

# 基于 Android 系统的储粮害虫图像识别软件设计与实现

赵彬宇, 周慧玲

(北京邮电大学自动化学院, 北京 100876)

摘要: 本文基于 Android 系统, 设计实现了一款储粮害虫图像识别软件。采用网络参数较少的 Resnet 模型训练储粮害虫图像识别算法, 实现了 6 类 10 种主要储粮害虫图像的种类识别, 平均识别准确率达到 0.9 左右。将训练好的算法导出为 pb 文件, 移植到 Android 手机上, 并采用 TensorFlow Lite 框架调用算法模型, 实现了在手机端本地识别储粮害虫图像。结果表明, 在手机端本地识别储粮害虫种类的方式具有可行性。相比在线识别的方式, 手机端本地识别可以免受网络状况和服务器性能的制约。该软件可以成为粮库保管人员和广大储粮农户的有益帮手。

关键词: 虫情监测; 储粮害虫; 图像识别; Resnet 算法; Android

中图分类号: TP311.1

## Design and Implementation of Android Software for Image Recognition of Stored Grain insects

Zhao Binyu, Zhou Huiling

(Automation school, Beijing University of Posts and Telecommunications, Beijing 100876)

Abstract: This paper designed and implemented a stored grain insect image recognition mobile phone software based on the Android system. The Resnet model with less network parameters was used to train a stored grain insect image recognition algorithm. The algorithm can identify 10 kinds of stored grain insects in 6 categories with an average accuracy of 0.9. In this paper, the trained algorithm is exported to pb file, and the file was transplanted to Android mobile phone. Then TensorFlow Lite was used to call the algorithm model, which realized the identification of the stored grain insect image on the Android mobile phone. The results show that it is feasible to identify the species of stored grain pests locally on the mobile phone. Compared with the online identification method, the local identification on the mobile phone can be protected from network conditions and server performance. This Android software could be an assistant for grain storage personnel and the majority of grain storage farmers.

Keywords: insect monitoring; stored grain insect; image recognition; Resnet algorithm; Android

## 0 引言

粮食在长期储存过程中会受到害虫的侵扰, 并且我国粮库储存粮食数量大, 受到害虫污染机会较大, 控制不好会使害虫繁殖迅速, 使粮食遭受大量损失<sup>[1]</sup>, 我国每年都要花费巨大的人力、财力用于保证粮库储粮安全<sup>[2]</sup>。为了实现储粮害虫情况的自动化监测, 许多研究者提出了基于图像、声学、近红外等不同技术的储粮害虫监测方法<sup>[3]</sup>。其中, 基于图像识别技

基金项目: 粮食公益性行业科研专项研究 (201513002-02)

作者简介: 赵彬宇 (1993-), 男, 硕士研究生在读, 主要研究方向: 测控技术与智能系统

通信联系人: 周慧玲 (1965-), 女, 教授, 主要研究方向: 嵌入式系统与测控网络. E-mail: huiling@bupt.edu.cn

40 术实现储粮害虫的定位和种类识别研究取得了快速发展，并且取得了一定程度的应用<sup>[4-6]</sup>。  
刘汉生设计并研发了一套陷阱式储粮害虫信息采集终端和系统<sup>[7]</sup>，通过服务器端的软件系统  
控制硬件设备定时采集粮仓内的害虫图像，并通过图像识别程序识别图片中害虫的种类和数  
量。这种方式采用固定式的硬件设备采集害虫图片，不具有便携性。

45 近年来随着移动互联网的发展，智能手机在人们学习和生活中扮演着越来越重要的角  
色。随着手机性能的快速提高，拍摄的图像越来越清晰，这使得研发手机端的便捷式害虫识  
别软件成为可能。目前图像识别类手机软件一般采用在线的方式进行识别，即使用手机拍照，  
然后将图片发送至服务器进行识别。这种方式要求手机连接网络，识别速度容易受到网络状  
况的影响。并且当识别任务并发量大时，会对服务器系统造成压力，需要增加服务器的数量，  
硬件成本较高。

