\boldsymbol{W}_2^{ii} = \frac{1}{\| \boldsymbol{A}^i \|_2 + \zeta}$$

令$\boldsymbol{B} = \left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right) - \boldsymbol{A}\left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right)$，$\boldsymbol{B}^{\boldsymbol{i}}$为其第$i$列，则$\boldsymbol{W}_1$可表示为

$$\boldsymbol{W}_1 = \begin{pmatrix} \frac{1}{\left\| \boldsymbol{B}^1 \right\|_2} & & \\ & \ddots & \end{pmatrix} \tag{17}$$

同样，当$\left\| \left(\left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right) - \boldsymbol{A}\left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right) \right)^i \right\|_2 = 0$时，有

$$\boldsymbol{W}_1^{ii} = \frac{1}{\left\| \left(\left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right) - \boldsymbol{A}\left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right) \right)^i \right\|_2 + \zeta}$$

首先，通过$\sqrt{\boldsymbol{W}_1}$给层间依赖特征添加一个权重，重要的特征被赋予更高的权重。其次，正则项$\boldsymbol{A}\sqrt{\boldsymbol{W}_2}$用于指导层间依赖特征重要性的评价，$\boldsymbol{W}_2^{ii}$越小意味着第$i$个层间依赖特征越重要，$\boldsymbol{W}_1^{ii}$越小意味着第$i$个层间依赖特征越离群。在优化模型式(15)的过程中，当$\boldsymbol{W}_2$中元素值较大时，$\| \boldsymbol{A}^i \|_2$被限制为较小的值，从而获得稀疏的$\boldsymbol{A}$。图3可视化了ResNet-50第一个卷积层中的$\boldsymbol{A}$。从图3中可以看出，$\boldsymbol{A}$中部分列全部接近0，反映了层间依赖特征集的冗余性。λ用于控制$\boldsymbol{A}$的列稀疏程度。当λ较大时，$\boldsymbol{A}$呈现明显列一致性稀疏，导致滤波器被大量移除，若此时误减掉重要的滤波器，会降低模型性能；反之，当λ较小时，$\boldsymbol{A}$呈现的列一致性稀疏较弱，仅少量冗余滤波器被移除，模型精度能够有效维持，但整体压缩效率受限。本文根据每层剪枝率的大小，采用网格搜索的方式寻找最优λ。

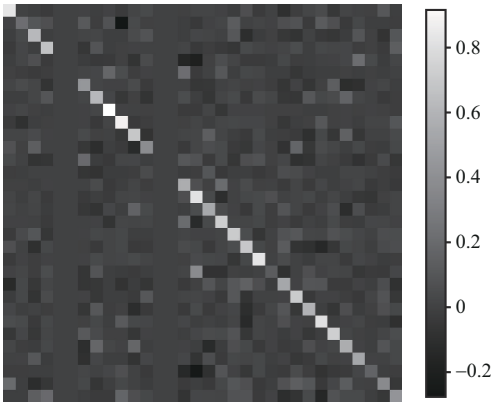


图3　系数矩阵$\boldsymbol{A}$可视化

特征重构模型优化：为有效解决模型式(15)中的最小化问题，模型优化过程采用了两步迭代的优化策略。

第一步，固定$\boldsymbol{\mu}$，优化$\boldsymbol{A}$。此时，优化问题转化为

$$\arg\min_{\boldsymbol{A}} \left\| \left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right) - \boldsymbol{A}\left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right) \sqrt{\boldsymbol{W}_1} \right\|_{\mathrm{F}}^2 + \lambda \left\| \boldsymbol{A} \sqrt{\boldsymbol{W}_2} \right\|_{\mathrm{F}}^2 \tag{18}$$

令式(18)对$\boldsymbol{A}$的导数为0，得到$\boldsymbol{A}$的更新规则，表示为

$$\boldsymbol{A} = \left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right) \boldsymbol{W}_1 \left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right)^{\mathrm{T}} \cdot \left(\lambda \boldsymbol{W}_2 + \left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right) \boldsymbol{W}_1 \left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right)^{\mathrm{T}} \right)^{-1} \tag{19}$$

这一步用于更新系数矩阵$\boldsymbol{A}$，以寻找最优的层间依赖特征，反映了层间依赖特征之间的相关性。

第二步，固定$\boldsymbol{A}$，优化$\boldsymbol{\mu}$。此时，优化问题转化为

$$\arg\min_{\boldsymbol{\mu}} \left\| \left(\left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right) - \boldsymbol{A}\left(\boldsymbol{Y} - \boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}} \right) \right) \sqrt{\boldsymbol{W}_1} \right\|_{\mathrm{F}}^2 \tag{20}$$

令式(20)对$\boldsymbol{\mu}$的倒数为0，可以得出优化$\boldsymbol{\mu}$的更新规则，表示为

$$\boldsymbol{\mu} = \frac{\left(\boldsymbol{Y}\boldsymbol{W}_1 - \boldsymbol{A}\boldsymbol{Y}\boldsymbol{W}_1 \right) \boldsymbol{1}}{\boldsymbol{1}^{\mathrm{T}} \boldsymbol{W}_1 \boldsymbol{1}} \tag{21}$$

这一步用于调整均值向量$\boldsymbol{\mu}$，以自动学习的方式获得更优的均值向量。通过这种两步迭代的策

略，在特征重构过程中交替优化$\boldsymbol{A}$和$\boldsymbol{\mu}$，逐步逼近最优解，其具体的优化过程如算法1所示。

算法1　模型式(15)的优化求解过程

输入　第l层的层间依赖特征集$\boldsymbol{Y}^l$，简写为$\boldsymbol{Y}$，参数λ

输出　系数矩阵$\boldsymbol{A}$

初始化 $\boldsymbol{W}_1=\boldsymbol{I}$，$\boldsymbol{W}_2=\boldsymbol{I}$，$\boldsymbol{\mu}=0$，$t=1$，$\varepsilon=0.1$

开始循环：

1) 更新$\boldsymbol{A}$：

$$\boldsymbol{A}=\left(\boldsymbol{Y}-\boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}}\right)\boldsymbol{W}_1\left(\boldsymbol{Y}-\boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}}\right)^{\mathrm{T}}\left(\lambda\boldsymbol{W}_2+(\boldsymbol{Y}-\boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}})\boldsymbol{W}_1(\boldsymbol{Y}-\boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}})^{\mathrm{T}}\right)^{-1}$$

2) 更新$\boldsymbol{\mu}$：

$$\boldsymbol{\mu}=\frac{(\boldsymbol{Y}\boldsymbol{W}_1-\boldsymbol{A}\boldsymbol{Y}\boldsymbol{W}_1)\boldsymbol{1}}{\boldsymbol{1}^{\mathrm{T}}\boldsymbol{W}_1\boldsymbol{1}}$$

