幽灵网络：低成本行动的更多特性

韩凯[1] 王云和[1] 田琪[1*] 郭建源[2] 徐春晶[1] 徐昌[31] 华为技术有限公司诺亚方舟实验室 [2] 北京大学。
([3]) 悉尼大学工程学院计算机科学系。
{kai.han,yunhe.wang,tian.qi1,xuchunjing}@huawei.comjyguo@pku.edu.cn c.xu@sydney.edu.au

摘要

在嵌入式设备上部署卷积神经网络（CNN）面临内存和计算资源受限的挑战。特征图中的冗余性是成功CNN的重要特征，但在神经网络架构设计中鲜少被研究。本文提出一种新型幽灵模块，通过低成本运算生成更多特征图。 基于一组内在特征图，我们通过低成本线性变换序列生成大量幽灵特征图，这些图能充分揭示内在特征背后的信息。该幽灵模块可作为即插即用组件升级现有卷积神经网络。通过设计幽灵瓶颈层堆叠幽灵模块，即可轻松构建轻量级幽灵网络。 基准测试表明，该幽灵模块可作为基线模型卷积层的卓越替代方案，在ImageNet ILSVRC-2012分类数据集上，幽灵网络以接近MobileNetV3的计算成本实现了更优识别性能（如75.7%的top-1准确率）。代码详见：https://github.com/huawei-noah/ghostnet。

1. 引言

深度卷积神经网络在多种计算机视觉任务中表现优异，例如图像识别[30, 13]、目标检测[43, 33]和语义分割[4]。 传统卷积神经网络通常需要大量参数和浮点运算（FLOPs）才能达到满意精度，例如ResNet-50[16]拥有约2560万个参数，处理224×224图像需消耗41亿次浮点运算。
处理224×224尺寸图像。因此，深度神经网络设计的最新趋势是探索便携且
高效的网络架构，以满足移动设备（如智能手机和自动驾驶汽车）的性能需求。

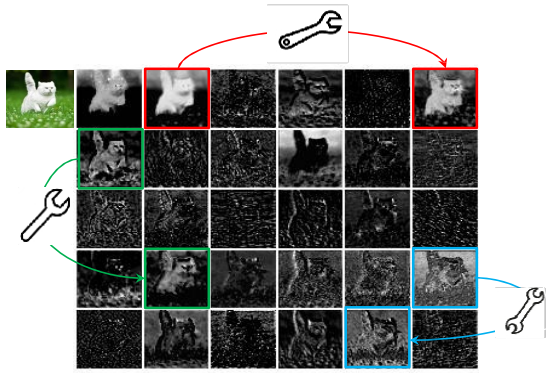


图1. ResNet-50模型中首个残差组生成的特征图可视化（）。图中标注了三组相似特征图对，每组特征图均用相同颜色框标注。通过廉价操作（标记为spanners）对其中一张特征图进行变换，即可近似获得另一张特征图。

多年来，人们提出了一系列方法来研究紧凑型深度神经网络，例如网络剪枝[14, 39]、低位量化[42, 26]、知识蒸馏[19, 57]等。Han等人[14]提出通过剪枝去除神经网络中不重要的权重。 [31]利用l_1范数正则化对卷积神经网络的滤波器进行修剪。
[42]通过将权重与激活值量化为1位数据实现大幅压缩与加速。[19]引入知识蒸馏技术，实现从大型模型向小型模型的知识迁移。然而这些方法的性能常受限于预训练深度神经网络基准的上限。

除上述方法外，高效神经网络架构设计在构建参数更少、计算量更低的高效深度网络方面具有巨大潜力，并已取得显著进展。此类方法还能为自动搜索算法提供新的搜索单元[62, 55, 5]。 例如，MobileNet[21,44,20]利用深度卷积与点卷积构建了替代大尺寸卷积滤波器的单元，实现了相当的性能表现。ShuffleNet[61,40]则进一步探索了通道洗牌操作

[*] 通信作者

以提升轻量化模型的性能。

训练有素的深度神经网络特征图中充沛甚至冗余的信息，往往能确保对输入数据的全面理解。例如图1展示了ResNet-50生成的输入图像部分特征图，其中存在大量相似特征图对，如同彼此的*幽灵*。特征图中的冗余性可能是深度神经网络成功的关键特征。与其规避冗余特征图，我们更倾向于以成本效益的方式加以利用。

本文提出一种新型幽灵模块，旨在通过减少参数数量生成更多特征。具体而言，将深度神经网络中的普通卷积层拆分为两部分：第一部分执行常规卷积操作，但严格控制卷积核总数；基于第一部分生成的内在特征图，通过一系列简单线性运算生成更多特征图。 相较于传统卷积神经网络，该幽灵模块在保持输出特征图尺寸不变的前提下，显著降低了整体参数需求与计算复杂度。基于幽灵模块，我们构建了高效神经网络架构——幽灵网（GhostNet）。首先通过替换基准神经网络架构中的原始卷积层验证幽灵模块的有效性，继而在多个基准视觉数据集上验证幽灵网的优越性。 实验结果表明：所提出的Ghost模块能在保持相似识别性能的前提下降低通用卷积层的计算成本；GhostNet在各类任务中不仅超越MobileNetV3[20]等前沿高效深度模型，更能在移动设备上实现快速推理。

本文其余部分组织结构如下：第2节简要总结该领域相关工作，第3节提出Ghost模块及GhostNet，第4节进行实验与分析，最后在第5节给出结论。

2. 相关工作

本文将从模型压缩与紧凑模型设计两个维度，重新审视现有神经网络轻量化方法。

2.1. 模型压缩

针对给定神经网络，模型压缩旨在降低计算、能耗及存储成本[14, 48, 11, 54]。 连接剪枝[15, 14, 50]通过剔除神经元间非关键连接实现轻量化。通道剪枝[51, 18, 31, 39, 59, 23, 35]进一步聚焦于移除冗余通道，以提升实际加速效果。模型量化[42, 24, 26]通过对权重或激活值进行量化处理实现轻量化。

在神经网络中，通过采用离散值实现压缩与计算加速。具体而言，仅使用1位值的二值化方法[24, 42, 38, 45]可通过高效二进制运算极大加速模型运行。张量分解[27, 9]则利用权重中的冗余性与低秩特性，有效减少参数数量或计算量。 知识蒸馏[19,12,3]通过让大型模型"教导"小型模型来提升后者性能。这些方法的效果通常依赖于预训练模型的质量，而基础运算与架构的改进将推动其发展。

