计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(9):58−64 [doi: 10.15888/j.cnki.csa.007026]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 Android 和卷积神经网络的鸟类识别系统①

肖理想[1,2], 罗　泽[1]

[1](中国科学院 计算机网络信息中心, 北京 100190)
[2](中国科学院大学, 北京 100049)
通讯作者: 罗　泽, E-mail: luoze@cnic.cn

摘　要: 随着深度学习的广泛应用和智能移动设备的普及, 将深度学习的应用迁移到移动设备上已经成为一种新的趋势. 本文设计了一种基于安卓平台和轻量级卷积神经网络的鸟类识别系统, 该系统不依赖任何外部的计算资源和存储资源. 本文提出以轻量级卷积神经网络作为基础模型的三种模型融合方法, 分别是加权平均融合、双线型融合和多图片单模型融合. 本文详细介绍了三种融合方式的结构和优缺点, 并且给出了模型选择和超参数选择的一些方法. 实验结果表明模型融合的方式相比单模型而言, 识别精度有显著提高, 可以更好的应用到安卓移动设备上.

关键词: 深度学习; 卷积神经网络; 图像细粒度分类; 迁移学习; 模型融合

引用格式: 肖理想,罗泽.基于 Android 和卷积神经网络的鸟类识别系统.计算机系统应用,2019,28(9):58–64. http://www.c-s-a.org.cn/1003-3254/7026.html

Bird Identification System Based on Android and Convolutional Neural Network

XIAO Li-Xiang[1,2], LUO Ze[1]

[1](Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China)
[2](University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: With the widespread use of deep learning and the popularity of smart mobile devices, it has become a new trend that migrates deep learning applications to mobile devices. This study designs a bird identification system based on Android platform and lightweight convolutional neural network. The system does not rely on any external computing and storage resources. This study also proposes three model stacking methods based on lightweight convolutional neural network as the basic model, which is weighted average, bilinear stacking, and multi-picture and single model stacking. In this paper, we introduce three stacking methods' structure, advantages, and disadvantages in detail. And we also give some selection methods of hyperparameters through experiments. The experimental results show that the model stacking is much better than the single model, and the accuracy of the model has been significantly improved, which can be better applied to Android mobile devices.

Key words: deep learning; convolutional neural network; fine-grained classification of images; transfer learning; model stacking

1　前言

深度学习的快速发展, 已经在许多领域取得了巨大的成功. 一方面, 卷积神经网络 (Convolutional Neural Networks, CNNs)[1,2]在图像识别上取得了的巨大的进步, 另一方面, 移动手机当前已经成为人类日常生活的随身携带之物, 手机上的各式各样的应用程序 (Applica-

① 基金项目: 中国科学院科技服务网络计划 (STS 计划)(Y82E01); 国家科技部国家科技基础条件平台项目 (DKA2017-12-02-18)
Foundation item: CAS Science and Technology Service Network Initiative (STS Initiative) (Y82E01); MOST National Science & Technology Infrastructure Program (DKA2017-12-02-18)
收稿时间: 2019-02-25; 修改时间: 2019-03-08; 采用时间: 2019-03-11; csa 在线出版时间: 2019-09-05

tions, APPs) 能帮助人类进行更好的决策. 目前, 智能终端上基于卷积神经网络模型的应用程序绝大多数都是采用“客户端-服务端”模式, 但是该模式不仅强依赖于网络性能 (如网速等) 而且会导致用户数据泄露. 数据表明, 我国部分地区还不能实现网络的全覆盖, 这会对一些强依赖网络的应用程序造成很大影响. 例如, 科研工作者工作在无网络的野外、海洋等地区, 这些强依赖网络的应用程序就不能使用. 因此, 将卷积神经网络迁移到移动智能终端上直接运行是必须的也是必要的, 本文的目的就是将鸟类识别的卷积神经网络模型迁移到安卓设备上, 使其能够在任务网络情况下使用.