50 本文基于 Android 手机系统，利用深度学习图像识别技术和 TensorFlow Lite 框架，设计  
实现了一款 Android 手机储粮害虫图像识别软件，实现了 6 类 10 种储粮害虫在 Android 手  
机本地的图像识别功能，可以帮助粮库保管员和广大储粮农户在储粮过程中快速鉴别害虫种  
类。

## 1 算法方案设计

55 传统的害虫图像识别算法基于害虫的形状特征、颜色特征和纹理特征，容易受到拍照角  
度和环境因素的影响。自从 2012 年 Alex Net<sup>[8]</sup>在 ImageNet 挑战赛中拔得头筹，使用卷积神  
经网络进行图像识别的技术研究得到快速发展，图像识别算法的性能大大提升。因此本文采  
用深度学习图像识别技术实现害虫图像的种类识别。

60 卷积神经网络是一种层次模型，其中每层包含大量的网络参数。在算法进行识别时，对  
这些参数进行计算，需要较高的计算能力和较大的内存空间。以具有 16 层网络的 VGG 模  
型<sup>[9]</sup>为例，其具有 138 兆个网络参数，运行时占用内存在 500MB 以上，在手机上运行难度  
较大。因此，为了能够在手机上运行害虫图像识别算法程序，本文采用了一种网络参数较少  
的 Resnet-50 模型<sup>[10]</sup>作为害虫图像识别算法，该模型具有 25.5 兆个网络参数，占用约 100MB  
内存空间。

## 65 2 算法训练和评估

### 2.1 储粮害虫图像数据集

鉴于储粮害虫的危害程度、分布范围等因素，本文选取了包含 6 类 10 种主要储粮害虫  
的图像数据集作为算法研究基础，这些害虫分别是：象甲类害虫（米象、玉米象）、谷蠹、  
扁谷盗类害虫（长角扁谷盗、锈赤扁谷盗、土耳其扁谷盗）、拟谷盗类害虫（赤拟谷盗、杂  
70 拟谷盗）、锯谷盗和烟草甲。

本文选用的数据集包含以白色纸板为背景的害虫图像和以黄色粘虫板为背景的害虫图  
像，共 117,424 张。每张图像中包含单头害虫，图像尺寸约为 100×100 像素。将图像数据

集分为训练集和测试集，如表 1 所示。图 1 所示为 6 类害虫图像示例。

表 1 图像数据集划分

Tab. 1 Division of image dataset

	象甲类	谷蠹	扁谷盗类	拟谷盗类	锯谷盗	烟草甲	总计
训练集 (头)	12,960	10,197	14,263	9,137	10,338	5,803	62,698
测试集 (头)	12,956	10,200	14,263	1,169	10,336	5,802	54,726
总计 (头)	25,916	20,397	28,526	10,306	20,674	11,605	117,424



图 1 6 类害虫单头图像示例

Fig. 1 image examples of stored grain insect in 6 categories

## 2.2 Resnet 算法

一般来说，神经网络的深度对模型的性能至关重要，随着网络层数的增加，网络可以提取更加复杂的特征模式，所以当网络更深时理论上可以取得更好的结果。但是实验发现，随着网络的加深，出现了退化问题（Degradation problem）：网络准确度出现饱和，甚至出现下降。

Resnet 是一种深度残差网络（Deep residual network），它引入图 2 所示的残差结构。ResNet 提出了两种映射：一种是恒等映射（identity mapping），指的是图中的“x”，另一种残差映射（residual mapping），指的是图中的“F(x)”。因此最后的输出为两者的叠加，即“y=F(x)+x”。残差指 y-x，即“F(x)”。这种结构能够解决网络退化的问题，使得网络的深度进一步增加，提取到更加抽象的特征信息，从而拥有更好的分类性能。

