百度云开发者大赛

植物移动检索推荐识别系统

目录

摘 要..... 3
一、 背景介绍..... 4
二、“植物识别”概要设计..... 5
2.1 预处理..... 5
2.1.1 阈值分割..... 5
2.1.2 去除根茎..... 7
2.1.3 分割结果..... 7
2.2 特征提取..... 8
2.2.1 Pyramid Histograms of Oriented Gradients (PHOG) 8
2.2.2 颜色特征..... 10
2.2.3 HSV..... 11
2.2.4 Wavelet feature..... 11
2.2.5 特征选择..... 13
三、“植物识别”的详细设计..... 13
3.1 叶片识别..... 13
3.2 语音合成..... 14
3.3 数据库设计..... 16
3.4 功能模块设计..... 17
3.4.1 主界面功能..... 17
3.4.2、选择图片功能..... 18
3.4.3.识别结果功能..... 18
3.4.5 图像细节..... 19
3.4.6.文字细节..... 20
3.4.7 拍照识别..... 20
3.4.8.联网识别..... 21
..... 21
3.5 软件系统介绍..... 21
四、市场前景展望..... 22

摘 要

植物种类自动识别对于业余植物学家、植物爱好者、教育工作者等都是非常 有必要和有必要的，在此，我将设计一款基于植物移动检索推荐识别系统，它通过植物叶片的图像识别出植物的种类，这个图像可以是已有的数字图像，也可以是通过摄像头获取的现实中的植物叶片图像，其可以本地识别也可以联网后台系统识 别。在识别过程中主要包含三个步骤：首先是叶片的分割，其次是特征的提取， 最后是识别的算法，在识别结束后，会显示给用户最接近被测图像的前几类植物的相关信息。在这里介绍的软件其训练数据库包含了竞赛 ImageCLEF2012 中的 126 类的植物叶片图像，系统输出显示的是最匹配测试图像的前几个植物类别， 同时还可以显示出各个类别中的详细信息，其中包括植物叶片、花卉、果实、树 皮等的高清图片，以及该类别的相应文本描述。同时通过语音合成技术，读取各种植物信息。同时经过测试，我们的软件在所 有的 android 设备上均运行良好。

关键字：数字图像处理　植物识别　语音合成

一、背景介绍

植物是地球上物种数量最多、分布最广泛的生命形式之一，通过光合作用维 持着大气中二氧化碳与氧气的平衡。同时，植物是人类的重要食物来源，也是人 类生产和生活所必需的资源。此外，植物在水土保持、抑制荒漠和改善气候等方 面也起着至关重要的作用。根据统计，地球上大约有 400,000 种植物，其中已被 植物学家命名和记录在案的约有 270,000 种。在我国仅高等植物就有 35,000 多 种，占世界总数的10.5%左右，是世界第二大植物物种资源库。近年来，日益增 加的人类生产活动已经严重地破坏了生态环境,据统计,世界上大约有3.4万种 植物物种已处于灭绝的边缘，占世界上已知的 27 万种植物的 12.6% ，对植物物 种的保护已刻不容缓。

我们知道，手机作为通信工具，在日常生活中占据重要的地位，手机性能也 在不断提升，如果在手机上开发出一款基于内容的植物识别软件，在软件中提供 相应的实用功能，将为用户提供极大的方便。

本软件基于不同种植物叶片的区别，从而区分植物的种类，在软件中可以通 过摄像头拍摄出植物的叶片或者直接选择已有的叶片数字图像,通过本地手机或 者后台服务器自动执行相关的操作从而识别出植物的种类，并展示出相应的信 息。在此所介绍的系统是基 于内容的图像检索技术（CBIR）的，其对于用户学习与识别植物是非常有用的。本软件中所用的特征提取算法并未出现在植物叶片识别中,同时本研究的另一个方向是多特征的融合,在制作软件过程中,本人同时提取了多种特征,进行融合并用 matlab 仿真，进行特征融合，取得了令人满意的效果。

在这里再简单介绍一下识别的主要步骤:（如图 1 所示 对于给定的查询图像， 系统会首先进行预处理,用阈值分割方法将其分割为二值图像,即黑白图像,再 利用 Tophat 算法去除叶片的根茎，然后再使用特征提取算法提取相关的特征， 并对所得的特征使用相关的识别算法进行识别,在系统的最后,返回给用户与输 入图像最匹配的前几幅图片。

