



Research on Rice Nitrogen Nutrition Diagnosis System Based on Deep Learning Model and Android

Qiang Yao, Bin Lyu*, Chao Su, Bo Li

Institute of Agricultural Science and Technology Information, Chongqing Academy of Agricultural Sciences, Chongqing, China

Email address:

94388062@qq.com (Qiang Yao), 479996341@qq.com (Bin Lyu)

*Corresponding author

To cite this article:

Qiang Yao, Bin Lyu, Chao Su, Bo Li. Research on Rice Nitrogen Nutrition Diagnosis System Based on Deep Learning Model and Android. *Science Discovery*. Vol. 10, No. 5, 2022, pp. 340-346. doi: 10.11648/j.sd.20221005.20

Received: September 16, 2022; **Accepted:** October 20, 2022; **Published:** October 24, 2022

Abstract: Nitrogen nutrition diagnosis is one of the key technologies to achieve high quality and high yield in rice. It is time-consuming and laborious to use traditional diagnosis methods of rice nitrogen nutrition. Rapid and intelligent diagnosis of rice nitrogen nutrition can be realized by using smart phones and image recognition technology. In order to use mobile equipment to carry out nitrogen nutrition diagnosis in rice anytime and anywhere, and providing suggestions and prescriptions for fertilization management, by migrating the deep learning model to the Android environment, the rice nitrogen nutrition diagnosis system based on Android has been developed according to the established deep learning model of rice nitrogen nutrition recognition based on TensorFlow. The diagnosis results of the developed system have been verified and analyzed using the collected image data. At first, comparative analysis was conducted on various nitrogen nutrition diagnosis methods for rice, and then, image processing technology, image recognition technology, the configuration of development environment, system design and implementation, verification and analysis of diagnosis results have been emphatically introduced. The techniques and development methods used in the experiments are feasible and reproducible. The results of rice nitrogen nutrition recognition using the Android-based rice nitrogen nutrition diagnosis system were the same as the validation results of the original model.

Keywords: TensorFlow, TensorFlow Lite, Android, Deep Learning, Rice, Nitrogen Nutrition Diagnosis

基于深度学习模型和Android的水稻氮素营养诊断系统研究

姚强, 吕斌*, 粟超, 李波

重庆市农业科学院农业科技信息研究所, 重庆, 中国

邮箱

94388062@qq.com (姚强), 479996341@qq.com (吕斌)

摘要: 氮素营养诊断是实现水稻优质高产的关键技术之一。传统的水稻氮素营养诊断方法耗时费力, 利用智能手机和图像识别技术可以实现水稻氮素营养的快速智能诊断。试验在已取得基于TensorFlow的水稻氮素营养识别深度学习模型基础上, 为解决使用移动设备开展随时随地诊断识别水稻氮素营养状况并提供施肥管理建议的问题, 将深度学习模型迁移至Android环境, 开发了一套基于Android的水稻氮素营养诊断系统, 使用采集的图像数据对开发的系统进行了诊断结果验证和分析。本文首先对各类水稻氮素营养诊断方法进行了对比分析, 然后着重介绍了图像处理技术、图像识别技术、系统开发环境配置、系统设计与实现、诊断结果验证和分析。结果表明: 实验采用的技术和开发方法可行且可复制; 使用系统对水稻氮素营养诊断识别的结果与原模型验证结果相同。

