

电子科技大学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

# 专业学位硕士学位论文

MASTER THESIS FOR PROFESSIONAL DEGREE



论文题目 基于图像处理的

水稻考种系统的设计与实现

专业学位类别 工程硕士

学 号 201651010169

作 者 姓 名 周 洪 垒

指 导 教 师 贾 海 涛 副 研 究 员

分类号\_\_\_\_\_密级\_\_\_\_\_

UDC<sup>注1</sup>\_\_\_\_\_

# 学 位 论 文

## 基于图像处理的水稻考种系统的设计与实现

周 洪 垒

指导教师	贾海涛	副研究员
	电子科技大学	成 都
	王雷	高 工
	成都思维硅信科技有限公司	成 都

(姓名、职称、单位名称)

申请学位级别 **硕士** 专业学位类别 **工 程 硕 士**

工程领域名称 **软件工程**

提交论文日期 **2019 年 3 月** 论文答辩日期 **2019 年 5 月**

学位授予单位和日期 **电子科技大学 2019 年 6 月**

答辩委员会主席\_\_\_\_\_

评阅人\_\_\_\_\_

注 1: 注明《国际十进分类法 UDC》的类号。

# **The Design and Implementation of Rice Seed Counting System Based on Image Processing**

A Master Thesis Submitted to  
University of Electronic Science and Technology of China

**Discipline:** Master of Engineering

**Author:** Zhou Honglei

**Supervisor:** Prof. Jia Haitao

**School:** School of Information and Communication Engineering



## 摘 要

水稻是最古老的食物之一，也是目前中国的第二大粮食作物。水稻种子的千粒重以及粒长、粒宽、周长、面积、等信息经常作为衡量和分析其质量的关键性指标。因为大多数传统的水稻种子计数方法是人工完成的，所以经过长时间的计数后，人眼往往会疲劳，从而导致计算错误并且不能保证精确度。为了保证水稻的高质量生产，提高农业数种的准确性和检测效率，本文采用图像处理技术结合 Qt 设计了水稻考种系统。本文的主要工作有：

(1) 水稻考种系统的硬件设计。选取 800 万像素的 USB 接口摄像头结合 19mm 金属设备工作指示灯，在固定箱体上进行安装，并设计抖动式托盘与外接天平，实现对水稻种子图像与重量信息的实时采集。

(2) 水稻考种系统的图像算法设计。OpenCV 是一个跨平台的计算机视觉库，运用其 C++ 接口，对采集的水稻种子图像进行处理。对图像进行了平滑处理，以减少噪声对算法后续识别处理产生干扰；对采集到的水稻种子图像进行二值化处理，然后对二值化图像进行开操作，从而剔除水稻种子的尾掉部分，得到只含有水稻种子主体信息的二值化图像；对黏连区域的分割，将距离变换与分水岭算法相结合，对多个种子黏连的区域进行分割；又根据种子的形态特征，对黏连区域提出了划线分割算法。为了加快算法处理速度，还在图像处理算法中加入了多线程。

(3) 水稻考种系统的软件界面实现。Qt 是一个跨平台的 C++ 图形用户界面应用程序开发框架，水稻考种系统的软件界面在 Qt 的平台上进行开发。软件中使用了设计模式中的观察者模式，对图像输出的相关统计数据以及显示界面完成联动操作。软件界面操作简单快捷，具有自动考种、水稻种子数据的显示与输出、水稻种子进行紧密排列等功能。

实验结果表明：本系统可以实现对水稻种子的自动计数，计数准确率可以达到 99.12%。本系统较适合对类圆锥形的农作物种子进行计数及形态学分析，因此，对于其他形状的种子，系统的适用性能的提升还需要后续研究。

**关键词：**自动计数，图像分割，距离变换，紧密排列，观察者模式

## ABSTRACT

Rice is one of the oldest foods and the second largest food crop in China. The 1000-grain weight of rice seeds, as well as grain length, grain width, perimeter, area, and other information are often used as key indicators for measuring and analyzing their quality. Because most traditional rice seed counting methods are performed manually, after a long period of time, the human eye tends to fatigue, resulting in calculation errors and inability to ensure accuracy. In order to ensure the high quality production of rice and improve the accuracy and detection efficiency of counting seeds of agriculture, this paper uses image processing technology combined with Qt to design the counting systems of rice. The main work of this paper is:

(1)Hardware design of counting rice systems. The 8 megapixel USB interface camera is combined with the working indicator of the 19mm metal device to install on the fixed box, and the dither tray and the external balance are designed to realize real-time collection of rice seed image and weight information.

(2)Image algorithm design for counting systems of rice. OpenCV is a cross-platform computer vision library that uses its C++ interface to process acquired rice seed images. The image is smoothed to reduce the noise and interfere with the subsequent recognition processing of the algorithm. The collected seeds of rice image binarization process, the binarized image and then opening operation, thereby excluding the tail portion out of rice seeds, rice seeds obtained containing only the body of binarized image information. For the segmentation of the adhesion region, the distance transformation and the watershed algorithm are combined to segment the regions where multiple seeds are adhered. According to the morphological characteristics of the seeds, a scribing algorithm is proposed for the adhesion region.

(3)The software interface implementation of counting rice systems. Qt is a cross-platform C++ graphical user interface application development framework. The software interfaces of counting rice systems are developed on the Qt platform. The MVC design pattern is used in the software, and the data and user interface can be effectively separated by using a model representing the data, a view representing the user interface, and a control defining the user's operation on the interface. The software interface is simple and fast to operate, with automatic counting rice seeds, display and

output of rice seed data, and tight alignment of rice seeds.

The experimental results show that: The system can realize automatic counting of rice seeds, and the counting accuracy is 99.12%. The system is more suitable for counting and morphological analysis of conical crop seeds. Therefore, for other shapes of seeds, the improvement of the applicable performance of the system requires further research.

**Keywords:** automatic counting, image segmentation, distance transformation, tightly arranged, observer mode

# 目 录

<b>第一章 绪 论</b> .....	1
1.1 课题研究的背景与意义 .....	1
1.2 国内外相关领域研究现状 .....	2
1.2.1 种子计数技术现状及发展趋势 .....	2
1.2.2 种子图像分割现状及趋势 .....	2
1.3 论文研究内容 .....	4
1.4 论文结构安排 .....	5
<b>第二章 图像相关技术与系统总体设计</b> .....	6
2.1 图像处理相关技术 .....	6
2.1.1 数字图像处理基础 .....	6
2.1.2 图像形态学处理 .....	6
2.1.3 图像边缘检测算法 .....	11
2.2 硬件系统设计 .....	12
2.2.1 工业相机以及相机镜头的选择 .....	13
2.2.2 照明系统的设计及电子天平的选择 .....	16
2.2.3 采集箱体的设计 .....	17
2.3 软件系统总体设计 .....	18
2.3.1 软件系统开发总体框架 .....	18
2.3.2 软件系统相关开发工具 .....	19
2.4 本章小结 .....	20
<b>第三章 种子图像的分割与识别算法</b> .....	21
3.1 种子图像分割算法 .....	21
3.1.1 轮廓凸包与缺陷点检测算法 .....	21
3.2.3 二值图像的距离变换 .....	22
3.2 种子黏连区域分割算法实现 .....	22
3.2.1 算法总体设计 .....	22
3.2.2 图像预处理 .....	23
3.2.3 种子黏连区域的分割 .....	25
3.3 种子紧密排列图像处理算法 .....	29
3.3.1 自动聚类 .....	30



3.3.2 单个种子的抠取与排列 .....	32
3.4 本章小结 .....	34
<b>第四章 系统核心功能详细设计 .....</b>	<b>35</b>
4.1 图像的采集与显示 .....	35
4.1.1 实时图像的显示与采集 .....	35
4.1.2 采集图像的显示与相关操作 .....	37
4.2 统计信息的计算与生成 .....	38
4.2.1 单个种子统计信息的计算与生成 .....	38
4.2.2 总体统计信息的计算与生成 .....	39
4.3 统计信息的增删查改与输出显示 .....	40
4.3.1 图层技术 .....	40
4.3.2 统计信息与显示图像的联动 .....	41
4.3.3 统计信息的 Excel 表格输出 .....	43
4.4 电子天平与 LED 灯的通信和控制 .....	43
4.4.1 电子天平的通信 .....	43
4.4.2 LED 灯的控制 .....	44
4.5 本章小结 .....	45
<b>第五章 系统的实现 .....</b>	<b>46</b>
5.1 系统主要操作界面 .....	46
5.2 系统主要操作流程 .....	47
5.2.1 图像采集的操作流程 .....	47
5.2.2 图像处理与相关输出信息操作流程 .....	53
5.2.3 对考种统计信息删除与定位的操作 .....	59
5.2.4 对天平与 LED 灯的操作 .....	60
5.3 系统测试 .....	62
5.3.1 软件系统兼容性测试 .....	62
5.3.2 算法稳定性测试 .....	63
5.4 系统验证 .....	66
5.5 本章小结 .....	68
<b>第六章 总结与展望 .....</b>	<b>69</b>
6.1 总结 .....	69
6.2 展望 .....	70
<b>致谢 .....</b>	<b>71</b>

参考文献 .....	72
攻读硕士学位期间取得的成果 .....	75

## 第一章 绪 论

### 1.1 课题研究的背景与意义

在我国当前的国情下，农业生产不仅对农业经济有相关影响，对整体经济的发展也起着重要的作用，所以需要重视农业并且将农业经济与整体经济紧密的联系起来。而水稻不仅是其中最大的粮食作物，也是其中单产量最高的粮食作物。近年来，通过农业技术推广和服务体系的不断完善，水稻品种的改良也十分明显，但水稻单产量仍增长缓慢。我国 60% 以上的人口以大米为主食，而且是在城乡居民中这个比例又要大幅上升<sup>[1]</sup>。水稻生产的发展对解决我国粮食供应安全具有重要作用，而考种就是水稻育种的重要一环，可以为选种者的选种提供重要的依据<sup>[2]</sup>。

在中国，农业是一个需要长期关注和发展的产业，整体经济趋势与农业经济密切相关。

水稻生产中，我们通常使用 1000 粒水稻种子的重量作为衡量种子质量的重要指标，以实现高质量的水稻生产，而测量 1000 粒种子重量必不可少的工作是就计算种子数量。每年，成千上万的农业机构和科研单位都需要对水稻种子进行测量统计，主要是统计一定批次内种子的个数以及测量粒长、粒宽等数据。此外，当使用自动机械化播种时，还必须预先计算传送带上的种子数量以确定播种量。然而，使用一些传统的种子计数方法却具有效率低和精度有限等缺点。目前，水稻种子的计数和相关参数的测量一般基于三种方法，即人眼的手动测量，光电测量仪器和基于图像处理的测量仪器<sup>[3-4]</sup>。其中，人工测量是最传统和最常用的方法，它主要依靠人眼来统计种子数量，依靠部分测绘工具统计部分水稻种子的重量和粒长、粒宽等数据。但是，人工测量劳动力成本高，测量时间长，工作量大，长期高强度容易引起视觉疲劳，造成二次统计误差，严重影响测量效率和准确性。光电测量仪器结构复杂，价格昂贵，并且工艺数量相对较少，因此最终只能获得种子的数量信息，并且需要手动记录<sup>[5]</sup>。近年来，计算机的计算能力不断发展，随着带来了图像处理技术的迅猛发展，研究人员与相关的开发人员开始逐步的将计算机图像处理技术运用到了农业的优选生产上，并开发了几种基于图像处理的考种设备<sup>[6-12]</sup>。然而，由于诸如功能性较差和未能很好的减少人工工作量等原因，这些仪器中的大部分都并未广泛传播和应用。

针对种子计数方式存在的一系列问题，在前人研究的基础上，本文设计了一款基于图像处理技术的水稻种子自动计数系统，可以实现种子计数、尺寸测量（单粒种子的长宽、面积、长宽比）千粒重等信息。

## 1.2 国内外相关领域研究现状

### 1.2.1 种子计数技术现状及发展趋势

在早期阶段，中国农业水平不高，处于不完全发展阶段，尚未开发出相关的农作物测量技术，大量的种子计数的工作都是由人工人眼来完成的。然而，人工人眼的判断具有很大的主观性和偶然性，加之人眼在长时间的高强度计数工作下，不同的个体会产生不同程度上的计数误差。

与人工人眼计数相比，光电测量仪的考种方式避免了主观和偶然性的误差产生。1995 年，张小津、刘顺康等<sup>[13]</sup>研究了光电管计数并在其基础上发明了一款震动电子考种仪；2011 年，宋初苏、蓝景针等<sup>[14]</sup>等研制出了转盘斜刮式自动光电考种仪，并具有计算千粒重的功能。这些都是在光电管计数的基础上开发出来的考种仪。然而，使用光电管计数的方法存在两个硬件上的缺点：（1）光电管具有滞后相应的属性，其灵敏度随时间急剧下降，需要在短时间内更换。此外，使用光电管进行计数所带来的滞后对计数的准确性有很大影响；（2）在计算不同尺寸的种子时，测量仪器需要匹配相同尺寸的部件，这使得操作的过程非常麻烦。

上述两种方法各方面都存在较大的缺陷，而农业生产的迅速发展也使得自动检测和计数技术得到了长足进步。基于数字图像处理的考种系统，与人工计数相比，具有十分显著的优势。计算机视觉可以迅速且快速地获取检测和识别所需要的特征信息，并根据需要测量数据自动处理和输出相关信息。相较于人类的视觉系统，在数字图像处理基础下的识别系统可以迅速获取得到非常多的图像特征信息，而与人眼相比，计算机不会产生眼睛过度使用带来的眼疲劳现象，这就使得使用计算机进行考种工作带来了操作便捷性与快速性的优势甚至消除了特定地点进行考种的限制。以往的一些因为危险系数过高的生产任务，随着机器视觉、图像处理技术的快速发展，也有了实现的可能<sup>[15]</sup>。目前，图像处理技术发展迅猛，物体识别、种子计数、形态参数测量等已经被广泛应用于作物育种、生产测量、和质量鉴别等方面<sup>[16-18]</sup>。

总之，在现代农业机械生产和数字图像处理技术中，计算机可以独立于人的主观因素来进行检测任务。这种优势使其不仅降低识别成本，也实现了高精度的识别，去除了识别过程中的主观评判性，其应用前景也十分的广阔。

### 1.2.2 种子图像分割现状及趋势

在数字图像处理技术的快速发展下，人们逐渐将研究重点放在了基于图像处理技术的自动考种系统，该系统能够准确且快速地计算采集的种子图像中包含的种子数量，并且输出衡量种子优劣的部分指标，如：千粒重、粒长、粒宽等。然