植物识别目前仍面临着许多问题，比如，同种植物的叶片在不同的季节其叶片的颜色会相差很大，叶片的根茎也很难统一，所以植物识别在数字图像处理技 术中也是一个热门的领域，有一些专门的竞赛为植物识别而举办，其中一个叫做 “ImageCLEF”[4,5]，关于软件所使用的

数据库就自于这个竞赛，这个数据库中 包含了法国地中海地区的 126 类的植物叶片数字图像。对于这款软件，在所有测 试的 android 设备中都运行良好。

二、“植物识别”概要设计

2.1 预处理

在这个系统中，为了得到更加精确的结果，我们需要在提取特征前对图像进 行分割操作，分割的目的是将叶片与背景区域分开，其中由于现分割算法与手机 性能的限制，输入图像必须是背景“干净”的数字图像，即背景中必须是尽量单 一的，不能有其他纹理及物品。因为考虑到软件的性能及运行所需要的实时性， 所以我采用了简单的阈值分割[6]，并对叶片进行了去根茎处理，如图 2 下面是 简单介绍：

2.1.1 阈值分割

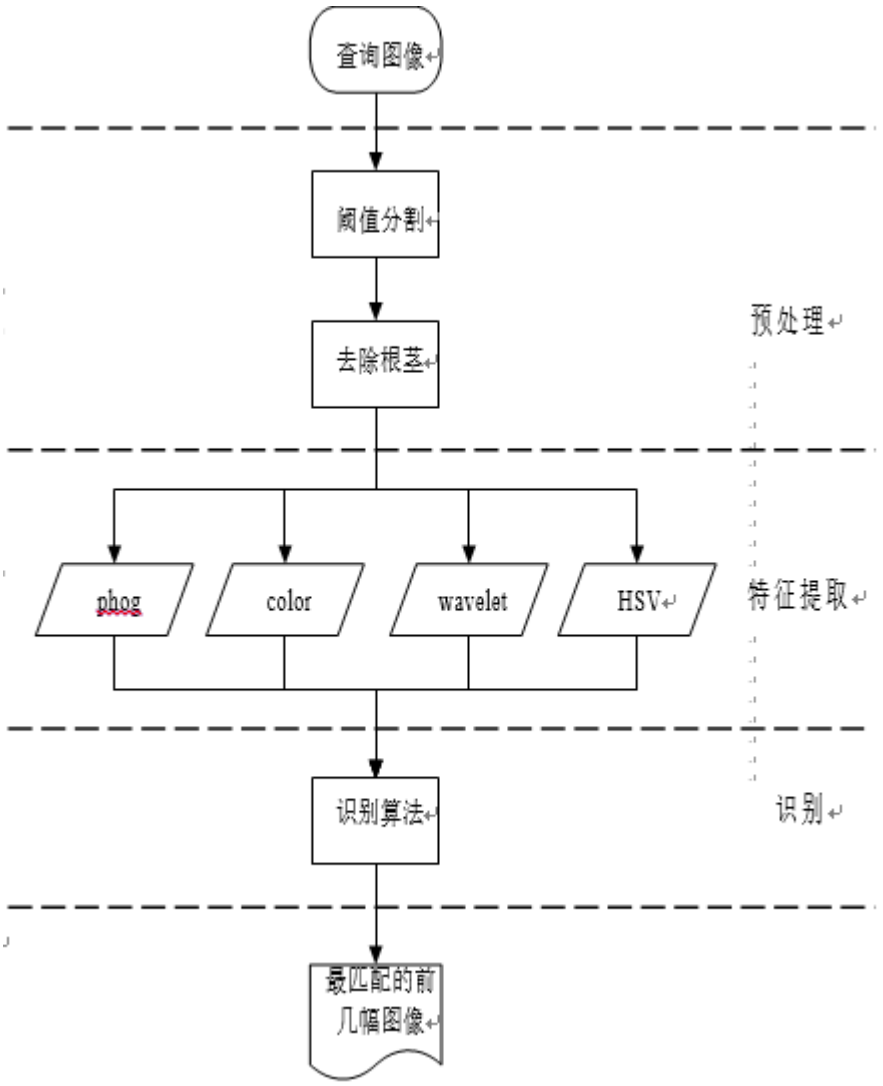
首先对图像利用如下公式将彩色图像变换成灰度图像：

Gray = R 0.299 G 0.587 B 0.114

其中 R、G、B 是彩色图像的三个通道，然后系统采用基于像素点的阈值分割方法（见图 2），分割出植物的叶片区域，阈值分割是图像分割中的一项重要的 方法，其实现非常的简单有效，它是基于一个阈值对图像实现分割。通过实验观 察，当阈值设置为图像像素的均值时，效果最好，所以定义阈值分割如下：

$$b(x, y) = \begin{cases} 1, & \text{if } g(x, y) \geq M \\ 0, & \text{if } g(x, y) < M \end{cases}$$

其中，x，y 是图像的像素坐标 ， g x, y 是灰度图像的像素点，b x, y 为二值 图像的像素点，M 为阈值。


查询图像
