# 基于数字图像的花卉种类识别系统的设计与实现

目录

[基于数字图像的花卉种类识别系统的设计与实现 1](#_Toc510610879)

[包含文件 3](#_Toc510610880)

[软件环境： 3](#_Toc510610881)

[图片数据库： 3](#_Toc510610882)

[1系统算法 4](#_Toc510610883)

[系统总体流程 4](#_Toc510610884)

[1.1切割图片 4](#_Toc510610885)

[1.1.1均值滤波 5](#_Toc510610886)

[1.1.2sobel算子 6](#_Toc510610887)

[1.1.3大津法自动阈值处理 6](#_Toc510610888)

[存在问题 7](#_Toc510610889)

[1.1.4闭操作 7](#_Toc510610890)

[1.1.5连通域轮廓提取 8](#_Toc510610891)

[1.1.6寻找中心点 8](#_Toc510610892)

[1.1.7漫水填充 9](#_Toc510610893)

[1.1.8腐蚀膨胀 9](#_Toc510610894)

[1.1.9掩膜处理 10](#_Toc510610895)

[1.2特征提取 10](#_Toc510610896)

[1.2.1提取颜色特征 10](#_Toc510610897)

[1.2.2提取轮廓特征 11](#_Toc510610898)

[1.2.3提取纹理特征 12](#_Toc510610899)

[1.2.4存在问题 13](#_Toc510610900)

[1.3识别模型 13](#_Toc510610901)

[1.3.1创建svm训练模型 13](#_Toc510610902)

[1.3.2结果预测 13](#_Toc510610903)

[Predictions\_GUI 13](#_Toc510610904)

[Train\_GUI 14](#_Toc510610905)

# 包含文件

软件在flowers\_recognition中包含两个文件夹：

**1，predictions目录中有如下重要文件**

flower\_data文件夹，里面存放部分花卉图片数据，可以用来识别。

category\_index.csv，存放花卉类别信息，用于软件识别时对照识别结果。

my\_image.model，svm识别模型（使用8189张图片进行训练过的）

predictions\_GUI.ipynb。jupyter notebook版本的识别界面程序

predictions\_GUI.py。相同的程序的py版本

---------------

predictions\_temp.csv。程序执行时自动生成的，存放识别图片的特征数据

predictions\_temp.jpg。程序执行时自动生成的，对识别图像切割出的花朵部分。

**2，train目录中有如下重要文件**

Test文件夹，里面存放了部分花卉图片数据，可用来对训练界面程序的测试。

imagelabels\_test\_pred.csv。Test文件中图片的种类标签，可用来对训练界面程序的测试。

train\_GUI.ipynb。jupyter notebook版本的训练界面程序

train\_GUI.py。相同的程序的py版本

-----------------------

111.model。识别程序执行时自动生成的，用test文件夹中图片训练出的模型

predictions\_feature.csv。识别程序执行时自动生成的，test文件夹中图片的特征数据。

predictions\_temp.jpg。识别程序执行时自动生成的，对识别图像切割出的花朵部分。

# 软件环境：

软件在anaconda 5.01 的jupyter notebook上编写。

anaconda 5.01编自带的，程语言是python3.63。

win10 64位

# 图片数据库：

图片数据库使用的是：102 Category Flower Dataset

该数据库包含102类花朵图片，每类图片数量在40张至258张。数据库一共包含8189张图片。

本系统使用全部图片进行svm模型训练，对8189张图片均可识别。

但由于该数据库并没有给出102类花朵的标签列表，因此在category\_index.csv文件中仅自己编辑了10中花朵类型名称信息。

预测时仅使用了数据库中的一部分图片进行识别。该部分在predictions/flower\_data目录下，true文件夹中保存的是预测正确的图片，共10类70张图片。false是预测错误图片