关键词：TensorFlow, TensorFlow Lite, Android, 深度学习, 水稻, 氮素营养诊断

1. 引言

水稻是我国最主要的粮食作物之一。水稻生产在我国农业生产和粮食安全方面具有举足轻重的地位。氮素——水稻生长过程中需要最多的营养元素，对水稻的生长、产量和品质影响十分显著[1]。如果施氮经验不足，在水稻种植过程中，氮肥施用量不合理，经常会导致水稻产量和质量下降[2, 3]。近年来，随着互联网+、大数据、人工智能等新技术的发展和应用，运用数字技术、人工智能技术开展作物氮素营养诊断的研究取得了很多成果。如：李岚涛等应用数字图像技术进行水稻氮素营养诊断方面的研究，得出数码相机进行水稻氮素营养诊断测试结果稳定的结论[4]；吴刚等采用一种基于卷积神经网络的方法搭建ResNet18卷积神经网络模型，对采集的多光谱玉米植株图像进行氮素含量、水分含量等进行分析识别，验证集中彩色照片的准确率达到84.7%，多光谱图像的准确率达到90.5% [5]；作者本人也基于深度学习的方法开展了水稻氮素营养诊断的初步探索，分别针对水稻的返青期、分蘖期、拔节孕穗期和灌浆期建立了深度学习模型，并对模型进行了验证。移动应用开发技术的成熟发展和TensorFlow对移动端的支持，使得将深度学习模型迁移到移动端开展诊断识别成为可能，国内在基于移动端开展图像识别方面已经开展了很多研究，也取得了很多成果。如：兰宁研究了基于Android平台的图像识别设计与实现方法，将使用TensorFlow训练出的CNN模型迁移到Android系统中，使用户可以在移动端对图像进行识别[6]；郑姣等设计了一款基于Android手机的水稻病害图像识别系统，通过分析水稻稻瘟病等四种病害的图像特征，采用图像识别方法实现水稻病害类型诊断[7]。本文实现将基于TensorFlow建立的水稻氮素营养诊断模型迁移到Android端，开发一个用于

水稻氮素营养诊断的Android版手机APP，帮助用户随时随地开展水稻氮素营养诊断。

2. 功能分析

2.1. 水稻氮素营养诊断方法分析对比

随着科技发展，水稻氮素营养诊断方法主要有外观诊断、化学诊断与现代氮素营养诊断[8]。现代氮素营养诊断方法中，目前比较流行的有：叶绿素仪诊断、机器视觉诊断、高光谱诊断、深度学习诊断等。各种诊断方法所采用的主要技术以及方法的优缺点对比如表1所示。

2.2. 系统功能目标

通过田间实验、图像数据采集、划分训练集和测试集、模型训练和测试，得到了基于TensorFlow的水稻氮素营养识别模型，具体实验设计、模型生成过程、验证结果在作者“深度学习在水稻氮素营养诊断中的应用初探”一文中做了详细阐述[9]。

基于Android的水稻氮素营养诊断系统主要功能目标是让用户能够通过手机、平板等安装有Android系统的移动设备获知水稻氮素营养状况及处理措施。系统基本功能需求如下：

- (1) 调取移动设备摄像头拍摄水稻图像的功能。
- (2) 从移动设备内置或连接的存储器上选取水稻图像的功能。
- (3) 使用深度学习模型进行水稻氮素含量判断的功能。
- (4) 根据水稻氮素含量判断结果推荐处理建议的功能。

表1 水稻氮素营养诊断技术对比。

诊断方法	主要技术	优点	缺点
外观诊断	通过观察水稻植株茎和叶的颜色、形态及生长速度等外观参数作出营养状态诊断。	简单方便，快捷有效，能够直观诊断当前状况。	依赖诊断者知识和经验，难以量化和精准化。
化学诊断	采用靛酚蓝比色法、凯氏定氮法、硝态氮测定法等化学方法测定水稻叶片或整株中氮或硝态氮含量，诊断营养状态。	精确性高，可量化。	成本高，耗时长，取样会破坏植株。
现代诊断方法	叶绿素仪 使用叶绿素仪测定叶片叶绿素含量，根据叶片含氮量与叶绿素含量的正相关关系计算含氮量，开展氮素营养状态诊断[10]。	可量化氮素含量，取样对植株无损。	测定值受光照影响大，有变异度，需多点多次测定。
	机器视觉 通过计算机模拟人视觉开展氮营养诊断[11]。利用图像处理技术对叶片、植株、冠层图像预处理，提取颜色、纹理、形态等特征，采用特定算法建立氮营养诊断模型。	对植株无损，实时、快速、准确、有效，对于缺氮或过氮症状不明显时以及早做出诊断[12]。	建模难度高、成本高、多停留在实验研究阶段，操作比较繁杂，不易推广。
	高光谱 获取作物高光谱数据，根据不同氮素营养状况下作物光谱特性，分析叶片含氮量与特定光谱关系，开展氮素营养诊断。	具有快速、便捷、无损害，可开展实时监测[13]。	设备昂贵，光谱反射率受温、湿度、光照等因素影响大，数据处理对专业要求高。
	深度学习 模仿人类视听和人脑思维机制的方法，从图像中自动提取特征，找出特征表示形式，建立卷积神经网络模型，使用模型开展氮素营养诊断。	容错率高，自学能力强，运行速度快，自适应性好，泛化能力优异，自动提取特征。	需大量样本图像数据，运算量大，对硬件要求高。