阈值分割
去除根茎
预处理
phog
color
wavelet
HSV
特征提取
识别算法
识别
最匹配的前
几幅图像

2.1.2 去除根茎

通过观察发现，对于植物叶片图像其根茎的排放没有一个统一的标准，比如 根茎摆放的位置，根茎的截取长度等等都很难统一，根茎的存在虽然可以适当的 增加不同种叶片之间的类间差异，但同时更能增加同种叶片之间的类内差异，这 往往会造成识别结果的错误，所以在完成阈值分割之后，系统须对其再经行处理， 去除叶片的根茎。本系统中，采用了Tophat 变换[7]，（见图2）它可以利用一个 预先定义的结构体（本系统结构体采用由0、1 两个数组成的矩阵所表示出的圆），
其定义为：

$$T \quad B \qquad B$$
$$B \qquad S$$

其中， 表示利用结构体侵蚀图像的操作。

2.1.3 分割结果

在去除根茎之后，对图像经行裁剪，裁剪出叶片区域的图像，再经行特征提 取，通过实验发现其效果总体令人满意，图 2 为分割过程的一个样例。

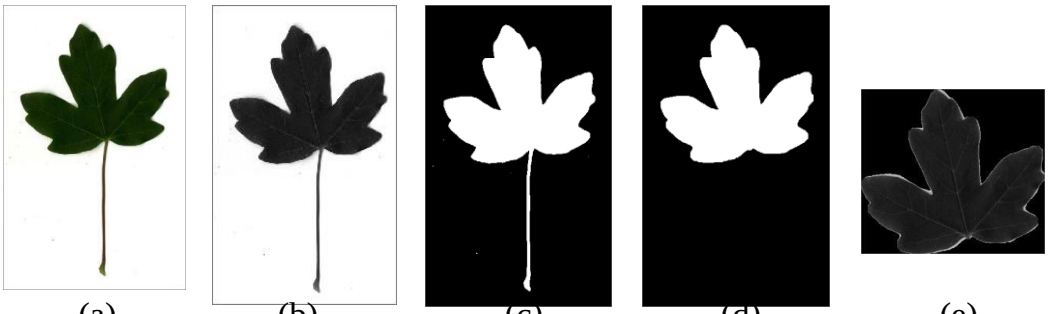


图 2. 系统首先将原始输入的 RGB 图像（a）转换为灰度图像（b），然后利 用阈值分割分割出叶片（c），并使用 tophat 去除其根茎（d），最后对其 经行裁剪操作（e）。

2.2 特征提取

为了正确经行识别，系统就需要一个好的识别算法，而为了满足软件对于实 时性的要求，就需要一个时间复杂度尽量低的算法，综合以上，为系统考虑了几 种“小”的特征，比如 PHOG，color，HSV 以及 wavelet 等，并将这些特征在 matlab 上进行仿真模拟实验，目的是选出优秀的特征并应用于今后的系统中，实验结果见表 1、表 2。其中叶片的颜色会随着季节的改变而改变，但是形状却基 本保持不变，通过实验发现，颜色并不能有效的提高识别的精度，所以其也不会 出现在以后的特征列表中，下面在简单介绍一下几个小特征：