众所周知, 卷积神经网络在图像识别上取得了巨大的进展, 从 LeCun 等人提出 LeNet 网络模型, 再到 2012 年 AlexNet[3]在 ImageNet 比赛上夺得冠军, 之后越来越多的科研机构和大型科技公司都在卷积神经网络的研究上上投入巨大的精力. 最近几年, 从 VGGNets[4]、InceptionV3[5]再到 ResNet 等网络在图像识别上的准确率逐步提高, 已经超过人类的识别精度. 但是这些都基于粗粒度分类和云端服务器的模型, 由于移动设备的计算能力和存储能力有限, 基于移动设备的网络模型一直发展缓慢. 随着谷歌在 2017 年提出一种基于移动设备的图像识别的深度学习模型 MobileNets[6,7], 之后旷世科技进一步提出了 ShuffleNet 网络. 这些基于移动设备网络模型在计算和存储能力有限的移动设备上取得不错的效果, 本文主要基于 MobileNets 网络进行模型融合探索和实验.

细粒度识别是近年来图像识别的热门任务之一, 细粒度识别的难点在于其图像所属类别的粒度更为精细, 不容易定位图像的局部敏感特征. 许多学者都在尝试提高细粒度识别的精度, Lin TY 等[8]人提出 Bilinear CNN 模型进行细粒度识别, 论文提出使用两种 CNN 模型进行特征提取, 之后使用外积组合得到局部敏感特征. Fu JL 等[9]人提出将注意力 (attention) 机制加入到图像识别中, 使用分类网络和标注网络交替训练, 提高分类准确率.

神经网络模型迁移到移动设备上也有一些工作者做出过尝试, 例如, 梁淑芬等人[10]将轻量级的 LeNet 网络应用到移动设备上进行身份证号码的识别, 南京航空航天大学的刘程等人[11]基于向量之间的比较将人脸识别程序迁移到移动设备上进行打卡试验. 上述两种应用都取得的不错的效果, 但是 LeNet 网络处理能力有限很难应对大规模的图像分类, 尤其在细粒度图像识别上表现的更差; 基于向量比较的人脸识别在预测的时候主要是通过设置的关键点进行特征提取之后再进行向量的比较.

本文设计了一个基于卷积神经网络的鸟类识别系统, 使用基于移动设备的卷积神经网络模型进行模型融合. 考虑到细粒度识别[12,13]的特点, 使用不同特征的加权平均进行训练, 为了得到图像的局部特征使用双线型模型融合, 分析了大量的图像并提出使用物理裁剪的方式来近似替代标注工作, 融合多图像特征进行训练. 最后将实验之后的模型, 迁移到安卓设备上运行.

2 系统架构设计

2.1 整体结构设计

系统的整体架构如图 1 所示. 系统采用功能模块化进行结构设计, 降低系统的耦合度, 使得系统设计、调试和维护等操作简单化, 具有更多的灵活性, 使得系统能够同时支持在线和离线两种鉴定方式. 本文的功能模块化设计不仅考虑到整体的耦合性, 而且可以在系统层面进行优化加速, 系统通过轮询当前设备的网络状况, 根据网络状况提前预加载资源或者释放资源来提高系统的运行效率.

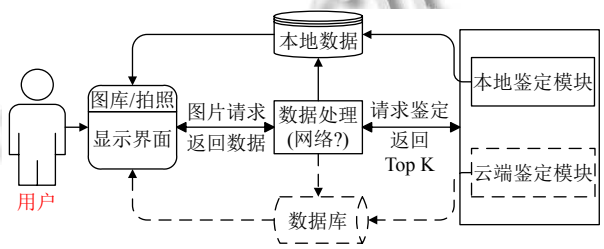

图 1 系统结构图

图 1 中分别用实线和虚线表示离线和在线两种鉴定方式, 可以看到这两种方式都通过数据处理模块进行调用, 这样可以根据用户的网络情况自动调用对应的服务, 给用户更好的体验效果. 当系统检测到移动设备上的网络状况很差的时候, 自动加载数据资源, 为用户接下来的离线鉴定做好准备, 系统通过网络状态轮询的方式, 当系统的网络状态良好时候, 释放加载的资源, 降低资源浪费率. 系统默认根据网络情况选择选择不同的服务. 具体的优化结构如图 2 所示.