3) 根据式(16)和式(17)更新$\boldsymbol{W}_1$和$\boldsymbol{W}_2$；

4) 计算模型损失：

$$\mathrm{obj}(t)=\|(\boldsymbol{Y}-\boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}})-\boldsymbol{A}(\boldsymbol{Y}-\boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}})\|_{2,1}+\lambda\|\boldsymbol{A}\|_{2,1}$$

5) 当$t>1$，计算ε：

$$\varepsilon=\left|\mathrm{obj}(t)-\mathrm{obj}(t-1)\right|$$

6) 收敛性判断：$\varepsilon\geqslant 10^{-4}$，继续执行循环，否则终止循环；

7) $t\leftarrow t+1$；

循环终止；

特征重构模型收敛性分析：在展开特征重构模型收敛性分析之前，首先介绍一个关键的数学前提，即引理1，它为后续证明模型收敛性提供了必要的理论基础。

引理1　对于任意非零向量$\boldsymbol{p}$和$\boldsymbol{q}$，不等式(22)始终成立[40]。

$$\|\boldsymbol{p}\|_2-\frac{\|\boldsymbol{p}\|_2}{2\|\boldsymbol{q}\|_2}\leqslant\|\boldsymbol{q}\|_2-\frac{\|\boldsymbol{q}\|_2}{2\|\boldsymbol{q}\|_2}\tag{22}$$

令$\tilde{\boldsymbol{A}}$和$\tilde{\boldsymbol{\mu}}$表示迭代更新后的$\boldsymbol{A}$和$\boldsymbol{\mu}$，根据$\boldsymbol{W}_1$和$\boldsymbol{W}_2$的定义可得

$$\sum_{i=1}^{n}\left(\frac{\left\|(\boldsymbol{Y}^i-\tilde{\boldsymbol{A}}\boldsymbol{Y}^i)-(\boldsymbol{I}-\tilde{\boldsymbol{A}})\tilde{\boldsymbol{\mu}}\right\|_2^2}{2\left\|(\boldsymbol{Y}^i-\boldsymbol{A}\boldsymbol{Y}^i)-(\boldsymbol{I}-\boldsymbol{A})\boldsymbol{\mu}\right\|_2}+\lambda\frac{\|\tilde{\boldsymbol{A}}^i\|_2^2}{2\|\boldsymbol{A}^i\|_2}\right)\leqslant\sum_{i=1}^{n}\left(\frac{\left\|(\boldsymbol{Y}^i-\boldsymbol{A}\boldsymbol{Y}^i)-(\boldsymbol{I}-\boldsymbol{A})\boldsymbol{\mu}\right\|_2^2}{2\left\|(\boldsymbol{Y}^i-\boldsymbol{A}\boldsymbol{Y}^i)-(\boldsymbol{I}-\boldsymbol{A})\boldsymbol{\mu}\right\|_2}+\lambda\frac{\|\boldsymbol{A}^i\|_2^2}{2\|\boldsymbol{A}^i\|_2}\right)\tag{23}$$

进一步，根据式(22)可得

$$\sum_{i=1}^{n}\left(\left\|(\boldsymbol{Y}^i-\tilde{\boldsymbol{A}}\boldsymbol{Y}^i)-(\boldsymbol{I}-\tilde{\boldsymbol{A}})\tilde{\boldsymbol{\mu}}\right\|_2-\frac{\left\|(\boldsymbol{Y}^i-\tilde{\boldsymbol{A}}\boldsymbol{Y}^i)-(\boldsymbol{I}-\tilde{\boldsymbol{A}})\tilde{\boldsymbol{\mu}}\right\|_2^2}{2\left\|(\boldsymbol{Y}^i-\boldsymbol{A}\boldsymbol{Y}^i)-(\boldsymbol{I}-\boldsymbol{A})\boldsymbol{\mu}\right\|_2}\right)\leqslant\sum_{i=1}^{n}\left(\left\|(\boldsymbol{Y}^i-\boldsymbol{A}\boldsymbol{Y}^i)-(\boldsymbol{I}-\boldsymbol{A})\boldsymbol{\mu}\right\|_2-\frac{\left\|(\boldsymbol{Y}^i-\boldsymbol{A}\boldsymbol{Y}^i)-(\boldsymbol{I}-\boldsymbol{A})\boldsymbol{\mu}\right\|_2^2}{2\left\|(\boldsymbol{Y}^i-\boldsymbol{A}\boldsymbol{Y}^i)-(\boldsymbol{I}-\boldsymbol{A})\boldsymbol{\mu}\right\|_2}\right)\tag{24}$$

同理可得

$$\sum_{i=1}^{n}\left(\|\tilde{\boldsymbol{A}}^i\|_2-\frac{\|\tilde{\boldsymbol{A}}^i\|_2^2}{2\|\boldsymbol{A}^i\|_2}\right)\leqslant\sum_{i=1}^{n}\left(\|\boldsymbol{A}^i\|_2-\frac{\|\boldsymbol{A}^i\|_2^2}{2\|\boldsymbol{A}^i\|_2}\right)\tag{25}$$

将式(23)和式(24)的结果相加，并累加式(25)的λ倍，可以推导出

$$\sum_{i=1}^{n}\left(\left\|(\boldsymbol{Y}^i-\tilde{\boldsymbol{A}}\boldsymbol{Y}^i)-(\boldsymbol{I}-\tilde{\boldsymbol{A}})\tilde{\boldsymbol{\mu}}\right\|_2+\lambda\|\tilde{\boldsymbol{A}}^i\|_2\right)\leqslant\sum_{i=1}^{n}\left(\left\|(\boldsymbol{Y}^i-\boldsymbol{A}\boldsymbol{Y}^i)-(\boldsymbol{I}-\boldsymbol{A})\boldsymbol{\mu}\right\|_2+\lambda\|\boldsymbol{A}^i\|_2\right)\tag{26}$$

式(26)等价于

$$\|(\boldsymbol{Y}-\tilde{\boldsymbol{\mu}}\boldsymbol{1}^{\mathrm{T}})-\tilde{\boldsymbol{A}}(\boldsymbol{Y}-\tilde{\boldsymbol{\mu}}\boldsymbol{1}^{\mathrm{T}})\|_{2,1}+\lambda\|\tilde{\boldsymbol{A}}\|_{2,1}\leqslant\|(\boldsymbol{Y}-\boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}})-\boldsymbol{A}(\boldsymbol{Y}-\boldsymbol{\mu}\boldsymbol{1}^{\mathrm{T}})\|_{2,1}+\lambda\|\boldsymbol{A}\|_{2,1}\tag{27}$$

