基于快速下采样的轻量化网络设计方法及人脸识别应用

王佳皓,徐树公,陆恒杰
(上海大学通信与信息工程学院,上海 200444)

摘 要: 高精度卷积神经网络推理成本往往较高,很难在资源受限的嵌入式设备上进行实时推理. 本文通过分析不同类型卷积对模型推理速度的影响因素,首次指出除了模型计算量,模型的特征图输出量也是影响推理速度的一个关键因素. 而现有基于深度分离卷积的轻量化方法仅把模型的计算量作为模型轻量化指标,并未考虑特征图输出量对模型推理速度的影响. 根据该发现,本文结合标准卷积提出一种基于快速下采样的模型轻量化加速方法,通过快速减少特征图尺寸来同时减少模型计算量和特征图输出量. 本文方法设计的轻量化模型的特征提取能力和不同平台的推理速度均优于现有的基于深度分离卷积的轻量化方法. 更进一步地,本文利用该方法针对人脸识别任务提出一个快速人脸识别模型FDFaceNet. 与现有的轻量化人脸识别模型相比,FDFaceNet准确率更高,在不同平台上的推理速度更快.

关键词: 轻量化网络模型设计;神经网络加速;轻量化人脸识别;人脸检测识别系统;嵌入式设备
基金项目: 国家自然科学基金(No.61871262)
中图分类号: TP391.4 **文献标识码:** A **文章编号:** 0372-2112(2023)08-2226-12
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20211031

Lightweight Network Design and Application for Face Recognition Based on Fast Down-Sampling

WANG Jia-hao, XU Shu-gong, LU Heng-jie
(*School of Communication & Information Engineering*, *Shanghai University*, *Shanghai* 200444, *China*)

Abstract: High-precision convolutional neural networks often come with high inference costs, making it difficult to perform real-time inference on resource-constrained embedded devices. We analyze the factors that influence the speed of model inference by different types of convolutions, and for the first time point out that in addition to the computational complexity of the model, the feature map throughput of the model is also a key factor affecting the inference speed. However, the existing lightweight methods based on the depth-wise separation convolution only use computational complexity as the model lightweight metric, not considering the influence of the feature map throughput on the model inference speed. Based on this discovery, we propose a model lightweight acceleration design method combined with standard convolution based on fast down-sampling module, which could reduce the computational complexity and feature map throughput of the model at the same time by rapidly reducing the size of the feature map. The performance and the inference speed on different platforms of the models designed by proposed method are better than the existing lightweight methods based on depth-wise separation convolution. Further, we utilize this method to propose a fast face recognition model FDFaceNet (Fast Down-sampling FaceNet) for face recognition tasks. Compared with the existing lightweight face recognition models, FDFaceNet has higher accuracy and faster inference speed on various platforms.

Key words: lightweight neural network design; neural network acceleration; lightweight face recognition; face detection and recognition system; embedded devices
Foundation Item(s): National Natural Science Foundation of China (No.61871262)

收稿日期:2021-08-01;修回日期:2022-01-14;责任编辑:王天慧

1 引言

卷积神经网络(Convolutional Neural Networks, CNN)已经成为深度学习领域的重要方法之一,在目标识别、目标检测以及实例分割等任务上都取得了很好的效果. 为了获得更高的模型准确度,传统的卷积神经网络往往需要更多的参数量和计算量. 而近些年在边缘设备上部署深度学习应用需求不断上升,边缘设备(如移动端或嵌入式平台)的存储空间和计算资源往往受到限制,因此在这些设备上部署较大的卷积模型是不实际的. 另外,为了实现更好的用户体验,一些嵌入式的深度学习应用(如人脸检测、人脸识别等)对模型还有实时性等要求.

为适应移动端深度学习模型应用,过去几年业内对模型压缩与加速的研究热度不断提升. 其中,主流方法包括模型剪枝、模型量化、知识蒸馏以及轻量化模型设计等. 按照压缩后模型加速效果分类,能将这些方法分为前端压缩和后端压缩两类,如表1所示. 其中,前端压缩不会改变模型权重的整体性,如通道剪枝[1]、知识蒸馏[2]等,因此在推理时使用模型压缩前的运行库就能在目标硬件上实现模型加速的效果;而后端压缩的方法为了追求极致的压缩比会极大地改变模型权重的整体结构,如非结构化剪枝[3]、模型量化[4]等. 因此,后端压缩方法需要开发配套运行库和硬件设备,才能实现理想的压缩和加速效果.

以上模型压缩加速方法中,轻量化网络设计的方法在设计高效模型方面具有很大的潜力,如Mobilenet[5,6]、Shufflenet[7]等,在神经网络搜索方面也获得了相当大的成功,如Efficientnet[8]. 不同于上述其他模型压缩方法均需要对预训练的模型进行压缩操作,轻量化网络设计是从头设计参数量和计算量较小的高效模型,推理时无须开发特殊的运行库和硬件设备就能实现加速效果.

在现有轻量化网络模型设计方法中,分组卷积和深度分离卷积已经成为轻量化模型设计的主要方法之一[9,10],并且大部分模型轻量化工作都把模型的参数量、计算量作为主要的轻量化指标[11,12]. 设计轻量化模型的初衷是希望模型能够在资源受限的移动或嵌入式设备上部署,并且提高模型的推理速度. 然而根据实验,本文发现除了计算量,模型特征图输出量也是影响推理速度的一个关键因素. 单个计算量指标并不能准确地反映模型的推理速度.

本文对不同卷积类型进行复杂度分析,并从模型计算量和特征图输出量两个角度研究了不同卷积在各个平台上对模型推理速度的影响,首次指出模型特征图输出量也是影响推理速度的一个关键因素,而不是一般认为的模型推理速度仅与模型的计算量有关. 同时,考虑计算代价和读写代价,本文给出了深度模型的推理速度关于嵌入式硬件算力与带宽和模型计算量与特征图输出量模型的关系式. 现有的基于深度分离卷积的轻量化方法仅考虑模型计算量指标,但会大幅增加模型特征图输出量,这对于模型在GPU平台的推理速度并不友好,并且会大幅减少模型在CPU平台上的每秒平均浮点计算数,降低计算效率.

表1　模型压缩与加速方法分类

分类	压缩方法	介绍	缺点
前端压缩	知识蒸馏	大型教师网络训练小型学生网络	需要设计蒸馏训练策略、精度损失
	通道剪枝	通过设计卷积核重要性策略,移除模型中冗余的通道	需要设计通道剪枝策略,需要微调网络、精度损失
	轻量化网络设计	手工设计方法:利用分组卷积或深度分离卷积设计轻量模型,增加模型层数弥补特征提取能力	模型深度增加导致模型输出量增多,占用内存增加
		网络结构搜索:在给定的网络搜索空间中用某种搜索策略得到更优的网络结构	模型结构不规则
后端压缩	模型量化	训练时量化:训练时采用低比特量化,需要设计配套的低比特训练方法	训练时量化需要设计配套训练方法;训练后量化模型损失精度,需要配套的运行库
		训练后量化:使用配套的运行库可以加速模型的推理.与FP32的模型相比,量化会导致精度损失	
	权重剪枝	权重剪枝的剪枝自由度比结构化剪枝更高,剪枝后模型的精度损失更小,但需要配套的运行库和硬件设备才能有效地加速网络	精度损失,需要配套的运行库

由此,本文提出一种新的基于快速下采样模块并结合标准卷积的模型轻量化加速方法,首次将模型特征图输出量和计算量同时作为模型轻量化的指标. 在不使用深度分离卷积和分组卷积的情况下,该方法能同时减少模型计算量与特征图输出量,并且达到模型推理加速的效果. 本文通过实验在图像分类和人脸识别任务上证明了该方法的特征提取能力和推理速度更优于基于深度分离卷积的轻量化模型.