2.2.1 Pyramid Histograms of Oriented Gradients (PHOG)

phog 特征[8]是基于 HOG 特征，并对以往的 HOG 特征[9]应用了空间金字塔 的思想[11]（如图 3）。为了提取图像的 PHOG 特征，首先需要对其进行边缘检测， 然后通过空间金字塔的思想对其划分网格，然后利用 HOG 思想对每个网格统计 梯度信息，并将梯度的方向信息划分到 N 个小块中，将每个网格中的信息统计 后并拼接起来，从而得到了所需要的特征，其中网格划分时，在 1 层中所包含的网格数为 21 。本系统中使用 Sobel 算子进行边缘检测：

$$sobel_x = \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ 2 & 0 & 2 \end{vmatrix}_y = \begin{vmatrix} 1 & 0 & 1 \\ 2 & 0 & 2 \\ 1 & 0 & 1 \end{vmatrix}, \quad sobel$$

则得到在 x，y 方向上的梯度：

$$I_x = I * sobel_y \qquad sobel_x , \ I_y$$

于是得到每个像素点上梯度的幅值：

$$|I| = \sqrt{I_x^2 + I_y^2}$$

在像素点上的梯度方向为：

$$\arctan(I_x / I_y)$$

在实验中，设置每个网格的梯度方向划分为 N=8 个小块，并将空间金字塔的

层数 L 设为 2 以及 3，分别对其做实验测试。

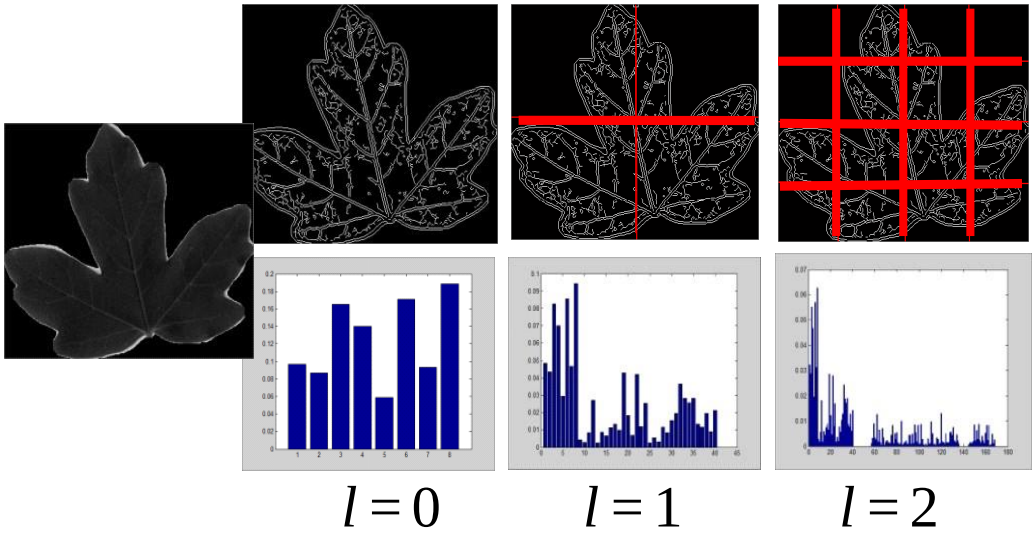


图 3. 对输入图像采用空间金字塔划分，上面的一行为层数设置为 0,1,2 时 的划分结果，下面一行为各层的直方图统计信息结果

2.2.2 颜色特征

颜色特征[15]是使用最广泛的特征之一，很多的方法能够被用来描述颜色的 特征,其中颜色矩方法有较低的维度以及较低的计算复杂度，较容易提取该特征， 所以其更适合于图像检索，Stricker 和 Orengo [17]使用了三种颜色矩作为颜色的
特征（均值 Ei 、标准差 i 以及偏度 si ），其定义如下：

$$E_i = \sum_{j=1}^{N} \frac{1}{N} p_{ij}$$

$$\sigma_i = \sqrt{(\frac{1}{N}\sum_{j=1}^{N}(p_{ij} - E_i)^2)}$$

$$s_i \sqrt{= \frac{1}{N}(\sum^{N}(p_{ij} - \ ^3}$$

提取特征时，首先将原始 RGB 图像转化为 HSV 图像，并采用划分网格的思
想，将图像划分为 3 3 的网格，对每个网格中的三个颜色通道（H、S、V）分 别计算颜色矩，从而得到了一个 81 维的向量（ 3 3 的网格中，对 3 个通道分 别计算 3 中颜色矩信息），这个 81 维的向量就作为最终的颜色特征。