至此，模型式(12)的收敛性得到验证。

3　剪枝流程

本文提出的剪枝方法流程如图4所示。针对一个预训练模型，剪枝该模型主要分为2个部分：滤波器剪枝和模型微调。

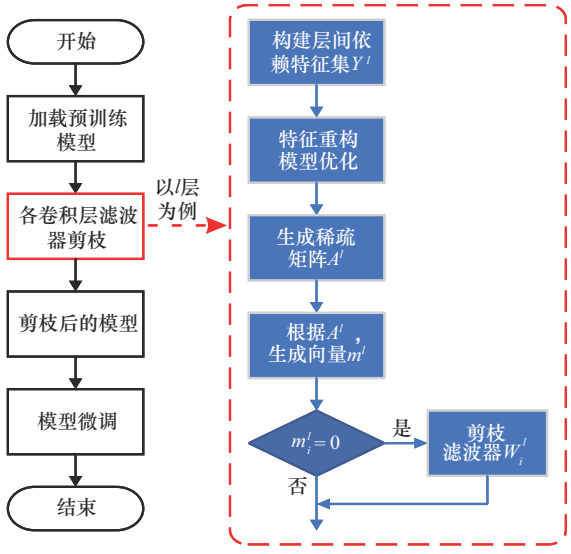


图4　本文模型剪枝方法流程

在滤波器剪枝过程中，本文采用逐层剪枝的方式，当对第 l 层卷积层实施滤波器剪枝操作时，包括构建层间依赖特征和特征重构 2 个步骤。层间依赖特征通过融合连续两层卷积层的滤波器相关信息而形成，特征重构通过对层间依赖特征集的重构获得列稀疏的系数矩阵 $\boldsymbol{A}^l$，然后计算矩阵 $\boldsymbol{A}^l$ 的列平方和得到向量 $\boldsymbol{R}^l \in \mathbb{R}^{N_l \times 1}$，$\boldsymbol{R}^l$ 中每个元素的计算式为

$$\boldsymbol{R}_j^l = \sum_{i=1}^{N_l} a_{ij}^2,\ j = 1,2,\cdots,N_l \tag{28}$$

其中，a_{ij} 表示 $\boldsymbol{A}^l$ 中第 i 行第 j 列的值。根据当前层的剪枝率 P_l，将 $\boldsymbol{R}_j^l$ 中较小的 N_lP_l 个元素置为 0，其他元素置为 1，得到滤波器冗余性判断向量 $\boldsymbol{m}^l \in \mathbb{R}^{N_l \times 1}$，其计算式为

$$\boldsymbol{m}_i^l = \begin{cases} 0, i \in \text{argsort}(\boldsymbol{R}^l)\left[:N_lP_l\right] \\ 1, \text{其他} \end{cases} \tag{29}$$

当 $\boldsymbol{m}_i^l$=0 时，表示滤波器 $\boldsymbol{W}_i^l$ 冗余，应该被删除，同时 $\boldsymbol{W}^{l+1}$ 中所有滤波器的第 i 个通道也被删除；当 $\boldsymbol{m}_i^l$=1 时，表示滤波器 $\boldsymbol{W}_i^l$ 被保留。

对所有卷积层完成剪枝操作后，虽然获得了轻量化模型以提升推理效率，但通常也会伴随模型性能的下降，本文采用对剪枝后的模型所有层一次性微调的方法，通过调整保留滤波器的参数以恢复模型性能。

4　实验分析

4.1　图像分类实验配置

为评估所提剪枝算法的泛化性与有效性，本文在 CIFAR-10[41] 与 ImageNet[42] 2 个主流数据集上进行了实验分析。剪枝评估指标包括 Top-1 和 Top-5 分类准确率及其下降率、参数量，以及以浮点运算总量（FLOPs）衡量的计算量。针对 CIFAR-10 数据集，采用随机水平翻转与标准化进行数据增强，利用 SGD 优化器对剪枝后的模型微调 150 个训练周期（epoch）。其他设置如下：动量为 0.9，批量大小为 256，初始学习率为 0.1，并在第 50 和 100 个 epoch 将学习率降为原来的 10%。针对 ImageNet 数据集，采用随机翻转、随机裁剪和标准化操作进行数据增强。在与大多数其他剪枝方法对比时，本文方法采用 SGD 优化器微调 90 个 epoch。其他实验配置如下：初始学习率为 0.1，并在第 30 和 60 个 epoch 将学习率降为原来的 10%，SGD 中的权重衰减参数设为 1×10^{-4}。在与 CHIP 和 FPUM 的对比实验中，采用了与二者相同的学习率调整策略，具体为使用余弦策略控制学习率并微调 180 个 epoch。

4.2　基于 CIFAR-10 图像分类的剪枝模型性能分析

针对 CIFAR-10 数据集，在单支和多支结构的网络上进行了剪枝对比实验分析。实验主要涉及单支结构的 VGG-16[43] 网络，以及多支结构的 ResNet-56、ResNet-110[44] 和 DenseNet-40[45] 网络。本文剪枝方法表示为本文-r，r 为 FLOP 的缩减率。

单支结构：表 2 展示了在单支结构网络 VGG-16 上的剪枝实验结果。首先，在将参数量减少至 3.28×10^6 时，本文方法的分类准确率（93.98%）较基准模型（93.96%）略有提升，相比 L1 和 FPGM 2 种滤波器参数剪枝方法分别提高 0.58% 和 0.44%。在计算量为 158.92×10^6~195.14×10^6 时，对比 4 种基于特征图的剪枝算法 MCTS[10]、VP[21]、SSS[22] 和 DNAL[25]，本文方法在准确率和计算效率方面均保持最优。其次，随着压缩程度的加深，在参数量为 2.04×10^6~2.67×10^6 时，本文方法也全面优于 HRank、GCNP[9] 和 GAL[24] 方法。与 FPUM 相比，本文方法在参数量和准确率方面更具优势。当参数量进一步压缩至 1.54×10^6 时，本文方法仍能保持 93.62% 的准确率，表现优于 FPUM 方法。综上所述，本文方法在单分支结构网络中展现出良好的适应性，能够在不同压缩程度下保持准确率与计算效率的平衡。