针对人脸识别应用,近年来的研究采用了不同的模型压缩与轻量化方法来减小识别模型的大小,加快人脸识别速度. 如ProxylessFaceNAS[13]将MobilenetV2[6]作为结构搜索空间针对CPU进行网络结构搜索

并获得了高效人脸识别模型. Lightfacenet[14]结合深度分离卷积、逐点卷积和瓶颈结构专为移动和嵌入式设备设计了轻量化识别模型. MobileID[15]使用知识蒸馏方法将教师模型蒸馏为一个大小为4.0 MB的紧凑学生模型. MobileFaceNet[16]使用谷歌提出的轻量型网络MobilenetV2作为其识别模型的特征提取网络,提高其在移动端的推理速度,并采用PReLu激活函数,以提高人脸识别精度. ShuffleFaceNet[17]使用轻量化通用网络ShufflenetV2作为其特征提取网络,进一步提高模型的推理速度.

结合所提的模型轻量化加速方法,本文针对人脸识别任务设计了一个实时、高精度的人脸识别模型FDFaceNet. 本文在统一嵌入式CPU和GPU硬件平台上比较现有不同轻量化人脸识别模型与本文人脸识别模型FDFaceNet的推理速度,以及它们在LFW(Labeled Faces in the Wild)展开人脸验证集上的准确率. 实验证明,所提的快速人脸识别模型在推理速度、特征图输出量和准确率方面都超过现有的轻量化人脸识别方法,实现了很好的模型识别精度与推理速度.

此外,本文对现有的基于RetinaFace[18]的人脸检测模型进行改进,适当减少模型大小和检测分支,提高人脸检测速度. 本文还配合所提的FDFaceNet人脸识别网络构建了一个高效人脸检测识别系统. 该系统在树莓派4B嵌入式开发板的Broadcom BCM2711四核CPU上最高运行帧率可达15FPS.

本文主要贡献总结为:

(1)通过研究不同类型卷积对模型推理速度的影响因素,首次指出模型特征图输出量是影响推理速度的一个关键因素,而不是一般认为的模型的推理速度仅与模型的计算量有关.

(2)基于上述发现,结合标准卷积提出了一种基于快速下采样的模型轻量化与加速设计方法,首次将模型特征图输出量和计算量同时作为轻量化指标. 实验证明,该方法在特征提取能力和推理速度方面均优于现有的基于深度分离卷积的轻量化方法.

(3)结合所提的基于快速下采样的模型轻量化与加速设计方法,针对人脸识别应用提出一个快速人脸识别模型FDFaceNet,并通过实验证明本文的人脸识别模型在推理速度、特征图输出量和准确率方面都超过现有的轻量化人脸识别方法. 此外,利用快速人脸检测模型并结合所提的FDFaceNet构建一个高效的人脸检测识别系统,在嵌入式设备上可进行实时人脸检测与识别.

2 模型复杂度与推理速度分析

设计轻量化模型的目的是希望能在存储空间和计算资源受限的移动端或嵌入式设备上部署深度学习应用,并提高模型在这些设备上的推理速度. 现有的基于分组卷积或深度分离卷积的模型轻量化工作[11,12]大多数仅把模型的参数量和计算量作为模型轻量化的指标. 然而,本文发现模型的计算量并不足以准确地反应模型推理时的实际速度,所以只有模型在硬件设备上的实际推理时间才是最重要的指标.

模型在硬件上的推理时间主要可以划分为计算用时和读写用时. 其中,影响计算用时的主要因素有模型计算量、硬件的计算能力以及推理框架针对硬件的对模型中算子的优化程度等. 影响读写用时的主要因素有模型特征图输出量、硬件内存和存储器的带宽等. 本文旨在研究模型结构对模型推理速度的影响. 模型结构主要会影响模型参数量、计算量和特征图输出量. 其他影响模型推理速度的因素(如推理框架针对硬件架构的优化等)实现上的优化方法本文不做重点讨论.

2.1 不同卷积类型的计算和空间复杂度分析

考虑模型推理时间分为计算用时和读写用时,计算用时主要来自模型的计算量,而读写用时主要来自模型推理时所产生的特征图输出量. 不同层特征图读和写的总和构成了整个模型的读写代价. 表2总结了不同类型的$K\times K$卷积的参数量、计算量和特征图输出量,其中K表示卷积核大小,M和N表示输入和输出特征图通道数,H和W表示输出特征图高和宽,g表示分组卷积的组数,特征图输出量F中的"4"表示每个特征值按照FP32存储将占用4 Bytes.

表2 不同卷积类型的参数量、计算量和特征图输出量对比

$K\times K$卷积	参数量	计算量	特征图输出量
标准卷积	K^2MN	K^2MNHW	$4NHW$
分组卷积	K^2MN/g	K^2MNHW/g	$4NHW$
深度分离卷积	$(K^2+N)M$	$(K^2+N)MHW$	$4(M+N)HW$
标准卷积(快速下采样)	K^2MN	$K^2MNH'W'$	$4NH'W'$

目前,轻量化网络设计方法中,分组卷积和深度分离卷积已经成为代替标准卷积最常用的方法. 分组卷积是指将输入特征图沿通道等分为g组,每个组内进行标准卷积,最后将g个组的输出特征图合并作为分组卷积的输出. 分组卷积的缺陷是每个组的特征图之间没有信息交互,特征提取能力较差. 深度分离卷积是指将一个$K\times K$标准卷积分为两步来完成,即一层$K\times K$逐通道卷积(depthwise convolution)和一层1×1逐点卷积(pointwise convolution),其中的第一步$K\times K$逐通道卷积是分组卷积的一个特例,将输入特征图的每个通道各自作为一组,即g等于M. $K\times K$逐通道卷积之后的1×1逐点卷积弥补了逐通道卷积每个组之间缺乏信息交流的

缺点. 表2中深度分离卷积的逐通道卷积的输入和输出通道数均为M,1×1逐点卷积的输入和输出通道数分别为M和N.