2.2.3 HSV

HSV(Hue, Saturation, Value)颜色模型是一种非线性的颜色模型，是根据颜色 的直观特性由 A. R. Smith 在 1978 年创建的一种颜色空间， 也称六角锥体模型 (Hexcone Model)。它将 RGB 转化为色调 H、饱和度 S 以及亮度值 V 三种信息， 其中 H：用角度度量，取值范围为 0°～360°，从红色开始按逆时针方向计算，红色为 0°，绿色为 120°，蓝色为 240°。它们的补色是：黄色为 60°，青色为 180°，品红为 300°；S：取值范围为 0.0～1.0；V：取值范围为 0.0(黑色)～1.0(白 色)。然后利用基于直方图统计的思想，利用相应的规则将 H 划分为 32 个部分， S 和 V 分别划分为 8 个部分，相互拼接，从而得到一个 48 维的向量。

2.2.4 Wavelet feature

小波分析[14]是当前数学中一个迅速发展的新领域，它同时具有理论深刻和 应用十分广泛的双重意义。小波变换的概念是由法国从事石油信号处理的工程师 J.Morlet 在 1974 年首先提出的，通过物理的直观和信号处理的实际需要经验的 建立了反演公式，小波分析的应用是与小波分析的理论研究紧密地结合在一起 地。小波分析也是在图像处理领域使用非常广泛的一个特征，在这里使用简单的哈尔变换来计算，相关定义如下：

$$\begin{cases} \varphi(x,y)=\varphi(x)\varphi(y) \\ \psi^{H}(x,y)=\varphi(y)\psi(x) \\ \psi^{v}(x,y)=\varphi(x)\psi(y) \\ \psi^{D}(x,y)=\varphi(x)\psi(y) \end{cases}$$

其中， H x, y ， V x, y ， D x, y 分别表示图像沿着水平、垂直以及对角线方向的变化率， i ,m,n(x, y) 为尺 度函数，

j ,m,n (x, y) 为变换基函数,j 2 (2i xy n)
则: m, 2 j

i ,m,n (x,
y) 2

2 j i j j
j ,m,n (x, y) 2 x m, y ,i H ,V , D
2 2 n

则对于大小为 M N 的离散小波分析函数 f (x, y) :

w j0 , m, n f (x, y) j ,m,n
(x, y)
MN x 0 y 0 0

1 M 1
1 N 1

w i j, m, n i
f (x, y) j ,m,n (x, y)
MN x 0
y 0

w j0 , m, n 为图像近似度; j, m, n 为水平、垂直及对角线方向的
协同因素。这里同样是划分 3 3 的网格，并且在三个不同的层次上计算水平、 垂直及对角线方向上的协同因素，从而得到 81 维的小波图像特征。

2.2.5 特征选择

对于植物的叶片，其颜色会随着季节的改变而改变，但是其形状基本会保持不变，所以从这个意义上来看，基于形状的特征要优于基于颜色的特征。试验中， 利用最近邻方法经行叶片自动识别（见第 4 节），通过 Matlab 仿真实验结果也证 实了上述的结论（表 1 和表 2），所以颜色特征在试验中被测试，但是在最终不 会出现在系统中。

三、“植物识别”的详细设计

3.1 叶片识别

系统需要对一个输入图像自动识别其类别，所以系统也需要一个分类器来自动对提取的特征做出判断，对于实验中所提取的不同特征，分别比较了直方图交 叉和以及欧式距离的度量，结果显示直方图交叉和的效果优于欧式距离。关于欧式距离的定义如下：

$$d(a,b)=1-\sum_{i}^{N}\min(a_i,b_i)$$

通过直方图交叉和计算出测试图片与数据库中图片的距离大小，对其进行排序，并去除重复项后，显示给用户，让用户自己判断，在所列的结果中，排序越 靠前，其特征越接近测试图片的特征。

