视频质量增强模型加速算法

Video Quality Enhancement Model Acceleration Algorithm

杨文哲/YANG Wenzhe，徐迈/XU Mai，白琳/BAI Lin
（北京航空航天大学，中国 北京 100191）
(Beihang University, Beijing 100191, China)

摘要： 提出了一种应用于视频质量增强算法的动态结构性剪裁算法Maskcut，它可以有效提高基于深度学习的视频质量增强算法的运行速度。Maskcut是一种通用的剪裁思路，支持绝大多数的基于卷积神经网络（CNN）深度学习网络模型的剪裁加速。基于原模型中已经训练好的参数数据，Maskcut使用一种针对剪裁加速的二次训练策略来进一步微调参数，从而在保证模型有效性损失不大的同时，缩短模型运行时间。以一种先进的视频质量增强算法——多帧质量增强2.0（MFQE 2.0）为目标，Maskcut剪裁后可以快速达到峰值信噪比（PSNR）指标损失低于1%、时间缩短10%以上的加速指标。

关键词： 模型加速；图像质量增强；结构性剪裁

Abstract: Maskcut, a dynamic structural clipping algorithm for video quality enhancement is proposed, which can effectively improve the speed of video quality enhancement algorithm based on deep learning. Maskcut is a general tailoring idea that supports most of the tailoring acceleration based on the deep learning network models for convolutional neural networks (CNN). Based on the trained parameter data in the original model, the secondary training for tailoring acceleration is carried out to further fine-tune the parameters. With an advanced video quality enhancement algorithm, the multi-frame quality enhancement 2.0 (MFQE 2.0) as the goal, the peak signal-to-noise ratio (PSNR) index is less than 1% and the time is shortened by more than 10% after Maskcut clipping.

Keywords: model acceleration; image quality enhancement; structural tailoring

DOI：10.12142/ZTETJ.202101006
网络出版地址：https://kns.cnki.net/kcms/detail/34.1228.TN.20210125.1047.004.html

网络出版日期：2021-01-25
收稿日期：2020-12-25

随着多媒体及5G时代的到来，视频传输的速率和带宽得到了有效提高，人们对于高清视频的需求也变得越来越高。但是由于许多拍摄软硬件条件不高或多层多级中转压缩过程复杂等原因，使得视频质量不够清晰，因此高清视频仍有着很大的提升空间。目前，针对图像视频质量增强的算法，大多是基于深度学习的庞大计算量。这些算法在应用时往往存在参数冗余、计算量大、时间耗费多等问题，这也是近两年来深度学习领域需要着重解决的问题。目前大量的优化算法大多是基于浮点运算次数（FLOPs）的仿真工作。面对实际问题和模型，如果没有底层的硬件支持和推理加速，很多算法使用时并不能获得理想的加速效果。本文中，我们主要聚焦于实际硬件加速效果，而非理论计算量的改变。

1 视频质量增强模型加速算法的相关工作

1.1 视频质量增强

在视频质量增强方面，由GUAN Z. Y.等提出的压缩视频质量提升2.0

(MFQE 2.0)模型[1],是目前实用性相对较好的一种深度学习算法。该算法以卓越的速度和性能效果优于同时期的其他视频质量增强算法。本文中,我们以此作为剪裁算法的实践模型对象,提出了一种具有通用性的剪裁算法。在MFQE 2.0模型中,增强算法分为两个子网络,分别对应数据特征提取运动补偿(MC)子网络和数据恢复质量增强(QE)子网络。其中,数据恢复QE子网络为卷积神经网络(CNN)模型,对于剪裁算法的操作性和兼容性较高。针对不同质量(QP)的视频数据集,MFQE 2.0能够训练对应的模型,并能根据视频前后好帧的同一物体的像素信息,对当前低帧图像进行质量增强。由于MFQE 2.0训练较慢,且所采用的从训练后的大模型剪裁小模型的方法也存在合理性[2],故在MFQE 2.0训练好的模型基础上进行剪裁,可以达到更快、更好的效果。

