

基于卷积神经网络的国兰种类识别系统

张子赫¹, 赵吟吟²

¹浙江农林大学, 信息工程学院, 浙江 杭州

²浙江农林大学, 环境与资源学院, 浙江 杭州

Email: 493688862@qq.com

收稿日期: 2020年11月27日; 录用日期: 2020年12月21日; 发布日期: 2020年12月28日

摘要

针对兰花品种众多, 外表相似, 导致没法准确识别兰花的种类这一问题, 本文构建了基于卷积神经网络的国兰识别模型, 设计并实现了模型在移动端的应用。文中通过多种途径完成常见国兰数据集的创建, 进而以Inception-ResNet-v2为卷积神经网络预训练模型, 使用迁移学习技术完成模型训练, 并基于Android平台完成系统的开发和测试。测试结果显示, 对传统国兰图像分类识别准确率达到91.51%。

关键词

国兰, Inception-ResNet-v2网络, 图像识别, Android

Chinese Orchid Species Recognition System Based on Convolutional Neural Network

Zihe Zhang¹, Yinyin Zhao²

¹College of Information Engineering, Zhejiang A&F University, Hangzhou Zhejiang

²College of Environmental and Resource Sciences, Zhejiang A&F University, Hangzhou Zhejiang

Email: 493688862@qq.com

Received: Nov. 27th, 2020; accepted: Dec. 21st, 2020; published: Dec. 28th, 2020

Abstract

Aiming at the problem that there are many varieties of orchids with similar appearances, which makes it impossible to accurately identify the types of orchids, a Chinese orchid recognition model

based on convolutional neural network is constructed, and its transplantation to the mobile terminal is designed and implemented. First, this paper completes the creation of the common Chinese orchid data set through multiple channels, and then uses the Inception-ResNet-v2 convolutional neural network as the pre-training model, uses the transfer learning technology to complete the model construction, and develops and tests the system based on the Android platform. Through experiments and actual tests, the accuracy of classification and recognition of traditional Chinese orchid images reached 91.51%.

Keywords

Chinese Orchid, Inception-Resnet-V2 Network, Image Identification, Android

Copyright © 2020 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着时代的发展,人们越来越注重精神世界的培养。国人对于兰花的需求与日俱增,基于兰花的观赏价值以及药用价值[1],加上国际市场的影响,我国兰花市场的发展十分迅速。根据海关数据统计,2017年我国兰花出口量约为六千万株,带来了巨大的经济价值。因而正确区分兰花的种类,确定其合理的价值十分重要。

近年来,深度学习发展迅速,基于各类神经网络的图像识别研究越来越广泛,然而由于数据量的缺乏,兰花方面的识别存在重大空缺。针对这一问题,本文通过网络搜集、实地拍摄和书籍扫描三种途径完成常见国兰数据集的创建,该数据集由春兰、蕙兰、建兰、墨兰、寒兰5类常见国兰种类构建得到。在此基础上基于迁移学习完成 Inception-ResNet-v2 [2]深度卷积模型的国兰识别训练,并将训练好的模型移植到移动端,验证兰花识别的效果。

2. Inception-ResNet-v2 网络

Inception-ResNet-v2吸收了 Inception V4和 ResNet [3]的优点,在 ImageNet数据集上的测试更加准确。该模型将 Inception-ResNet-A、Inception-ResNet-B、Inception-ResNet-C 3种 Inception-ResNet块加入直连,从而使得通道多样化。相较 Inception V4, Inception-ResNet-v2参数少,收敛快,同时对机器的性能要求也有一定的降低,可以在同实验环境下进行更高的参数设置; Inception-ResNet-v2设计了 BN层,这样可以有效实现数据的输出不会发生偏移,数据规范化传输可以有效提高识别的准确度。该模型卷积核的体积很小,但是功能却很强大。Inception-ResNet-v2网络结构如图1所示。

3. 数据集

针对国兰图像数据集空缺的现状,本文通过中国兰花交易网下载、经典书籍《中国兰花精粹》扫描以及实地拍摄三种途径相结合初步获取5类3000余张国兰图像。经过筛选和预处理,保留1982张高质量的国兰图像进行数据集的构建。为解决因国兰图像不足造成的过拟合现象,本文采用水平翻转,随机裁剪,添加高斯噪声:对原始图像添加了30%的高斯噪声,图像亮度调节(调亮或者调暗),随机旋转变换(60°、90°、180°、270°)五种数据增强方式[4]对数据集进行扩充,最终得到12,150张国兰图像。以春建兰为例,数据增强前后对比如图2所示。