3.2 语音合成

植物移动检索推荐识别系统利用百度合成技术，读取植物的信息。可以自动识别植物的类别，对不同的植物做出不同的判断。用户点击主界面，选取一种植物简介。该系统就会帮助用户读取植物信息。这样提高了该系统的性能，给用户节约了时间，增强了用户体验。

下面是该植物移动检索推荐系统植物的百度语音合成技术的接口设计：

```
//百度合成

private void initialEnv() {
    if (mSampleDirPath == null) {
        String sdcardPath = Environment.getExternalStorageDirectory().toString();
        mSampleDirPath = sdcardPath + "/" + SAMPLE_DIR_NAME;
    }
    makeDir(mSampleDirPath);
    copyFromAssetsToSdcard(false, SPEECH_FEMALE_MODEL_NAME, mSampleDirPath + "/" + SPEECH_FEMALE_MODEL_N
    copyFromAssetsToSdcard(false, SPEECH_MALE_MODEL_NAME, mSampleDirPath + "/" + SPEECH_MALE_MODEL_NAME)
    copyFromAssetsToSdcard(false, TEXT_MODEL_NAME, mSampleDirPath + "/" + TEXT_MODEL_NAME);
    copyFromAssetsToSdcard(false, LICENSE_FILE_NAME, mSampleDirPath + "/" + LICENSE_FILE_NAME);
    copyFromAssetsToSdcard(false, "english/" + ENGLISH_SPEECH_FEMALE_MODEL_NAME, mSampleDirPath + "/"
            + ENGLISH_SPEECH_FEMALE_MODEL_NAME);
    copyFromAssetsToSdcard(false, "english/" + ENGLISH_SPEECH_MALE_MODEL_NAME, mSampleDirPath + "/"
            + ENGLISH_SPEECH_MALE_MODEL_NAME);
    copyFromAssetsToSdcard(false, "english/" + ENGLISH_TEXT_MODEL_NAME, mSampleDirPath + "/"
            + ENGLISH_TEXT_MODEL_NAME);
}

private void makeDir(String dirPath) {
    File file = new File(dirPath);
    if (!file.exists()) {
        file.mkdirs();
    }
}

/**
 * 将sample工程需要的资源文件拷贝到SD卡中使用（授权文件为临时授权文件，请注册正式授权）
 *
 * @param isCover 是否覆盖已存在的目标文件
 * @param source
 * @param dest
 */
private void copyFromAssetsToSdcard(boolean isCover, String source, String dest) {
    File file = new File(dest);
    if (isCover || (!isCover && !file.exists())) {
        InputStream is = null;
        FileOutputStream fos = null;
        try {
            is = getResources().getAssets().open(source);
            String path = dest;
            fos = new FileOutputStream(path);
            byte[] buffer = new byte[1024];
```

```
private void initialTts() {
    this.mSpeechSynthesizer = SpeechSynthesizer.getInstance();
    this.mSpeechSynthesizer.setContext(this);
    this.mSpeechSynthesizer.setSpeechSynthesizerListener(this);
    // 文本模型文件路径 (离线引擎使用)
    this.mSpeechSynthesizer.setParam(SpeechSynthesizer.PARAM_TTS_TEXT_MODEL_FILE, mSampleDirPath + "/"
            + TEXT_MODEL_NAME);
    // 声学模型文件路径 (离线引擎使用)
    this.mSpeechSynthesizer.setParam(SpeechSynthesizer.PARAM_TTS_SPEECH_MODEL_FILE, mSampleDirPath + "/"
            + SPEECH_FEMALE_MODEL_NAME);
    // 本地授权文件路径,如未设置将使用默认路径.设置临时授权文件路径, LICENCE_FILE_NAME请替换成临时授权文件的实际路径, 仅在使用临时
    // 如果合成结果出现临时授权文件将要到期的提示, 说明使用了临时授权文件, 请删除临时授权即可。
    this.mSpeechSynthesizer.setParam(SpeechSynthesizer.PARAM_TTS_LICENCE_FILE, mSampleDirPath + "/"
            + LICENSE_FILE_NAME);
    // 请替换为语音开发者平台上注册应用得到的App ID (离线授权)
    this.mSpeechSynthesizer.setAppId("8901019"/*这里只是为了让Demo运行使用的APPID,请替换成自己的id。*/);
    // 请替换为语音开发者平台注册应用得到的apikey和secretkey (在线授权)
    this.mSpeechSynthesizer.setApiKey("6vAfXMvOUAXNtxTR67Gk3546",
            "f0e5c982062aa505e2e310d4e467fa31"/*这里只是为了让Demo正常运行使用APIKey,请替换成自己的APIKey*/);
    // 发音人 (在线引擎) , 可用参数为0,1,2,3。。。 (服务器端会动态增加, 各值含义参考文档, 以文档说明为准。0--普通女声, 1--普通男声,
    this.mSpeechSynthesizer.setParam(SpeechSynthesizer.PARAM_SPEAKER, "0");
    // 设置Mix模式的合成策略
    this.mSpeechSynthesizer.setParam(SpeechSynthesizer.PARAM_MIX_MODE, SpeechSynthesizer.MIX_MODE_DEFAULT);
    // 授权检测接口(只是通过AuthInfo进行检验授权是否成功。)
    // AuthInfo接口用于测试开发者是否成功申请了在线或者离线授权, 如果测试授权成功了, 可以删除AuthInfo部分的代码 (该接口首次验证时比
    AuthInfo authInfo = this.mSpeechSynthesizer.auth(TtsMode.MIX);
```

```
private void speak() {
    String text = this.knowledgeStringB.toString();
    //需要合成的文本text的长度不能超过1024个GBK字节。
    if (TextUtils.isEmpty(knowledgeStringB.toString())) {
        text = "欢迎使用百度语音合成SDK,百度语音为你提供支持。";

    }
    int result = this.mSpeechSynthesizer.speak(text);
    if (result < 0) {
          toPrint("error,please look up error code in doc or URL:http://yuyin.baidu.com/docs/
        Toast.makeText(this, "语音阅读失败", Toast.LENGTH_SHORT).show();
    }
}

private void pause() {
    this.mSpeechSynthesizer.pause();
}

private void resume() {
    this.mSpeechSynthesizer.resume();
}

private void stop() {
    this.mSpeechSynthesizer.stop();
}

private void synthesize() {
    String text = this.knowledgeStringB.toString();
    //需要合成的文本text的长度不能超过1024个GBK字节。
    int result = this.mSpeechSynthesizer.synthesize(text);
}

private void batchSpeak() {
    List<SpeechSynthesizeBag> bags = new ArrayList<SpeechSynthesizeBag>();
    bags.add(getSpeechSynthesizeBag("123456", "0"));
    bags.add(getSpeechSynthesizeBag("你好", "1"));
    bags.add(getSpeechSynthesizeBag("使用百度语音合成SDK", "2"));
    bags.add(getSpeechSynthesizeBag("hello", "3"));
```