表 2　基于 CIFAR-10 数据集的 VGG-16 剪枝结果

方法	准确率	参数量	计算量
VGG-16	93.96%	14.98×10^6	313.73×10^6
FPGM	93.54%	—	206.43×10^6
L1	93.40%	5.40×10^6	206.12×10^6
DNAL	93.53%	3.73×10^6	195.14×10^6
VP	93.18%	3.92×10^6	190.00×10^6
SSS	93.02%	3.93×10^6	183.13×10^6
MCTS	93.90%	—	171.00×10^6
本文-0.49	93.98%	3.28×10^6	158.92×10^6
GAL	93.42%	2.67×10^6	171.89×10^6
HRank	93.43%	2.51×10^6	145.61×10^6
FPUM	93.61%	2.27×10^6	73.09×10^6
GCNP	93.27%	2.21×10^6	134.22×10^6
本文-0.58	93.85%	2.04×10^6	133.16×10^6
本文-0.62	93.62%	1.54×10^6	119.12×10^6

多支结构：实验选择普遍应用的ResNet-56、ResNet-110和DenseNet-40这3种多支结构网络作为基准模型。在多支连接处，实验中通过联合剪枝的方式同时剪枝不同分支中的相同通道。对于DenseNet-40，实验中为了简化操作，当不同分支的特征图分辨率相同时，采用同样的通道保存比例。

表3展示了基于ResNet-56模型的剪枝结果。实验结果表明，本文方法在不同的压缩程度下均表现出优异的性能。具体而言，当参数量减少至0.66×10^6时，本文方法取得了94.08%的准确率，不仅超越了基准模型93.26%的准确率，而且在计算量接近的情况下，较L1、HRank和Sketch方法至少提升0.43%，剪枝后模型的参数量小于上述3种方法。当参数量减少至0.38×10^6时，本文方法的准确率为93.73%，仍然超过基准模型的准确率，同时显著优于其他方法。例如，与GAL方法相比，本文方法在参数量减少0.37×10^6、计算量降低4.15×10^6的情况下，仍保持更高的准确率（93.73% vs 92.98%）。当参数量减少至0.24×10^6时，本文方法和FPUM和HRank在相近的参数量和计算量情况下，表现出最优的分类性能，分类准确率较FPUM和HRank分别提升0.30个百分点和0.46个百分点。

表3 基于CIFAR-10数据集的ResNet-56剪枝结果

方法	准确率	参数量	计算量
ResNet-56	93.26%	0.85×10^6	125.49×10^6
L1	93.06%	0.73×10^6	90.90×10^6
HRank	93.52%	0.71×10^6	88.72×10^6
Sketch	93.65%	0.68×10^6	88.05×10^6
本文-0.28	94.08%	0.66×10^6	90.35×10^6
GAL	92.98%	0.75×10^6	78.30×10^6
HRank	93.17%	0.49×10^6	62.72×10^6
MCTS	93.56%	—	57.00×10^6
CP	91.80%	—	62.00×10^6
CLR-RNF	93.27%	0.38×10^6	54.00×10^6
本文-0.57	93.73%	0.38×10^6	54.45×10^6
FPUM	92.48%	0.24×10^6	34.78×10^6
HRank	92.32%	0.24×10^6	34.78×10^6
本文-0.72	92.78%	0.24×10^6	34.56×10^6
本文-0.92	90.23%	0.07×10^6	10.35×10^6

表4展示了基于ResNet-110模型的剪枝结果。在参数量为1.04×10^6的情况下，本文方法取得了94.34%的分类准确率，优于HRank（94.23%）和基准模型（93.50%），同时计算量较HRank方法降低8.16×10^6。当参数量减少至0.54×10^6时，本文方法的准确率为93.73%，仍然高于基准模型的准确率。尽管FPGM方法在准确率上的表现略优于本文方法（93.85% vs 93.73%），但其计算量为121.00×10^6，是本文方法（71.69×10^6）的1.69倍。除FPGM方法外，对比其他方法，本文方法在获得更高准确率的同时，实现了更显著的计算量和参数量缩减。例如，本文方法相比GAL方法计算量降低58.51×10^6、参数量降低0.41×10^6，同时保持更高的准确率（93.73% vs 92.55%）。

表4 基于CIFAR-10数据集的ResNet-110剪枝结果

方法	准确率	参数量	计算量
ResNet-110	93.50%	1.72×10^6	254.99×10^6
HRank	94.23%	1.04×10^6	148.70×10^6
本文-0.45	94.34%	1.04×10^6	140.54×10^6
FPGM	93.85%	—	121.00×10^6
GAL	92.55%	0.95×10^6	130.20×10^6
HRank	93.36%	0.70×10^6	105.70×10^6
Sketch	93.44%	0.69×10^6	92.84×10^6
CLR-RNF	93.71%	0.53×10^6	86.80×10^6
本文-0.72	93.73%	0.54×10^6	71.69×10^6
本文-0.90	91.25%	0.15×10^6	25.91×10^6

基于DenseNet-40模型的剪枝结果如表5所示。实验结果表明，虽然HRank方法在计算效率方面优于本文方法，但在参数量缩减和准确率的表现上本文方法更优。例如，本文方法在参数量为0.62×10^6、计算量为173.39×10^6的情况下，准确率仍高达94.44%，相比基准模型的准确率仅损失0.37%。HRank方法在参数量为0.66×10^6、计算量为167.41×10^6的情况下，准确率为94.24%。当参数量减少至0.39×10^6时，本文方法的准确率（93.71%）优于对比的GAL（93.53%）和VP（93.16%）方法，同时，在计算量和参数量的缩减方面，本文方法均优于这两种方法。上述3种多支结构模型的实验结果表明，本文方法在对多支结构网络进行模型剪枝时仍然高效。

表5　基于CIFAR-10数据集的DenseNet-40剪枝结果

方法	准确率	参数量	计算量
DenseNet-40	94.81%	1.04×10^6	282.00×10^6
HRank	94.24%	0.66×10^6	167.41×10^6
本文-0.39	94.44%	0.62×10^6	173.39×10^6
HRank	93.68%	0.48×10^6	110.15×10^6
GAL	93.53%	0.45×10^6	128.11×10^6
VP	93.16%	0.42×10^6	156.00×10^6
本文-0.60	93.71%	0.39×10^6	113.08×10^6
本文-0.88	90.54%	0.10×10^6	33.45×10^6

4.3　基于ImageNet图像分类的剪枝模型性能分析

为验证本文方法在大规模数据集（ImageNet）上的性能，本文以ResNet-50作为基准模型进行剪枝实验，实验结果如表6所示。首先，在轻度压缩实验中，本文方法在参数、计算量和准确率下降方面均优于SSS、GAL、HRank和CLR-RNF[35]方法。尽管计算量略高于MCTS和DNAL，但本文方法在Top-1和Top-5下降率上表现出明显的优势。虽然Sketch的参数量和计算量都略低于本文方法，但其Top-1下降率比本文方法高1.09%。在中等和重度