1.2 模型剪裁加速

在剪裁加速方面,近两年来相关论文的成果众多,如HAN S.等[3]提出的剪裁-量化-编码的三部曲结构,是压缩模型比较经典的方法。本课题也是从这种压缩技巧入手,但目的是缩短模型的运行时间。本文中的剪裁方式主要为随机剪裁,即根据网络中的参数大小进行剪裁。这种方法能够减少计算量和参数数量,但剪裁结果为稀疏性矩阵,而运算时许多框架对于稀疏性矩阵的卷积(底层会转为乘法运算)并无有效加速;因此,时间上的加速效果不明显。一般做法是,底层运算采用特定的计算库,但是这种做法的通用性较差,与框架结合效果不佳。而李浩等[4]提出的基于滤波器剪裁的加速方法,以通道为单位,并以滤波器的标准差作为衡量重要性标准来进行剪裁。虽然这种思路能够在不依赖其他框架的情况下,有效缩短模型的运行时间,但依然存在优化空间,如缺乏在重训过程中的更自由的调整策略和自动调整剪裁阈值的机制。本文所提的加速算法也在此基础上进行了改进。

面对工业界亟待解决的问题,一种基于算法层面的有实际加速效果的剪裁就显得非常重要。因此,本文主要针对深度学习算法的时间耗费问题,并以一种视频质量增强算法为例,提出一种可以通过自动调整剪裁标准进行通道性剪裁的方法。同时,实验结果表明,该方法切实缩短了模型的运行时间,加速模型在工业界的落地应用。

2 Maskcut动态剪裁方法

2.1 动态通道剪裁

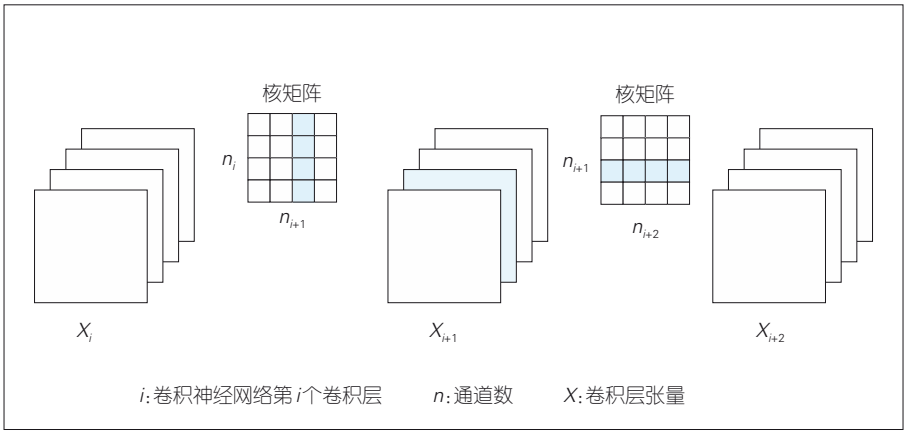
基于李浩提出的通道性剪裁的基础概念,本文的剪裁算法得以提出。在以往的通道性剪裁中(如图1所示),每一层的滤波器维度为四维,图1的每一个卷积核被视作一个质点;核矩阵为滤波器在输入通道、输出通道维度的二维表示。无论是逐层剪裁还是整体剪裁,都要先通过L1-norm或核矩阵的标准差等指标排序来确定要剪裁掉的通道,然后重训微调,提高模型准确率,并同时对应剪掉下一层的核矩阵对应通道的输出。

在剪裁后重训的过程中,滤波器的数值会经损失函数反向梯度传播而不断改变。如果我们依然按照原先的标准,那么此时可能有一些通道变得不符合规则。模型在剪裁后都是需要训练的,所以我们可以将训练过程中的剪裁做形式上的改变:不需要立刻导出小通道数模型,而是如图2所示,将对应通道的前向传递函数和反向梯度传递函数进行可控阻断,在保留被剪裁参数的同时,取得和剪裁滤波器通道一样的效果;在重训过程中,也可以随时更改开启关闭的通道。