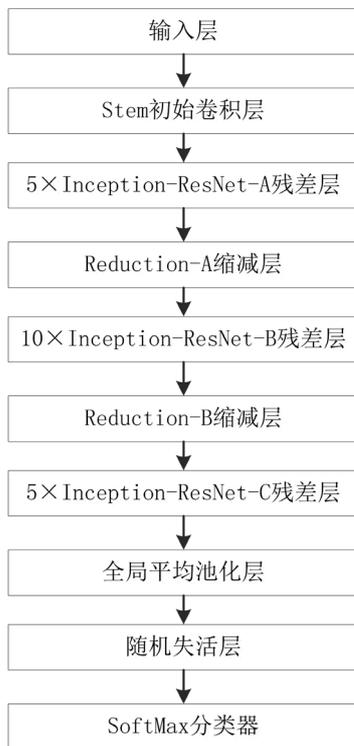
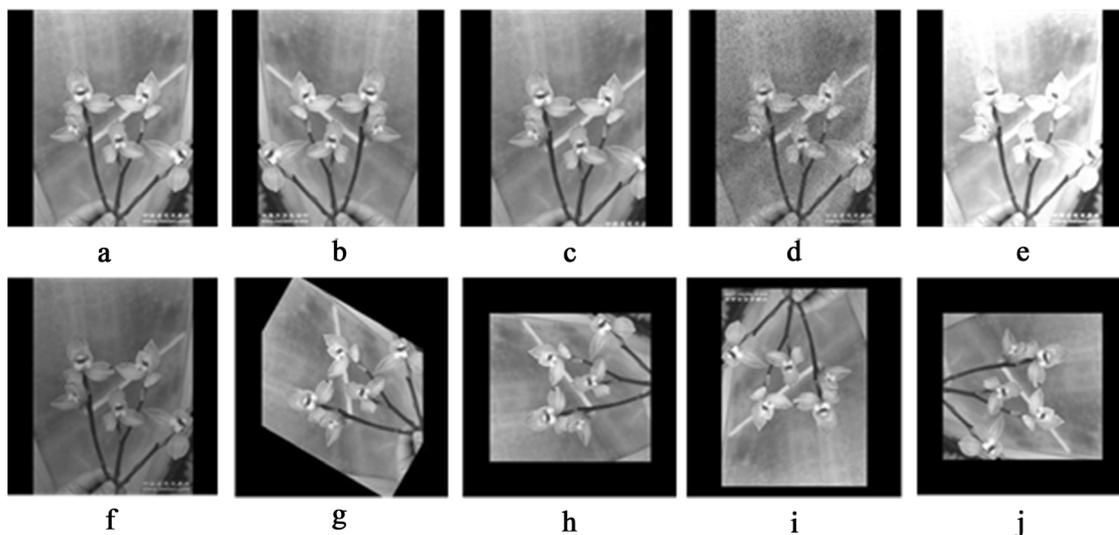


Figure 1. Inception-Resnet-v2 network structure
图 1. Inception-Resnet-v2 网络结构



注: a 是未增强的图片, b-j 是增强后的图片, 对应的增强方法分别是水平翻转、随机剪裁、添加高斯噪声、调亮、调暗、旋转 60°、旋转 90°、旋转 180°、旋转 270°。

Figure 2. Comparison of original and enhanced images of *Cymbidium tortisepalum* var. *longibracteatum*
图 2. 春剑兰图像增强前后对比图

4. 实验及结果分析

为检验模型效果, 分别使用 VGG16、VGG19、InceptionV3、InceptionV4、ResNet-50、ResNet-101、ResNet-152、InceptionResNet-v2, 8 种预训练的卷积神经网络模型进行兰花识别实验, 实验方案编号记为

1-8, 实验结果如表 1 所示。其中卷积神经网络优化器均为 Adam [5], 迁移训练策略为重新训练预训练网络模型的全部网络层。

Table 1. Comparison of different CNN models

表 1. 不同网络模型比较

实验编号 Experiment No.	网络模型 network models	Top-1 准确率/% Top-1 accuracy/%	平均训练时间/s Average Training time/s
1	VGG16	88.33	0.675
2	VGG19	86.20	0.625
3	InceptionV3	87.95	0.297
4	InceptionV4	89.20	0.547
5	ResNet-50	85.87	0.187
6	ResNet-101	88.58	0.312
7	ResNet-152	89.37	0.453
8	Inception-ResNet-V2	91.51	0.676