表6　基于ImageNet数据集的ResNet-50剪枝结果

方法	Top-1	Top-1 ↓	Top-5	Top-5 ↓	计算量	参数量
SSS	76.12%→74.18%	1.94%	92.86%→91.91%	0.95%	2.82×10^9	18.60×10^6
CP	74.99%→72.84%	2.15%	92.20%→90.80%	1.40%	2.73×10^9	—
Sketch	76.13%→75.22%	0.93%	92.86%→92.41%	0.45%	2.64×10^9	16.95×10^6
CLR-RNF	76.01%→74.85%	1.30%	92.96%→92.31%	0.65%	2.45×10^9	16.92×10^6
SFP	76.15%→74.61%	1.54%	92.87%→92.03%	0.84%	2.38×10^9	—
FPGM	76.15%→75.59%	0.56%	92.87%→92.63%	0.24%	2.36×10^9	—
GAL	76.15%→71.95%	4.20%	92.87%→90.04%	2.83%	2.33×10^9	21.20×10^6
HRank	76.15%→74.98%	1.17%	92.87%→92.33%	0.54%	2.30×10^9	16.15×10^6
Sketch	76.13%→74.68%	1.45%	92.86%→92.17%	0.69%	2.23×10^9	14.53×10^6
MCTS	77.34%→76.80%	0.54%	93.27%→93.00%	0.27%	2.21×10^9	—
DNAL	75.19%→74.07%	1.12%	92.56%→92.02%	0.54%	2.07×10^9	15.34×10^6
本文-0.45	76.15%→75.79%	0.36%	92.87%→92.76%	0.11%	2.26×10^9	15.09×10^6
FPGM	76.15%→74.83%	1.32%	92.87%→92.32%	0.55%	1.90×10^9	—
GDP	75.13%→71.19%	3.94%	92.30%→90.71%	1.59%	1.88×10^9	—
GAL	76.15%→71.80%	4.35%	92.87%→90.82%	2.05%	1.84×10^9	19.31×10^6
ThiNet	72.88%→71.01%	1.87%	91.14%→90.30%	0.84%	1.82×10^9	12.40×10^6
DNAL	75.19%→73.65%	1.54%	92.56%→91.74%	0.82%	1.75×10^9	12.75×10^6
TSFR	74.99%→71.45%	3.54%	92.20%→90.64%	1.56%	1.70×10^9	12.30×10^6
HRank	76.15%→71.98%	4.17%	92.87%→91.01%	1.86%	1.55×10^9	13.77×10^6
Sketch	76.13%→73.04%	3.09%	92.86%→91.18%	1.68%	1.51×10^9	10.40×10^6
本文-0.63	76.15%→74.67%	1.48%	92.87%→92.13%	0.74%	1.52×10^9	11.05×10^6
DNAL	75.19%→72.86%	2.33%	92.56%→91.34%	1.22%	1.44×10^9	10.94×10^6
GAL	76.15%→69.31%	6.84%	92.87%→89.12%	3.75%	1.11×10^9	10.21×10^6
CLR-RNF	76.01%→72.67%	3.34%	92.96%→91.09%	1.87%	1.23×10^9	9.00×10^6
ThiNet	72.88%→68.42%	4.46%	91.14%→88.30%	2.84%	1.10×10^9	8.66×10^6
HRank	76.15%→69.10%	7.05%	92.87%→89.58%	3.29%	0.98×10^9	8.27×10^6
Sketch	76.13%→69.43%	6.72%	92.86%→89.23%	3.63%	0.93×10^9	7.18×10^6
本文-0.77	76.15%→72.93%	3.22%	92.87%→91.15%	1.72%	0.95×10^9	8.02×10^6

压缩的实验中，Sketch也表现出类似的现象，其Top-1下降率分别高出本文方法1.61%和3.50%。上述结果表明，随着压缩程度的加深，本文方法相比Sketch方法的优势愈发明显。其次，中等压缩实验中的FPGM方法和重度压缩实验中的DNAL方法在计算量高出本文方法9.25%和11.92%的情况下，Top-1下降率仅减少0.16%和0.89%。此外，除GDP[23]和Sketch方法外，本文方法在中等和重度压缩实验中，多个指标均全面领先其他对比方法。

表6中本文方法的实验结果均采用微调90个epoch的方式获得，为了确保对比的公平性，在与CHIP和FPUM方法比较时，实验采用与这2种方法相同的学习率调整策略（即采用余弦退火调整学习率并微调180个epoch），实验结果如表7所示。在轻度压缩实验中，本文方法的Top-1和Top-5下降率分别为0.64%和0.32%，优于CHIP和FPUM，同时计算量也略低于CHIP和FPUM。在重度压缩实验中，CHIP、FPUM在77%的计算量和67%的参数量缩减情况下，分别实现了2.85%和2.97%的Top-1下降，相比之下，本文方法在75%的计算量和74%参数量缩减情况下，实现了仅2.49%的Top-1下降。整体实验结果表明，本文方法在提升剪枝效率的同时，在剪枝性能方面具有一定的优势。

4.4 基于语义分割的剪枝模型性能分析

为验证所提剪枝方法在不同任务中的通用性，针对语义分割任务，以DeepLabv3[46]为基础框架，采用经不同剪枝方法处理后的ResNet-50预训练模型作为骨干网络，在PASCAL VOC2012[47]数据集上评估剪枝方法性能。实验中ASPP模块参数固定为(6,12,18)，Multi Grid策略参数设为(1, 1, 1)。其他参数设计严格按照DeepLabv3官方配置，评价指标采用参数量、计算量和平均交并比（mIoU, mean intersection over union）。同时对比了L1、HRank和FPUM方法剪枝后模型的语义分割的性能，实验结果如表8所示。

表8 基于语义分割任务的模型剪枝结果

方法	mIoU	参数量	计算量
Baseline	75.03%	39.05×10^6	50.72×10^9
L1	73.49%	24.97×10^6	30.13×10^9
HRank	73.78%	24.97×10^6	30.13×10^9
FPUM	74.02%	24.54×10^6	29.53×10^9
本文-0.65	74.33%	24.49×10^6	29.32×10^9