2.2 半自动剪裁标准调整机制

通道性剪裁有着简单直接的优势,但一个比较突出的问题是每层裁剪的数量需要预先设定。无论剪裁标准是滤波器的标准差、L1-norm值,还是其他,都只是标准的差异而已,而选定的剪裁比例或阈值却没有对应的优化机制。本文提出了一种半自动剪裁标准调整机制,对于动态剪裁可以考虑选择。

以输出通道为单位,进行权重排



▲图1 传统通道性剪裁

序，记作 $list$，则各卷积层中每个输出通道对应的 L1-norm 正则化值的百分比为：$y=\frac{list(x)-\min[list]}{\max[list]-\min[list]}$。根据 y 的分布情况可首先获得分布曲线，再根据趋势来合理设定阈值或百分比，以决定哪些通道应该被剪掉[4]。通过分析此类曲线斜率的物理意义，我们可以得出这样的结论：如果某处斜率过高，那么对应一阶导数的极值点，就可得到一个由参数数据大小决定的简单阈值；我们通过拟合样条曲线后进行求导（或差分），并根据导函数极大值点或最大值点来确定对应的合适分割位置，并将其作为一种近似的剪裁阈值。

通过以上的导函数寻找极大值点，可以辅助寻找适合被剪裁的通道。如预设剪裁 $[a,b]$ 范围的通道数（百分比），然后使用寻找极大值点的方法，从 $[a,b]$ 区间内寻找导函数最大值点，以此作为当前剪裁的真正比例。通过动态剪裁的方法，可以在每轮重新训练时，不断地重新确认剪裁的比例值。

2.3 Maskcut 剪裁算法

本文使用 L1-norm 作为衡量通道重要性的指标。通过对 L1-norm 进行排序，算法将通道整体的重要度进行区分，剪裁那些不太重要的通道，并采用动态裁剪的方式随时调整选择的通道位置，或利用半自动剪裁标准调整机制，随时调整选择的通道数量。

剪裁算法的步骤具体如下：

（1）以输入通道为主，以卷积层为单位，计算各个通道的 L1-norm 并排序；

（2）设定各层的初始剪裁比例或范围；

（3）根据比例范围和半自动剪裁标准调整机制，找出此轮真正的剪裁比例；

（4）确认被剪裁的通道对应的开关关闭，对模型进行微调重训；

（5）重复步骤（3）—（4），直至满足要求，最后将开通道对应参数导出至新模型，完成剪裁。

其中，步骤（3）非必需，可根据实际情况选择是否进行。

2.4 动态剪裁实践流程

采用通道开关后，算法由原来的筛选-重训的流程变成了如图 3 所示的流程。在每轮训练中，新的流程可以更自由地重新修改训练通道。最重要的是，通过不断调整训练的通道数量，并搭配自动选择阈值分割线的机制，就可以进一步实现动态剪裁和重训，从而将两者有机结合起来。待训练效果可以接受时，再通过导出开通道的参数，原模型就可以转变为一个简单且低维的小模型，从而完成剪裁。