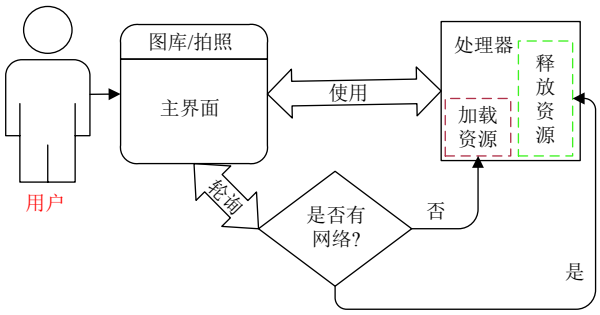

图 2　系统轮询优化

3　相关工作

3.1　相关工具

Tensorflow[14]是一个基于数据流编程 (dataflow programming) 的符号数学系统, 被广泛应用于各类机器学习 (machine learning) 算法的编程实现, 是目前使用最广泛的深度学习框架. TensorFlow Lite 是 Tensor Flow 针对移动和嵌入式设备的轻量级解决方案. 它使基于设备的机器学习模型预测具有低延迟和低功耗. TensorFlow Lite 还支持硬件加速的 Android 神经网络应用程序编程接口 (Application Programming Interface, API). Keras 是一个高层神经网络的 API, Keras 由纯 Python 编写而成并基于 TensorFlow、Theano 和 CNTK 后端. 由于 Keras 使用简单快速, 能够快速的开发深度学习模型, 受到了很多科研工作者的青睐.

3.2　迁移学习

本文使用迁移学习[15,16]训练模型, 基于 ImageNet 的权重进行预训练, 之后进行微调. 基于迁移学习的模型训练方式一般有两种, 第一, 将预训练的模型作为图片的特征提取器, 将整个任务作为一个多分类问题; 第二, 先训练顶层网络参数, 微调原有网络部分层的参数. 详细信息如表 1 所示.

表 1　迁移学习方式

方式	内容
迁移学习	预训练网络作为特征提取器; 重新训练全连接层参数
微调网络	预训练网络参数; 训练全连接层参数; 微调网络结构上层部分层参数

为了判断迁移学习模型提取特征的有效性, 本文通过 PCA[17]降维进行特征可视化, 随机选择三类鸟进行二维平面的显示, 显示结果如图 3 所示. 从图 3 中可以看出在二维平面上三类鸟的散点基本分散, 可以认为模型提取的特征是有效的特征.

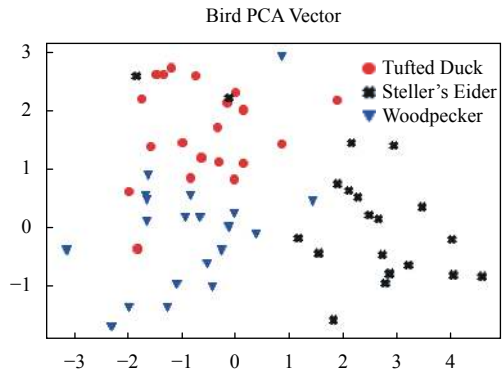

图 3　迁移学习特征降维图

3.3　数据存储

本系统的数据主要包括鸟类的基本信息和每个鸟的代表图. 由于大小的限制, 移动设备上不存储鸟类的图集和声音数据. 对于数据库中所有的鸟, 选择一个最具代表性的图片进行压缩至适中的缩略图来代表这个鸟进行显示. 本文提取数据库中的基本信息并存储在 Json 文件中, 将每个鸟的缩略图重命名为其对应的 id. 鸟类的基本信息如下:

```
{
  "1": {
    "id": "1",
    "name": "雪鹑",
    "latin": "Snow Partridge",
    "thumbnail": "bird_1.jpg",
    "describ": "中等体型 (35 厘米) ……"
  },
  ⋮
}.
```