从表 1 中可以看出, ResNet-50 网络模型训练结果准确率最低, 而 Inception-ResNet-v2 最高, 可以发现当模型深度逐渐上升时, 模型的准确率是总体呈现上升趋势的, 这也说明了卷积层的深度会对模型识别的准确率造成一定的影响, 更深的网络结构会有更高的识别准确率。

本文使用识别效果最佳的 Inception-ResNet-v2 网络作为本文的预训练模型, 设置初始学习率为 0.01, 批处理量为 16, 选取 Relu [6] 函数作为激活函数, 并在此基础上, 使用 Adam 优化算法对网络参数进行优化, 采用迁移学习方法重新训练网络所有参数层。

本文研究目标是要完成国兰图像分类识别问题, 因此采用多分类交叉熵(Cross entropy)作为损失函数, 其表达式如公式 1 所示:

$$C_{loss} = -\frac{1}{m} \sum_{j=1}^m \sum_{i=1}^n y_{ji} \log(\hat{y}_{ji}) \quad (1)$$

式中, m 为当前一批输入网络的国兰样本数量, n 为国兰类别的总数, y_{ji} 为国兰图像真实的标签, \hat{y}_{ji} 为国兰图像预测的标签, C_{loss} 为损失值。交叉熵刻画了实际输出概率与期望输出概率分布之间的距离, 交叉熵的值越小, 表示模型训练过程的中学习效果越好。

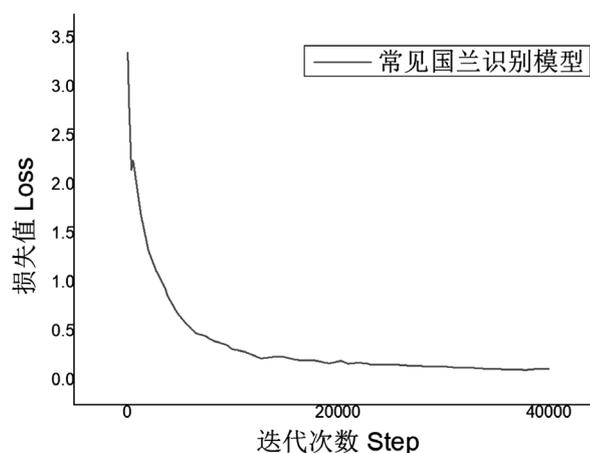


Figure 3. Loss curve of common Chinese orchid data set

图 3. 常见国兰数据集损失曲线

由图3训练结果可知,随着迭代次数的增加,交叉熵损失函数呈整体下降趋势,当迭代次数达到40,000步时,交叉熵损失函数基本完成收敛,故选取40,000步作为本模型训练的迭代次数。经测试,在迭代结束时交叉熵损失值为0.1165,经验证对应的准确率为91.51%,具有较高的识别准确率,可以辅助人们更好的识别兰花种类,达到预期目标。

5. 系统设计与实现

本文国兰识别系统的流程如图4所示,主要由图片获取模块和识别模块组成,最终由信息展示模块输出。通过前文对比实验结果展示,采用迁移学习训练 Inception-ResNet-v2 预训练模型的全部参数可以达到识别精度高并且训练时间短的要求,其中主要利用该模型卷积层的特征提取功能完成对国兰图像颜色花型等局部特征的提取。

此系统中,用户可以通过手机拍摄和相册上传两种途径将国兰图片上传到 Web 服务器端,此后服务端会通过国兰模型对接收到的图片进行识别分类。并根据识别结果将对应的国兰品种详情以及相应的养护知识一起打包发送给手机端屏幕显示。