2.5 理论分析

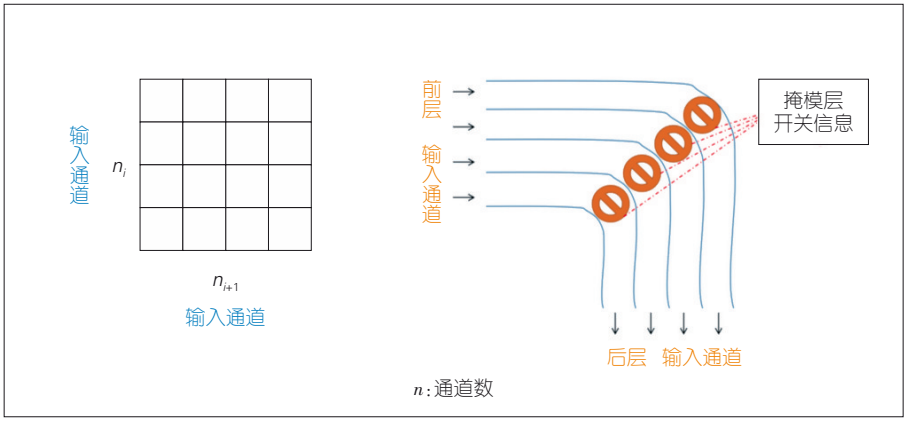
本文所提的剪裁算法的最终目的是实现时间加速，但为了保持内容完整，我们同样进行了 FLOPs 的理论分析。该算法减少的计算量是相同的[4]，以图 1 的通道为例，假设相邻两个特征图之间卷积核维度为 $k\times k\times n_i\times n_{i+1}$，输出特征图维度为 $w_{i+1}\times h_{i+1}\times n_{i+1}$，故此时 FLOPs 为：

$$FLOPs=n_{i+1}\times n_i\times k^2\times h_{i+1}\times w_{i+1}。\quad(1)$$

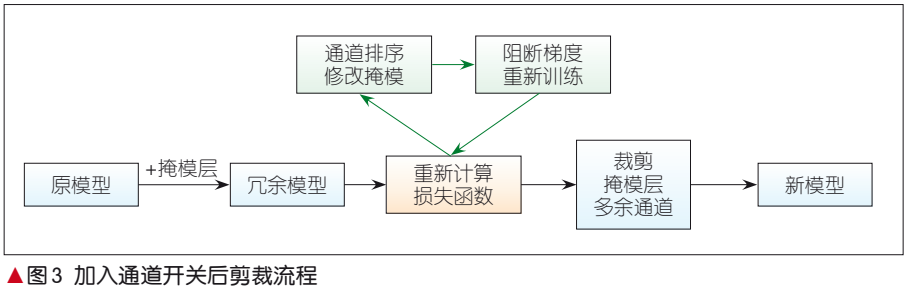
假设裁剪一个通道（如图 2 中的蓝色部分），那么减少的浮点计算量包括相邻两个卷积层的影响，总共减少了 $\left(n_i h_{i+1} w_{i+1}+n_{i+2} h_{i+2} w_{i+2}\right)k^2$ 的计算量。

3 实验结果

本文所提的基础模型是基于 Tensorflow 1.0 版本的 MFQE 2.0 模型，故我们以 Tensorflow 1.14 为运行框架环境，来测试 MFQE 2.0 的 QP32 视频



▲图 2 通道开关实现剪裁效果



▲图 3 加入通道开关后剪裁流程

数据集及模型。

3.1 MFQE 2.0模型分析

3.1.1 概述

在剪裁前，应对模型的运行时间、计算时间开支进行分析和了解。

我们利用Tensorflow内置的时间分析工具timeline进行测量。由于timeline是官方内嵌的时间测量工具，虽有一定的波动性，但相对来说可靠得多。通过timeline分析，不仅可以计算出整个模型的单次运算时间，还可以获得每个卷积层的运算时间；因此对于卷积层的通道剪裁来说，有着更明显的对比效果，故我们以此工具作为时间测量的手段。

在MFQE 2.0网络中，前面紧凑计算对应的是特征提取网络，而后面相对耗时长的部分是重建网络，因此优先剪裁的对象应是QE重建网络中的CNN那一部分。这种做法对于其他算法模型来说也有着很强的通用价值。

3.1.2 参数分析

为了进一步了解模型和参数，应对模型的参数进行分析。首先我们以层为单位，对MFQE 2.0中比较容易剪裁的QE重建网络部分中的各个卷积层的参数进行分析，并且对每个权重值的绝对值进行排序和计数，对权重值和索引比进行归一化，从而能够统计出权重的分布，具体的情况如图4所示。