3. 系统关键技术

基于Android的水稻氮素营养诊断系统主要使用了图像处理技术和图像识别技术。

3.1. 图像处理技术

在将用户拍摄的水稻图像传递给模型开展识别诊断前，需按模型要求对图像进行处理，以形成张量。应用于Android平台的图像处理方法主要有两种：一种是使用Java程序语言编写图像处理算法，另一种是调用第三程序库。两种方法各有各优点。编写图像处理算法的方法可以根据系统需要编写程序，减少代码量，提供程序性能。调用第三程序库的方法只需调用接口，降低开发难度，提升开发速度，降低成本。系统采用了调用第三程序库的方法进行图像处理。

Open CV(Open Source Computer Vision Library)是一个开源的图像处理库，有超过2500个图像处理算法，可以应用于多种图像处理场景，如：目标检测、对象识别、三维模型提取等，提供了C、C++、Java、Python等多种程序语言接口，支持Windows、Android、Linux、Mac OS等操作系统[14]。

3.2. 图像识别技术

图像识别是利用图像采集设备获取图像，将图像传输到计算设备中，经过图像处理、分割、特征提取、特征匹配等操作，识别不同目标对象的技术，主要过程有图像采集、图像处理、特征提取、特征匹配。随着科技发展，在我国农业向智能化、精准化发展过程中，图像识别技术在杂草识别、病虫害预警、作物采摘等越来越多的方面得到应用[15]。

常用的图像识别技术有基于机器学习的，有基于深度学习的。

基于机器学习的图像识别技术自动将特征向量分类或者聚类，达到图像识别的目的，在一定程度上减少了人工参与，提高了图像识别效率和准确率。这种方法在特征提取阶段需要人工参与，且对技术要求较高，增加了获取模型的难度。

基于深度学习的图像识别技术通过建立模拟人脑进行分析学习的神经网络，模仿人脑机制解释数据，模拟人类视觉系统识别图像中的目标事物。深度学习采用无监督式学习，学习和特征提取过程均无需人工参与，其信息处理过程是从具体信息到高维抽象特征的层层抽象。

4. 系统设计与实现

4.1. 系统设计

4.1.1. 系统架构设计

基于Android的水稻氮素营养诊断系统采用基于MVC的三层架构，所有程序打包在Android移动端运行。View层完成用户与系统交互，响应用户操作（包括调取摄像头拍摄图像数据、打开文件系统选择图片、发起诊断等）事

件，以及展示诊断结果和处理建议等。Model完成数据处理，主要包括图像的通道变换、尺寸缩放、像素值归一化，图像数据转换成Tensor，以及根据模型输出查询数据库获取处理建议等。Controller完成用户操作和业务之间的映射和控制，根据用户操作调用对应Model，将Model处理结果反馈到View展示给用户。系统架构设计如图1所示。