2.2. 紧凑模型设计

随着神经网络在嵌入式设备上的部署需求，近年来提出了一系列紧凑模型[7, 21, 44, 20, 61, 40, 53, 56]。Xception[7]利用深度卷积操作更高效地利用模型参数。 MobileNets[21]是基于深度可分离卷积的轻量级深度神经网络系列。 MobileNetV2[44]提出反向残差块，MobileNetV3[20]进一步运用AutoML技术[62,55,10]，在减少浮点运算量的同时提升性能。 ShuffleNet[61]引入通道洗牌操作以增强通道组间信息流动交换。ShuffleNetV2[40]进一步考虑目标硬件的实际运行速度进行紧凑模型设计。尽管这些模型以极低浮点运算量获得卓越性能，但特征图间的关联性与冗余性始终未被充分挖掘。

3. 方法论

在本节中，我们将首先介绍Ghost模块，通过使用几个小型滤波器从原始卷积层生成更多特征图，然后开发出具有极高效能架构和高性能的新型GhostNet。

3.1. Ghost模块实现特征扩展

深度卷积神经网络[30,46,16]通常包含大量卷积操作，导致计算成本高昂。尽管近期研究如MobileNet[21,44]和ShuffleNet[40]引入了深度卷积或洗牌操作，通过缩小卷积滤波器（浮点运算）构建高效卷积神经网络，但剩余的1×1卷积层仍会占用大量内存和浮点运算资源。

运算），但剩余的1×1卷积层仍会占用大量内存和浮点运算资源。

鉴于主流卷积神经网络计算的中间特征图普遍存在冗余（如图1所示），我们提出通过减少生成这些特征图所需的卷积滤波器数量来降低资源消耗。具体而言，

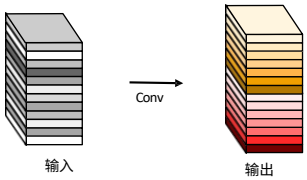
给定输入数据 $X \in \mathrm{R}^{c\times h\times w}$ ，其中 c 为输入通道数，h 和 w 分别为输入数据的高度和宽度

，任意卷积层生成n张特征图的运算可表示为：

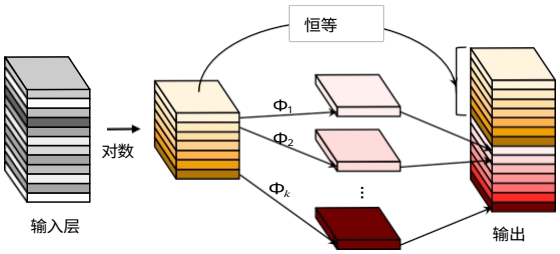
$$Y = X * f + b, \tag{1}$$

其中 $*$ 表示卷积运算，b 为偏置项，$Y \in \mathbb{R}^{h' \times w' \times n}$ 表示具有n通道的输出特征图，其中 $f \in \mathbb{R}^{c \times k \times k \times n}$ 为本层卷积滤波器。在此外，h' 和 w' 分别表示输出数据的高度和宽度，$k \times k$ 则是卷积滤波器 f 的核尺寸。

分别。在此卷积过程中，所需浮点运算次数可计算为 $n \cdot h' \cdot w' \cdot c \cdot k \cdot k$，由于滤波器数量 n 和通道数 c 通常非常大（例如256或512）。



(a) 卷积层。



(b) 幽灵模块。

图2. 卷积层与所提Ghost模块的示意图，用于输出相同数量的特征图。Φ代表低成本操作。

根据式(1)，待优化的参数数量（f和b中）由输入与输出特征图的维度明确决定。如图1所示，卷积层的输出特征图常包含大量冗余信息，其中部分特征图彼此相似。 我们指出，无需耗费大量浮点运算和参数逐一生成这些冗余特征图。假设输出特征图是少数内在特征图经廉价变换后的"幽灵"。这些内在特征图通常尺寸较小，由普通卷积滤波器生成。具体而言，m个内在特征图 $Y' \in \mathbb{R}^{h' \times w' \times m}$ 通过主卷积生成：

$$Y' = X * f, \tag{2}$$

其中 $f \in \mathbb{R}^{c \times k \times k \times m}$ 为所用滤波器，$m \le n$，为简化起见省略偏置项。超参数例如滤波器尺寸、步长、填充参数均与普通卷积（式（1））保持一致，以确保输出特征图的空间尺寸（即h'和w'）保持一致。为进一步获得所需的n张特征图，我们建议对Y'中的每条内在特征应用一系列低成本线性运算，根据下列公式生成s张虚特征：

函数：

$$y_{ij} = \Phi_{i,j}(y'_i), \quad \forall\, i = 1, \ldots, m, j = 1, \ldots, s, \tag{3}$$

其中y'_i是Y'中的第i个内在特征图，$\Phi_{i,j}$在生成第 j 个鬼特征图 y_{ij}的第 j 个（除最后一个外）线性运算，该也就是说，y'_i 可能包含一个或多个鬼特征图 $\{y_{ij}\}_{j=1}^{s}$. 最后一个$\Phi_{i,s}$是预处理阶段的恒等映射

如图2(b)所示，输出内在特征图。通过利用式(3)，可获得$n = m \cdot s$个特征图$Y = [y_{11}, y_{12}, \cdots, y_{ms}]$作为幽灵调制器的输出数据（如图2(b)所示）。注意线性运算Φ

注意，线性运算Φ

对每个通道进行操作，其计算成本远低于普通卷积。实际应用中，单个Ghost模块可能包含多种线性运算，例如3×3和5×5线性核，将在实验部分进行分析。

与现有方法的差异。所提出的Ghost模块与现有高效卷积方案存在显著差异：i)相较于文献[21, 61]中的单元广泛采用1×1点卷积，Ghost模块的主卷积可支持自定义核尺寸尺寸。ii) 现有方法[21,44,61,40]采用点卷积处理跨通道特征，再通过深度卷积处理空间信息。 而幽灵模块采用普通卷积处理首先生成若干内在特征图，随后利用低成本线性运算增强特征并增加通道数。iii) 在先前的高效架构中[21, 61, 53, 28]，处理每张特征图的操作仅限于深度卷积或移位操作，而Ghost模块中的线性操作可具有多样性。iv) 此外，Ghost模块中通过将恒等映射与线性变换并行处理来保留内在特征图。