当模型通道数不变时,使用深度分离卷积代替标准卷积,两者计算量比值R_B如式(1)所示. 输出通道数N一般远大于K^2,假设N=128,K=3,则深度分离卷积可达8倍左右的计算量压缩比.

$$R_B=\frac{B_{\text{标准}}}{B_{\text{深度分离}}}=\frac{K^2N}{K^2+N} \tag{1}$$

深度分离卷积能够大幅减少模型计算量,但由于其将一层卷积变成两步操作,因此其特征图输出量是标准卷积的近2倍,这将导致模型推理时的读写用时增加. 特征图输出量比值R_F如式(2)所示. 一般来说,通道数M和N的数量级相同.

$$R_F=\frac{F_{\text{标准}}}{F_{\text{深度分离}}}=\frac{N}{M+N} \tag{2}$$

在特征提取能力方面,虽然深度分离卷积中逐点卷积层弥补了特征图组间的信息交互,但之前工作证明,深度分离卷积的特征提取能力依然不如标准卷积. 因此通常需要堆叠更多的深度分离卷积层或增加通道数,通过增加网络深度或宽度的方式来弥补准确率的损失,而此方式会进一步增加整个模型的特征图输出量,导致读写用时增加. 因此,深度分离卷积可以理解为一种以模型内存占用的增加交换模型计算复杂度减少的方法.

2.2　不同卷积类型对推理速度的影响

本文旨在研究轻量化模型结构对推理速度的影响,模型结构主要会影响模型的参数量、计算量和特征图输出量. 参数量主要影响模型的大小和性能,而计算量和特征图输出量主要影响模型的计算和读写代价.

为研究模型计算量、特征图输出量对推理速度的影响,本文设计并对比了4个卷积网络:由标准卷积组成的8层卷积模型(CNN-8),以及由深度分离卷积组成的8组、14组深度分离卷积模型(CNN-8-DS,CNN-14-DS和 CNN-14-DS×1.25). 其中"×1.25"表示每层通道数扩大为原始的1.25倍. 表3统计了以上4个模型的计算量、特征图输出量以及在嵌入式CPU与GPU设备上的推理速度. 其中CPU推理速度采用开源的NCNN推理框架在树莓派4B四核CPU上进行测试,GPU速度采用PyTorch框架在Nvidia Jetson TX2上测试,具体配置见表8,输入均为224×224. 分析表3前3列可知,采用深度分离卷积代替标准卷积层,即使在模型深度或宽度适当增加的情况下也可以大幅减少模型的计算量B,但由于模型层数的增加,模型的特征图输出量F会成倍增加. 表3第6列列出了推理每个模型的CPU每秒平均浮点运算次数(FLoating-point Operations Per Second,FLOPS),计算方式如式(3)所示. 其中,B表示模型计算量,t表示CPU推理时间.

$$\text{FLOPS}_{\text{CPU}}=B/t \tag{3}$$

通过表3可以发现,虽然深度分离卷积模型在CPU上的总推理时间有所减少,但是8层标准卷积网络(CNN-8)的CPU每秒平均浮点运算次数反而更高,且远大于其他基于深度分离卷积的轻量化模型. 从特征图输出量的角度定性分析表3,采用标准卷积的模型在推理时每秒平均用于特征图传输的用时更少,用于计算的用时更多,因此基于标准卷积的模型CPU每秒平均浮点运算次数更多.

表3　标准卷积与基于深度分离卷积模型在嵌入式CPU和GPU的推理速度

模型	参数量/M	计算量/M	特征图输出量/MB	CPU推理时间/ms	CPU每秒平均浮点运算次数/(M/s)	GPU推理时间/ms
CNN-8	3.4	2.6	58.20	385.3	6 748	9.8
CNN-8-DS	0.9	333	84.86	159.8	2 083	13.4
CNN-14-DS	1.1	472	110.14	196.4	2 403	23.1
CNN-14-DS×1.25	1.5	719	137.68	282.5	2 545	22.8

2.3　嵌入式系统的模型推理性能分析模型

同时考虑计算成本和读写成本,模型在一个计算平台上的推理时间T取决于硬件的算力P与带宽D和模型自身的计算量B和特征图输出量F,其关系式如式(4)所示. 其中,α和θ分别为平台的算力效率系数和带宽效率系数,取值范围为(0,1.0],当$\alpha=\theta=1.0$时,表示以最大算力与最大带宽推理模型,τ为其他用时(如加载模型和权重等).

$$T=\frac{B}{\alpha P}+\frac{F}{\theta D}+\tau \tag{4}$$

当模型在同一嵌入式平台上以相同算力和带宽推理时,其推理速度与模型计算量和特征图输出量呈线性正相关关系,如图1所示. 所以,要想提高模型的推理速度,不仅要减少模型整体计算量B,也要考虑特征图的读写用时,即特征图输出量F在其中的作用. 假设标准卷积模型和深度分离卷积模型的计算量相同,则特征图输出量会成为影响模型推理速度的主导因素.

结合以上对模型在嵌入式端推理速度影响的分析,可以总结出:除了减少模型的计算量,减少模型特征图输出量也是加速模型推理的一个关键因素.

计算量会影响推理时的计算用时,而特征图输出量会影响推理时的读写用时. 因此,在轻量化模型设计

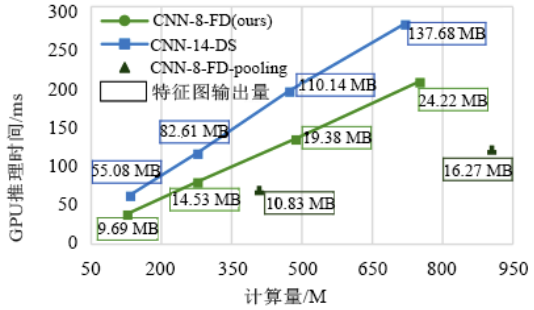


图1 两种轻量化模型的CPU推理时间

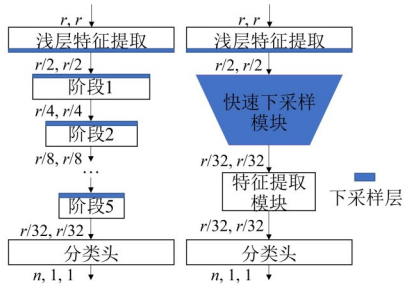
的方法中，在保持模型准确率的前提下，要想提升模型的推理速度，不仅要减少模型的计算量，也要考虑推理时模型的特征图输出量. 然而显然现有的深度分离卷积仅考虑到了计算量，因此本文在第3节提出基于快速下采样的模型轻量化设计方法.

3 基于快速下采样的模型轻量化与加速设计方法

根据以上分析，本文提出一种基于快速下采样模块并结合标准卷积的轻量化加速模型设计方法. 该方法可以同时减少模型的计算量和特征图输出量，并减少推理时的占用内存.

3.1 基于快速下采样模块的模型轻量化设计方法

不同于大部分工作中采用的阶段式下采样的模型设计范式[19]，如图2(*a*)所示，其中r表示特征图高和宽，n表示模型输出的向量维度. 本文提出的模型轻量化加速方法是在模型的浅层特征提取后添加一个快速下采样模块，并将模型的下采样层全部集中在快速下采样模块中，后续的特征提取模块中不再包含对特征图进行下采样的模块，如图2(*b*)所示.



(*a*) 阶段式下采样的模型设计范式 (*b*) 基于快速下采样模块的模型加速方法

图2 两种模型设计方法

本文优选使用带步长的标准卷积层作为快速下采样模块的下采样层，因为其相较于深度分离卷积的特征图输出量更少，而且标准卷积具有更好的特征提取能力. 卷积层步长为2时，特征图高和宽两个维度上需要进行卷积的位置均减半，计算量可减少为原来的1/4. 带有步长的卷积层同时具有池化的作用，可以省去池化层带来的额外的特征图输出量. 另外，尽量采用卷积核尺寸相同的标准卷积层来构建网络，防止模型中的算子过于碎片化导致推理时硬件利用效率降低，本文采用最常用的3×3卷积核，因为它在各种推理框架中受到更多优化.