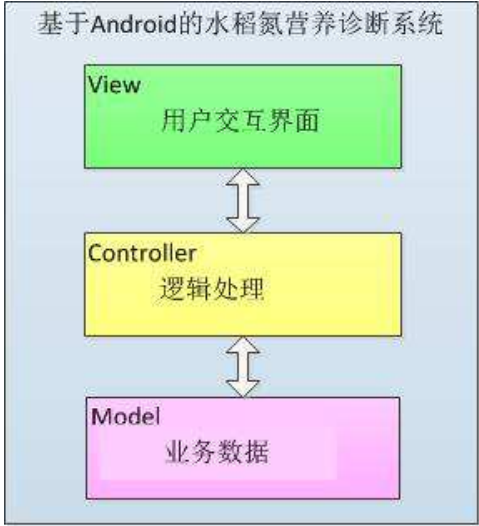


图1 系统架构。

4.1.2. 数据库设计

基于Android的水稻氮素营养诊断系统涉及3类数据：分类含量数据，深度学习模型数据，以及处理建议数据。

基于作者“深度学习在水稻氮素营养诊断中的应用初探”文中所述实验方法，将水稻生长周期划分为返青期、分蘖期、拔节孕穗期和灌浆期共4个时期阶段，每个时期根据施肥量不同划分为对照、低氮、低中氮、中氮、中高氮、高氮、超高氮7个等次，每个时期阶段都在不同等次稻田采集样本，测量氮素营养含量值作为参考。这些时期阶段、等次和参考值构成了分类含量数据。

深度学习模型数据包括模型的输入参数、输出参数、文件名和存放位置。输入参数主要是模型能接受的张量尺寸，输出参数主要是模型的分类数。系统使用的水稻氮素营养识别过程是首先识别出水稻处在那一生长时期阶段，然后识别出水稻对应的参考等次。系统用到的深度学习模型包括1个识别生长时期阶段的模型和4个识别等次的模型。随着后续实验的跟进和模型修正，会产生更加优异的深度学习模型，为了方便模型的更新替代，系统将模型对应的输入参数、输出参数、文件名称和存放路径记录在数据库中。

处理建议数据的主要构成是针对不同氮素营养含量范围的参考施肥建议和管理建议。

4.1.3. 功能模块及流程设计

分析系统功能目标可知，基于Android水稻氮素营养诊断系统主要包括图像采集、水稻氮素营养诊断、处理建议3个功能模块，其中水稻氮素营养诊断模块又包含图像处理 and 识别诊断两个子过程。

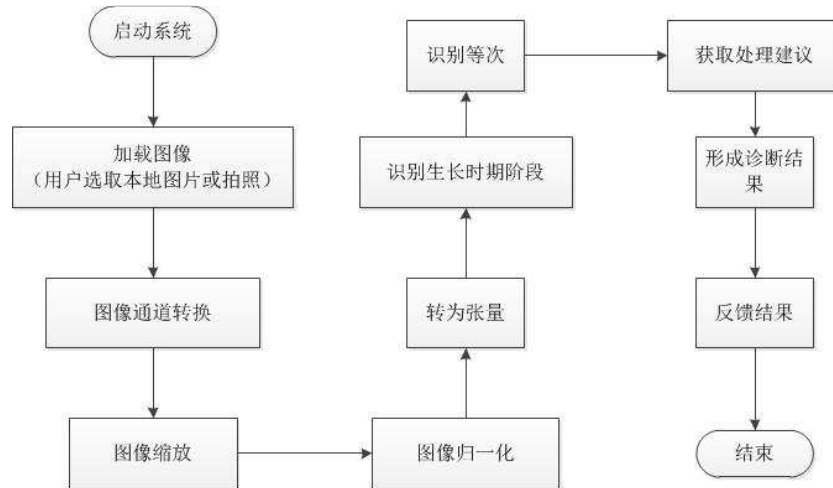


图2 系统工作流程。

系统主要工作流程为：系统启动时加载所有深度学习模型，用户在系统主界面通过点击按钮打开摄像头，拍摄水稻图像后返回系统主界面，点击诊断按钮触发氮素营养诊断过程，该过程首先对图像进行通道顺序变换，然后对图像进行尺寸缩放，再对像素值进行归一化处理，将图像数据转换为张量，调用深度学习模型识别水稻生长时期阶段和氮素营养参考值，根据识别结果查询处理建议，最后将诊断结果和处理建议反馈给用户。系统工作流程如图2所示。