复杂度分析。由于可利用式(3)中提出的幽灵模块生成与普通卷积层相同数量的特征图，我们能轻松将其集成到现有成熟神经网络架构中以降低计算成本。本文进一步分析采用幽灵模块在内存使用和理论加速率方面的收益。例如，存在1个恒等映射和$m \cdot (s-1) = (n) \cdot (s-1)$个非恒等映射。

例如，存在1个恒等映射和$m \cdot (s-1) = \frac{n}{s} \cdot (s-1)$线性运算，且每项线性运算的平均核尺寸等于$d \times d$。理想情况下，$n \cdot (s-1)$项线性运算可具有不同形状和参数，但在线推理将受到阻碍，尤其考虑到

CPU或GPU卡的实用性。因此，我们建议在单个Ghost模块中采用相同尺寸的线性运算（如3×3或5×5），以实现高效实施。理论上通过Ghost模块升级普通卷积的理论加速比为

$$r_s = \frac{n \cdot h' \cdot w' \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot h' \cdot w' \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot h' \cdot w' \cdot d \cdot d} = \frac{c \cdot k \cdot k}{\frac{1}{s} \cdot c \cdot k \cdot k + \frac{s-1}{s} \cdot d \cdot d} \approx \frac{s \cdot c}{s + c - 1} \approx s, \tag{4}$$

其中 $d \times d$ 与 $k \times k$ 具有相近的量级，且 $s \ll c$。同样地，压缩比可计算为

$$r_c = \frac{n \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot d \cdot d} \approx \frac{s \cdot c}{s + c - 1} \approx s, \tag{5}$$

这相当于利用所提出的Ghost模块所获得的加速比。

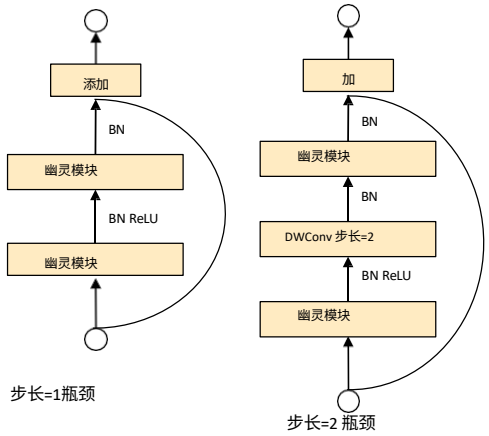


图3. Ghost瓶颈结构。左：步长=1的Ghost瓶颈；右：步长=2的Ghost瓶颈。

3.2. 构建高效卷积神经网络

幽灵瓶颈。基于幽灵模块的优势，我们提出专为小型卷积神经网络设计的幽灵瓶颈（G-bneck）。如图3所示，幽灵瓶颈与ResNet[16]中的基础残差块相似，整合了多个卷积层和短路连接。 所提出的幽灵瓶颈主要由两个堆叠的幽灵模块构成。首个幽灵模块作为扩展层增加通道数量，我们将输出通道数与输入通道数的比值称为*扩展比*。第二个幽灵模块则缩减通道数量以匹配短路路径。 随后在两个幽灵模块的输入与输出之间建立捷径连接。除遵循MobileNetV2[44]建议在第二个幽灵模块后不使用ReLU外，其余各层均应用批量归一化(BN)[25]与ReLU非线性映射。

表1. GhostNet整体架构。G-bneck表示幽灵瓶颈层。#exp表示扩展层规模。#out表示输出通道数。运算符表示是否使用SE模块。输出

输入	运算符	#exp	#out	SE	步长
$224^2 \times 3$	Conv2d 3×3	-	16	-	2
$112^2 \times 16$	G-bneck	16	16	-	1
$112^2 \times 16$	G-bneck	48	24	-	2
$56^2 \times 24$	G-bneck	72	24	-	1
$56^2 \times 24$	G-bneck	72	40	1	2
$28^2 \times 40$	G-bneck	120	40	1	1
$28^2 \times 40$	G-bneck	240	80	-	2
$14^2 \times 80$	G-bneck	200	80	-	1
$14^2 \times 80$	G-bneck	184	80	-	1
$14^2 \times 80$	G-bneck	184	80	-	1
$14^2 \times 80$	G-bneck	480	112	1	1
$14^2 \times 112$	G-bneck	672	112	1	1
$14^2 \times 112$	G-bneck	672	160	1	2
$7^2 \times 160$	G-bneck	960	160	-	1
$7^2 \times 160$	G-bneck	960	160	1	1
$7^2 \times 160$	G-bneck	960	160	-	1
$7^2 \times 160$	G-bneck	960	160	1	1
$7^2 \times 160$	Conv2d 1×1	-	960	-	1
$7^2 \times 960$	平均池 7×7	-	-	-	-
$1^2 \times 960$	Conv2d 1×1	-	1280	-	1
$1^2 \times 1280$	FC	-	1000	-	-

上述幽灵瓶颈针对步长=1的情况。当步长=2时，捷径路径通过下采样层实现，并在两个幽灵模块之间插入步长=2的深度卷积层。实际应用中，幽灵模块的主卷积采用点卷积以提升效率。

幽灵网络。 基于幽灵瓶颈结构，我们提出

如表7所示的GhostNet架构。我们基本沿用了MobileNetV3[20]的架构，因其卓越性能，并将MobileNetV3中的瓶颈模块替换为我们设计的Ghost瓶颈模块。GhostNet主要由多层Ghost瓶颈模块堆叠构成，以Ghost模块为基本单元。 第一层为标准卷积层（含16个滤波器），随后依次排列多组通道数逐步增加的Ghost瓶颈模块。这些Ghost瓶颈模块根据输入特征图尺寸划分为不同阶段。 除各阶段末层步长设为2外，其余幽灵瓶颈层均采用步长=1。最终通过全局平均池化与卷积层将特征图转换为1280维特征向量进行分类。 部分幽灵瓶颈层的残差连接处还应用了挤压激发（SE）模块[22]（详见表7）。与MobileNetV3不同，我们未采用硬锯齿非线性函数，因其存在较大延迟问题。本文提出的架构提供基础设计参考，后续可通过幽灵模块进行超参数调优或自动架构搜索。