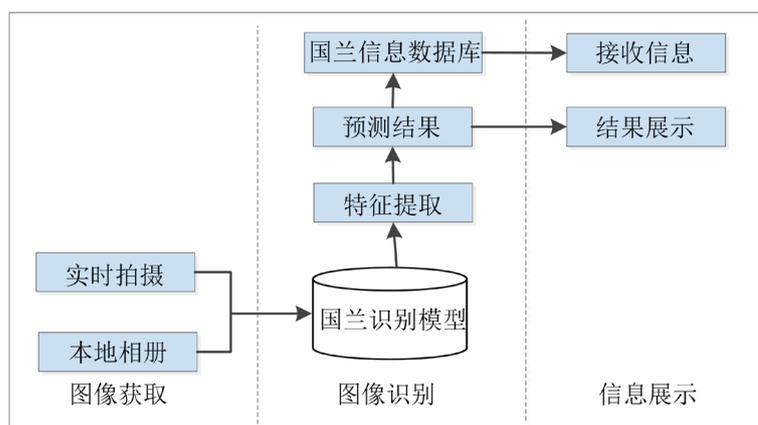


Figure 4. Flow chart of Chinese orchid identification system

图4. 国兰识别系统流程图

5.1. 图像获取模块

图像获取模块主要负责国兰图像的采集[7],以供识别模块调用识别,该模块需能调用智能手机的相机拍摄功能和手机本地相册读取功能,其工作流程如图5所示。

OpenCV 提供的库函数可以完成图像采集部分,主要是由于它所在的图像获取模块由摄像头承接主要任务,而该函数可以很好地完成对摄像头的调用[8]。OpenCV 是一个可以在很多开发环境上进行运行的开源视觉库,这是由于很大数目的 Java 接口存在的结构,而且 API 接口函数获得方法比较简单即在线获得,还能很轻松的对 Android 应用主要使用的 Android NDK [9]库进行调用。结合 JNI 接口实现本地功能代码,由 Android NDK 对其进行编译,生成可由 Java 调用的动态库文件,该动态库文件可被打包编译,以实现国兰识别 Android 程序。

5.2. 图像识别模块

图像识别模块的组成主要有三部分,包括模型的部署、结果的预测和信息的查询这几个部分。这一模块也可以算得上是本应用程序的核心模块,工作流程如图6所示。在接受到国兰图像后,服务器端会将收到的图像输入到国兰识别模型中去,然后根据预测出的结果到数据库中寻找对应种类的国兰信息和

养护信息等, 其中国兰识别模型部分由前文提到的改进的 Inception-ResNet-v2 网络国兰图像识别模型来承接。

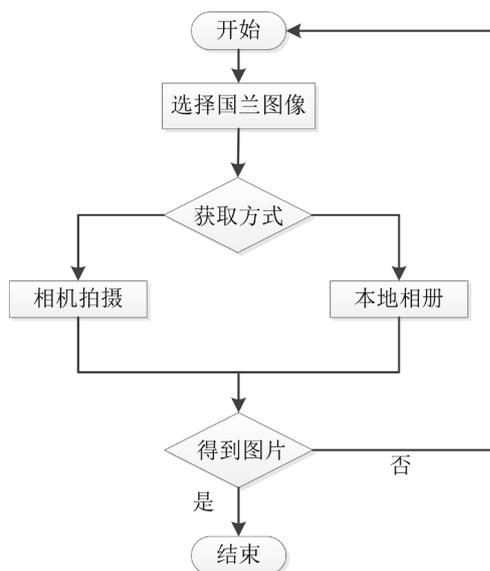


Figure 5. Flow chart of image acquisition module

图 5. 图像获取模块流程图

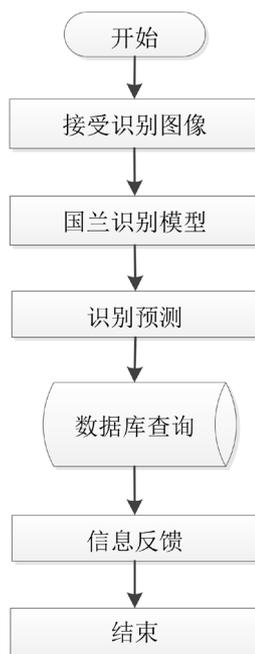


Figure 6. Flow chart of Chinese orchid identification module

图 6. 国兰识别模块流程图

5.3. 信息展示模块

信息展示模块的组成相对简单, 主要包含接受和展示两部分, 工作流程见图 7。接受部分首先需要判断网络的状况是否良好, 如果网络连接有问题, 则直接显示出错, 并提示及时检查网络连接。如果检