假设特征图的高和宽相等(即边长$r=H=W$)，那么由表2可看出，标准卷积的计算量B和特征图输出量F与特征图边长r的平方成正比，即

$$B \propto r^2 \tag{5}$$

$$F \propto r^2 \tag{6}$$

模型浅层的快速下采样模块作用是快速减小特征图的边长r，由式(5)和式(6)可知，边长r快速变小使模型深层通道数更大的卷积层所需的计算量和特征图输出量大幅降低，从而可提高模型推理速度.

分析表2中的不同卷积计算量与参数量和特征图输出量的关系可知，本文提出的轻量化方法是通过减小特征图尺寸来减少计算量. 在通道数相同的情况下，深度分离卷积与基于快速下采样的标准卷积的计算量之比R'_B如式(7)所示，其中，R表示输出特征图边长的比值，即$R=H/H'=W/W'$.

$$R'_B=\frac{B_{\text{深度分离}}}{B_{\text{标准(快速下采样)}}}=\frac{(K^2+N)}{K^2N}R^2\approx\frac{R^2}{K^2} \tag{7}$$

通常，深度网络的通道数N远大于K^2，则$R'_B\approx R^2/K^2$. 使用3×3卷积时(K=3)，根据图2中两种模型设计方法的下采样位置，相同层的下采样倍数$R\in\{1,2,4,8,16,32\}$. 当$R\leqslant 2$时，深度分离卷积计算量更少；当$R\geqslant 4$时，基于快速下采样的标准卷积计算量更少.

然而，由于深度分离卷积特征提取能力不如标准卷积，需要增加网络深度或宽度的方式来弥补准确率的损失，则需要累加的卷积层比标准模型更多些. 总的来说，如果调整合适的模型深度，基于深度分离卷积的模型和基于本文提出的轻量化模型设计方法的模型整体计算量可以达到同一水平，可参考表4.

进一步地，本文在表4中证明了所提方法在推理速度方面的优势. 使用所提的模型轻量化加速设计方法，按照第2.2节同样设计了8层标准卷积模型CNN-8-FD，并对其每层通道数添加了缩放系数.

表4对比了不同通道缩放系数下的两种轻量化模型设计方法的嵌入式CPU与GPU推理速度，如图1所示，本文所提的基于快速下采样结合标准卷积的轻量化模型与基于深度分离卷积的模型具有相同的计算量，但是由于减少了特征图输出量，本文提出的模型在不同计算量情况下的CPU推理时间明显减少. 在嵌入

式GPU上,本文的模型在相同计算量情况下GPU推理速度明显快于基于深度分离卷积的模型,如图3所示.

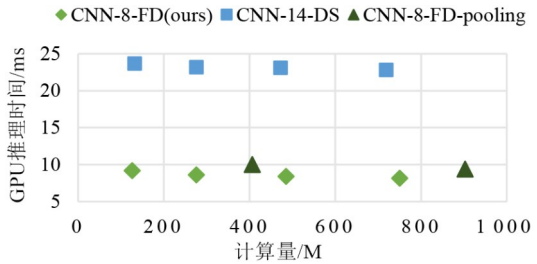


图3　两种轻量化模型的GPU推理时间

表4中的CNN-8-FD-Pooling为带有池化层的快速下采样模型,分析可知采用去除池化层的模型CNN-8-FD在模型计算量、特征图输出量和推理速度等指标上更有优势.

结合以上分析,本文提出的基于快速下采样模块的轻量化与加速方法能够同时减少模型的计算量与特征图输出量,并且能有效地减少模型在不同设备上的推理时间,在相同计算量情况下优于基于深度分离卷积的轻量化模型.

总结模型设计时需要遵循的几点设计思路:首先,基于快速下采样的模型加速设计方法配合标准卷积可以达到更好的效果,因为标准卷积具有更好的特征提取能力,并且其产生的内存占用更少;其次,采用带有步长的卷积层作为下采样层,减少模型计算量和特征图输出量;另外,如果在模型中使用了BN层,为进一步减少模型前向传播时的特征图输出量,推理前可利用Conv-BN融合技巧将BN层中的参数等效地与其前一个卷积层融合,其具体等效过程如下.

表4　两种轻量化模型在嵌入式CPU和GPU的推理速度

模型	计算量/M	特征图输出量/MB	CPU推理速度(Pi 4B)/ms	GPU推理速度(TX2)/ms
CNN-14-DS ×0.5	132	55.08	63.0	23.7
CNN-14-DS ×0.75	276	82.61	116.8	23.2
CNN-14-DS ×1.0	472	110.14	196.4	23.1
CNN-14-DS ×1.25	719	137.68	282.5	22.8
CNN-8-FD ×0.5	126	9.69	38.6	9.2
CNN-8-FD ×0.75	276	14.53	79.9	8.6
CNN-8-FD ×1.0	485	19.38	135.2	8.4
CNN-8-FD ×1.25	750	24.22	208.4	8.2
CNN-8-FD-Pooling ×0.5	406	10.83	70.6	10.0
CNN-8-FD-Pooling ×0.75	903	16.27	122.2	9.4

输入特征图$M^{(1)}$经过卷积层后再经过BN层进行归一化输出特征图$M^{(2)}$,该过程为式(8).其中W表示卷积层参数;μ,σ,γ和β分别表示BN层均值、标准差和可训练缩放因子与偏置.BN计算过程如式(9)所示,其中i表示特征图第i个通道.

$$M^{(2)}=\mathrm{BN}(M^{(1)}*W,\ \mu,\sigma,\gamma,\beta) \tag{8}$$

$$\mathrm{BN}\left(M,\mu,\sigma,\gamma,\beta\right)_{:,i,:,:}=(M_{:,i,:,:}-\mu_i)\frac{\gamma_i}{\sigma_i}+\beta_i \tag{9}$$

如果令

$$W'_{i,:,:,:}=\frac{\gamma_i}{\sigma_i}W_{i,:,:,:},\ b'_i=-\mu_i\frac{\gamma_i}{\sigma_i}+\beta_i \tag{10}$$

则式(8)可改写为

$$\mathrm{BN}\left(M*W,\ \mu,\sigma,\gamma,\beta\right)_{:,i,:,:}=(M*W')_{:,i,:,:}+b'_i \tag{11}$$

式(11)表明BN层与前一层卷积融合在数学上等效,因此不会影响推理模型的精度.同时,BN融合能够减少因BN层产生的额外特征图输出量,减少模型推理时的读写用时.

基于本节提出的轻量化设计方法,在第4节针对人脸识别任务构建了快速识别网络,结构如图4(a)所示.

3.2　两种轻量化设计方法的模型性能对比

为了验证采用基于深度分离卷积的轻量化模型与所提出的基于快速下采样模块的轻量化模型的特征提取能力,表5在4个CNN模型上进行对比实验,其中基线模型CNN-8为标准卷积网络,CNN-8-DS和CNN-14-DS采用了基于深度分离卷积的轻量化方法,CNN-8-FD采用了本文提出的基于快速下采样的模型轻量化方法.