4　模型融合方法

4.1　加权平均融合

本文使用迁移学习训练模型, 将卷积神经网络模型作为特征提取器, 加权平均融合采用一主一辅的策略进行模型融合, 主模型特征的权重是$1-a$, 辅模型特征的权重是$1-a$. 训练时特征的计算公式如下:

$$Vector = a^{*}MainVec + (1-a)^{*}AssisVec \tag{1}$$

训练完成后需要保存主模型和训练好的全连接层, 以及辅模型在训练数据上所有特征的平均值. 测试的

时候的使用特征的计算公式如下:

$$Vector = a^{*} MainVec + (1-a)^{*} AssistAvgVec \quad (2)$$

其中, $MainVec$表示主模型提取的特征向量, $AssistVec$是辅模型提取的特征向量, $AssistAvgVec$是辅模型在训练数据上的特征平均值向量. 加权平均的融合方式可以从方差的角度进行分析, 可以认为是在主模型特征中加入了部分扰动, 在图像特征中加入扰动就相当于在图像中加入噪声, 这种噪声可以认为测试数据是模糊的、缺失的低质量图像. 这种融合方式的优点是仅仅需要保存主模型和辅模型在训练数据上的平均特征向量, 可以很好的预测低质量的图像. 缺点是超参数a不好确定, 在实际使用中可以通过交叉验证得到最优的超参数. 模型结构如图 4 所示.

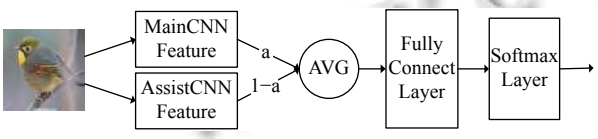

图 4 加权平均融合

4.2 双线型融合

加权平均融合是特征的加权平均, 从结构来看, 这是双线型模型的一种特例, 双线型融合的策略是使用两个卷积神经模型进行特征提取, 之后进行叉乘操作得到局部敏感特征. 由于叉乘之后的数据维度过高, 本文使用 PCA 进行降维处理, 模型的架构如图 5 所示.

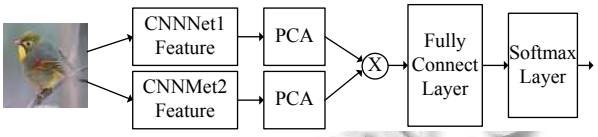

图 5 双线型模型训练方式

双线型融合模型的优点是使用叉乘扩充特征, 有可能得到局部敏感特征, 加权平均的方式关注的是低质量的图像, 双线型模型更关注细粒度分类的本身特点, 希望找到更精细的识别特征, 提升模型的准确率. 双线型融合的缺点是 PCA 的降维的参数不好设置, 考虑到叉乘之后的维度不能太大, 建议 PCA 的维度设置在[20, 30]之间, 同时可以使用交叉验证的方式, 选择最优的 PCA 维度. 另外一个缺点是双线型融合需要保存两个模型, 这使得模型的大小扩大一倍, 除此之外, 由于模型中使用了 PCA 降维, 使得整个模型不是端到端的模式.

4.3 多图片单模型融合

双线型模型只考虑了提高细粒度鸟类识别的精度问题, 模型却扩大一倍. 通过结构图可以得到, 双线型融合是对同一图像使用不同的特征提取方式, 目的是得到更精细的分类特征. 对于细粒度识别问题, 文献[9]中指出可以直接关注局部位置例如鸟喙和翅膀等, 例如使用 YOLO 等模型先标记出鸟喙和翅膀等位置, 裁剪出来局部位置, 使用同一个模型分别提取原始图像和局部位置图像特征. 考虑到移动设备的存储能力和计算能力, 不能直接使用 YOLO 等类似模型. 本文提出了一种近似的物理裁剪的替代方法, 如果使用随机标记裁剪的方式, 由于具有很大随机性的会对模型有很大的扰动. 本文分析了大量鸟的图像和用户使用智能手机的拍照习惯, 发现一般情况下鸟都会在图片的中心, 因此本文使用中心裁剪替代随机裁剪. 我们将这种融合方式叫做多图片单模型融合. 多图片单模型融合的优点是简单, 相对双线型融合网络模型没有增大, 适合在移动设备使用. 缺点是, 如果鸟的位置不在图像的中心, 会得到一个较差的结果. 具体的模型结构如图 6 所示.