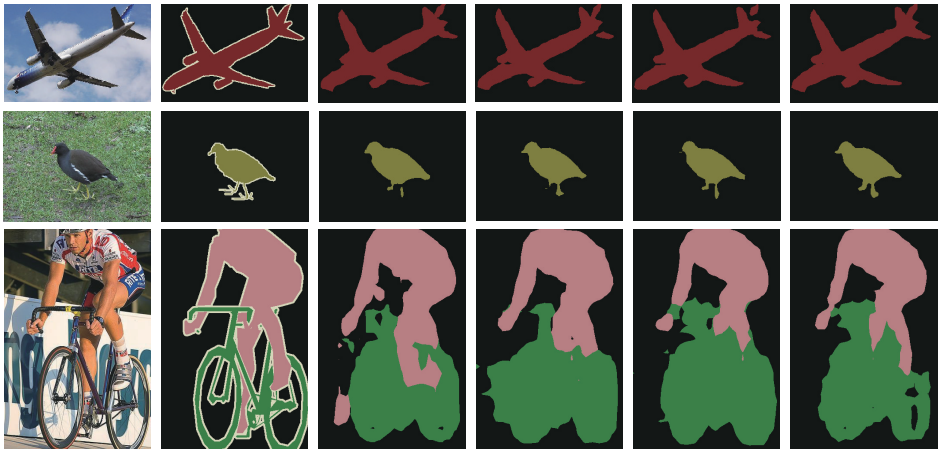
从表8可以看出，本文剪枝方法在更少的参数量和计算量下实现了更高的mIoU。同时，对不同方法剪枝后的模型性能进行了可视化分析，如图5所示，其中Ground Truth是像素级标注的真实类别区域。从图5可以看出，L1、HRank、FPUM方法的分割结果在物体细节处理上存在明显不足，如飞机轮廓不完整、鸟类形态模糊、人物与自行车边界混淆等问题。相比之下，本文方法生成的分割结果与真实标注高度吻合，不仅完整保留目标区域，还能精准还原物体边界与形态特征。在细节保留与分割精度方面，本文方法展现出显著优势，进一步验证了该剪枝方法在维持模型语义信息捕捉能力的同时，对语义分割任务具备良好的通用性。

4.5 消融实验

为了验证层间依赖特征的有效性，在CIFAR-10数据集上对ResNet-56、ResNet-110和VGG-16进行了消融实验分析。实验中，除特征重构输入数据不同外，其他实验设置均保持一致。对比使用层间依赖特征和使用层内滤波器参数指导剪枝的分类准确率。首先，为排除微调的影响，剪枝后的模型均未进行微调，并测试其分类准确率。在VGG-16

表7 基于ImageNet数据集的ResNet-50剪枝结果（微调采用余弦退火策略调整学习率）

方法	Top-1	Top-1↓	Top-5	Top-5↓	计算量	参数量
CHIP	76.15%→75.26%	0.89%	92.87%→92.53%	0.34%	1.52×10^9	11.05×10^6
FPUM	76.15%→74.80%	1.35%	92.87%→92.39%	0.48%	1.52×10^9	11.05×10^6
本文-0.66	76.15%→75.51%	0.64%	92.87%→92.55%	0.32%	1.38×10^9	11.05×10^6
CHIP	76.15%→73.30%	2.85%	92.87%→91.48%	1.39%	0.95×10^9	8.02×10^6
FPUM	76.15%→73.18%	2.97%	92.87%→91.32%	1.55%	0.95×10^9	8.02×10^6
本文-0.75	76.15%→73.66%	2.49%	92.87%→91.71%	1.16%	1.02×10^9	6.56×10^6



(a) 原图　(b) Ground Truth　(c) L1　(d) HRank　(e) FPUM　(f) 本文-0.65

图5　不同剪枝方法在语义分割任务的结果可视化对比

模型中，每层卷积层按相同比例剪枝；在ResNet-110模型中，对每个Block的第一个卷积层进行相同比例的剪枝。剪枝后的模型分类准确率如图6所示。

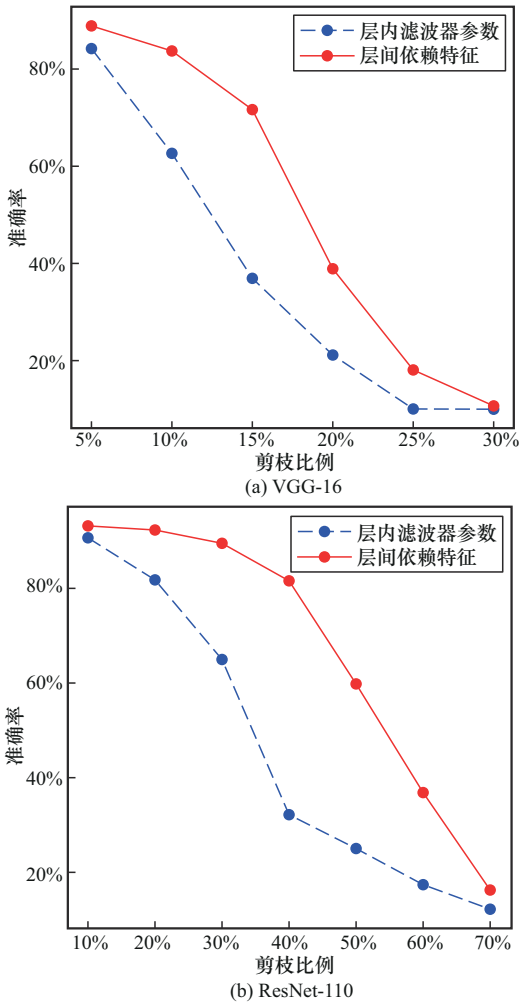


图6　针对VGG-16和ResNet-110剪枝后不微调的分类准确率

从图6可以看出，采用层间依赖特征指导的剪枝方法在不同剪枝百分比下的分类准确率均显著优于仅使用层内滤波器参数进行剪枝的方法。此外，如表9所示，微调后的性能也同样优于仅依赖层内滤波器参数的剪枝方法。上述2个实验结果均验证了层内依赖特征的有效性。同时，随着剪枝比例的提升，2种模型在剪枝不进行微调时，准确率会随剪枝程度加深而快速下降，且显著低于原始模型性能，表明了微调对恢复模型性能的必要性。

表9　基于层内滤波器参数与层间依赖特征的剪枝结果

骨干网络	层内滤波器参数剪枝准确率	层间依赖特征剪枝准确率
VGG-16	93.26%	93.98%
ResNet-56	93.21%	93.73%
ResNet-110	93.88%	94.34%

为证明层间依赖特征重构方法指导剪枝的有效性，在相同压缩率下，将其与Baseline（未剪枝）、Random（随机剪枝滤波器）、Reverse（剪枝本文方法认为重要的滤波器）对比。由图7分类准确率可知，3种模型下，各方法精度损失均不显著，表明CIFAR-10上3种模型存在滤波器冗余。此外，本文方法分类准确率最优，验证其有效性。同时Random性能高于Reverse，进一步证明本文方法在筛选重要滤波器上的有效性。