本文采用相同的训练策略在CIFAR10图像分类任务上训练相同的迭代次数,表5中列出每个模型的测试集准确率.在模型深度相同的情况下,由于CNN-8-FD采用的是标准卷积层,其测试集准确率也较优于采用深度分离卷积的CNN-8-DS.与模型深度更深的CNN-14-DS相比,在计算量相当的情况下,采用了本文提出的轻量化加速方法而设计的CNN-8-FD的测试准确率相比基线模型下降得更少,同时能够大幅降低模型特征图输出量.在第4节中,本文的轻量化方法在人脸识别任务上的准确率也可以超越基于深度分离卷积的方法,并且在嵌入式设备上的推理深度更优.

表5 CIFAR10任务上模型性能对比

模型	准确率/%	计算量/M	特征图输出量/MB
CNN-8	90.53	53	1.19
CNN-8-DS	84.62	6.8	1.73
CNN-14-DS	85.34	9.6	2.25
CNN-8-FD	**85.42**	9.8	**0.40**

4 基于快速下采样模块的人脸识别网络与高效人脸检测识别系统

本节中提出一个基于快速下采样模块的人脸识别网络. 第4.1节阐述了人脸识别网络的一般过程和复杂度,第4.2节介绍了结合标准卷积建立基于快速下采样模块的人脸识别网络FDFaceNet的过程和具体细节. 第4.3节中对现有人脸检测网络进行改进,结合所提的快速人脸识别网络组成高效人脸检测识别系统.

4.1 人脸识别模型一般过程

现有基于深度学习的人脸识别的通常做法是把一张对齐的人脸图像通过人脸识别网络,通过度量学习来进行1:1人脸验证或1:N人脸识别任务[20,21]. 人脸识别网络可以理解为一个编码器E,假设x为一张对齐人脸图像,y为输出的人脸特征向量(一般为128维、256维或512维),则有

$$y = E(x) \tag{12}$$

为得到质量更好的人脸特征向量,近些年的大多数人脸识别工作主要集中在改进人脸识别网络的训练损失函数,如CosFace[22]和ArcFace[23]试图在余弦空间最小化相同ID的人脸特征向量距离,同时最大化不同ID的人脸特征向量距离,这些工作中的特征提取网络一般采用通用的特征提取网络VGG16[24]或ResNet18/50[25]. 在人脸识别损失函数的监督下,这些通用特征提取网络可以在大多数人脸验证数据集上获得很高的准确率.

但在实际应用中,由于通用特征提取网络所需的计算资源过多,而且人脸识别模型还需人脸检测作为上游任务,因此整个人脸检测识别系统很难在资源受限的嵌入式设备上获得高实时性. 表6所示为VGG和ResNet的参数量、计算量和嵌入式CPU平台上的平均推理速度. 显然,通用特征提取网络无法满足人脸识别应用的实时性.

表6 通用模型在嵌入式设备上的推理速度对比

模型	参数量/M	计算量/G	4核CPU推理速度/ms	
			树莓派4B	RK3288
VGG16	138.36	15.61	1 116	966
ResNet18	11.69	1.82	258	191
ResNet50	25.56	4.14	508	445

4.2 基于快速下采样模块的人脸识别模型设计

结合标准卷积,本文采用基于快速下采样模块的轻量化方法,针对现有的十层人脸识别方法VIPLFaceNet[26]进行改进,并提出了快速人脸识别模型FDFaceNet. 在通道数配置和网络深度方面采用了相同的配置.

图4(a)为提出的基于快速下采样模块的人脸识别网络FDFaceNet结构,主要由4个模块组成:快速下采样模块、特征提取模块、全局逐通道卷积层以及全连接层. 为了满足计算量和特征图输出量更少的设计要求,并保持模型整体结构一致性,FDFaceNet完全采用3×3卷积来构建模型的快速下采样模块和特征提取模块,并且采用步长为2的卷积层来代替步长为2的池化层. 在模型最后采用一层全连接来输出人脸特征向量,输出维度为512. 另外,受MobileFaceNet[16]中的启发,本文在全连接层之前添加了全局逐通道卷积层(Global Depthwise Convention,GDC),该模块对卷积层输出的最后一层特征图的每个通道上添加空间维度的注意力,同时起到全局平均池化的作用,将特征图尺寸池化至1×1,降低了最后一层全连接的输入维度,减少了全连接层的参数量和运算量.

全局逐通道卷积的过程可以写为

$$M^{(2)}_{i,:,:} = M^{(1)}_{i,:,:} * W_{i,:,:,:} \tag{13}$$

其中,$M^{(1)} \in \mathbb{R}^{C\times r\times r}$和$M^{(2)} \in \mathbb{R}^{C\times 1\times 1}$表示GDC模块输入和输出特征图,$C$为特征图通道数,$r$为特征图高宽,$W_{i,:,:,:} \in \mathbb{R}^{C\times 1\times r\times r}$表示第$i$个输出通道对应的卷积核.

FDFaceNet的具体参数如表7所示. 模型浅层的快速下采样模块由5个步长为2的CONV-P模块组成的快速下采样模块来快速减小特征图的尺寸. 快速下采样模块的输出再经过3层步长为1的CONV-P模块进行进一步的特征提取. 输出的最后一层特征图再经过一层全局逐通道卷积层GDC将特征图尺寸减至1×1,经过维度变换后再通过一层全连接层和一维BN层输出最终的512维人脸特征向量. CONV-P模块如图4(b)所示,每个CONV-P模块中包含一层3×3标准卷积,卷积之后连接了BN层和PReLu激活函数层. 另外,本文基于深度分离卷积设计了DSConv-P模块用于之后的对比实验,由一层3×3逐通道卷积层和一层1×1逐点卷积层组成,如图4(c)所示.

模型训练完成后,最后利用Conv-BN融合的方法将所有卷积层与其后面的BN层融合为一层,进一步减少模型推理时的特征图输出量,提高模型在不同设备上的推理速度.

4.3 快速人脸检测识别系统

本文对现有的基于RetinaFace的人脸检测模型进行了改进,以提高在嵌入式上的推理速度,并将其作为

表 7　FDFaceNet 结构

层	通道数	核大小	步长	填充
CONV-P 1	48	3×3	2	1
CONV-P 2	128	3×3	2	1
CONV-P 3	128	3×3	2	1
CONV-P 4	256	3×3	2	1
CONV-P 5	192	3×3	2	1
CONV-P 6	192	3×3	1	1
CONV-P 7	256	3×3	1	1
CONV-P 8	512	3×3	1	1
GDC	512	4×4	1	0
FC1	512	-	-	-
BN-1d	512	-	-	-

上游任务与所提出的快速人脸识别模型 FDFaceNet 级联,最终在嵌入式设备上构建了一个实时的人脸检测识别系统.