# 1系统算法

## 系统总体流程

1. 切割图片
2. 提取颜色特征
3. 提取轮廓特征
4. 提取纹理特征
5. 创建svm训练模型
6. 结果预测

## 1.1切割图片

从一张图片中切割出花朵的部分，去掉背景等对识别没用的图像。

切割图片使用到数字图像处理中的方法，算法流程，算法包括：

这部分参考车牌识别的切割方法。算法原理等可参考 基于SVM与ANN的车牌识别，文件

该部分可以参考picture2文件。

**原始图像：**

Python的opencv读取出的图像时BGR格式的，在显示原始图像时，要先转成RGB图像再显示。

读取使用imread函数自动就是BGR通道，存储使用imwrite函数由BGR存储，保存后也是正常颜色通道。所以在程序里显示要想显示正常要转RGB，但不能用这个处理。



### 1.1.1均值滤波

这是沿用车牌识别的方法，：在寻找垂直边缘之前，我们需要把彩色图像转换为灰度图像，因为彩色图像在我们的任务中没有帮助，并且用均值滤波的方法来移除来自相机或者外界对图像的噪声。如果我们不应用去噪方法，我们将得不到许多的垂直边缘，将会产生检测失败。

均值滤波：使用3\*3模板，对灰度图像进行处理，9个像素计算均值，给中间像素重新赋值，取出噪声。

但是这步貌似没有什么用，因为图像没有什么噪声。



读取进来的原始灰度图像

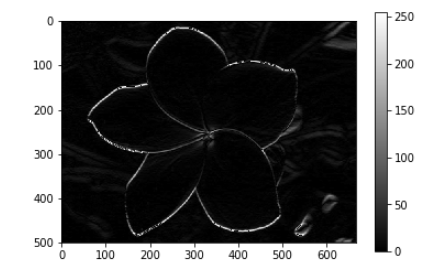
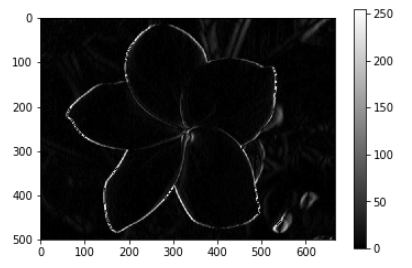


均值滤波处理后的图片

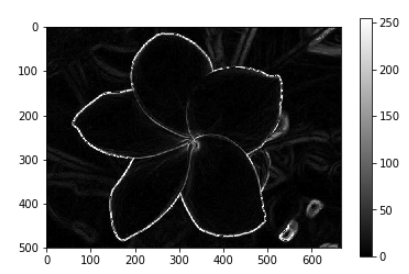
### 1.1.2sobel算子

Sobel算子：得到纹理图像

对灰度图像处理，分别得到X,Y两个方向上的sobel算子图像，两个方向叠加使轮廓更明显。



X方向上的Sobel图像 Y方向上的Sobel图像



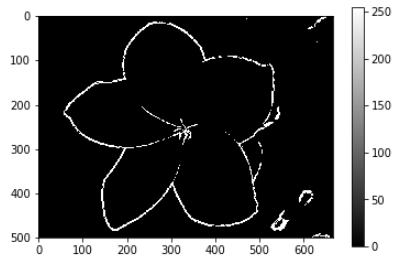
两方向叠加的轮廓

### 1.1.3大津法自动阈值处理

Sobel滤波后，我们应用一个阈值滤波器来获得一个二值图像，阈值通过Otsu方法获得。

大津法自动阈值处理：得到二值图像。这样得到的轮廓更明显

二值图像，仅有黑色，白色。黑色是0，白色是255



### 存在问题

使用上述方法处理得到轮廓图像后，是没有轮廓坐标信息的，仅有一张图片，所以上边是不是有轮廓，不知道

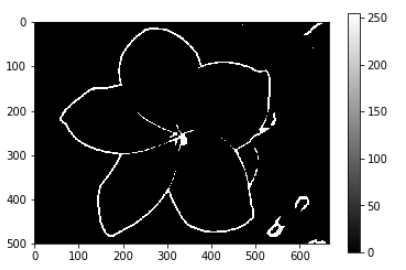
处理后的图像，轮廓连通效果，未知。可能花朵边缘出现断开

除花朵轮廓外，也有其他轮廓

### 1.1.4闭操作

车牌识别：接下来，通过应用一个闭操作（先膨胀,再腐蚀），我们能够去掉每个垂直边缘线的空白黑色部分。并且连接含有边缘数量很多的所有区域。在这一步，我们得到可能的含有车牌的区域。