4.2. 开发环境

基于Android的水稻氮素营养诊断系统使用Android Studio+SDK+OpenCV+Tensorflow Lite e开发实现，开发环境配置如下。

操作系统：Windows 10 64位版
 Java开发工具：JDK1.8
 IDE：Android Studio4.1.3
 Android SDK版本：
 minSdkVersion 26
 targetSdkVersion 30
 compileSdkVersion 30
 OpenCV版本：4.5.1
 TensorFlow-Lite版本：2.3.0

因系统使用的深度学习模型是在Python环境下训练生成，为使模型迁移至Android平台后有相同的识别效果，Android中使用的OpenCV和TensorFlow-Lite版本与Python中保持一致。引入OpenCV和TensorFlow-Lite最简便的方法是在build.gradle文件的dependencies 段添加下面的配置选项。

```
implementation 'org.tensorflow:tensorflow-lite:2.3.0'
implementation 'com.quickbirdstudios:opencv:4.5.1'
```

4.3. 系统实现

4.3.1. 用户界面实现

为达到用户操作简便、界面简洁的目标，系统所有操作均在一个用户界面完成，诊断结果及建议以弹出信息的

形式展示给用户。用户界面用到Button组件、ImageButton组件、Image View组件和TextView组件各一个。通过ImageButton的Click事件调取摄像头拍摄图像数据，完成后图像显示在ImageView中。通过Image View的Click事件打开本地存储空间，供用户选择存储在本地的图像数据，选择的图像显示在ImageView中，文件名显示在TextView中。通过Button组件的Click事件调用图像处理和识别诊断程序。整个界面采用ConstraintLayout布局，组件的属性和事件在xml文件中设置。用户界面采用ConstraintLayout布局，实现的用户界面如图3所示。



图3 系统用户界面。

4.3.2. 功能模块实现

(1) 图像采集模块实现

系统可通过两种方式采集图像数据：一种是调用移动设备自带摄像头拍摄图像数据，另一种是先使用其他摄像设备获取图像然后传送到移动设备上，然后在系统中浏览并选择图像数据。

第一种采集方式实现过程为：首先，创建一个Intent，将Intent的Action属性设置为ACTION_IMAGE_CAPTURE，然后通过Activity的startActivityForResult()方法将Intent发送给Android系统，并指定requestCode以便在Activity中对采集到的图像数据进行处理。主要实现代码如下：

```
Intent cameraIntent=new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
startActivityForResult(cameraIntent,1003);
onActivityResult(int requestCode, int resultCode, Intent data){
.....
if(requestCode==1003&&resultCode==RESULT_OK){
selectedimg=(Bitmap)data.getExtras().get("data");
img_photo.setImageBitmap(selectedimg);
}
```

第二种采集方式实现过程与第一种类似，创建一个Intent，将Intent的Action属性设置为ACTION_OPEN_DOCUMENT，Type属性设置为"image/*"，通过Type属性过滤显示的文件类型为图片。浏览和选择图像需要访问移动设备的文件系统，会用到读取存储设备的权限。系统使用到的权限均需AndroidManifest.xml中提前申明，申明如下：

```
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"></uses-permission>
```

系统采用动态申请的方式获取权限，用户首次访问文件系统时提出权限申请。

(2) 图像处理功能实现

图像处理功能借助OpenCV库实现。在Android程序中加载OpenCV库，使用OpenCV库中的Utils和Mat类将Bitmap格式的图像数据转化为矩阵数据。Android中Bitmap格式的图像数据通道顺序为RGBA，而基于TensorFlow的水稻氮素营养诊断模型需要图像通道格式为BGR。使用OpenCV库中Imgproc类的cvtColor()方法完成通道变换。使用Imgproc类的resize()方法将图像大小缩放至模型需要的尺寸。在归一化处理前需将矩阵中的像素数据从整型转化为浮点型，使用Mat类的convertTo()方法完成数据类型转换。