而，在实现过程中，最核心也是最难解决的问题还是在种子图像处理过程中，需要对图像中的黏连的种子区域进行分割并将其准确的识别为单个种子。近年来，国内外许多研究人员对这方面进行了研究并取得了一些良好的进展，并且能够初步的替代传统考种方式。

在对黏连区域分割研究的早期，大部分的成果主要还是来自于医学图像的研究。在国外的研究中，研究人员 Gallas<sup>[19]</sup>完成了对染色体条纹带形状的提取与刻画。为了显示出图像中的局部特性，其首先利用图像增强技术凸显出图像的局部特性，然后滤除图像中的各类噪声，最后将输出的图像进行闭值分割处理，从而达到染色体条纹带形状的提取目的。虽然可以实现目标，但是这种处理方法需要等待的时间太长且识别的精度不太理想。Jean<sup>[20]</sup>也将黏连区域的分割技术运用到了医学图像领域，完成了癌细胞的参数提取，主要涉及到的是数学上的形态学方法在图像处理中的应用，将其与正常细胞区别开来，这个算法运行时间在可接受的范围之内且识别的精度较高，但该算法的适用性不强。在 2002 年，王金涛、刘文耀<sup>[21]</sup>等将流域分割算法应用到了细胞图像分割当中。2008 年，闫文忠、沈树群等人<sup>[22]</sup>在第六届全国信息获取与处理学术会议上提出了基于分水岭算法的黏连染色体图像分割。2009 年，马保国、乔玲玲、贾寅波等人<sup>[23]</sup>在期刊计算机应用研究中提出了对采集到的细胞图像进行分割的新技术，这项技术的是基于部分自适应阈值算法进行实现的，具有重要的研究意义。

随着农业的不断发展，农业现代化需求的不断增强，越来越多的图像处理技术开始应用到农业生产当中，图像黏连区域的分割技术就是其中应用的一个重要部分。近年来，许多研究人员正在研究几种粘附种子图像的分割，并且取得了非常不错的成果。Van den Berg EH<sup>[28]</sup>对农产品对它的纹理和其他信息进行提取，然后进行识别。Zayas.I.Y<sup>[29]</sup>使用小麦的切线值，长度和宽度值以及长宽比系数来识别软冬小麦和硬冬小麦，最终的识别正确率可以在 80%以上。Talukde<sup>[30]</sup>利用灰度变换，噪声去除滤波等对图像进行预处理，然后使用分水岭算法对图像进行分割，以实现多种农业产品的识别和检测，该算法适用性较强，但是存在一定的误差。2009 年，毋桂萍、杨红卫等人<sup>[31]</sup>在期刊农业化研究中提出了一种改进的 Otsu 算法对粮食颗粒图像进行分割。2012 年，刘娜等人<sup>[32]</sup>也对颗粒黏连图像的分割作了一定的研究，在文章中，提出了一种基于灰度形态学重建的分割算法，其识别精度与识别速度与前人的相比又有了很大的提升。

综上所述，随着计算机计算能力的不断进步与发展，越来越多的研究人员开始使用图像处理相关的技术知识来分割粘附种子的图像，这样不仅可以确保识别速度，还可以确保识别的准确，相比传统的考种方式，又可以减少大部分的物力

与人力，减轻农业生产优化的成本。随着研究的深入，软件、硬件配套设备适配性的不断增强，这种方法在一定会在现在和未来加以利用，以使农业生产更高效、更便捷。

### 1.3 论文研究内容

本论文研究了基于数字图像处理的水稻考种系统。在前人研究的图像分割方法的基础上，设计了一种适用于水稻种子的考种系统。针对现有的低效数种方法与低精度的黏连区域分割算法，结合种子的形态特征、面向用户的使用习惯与使用需求，完成了以下几方面的工作：

（1）分析水稻考种系统的需求关系，组建水稻考种系统的硬件部分。

对水稻考种系统的需求做了详尽分析，分析其输入与输出，将硬件部分大致分为三个模块：图像采集模块、照明模块、采集箱体模块。根据实际情况完成对图像采集镜头、照明 LED 灯以及箱体的选型。

（2）改进基于分水岭算法的黏连区域分割算法。

目前基于分水岭算法的黏连区域分割算法应用相对广泛，但分割的效果不算太好，且大多数都需要与迭代腐蚀想结合，这势必会造成极大的时间损耗。针对这个问题，本文对基于分水岭算法的黏连区域分割算法进行了改进，将分水岭算法与二值图像距离变换相结合，从而降低时间损耗，且分割效果也更为理想。

（3）结合种子形态特征，改进黏连区域划线分割算法。

种子具有椭圆的形态特征，根据黏连区域产生的凹凸点，结合缺陷点检测技术，得到凸包缺陷的起点、终点、最远点等点集，对可疑区域进行划线，实现黏连区域的二次分割，从而提高了分割的成功率以及种子识别的精度。

（4）针对农业生产科研人员需求，改进种子紧密排列图像处理算法。

为了便于水稻生产研究人员对不同种型的种子进行粒长、粒宽的对比，对种子紧密排列的图像处理算法进行了改进，并且结合 K 均值聚类算法，对图像中不同摆放位置的种子进行自动聚类，从而达到自动分类的效果。

（5）搭建水稻考种系统的用户图像界面。

使用 Qt 开发框架搭建整个系统的用户图形界面。实现了用户界面与硬件系的有效通信。为了便于用户操作，对 Qt 中的部分控件进行了重写，还将 PhotoShop 中的图层技术引入其中。对于算法输出的统计信息，使用了设计模式中的观察者

模式，并提供了输出统计信息到相关文档的功能。

## 1.4 论文结构安排

针对上述的研究内容，本论文的安排如下：

第一章，绪论。首先详细分析了课题的探究意义、背景，此外还综述了国内外对此的研究现状。最后，还对本论文的主要研究内容和结构安排做了简单介绍。

第二章，图像相关技术与系统的总体设计的介绍。介绍了图像处理的基础理论知识。详细说明了水稻考种系统的硬件部分的设计，说明了系统运行的环境与硬件的需求，阐述了硬件选取的原则与相关原理。对系统的软件部分做了总体的描述。

第三章，种子图像的分割与识别算法。详细介绍解释了分割处理算法中涉及到的图像处理基础知识，分块对算法中的每一个模块进行了详细阐述。

第四章，系统核心功能详细设计。展示了软件系统中的用户图形界面的流程图，详细阐述了其中的关键技术设计。

第五章，系统的实现。对软件的整体界面以及各项功能的操作做了介绍说明。对考种系统的适配性以及算法的稳定性做了相应测试与验证。

第六章，总结与展望。反思系统存在的不足，对后续的研究工作进行了展望。

## 第二章 图像相关技术与系统总体设计

### 2.1 图像处理相关技术

#### 2.1.1 数字图像处理基础

数字图像指的是二维或三维的像素矩阵，像素指代的是图像空间的坐标，像素值即矩阵对应坐标处所存储的数值。一般我们认为的数字图像处理技术指的是对转换成数字信号的数字图像进行像素级别的处理。图像处理技术起源于 20 世纪 60 年代，发展至今，已经取得了惊人成绩，在很多领域都有着良好的应用。

亮度、对比度、直方图是图像的三个基本属性。其中亮度的大小代表着人视觉上的明暗效果，在某些情境下，也可以称之为灰度。合适的对比度大小可以给人眼更快速的识别图像中特定的识别目标。直方图是关于图像灰度值的一个统计图，可以较为清晰明了的观察图像灰度分布，便于后续处理。

数字图像处理有常用的方法包括：图像描述、图像编码与压缩、图像增强与复原、图像变换、图像分割、图像分类等。图像描述指的图像需要识别或处理任务的特征表征，为后续的物体识别以及其它形式的图像处理做好相关的准备。图像压缩与编码指的是图像存储的相关技术，不同的压缩编码技术会以不同大小以及不同的格式存储在数字存储设备中。对图像进行增强或复原操作后，可以凸显出所需要的任务区域或者将任务区域更高质量的呈现。图像变换指的是图像转换到时域以外的域，例如转换成频域的傅里叶变换等，图像变换可以对图像进行多域处理，增加处理任务完成的可能性。图像分割是图像处理技术中最常涉及的技术之一，主要是在给定图像根据需求，将特定的目标与图像的背景分割开来，但由于图像中的背景或者目标都十分复杂，图像分割技术直至今日，也是图像处理领域的一大研究热点。图像分类是将图像中特殊的场景或者物体进行分类，例如：给定一张拍摄到的动物图像，需要判别其是狗、猫亦或是其它的动物。

数字图像处理中，通常都需要首先对噪声进行处理。处理噪声的方法主要是通过滤波，使用图像变换将图像转移到频域，然后进行高通滤波或低通滤波等操作，滤除图像中的噪声。

#### 2.1.2 图像形态学处理

##### 2.1.2.1 图像处理中形态学的基本概念

图像中的形态学因为与数学上的形态学知识体系相接近，所以也可以称之为数



学形态学。数学形态学是一种形态实现相应问题分析的数学工具，格论和拓扑学是其中的理论基础<sup>[33]</sup>。

图像中的形态学处理在实际的运用中十分常见，其处理分析原理主要是通过通过对前期的图像特征识别、特征提取，进而对图像后期的特征分析以及物体检测与识别做好相应的准备工作。其主要的基本运算包括：二值腐蚀和膨胀、二值开闭运算、骨架抽取、极限腐蚀、击中击不中变换、形态学梯度、Top-hat 变换、颗粒分析、流域变换、灰值腐蚀和膨胀、灰值开闭运算、灰值形态学梯度等<sup>[34]</sup>。

在考种系统的图像处理算法中，主要涉及到的形态学知识包括二值腐蚀和膨胀、二值开闭运算、分水岭算法等等。通过这些形态学操作，可以实现对水稻种子边缘特征的提取，以便于下一步的识别与处理。

### 2.1.2.2 腐蚀和膨胀

首先介绍一下结构元素的概念，设有一副图像  $D$ ，表示为被结构元腐蚀或膨胀的图像；又有一副图像  $S$ ，表示为对图像  $D$  进行相应的腐蚀或膨胀操作的结构元素。类似  $S$  的结构元素在一般的情况下都会将其定义为较小的尺寸， $D$  与  $S$  之间的关联可以类比于图像滤波处理中的模板与原始图像之间的关系。其次所有形态学运算都是针对图像中的前景物体进行的，而在实际的 OpenCV 处理中，规定需要识别或者提取特征的物体对象使用黑色表示，而其它的背景则使用白色来表示。在本章节所演示的形态学处理图中，由于使用的 Matlab 中的图像处理库函数进行处理，所以处理时黑色代表背景，白色代表前景。

图像的腐蚀和膨胀在形态学处理中具有举足轻重的地位，因为这两个处理方式是后来进行的许多形态学操作的基础，图像腐蚀与膨胀相结合可以组成其它的许多形态学算法。

腐蚀的运算符是“ $\ominus$ ”，其定义如式 2-1 所示。定义式上表示的是结构元素  $B$  对  $A$  的一个腐蚀，腐蚀的结果是用  $z$  平移的  $B$  包含在  $A$  中的所有的点  $z$  的集合。

简单来说，就是让  $B$  在  $Z$  平面上进行上下左右的平移，在  $B$  移动的过程中，若其中心原点可以全部都包含在  $A$  中，则将符合这种条件的所有点都集合起来，即可得到腐蚀操作的结果。腐蚀操作可以达到消除物体对象之间边界的目的，不同大小的结构元能够造成不同效果的腐蚀情况<sup>[35]</sup>，若结构元过大，可能出现物体完全消失的现象，所以选择好结构元的大小尺寸是腐蚀操作的关键之处。图 2-1 展示了结构元大小为  $3 \times 3$  的情况下，对图像的腐蚀效果，其中左图为原图，右图为腐蚀后的效果图。

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (2-1)$$



图 2-1 腐蚀效果对比图

膨胀的运算符是“ $\oplus$ ”，其定义如式 2-2 所示。定义式上表示的是结构元素  $B$  对  $A$  的膨胀，是所有位置  $z$  的集合。与腐蚀操作恰恰相反，膨胀操作的作用为“粗化”二值图像中的物体，同样，膨胀操作的具体效果要视图像本身与结构元的大小而定。图 2-2 展示了结构元大小为  $3 \times 3$  的情况下，对图像的膨胀效果，其中，左图为原图，右图为膨胀后的效果图。

$$A \oplus B = \{z | (\widehat{B})_z \cap A \neq \emptyset\} \quad (2-2)$$



图 2-2 膨胀效果对比图

### 2.1.2.3 开运算和闭运算

开运算主要是对目标图像先进行腐蚀操作再进行膨胀操作，其运算符为“ $\circ$ ”，其定义如式 2-3 所示。在定义式中， $B$  表示的是结构元素，即  $B$  对  $A$  先进行腐蚀再进行膨胀则得到开运算的结果。开运算的作用主要是去除噪声和小物体、平滑较大物体的边界的同时不明显的改变其面积、在纤细点处分离物体等。图 2-3 展示了结构元大小为  $3 \times 3$  的情况下，对图像的开运算效果，其中，左图为原图，右图为开操作后的效果图。

$$A \circ B = (A \ominus B) \oplus B \quad (2-3)$$



图 2-3 开运算效果对比图

闭运算主要是对目标图像先进行膨胀操作再进行腐蚀操作，其运算符为“ $\cdot$ ”，其定义如式 2-4 所示。在定义式中， $B$  表示的是结构元素，即  $B$  对  $A$  先进行膨胀再进行腐蚀则得到闭运算的结果。闭运算的作用主要是平滑物体轮廓、连接相对窄的间断点和沟壑、填补断裂的轮廓线等。图 2-10 展示了结构元大小为  $3 \times 3$  的情况下，图像的闭运算的效果，其中，左图为原图像，右图为闭运算后的效果图。

$$A \cdot B = (A \oplus B) \ominus B \quad (2-4)$$



图 2-4 闭运算效果对比图

#### 2.1.2.4 分水岭算法

分水岭算法将灰度图像视为拓扑平面，灰度值高的区域看成山峰，灰度值低的区域看成山谷，我们向图像上所有的“山谷”注入不同颜色的水，不断的注水，水位则不断上升，注入的水将灌满山谷，并可能淹没山峰，为了防止不同颜色的山谷中的水溢出汇合，我们可在汇合的地方筑起堤坝，故可将堤坝看作是对图像的分割后形成的边界。通俗来说，就是将图像比作一幅地貌，然后通过最低点和最高点去分类。

原始的分水岭算法主要分四步进行。第一步需要找到图像的局部最低点，如图 2-5 中的①所示两个为局部最低点，可以使用一个内核去寻找，也可以通过遍历比较来寻找；第二步，从最低点开始注水，水沿着梯度上涨，其中那些最低点已经被标记，不会被淹没，那些中间点是被淹没的，在图 2-5 中注水沿着箭头的方向；第三步，寻找局部最高点，在图 2-5 中为③表示的两个点；第四步，基于局部最小值和局部最大值，对图像进行分割。但是原始的分水岭算法由于图像上存在噪声和图局部不连续的情况，常常出现过度分割的情况，如图 2-6 所示。其中，左图为原图，右图为出现过分割情况的效果图。