进一步提升性能。

宽度乘数。尽管表7所示架构已能提供低延迟和保证精度，但在某些场景下我们可能需要更小更快模型或特定任务的更高精度。为定制网络以满足需求，只需在每层通道数上均匀乘以*因子*α。该*因子*α称为*宽度乘数*，因其可改变整个网络的宽度。我们将宽度乘数α的GhostNet称为GhostNet-α×。宽度乘数可二次控制模型规模与计算成本。

为α的GhostNet称为GhostNet-α×。宽度乘数通过

通常*α值*较小时延更低但性能下降，反之亦然。

4. 实验验证

在本节中，我们首先用提出的Ghost模块替换原始卷积层以验证其有效性。随后，基于新模块构建的GhostNet架构将在图像分类和目标检测基准测试中进行进一步验证。

数据集与设置。为验证所提Ghost模块及GhostNet架构的有效性，我们在多个基准视觉数据集上开展实验，包括CIFAR-10[29]、ImageNet ILSVRC 2012数据集[8]以及MS COCO目标检测基准[34]。

CIFAR-10数据集用于分析所提方法的特性，该集包含10类共60,000张32×32彩色图像，其中训练图像50,000张，10,000张测试图像。采用包含随机裁剪和镜像翻转的通用数据增强方案[16,18]。ImageNet作为大规模图像数据集，包含超过*包含120*万张训练图像和*5万*张验证图像，共涵盖1000个类别。训练过程中采用包含随机裁剪和翻转的通用数据预处理策略[16]。我们还基于MS COCO数据集[34]开展了目标检测实验。 遵循常规做法[32, 33]，我们在COCO *trainval35k*数据集（包含*80K*训练图像与验证集随机抽取的*35K*子集）上训练模型，并在包含*5K*图像的*minival*数据集上进行评估。

4.1. 幽灵模块效率

4.1.1 简易实验。

图1示意图揭示了某些相似特征图对可通过高效线性运算生成。我们首先通过玩具实验观察原始特征图与生成的幽灵特征图之间的重建误差。以图1中三组特征对（红色、绿色、蓝色）为例，采用ResNet-50模型[16]的首个残差模块。 将左侧特征作为输入，右侧特征作为输出，我们采用小型深度卷积滤波器学习映射关系，即两者间的线性操作Φ。卷积滤波器*尺寸d*取值范围为1至7，不同*d值*对应各特征对的均方误差(MSE)值见表2。

表2. 均方误*差与*不同核尺寸*关系*。

MSE (10^{-3})	d=1	d=3	d=5	d=7
红色对	4.0	3.3	3.3	3.2
绿色对	25.0	24.3	24.1	23.9
蓝色对	12.1	11.2	11.1	11.0

从表2可以看出，所有均方误差值都极小，这表明深度神经网络中的特征图之间存在强相关性，这些冗余特征图可由若干内在特征图生成。除上述实验中使用的卷积操作外，我们还可探索其他低成本线性运算来构建Ghost模块，例如仿射变换和小波变换。 然而卷积作为当前硬件已充分支持的高效运算，可覆盖平滑、模糊、运动*等多种*常用线性操作。此外，尽管可针对线性操作Φ学习每个滤波器的尺寸，但不规则模块将降低计算单元（如CPU和GPU）的效率。 因此，我们建议在后续实验中将幽灵模块中的*d*设为固定值，并采用深度卷积实现式(3)，以构建高效的深度神经网络。

表3. 不同*d值*下Ghost模块在CIFAR-10数据集上的性能表现。

d	权重 (M)	浮点运算次数 (M)	准确率（%）
VGG-16	15.0	313	93.6
1	7.6	157	93.5
3	7.7	158	93.7
5	7.7	160	93.4
7	7.7	163	93.1

表 4. 不同 *s* 值下，在 CIFAR-10 数据集上所提 Ghost 模块的性能表现。

s	权重 (M)	浮点运算次数 (M)	准确率 (%)
VGG-16	15.0	313	93.6
2	7.7	158	93.7
3	5.2	107	93.4
4	4.0	80	93.0
5	3.3	65	92.9

4.1.2 CIFAR-10。

我们在两个流行的网络架构上评估了提出的Ghost模块，即VGG-16[46]和ResNet-56[16]。

在CIFAR-10数据集上。由于VGG-16最初是为ImageNet设计的，我们采用其变体[60]（该变体在文献中被广泛使用）进行后续实验。这两个模型中的所有卷积层均被替换为提出的Ghost模块，新模型分别命名为Ghost-VGG-16和Ghost-ResNet-56。 训练策略严格遵循[16]中的设置，包括动量、学习*率等*参数。我们首先分析Ghost模块中两个超参数*s和d*的影响，随后将Ghost模型与当前最先进方法进行对比。

超参数分析。如式(3)所述，高效深度神经网络的Ghost模块包含两个超参数：用于*生成*m=
n/s 个本征特征图，以及用于计算幽灵特征图的线性运算核尺寸 *d* × *d*（即深度卷积滤波器的尺寸）。
计算鬼特征图。在VGG-16架构上测试了这两个参数的影响。

首先*固定*s=2，在{1,3,5,7}范围内*调整d*，并将CIFAR-10验证集结果列于表3。可见
*当*d=3时，所提出的Ghost模块表现优于较小或较大的d值。这是因为当d=3时，卷积核的尺寸恰好能有效捕捉深度方向特征，同时避免过度分散注意力。
尺寸为1×1的核无法在特征图中引入空间信息，而更大的核（如d=5或d=7）则会导致过拟合并增加计算量。因此，为兼顾效果与效率，后续实验中我们采用d=3的核尺寸。
导致过拟合和更多计算。因此，在后续实验中我们采用*d*=3以兼顾效果与效率。

在研究了提案Ghost模块中使用的核尺寸后，我们保持d=3，并将其他超*参数*s在{2, 3, 4, 5}范围内进行调优。事实上，s值直接影响最终网络的计算成本，
也就是说，更大的*s*值会带来更高的压缩率和加速比，正如式(5)和式(4)所分析的那样。从表4的结果可见，当我们增加*s时*，浮点运算次数显著减少，而准确率逐渐下降，这与预期一致。
预期。尤其当s=2（即对VGG-16模型进行2倍压缩）时，本方法的性能甚至略优于原始模型，这表明所提出的幽灵模块具有显著优势。
所提出的Ghost模块的优势。