使用OpenCV进行图像数据归一化处理的过程是：先定义一个尺寸大小与模型需求尺寸一致的矩阵，将矩阵中所有元素值设为255.0，然后使用OpenCV库中Core类的divide()方法对缩放后的矩阵数据和新定义的元素值为255.0的矩阵数据做除法运算。

深度学习模型的输入是张量(tensor)数据，因此调用模型前要将矩阵数据转换为张量数据。TensorFlow Lite库中TensorBufferFloat类的loadArray ()方法可以将浮点数

组转换为张量。首先使用Mat类的reshape()方法将归一化后的矩阵转换为单行矩阵，再使用MatOfFloat类定义一个浮点数矩阵，使用Mat类的copyTo()方法将单行矩阵数据复制到浮点数矩阵，最后使用TensorBufferFloat类的loadArray ()方法生成张量数据。

图像处理的实现代码如下：

```
Mat img= new Mat();
Utils.bitmapToMat(selectedimg,img);
Mat imgrgb=new Mat(img.rows(),img.cols(), img.type());
Imgproc.cvtColor(img,imgrgb,Imgproc.COLOR_RGBA2BGR);
Mat res_img=new Mat();
Imgproc.resize(imgrgb,res_img,new Size(rows,height));
res_img.convertTo(res_img,CvType.CV_32FC(3));
Mat tempMat=new Mat(rows,height,CvType.CV_32FC(3),new
Scalar(255.0,255.0,255.0));
Mat dvided_image=new Mat(rows,height,CvType.CV_32FC(3));
Core.divide(res_img,tempMat,dvided_image);
dvided_image = dvided_image.reshape(1,1);
MatOfFloat input_img=new MatOfFloat();
dvided_image.copyTo(input_img);
float[] input_image=input_img.toArray();
int[] shape= {1,rows,height,3};
TensorBufferFloat tensor=(TensorBufferFloat) TensorBuffer.
createFixedSize(shape,DataType.FLOAT32);
tensor.loadArray(input_image);
```

(3) 识别诊断功能实现

识别诊断过程的实现是先用TensorFlow Lite库中Interpreter类创建一个解释器，并加载模型文件，然后调用run()方法开始识别诊断。run()方法将张量数据传入模型，并将识别结果存入输出。识别诊断的实现代码如下：

```
tfLite=new Interpreter(filepath,filename), Option s());
tfLite.run(tensor,result);
```

(4) 处理建议功能实现

识别诊断出水稻生长时期阶段和等次后，使用SQLiteOpenHelper类的getReadableDatabase()方法打开数据库，使用的SQLiteDatabase的query()方法，查询出该阶段和等次的处理建议，然后以弹出提示的方式反馈给用户。

5. 诊断结果验证与对比

从采集的图像数据中，每个水稻生长时期阶段和等次随机选取10张图像，分别在python环境下使用基于TensorFlow的模型和在Android环境下使用转换后的TensorFlow Lite模型开展诊断识别，记录诊断正确数量和错误数量，计算正确率，验证和对比诊断效果差异。验证结果和对比结果如表2所示。

表2 验证结果和对比

时期 阶段	程序 环境	验证总 数(张)	正确数量							错误数量							正确率(%)						
			1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7
返青	Python	70	10	9	3	7	10	6	8	0	1	7	3	0	4	2	100	90	30	70	100	60	80
	Android	70	10	9	3	7	10	6	8	0	1	7	3	0	4	2	100	90	30	70	100	60	80
分蘖	Python	70	7	3	4	2	6	9	7	3	7	6	8	4	1	3	70	30	40	20	60	90	70
	Android	70	7	3	4	2	6	9	7	3	7	6	8	4	1	3	70	30	40	20	60	90	70
拔节	Python	70	10	5	5	4	9	10	7	0	5	5	6	1	0	3	100	50	50	40	90	100	70
孕穗	Android	70	10	5	5	4	9	10	7	0	5	5	6	1	0	3	100	50	50	40	90	100	70
灌浆	Python	70	3	2	4	3	9	5	4	7	8	6	7	1	5	6	30	20	40	30	90	50	40
	Android	70	3	2	4	3	9	5	4	7	8	6	7	1	5	6	30	20	40	30	90	50	40