具体地,本文用轻量化特征提取网络(Mobilenet, Shufflenet 等)来替换检测网络原始结构中的主干特征提取网络,并将检测分支分别减少为 3. 另外,针对人脸检测网络,除了输出人脸置信度和人脸框位置,本文还在检测头中加入额外的人脸关键点预测分支,以输出每个人脸预测框中 5 个人脸关键点(左眼、右眼、鼻子、左嘴角、右嘴角),这些人脸关键点信息后续将用于人脸对齐. 本文在实验中根据不同检测模型的检测准确率与推理速度的折中关系选择出最优检测模型.

5　实验及结果分析

5.1　人脸识别模型训练与测试

本文训练人脸识别模型所用的数据集均为 MSCeleb-Arcface 人脸数据集,包含约 580 万张人脸图像,约 85 000 个 ID. 用于训练的人脸图像尺寸统一为 112×112,并经过 5 个关键点对齐处理. 模型输出的人脸特征向量均采用 512 维向量.

为了公平比较不同模型的性能,所有模型均采用 ArcFace loss 从随机权重初始化开始训练. 训练的 batchsize 设为 256,训练的优化器采用随机梯度下降(Stochastic Gradient Descent, SGD),其权重衰减设置为 0.000 5,动量项设为 0.9,初始学习率设为 0.01,并在第 160 k, 210 k, 250 k, 290 k 次迭代周期时衰减 10 倍. 测试数据集采用 LFW 人脸验证数据集,包含来自 5 749 个 ID 的 13 233 张人脸图像,并从中抽取 6 000 对人脸用于人脸验证.

测试时,先利用人脸检测器 Retinaface 对输入的 LFW 数据集图像进行检测,再利用检测到的 5 个人脸关键点将人脸区域进行人脸对齐,输出尺寸的 112×112 对齐人脸图像最后再送入人脸识别模型进行特征提取.

本文在统一的嵌入式平台上对不同的轻量化人脸识别模型进行平均推理速度测试. 具体硬件配置如表 8 所示,在 ResberryPi 4B 和 Firefly RK3288 上利用开源的 C++推理框架 NCNN 来测试模型在 ARM CPU 上的推理速度,在 NVIDIA Jetson TX2 上利用 Pytorch 1.7 框架测试模型在 ARM GPU CUDA 上的推理速度.

表 8　模型推理速度测试平台配置

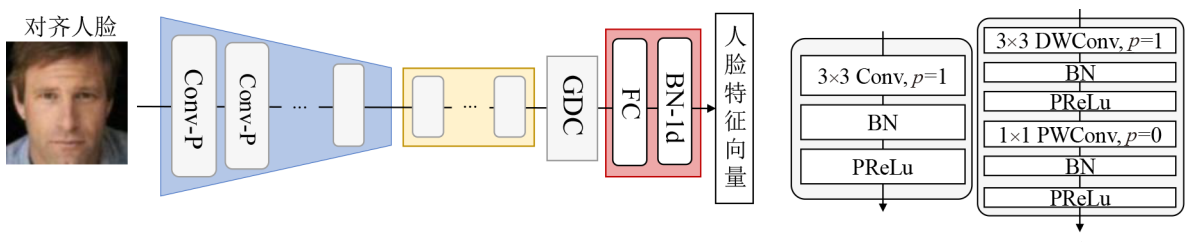
	CPU 平台		GPU 平台
开发平台	Firefly RK3288	ResberryPi 4B	NVIDIA Jetson TX2
CPU	Rockchip RK3288	Broadcom BCM2711	Cortex A57 4 核 Denver 2 核
GPU	无	无	Pascal CUDA
推理框架	NCNN(C++)	NCNN(C++)	Pytorch1.7(CUDA)

5.2　实验结果

5.2.1　消融实验

为了验证提出的 FDFaceNet(FDFN)在准确率和推理速度方面的优越性,本文设计了两种结构作为对照实验:NFaceNet(NFN)和 DWFaceNet(DWFN). 其中,NFN 采用了标准卷积和传统的阶段式下采样模型设计范式,DWFN 在 NFN 基础上采用了基于深度分离卷积的模型轻量化方法,使用图 4(*c*)所示的 DSConv-P 模块来构建模型.

本文使用第 5.1 节中的训练配置对以上模型进行训练,并在统一的嵌入式硬件平台上测试不同模型的平均推理速度.



(*a*) 快速人脸识别网络结构　　(*b*) Conv-P 模块　　(*c*) DSConv-P 模块

图 4　快速人脸识别网络结构及其内部模块

(1)GDC模块的消融实验

表9列出了NFN*,FDFN*和FDFN的LFW验证准确率、参数量、计算量以及在树莓派4B CPU和Jetson TX2 GPU上的推理速度,其中"*"表示未使用GDC模块.

表9 GDC模块的消融实验

模型	GDC	LFW准确率/%	参数量/M	计算量/M	推理时间/ms	
					CPU	GPU
NFN*	×	99.13	7.10	660	87.5	8.2
FDFN*	×	99.31	7.10	134	55.6	8.7
FDFN	√	**99.35**	**3.17**	**130**	**44.1**	9.5

对比表9中第1、2行(NFN*和FDFN*),在未使用GDC模块的情况下,采用本文提出的基于快速下采样的模型轻量化方法可有效减少模型的计算量,同时大幅减少模型在CPU上的推理时间.

对比表9中第2、3行(FDFN*和FDFN),采用GDC模块有效地提高了LFW准确率.而且,因为GDC起到了全局池化的作用,FDFN最后一层的全连接层参数量和计算量大幅降低,进一步减少了模型在CPU上的推理时间.

(2)两种轻量化方法的对比实验

表10列出传统CNN模型与两种轻量化模型的LFW验证准确率、计算量、特征图输出量以及在树莓派4B CPU和Jetson TX2 GPU上的推理速度.对于基于深度分离卷积的模型DWFN,采用GDC模块会导致其模型参数量过少(小于0.6 M)且识别准确率很低,因此在对比实验中未对其添加GDC模块,并对其通道数进行缩放(×1.25).

对比表10中第2、3行(DWFN*×1.25和FDFN),在相同的计算量下,本文提出的基于快速下采样的轻量化人脸识别模型的LFW准确率更高,特征输出量更少,而且更重要的是其CPU和GPU推理时间均优于基于深度分离卷积的模型.

表10 基线模型与两种轻量化模型对比

模型	LFW准确率/%	计算量/M	特征图输出量/MB	推理时间/ms	
				CPU	GPU
NFN	99.26	656	21.92	84.4	9.6
DWFN* ×1.25	99.20	132	26.70	51.1	13.9
FDFN	**99.35**	**130**	**7.39**	**44.1**	**9.5**

表10说明基于快速下采样模块配合标准卷积设计的人脸识别模型FDFN可以在人脸识别任务上实现更好的识别准确率与推理速度的权衡.

以上实验证明,基于快速下采样的模型轻量化加速方法配合标准卷积不仅可以有效减少模型的参数量和计算量,提高模型在计算受限设备上的推理速度,而且解决了深度分离卷积使模型前向推理时特征图内存用量变多的问题,从而避免了模型由于特征图传输量的增加,其在计算能力更强硬件设备上的推理速度减慢的问题.