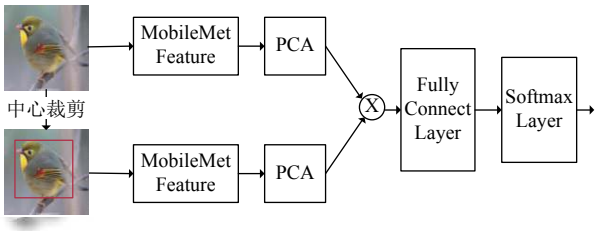

图 6 多图片单模型融合

5 实验结果分析

5.1 实验数据和环境

本文的实验数据是 100 类鸟, 训练数据共有 36,782 张鸟的图像, 测试数据和验证数据每类鸟各有 20 张图像. 本文使用两种测试环境分别是台式机测试环境和安卓手机测试环境, 台式机测试环境具体的参数是: Ubuntu16.04LTS 操作系统, 8 核 Intel Core i7 处理器, 12 GB 内存. 安卓测试环境具体参数是: Android 5.1 操作系统, 联发科 8 核处理器, 2 GB 运行内存.

5.2 加权平均融合

加权平均融合有两个难点, 分别是模型的选择和

超参数a的设置, 本文主模型选择使用 MobileNet 卷积神经网络模型, 辅模型选择 InceptionV3 卷积模型和 Xception 卷积模型进行对比实验, 而且对比了超参数$a = 0.9$和$a = 0.95$的实验结果. 具体实验结果如表 2 所示.

表 2 加权平均融合实验结果

模型	超参数 a	Top-1 准确率	Top-5 准确率	时间 (ms)	安卓时间 (ms)
InceptionV3	—	0.774	0.956	57.79	—
Xception	—	0.781	0.961	170.11	—
MobileNet	—	0.756	0.945	46.42	400
MobileNet+Xception	0.9	0.721	0.93	246.35	400
MobileNet+InceptionV3	0.9	0.727	0.932	125.67	400
MobileNet+Xception	0.95	0.731	0.935	242.47	400
MobileNet+InceptionV3	0.95	0.732	0.939	127.72	400

从表中可以看出, 当超参数$a = 0.95$相比$a = 0.9$, 融合模型的准确率更高. 从方差的角度来看, 辅模型的权重越大, 相当于加入的扰动方差越大, 对模型的结果影响较大, 本文的测试数据和训练数据都是高质量的图像, 因此辅模型的权重越小, 更能得到好的结果. 目前, 智能设备的拍摄的图像质量越来越高, 因此建议使用较小的辅模型权重. 加权平均的融合方式比单模型 MobileNet 的准确率更低的原因是因为加权平均融合加入了方差扰动, 当图片模糊的时候会取得更好的结果, 但是测试数据集中的图片质量较高, 加权只是增加了扰动, 不能得到更好的结果.

5.3 双线型融合

加权平均融合对模糊的图像更友好, 从结构上来看, 加权平均模型是双线型模型的一种特例. 加权平均融合区分主辅模型, 双线型融合没有模型的主次之分, 同时使用两种模型进行特征提取. 本文选择基于移动设备的网络模型 MobileNet 和 MobileNetV2 作为双线型融合方法的基特征提取器. PCA 的维度选择了 20 和 25 进行实验, 为了更好的对比实验结果, 本文还选择了 MobileNet 和 Xception 的双线型融合进行对比实验. 实验结果如表 3 所示.