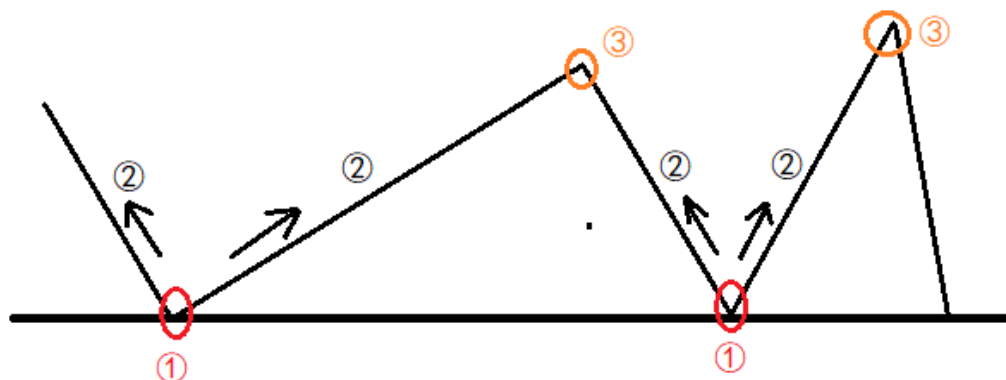


图 2-5 原始分水岭算法示意图

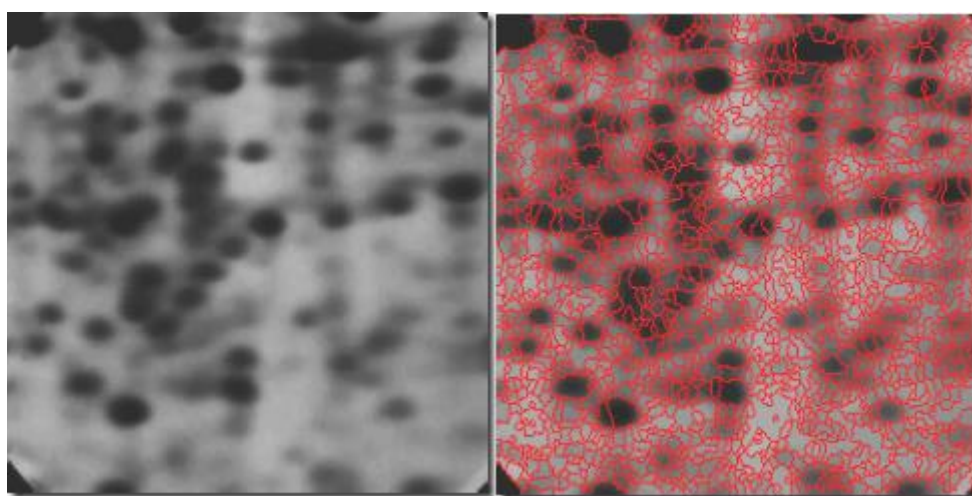


图 2-6 原始分水岭算法的过分割示意图

分水岭算法的改进点之一是使用了标记图像，然后利用标记图像来进行分水岭算法分割。由于噪声的存在以及图像中连接物体的一系列特征，传统的分水岭算法对包含连接物体的灰度图像很难取得满意的分割结果。所以我们可以先通过先验知识来指导分水岭算法，加入一定的标记图像，从而对原始的分水岭算法进行了改进。在一般的情况下，标记图像都是在某个区域之内定义了一些灰度层级，而在这个区域在洪水淹没过程中，它的水平面都是从定义的高度开始的，这样的话可以避免一些噪声很小的极值区域被分割，从而避免过分割现象与误分割现象的发生。使用改进的分水岭算法的效果如图 2-7 所示，其中左图展示的是带标记的原始图像，右图展示的是改进的分水岭算法的效果图，可以很明显的看出这减少了过分割，取得了更好的分割效果。

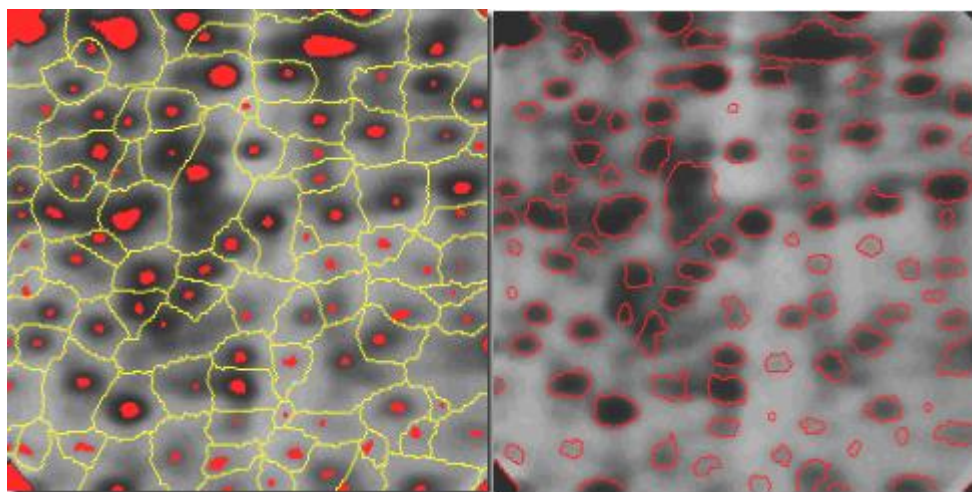


图 2-7 改进的分水岭算法分割效果

### 2.1.3 图像边缘检测算法

梯度算子可分为 3 类：第一类是使用差分近似图像函数导数的算子。有些是具有旋转不变性的(如：Laplacian 算子)，因此只需要一个卷积掩模来计算。其它近似一阶导数的算子使用几个掩模；第二类是基于图像函数二阶导数过零点的算子；第三类是试图将图像函数与边缘的参数模型相匹配的算子。

可以将梯度算子大体上分为三类，其中第一类代表的是类似于求图像的梯度的算子，其涉及到了梯度的计算方式，一般来说采用的是差分的方法。

Laplacian(拉普拉斯)算子是最简单的各向同性微分算子，它具有旋转不变性。一个二维图像函数的拉普拉斯变换是各向同性的二阶导数，其定义为式 2-5。图 2-8 展示了拉普拉斯算子的四种模板，其中图中 a 表示的是式 2-5 的实现，b 是 a 的拓展的模板，c、d 都是实践中比较常用的模板。

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (2-5)$$

0	1	0	1	1	1
1	- 4	1	1	- 8	1
0	1	0	1	1	1
<b>a</b>			<b>b</b>		
0	- 1	0	- 1	1	- 1
- 1	4	- 1	1	8	- 1
0	- 1	0	- 1	1	- 1
<b>c</b>			<b>d</b>		

图 2-8 常见拉普拉斯算子的模板

Canny 边缘检测技术具有广泛的应用，算法核心是从获取到的图像中获取后续处理所需要的结构信息，这可以达到缩小处理数据范围的目的。算法首先需要使用高斯滤波器对图像进行平滑处理，因为噪声可能会对算法处理的结果造成一定的影响。然后需要用一阶偏导有限差分来计算图像的梯度值，图像中的边缘可以指向各个方向，因此 Canny 算法使用四个算子来检测图像中的水平、垂直和对角边缘。第三步需要对梯度幅值进行非极大值抑制，非极大值抑制是一种边缘稀疏技术，非极大值抑制的作用在细化边缘部分。第四步是用双阈值算法检测和连接边缘。一般的边缘检测算法用一个阈值来滤除噪声或颜色变化引起的小的梯度值，而保留大的梯度值。Canny 算法应用双阈值，即一个高阈值和一个低阈值来区分边缘像素。如果边缘像素点梯度值大于高阈值，则被认为是强边缘点。如果边缘梯度值小于高阈值，大于低阈值，则标记为弱边缘点。小于低阈值的点则被抑制掉。图 2-9 展示的是 Canny 边缘检测的效果图，其中，左图为原图，右图为 Canny 边缘检测的效果图。



图 2-9 Canny 边缘检测效果图

水稻考种系统是收集到需要进行检测的一批次水稻种子或者部分水稻种子后，然后对这一批次种子的千粒重、粒长、粒宽等信息进行输出显示或者输出报告报表。本系统的组成部分主要包括硬件层面的图像采集系统和计算机软件系统两大部分。其中硬件层面的图像采集系统主要包括采集箱体、工业摄像头、照明系统、称重天平四部分。计算机软件系统又可以分为用户可视化操作界面和种子图像处理分割算法两部分。

## 2.2 硬件系统设计

硬件系统中的图像采集系统将采集到的同一批次的种子图像进过一系列采样与量化操作后，转换为可在计算机上使用的数字图像。计算机系统上为水稻考种开发的软件接收到采集到的种子图像后，在用户可视化操作界面上进行显示并为

用户提供包括图像处理等一系列操作和设置的接口。通过图像处理算法可以将采集到的图像中的种子进行识别并输出相关参数。在这一节，首先介绍水稻考种仪的硬件系统设计，图 2-10 给出了硬件系统主要部分的设计框架。

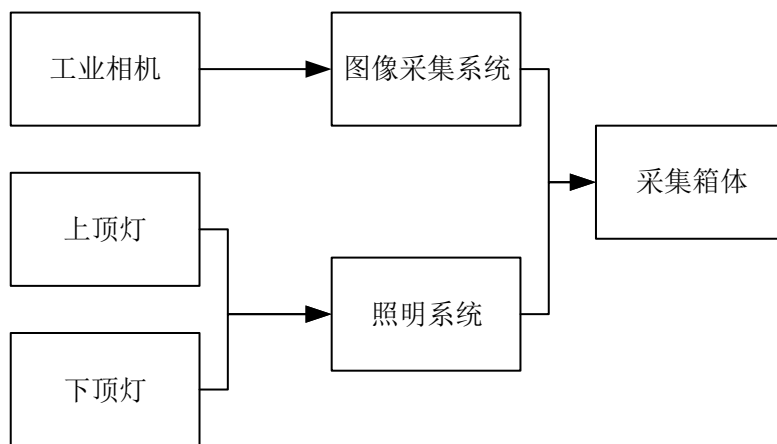


图 2-10 硬件系统主要部分的设计框架

要想实现基于图像处理的水稻考种，显而易见，我们首要需要考虑的是图像来源的问题，水稻考种仪根据实际情况以及实际生产研究中的应用设计了本系统的硬件部分也即图像采集系统。优良的图像采集设备以及干扰性小的图像采集背景环境，往往能够对图像处理算法减轻很大的负担，这样就可以让算法专注于分割本身，而不会花费太多的精力在去噪、去干扰这一方面。所以，设计一套干扰小、清晰度高、抗噪能力强的种子图像采集系统是十分有必要的，也可以说这会直接关系到最后水稻种子识别的精度甚至是对种子能否识别产生重大的影响。

本文中的图像采集系统主要是由采集箱体、工业相机、照明系统和称重天平这四部分构成。工业相机嵌入采集箱体顶端的正中央，以达到覆盖底部所有视角的目的；照明系统包括上顶灯和下底灯，可以保证采集过程中各个角度的光源；称重天平可以将测量的一批次的种子质量精确到 0.01 克；采集箱体底部加入了震动功能，可以使种子通过抖动变得更为分散，从而提高识别精度；采集箱体采用封闭的形式，以避免箱体外部噪声的干扰。

### 2.2.1 工业相机以及相机镜头的选择

采集相机，它就像是人的眼睛，将看到的信息传递给大脑。因此这个“眼睛”是硬件部分图像采集系统中非常关键的一个环节。而这个“眼睛”的关键之处又在于其两大部件。第一个部件是感光芯片，其主要作用是将镜头产生的光学图像转换为计算机可以接收到的数字信号。第二个部件是数据接口，它主要负责的是将这些信号进行有序的输出传送到计算机主机。给系统搭配一个合适的相机，不仅可以



使系统采集得到高质量的种子图像，还将简化采集箱体的搭建。由于在进行水稻种子识别时，我们不仅要完成计数的任务，还需要输出单粒种子的形态学参数，这使得我们对相机的像素以及成像精确性有了较高的要求。

为了更好的选择相机，我们先了解一些机器视觉中工业镜头中的一些概念以及计算公式<sup>[36-39]</sup>。首先是物距也即工作距离（Work Distance, WD），在本课题中代表的是镜头到采集箱体底部放置种子位置的垂直距离；其次是视场也即视野（Field of View, FOV），表示相机实际拍到区域的尺寸，在本课题中代表的是镜头所能拍摄到的采集箱体底部的长度和宽度；再其次是景深（Depth of Field, DOV），景深是指在被摄物体聚焦清楚后，在物体前后一定距离内，其影像仍然清晰的范围；PMAG 代表的是镜头的放大倍数，可由式 2-6 计算求得；f 代表的是镜头的焦距，可由式 2-7 计算求得。

$$PMAG = \frac{Sensor\ Size}{FOV} \quad (2-6)$$

$$f = \frac{WD \times PMAG}{1 + PMAG} \quad (2-7)$$

首先是针对相机接口的选择，由于考种系统需要将采集到的图像数据传输到计算机中，然后再调用图像处理算法进行处理，考虑到考种仪系统使用用户的适配性，因为 USB 接口在较为常见，所以将接口类型限制为 USB 接口更为便捷。

其次是相机的选择，因为在实际的采集环境中，若使用低分辨率的相机进行采集，会使得产生较大的噪声干扰，进而对识别造成影响，为了提高系统以及识别算法的可靠性与稳定性，我们将每毫米所占的像素个数规定在 10 个以上，根据饱和式采集本课题最终将相机的分辨率定为 800 万像素。

然后还需要考虑镜头的畸变因素，由于水稻考种需要获得的精确的参数指标，所以要求镜头为无畸变镜头，以减小畸变的因素对最后输出相关技术指标数据的影响。

然后还有感光芯片的选择，感光芯片主要由 CCD 与 CMOS 两种，如果要求采集的对象是运动的，要处理的对象也是实时运动的，那么选择搭载 CCD 感光芯片的相机是最为适宜的。但水稻考种仪系统采集的对象是静态的，且处理的对象也是静态的，所以水稻考种系统的图像采集采用的是静态采集的方式，因此搭载 CMOS 的感光芯片可以符合要求。

结合以上要求，本课题选取瑞尔威视的 RER-USB8MP02G 高清工业摄像头模组，支持 USB2.0 的高速接口，采用索尼的 IMX179 CMOS Sensor 作为感光芯片，同时搭载 M12 1000 万像素高清镜头作为系统的“眼睛”，最高有效像素为 880 万，