闭操作：让图像花朵轮廓更明显

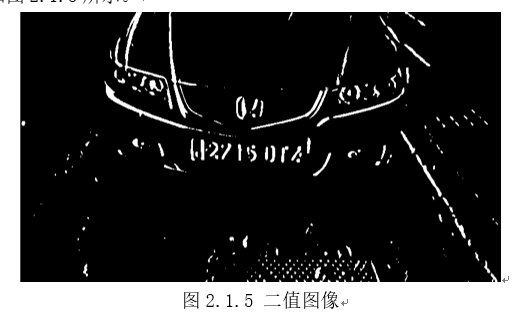
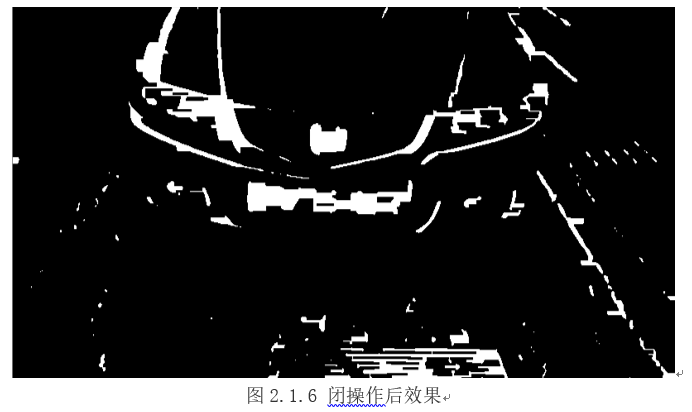


此处闭操作效果

**问题**

此处闭操作是否应该改成膨胀

模板大小是否要再增大。对比车牌识别，图片如下

### 1.1.5连通域轮廓提取

连通域轮廓提取：提取出图像中的连通域轮廓边缘。这一步就拿到了轮廓的位置信息，坐标点数组。

车牌识别：根据闭操作后的图像可使用opencv库中的findContours函数获得连通域轮廓，然后使用drawContours函数将轮廓曲线，画到彩色图像上。**同时也获得一些图像的最小外接矩形。我们基于面积和宽高比，对于检查到的所有外接矩形区域做一下确认，进行估算筛选。 这步骤没做，花朵形状不一定**

取出最大连通域的轮廓：用contourArea函数计算连通域内部面积。图像中花朵占大部分面积，因此轮廓最大的就是花朵边缘。然后用drawContours函数绘制轮廓

此步骤问题：drawContours函数绘制轮廓时，轮廓厚度参数设置成-1，可以向内填充。这样就可以将花朵部分填充成白色。用掩膜操作就可以得到花朵部分。但是用上述方法处理后得到的轮廓，向内填充无效果，怀疑是轮廓不是一个闭合的曲线，有缺口。

因此绘制轮廓前，再进行闭操作，得到更明显的轮廓图像



### 1.1.6寻找中心点

寻找中心点：花朵中心

拿到上述的连通域坐标数组后，计算x、y坐标平均值，得到中心点。这个中心点不一定是花朵中心（花蕊那块），可能存在拍摄角度差异。

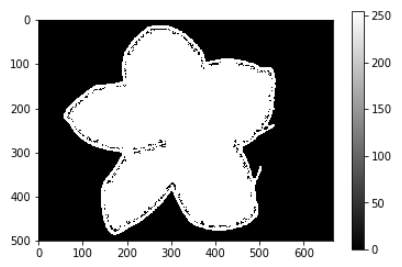


在中心点个红点，就是这个位置

### 1.1.7漫水填充

漫水填充：填充花朵部分

拿到上边的中心点，用floodFill函数进行漫水填充，从中心点向外扩张，扩张到轮廓附近就翻不过去了。此部分是用1.1.5拿到的连通域绘制的轮廓图。填充成白色就能当成掩膜蒙版。