3.3 数据库设计

本系统所采用的数据库来自于一个关于植物叶片分类的竞赛“ImageCLEE2012”，这个数据库中包含了来自法国地中海地区的 126 类植物叶 片的不同图像[12]，数据库中的图像分为三种图片：背景“干净”的图片，带不 含特殊纹理背景的图片，还有就是自然场景的图片（如图 4）。我们的系统中由 于分割算法的限制，只使用了前两类的图片，一共包含 126 类的 4870 张图片。 由

于手机性能的局限性，我们在试验中利用 matlab 进行仿真，将数据库中图片 随机分成 2500 张做训练图像，其他的 2370 张做为测试图像，得到的结果见表 1、 表 2。



(a) (b) (c)

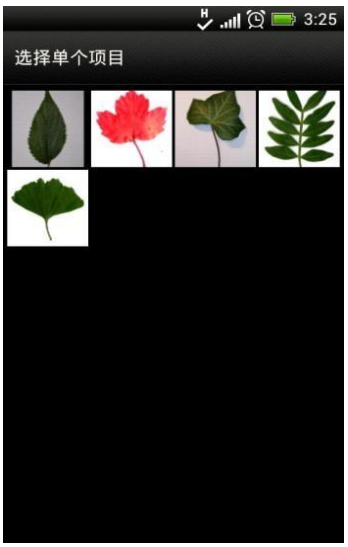
图 4. 数据库中的一个样例，（a）为背景“干净”的，（b）稍有背景的，
（c）自然场景图片。