且镜头为无畸变的高拍仪镜头。此套相机模组符合水稻考种系统的主要需求，且性能在同价位内为中等偏上水平。此产品的外观如图 2-11 所示，其具体参数可参考表 2-1。

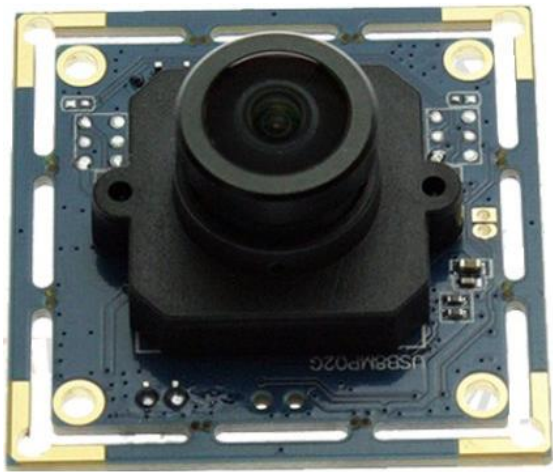


图 2-11 高清工业摄像头模组

表 2-1 RER-USB8MP02G 产品性能参数表

产品型号	RER-USB8MP02G
Sensor	索尼 IMX179
Lens Size	1/3.2inch
像元尺寸	1.4um
图像区域	6.18mm x 5.85mm
最高有效像素	3264(H) X 2448(V) 800 万像素
输出图像格式	MJPEG / YUV2 (YUYV)
支持的分辨率及帧率	3264X2448 @ 15fps / 2592X1944@ 20fps 2048X1536 @ 20fps / 1600X1200@ 20fps 1280X960 @ 20fps / 1024X768@ 30fps 800X600 @ 30fps / 640X480@ 30fps
接口类型	USB2.0 High Speed
可调节参数	亮度 对比度 色饱和度 色调 清晰度 伽玛 增益 白平衡 逆光对比 曝光度
镜头参数	高拍仪镜头 无畸变 尺寸 1/3.2 焦距 3.6mm 光圈 F3.5 视角 75
板机尺寸	38X38mm (兼容 32X32mm)

### 2.2.2 照明系统的设计及电子天平的选择

光源作为图像最重要的一个成像条件，对图像质量的好坏具有决定性的因素，它对成像后图像的像素灰度信息有很大的影响<sup>[40]</sup>。在水稻考种系统中照明系统的主要作用是创造良好的光源背景，在图像采集时可以为相机提供其所需要的光源。水稻考种中光源的首要意图是把种子同托盘中的背景有尽可能较高的对比度，以获得高质量的种子图像。

符合水稻考种仪需求的照明系统的完美的照明光源应该是发光稳定且均匀的。光源的种类较多，不同的光源有不同的优缺点以及不同的应用场景，它们大概可以分为 LED 光源、高频荧光灯、卤素灯（光纤光源），每个都具有不同的优点和缺点。高频荧光灯发光原理与日光灯类似，区别是其灯管为工业级产品，适合大面积照明，亮度较高；卤素灯具有亮度高的特点，但是其寿命太短，需要经常更换灯泡；结合水稻考种仪的要求，本课题最终定制了一款 LED 平行面光源灯，并且选择的是背光的形式，LED 光源具有寿命长、稳定性好、功耗小等特点。定制的 LED 背光源外观如图 2-12 所示。



图 2-12 背光源外观

由于水稻种子的千粒重在 18~34 克这个范围，所以电子天平最大量程在 100 克以内完全可以满足水稻种子的重量称取需求。且千粒重这一项数据需要较高的精度，所以需要将测量的精度精确到 0.01 克。综上所述的需求，本课题选择的是南京伯尼塔科学仪器有限公司生产的梦龙 DNA 精密天平。其外观如图 2-13 所示，其最大量程为 200 克，误差在 0.002 克以内，配置 RS232 和 USB 通讯端口连接外围设备，可以满足本课题的称重需求。



图 2-13 精密天平外观

### 2.2.3 采集箱体的设计

采集箱体是水稻考种仪图像采集系统的整体框架，在其内部内嵌或者直接接入了其他硬件设备，相当于一个搭载平台。照明系统安装在采集箱体的底部与顶部，工业相机与镜头内嵌在采集箱体的顶部。在整个系统中，生成和采集到种子图像质量的优劣，将直接对后续的图像算法处理时种子的计数以及相关数据的输出产生影响。又由于封闭的采集环境可以有效避免外部光源与噪声对成像与采集时产生的干扰，因此一个好的采集箱体对于后续的处理以及整个考种仪都是非常重要的。

首先确定采集箱体底部的尺寸，由 2.2.1 小节可知，采集箱体底部的尺寸大小即为 FOV 的大小。根据在一个批次中需要测试水稻种子的最大数量，将采集箱体底部的尺寸设置为  $300\text{mm} \times 200\text{mm}$ 。这使得一个批次可以测试的种子数量最多可以达到 1000 个。其次需要确定采集箱体的高度，采集箱体的高度即为物距。根据式 2-6 以及确定的 FOV 大小，可以求得 PMAG 的大小，又因为镜头为定焦相机，焦距参数已知，所以可以求得物距也即采集箱体高度应该设置为  $280\text{mm}$  左右。结合箱体的外在美观性，以及对铝合金外壳材料本身的重量以及厚度的考虑，将采集箱体的尺寸设计为  $320\text{mm} \times 220\text{mm} \times 300\text{mm}$ 。图像采集箱体最后搭建完成的效果图如图 2-14 所示，其中①代表的是内嵌摄像头，②代表的是采集箱体整体，③代表的是背光源 LED 灯。设计完成的采集箱体外观良好，且实用性较强。

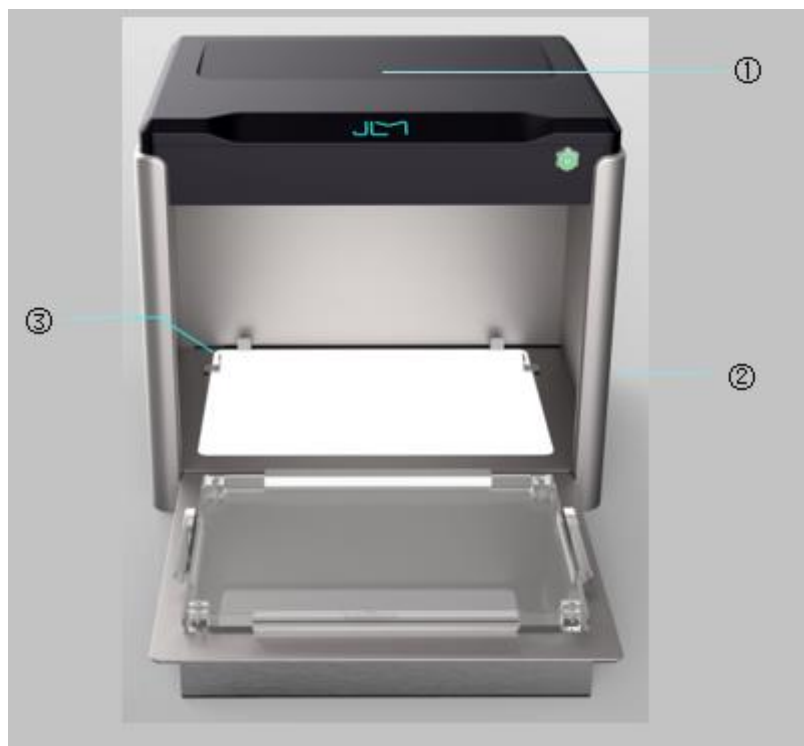


图 2-14 采集箱体整体效果图

## 2.3 软件系统总体设计

水稻考种仪系统的软件部分主要包括图像处理算法与用户图形界面设计。软件系统在 Windows 下开发，利用 OpenCV 库函数以及 Qt 开发框架，完成水稻考种仪系统的软件设计。其中调用 OpenCV 库函数可以完成图像处理中的绝大部分的功能开发。Qt 可以完成对系统的用户操作界面的设计，以及输出处理信息的展示与存储。最终，将图像处理算法与用户图形界面使用 Qt 糅合在一起，并使用 Qt 发布可供用户独立运行的可执行文件。

### 2.3.1 软件系统开发总体框架

图 2-6 展示了水稻考种仪的软件系统的总体框架设计，其中软件系统主要包括图像处理算法与用户图形界面两大部分。

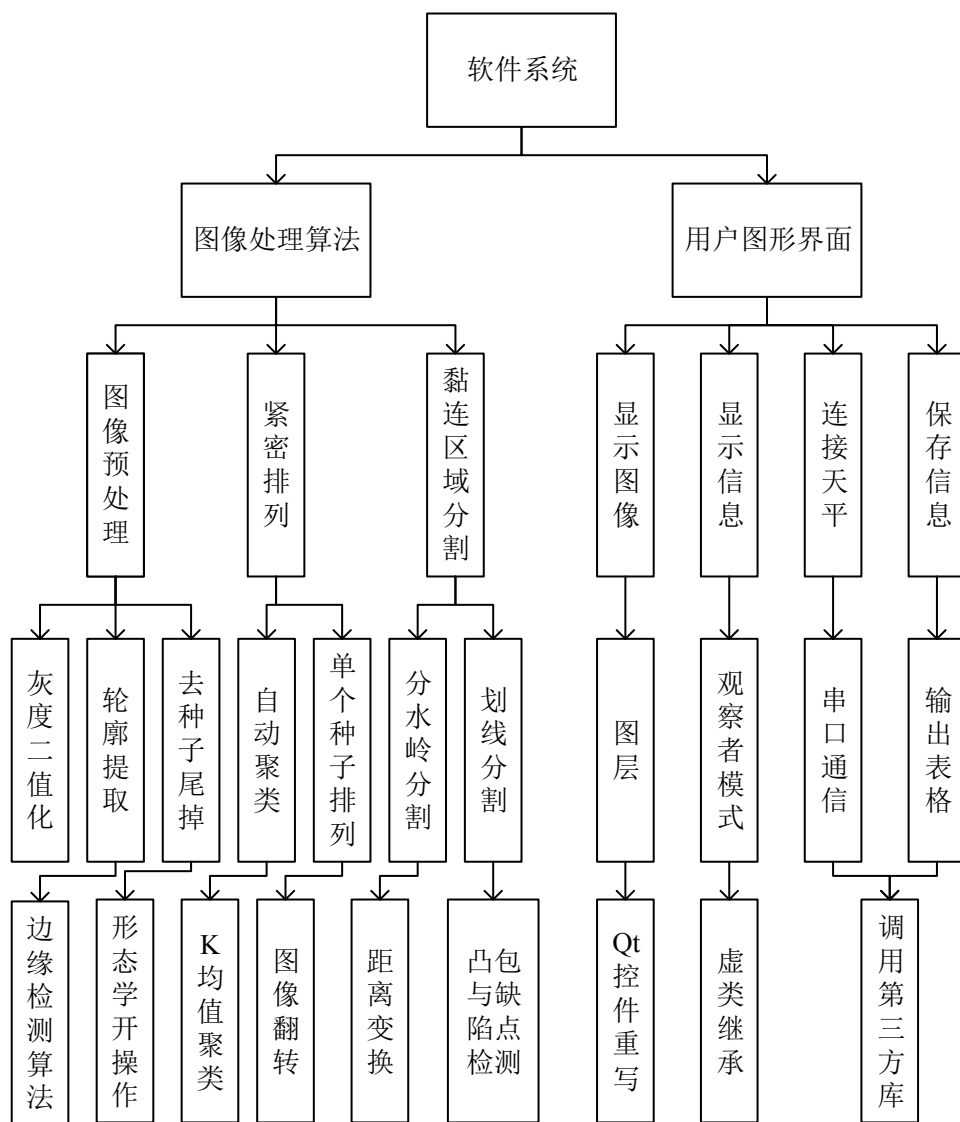


图 2-15 软件系统总体框架

图 2-15 中的图像处理算法主要包括图像预处理、图像黏连区域分割、紧密排列等部分。图像预处理部分又可以分为图像平滑、灰度二值化等操作，图像黏连区域分割又可以分为分水岭分割与划线分割两部分。用户图形界面主要包括显示图像、显示统计信息、连接天平、保存输出信息等部分。显示图像与显示图像处理输出信息涉及到了图层技术与设计模式中的观察者模式。

### 2.3.2 软件系统相关开发工具

OpenCV 早年由英特尔建立，其全称为 Open Source Computer Vision Library，中文的翻译则为开源的计算机视觉库。它具有绝大部分通用编程语言的调用接口，且适用于现在绝大部分的主流操作系统<sup>[41]</sup>。OpenCV 有着较高的计算效率，且专注于实时应用，其应用领域非常广阔，包括工业产品品质检验、图像的分割、图像

的形态学处理、医学图像处理、相机校正、双目视觉以及机器人学等方面。

Qt 是一个跨平台的 C++ 图形用户界面应用程序开发框架，Qt 十分庞大，它有自身的 GUI 组件以及内置的“一站式”解决方案<sup>[42]</sup>。Qt 提供信号与槽的机制，可以让任意两个对象之间进行消息处理，其作用就是让一个对象产生的信号能够被另一个对象接受并处理。Qt 基本所有的对象都集成在 QObject 对象中，在这个对象中有一个静态函数 connect，该函数可以让一个对象产生的信号能够被另一个对象接受并处理。

QObject::connect(&button, SIGNAL(clicked()), &w, SLOT(close())), 前面两个参数称为信号，后两个参数称为槽。经过这样的连接后，按钮 button 的 clicked 函数和窗口对象 w 的 close() 函数就进行了绑定，调用 button 的 clicked() 函数就相当于调用了窗口 w 的 close() 函数。这样做的好处是可以将两个独立的模块通过第三方连接起来，降低设计的耦合性。

在水稻考种仪系统中，将 OpenCV 配合 Qt 使用，发挥 OpenCV 在图像处理方面的高效性，Qt 在图形用户界面上开发的便捷简易性，完成系统软件部分的开发，以实现系统的设计。

## 2.4 本章小结

本章介绍了数字图像处理的相关技术，还介绍了考种系统的硬件部分与软件部分，详细说明了摄像头的选取、照明系统与采集箱体的设计。本章最后介绍了考种系统的软件部分以及软件部分的总体流程。

## 第三章 种子图像的分割与识别算法

### 3.1 种子图像分割算法

传统的种子图像识别算法都是通过迭代腐蚀来对种子的黏连区域进行分割，但是由于腐蚀需要遍历整个图像，在黏连区域很大的情况下，迭代腐蚀算法需要损耗大量时间。并且迭代腐蚀对于连通性较大的黏连区域的分割效果不理想，所以本文对传统的种子识别算法进行了改进，提出了一种距离变换、分水岭算法与凸包缺陷检测、划线分割相结合的黏连区域分割算法。