这里，我们以QP=32数据集下的MFQE 2.0模型为例，提取QE子网中各卷积层的参数，并以卷积层为单位，分析权重绝对值的分布情况。图4中x为参数排序后的数组$list$索引比例，y为排序后x对应索引序号处的比例值，$y=\frac{list(x)-\min[list]}{\max[list]-\min[list]}$。我们从曲线斜率的变化可以看出：绝大多数的参数都比较小，CNN网络中参数较大的相对较少。由于参数绝对值的大小一般表征着重要性，绝对值大的权重对于计算结果影响更大。所以相对来说，应该优先剪裁那些权重较小的值。从图中也可以看出，在模型参数的分布上体现了模型目前仍然具有冗余性，并且存在可以剪裁的空间。

3.2 剪裁实验结果

我们以QP=32的视频数据为例，对MFQE 2.0的QE部分网络进行剪裁加速实验。

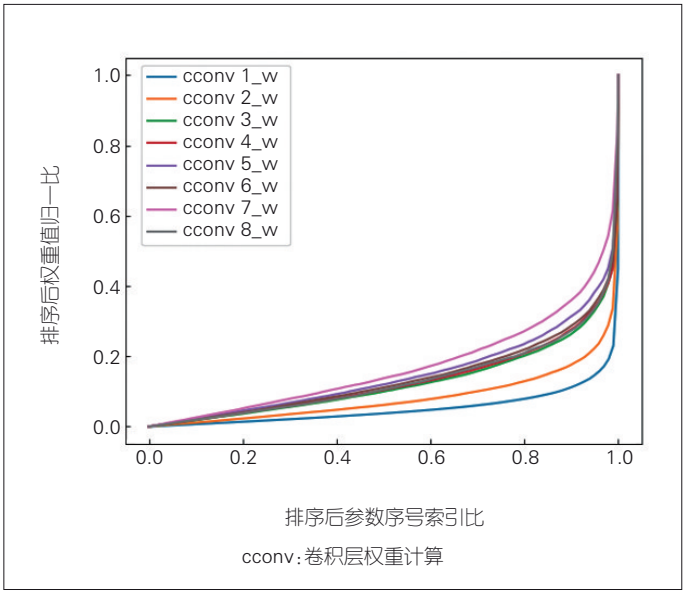
QE部分网络的原始维度主要有7层，通道数均为32，我们在此基础上进行整体剪裁加速实验。为了方便统计和测试，采取各层剪裁的通道数时刻保持一致，通道数不断调整的方法进行实验。

3.2.1 剪裁时间结果

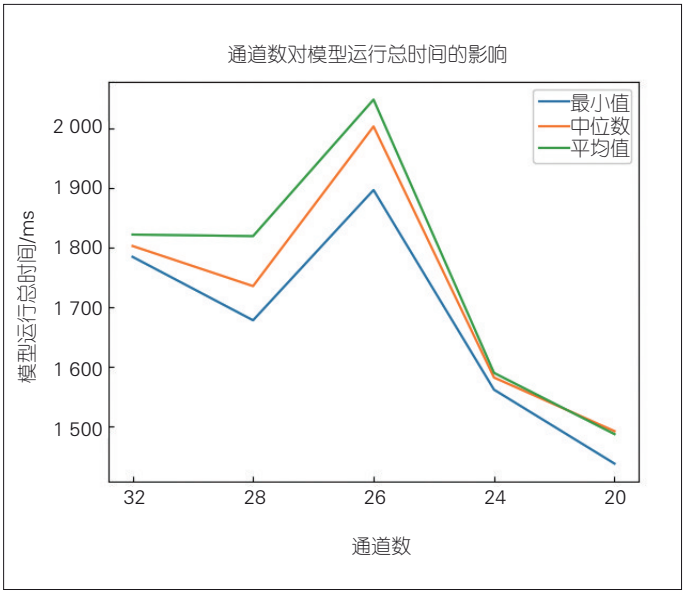
Tensorflow静态图的特点导致在模型剪裁时，只能采取稀疏+参数数据迁移的方案。而在稀疏之前，需要先对不同剪裁的具体时间加速效果进行实验，这也能给后面的稀疏实验提供优化对比。