表5. CIFAR-10数据集上VGG-16与ResNet-56压缩方法的最新技术对比。-表示未见相关报道结果。

模型	权重	浮点运算次数	准确率（%）
VGG-16	15M	313M	93.6
l1-VGG-16 [31, 37]	540万	206M	93.4
SBP-VGG-16 [18]	-	1.36亿	92.5
幽灵VGG-16 (s=2)	770万	1.58亿	**93.7**
ResNet-56	0.85M	1.25亿	93.0
CP-ResNet-56 [18]	-	63M	92.0
l1-ResNet-56 [31, 37]	0.73M	91M	92.5
AMC-ResNet-56 [17]	-	63M	91.9
幽灵ResNet-56 (s=2)	0.43M	63M	**92.7**

与最先进模型的比较。我们在VGG-16和ResNet-56两种架构上，将Ghost-Net与若干具有代表性的最先进模型进行了对比。 对比方法涵盖不同类型的模型压缩技术，*包括*l_1剪枝[31, 37]、SBP[18]、通道剪枝(CP)[18]及AMC[17]。在VGG-16架构下，我们的模型在实现2倍加速的同时，准确率略高于原始模型
2倍加速，表明VGG模型存在显著冗余。我们的Ghost-VGG-16（s=2）
在显著降低浮点运算量的同时，以93.7%的最高性能超越竞争对手。对于体量远小于VGG-16的ResNet-56，我们的模型
在速度提升2倍的同时达到与基线相当的准确率。我们还发现，其他计算成本相当或更高的尖端模型
具有相似或更高计算成本的其他尖端模型，其准确率均低于我们的模型。

特征图可视化。我们还可视化了幽灵模块的特征图，如图4所示。尽管生成的特征图源自原始特征图，但它们存在显著差异，这表明生成的特征具有足够的灵活性以满足特定任务的需求。

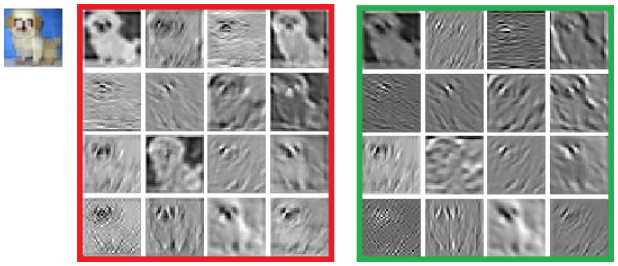


图4. Ghost-VGG-16模型第二层特征图。左上图为输入图像，左侧红色框内为原始卷积层特征图，右侧绿色框内为深度变换后的特征图。

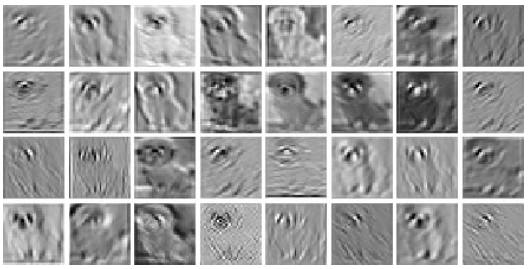


图5. 基础VGG-16模型第二层的特征图。

4.1.3 ImageNet上的大型模型

随后我们将Ghost模块嵌入标准ResNet-50[16]模型，并在大规模ImageNet数据集上开展实验。ResNet-50约含2560万参数，
4.1B FLOPs，前5错误率为7.8%。我们采用Ghost

表6. ImageNet数据集上ResNet-50压缩的尖端方法对比。

模型	权重（M）	浮点运算次数 (B)	Top-1 准确率 (%)	Top-5 准确率 (%)
ResNet-50 [16]	25.6	4.1	75.3	92.2
Thinet-ResNet-50 [39]	16.9	2.6	72.1	90.3
NISP-ResNet-50-B [59]	14.4	2.3	-	90.8
多功能ResNet-50 [49]	11.0	3.0	74.5	91.8
SSS-ResNet-50 [23]	-	2.8	74.2	91.9
幽灵ResNet-50 (s=2)	13.0	2.2	**75.0**	**92.3**
Shift-ResNet-50 [53]	6.0	-	70.6	90.1
Taylor-FO-BN-ResNet-50 [41]	7.9	1.3	71.7	-
可压缩ResNet-50 *0.5倍* [58]	6.9	1.1	72.1	-
MetaPruning-ResNet-50 [36]	-	1.0	73.4	-
幽灵-ResNet-50 (s=4)	6.5	1.2	**74.1**	**91.9**

模块替换ResNet-50中的所有卷积层以获得紧凑模型，并将结果与几种最先进方法进行比较，详见表6。为确保公平比较，训练设置（如优化器、学习率和批量大小）完全与[16]保持一致。

从表6结果可见，我们的Ghost-ResNet-50（s=2）在保持与原始ResNet-50相同精度的同时，实现了约2倍的加速率和压缩率。

原始ResNet-50的准确率。与包括Thinet[39]、NISP[59]、多功能滤波器[49]和稀疏结构选择(SSS)[23]在内的最新方法相比，本方法在2倍加速设置。当我们将s进一步增至4时，基于幽灵模型的准确率仅下降0.3%，同时实现了

计算速度提升约4倍。相比之下，采用相似权重或浮点运算量的比较方法[53, 58]

低于我们的模型。

4.2. GhostNet在视觉基准测试中的表现

在验证了所提Ghost模块高效生成特征图的优势后，我们分别采用图像分类和目标检测任务评估了表7所示的精心设计的GhostNet架构（采用Ghost瓶颈层）。

4.2.1 ImageNet分类任务

为验证所提GhostNet的优越性，我们在ImageNet分类任务上开展实验。除初始学习率设定为0.4（批量大小为1,024，使用8块GPU）外，其余训练设置均遵循[61]中的配置。 所有结果均基于ImageNet验证集的单裁剪Top-1性能报告。为简化实现，GhostNet在主卷积层设置核尺寸k=1，所有Ghost模块中s=2且d=3。

选取若干现代小型网络架构作为对比对象，包括MobileNet系列[21, 44, 20]，

ShuffleNet系列[61, 40]、ProxylessNAS[2]、FBNet[52]、MnasNet[47]等模型。结果汇总于表7。

模型按计算复杂度分为三类，通常适用于移动应用场景：约50、约150及200-300 MFLOPs。结果表明，在这些小型网络中，更高的浮点运算能力通常能提升准确率，这印证了其有效性。我们的Ghost-Net在不同计算复杂度层级中持续超越其他竞争者，因其能更高效地利用计算资源生成特征图。