#### 3.1.1 轮廓凸包与缺陷点检测算法

凸包(Convex Hull)是一个计算几何(图形学)中的概念，在二维欧几里得空间中，凸包可以想象为一条刚好包着所有点的橡皮圈。通俗来说，给定二维平面上的点集，凸包就是将最外层的点连接起来构成的凸多边形，它能包含点集中所有的点<sup>[45]</sup>。Jarvis 步进法是寻找凸包的一种较为常用的方法，这种方法首先找到横坐标最小的点，若横坐标最小的点存在多个，则取其中纵坐标最小的那个点，然后将这点作为起点，通过求取叉积并对所有的结果进行对比，找出最靠近外侧的点，对区域内所有的点都进行循环调用，当循环回到起点时，完成凸包的寻找。

对于一个非凸包的轮廓而言，它包括一些列的凹陷区域，这些区域称作缺陷。根据每一个需要识别物体的特定的形态，对于其缺陷点的检测可以有效的帮助后续的分割任务。

OpenCV 提供了 `convexHull()` 函数来查找物体中图像的凸包，同时 OpenCV 也提供了 `convexityDefects()` 来检测图像中的非凸包缺陷。图 3-1 展示了凸包与缺陷检测的效果图，其中，左图为原图，右图为凸包与缺陷检测的效果图。红线展示的是根据原图找到的凸包，蓝点表示的是缺陷的开始点和结束点，红点表示的是缺陷的最远点，绿线表示的是一个缺陷中，开始点与结束点与最远点的连线。



图 3-1 凸包与缺陷检测

### 3.1.2 二值图像的距离变换

距离变换是二值图像处理与操作中的常用算法，在骨架提取、图像窄化中应用较多。距离变换的结果是得到一张与输入图像类似的灰度图像，但是灰度值只出现在前景区域，这使得靠近的背景的像素具有较小的值，而离背景越远的像素所具有的像素值越大。

根据度量距离的方法不同，距离变换有几种不同的方法。假设有两个像素点  $P1(x1, y1)$ 、 $P2(x2, y2)$ 。则计算距离的方法大概可以归纳为以下三种。

欧几里得距离（Euclidean distance）如式 3-1 所示。欧氏距离也即通常情况下我们计算的距离，适用性较广。

$$d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \quad (3-1)$$

曼哈顿距离（City Block Distance）如式 3-2 所示。它表示的是两个点在标准坐标系上的绝对轴距总和。

$$d = |x1 - x2| + |y1 - y2| \quad (3-2)$$

象棋格距离（Chessboard Distance）如式 3-3 所示。象棋格距离取的是两个点在水平和垂直方向上距离的最大值。

$$d = \max(|x1 - x2|, |y1 - y2|) \quad (3-3)$$

## 3.2 种子黏连区域分割算法实现

### 3.2.1 算法总体设计

算法的总体流程图如图 3-2 所示。首先对图像进行预处理，预处理包括灰度二值化和形态学的开操作以及所有轮廓的查找等；然后对预处理输出的图像进行分割识别，分割识别包括对非黏连种子区域以及黏连种子区域的识别，识别中，对面积不符要求的轮廓区域进行过滤。若面积符合则判定为单个种子；若面积过小，则判定为干扰物；若面积过大，则判定为黏连区域，放置后续黏连区域处理分割。识别要直到图像中的种子完全识别成功才能结束算法。单若有对比种子相关粒长、粒宽信息的需求，还可对识别完的种子进行紧密排列。



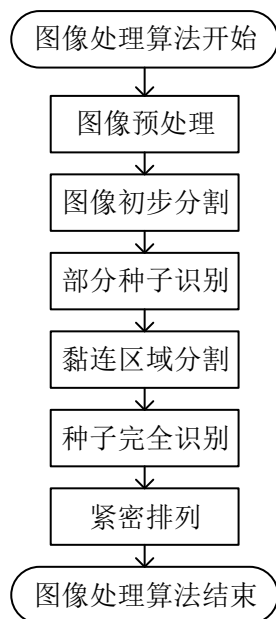


图 3-2 算法总体流程图

### 3.2.2 图像预处理

图像预处理的流程图如图 3-3 所示。

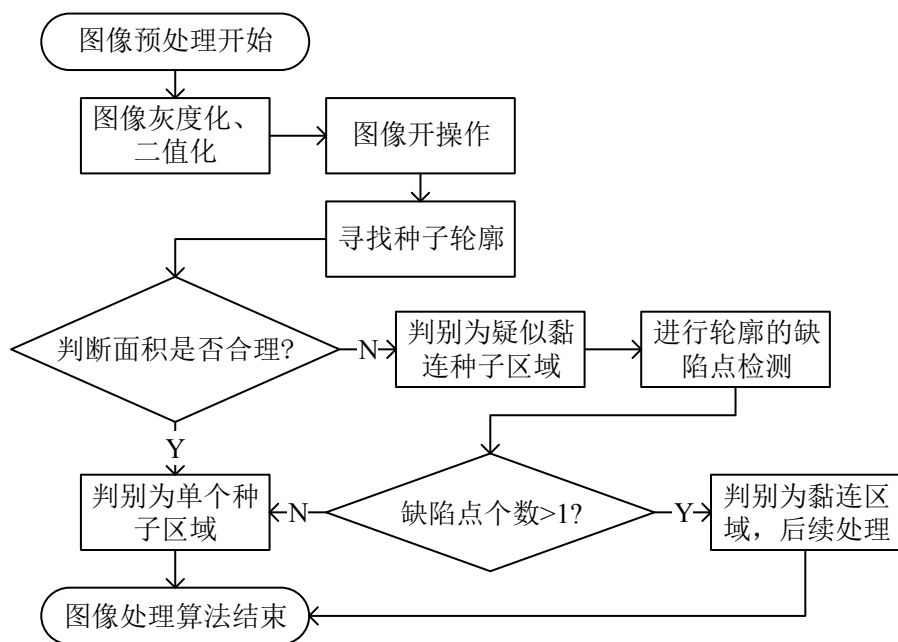


图 3-3 图像预处理流程图

图像预处理的第一步是对图像进行灰度二值化处理。首先将采集到的原图像由单通道转换为单通道的灰度图像，使用 OpenCV 的 `cvtColor()` 函数，然后将灰度图像转换为二值化图像，使用 OpenCV 的 `threshold()` 函数。原图二值化后的对比图如

图 3-4 所示。



图 3-4 二值化图像

图像预处理的第二步是对图像进行开操作。对图像进行开操作的目的是去除种子的尾掉部分，以避免其对最终统计数据的影响。使用 OpenCV 的 `morphologyEx()` 函数可以实现开操作。去尾掉的效果如图 3-5 所示，其中左边的是未采用开操作的原图，右边的是使用开操作去掉了尾掉后的效果图。



图 3-5 去尾掉效果图

预处理的第三步是对图像进行轮廓查找，然后将轮廓区域面积在合理范围内的识别为单个种子区域，面积过大的区域识别为黏连种子区域。再然后对黏连区域进行凸点缺陷检测，当凸点缺陷不存在时，认定其为单个种子区域，其余的黏连区域留给黏连区域的分割处理。图 3-6 展示了两轮判断后仍然黏连的区域。



图 3-6 两次判断后仍然黏连的区域

图像预处理的相关伪代码如下方的伪代码 1 所示。其中 `suspect_contours` 用于存储预处理后认定为黏连的区域，以二维点集的方式返回。`get_contours` 函数为获取二值化图像中的轮廓信息。对种子二值化图像中的轮廓进行遍历，然后在 `get_contour_type` 函数中对每一个点集都进行面积的判别，若判别为单个种子，则返回枚举类型的 `SINGLE` 值且调用 `set_seed` 函数将单个种子存储起来，若判别面

积过小或属于边界的干扰信息，则返回枚举类型的 GARBAGE，否则返回 ADHESION 且将黏连的区域存储起来。最后返回 suspect\_contours 以便后续处理。

伪代码 1:

```
{ vector<vector<Point>> suspect_contours;
  vector<vector<Point>> contours = get_contours();
  for (i<contours 的大小){
    Contour_Type contour_type = get_contour_type(contours[i]);
    if (contour_type == GARBAGE)//杂质干扰
      continue;
    else if (contour_type == ADHESION)//面积过大视为可疑黏连区域
      suspect_contours.push_back(contours[i]);
    else if (contour_type == SINGLE)//单个种子区域
      set_seed(contours[i], ++seed_num, SeedInfo::SINGLE_SEED);
  }
  return suspect_contours;
}
```

### 3.2.3 种子黏连区域的分割

黏连区域的分割算法总流程图如图 3-7 所示。首先结合距离变换和分水岭算法进行分割，此步骤可以分割绝大部分的黏连区域，且剩下无法分割的黏连区域都为两个种子的黏连区域，然后再根据种子的形态特征进行划线分割，完成黏连区域的分割。

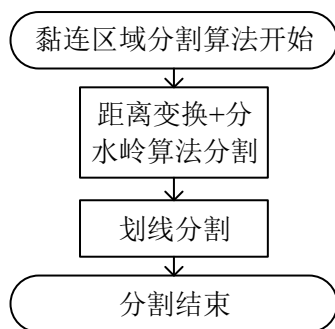


图 3-7 黏连分割算法总流程图

#### 3.2.3.1 水岭算法与距离变换相结合的分割

距离变换与分水岭算法相结合的第一次分割算法的流程图如图 3-8 所示。

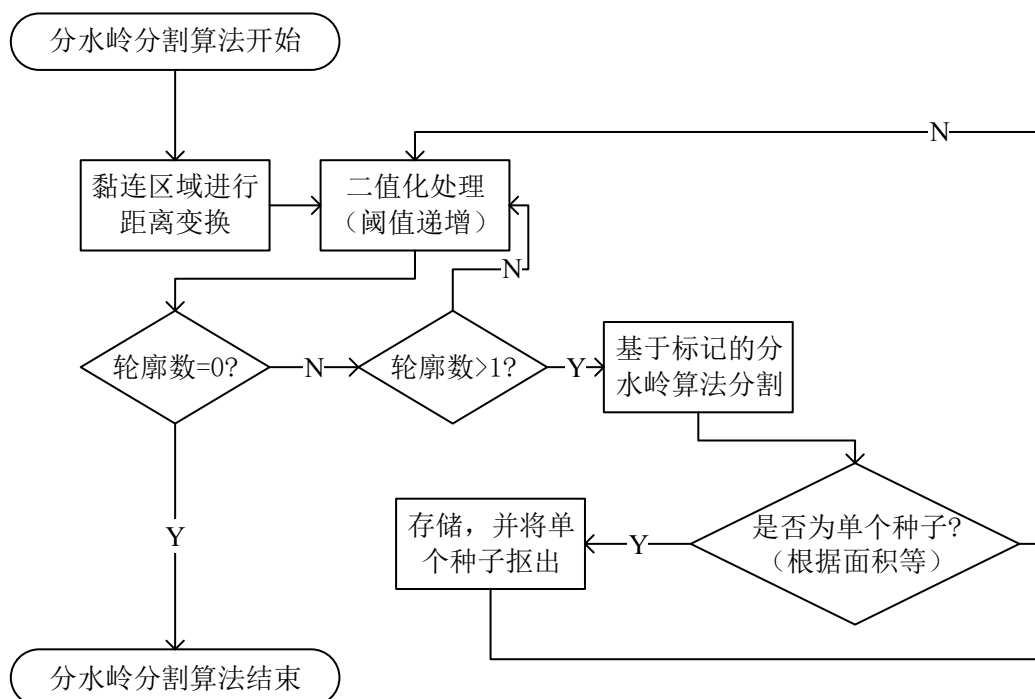


图 3-8 第一次分割的算法流程图

首先对黏连区域进行距离变换，然后对距离变换结果进行二值化处理，初次二值化阈值为初始默认值，在本算法中，设置为 1。图 3-9 展示了 a-h 阈值分别为 1-8 的情况下的二值化图像，可以看到阈值为 1、2 的时候，二值化图像仍然黏连，直到阈值为 3 的时候，才分离开最左边的种子。然后阈值一直到 8 的时候中间和最右边的种子才分离开，完成了整个的分割。

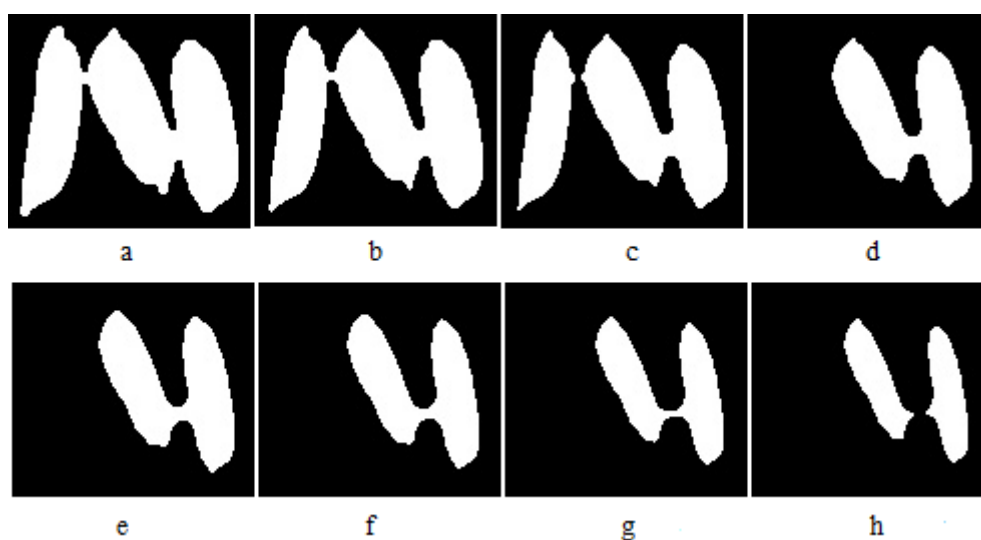


图 3-9 距离变换的二值化处理过程

在图 3-9 的 c 中，由于已经分割出来一个种子，所以将这两个轮廓作为分水岭算法的标记点，从而进行分水岭分割，或者说使用分水岭算法将距离变化分割开

的单个种子进行复原操作。图 3-10 展示了阈值为 3 的二值化后对原图进行分水岭算法的结果，其中，左图为分割成功的单个种子的图像，右图为仍黏连种子的图像。



图 3-10 第一次分水岭分割的结果

在图 3-9 的 h 中，中间与最右边的黏连种子分割开来，所以又将这两个轮廓作为分水岭算法的标记点，再进行分水岭分割。图 3-11 展示了阈值为 8 时进行分水岭分割的结果。