部分验证结果截图如图4、图5所示。图4是Python下验证目前准确率最高的返青期模型时的部分截图。图5是在基于Android的水稻氮素营养诊断系统中验证返青期图像的部分结果。

```
验证返青期等次 6 *****
\DJI_0308.JPG 有 50.19% 的可能性是属于 返青中高氮 类别
\DJI_0336.JPG 有 89.83% 的可能性是属于 返青中高氮 类别
\DJI_0370.JPG 有 85.58% 的可能性是属于 返青中高氮 类别
\DJI_0393.JPG 有 89.86% 的可能性是属于 返青中高氮 类别
\DJI_0408.JPG 有 72.11% 的可能性是属于 返青高氮 类别
\DJI_0447.JPG 有 97.19% 的可能性是属于 返青高氮 类别
\DJI_0494.JPG 有 97.78% 的可能性是属于 返青高氮 类别
\DJI_0500.JPG 有 97.06% 的可能性是属于 返青高氮 类别
\DJI_0517.JPG 有 99.19% 的可能性是属于 返青高氮 类别
\DJI_0521.JPG 有 99.47% 的可能性是属于 返青高氮 类别

验证返青期等次 7 *****
\DJI_0032.JPG 有 90.97% 的可能性是属于 返青超高氮 类别
\DJI_0061.JPG 有 99.03% 的可能性是属于 返青超高氮 类别
\DJI_0108.JPG 有 86.86% 的可能性是属于 返青超高氮 类别
\DJI_0124.JPG 有 85.85% 的可能性是属于 返青高氮 类别
\DJI_0130.JPG 有 85.88% 的可能性是属于 返青高氮 类别
\DJI_0170.JPG 有 97.49% 的可能性是属于 返青超高氮 类别
\DJI_0195.JPG 有 100.00% 的可能性是属于 返青超高氮 类别
\DJI_0205.JPG 有 99.98% 的可能性是属于 返青超高氮 类别
\DJI_0226.JPG 有 99.99% 的可能性是属于 返青超高氮 类别
\DJI_0238.JPG 有 99.96% 的可能性是属于 返青超高氮 类别
```

图4 返青期TensorFlow模型验证结果。

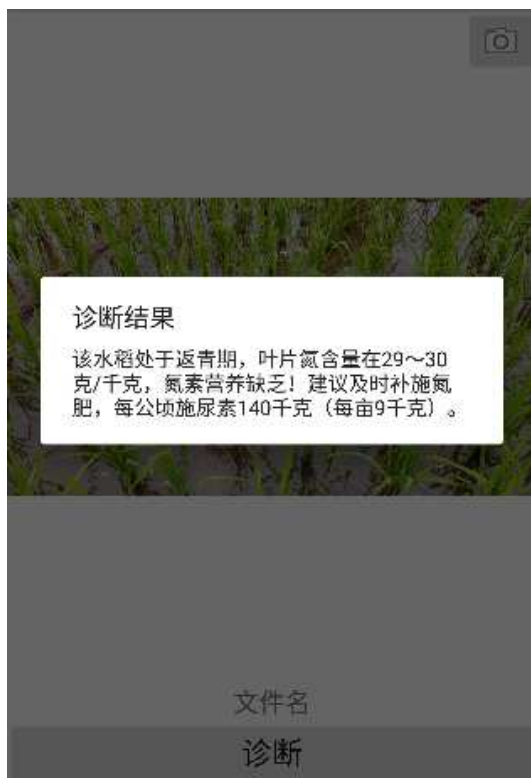


图5 模型迁移Android后验证结果。

6. 结论

实验在已经产生基于TensorFlow的水稻氮素营养诊断模型的基础上，开发基于Android的水稻氮素营养诊断系统，完成模型从Python环境到Android系统的迁移，借助了TensorFlow Lite、OpenCV等程序库，对比了模型转换前后以及在两种不同程序和运行环境下开展水稻氮素营养诊断的实验效果。对比验证结果得出结论：所开发的系统在诊断效果上与原模型完全相同，使用的技术和开发方法可行且可复制。