因此，我们将原方案32通道的模型做更改，并用timeline来测试时间，经平均处理消除不确定性，其结果如图5所示。

可以看出，随着通道数的减少，模型运算的总时间也相应减少。虽然只是裁剪部分层的通道，但由于QE网络所占运行时间很长，所以相



▲图4 各层参数绝对值统计曲线(QP=32)



▲图5 模型总时间与通道数折线图

对来说只检测QE网络部分的时间加速效果依然不错。

当方案中的通道数为26时，数据平均处理后的时间仍比原模型时间更久一些。经查阅资料后得知，在计算机底层计算时，由于通道数是卷积层滤波器四维矩阵中的两个维度，因此在计算时都是按照感受野为单位进行的，可能会因维度数目和计算机底层的一些内存单元大小的匹配，存在时间长短的区分。原模型通道数仅为32，因此在通道数目减小的过程中，如果不是减少量显著，则存在时间消耗不减反增的可能，这与计算机底层内存块的空间大小等都有关系。

总的来说，通过剪裁通道数能够很方便地实现时间加速，但至于性能如何，则需要通过稀疏实验进行验证。

3.2.2 剪裁结果分析

通过动态的通道剪裁方案，算法能够在每次训练时不直接将某些参数置为0，而是通过对掩模层的“开关”进行学习，这样就能实现损失函数和反向梯度传播的对应更新，以更快速准确地实现视频质量增强的网络剪裁效果。我们对之前提到的算法和方案进行了落实，在MFQE 2.0网络中，以QP=32的数据和训练好的模型对低质量帧（non-PQF）进行实验。

如果不进行相应的参数恢复，而直接进行剪裁，那么峰值信噪比（PSNR）和结构相似性比（SSIM）的运行结果如表1所示。

可以看出，如果直接进行剪裁，会导致模型的性能变得非常差。也就是说，虽然模型存在着大量权重小的参数，但仍不能简单忽略。如果直接删除一些小权重的通道，而不对其他的通道进行适当的修改和调整，那么准确率和质量增强效果依然会大打折扣。

使用掩模层进行不断地重训后，通过调整通道开关闭合，就可以计算损失函数，阻断部分梯度反向传播的更新。

经上述训练后，我们分别测试了所得模型的时间和效果，结果如表2所示。分析PSNR对应列与直接剪裁不重训，我们可以看到模型重训对模型的性能恢复效果。

我们采用PSNR作为质量增强效果的客观标准。基于此种动态通道的剪裁方法有着较好的卷积神经网络的通用性和一般规律性，并在本文的视频质量增强任务上实现了有效的效果。

总的来看，如果主要分析ΔPSNR和总时间的平衡，分析结果如表3所示。在目前的实验结果中，在通道数为24时，能够实现以0.7%的性能损失换取12.5%的总时间缩短，达到预期指标。

▼表1 直接剪裁峰值信噪比结果对比

模型	原模型32通道	28通道	24通道
PSNR	0.305	0.108	0.007
损失比/%	—	64.6	97.7

PSNR：峰值信噪比

▼表2 重训后峰值信噪比结果对比

通道数	ΔPSNR	损失比/%
32	0.305	—
28	0.291	4.6
24	0.303	0.7
20	0.230	24.6

PSNR：峰值信噪比

▼表3 重训后SSIM与增强每帧总时间指标情况

通道数	ΔSSIM	损失比	总时间	加速比/%
32	0.011	—	1 783.9	—
28	0.009	18.2	1 677.4	6.0
24	0.009	18.2	1 560.9	12.5
20	0.008	27.3	1 436.8	19.5