图 3-11 第二次分水岭分割的结果

距离变换与分水岭算法相结合分割的关键部分伪代码如下的伪代码 2 所示。其中 MAX\_DISTANCE\_THRESHOLD 为二值化阈值的上限，设置上限也可控制循环次数。seed\_contours\_beforedistance\_transform 函数是距离变换的实现。若距离变换后的点集大小为 0，则表示距离变换以及完全“腐蚀”掉了黏连区域，直接跳出循环，若距离变换结果中点集的大小与前一次距离变换结果点集的大小相同，则表示此次距离变换与二值化阈值未能分割成功，阈值递增，进行下一次循环。否则调用函数 get\_watershed\_mats，进行分水岭算法的分割，分水岭分割返回的结果为 Mat 类型的数组，存储的为分割成功的种子的图像。后续对分水岭算法分割成功的种子图像的识别由于篇幅有限，此处不再贴出。

伪代码 2:

```
for(i < MAX_DISTANCE_THRESHOLD){
    vector<vector<Point>> contours = seed_contours_beforedistance_transform();
    if(距离变换后检测到轮廓大小为 0)
        break;
```

```

else if(测得轮廓数与前一次二值化阈值的距离变换测得轮廓数相同)
    continue;
else(则表明分割成功)
    vector<Mat> seed_mats_after_water_shed = get_watershed_mats();
}

```

### 3.2.3.2 划线的分割

划线分割主要是根据凸缺陷检测来实现的，其分割的流程图如图 3-12 所示。

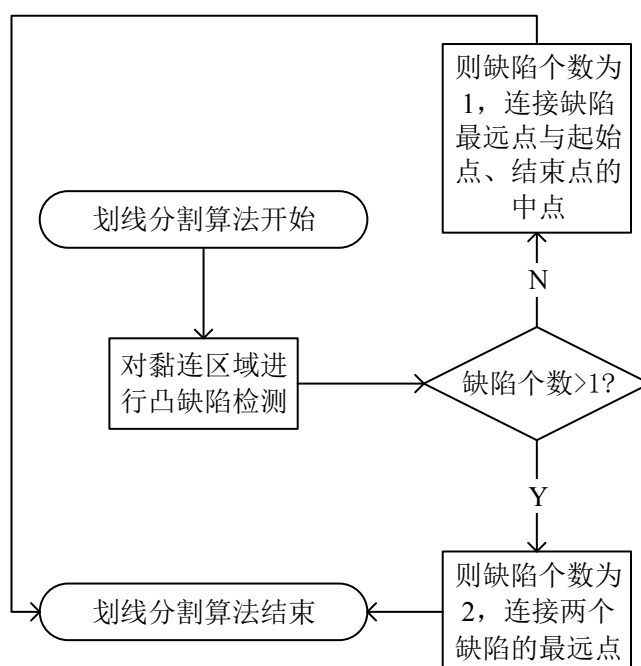


图 3-12 划线分割算法流程图

首先对黏连种子进行凸缺陷检测，然后判断缺陷点的个数，通常缺陷点的个数都为 1 或 2。对缺陷点个数为 1 的黏连区域直接划线在缺陷起始点与结束点的中间位置与缺陷最远点的线段上。如图 3-13 所示，为缺陷点数为 1 时的分割效果图。对缺陷点数为 2 的黏连区域直接划线在两个缺陷的最远点的连接线段处。如图 3-14 所示，为缺陷点数为 2 时的分割效果图。

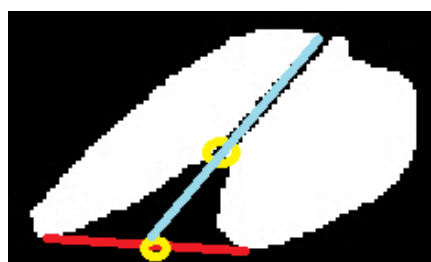


图 3-13 凸缺陷个数为 1 时的划线分割

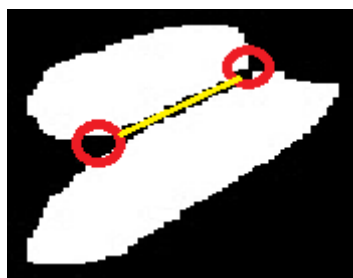


图 3-14 凸缺陷个数为 2 时的划线分割

划线分割算法的关键部分伪代码如下面所贴出的伪代码 3 所示。其中 `watershed_suspect_contours` 为距离变换与分水岭算法结合分割失败的黏连种子区域，对这些种子区域遍历进行划线分割。首先调用 `get_part_mat` 函数对黏连区域进行抠取。之后调用 `get_defects_param` 函数获取相应区域的凸缺陷的起始点、结束点、最远点等参数。然后判断其中任意一个点集的大小，图中代码所示的是对点集大小为 2 的情况时的处理。将点集大小为 2 的黏连区域调用 `seed_draw_line` 函数进行划线分割，其效果为将两个凸缺陷的最远点划上一条灰度值为 0，宽度为 3 的线，使黏连区域分离开来，达到分割的效果。后续对其它情况的处理由于篇幅有限，不再贴出。

伪代码 3:

```
for(i < watershed_suspect_contours 的大小){
    Mat part_src_mat = get_part_mat();
    get_defects_param();
    if(缺陷点集大小为 2)
        seed_draw_line();
    else if(缺陷点集大小不为 2)
        //连接最远点与起点与终点的中点
}
```

### 3.3 种子紧密排列图像处理算法

水稻种子的紧密排列指的是将识别到的单个种子以横排或者竖排的形式依次排好，这样做的目的是为了水稻相关的研究人员对不同品种的水稻种子进行表面形状的分析对比。

本文中的紧密排列主要流程图如图 3-15 所示。首先对种子进行识别，得到相应的单个种子的轮廓信息；然后根据轮廓信息中的单个种子的中心坐标点进行聚类操作，即将摆放在不同位置、不同堆的种子自动识别为不同品种的种子；最后将种子都从原图中抠出来，按照聚类的结果排成不同的行或列。

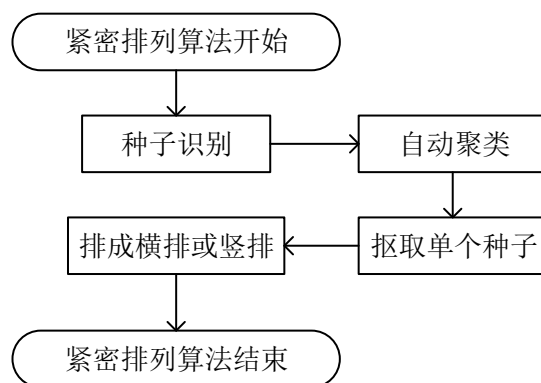


图 3-15 紧密排列算法流程图

### 3.3.1 自动聚类

首先介绍一下类与聚类的概念。类指的是具有相似性的集合，而聚类就是指将数据集划分为若干类，各类之间数据相似性差别尽可能的大。在水稻种子紧密排列中使用了 K-means (K 均值) 算法进行迭代型聚类，采用距离作为相似性指标，从而发现给定数据集中的 K 个类，且每个类的中心是根据类中所有值的均值得到，每个类用聚类中心来描述。聚类目标是使得各类的聚类平方和最小，在种子的聚类中，需最小化其欧式距离，即最小化式 3-4。

$$J = \sum_{k=1}^K \sum_{i=1}^N \|p_k - m_{ki}\|^2 \quad (3-4)$$

K-means 聚类算法是一个反复迭代的过程。首先需要选取数据空间中的 K 个对象作为初始中心，每个对象代表一个聚类中的质心。对于样本中的数据对象，根据它们与这些质心的欧氏距离，按距离最近的准则将它们分到距离它们最近的聚类中心所对应的类。然后是更新聚类中心，将每个类别中所有对象所对应的均值作为该类别的聚类中心，再与之前的聚类质心对比，若相同则表示聚类结束，若不同则迭代进行聚类<sup>[46]</sup>。在紧密排列中，聚类的流程图如下图 3-16 所示。图 3-17 展示了聚类的效果图，其中左侧为原图，右侧为聚类的效果图，其中每一类都使用了不同颜色的轮廓线标出。



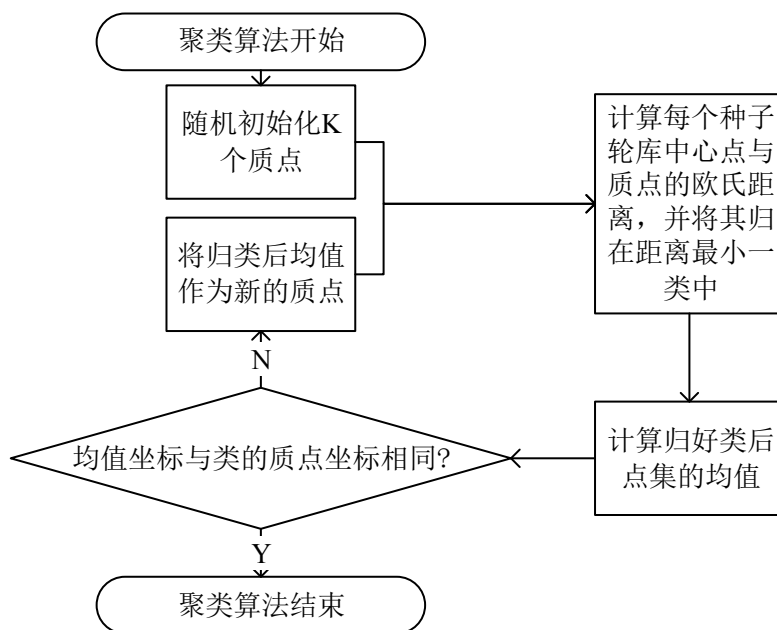


图 3-16 自动聚类算法流程图



图 3-17 聚类效果图

自动聚类的部分关键伪代码如伪代码 4 所示。代码中展示的是聚类的迭代部分，首先定义一个 `minDist` 变量，存储计算点与质心的欧氏距离最小值，将其初始化为 `0x7fffffff` 为 `int` 型变量的最大值。定义一个 `minIndex` 变量，存储计算点与质心的欧氏距离最小值的质心的索引。然后算出计算点与 `K` 个质心的欧氏距离，并通过对比，存储最小值与其对应的索引值，最后将计算点的类别归为最小值的索引。若此时的计算点的类别改变了，则表示还需要进行迭代，将标志位置为真，若所有计算点的类别都未改变，则表示聚类成功，停止迭代。

伪代码 4:

```
while(质心发生了改变){
```

```

for(i<点集的大小){
    float minDist = 0x7fffffff;
    int minIndex = -10;
    for(j < k 值){
        float distance = 欧氏距离(i,j);
        if(distance < minDist)
            更新 minDist 并存储最小值的索引;
    }
    if(类别发生了改变)
        flag = true;
}
if(flag == false)
    Break;
}

```

### 3.3.2 单个种子的抠取与排列

单个种子的抠取主要指将识别到的单个种子从原图像中抠取下来，作为一个单独的 Mat 对象，并将这个 Mat 对象进行灰度与二值化等操作，使其背景透明化，使种子为暗色。种子的排列是指：将抠取且透明化后的种子进行旋转，使种子竖直的方向与水平面垂直，最后将单个种子按照分类情况进行排列。种子的抠取与排列的流程图如图 3-18 所示。

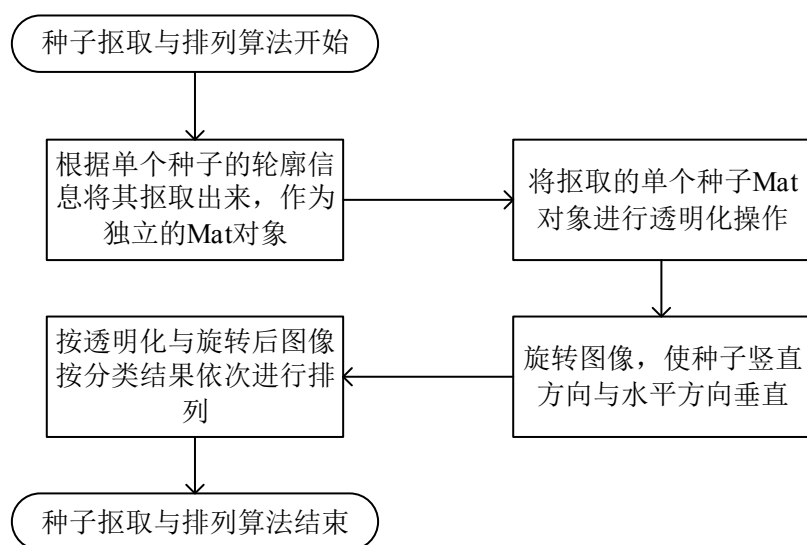


图 3-18 种子的抠取与排列算法流程图

种子的抠取主要利用的是识别到的单个种子的轮廓信息，然后复制轮廓点集所处原图中坐标的像素值到一个新的 Mat 对象，即可完成单个种子的抠取。图 3-19 展示了单个种子抠取的效果图。背景的透明化主要是将种子进行背景统一为白色，种子则显示为暗色，透明化效果如图 3-20 所示。



图 3-19 单个种子抠取效果图



图 3-20 透明化效果图

种子的排列首先需要将抠取与透明化后的种子进行旋转，即使种子的竖直方向与水平面垂直。先根据种子的轮廓点集绘制其旋转矩形框，然后由旋转矩形框可以获得种子的偏移角度，再由此偏移角度对 Mat 矩阵进行旋转，则得到旋转后的图像。旋转后的效果图如图 3-21 所示。最后按照类别将旋转后的图像进行拼接即可得到最终的紧密排列结果，其效果图如图 3-22 所示，其排列的对象是图 3-17 中的原图，其中红色的数字表示的是那一列所有种子的长度。



图 3-21 旋转效果图

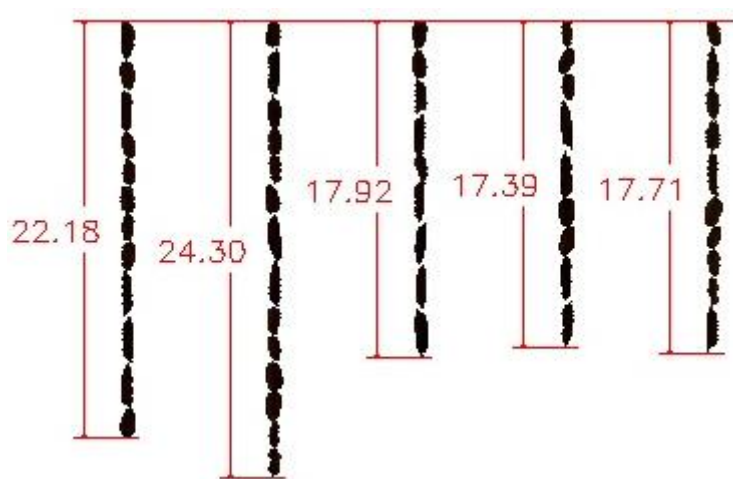


图 3-22 紧密排列最终结果

### 3.4 本章小结

本章首先介绍了到种子图像处理识别所涉及的一些算法，然后分析了传统算法存在的不足之处，提出了改进的算法。然后在此基础上，说明了考种系统图像处理算法的总体流程图，包括预处理和黏连区域分割两大部分，并分别对这两大部分都进行了详尽的描述，给出了分块处理的流程图与一些测试的图例。