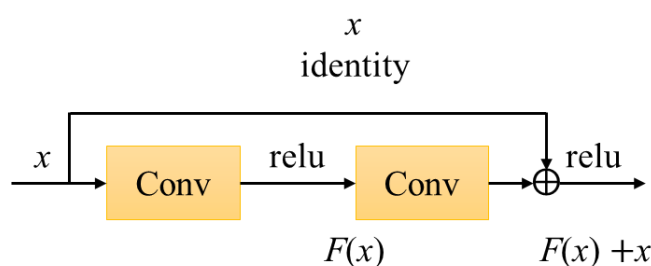
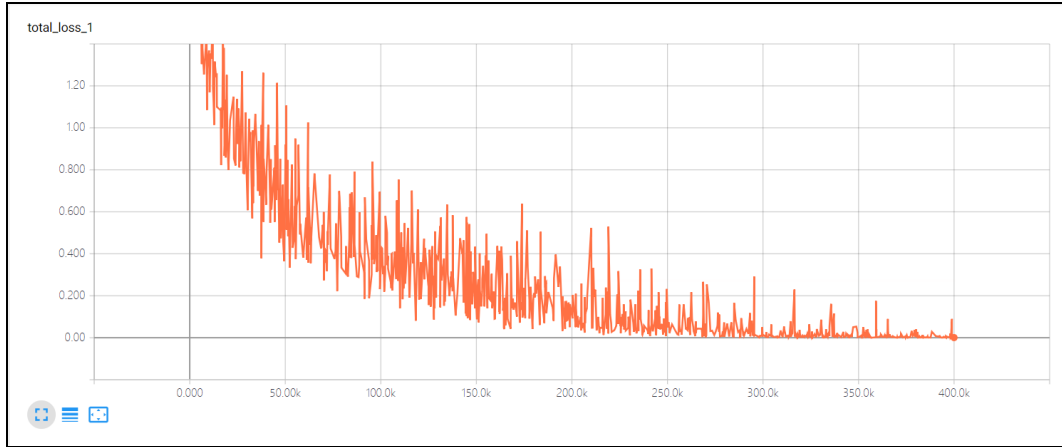


图 2 残差结构

Fig. 2 Residual structure

## 2.3 算法训练过程

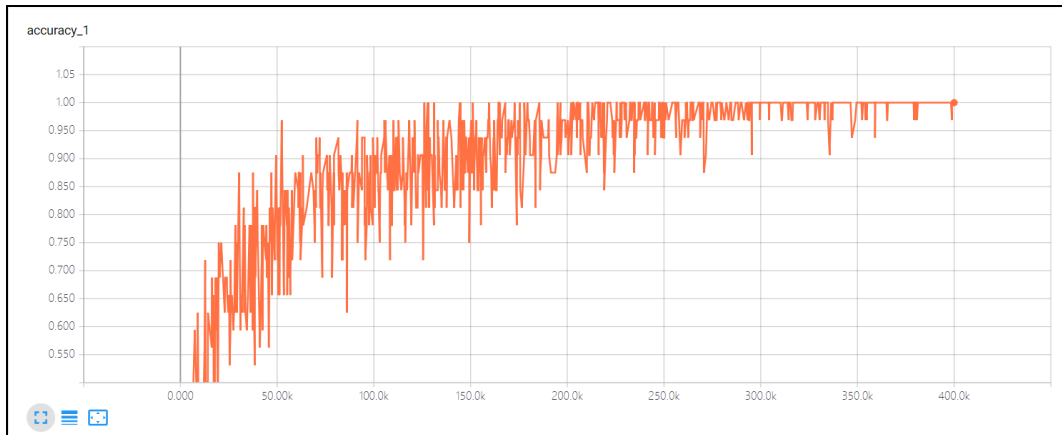
本文使用 TensorFlow 框架进行算法的开发，算法训练时调整图像的输入大小为 100×100 像素，batch size 为 32，迭代次数为 40 万次，初始学习率为 0.001。图 3 为训练过程中的 loss 变化曲线，图 4 为准确率变化曲线。训练结束后，将模型导出为 pb 格式的文件，设置输入节点名称为“shuffle\_batch”，输出节点名称为“Softmax”，以便软件对于算法的调用。