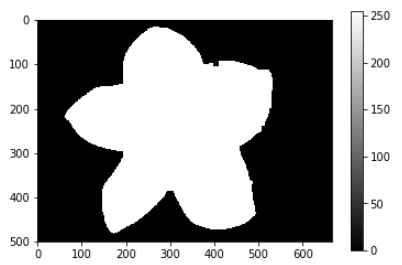
效果如图所示

填充后边缘附近会填充不上，不知道为什么

### 1.1.8腐蚀膨胀

膨胀：得到更明显的花朵部分

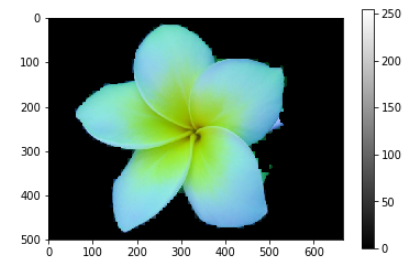
提取连通域，漫水填充后还有杂边。所以想先进性腐蚀，去除杂边，在进行膨胀，填充上花朵里的黑点。此部分腐蚀2次， 膨胀6次。操作次数的科学性有待考量。估计是仅对着一张图好，其他图不一定行



### 1.1.9掩膜处理

掩膜：得到花朵图像

上面拿到的二值图像，当做蒙版，与彩色图像进行与操作bitwise\_and函数，就能得到花朵



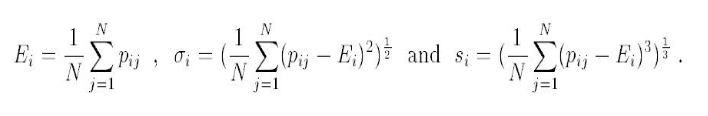
效果如图所示。同样这里显示是用RGB通道显示。保存时要传成BGR通道。

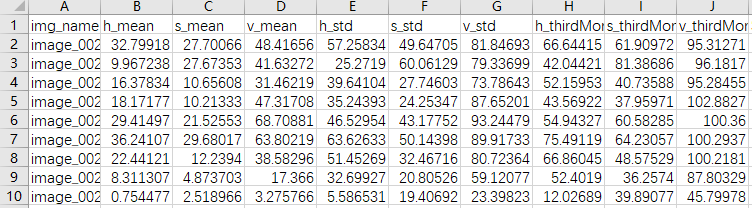
## 1.2特征提取

### 1.2.1提取颜色特征

计算花朵图像的HSV颜色特征。包括HSV三个通道的一阶矩、二阶矩、三阶矩的共九个特征。

颜色矩是一种简单有效的颜色特征表示方法，有一阶矩(均值,mean)、二阶矩(方差,viarance)和三阶矩(斜度,skewness)等，由于颜色信息主要分布于低阶矩中，所以用一阶矩，二阶矩和三阶矩足以表达图像的颜色分布，颜色矩已证明可有效地表示图像中的颜色分布。

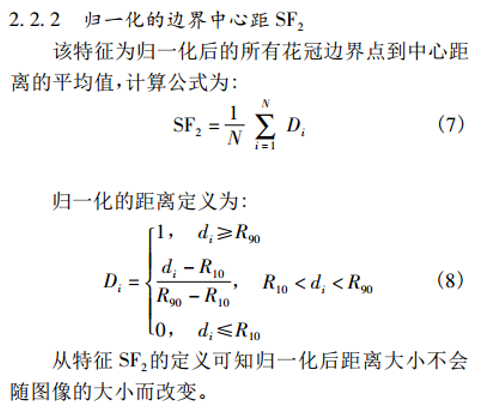
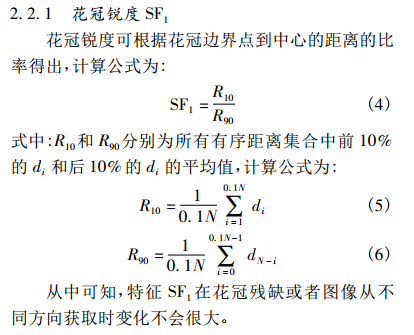