5.2.2 与现有轻量化人脸识别方法对比

为了将所提的FDFN与之前轻量化人脸识别工作进行比较,本文在相同的数据集上复现了之前的轻量化人脸识别模型,在表11中用"#"表示复现结果,并同样在第5.1节所述的嵌入式硬件平台上测试不同模型的CPU和GPU推理速度,输入图片大小均为112×112.表11中记录了不同轻量化识别模型的参数量、计算量、推理速度以及在LFW数据集上的准确率.

表11 与现有轻量化人脸识别模型的结果对比

轻量化人脸识别网络	LFW准确率/%	参数量/M	特征图输出量/MB	计算量/M	CPU推理速度/ms		GPU推理速度/ms
					ResberryPi 4B	RK3288	Jetson TX2
Light CNN-9 [27]	98.80	5.6	-	-	-	-	-
Light CNN-29 [27]	99.33	12.6	-	-	-	-	-
MobileID [15]	97.32	1.0	-	-	-	-	-
MobileFaceNet # [16]	99.20	1.0	73.4	230	66.7	85.4	40.3
ShuffleFaceNet # (×1.0) [17]	99.21	1.4	37.7	143	54.8	75.5	54.2
Lightfacenet [14]	99.32	1.1	48.08	227	60.2	83.4	80.3
ProxylessFaceNAS [13]	99.20	4.9	200.78	481	176.2	171.8	52.0
FDFN*(本文)	99.31	7.1	**4.9**	134	55.6	66.2	**8.7**
FDFaceNet (本文)	**99.35**	3.2	**7.4**	**130**	**44.1**	**54.6**	**9.5**

分析表11的实验数据可知,在相同的训练集和训练配置下,FDFaceNet的LFW人脸验证集准确率超过了轻量化人脸识别模型MobileFaceNet和ShuffleFaceNet(×1.0).在嵌入式平台的推理速度方面,本文识别模型在ARM CPU上的四核推理速度约为44 ms,在TX2 GPU上的推理速度约为9.5 ms,比MobileFaceNet分别快了22 ms和30 ms.

ShuffleFaceNet(×1.0)与所提的FDFaceNet具有相似计算量,但其在ARM CPU上的推理速度慢于FDFaceNet.原因是其前向传播时特征图输出量更大,用

于特征图传输的用时更多. 该现象在算力更高的GPU平台上更加明显,FDFaceNet在TX2 GPU上比ShuffleFaceNet快了约45 ms. 这证明了模型在不同平台上的推理速度不仅和模型的计算量有关,而且与模型特征图输出量有关. 本文的方法在嵌入式CPU和GPU平台上的推理速度均超过了现有的基于分组卷积和深度分离卷积的人脸轻量化网络.

图5和图6为现有不同轻量化人脸识别模型LFW准确率和CPU/GPU推理时间对比图. 本文基于快速下采样并结合标准卷积设计的快速人脸识别网络FDFaceNet在不同的平台上都取得了很好的推理速度与识别准确率的权衡.

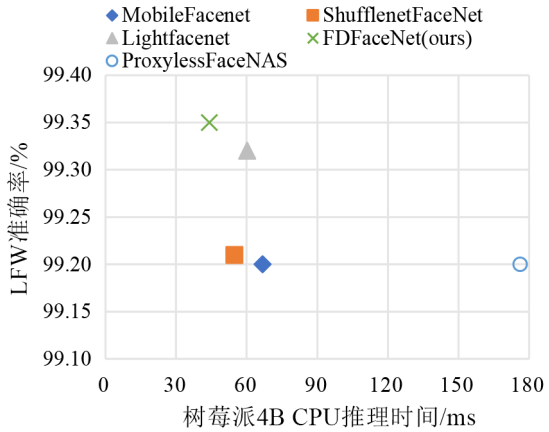


图5　现有轻量化人脸识别模型准确率与CPU推理时间对比

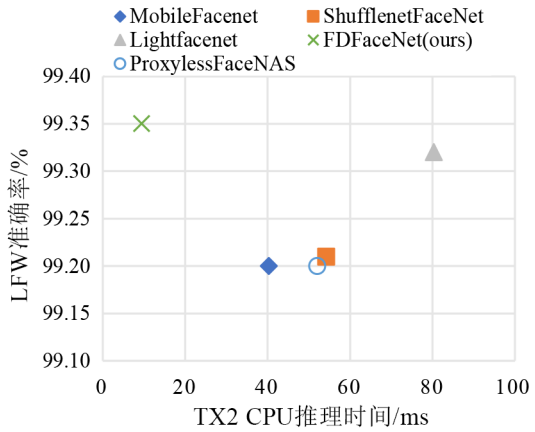


图6　现有轻量化人脸识别模型准确率与GPU推理时间对比

5.3　人脸检测识别系统性能

如第4.3节所述,在人脸识别网络之前加入了基于预设框的人脸检测模型和对齐模块以组成完整的人脸检测识别系统. 对于该系统,本文首先测试了不同主干网络组成的人脸检测模型准确率以及检测模型在嵌入式设备上的平均推理速度,然后测试了人脸检测识别系统的整体的运行速度.

本文改进了基于SSD[28]和基于RetinaFace的单阶段人脸检测网络,人脸检测网络的训练数据集采用带有关键点标注的WiderFace数据集,测试集采用的是FDDB数据集. 表12中第1和第2行比较了不同主干网络含4个检测分支的SSD检测模型,第3和第4行比较了含3个检测分支的RetinaFace检测模型,分析发现,以MobilenetV1×0. 25为主干网络的RetinaFace人脸检测模型在FDDB数据集上的检测结果最佳,而且其推理速度在树莓派4B的CPU最快. 因此在后续的人脸检测识别系统中采用了3-Retina-MBV1×0.25模型.

配合人脸检测模型与本文提出的快速人脸识别模型FDFaceNet,该人脸检测识别系统能够在嵌入式设备上对摄像头采集的视频帧进行实时检测和识别. 本文利用推理框架在嵌入式设备上搭建整个人脸检测识别系统,其在树莓派4B四核CPU上的最高系统运行帧率可达15FPS,在i5的四核CPU上最高可达40FPS,如表13所示.

表12　人脸检测模型性能和速度对比

人脸检测网络	FDDB Average AP	FLOPs/M	ResberryPi 4B推理速度/ms
4-SSD-MBV2	92.4	113	73.1
4-SSD-ShuffleV2×0.75	91.3	132	68.8
3-Retina-MBV1×0.25	**93.1**	254	**67.4**
3-Retina-ShuffleV2×0.75	93.0	264	81.7

表13　人脸检测识别系统性能

人脸检测	人脸识别	CPU平台	最高帧率/FPS
3-Retina-MBV1×0.25	FDFaceNet	X86平台 i5-7500 CPU	40
		树莓派4B CortexA72 4核CPU Broadcom BCM2711	15

6　结论

本文通过分析不同卷积类型的复杂度以及对推理速度的影响,发现模型的推理速度不仅取决于模型的计算量,同时也受到模型特征图输出量的影响. 由此本文提出一种基于快速下采样并结合标准卷积的模型轻量化与加速设计方法. 这是首个能够同时优化模型计算量和特征图输出量的模型轻量化加速方法. 不同于常用的基于深度分离卷积或分组卷积的轻量化方法,

本文的方法仅结合标准卷积即可有效减少模型的计算量和特征图输出量,并且其特征提取能力和推理速度均优于现有的基于深度分离卷积的轻量化方法. 利用该轻量化加速设计方法,本文针对人脸识别应用提出一个快速人脸识别网络FDFaceNet. 实验证明,所提的方法在不同CPU和GPU平台上的推理速度均优于现有的轻量化人脸识别模型,同时在LFW数据集上有更高的验证准确率. 另外,配合轻量化人脸检测模型,完整的人脸检测识别系统能够在嵌入式设备上完成实时的高精度人脸检测和识别.