100

图 3 训练过程中 Loss 的变化曲线

Fig. 3 Loss curve during training



105

图 4 训练过程中准确率的变化曲线

Fig. 4 Accuracy rate curve during training

## 2.4 算法评估

算法在测试集上的识别结果混淆矩阵如图 5 所示, 基于该混淆矩阵计算的召回率和准确率如表 2 所示。依据公式(1)和(2), 计算得到平均召回率为 0.902, 平均准确率为 0.897。



图 5 测试结果混淆矩阵

Fig. 5 confusion matrix of test result

表 2 不同害虫图像识别的召回率和准确率

Tab. 2 Recall rate and accuracy of image recognition different insects

	象甲类	谷蠹	扁谷盗类	拟谷盗类	烟草甲	锯谷盗
召回率	0.95	0.91	0.92	0.84	0.92	0.79
准确率	0.94	0.86	0.91	0.55	0.93	0.91

其中,  $accuracy_i$  为每类害虫图像的识别准确率,  $recall_i$  为每类害虫图像的识别召回率,  $amount_i$  为每类害虫图像的数量。

### 3 软件设计与实现

本文采用 Android 系统进行储粮害虫图像识别 APP 软件的开发。软件的害虫图像识别功能分为个 4 步骤: 选取害虫图像、裁剪图像、图像识别、结果展示。

#### 3.1 选取和裁剪害虫图像

储粮害虫图像识别的第一步是拍摄或选取相册中的害虫图片, APP 采用第三方插件 PictureSelector 作为图片选择器。PictureSelector 适配 Android 6.0 版本及以上的系统, 支持拍摄照片和选择相册图片, 支持图片预览和图片多种比例裁剪, 支持多种主题设置。APP 使用的 PictureSelector 参数配置如表 3 所示。

表 3PictureSelector 参数配置

Tab. 3PictureSelector parameter configuration

参数	设置值	效果
selectionMode()	PictureConfig.SINGLE	单选
previewImage()	true	可预览图片
isCamera()	true	显示拍照按钮
compress()	false	不压缩图片
enableCrop()	true	激活图片裁剪功能
hideBottomControls()	false	显示 uCrop 工具栏
showCropFrame()	true	显示裁剪矩形边框
showCropGrid()	true	显示裁剪矩形网格
previewEggs()	true	预览图片时增强左右滑动图片体验
rotateEnabled()	true	裁剪时可旋转图片
scaleEnabled()	true	裁剪时可放大缩小图片
freeStyleCropEnabled()	true	裁剪框可拖拽

130

当用户点击拍摄选取图片按钮后, PictureSelector 启动一个 Activity, 用户可以选择拍摄照片和在相册中选取图片。由于储粮害虫图像识别功能采用的识别算法为分类算法, 不涉及害虫目标检测问题, 为了达到更好的害虫识别效果, 提高识别成功率, 在用户选取害虫图片后, 要使用 PictureSelector 插件内的 uCrop 图片裁剪工具对选取的图片进行裁剪, 确保裁剪后的图片中仅包含单头害虫。

135

### 3.2 算法调用

对害虫图像进行选取和裁剪操作后, 就需要在 Android 手机本地调用图像识别算法对图像进行识别。本文采用 TensorFlow Lite 在 Android 手机上调用训练好的算法模型识别害虫图片。这需要在 Android 工程中添加两个依赖文件, 分别是 libandroid\_tensorflow\_inference\_java.jar 和 libtensorflow\_inference.so, 然后将导出的 pb 文件存放于 Android 项目中“app/src/main/assets”文件夹下, 随软件安装到手机本地。

140

调用算法时, 首先创建一个 TensorFlowInferenceInterface 对象 inferenceInterface, 这将加载算法的 pb 文件, 然后将要识别的图片转换为 float 类型的数组, 通过 inferenceInterface.feed() 方法将其传入算法的输入节点“shuffle\_batch”。接着调用 inferenceInterface.run() 方法开始对图像进行识别, 识别完成后调用 inferenceInterface.fetch() 方法从输出节点“Softmax”获取到 6 个 float 类型的置信度。这些置信度表示图像中的害虫属于 6 类害虫的可能性, 根据置信度的具体大小倒序排序, 然后在界面上显示置信度及其对应的害虫图片。