## 第四章 系统核心功能详细设计

系统的核心功能主要涉及有图像的采集与显示、天平的连接与通信、种子统计信息的增删查改与输出打印等。其中图像的采集与显示、种子统计信息的增删查改与输出打印是其中设计的重点与难点。图 4-1 展示了系统核心功能的总体设计。

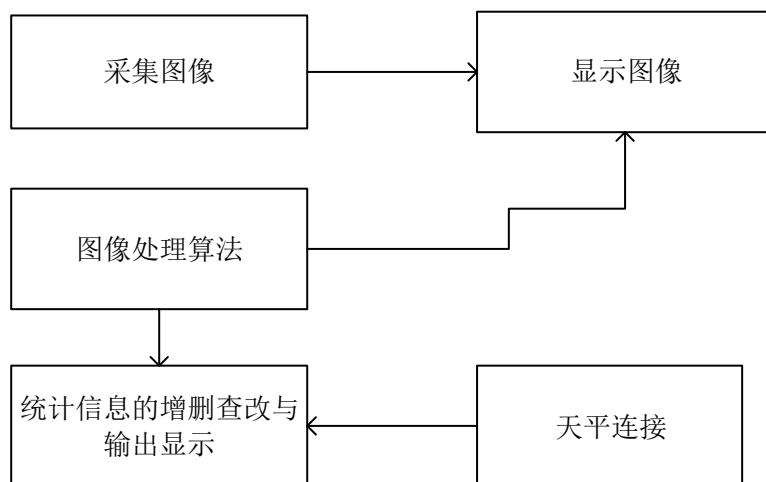


图 4-1 系统核心功能总体设计

### 4.1 图像的采集与显示

图像的采集与显示主要用到了 Qt 中的一系列控件信号与槽机制。在用户界面上放置打开摄像头以及采集图像的按钮，点击时发出信号，对应到相应的槽函数执行相应的操作。

#### 4.1.1 实时图像的显示与采集

使用 Qt 中的 `QCameraViewfinder` 控件对摄像头的实时图像进行显示，使用 Qt 中的 `QLabel` 控件对采集的图像进行实时显示。其中 `QCameraViewfinder` 与 `QLabel` 放置在界面上的同一区域，所以使用 `QscrollArea` 实现对这两个控件的管理。将 `QCameraViewfinder`、`QLabel` 根据不同的状态装载入或者移出 `QscrollArea` 控件中，例如在图像采集状态时，则利用 `QscrollArea` 的 `takeWidget` 函数将其中现有的控件拿出，再使用 `setWidget` 函数将控件 `QcameraViewfinder` 装入 `QscrollArea` 中。将 `QscrollArea` 控件视为一个容量大小为 1 控件容器，则它的装载与移除如图 4-2 所示，上图展示了摄像头显示状态下对 `QscrollArea` 容器的操作，下图展示了采集图像显示状态下对 `QscrollArea` 容器的操作。

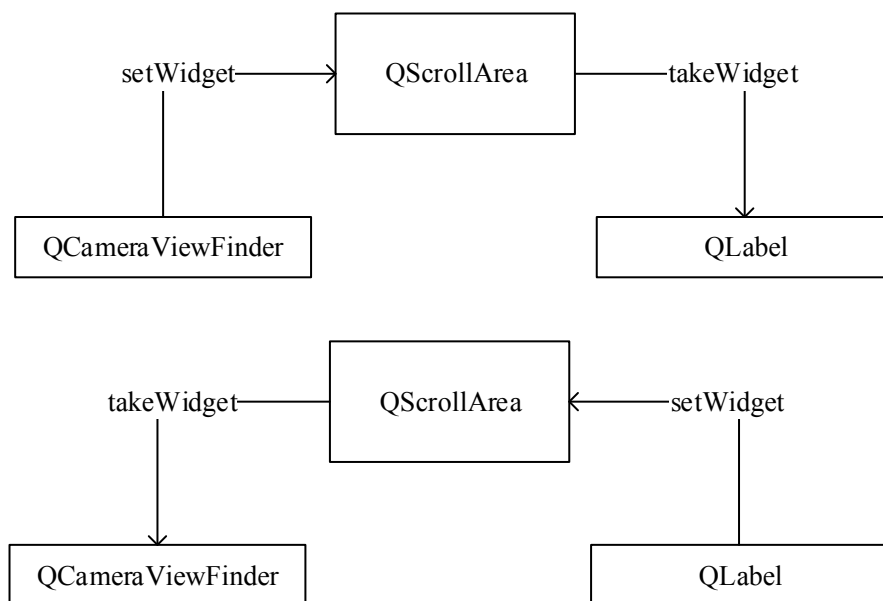


图 4-2 不同状态的控件容器管理

由于 QCameraViewfinder 封装的良好性，仅需传入摄像头的设备号即可对摄像头进行打开、关闭以及图像采集的操作，十分便捷。根据查阅相应的官方文档，摄像头的设备号为“\VID\_05A3&PID\_8830&MI\_00\7&27d2624d&0&0000”。采集图像时也只需调用 Capture 函数进行拍取。根据不同的分辨率需求，还可调用 QCameraViewfinder 成员函数，对摄像头的分辨率进行相应的修改。使用 Qt 的 QcameraViewfinder 控件对摄像头的相关操作流程如图 4-3 所示，首先输入摄像头的设备号以实例化 QcameraViewfinder 类，然后载入对应设备号的摄像头到 QcameraViewfinder 中，若需要对摄像头的相关参数进行特定的设置，则需要调用 setViewfinderSettings 函数对相关参数进行设定。设定完成或者使用选择使用默认的参数后就可以调用 start 函数，将摄像头的实时画面显示到 QcameraViewfinder 控件之中。当需要捕捉当前的实时画面时，使用 capture 功能即可捕捉到当前的实时画面。最后 close 功能可对打开的摄像头进行关闭操作。

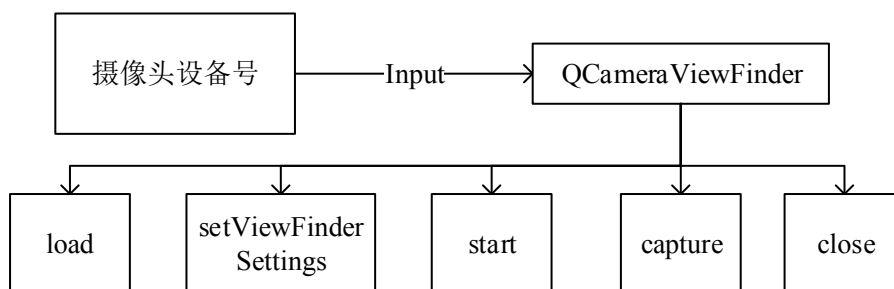


图 4-3 对摄像头的相关操作

为了方便查看采集到种子图像的细节，对 QLabel 进行了重写，增加了图像的放大和缩小功能；为了方便后续在图像上进行增删查改操作，对 QLabel 进行了重写，增加了右键操作，包括删除误识别的种子和新增种子的功能。

### 4.1.2 采集图像的显示与相关操作

对 QLabel 的重写主要涉及到的是对其鼠标事件、滑轮事件以及绘图事件的重写。图 4-4 展示了空间类的继承关系与类内的成员变量、接口函数等。其中 QLabel 为 Qt 自带的控件类，作为后续继承的基类。CTZoomLabel 类继承 QLabel 类，主要实现的是图像的放大缩小功能，即以鼠标所在处为焦点进行放缩。CTPaintLabel 类继承 CTZoomLabel 类，实现了在控件范围内的右键操作，为后续的增删种子提供基础。

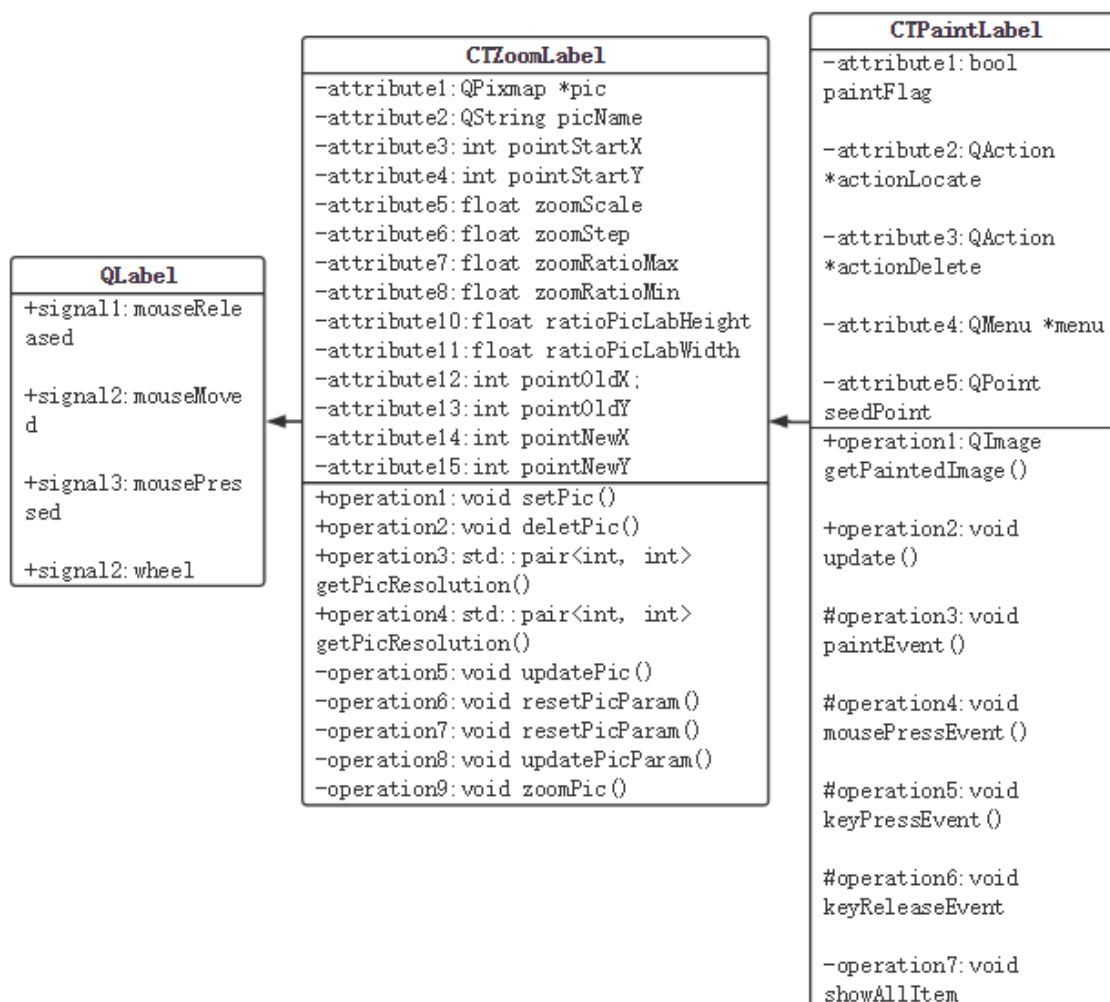


图 4-4 图像显示控件类图

其中 CTZoomLabel 类实现了对图像的放大缩小操作，其主要的实现原理如图

4-5 所示。首先对鼠标滑轮进行监视，一旦鼠标滑轮发生了改变，则通过 `wheelEvent` 获取滑轮的滑动情况，即滑动的方向代表放大或者缩小，滑动的范围代表放大缩小系数。然后调用 `zoomPic` 函数，根据鼠标当前的坐标位置，以及放大缩小的系数计算放大后图像的坐标位置。最后调用 `update` 函数对图像进行缩放，且缩放后的焦点保持与缩放前位置的一致。

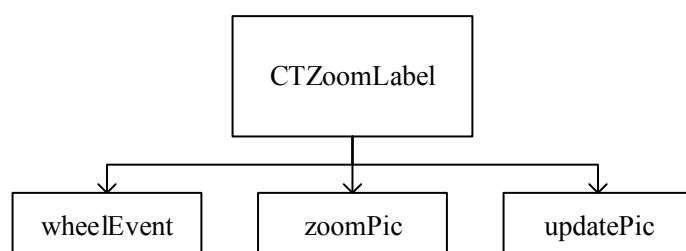


图 4-5 CTZoomLabel 放大缩小

CTPaintLabel 实现在 QLabel 图像显示区域的右键操作，其主要的实现原理如图 4-6 所示。首先对鼠标的右键进行监视，当鼠标的右键按下时，`mousePressEvent` 则对此时刻鼠标所在坐标进行判断，即判断此时的坐标是否在识别到的种子矩形框区域内。此时的坐标点存在种子的矩形区域内则下一步调用 `update` 函数，`update` 函数是继承于 QLabel 的重载函数，一旦调用 `update`，则会调用 `paintEvent` 对整个显示区域进行重新载入。在 `paintEvent` 中调用 `showAllItem` 将右键操作的选项显示在图像显示区域中。

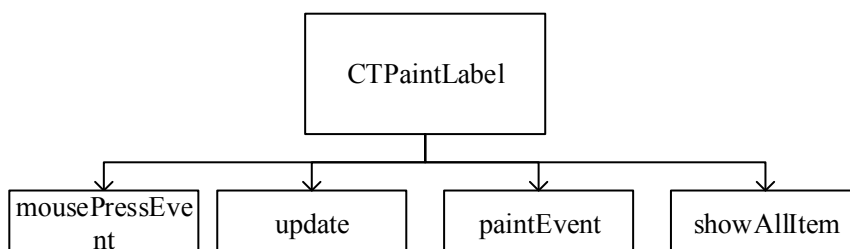


图 4-6 CTPaintLabel 右键操作

## 4.2 统计信息的计算与生成

### 4.2.1 单个种子统计信息的计算与生成

采集到的待识别种子图像输入到种子识别算法当中，算法输出识别到种子的轮廓信息。根据输出的单个种子的轮廓数据，分别计算单个种子的相关统计信息，包括种子的粒长、粒宽、周长、面积等，相关流程如图 4-7 所示。