查网络状况正常, 则接受从 Web 服务器端[10]传输来的关于国兰图像种类的详情以及对应养护知识, 显示到屏幕上, 完成本次识别操作。

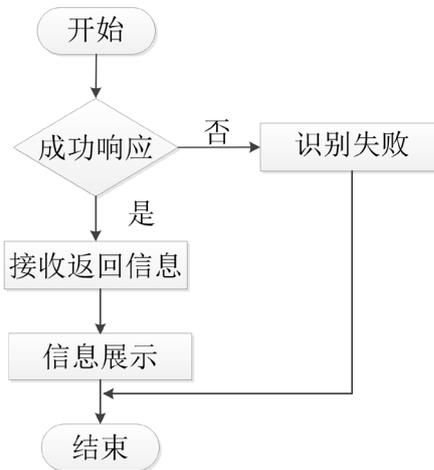


Figure 7. Flow chart of the result display module
图 7. 结果展示模块流程图

5.4. 系统界面

为了对该系统应用进行测试, 并且对本研究的实时便捷性进行验证。使用型号为 OPPO R9s Plus, 搭载 Android 6.0.1 操作系统的智能手机来完成本次对系统的测试, 如果发现不足, 及时进行改进和调整。图 8 展示了系统运行的所有界面。整体页面设计相对简单清晰且易于操作, 满足设计该应用程序的初衷。



Figure 8. Test chart of Chinese orchid identification system
图 8. 国兰识别系统测试图

使用者点击开始识别之后, 系统会跳转到相机功能页面进行拍照。拍照完成后, 系统对接收到的兰花图像自动分类查询, 并将对应的国兰种类信息等反馈到详情页面。或者根据需求选择本地相册中的兰花图像进行识别, 只需在拍照识别界面点击左下角的相册按钮便可进入国兰相册界面进行选择, 选定待识别图像后, 点击确定按钮便可自动识别并获得对应的兰花种类详情和养护信息。也可以对相册中的待识别图像重新进行选择, 只需点击重选按钮即可撤回操作。兰花详情页面, 可以点击页面上的养护知识按获取该种类兰花更多的养护知识。实验显示, 从选定图片到获得对应的种类信息大概需要 2.5 秒, 表明该系统较快速便捷, 达到了设计的预期。

6. 结束语

本文通过多种途径获取国兰图像完成了常见国兰图像数据集的构建, 并且基于 Inception-ResNet-v2 网络和迁移学习优化训练完成国兰识别模型。经过实验表明, 所设计的模型对兰花品种具有较高的识别准确率, 填补了深度学习应用于国兰识别的空缺, 同时 Android 端的应用能够达到辅助人们进行国兰识别的目的。通过多次实验验证, 该系统在目前搭载的 Android 6.0.1 操作系统上具有较强的稳定性, 识别准确率较为理想。但由于 IOS 操作系统的不开源问题, 该系统的兼容性还有待进一步提升, 这也是未来的研究的重点。

参考文献

- [1] 邵伟丽. 福建省野生兰科植物种质资源调查与保育策略研究[D]: [硕士学位论文]. 福州: 福建农林大学, 2008.
- [2] Szegedy, C., Ioffe, S., Vanhoucke, V., et al. (2016) Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *Thirty-First AAAI Conference on Artificial Intelligence*, 4278-4284.
- [3] He, K., Zhang, X., Ren, S., et al. (2016) Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 770-778.
<https://doi.org/10.1109/CVPR.2016.90>
- [4] 杨明欣, 张耀光, 刘涛. 基于卷积神经网络的玉米病害小样本识别研究[J/OL]. 石家庄: 中国生态农业学报(中英文): 1-9. <https://doi.org/10.13930/j.cnki.cjca.200375>, 2020-11-19.
- [5] 汤焯, 陆卫忠, 陈成, 等. 基于 Adam 算法和神经网络的照度计算方法[J]. 照明工程学报, 2019, 30(2): 50-54.
- [6] 郭子琰, 舒心, 刘常燕, 等. 基于 ReLU 函数的卷积神经网络的花卉识别算法[J]. 计算机技术与发展, 2018, 28(5): 154-157.
- [7] 周毅敏, 陈榕. Dalvik 虚拟机进程模型分析[J]. 计算机技术与发展, 2010, 20(2): 83-86.
- [8] Sivkov, S., Novikov, L., Romanova, G., et al. (2020) The Algorithm Development for Operation of a Computer Vision System via the OpenCV Library. *Procedia Computer Science*, **169**, 662-667.
<https://doi.org/10.1016/j.procs.2020.02.193>
- [9] 岳智慧. Android 平台上的细分插值算法研究[D]: [硕士学位论文]. 郑州: 郑州大学, 2019.
- [10] 李群. 基于 OkHttp 的文件传输设计与实现[J]. 电子技术与软件工程, 2018(13): 180-181.