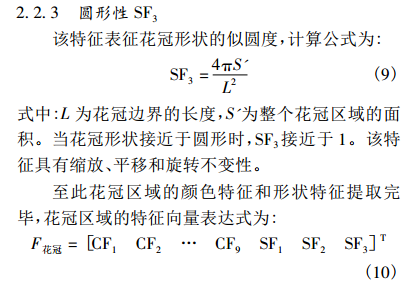


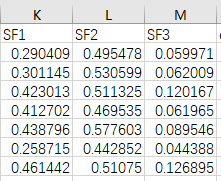
### 1.2.2提取轮廓特征

提取花朵边缘的轮廓特征。包括花冠锐度、中心距、圆形性的三个特征

轮廓特征信息是将用上面连通域轮廓坐标数组计算。花冠锐度SF1、归一化边界中心距SF2、圆形性SF3均参考论文《基于多特征融合的花卉种类识别研究\_吴笑鑫.pdf》







### 1.2.3提取纹理特征

提取出花朵的纹理特征。计算图像的灰度共生矩阵，然后得到能量、对比度和相关性三个特征。

纹理特征参考文章《基于数字图像的花卉种类识别技术研究\_裴勇》。

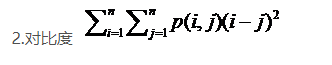
首先用greycomatrix函数获得图像的灰度共生矩阵，然后再用greycoprops计算特征

**能量**

C:\Users\zihao\AppData\Local\Temp\1522139017(1).png

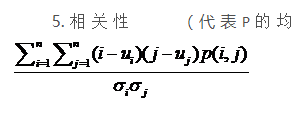
- 'ASM': :math:`\sum\_{i,j=0}^{levels-1} P\_{i,j}^2`

**对比度**



- 'contrast': :math:`\sum\_{i,j=0}^{levels-1} P\_{i,j}(i-j)^2`

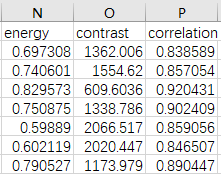
**相关性**



'correlation':

.. math:: \sum\_{i,j=0}^{levels-1} P\_{i,j}\left[\frac{(i-\mu\_i) \

(j-\mu\_j)}{\sqrt{(\sigma\_i^2)(\sigma\_j^2)}}\right]



### 1.2.4存在问题

选取特征后未确定特征是否合适可用，直接训练的

## 1.3识别模型

### 1.3.1创建svm训练模型

模型使用svm。用8189张图片训练模型。

模型有上述的共15个特征作为输入，输出为类别

### 1.3.2结果预测

# Predictions\_GUI

预测界面的程序。包括

图像处理函数：切割花朵

颜色特征提取

轮廓特征提取

纹理特征提取

图片识别

GUI绘图

# Train\_GUI

训练界面程序。

图像处理函数：切割花朵

颜色特征提取

轮廓特征提取

纹理特征提取

图片识别

GUI绘图

# 程序打包问题

编写完程序后使用pyinstaller进行打包，打包时出现UTF-8的编码错误，用 chcp 65001 命令将命令行窗口编码格式改为utf-8格式后，可以正常打包。打包命令为 pyinstaller -F 文件名.py。但在打包过程中出现多个错误找不到dll。打包后运行exe文件，出现找不到dll错误，不能执行

参考链接

https://www.cnblogs.com/mimi0107/p/5755946.html

https://blog.csdn.net/u011529752/article/details/54892488

然后用cx-freeze进行打包，过程中出现错误，未解决。

打算用其他方式，生成发布版本软件。