SSIM：结构相似性比

4 结束语

本文提出了一种动态的通道剪裁方法，以L1-norm作为衡量滤波器通道重要程度的标准，通过动态剪裁、统一设定剪裁比例，对已经训练好的MFQE 2.0模型进行剪裁加速。在MFQE 2.0的QP 32数据集中，通过微调模型后，我们发现：当将QE网络的32通道数剪裁至24时，可以达到以0.7%的性能损失换取12.5%的总时间缩短的指标。在剪裁后网络最终导出之前，随着迭代次数的推进，模型剪裁效果会更好，而且具有一定外扩性，即最后的模型再适当增大通道时，由于动态剪裁的选通机制，可以相对更轻松微调至其他数目通道的模型。通过动态通道剪裁，完成了MFQE 2.0的视频质量增强的模型有效加速。

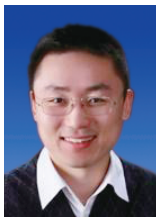
本文还提出一种半自动剪裁标准调整策略，通过拟合函数的一阶导数极大值或最大值寻找，辅助决定剪裁最佳比例，后续应尝试从结果反馈信息或二分类辅助自动决定剪裁比例。

参考文献
[1] GUAN Z Y, XING Q L, XU M, et al. MFQE 2.0: a new approach for multi-frame quality enhancement on compressed video [J]. IEEE transactions on pattern analysis and machine intelligence, 2019: 1. DOI: 10.1109/tpami.2019.2944806
[2] ZHU M, GUPTA S. To prune, or not to prune: exploring the efficacy of pruning for model compression [EB/OL]. (2018-06-23) [2020-12-22]. http://arxiv.org/abs/1710.01878
[3] HAN S, MAO H Z, DALLY W J. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding [EB/OL]. [2020-12-22]. https://arxiv.org/abs/1510.00149
[4] 李浩, 赵文杰, 韩波. 基于滤波器裁剪的卷积神经网络加速算法 [J]. 浙江大学学报(工学版), 2019, 53(10): 1994-2002
[5] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient ConvNets [EB/OL]. [2020-12-22]. https://arxiv.org/abs/1608.08710
[6] LEBEDEV V, LEMPITSKY V. Fast ConvNets using group-wise brain damage [C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, 2016: 2554-2564. DOI: 10.1109/cvpr.2016.280
[7] WEN W, WU C P, WANG Y D, et al. Learning structured sparsity in deep neural networks [EB/OL]. [2020-12-22]. https://arxiv.org/abs/1608.03665
[8] HOWARD A G, ZHU M L, CHEN B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications [EB/OL]. [2020-12-22]. https://arxiv.org/abs/1704.04861
[9] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient ConvNets [EB/OL]. [2020-12-24]. https://arxiv.org/abs/1608.08710
[10] POLYAK A, WOLF L. Channel-level acceleration of deep face representations [J]. IEEE access, 2015, 3: 2163-2175. DOI: 10.1109/access.2015.2494536

作 者 简 介

杨文哲，北京航空航天大学电子信息工程学院在读硕士研究生；研究方向包括深度学习、图像压缩等。

徐迈，北京航空航天大学电子信息工程学院教授、教育部“青年长江学者”、中国图象图形学学会青工委副主任；研究方向包括图像处理、视频压缩、视频通信、计算机视觉与人工智能等；2016年获教育部霍英东青年基金资助，2017年获人工智能学会技术发明一等奖(第二完成人)，2018年获教育部科技进步一等奖、中国电子学会优秀科技工作者，2019年获国家优秀青年基金资助，2020年获北京市杰出青年基金资助；发表论文100余篇。



白琳，北京航空航天大学网络空间安全学院教授；主要研究方向包括通信网络安全、无线通信、物联网、无人机通信等领域等；主持国家自然科学基金项目3项、国家重点研发计划项目子课题1项，获国家自然科学基金优秀青年科学基金资助，获第四届中国出版政府奖、中国电子学会自然科学二等奖、国家科技进步二等奖；发表SCI期刊文章66篇，著有英文专著2部、中文专著3部。