由于批归一化（BN, batch normalization）层对卷积层输出的调节作用，为了研究BN层对本文剪枝算法的影响，实验引入重参数技术，将卷积层与BN层融合为新的卷积层。基于融合后的卷积层参

数实施本文方法的剪枝操作。针对ResNet-56以及带有BN层的VGG-16模型，在CIFAR-10数据集上进行消融实验分析，结果如表10所示。实验结果表明，基于重参化的剪枝方法在CIFAR-10数据集上的准确率均有提升。为了公平地对比其他方法，突出所提剪枝方法的有效性，本文中的其他实验中仍然采用未使用重参化的剪枝结果。关于重参化技术对其他基于滤波器参数剪枝方法的有效性，尚需通过进一步的实验对比深入分析。

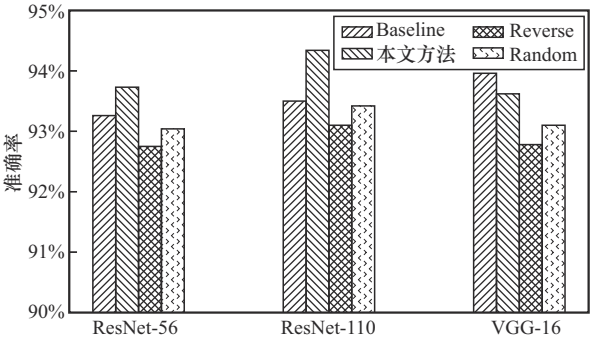


图7　针对3种模型多种方法的分类准确率

表10　重参化技术对本文剪枝方法的效果影响

Baseline	重参化	准确率	参数量	计算量
VGG-16	×	93.98%	3.28×10^6	158.92×10^6
	√	94.17%	3.28×10^6	158.92×10^6
	×	93.85%	2.04×10^6	133.16×10^6
	√	93.93%	2.04×10^6	133.16×10^6
ResNet-56	×	93.73%	0.38×10^6	54.45×10^6
	√	93.88%	0.38×10^6	54.45×10^6
	×	92.78%	0.24×10^6	34.56×10^6
	√	93.14%	0.24×10^6	34.56×10^6

5　结束语

本文提出了一种基于融合特征稀疏重构的剪枝方法，旨在解决现有滤波器剪枝方法在层间依赖关系建模上的不足。通过分析相邻卷积层滤波器参数的依赖关系，构建层间依赖特征，并将滤波器剪枝问题转换为$\ell_{2,1}$范数约束下的层间依赖特征重构问题。通过重构的稀疏特征表示，实现了对重要滤波器的筛选。本文已经证明通过重参化技术可以提升所提剪枝算法的性能，未来研究将深入探讨重参化技术对其他滤波器参数剪枝方法的通用性。

参考文献：

[1] 于舒娟, 魏玉尧, 蔡良隆, 等. 基于深度残差定点网络的太赫兹UM-MIMO系统信道估计算法[J]. 通信学报, 2025, 46(5): 77-90.
YU S J, WEI Y Y, CAI L L, et al. THz UM-MIMO system channel estimation algorithm based on deep residual block fixed-point network[J]. Journal on Communications, 2025, 46(5): 77-90.

[2] 杨凡, 杨成, 黄杰, 等. 6G密集网络中基于深度强化学习的资源分配策略[J]. 通信学报, 2023, 44(8): 215-227.
YANG F, YANG C, HUANG J, et al. Resource allocation strategy based on deep reinforcement learning in 6G dense network[J]. Journal on Communications, 2023, 44(8): 215-227.

[3] TONG S Y, YU X X, LI R P, et al. Alternate learning-based SNR-adaptive sparse semantic visual transmission[J]. IEEE Transactions on Wireless Communications, 2025, 24(2): 1737-1752.

[4] 张平, 戴金晟, 张育铭, 等. 面向语义通信的非线性变换编码[J]. 通信学报, 2023, 44(4): 1-14.
ZHANG P, DAI J C, ZHANG Y M, et al. Nonlinear transform coding for semantic communications[J]. Journal on Communications, 2023, 44(4): 1-14.

[5] 张平, 牛凯, 姚圣时, 等. 面向未来的语义通信: 基本原理与实现方法[J]. 通信学报, 2023, 44(5): 1-14.
ZHANG P, NIU K, YAO S S, et al. Semantic communications for future: basic principle and implementation methodology[J]. Journal on Communications, 2023, 44(5): 1-14.

[6] 程旗, 李捷, 高晓利, 等. 基于深度稀疏低秩分解的深度神经网络轻量化方法[J]. 控制与决策, 2023, 38(3): 751-758.
CHENG Q, LI J, GAO X L, et al. Lightweight method of deep neural network based on deep sparse low rank decomposition[J]. Control and Decision, 2023, 38(3): 751-758.

[7] GHOLAMI A, KIM S, DONG Z, et al. A survey of quantization methods for efficient neural network inference[J]. arXiv Preprint, arXiv: 2103.13630, 2021.

[8] GOU J P, YU B S, MAYBANK S J, et al. Knowledge distillation: a survey[J]. International Journal of Computer Vision, 2021, 129(6): 1789-1819.

[9] JIANG D, CAO Y, YANG Q. On the channel pruning using graph convolution network for convolutional neural network acceleration[C]//Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence. Piscataway: IEEE Press, 2022: 3107-3113.

[10] WANG Z, LI C C. Channel pruning via lookahead search guided reinforcement learning[C]//Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). Piscataway: IEEE Press, 2022: 3513-3524.

[11] HU H, PENG R, TAI Y W, et al. Network trimming: a data-driven neuron pruning approach towards efficient deep architectures[J]. arXiv Preprint, arXiv: 1607.03250, 2016.

[12] LIN M B, JI R R, WANG Y, et al. HRank: filter pruning using high-rank feature map[C]//Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 1526-1535.

[13] LUO J H, WU J. An entropy-based pruning method for cnn compression[J]. arXiv Preprint, arXiv: 1706.05791, 2017.

[14] ZHANG S, GAO M Q, NI Q, et al. Filter pruning with uniqueness mechanism in the frequency domain for efficient neural networks[J]. Neurocomputing, 2023, 530: 116-124.

[15] SUI Y, YIN M, XIE Y, et al. Chip: channel independence-based pruning for compact neural networks[J]. Advances in Neural Information Processing Systems, 2021, 34: 24604-24616.

[16] 廖威, 李光辉, 代成龙, 等. 引入余弦空间相关性的两阶段滤波器剪枝[J]. 中国图象图形学报, 2024, 29(12): 3628-3643.
LIAO W, LI G H, DAI C L, et al. Two-stage filter pruning incorporating cosinespatial correlation[J]. Journal of Image and Graphics, 2024, 29(12): 3628-3643.