在这些小型网络中，浮点运算能力越强通常意味着精度越高，这证明了它们的有效性。我们的Ghost-Net在不同计算复杂度级别上始终优于其他竞争者，因为GhostNet在生成特征图时能更高效地利用计算资源。

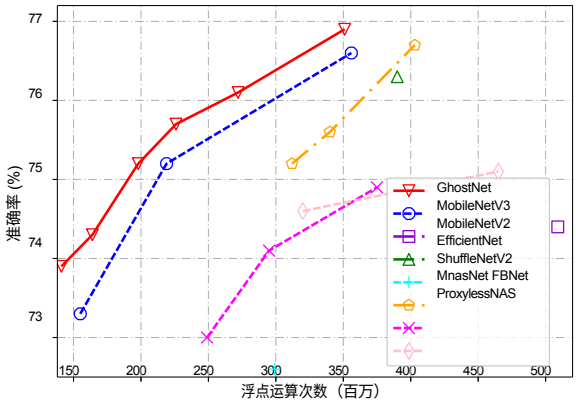


图6. ImageNet数据集上Top-1准确率与FLOPs*的关系*。

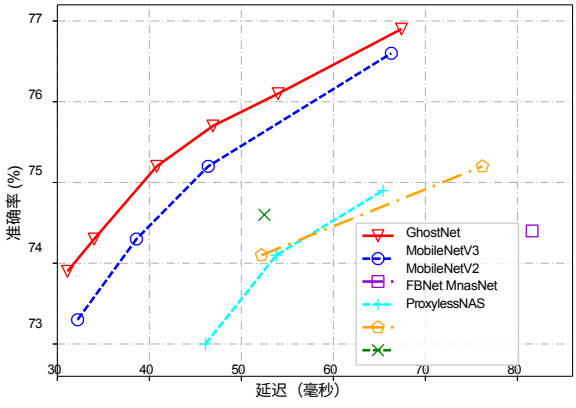


图7. 在ImageNet数据集上的Top-1准确率*与*延迟*对比*。

表7. 在ImageNet数据集上，对最先进小型网络在分类准确率、权重数量和浮点运算次数方面的比较。

模型	权重（百万）	浮点运算次数（百万）	前1名命中率（%）	Top-5 准确率（%）
ShuffleNetV1 0.5× (g=8) [61]	1.0	40	58.8	81.0
MobileNetV2 0.35× [44]	1.7	59	60.3	82.9
ShuffleNetV2 0.5× [40]	1.4	41	61.1	82.6
MobileNetV3 Small 0.75× [20]	2.4	44	65.4	-
幽灵网 0.5×	2.6	42	**66.2**	**86.6**
MobileNetV1 0.5× [21]	1.3	150	63.3	84.9
MobileNetV2 0.6× [44, 40]	2.2	141	66.7	-
ShuffleNetV1 1.0× (g=3) [61]	1.9	138	67.8	87.7
ShuffleNetV2 1.0× [40]	2.3	146	69.4	88.9
MobileNetV3 Large 0.75× [20]	4.0	155	73.3	-
幽灵网 1.0×	5.2	141	**73.9**	**91.4**
MobileNetV2 1.0× [44]	3.5	300	71.8	91.0
ShuffleNetV2 1.5× [40]	3.5	299	72.6	90.6
FE-Net 1.0× [6]	3.7	301	72.9	-
FBNet-B [52]	4.5	295	74.1	-
无代理NAS [2]	4.1	320	74.6	92.2
MnasNet-A1 [47]	3.9	312	75.2	92.5
MobileNetV3 Large 1.0× [20]	5.4	219	75.2	-
幽灵网 1.3×	7.3	226	**75.7**	**92.7**

实际推理速度。由于提出的GhostNet专为移动应用设计，我们进一步使用TFLite工具[1]在基于ARM的移动设备上测量了GhostNet的实际推理速度。 遵循[21, 44]中的通用设置，我们采用单线程模式且批量大小为1。从图7的结果可见，在相同延迟下，GhostNet的top-1准确率比MobileNetV3高出约0.5%，且GhostNet以更短的运行时间实现了相近性能。 例如，准确率为75.0%的GhostNet仅需40毫秒延迟，而准确率相近的MobileNetV3处理单张图像则需约45毫秒。总体而言，我们的模型普遍优于知名尖端模型，即MobileNet系列[21, 44, 20]。ProxylessNAS [2]、FBNet [52] 和 MnasNet [47]。

4.2.2 目标检测

为进一步评估GhostNet的泛化能力，我们在MS COCO数据集上开展目标检测实验。*采用trainval35k*数据集作为训练数据，并参照[32, 33]*在minival*数据集上报告平均精度均值(mAP)结果。 实验框架采用两阶段Faster R-CNN（集成特征金字塔网络FPN）[43, 32]与单阶段RetinaNet[33]，其中GhostNet作为骨干特征提取器的直接替代方案。所有模型均基于ImageNet预训练权重，采用[32,33]建议的超参数，通过SGD训练12个 epoch。输入图像尺寸调整为短边800像素，长边不超过

表8展示了检测结果，其中浮点运算次数（FLOPs）采用行业惯例以224×224图像为基准计算。GhostNet在计算成本显著降低的同时在单阶段RetinaNet和双阶段Faster R-CNN框架中均实现了与MobileNetV2和MobileNetV3相当的mAP，无论采用单阶段RetinaNet还是双阶段Faster R-CNN框架。

表8. GhostNet在MS COCO数据集上的 结果。

主干网络	检测框架	主干网络FLOPs	mAP
MobileNetV2 1.0× [44]	视网膜网络	3亿	26.7%
MobileNetV3 1.0× [20]		2.19亿	26.4%
GhostNet 1.1×		1.64亿	26.6%
MobileNetV2 1.0× [44]	Faster R-CNN	300M	27.5%
MobileNetV3 1.0× [20]		2.19亿	26.9%
幽灵网络 1.1×		1.64亿	26.9%

5. 结论

为降低近期深度神经网络的计算成本，本文提出一种新型幽灵模块以构建高效神经架构。基础幽灵模块将原始卷积层拆分为两部分，利用更少滤波器生成若干内在特征图，随后通过若干低成本变换操作高效生成幽灵特征图。 在基准模型和数据集上的实验表明，该方法作为即插即用模块，能在保持可比性能的前提下将原始模型转换为紧凑模型。此外，基于该新模块构建的GhostNet在效率和准确性方面均超越了当前最先进的可移植神经网络架构。