分析验证结果发现随着水稻生长时期阶段的变化，模型的准确率逐渐降低，主要原因与季节有关，进入夏季后光照强度高，拍摄照片容易曝光过度，导致原始图像数据质量差。另外，在Python环境下训练生成模型时所使用的网络模型和参数的配置也会对模型准确率产生影响。

后续工作重点在提高模型准确率上。一方面尝试使用不通的网络模型开展模型训练。另一方面，需要持续采集更新图像数据，提升图像质量。

致谢

本文为重庆市农业发展基金重大产业技术创新项目《手机+图像识别构建水稻氮素营养诊断系统研究》(NKY-2021AB009)的阶段性成果之一。

参考文献

- [1] 孙棋. 基于数字图像处理技术的水稻氮素营养诊断研究[D]. 杭州: 浙江大学, 2008: 1-3.
- [2] 杨红云, 罗建军, 万颖等. 计算机视觉技术在水稻氮素营养诊断中应用的研究进展 [J]. 中国农学通报, 2020, 36 (16): 149-155.
- [3] 曹彦圣, 付子轼, 孙会峰等. 施氮水平对水稻氮肥利用率和径流负荷的影响 [J]. 土壤, 2016, 48 (5): 868-872.
- [4] 李岚涛, 张萌, 任涛等. 应用数字图像技术进行水稻氮素营养诊断 [J]. 植物营养与肥料学报, 2015, 21 (1): 259-268.
- [5] 吴刚, 彭要奇, 周广奇等. 基于多光谱成像和卷积神经网络的玉米作物营养状况识别方法研究 [J]. 智慧农业 (中英文), 2020, 2 (01): 111-120.
- [6] 兰宁. 基于Android平台的图像识别设计方法与实现 [J]. 电子技术与软件工程, 2021, (19): 61-64.
- [7] 郑姣, 刘立波. 基于Android的水稻病害图像识别系统设计与应用 [J]. 计算机工程与科学, 2015, 37 (07): 1366-1371.
- [8] 杨红云, 周琼, 杨珺等. 水稻氮素营养诊断方法研究进展 [J]. 中国稻米, 2020, 26 (02): 5-8+13.
- [9] 姚强, 栗超, 李波等. 深度学习方法在水稻氮素营养诊断中的应用初探 [J]. 南方农业, 2021, 15 (31): 125-129.

- [10] CARTER G A, KNAPP A K. Leaf optical properties in higher plants: Linking spectral characteristics to stress and chlorophyll concentration [J]. *Am J Bot*, 2001, 88 (4): 677-684.
- [11] 罗建军. 基于计算机视觉技术与高光谱技术的水稻氮素营养诊断研究[D]. 南昌: 江西农业大学, 2015: 17-19.
- [12] 黄喜梅, 毕建杰, 张楠等. 计算机视觉技术在农业上的应用 [J]. *农业科学与技术*, 2017, 18 (11): 2158-2162.
- [13] 张浩, 胡昊, 陈义等. 水稻叶片氮素及籽粒蛋白质含量的高光谱估测模型 [J]. *核农学报*, 2012, 26 (1): 135-140.
- [14] 吴子龙. 基于Android移动终端的烟草病虫害图像智能识别系统研究[D]. 昆明: 云南农业大学, 2015: 17-19.
- [15] 刘鹏, 庄卫东. 图像识别技术在农业中的应用浅析 [J]. *现代化农业*, 2021, (12): 20-21.