145

### 3.3 软件界面

软件的图像识别流程页面如所示。图(a)为图像识别功能首页, 点击上方的相机按钮进入图(b)所示页面。在图(b)所示页面中可以调用相机拍照或者选择相册中的图像。图(c)展示的是对图像进行裁剪。图(d)中上面展示的为原始图像, 下面为识别结果列表, 图中所示结果为象甲类的害虫玉米象和米象, 置信度为 100%。

150

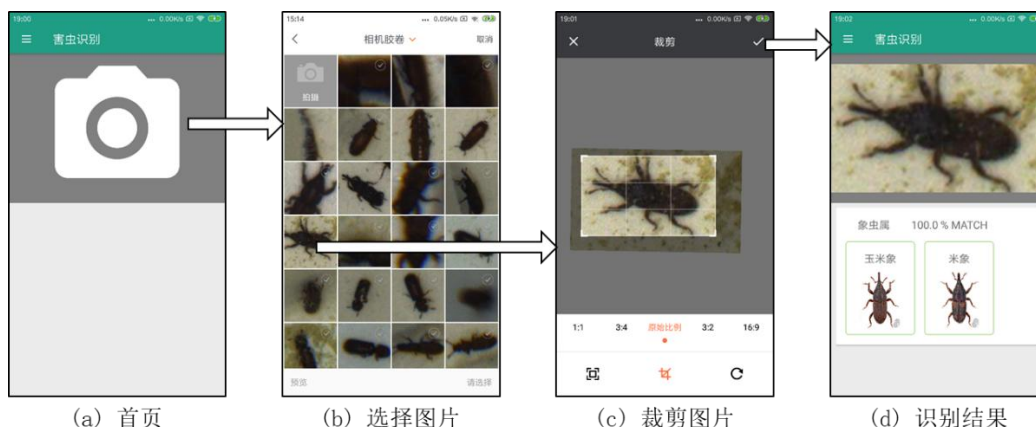


图 6 害虫识别流程界面

Fig. 6Interface of insect image recognition process

155

## 4 结论

本文基于储粮害虫图像数据集，采用 Resnet 模型训练了一种网络参数较少的储粮害虫图像识别算法，能够对 6 类 10 种储粮害虫的图像进行识别，平均准确率达到 0.90。本文设计实现了一款储粮害虫图像识别手机软件，采用 TensorFlow Lite 框架将算法移植到 Android 手机上，实现了在手机本地的储粮害虫图像种类识别。该软件可以为粮库保管员和广大储粮农户鉴别害虫的种类提供参考，成为他们的有益帮手。

160

### [参考文献] (References)

[1] 胡丽华, 郭敏, 张景虎, 等. 储粮害虫检测新技术及应用现状[J]. 农业工程学报, 2007, 23(11):286-290.

[2] 王海修. 储粮害虫检测与识别技术研究进展[J]. 粮油仓储科技通讯, 2011, 06(1):20-23.

[3] 马彬, 金志明, 蒋旭初, 等. 储粮害虫在线监测技术的研究进展[J]. 粮食储藏, 2018, (2):27-31.

[4] 邱道尹, 张红涛, 陈铁军, 等. 基于机器视觉的储粮害虫智能检测系统软件设计[J]. 农业机械学报, 2003, (2):83-85. DOI:10.3969/j.issn.1000-1298.2003.02.025.

[5] Defa Wang. Research on Image Acquisition and Recognition for Stored Grain Pests[A]. Science and Engineering Research Center. Proceedings of 2016 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE2016)[C]. Science and Engineering Research Center, 2016:5.

[6] Yufeng Shen, Huiling Zhou, Jiangtao Li, Fujijian, Digvir S. Jayas. Detection of stored-grain insects using deep learning[J]. Computers and Electronics in Agriculture, 2018, 145.

[7] 刘汉生. 陷阱式储粮害虫信息采集终端及其系统的研究与实现[D]. 北京邮电大学, 2018.

[8] Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with Deep Convolutional Neural Networks[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012.

[9] Karen Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint, arXiv:1409-1556.

[10] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. 2015.

165

170

175

180