致谢

感谢匿名审稿人的宝贵意见。徐畅的研究获得澳大利亚研究理事会项目DE180101438资助。

参考文献

[1] 马丁·阿巴迪、阿什什·阿加瓦尔、保罗·巴勒姆、尤金·布雷夫多、陈志锋、克雷格·西特罗、格雷格·S·科拉多、安迪·戴维斯、杰弗里·迪恩、马蒂厄·德文、桑杰·格玛瓦特、伊恩·古德费洛、安德鲁·哈普、杰弗里·欧文、迈克尔·伊萨德、 贾阳青、拉法尔·约瑟夫维奇、卢卡什·凯泽、曼朱纳特·库德勒、乔什·莱文伯格、丹德利昂·曼尼、拉贾特·蒙加、雪莉·摩尔、德里克·默里、克里斯·奥拉、迈克·舒斯特、乔纳森·施伦斯、布诺·施泰纳、伊利亚·苏茨克弗、库纳尔·塔尔瓦尔、保罗·塔克、文森特·范霍克、维杰·瓦苏德万、 费尔南达·维加斯、奥里奥尔·维尼亚尔斯、皮特·沃登、马丁·瓦滕伯格、马丁·维克、于元、郑晓强。TensorFlow：异构系统上的大规模机器学习，2015。软件 可从tensorflow.org获取。

[2] 蔡汉、朱立刚、韩松。ProxylessNAS：基于目标任务与硬件的直接神经架构搜索。收录于 *ICLR*，2019年。

[3] 陈汉庭、王云和、徐畅、杨昭辉、刘传健、石博鑫、徐春静、徐超、田启 . 无数据学生网络学习. *发表于ICCV*, 2019.

[4] 陈亮杰、George Papandreou、 Iasonas Kokkinos、 Kevin Murphy、 Alan L Yuille。基于深度卷积网络与全连接CRF的语义图像分割。*发表于ICLR* 2016。

[5] 陈武阳、龚新宇、刘先明、张倩、李远、王张阳。Fasterseg：更快实时语义分割的卷积神经网络搜索。发表于*ICLR*，2020年。

[6] 陈伟杰、谢迪、张远、蒲世亮。仅需少量位移：设计高效卷积神经网络实现图像分类。发表于*CVPR*，2019年。

[7] 弗朗索瓦·肖莱特. Xception：基于深度可分离卷积的深度学习. 发表于*CVPR会议*论文集，第1251–1258页，2017年.

[8] 邓嘉、董伟、Richard Socher、李立佳、李凯、李飞飞。Imagenet：大规模分层图像数据库。 发表 *于CVPR*，第248–255页。IEEE，2009年。

[9] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann Le-Cun, Rob Fergus. 基于卷积网络内线性结构的高效评估方法. *发表于NeurIPS*, , 2014年, 第1269–1277页.

[10] 龚新宇、常世宇、姜一凡、王张阳. Autogan：生成对抗网络的神经架构搜索. 发表于*ICCV*, 2019.

[11] 桂树鹏、王浩涛、杨海川、于晨、王张阳、刘吉. 兼具对抗鲁棒性的模型压缩：统一优化框架. 发表于 *NeurIPS*, 2019.

[12] 韩博, 姚全明, 于兴瑞, 牛刚, 徐淼, 胡伟华, 曾伊沃, 杉山正志. 协同教学：基于极端噪声标签的深度神经网络鲁棒训练. *发表于NeurIPS*, 2018.

[13] 韩凯、郭建远、张超、朱明健。用于精细表示学习的属性感知注意力模型。*发表于ACM MM*，2018年。

[14] 宋汉、毛慧姿、William J Dally. 深度压缩：基于剪枝、训练量化与霍夫曼编码的深度神经网络压缩. 发表 *于ICLR*, 2016.

[15] 宋汉、Jeff Pool、John Tran、William Dally. 同时学习权重与连接的高效神经网络. 发表于 *NeurIPS*, 1135–1143页, 2015.

[16] 何凯明、张向宇、任绍清、孙健。图像识别中的深度残差学习。发表于*CVPR*， 770–778页，2016年。

[17] 何一辉, 林吉, 刘志坚, 王翰瑞, 李立佳, 韩松. Amc: 移动设备上基于自动机器学习的模型压缩与加速 . *发表于ECCV*, 2018.

[18] 何一辉、张翔宇、孙健。通道修剪加速超深神经网络。*发表于ICCV*，2017年。

[19] Geoffrey Hinton, Oriol Vinyals, Jeff Dean. 神经网络知识蒸馏. *arXiv 预印本 arXiv:1503.02531*, 2015.

[20] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan 等. 《移动卷积神经网络v3的搜索》. *发表于ICCV*, 2019.

[21] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam. 移动神经网络：面向移动视觉应用的高效卷积神经网络. *arXiv 预印本 arXiv:1704.04861*, 2017.

[22] 胡杰、沈莉、孙刚。挤压与激励 网络。*CVPR*会议论文，2018年。

[23] 黄泽浩与王乃彦. 深度神经网络的数据驱动稀疏结构选择. 收录于 *ECCV*会议论文集，第304–320页， 2018.

[24] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 《二值化神经网络》. 收录于 *NeurIPS*, 第4107–4115页, 2016年.

[25] Sergey Ioffe 和 Christian Szegedy. 《批量归一化：通过减少内部协变量偏移加速深度网络训练》. 发表于 *ICML*, 2015.

[26] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, Dmitry Kalenichenko. 神经网络量化与训练实现高效整数运算推理. *发表于CVPR*, , 2704–2713页, 2018.

[27] Max Jaderberg, Andrea Vedaldi, Andrew Zisserman. 基于低秩 展开加速卷积神经网络. *发表于BMVC*, 2014.

[28] 全允浩与金俊模. 通过卷积分解构建快速网络 . 发表于*NeurIPS*, 2018.