表 3 双线型融合实验结果

模型	PCA 维度	Top-1 准确率	Top-5 准确率	时间 (ms)	安卓时间 (ms)
MobileNet+MobileNetV2	20	0.764	0.951	82.83	800
MobileNet+Xception	20	0.768	0.952	234.35	—
MobileNet+MobileNetV2	25	0.766	0.949	84.14	800
MobileNet+Xception	25	0.769	0.951	246.22	—

上文提到可以通过交叉验证得到最优的 PCA 维度, 这样虽然可以得到很好的结果, 但是训练速度却很慢. 本文给出一种简单的 PCA 维度的选择方法, 可以通过可视化叉乘之后的特征来判断选择何种维度进行降维, 图 7 从左到右依次是原图、PCA 维度等于 20 的热力图和 PCA 维度等于 25 的热力图. 从图 7 中的热力图可以看出 PCA 维度等于 20 维的时候, 特征更关注鸟喙的位置, 可以认为得到了更好的敏感特征.

5.4 多图片单模型融合

双线型融合的优点在于能够得到局部敏感特征, 缺点是需要保存两个模型. 多图片模型融合基于 Bounding Box 思想, 使用物理方式裁剪局部敏感位置, 是一种基于图像的双线型融合方式. 本文使用 MobileNetV2 模型进行对比实验, 分别对比了随机裁剪和中心裁剪两种裁剪方式, 设置裁剪框的大小是原图的 1/2. 实验结果如表 4 所示.

图 7 PCA 维度热力图

表 4 多图片单模型融合实验结果

模型	裁剪方式	Top-1 准确率	Top-5 准确率	时间 (ms)	安卓时间 (ms)
MobileNetV2	随机化裁剪	0.746	0.939	101.92	550
MobileNetV2	中心裁剪	0.759	0.954	98.45	600

从表 4 中可以看出中心裁剪得到的结果更好, 主要因为随机裁剪具有很大地不稳定性, 并且训练和测试数据中鸟一般出现在图像的中心位置, 这也更符合用户的习惯, 相对随机裁剪的方式更稳定. 本文给出了中心裁剪、随机裁剪和原图叉乘之后的热力图结果, 图 8 从左到右依次是原图裁剪、中心裁剪的热力图和随机裁剪的热力图. 从图 8 可以看出, 中心裁剪的方式更多的关注翅膀和头的位置, 得到了相对较好的局部特征.

5.5 融合算法对比

3 种不同的融合方式, 各有其优缺点. 加权平均融合对模糊图像有很好的表现, 但对于高清图像误差较大; 双线型融合可以提取图像的局部敏感特征便于细粒度识别, 但对于移动设备而言它的模型相对较大; 多图片单模型是一种基于图像的双线型模型, 在提取敏感特征的同时不会增加模型大小, 是一种较好的算法模型, 缺点是裁剪方式和图片有很大关系. 表 5 给出 3 种融合方式最优实验结果的对比结果.

图 8 裁剪方式热力图

表 5 融合方式最优实验结果对比

模型	融合方式	Top-1 准确率	Top-5 准确率	时间 (ms)	安卓时间 (ms)
MobileNet+InceptionV3	加权平均	0.732	0.939	127.72	400
MobileNet+MobileNetV2	双线型	0.766	0.949	84.14	800
MobileNetV2	多图片单模型	0.759	0.954	98.45	600

6 系统设计

6.1 模型选择

本文通过对比上述所有融合方法的实验结果, 综合考虑移动设备的计算能力和存储能力以及模型融合的准确率, 系统最后选择了基于多图片单模型融合的方式, 使用这种方式训练得到 MobileNetV2 模型投入使用. 由于多图片单模型融合对图像中鸟的位置很敏感, 为了提高 APP 的友好性, 系统也提供了原始 MobileNet 的模型供用户使用.

6.2 系统测试

用户进入主界面的时, 系统自动进行网络状态检测, 如果检测到没有网络或者网络状态很差, 就将模型和数据文件提前加载到内存中, 当用户使用本地模型进行测试的时候, 直接使用加载好的模型和鸟类的基本信息, 可以直接进行预测. 网络状况的检测是轮询进行的, 当检测到网络存在时, 会自动调用云端的模型, 这时就释放已经加载好的资源, 减少内存使用. 离线模型的运行结果如图 9 所示.