[17] 施瑞文, 李光辉, 代成龙, 等. 一种基于特征导向解耦网络结构的滤波器修剪方法[J]. 计算机研究与发展, 2024, 61(7): 1836-1849.
SHI R W, LI G H, DAI C L, et al. Feature-oriented and decoupled network structure based filter pruning method[J]. Journal of Computer Research and Development, 2024, 61(7): 1836-1849.

[18] LUO J H, WU J X, LIN W Y. ThiNet: a filter level pruning method for deep neural network compression[C]//Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2017: 5068-5076.

[19] HE Y H, ZHANG X Y, SUN J. Channel pruning for accelerating very deep neural networks[C]//Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2017: 1398-1406.

[20] LIANG Y S, LIU W, YI S Y, et al. Filter pruning-based two-step feature map reconstruction[J]. Signal, Image and Video Processing, 2021, 15(7): 1555-1563.

[21] ZHAO C L, NI B B, ZHANG J, et al. Variational convolutional neural network pruning[C]//Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2019: 2775-2784.

[22] HUANG Z H, WANG N Y. Data-driven sparse structure selection for deep neural networks[C]//Proceedings of the European Conference on Computer Vision. Berlin: Springer, 2018: 317-334.

[23] LIN S H, JI R R, LI Y C, et al. Accelerating convolutional networks via global & dynamic filter pruning[C]//Proceedings of the 27th International Joint Conference on Artificial Intelligence. New York: ACM, 2018: 2425-2432.

[24] LIN S H, JI R R, YAN C Q, et al. Towards optimal structured CNN pruning via generative adversarial learning[C]//Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2019: 2785-2794.

[25] GUO Q B, WU X J, KITTLER J, et al. Differentiable neural architecture learning for efficient neural networks[J]. Pattern Recognition, 2022, 126: 108448.

[26] HAO L, ASIM K, IGOR D, et al. Pruning filters for efficient convnets[J]. arXiv Preprint, arXiv: 1608.08710, 2016.

[27] RONG J T, YU X Y, ZHANG M Y, et al. Soft Taylor pruning for accelerating deep convolutional neural networks[C]//Proceedings of the IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society. New York: ACM Press, 2020: 5343-5349.

[28] 陈锦前, 郭少勇, 刘畅, 等. 数据处理单元赋能的智算中心网络拥塞控制机制[J]. 通信学报, 2025, 46(2): 1-17.
CHEN J Q, GUO S Y, LIU C, et al. DPU empowered intelligent congestion control mechanism for the intelligent computing center network[J]. Journal on Communications, 2025, 46(2): 1-17.

[29] 刘静, 慕泽林, 赖英旭. 面向异构环境的物联网入侵检测方法[J]. 通信学报, 2024, 45(4): 114-127.
LIU J, MU Z L, LAI Y X. Intrusion detection method for IoT in heterogeneous environment[J]. Journal on Communications, 2024, 45(4): 114-127.

[30] HE Y, LIU P, WANG Z W, et al. Filter pruning via geometric Median for deep convolutional neural networks acceleration[C]//Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2019: 4335-4344.

[31] LIN M B, CAO L J, LI S J, et al. Filter sketch for network pruning[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(12): 7091-7100.

[32] KHAN N A, SAADMAN RAFAT A M. Pruning convolution neural networks using filter clustering based on normalized cross-correlation similarity[J]. Journal of Information and Telecommunication, 2025, 9(2): 190-208.

[33] WANG W, FU C, GUO J, et al. Cop: customized deep model compression via regularized correlation-based filter-level pruning[J]. arXiv Preprint, arXiv: 1906.10337, 2019.

[34] WANG Z, LI C C, WANG X Y. Convolutional neural network pruning with structural redundancy reduction[C]//Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2021: 14908-14917.

[35] LIN M B, CAO L J, ZHANG Y X, et al. Pruning networks with cross-layer ranking & k-reciprocal nearest filters[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(11): 9139-9148.

[36] WANG X R, WANG J, TANG X, et al. Filter pruning via filters similarity in consecutive layers[C]//Proceedings of the ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Piscataway: IEEE Press, 2023: 1-5.

[37] MONDAL M, DAS B, LALL B, et al. Feature independent filter pruning by successive layers analysis[J]. Computer Vision and Image Understanding, 2023, 236: 103828.

[38] ZU X, LI Y, YIN B Q. Consecutive layer collaborative filter similarity for differentiable neural network pruning[J]. Neurocomputing, 2023, 533: 35-45.

[39] HU Y M, SUN S Y, LI J Q, et al. Multi-loss-aware channel pruning of deep networks[C]//Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP). Piscataway: IEEE Press, 2019: 889-893.

[40] NIE F, HUANG H, CAI X, et al. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization[J]. Advances in Neural Information Processing Systems, 2010, 23: 1813-1821.

[41] KRIZHEVSKY A, HINTON G. Learning multiple layers of features from tiny images (technical report)[R]. 2009.

[42] RUSSAKOVSKY O, DENG J, SU H, et al. ImageNet large scale visual recognition challenge[J]. International Journal of Computer Vision, 2015, 115(3): 211-252.

[43] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. arXiv Preprint, arXiv: 1409.1556,

2014.
[44] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2016: 770-778.
[45] HUANG G, LIU Z, VAN DER MAATEN L, et al. Densely connected convolutional networks[C]//Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2017: 2261-2269.
[46] CHEN L C, PAPANDREOU G, SCHROFF F, et al. Rethinking atrous convolution for semantic image segmentation[J]. arXiv Preprint, arXiv: 1706.05587, 2017.
[47] EVERINGHAM M, ALI ESLAMI S M, VAN GOOL L, et al. The pascal visual object classes challenge: a retrospective[J]. International Journal of Computer Vision, 2015, 111(1): 98-136.

[作者简介]

杨火祥（1992– ），男，湖北荆州人，深圳大学博士生，主要研究方向为模型轻量化、低秩分解、稀疏表达。

仪双燕（1987– ），女，山东菏泽人，博士，深圳信息职业大学讲师，主要研究方向为模式识别、机器学习。

孟凡阳（1986– ），男，河南南阳人，博士，鹏城实验室副研究员，主要研究方向为信源信道编码、智能视频编码。

柳伟（1973– ），男，湖南长沙人，博士，深圳信息职业技术大学教授，主要研究方向为人工智能、视觉媒体处理。

李宗鹏（1977– ），男，加拿大人，博士，清华大学教授，主要研究方向为计算机网络优化、网络编码和网络安全。

梁永生（1971– ），男，黑龙江肇东人，博士，哈尔滨工业大学教授，主要研究方向为软硬件协同优化、信源–信道–网络联合优化编码。