3.4 功能模块设计

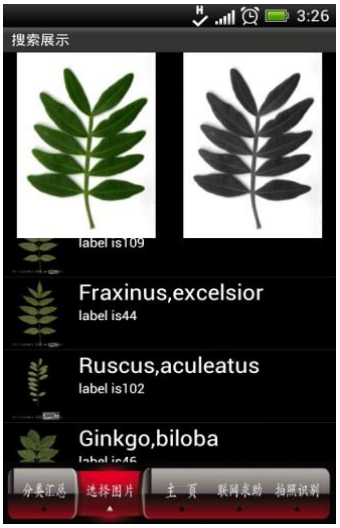
3.4.1 主界面功能



3.4.2、选择图片功能



3.4.3.识别结果功能



3.4.4.分类汇总



3.4.5 图像细节



3.4.6.文字细节



3.4.7 拍照识别



3.4.8.联网识别



3.5 软件系统介绍

在这里我们描述了第一个基于 android 平台的植物叶片识别软件，它的数据库 中包含了来自法国地中海地区的 126 类的植物的相关信息，包括简单的文本描 述、植物的叶片、花卉、果实等的高清图片等信息。当用户遇到一个植物时，他 就可以向本软件寻求帮助，软件将会显示出一组相似的植物信息列表，其中就包 含文本及图片信息，供用户自行查阅，做出判断。

当前版本的软件，其主要包含以下几个部分（图 5 a - i）： 主页：在主页中，设置了一些精美的植物的图片循环显示，供用户观赏（a）。 选择图片页：当用户点击页面下方的“选择图片”按钮时，软件就会自动调用手 机中的图片库系统（b），然后用户可以在图片库中找到自己想要识别的叶片的数 字图片，点击该图片之后，系统将会自动识别，并返回一个按照识别的相似度排 列的列表，其中排在第一位的是与输入图像最相似的类别，然后用户即可通过查看这些植物的文本及图片的详细信息（f、g），详细的植物的叶片、花卉、树皮等的高清图片信息，以及植物的文本描 述信息。



分类显示页：点击这个按钮，系统会展示出数据库中所有的 126 类的植物信息
（d），和“选择图片页”一样，点击其中的某个项目可以展示出其中的细节信息， 在该页中还可以调用搜索功能，通过植物的名称快速搜索出需要的信息（e）。联网识别页：如果想要获得更加精确的识别结果，则需要更加复杂的算法，但是 考虑到手机的局限性，所以系统还提供了联网识别的功能，即如果用户的手机可 以连接网络，则可以点击“联网识别按钮”（然后和“选择图片”页类似操作， 但实际流程完全不同），点击“联网识别”并选择图片后，首先系统会将该图片 上传至服务器端，上传完毕后，会自动打开浏览器页面，识别的结果在浏览器上 显示（服务器端是利用 JSP 所编写）。 拍照识别：如果用户遇到的是现实中的叶片，点击“拍照识别”按钮后，系统会 调用摄像头程序，当用户按下拍照键后几秒便会返回如前面一样的识别结果列
表。

比较苹果操作系统中的相关软件，本软件实现了其大部分功能，同时本软件 工作在更流行的操作系统——android 平台之上，使用了更加小巧的特征，并利 用了特征融合的技术，而且苹果上的设备只能工作在联网的状态下并只能用摄像 头拍摄识别，而本软件不仅可以联网使用，还可以本地识别，而且还可以选择本 地设备上已有的图像经行识别。

四、市场前景展望

本系统是基于计算机视觉的数字图像处理技术，其中识别部分主要包括图像分割、特征提取以及识别算法，结果为向用户显示出与输入图像相似的一组植物信息列表，但是当前版本的系统只包含 126 类植物信息，未来我们会不断扩充数 据库，使其包含更多的类别；同时，我们还将致力于寻找更适合的特征提取算法， 从而提高精度以及识别速度；我们也会研究更好的分割算法，从而让拍照识别不 在受限制；同时设计服务器端与客户端分开的思想是在服务器端使用更加复杂，

更加精确的算法，从而拟补客户端对手机性能的缺点，而现阶段，服务器端的算 法仍旧使用和手机客户端一样的特征提取算法，未来我们也会研究出更好的算 法，并将其置于服务器端，供用户使用，让用户更加方便的使用本软件。