[29] 亚历克斯·克里泽夫斯基与杰弗里·辛顿。从微型图像中学习多层特征。技术报告，Citeseer， 2009年。

[30] 亚历克斯·克里泽夫斯基、伊利亚·苏茨克弗、杰弗里·E·辛顿。基于深度卷积神经网络的Imagenet分类。 载于*NeurIPS*，第1097–1105页，2012年。

[31] 李昊、阿西姆·卡达夫、伊戈尔·杜尔达诺维奇、哈南·萨梅特、汉斯·彼得·格拉夫。高效卷积神经网络的修剪滤波器。发表于 *ICLR*，2017年。

[32] 林宗毅、Piotr Dolla´r、Ross Girshick、何凯明、Bharath Hariharan、Serge Belongie. 基于特征金字塔网络的物体检测. *发表于CVPR*, 2017.

[33] 林宗毅、普里亚·戈亚尔、罗斯·吉尔希克、何凯明、皮奥特·多拉尔。用于密集目标检测的焦点损失。*发表于ICCV*，2017。

[34] 林宗毅、Michael Maire、Serge Belongie、James Hays、Pietro Perona、Deva Ramanan、Piotr Dolla´r、C Lawrence Zitnick.《Microsoft COCO：上下文中的常见物体》. 收录于 *ECCV*. Springer出版社，2014年.

[35] 刘传健、王云和、韩凯、徐春静、徐畅。基于实例稀疏性学习加速 深度模型。*发表于IJCAI*，2019年。

[36] 刘泽春、穆昊远、张向宇、郭子超、杨鑫、Tim Kwang-Ting Cheng、孙健。元修剪：用于自动神经网络通道修剪的元学习。发表于 *ICCV* 2019。

[37] 刘壮，孙明杰，周廷辉，黄高，Trevor Darrell. 重新审视网络剪枝的价值. 发表于 *ICLR*, 2019.

[38] 刘泽春，吴宝源，罗文涵，杨欣，刘伟，程光廷。双实数网络：通过增强表示能力和先进训练算法提升1位卷积神经网络性能。*发表于ECCV*，2018年。

[39] 罗建浩、吴建新、林伟尧。Thinet：深度神经网络压缩的滤波器层级剪枝方法。 发表于*ICCV*，第5058–5066页，2017年。

[40] 马宁宁、张翔宇、郑海涛、孙健。Shufflenet v2：高效卷积神经网络架构设计的实用指南。*发表于ECCV*，2018年。

[41] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, Jan Kautz. 神经网络剪枝的重要性估计. *发表于CVPR*, 2019.

[42] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, Ali Farhadi. Xnor-net：基于二进制卷积神经网络的ImageNet分类. 收录于*ECCV会议*论文集，第525–542页. Springer出版社，2016年.

[43] 任少清、何凯明、罗斯·吉尔希克、孙健。Faster R-CNN：基于区域提案网络的实时目标检测。发表于*NeurIPS*，2015年。

[44] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zh-moginov, Liang-Chieh Chen. Mobilenetv2：反向残差与线性瓶颈网络. 发表于*CVPR*会议，第4510–4520页， 出版社，2018年.

[45] 申明珠、韩凯、徐春静、王云和. 寻找精确的二元神经网络架构. *发表于ICCV 研讨会*, 2019.

[46] K. Simonyan 与 A. Zisserman. 用于大规模图像识别的超深卷积神经网络. 发表于 *ICLR*, 2015.

[47] 谭明兴、陈博、庞若明、Vijay Vasudevan、Mark Sandler、Andrew Howard、Quoc V Le. Mnasnet：面向移动设备的平台感知神经架构搜索. 发表于 CVPR，页码2820–2828，2019年.

[48] 王悦、江子宇、陈晓涵、徐鹏飞、赵阳、林颖彦、王张阳。E2-train：实现80%以上节能的尖端卷积神经网络训练方案。发表于*NeurIPS*，2019年。

[49] 王云和、徐畅、徐春静、徐超、陶大成. 学习多功能滤波器以实现高效卷积神经网络. 发表于*NeurIPS*, 2018.

[50] 王云和、徐畅、尤山、陶大成、徐超. Cnnpack：卷积神经网络在频域中的压缩. *发表于*NeurIPS，第253–261页，2016年.

[51] 文伟、吴春鹏、王艳丹、陈怡然、李海。深度神经网络中结构化稀疏性的学习。发表于 *NeurIPS*，第2074–2082页，2016年。

[52] 吴碧辰, 戴晓亮, 张培昭, 王阳涵, 孙飞, 吴一鸣, 田远东, Peter Vajda, 贾阳青, Kurt Keutzer. Fbnet: 基于可微神经架构搜索的硬件感知高效卷积神经网络设计. 发表于 *CVPR*, 页码10734–10742, 2019.

[53] 吴碧辰、万阿尔文、岳翔宇、金彼得、赵思诚、诺亚·戈尔曼特、阿米尔·戈拉米内贾德、约瑟夫·冈萨雷斯、库尔特·科伊策尔。《Shift：空间卷积的零浮点运算、零参数替代方案》。*发表于CVPR*，2018年。

[54] 徐一兴, 王云和, 陈翰廷, 韩凯, 徐春静, 陶大成, 徐畅. 基于正样本-无标签数据的云端压缩. *发表于NeurIPS*, 2019.

[55] 杨昭辉、王云和、陈星浩、石博新、徐超、徐春静、田奇、徐畅。CARS：高效神经网络架构搜索的连续进化算法。*arXiv 预印本 arXiv:1909*.04977，2019年。

[56] 杨昭辉、王云和、刘传健、陈翰廷、徐春静、石博新、徐超、徐畅. Legonet：基于乐高滤波器的高效卷积神经网络. 发表于 *ICML*, 2019.

[57] 尤山、徐畅、徐超、陶大成. 基于多教师网络的学习式深度学习（）. *发表于SIGKDD*, 2017.

[58] 于佳辉，杨林杰，徐宁，杨建超，黄托马斯· 。可压缩神经网络。发表于*ICLR*，2019年。

[59] 于瑞琪、李昂、陈春富、赖瑞欣、Vlad I Morariu、韩新彤、高明飞、林清勇、Larry S Davis。Nisp：基于神经元重要性评分传播的网络剪枝。发表*于CVPR*，2018年。

[60] Sergey Zagoruyko. Torch实现CIFAR-10分类率达92.45%，2015. *URL http://torch.ch/blog/2015/07/30/cifar.html*.

[61] 张翔宇、周欣宇、林梦晓、孙健。ShuffleNet：面向移动设备的超高效率卷积神经网络。*CVPR*，2018。

[62] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, Quoc V Le. 可扩展图像识别中可迁移架构的学习. *收录于CVPR*, 第8697–8710页, 2018.