参考文献

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

[2] CHEN H T, WANG Y H, XU C, et al. Data-free learning of student networks[C]//2019 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE, 2020: 3513-3521.

[3] 胡骏, 黄启鹏, 刘嘉昕, 等. 引入概率分布的深度神经网络贪婪剪枝[J]. 中国图象图形学报, 2021, 26(1): 198-207.
HU J, HUANG Q P, LIU J X, et al. Greedy pruning of deep neural networks fused with probability distribution [J]. Journal of Image and Graphics, 2021, 26(1): 198-207. (in Chinese)

[4] ZHUANG B H, TAN M K, LIU J, et al. Effective training of convolutional neural networks with low-bitwidth weights and activations[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(10): 6140-6152.

[5] HOWARD A G, ZHU M, CHEN B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications[EB/OL]. (2017-04-17) [2021-08-01]. https://arxiv.org/abs/1704.04861.

[6] SANDLER M, HOWARD A, ZHU M L, et al. MobileNetV2: inverted residuals and linear bottlenecks[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2018: 4510-4520.

[7] MA N N, ZHANG X Y, ZHENG H T, et al. ShuffleNet V2: Practical guidelines for efficient CNN architecture design[C]//Computer Vision - ECCV 2018: 15th European Conference. New York: ACM, 2018: 122-138.

[8] HUMPHREY E J, BELLO J P. Rethinking automatic chord recognition with convolutional neural networks[C]// 2012 11th International Conference on Machine Learning and Applications. Piscataway: IEEE, 2013: 357-362.

[9] CHOLLET F. Xception: Deep learning with depthwise separable convolutions[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2017: 1800-1807.

[10] 权宇, 李志欣, 张灿龙, 等. 融合深度扩张网络和轻量化网络的目标检测模型[J]. 电子学报, 2020, 48(2): 390-397.
QUAN Y, LI Z X, ZHANG C L, et al. Fusing deep dilated convolutions network and light-weight network for object detection[J]. Acta Electronica Sinica, 2020, 48(2): 390-397. (in Chinese)

[11] QIU J, CHEN C, LIU S, et al. SlimConv: Reducing channel redundancy in convolutional neural networks by weights flipping[EB/OL]. (2020-05-16) [2021-08-01]. https://arxiv.org/abs/2003.07469.

[12] HAN K, WANG Y H, TIAN Q, et al. GhostNet: more features from cheap operations[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 1577-1586.

[13] MARTÍNEZ-DÍAZ Y, NICOLÁS-DÍAZ M, MÉNDEZ-VÁZQUEZ H, et al. Benchmarking lightweight face architectures on specific face recognition scenarios[J]. Artificial Intelligence Review, 2021, 54(8): 6201-6244.

[14] 张典, 汪海涛, 姜瑛, 等. 基于轻量级网络的实时人脸识别算法研究[J]. 计算机科学与探索, 2020, 14(2): 317-324.
ZHANG D, WANG H T, JIANG Y, et al. Research on real-time face recognition algorithm based on lightweight network[J]. Journal of Frontiers of Computer Science & Technology, 2020, 14(2): 317-324. (in Chinese)

[15] LUO P, ZHU Z Y, LIU Z W, et al. Face model compression by distilling knowledge from neurons[C]//Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. New York: ACM, 2016: 3560-3566.

[16] CHEN S, LIU Y, GAO X, et al. MobileFaceNets: Efficient CNNS for accurate real-time face verification on mobile devices[M]//Biometric Recognition. Cham: Springer International Publishing, 2018: 428-438.

[17] MARTINDEZ-DÍAZ Y, LUEVANO L S, MENDEZ-VAZQUEZ H, et al. ShuffleFaceNet: A lightweight face architecture for efficient and highly-accurate face recognition[C]//2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). Piscataway: IEEE, 2020: 2721-2728.

[18] DENG J, GUO J, ZHOU Y, et al. RetinaFace: Single-

stage dense face localisation in the wild[EB/OL]. (2019-05-02)[2021-08-01]. https://arxiv.org/abs/1905.00641.

[19] RADOSAVOVIC I, KOSARAJU R P, GIRSHICK R, et al. Designing network design spaces[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 10425-10433.

[20] 徐先峰, 张丽, 郎彬, 等. 引入感知模型的改进孪生卷积神经网络实现人脸识别算法研究[J]. 电子学报, 2020, 48(4): 643-647.
XU X F, ZHANG L, LANG B, et al. Research on inception module incorporated Siamese convolutional neural networks to realize face recognition[J]. Acta Electronica Sinica, 2020, 48(4): 643-647. (in Chinese)

[21] 吴长虹, 苏剑波, 陈叶飞. 抗年龄干扰的人脸识别[J]. 电子学报, 2018, 46(7): 1593-1600.
WU C H, SU J B, CHEN Y F. Age invariant face recognition[J]. Acta Electronica Sinica, 2018, 46(7): 1593-1600. (in Chinese)

[22] WANG H, WANG Y T, ZHOU Z, et al. CosFace: large margin cosine loss for deep face recognition[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2018: 5265-5274.

[23] DENG J K, GUO J, YANG J, et al. ArcFace: Additive angular margin loss for deep face recognition[C]//IEEE Transactions on Pattern Analysis and Machine Intelligence. Piscataway: IEEE, 2021: 5962-5979.

[24] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition [EB/OL]. (2014-09-04)[2021-08-01]. https://arxiv.org/abs/1409.1556.

[25] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2016: 770-778.

[26] LIU X, KAN M N, WU W L, et al. VIPLFaceNet: An open source deep face recognition SDK[J]. Frontiers of Computer Science, 2017, 11(2): 208-218.

[27] WU X, HE R, SUN Z N, et al. A light CNN for deep face representation with noisy labels[J]. IEEE Transactions on Information Forensics and Security, 2018, 13(11): 2884-2896.

[28] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single shot MultiBox detector[M]//Computer Vision - ECCV 2016. Cham: Springer International Publishing, 2016: 21-37.

作者简介

王佳皓　男,1997年生. 上海大学通信与信息工程学院硕士研究生. 主要研究方向为模型压缩与加速、人脸识别等.
E-mail: nikkonew@shu.edu.cn

徐树公(通讯作者)　男,1969年生. 上海大学通信与信息工程学院教授. 主要研究方向为无线通信系统、模式识别与机器学习.
E-mail: shugong@shu.edu.cn

陆恒杰　男,1998年生. 上海大学通信与信息工程学院博士研究生. 主要研究方向为深度补全、人脸属性识别和人脸识别等.
E-mail: luhengjie@shu.edu.cn