图 9 系统测试图

7 结论与展望

本文构建了一个基于安卓平台和卷积神经网络的离线鸟类识别系统, 提出了基于细粒度识别的三种模型融合方式, 分别使用加权平均融合、双线型融合和多图片单模型融合的方法在鸟类数据上进行了实验. 本文训练的模型直接运行在移动设备上, 不依赖任何外部的计算资源和存储资源. 为了进一步提高鸟类识别的准确率, 采用融合思想进行训练模型. 从整体来看, 本文使用迁移学习降低模型训练的时间, 得到相对较优的算法模型, 并且在移动设备上取得了预期的效果. 本文虽然取得了不错的效果, 但是将深度学习模型迁移到移动设备[18]上还有很长的路要走. 为了在计算和存储能力都有限的移动设备上运行深度学习模型还需要进一步的研究, 例如提出更优的适合移动设备的网络结构, 提高移动设备的计算能力和储存能力.

参考文献

1 周飞燕, 金林鹏, 董军. 卷积神经网络研究综述. 计算机学报, 2017, 40(6): 1229–1251. [doi: 10.11897/SP.J.1016.2017.01229]

2 卢宏涛, 张秦川. 深度卷积神经网络在计算机视觉中的应用研究综述. 数据采集与处理, 2016, 31(1): 1–17.

3 Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. Proceedings of the 25th International Conference on Neural Information Processing Systems. Lake Tahoe, NV, USA. 2012. 1097–1105.

4 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv: 1409.1556, 2014.

5 Szegedy C, Vanhoucke V, Ioffe S, *et al*. Rethinking the inception architecture for computer vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 2818–2826.

6 Howard A G, Zhu ML, Chen B, *et al*. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv: 1704.04861, 2017.

7 Sandler M, Howard A, Zhu ML, *et al*. MobileNetV2: Inverted residuals and linear bottlenecks. Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA. 2018. 4510–4520.

8 Lin TY, RoyChowdhury A, Maji S. Bilinear CNN models for fine-grained visual recognition. Proceedings of the IEEE International Conference on Computer Vision. Santiago, Chile. 2015. 1449–1457.

9 Fu JL, Zheng HL, Mei T. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA. 2017. 4476–4484.

10 梁淑芬, 胡帅花, 秦传波, 等. 基于深度学习的数字识别模块在安卓系统的实现. 五邑大学学报 (自然科学版), 2017, 31(1): 40–45. [doi: 10.3969/j.issn.1006-7302.2017.01.009]

11 刘程, 谭晓阳. 一种基于深度学习的移动端人脸验证系统. 计算机与现代化, 2018, (2): 107–111, 117. [doi: 10.3969/j.issn.1006-2475.2018.02.022]

12 陈淑娴, 刘建明. 基于部位特征和全局特征的物体细粒度识别. 计算机与现代化, 2017, (10): 1–4, 9. [doi: 10.3969/j.issn.1006-2475.2017.10.001]

13 李新叶, 王光陛. 基于卷积神经网络语义检测的细粒度鸟类识别. 科学技术与工程, 2018, 18(10): 240–244. [doi: 10.3969/j.issn.1671-1815.2018.10.041]

14 Abadi M, Agarwal A, Barham P, *et al*. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv: 1603.04467v1, 2016.

15 庄福振, 罗平, 何清, 等. 迁移学习研究进展. 软件学报, 2015, 26(1): 26–39.

16 Pan S J, Yang Q. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(10): 1345–1359. [doi: 10.1109/TKDE.2009.191]

17 Wold S, Esbensen K, Geladi P. Principal component analysis. Chemometrics and Intelligent Laboratory Systems, 1987, 2(1-3): 37–52. [doi: 10.1016/0169-7439(87)80084-9]

18 雷杰, 高鑫, 宋杰, 等. 深度网络模型压缩综述. 软件学报, 2018, 29(2): 251–266. [doi: 10.13328/j.cnki.jos.005428]