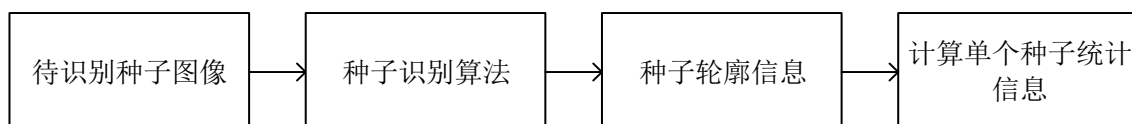


图 4-7 单个种子统计信息生成流程图

在已知种子轮廓信息的情况下，要想获取种子的粒长、粒宽，只需根据轮廓信息获取种子的最小外接矩形框，然后由矩形框的长、宽来得到粒长、粒宽信息。种子最小外接矩形框可利用 OpenCV 的 `minAreaRect` 函数获取。种子的周长信息可以使用 OpenCV 的 `arcLength` 函数计算得出，此函数只需待计算区域的轮廓点集，即可得到轮廓的周长信息。种子的面积信息可使用 OpenCV 的 `contourArea` 函数计算得出，`contourArea` 函数主要是运用格林公式来计算轮廓区域的面积。图 4-8 展示了单个种子统计信息的计算方式。

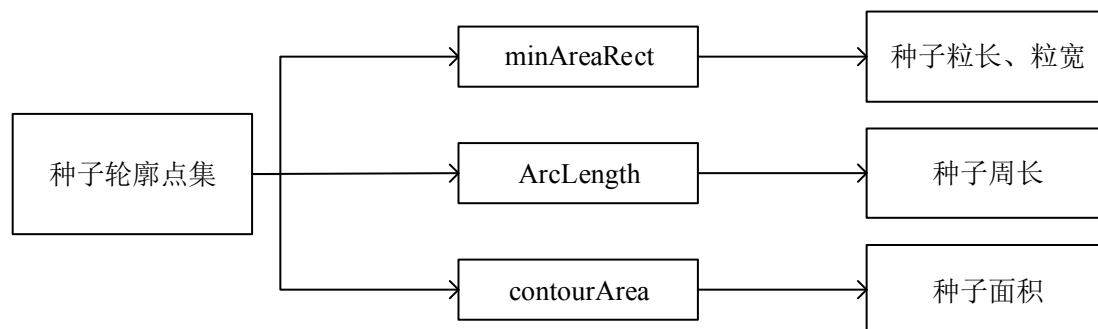


图 4-8 单个种子统计信息计算方式

## 4.2.2 总体统计信息的计算与生成

总体统计信息主要包括：样本计数、粒长（粒宽）均值、粒长（粒宽）标准差、周长（面积）均值、周长（面积）标准差以及千粒重等。

其中，均值即为所有识别到单个种子的相关指标的平均值，可以表示一组数据集中趋势的量数，其可由式 4-1 计算求得，其中  $n$  代表样本计数值。通过对种子统计信息的均值数据计算，可以分析得到一批次种子的大致外观形态。对比不同批次种子的均值数据可以更为准确的反映品种间的形态差异性。

$$\bar{x} = \frac{x_1 + x_2 + x_3 \dots + x_n}{n} \quad (4-1)$$

标准差就是我们数学上所说的均方差，它可以反映一个数据集的离散程度，其可由式 4-2 计算求得，其中  $N$  表示样本计数值。通过对种子统计信息的标准差数据计算，可以分析得到一批次种子形态的离散程度，进而成为分析种子生长稳定性的指标之一。

$$d = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (4-2)$$

千粒重表示的是一千粒种子的重量，它是体现种子大小与饱满程度的一项指标。是检验种子质量和作物考种的内容，也是田间预测产量时的主要依据。千粒重可由式 4-3 计算得到。

$$P = \frac{W}{N} \times 1000 \quad (4-3)$$

### 4.3 统计信息的增删查改与输出显示

种子的统计信息来自图像处理算法的输出，考虑到算法的容错空间与操作的可行性，对算法输出的统计信息添加了增删查改的功能，由于统计信息与种子图像中同时具有算法的输出信息，在任何一面进行增删查改操作时，另一面都需要同时更新，联动显示。所以此部分主要介绍图像显示的图层部分以及统计信息与图像显示的联动设计。

#### 4.3.1 图层技术

图层是 Photoshop 中的一个图片处理核心版块，传统意义上来说，图层就像是含有文字或图形等元素的胶片，一张张按顺序叠放在一起，组合起来形成页面的最终效果。图层中可以加入文本、图片、表格、插件，也可以在里面再嵌套图层。打个比方说，在一张张透明的塑料纸上作画，透过上面的塑料纸可以看见下面纸上的内容，但是无论在上一层上如何涂画都不会影响到下面的塑料纸，上面一层会遮挡住下面的图像。最后将塑料纸叠加起来，通过移动各层玻璃纸的相对位置或者添加更多的玻璃纸即可改变最后的合成效果<sup>[47]</sup>。图 4-9 是图层概念的示意图，最右侧将人脸、眼睛与嘴巴分为了三个部分，分别分布在三个塑料纸中，然后将三个塑料纸从上往下依次叠放，最后形成了最左边的效果样式。

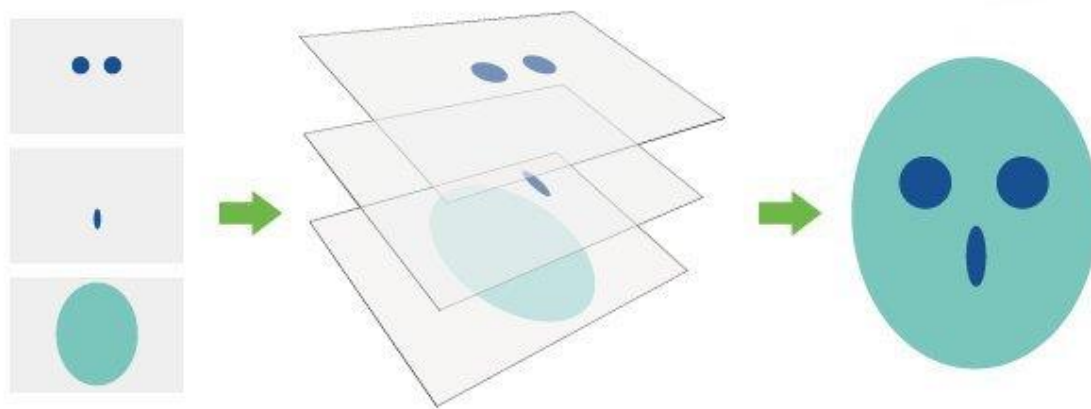


图 4-9 图层概念示意图

考种仪系统将图层的概念加入到图像显示中的种子轮廓标注中,由于增删查改操作需要改变种子的轮廓标注,若采用传统的图像显示方案,整个图像需要重新生成,而使用图层的设计则只需在轮廓标注层进行增删查改,这样的话可以大大减少图像绘制的时间,从而提高系统的效率。

Qt 的 `QPainter` 类可以实现对控件的绘画功能。首先设计一个 `CTSeedInfo` 类将算法输出的识别信息存储起来,并实现识别到种子的绘制操作,再将此类设计为单例模式,并利用 Qt 中设计的 `QundoStack` 类实现撤销与反撤销操作。然后将采集到的图像依附在 `CTPaintLabel` 中显示,最后将 `CTPaintLabel` 作为控件参数传入 `CTSeedInfo` 中的绘制种子轮廓成员函数,完成图层绘制。其中原图像为单独的一层“塑料纸”,`CTPaintLabel` 为另一层“塑料纸”以透明的方式附在原图像上方。

图 4-10 展示了考种仪软件显示中图层显示的效果图,左图表示的是通过算法传递参数进行绘制的轮廓图,右图表示的是将上方种子进行删除即除去了识别轮廓线。可以看到,这样的设计对原图不会产生任何影响,加快了图像处理的效率与显示的速度。

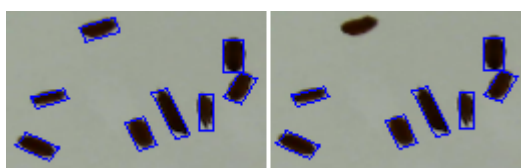


图 4-10 图层效果图

### 4.3.2 统计信息与显示图像的联动

统计信息与显示图像的联动的设计主要是利用了观察者模式。观察者模式定义了对对象间的一种一对多的依赖关系,即当一个对象的状态发生改变时,所有依赖于它的对象都将得到通知并被自动更新。观察者模式主要解决了一个对象状态改变给其它对象通知的问题,而且具有易用性和低耦合性,保证高度的协作<sup>[48]</sup>。

图 4-11 是观察者模式的示意图,下面介绍一下观察者模式的主要流程。首先需要定义两个基类,它们分别是 `Subject` (目标) 和 `Observer` (观察者)。`Subject` 中需要提供注册和删除观察者对象的接口,并提供一个 `notify` 接口用于对所有依赖于它的观察者进行同步。`Observer` 为那些在 `Object` 发生改变时需获得通知的对象定义一个更新接口。基类的定义完成之后就需要定义两个类来继承它们,这两个类分别是 `ConcreteSubject` (具体目标) 和 `ConcreteObserver` (具体观察者)。在 `ConcreteSubject` 中将有关状态存入具体观察者对象,并在具体目标发生变化时,给所有注册过的观察者发出通知,对观察者们进行相关的调度。在 `ConcreteObserver` 中将基类中的更新接口实现,以便使本身的状态与具体目标的状态相协调。

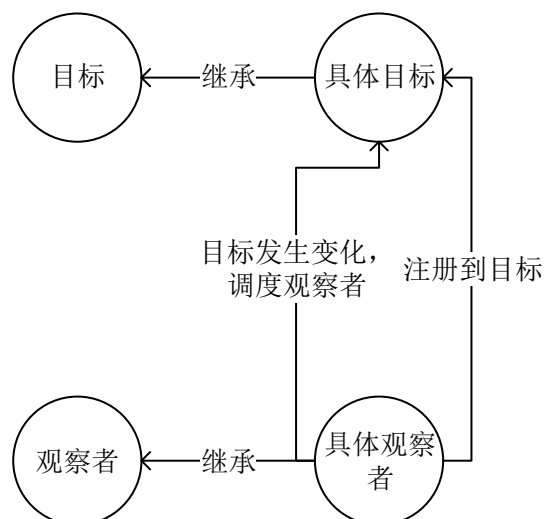


图 4-11 观察者模式示意图

在统计信息与显示图像的联动的设计中，同上述流程一致，首先定义两个基类，然后将统计信息类继承 **Subject** 基类，将统计信息显示栏、图像与轮廓显示类继承 **Observer** 基类。然后将观察者对象与目标对象绑定起来。当统计信息类中出现了增删查改时，则调用 **notify** 函数，通知统计信息显示栏类、图像与轮廓显示类进行更新操作，从而完成了两者的联动。图 4-12 展示了本系统联动设计的观察者模式搭建的类图。

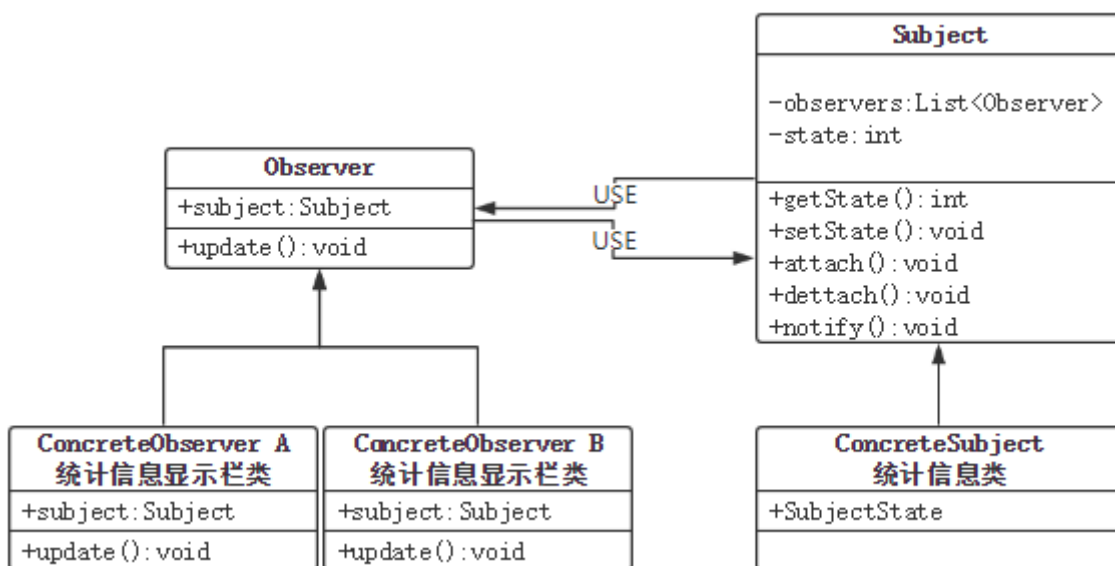


图 4-12 观察者模式搭建类图

### 4.3.3 统计信息的 Excel 表格输出

统计信息的 Excel 表格输出利用的是 QtXlsx 这个第三方库。QtXlsx 是一个可以读取和写入 Excel 文件的库，它不需要 Microsoft Excel，可以在 Qt5 支持的任何平台上使用。该库可以用来从头开始生成一个新的.xlsx 文件、从现有的.xlsx 文件中提取数据以及编辑现有的.xlsx 文件等。

统计信息保存为 Excel 表格的流程如图 4-13 所示。首先使用结构体 Document 实例化一个 Excel 文档对象，然后使用 addSheet 函数对当前的文档对象新建页。再使用 selectSheet 函数选择当前文档中需要编辑的页。选择好要编辑的页后即可使用 write 方法对相关数据进行写入，再调用 selectSheet 函数选择当前需要编辑的页，然后直接使用 write 函数即可对相应的数据进行写入。写入完成后，使用 saveAs 方法即可对文档保存到指定目录之下。

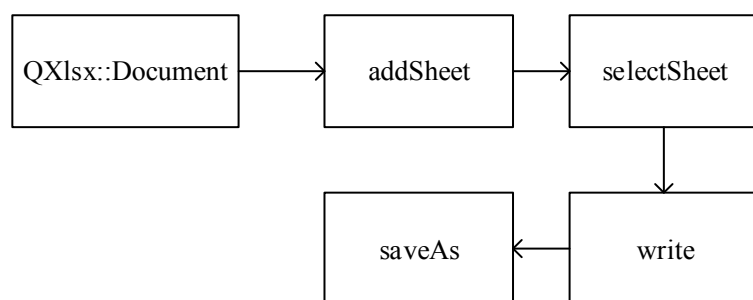


图 4-13 保存统计信息为 Excel 文件操作流程图

## 4.4 电子天平与 LED 灯的通信和控制

### 4.4.1 电子天平的通信

电子天平与 PC 的连接线为 USB 转串口线，即计算机 USB 接口到通用串口之间的转换。USB 转串口线能够给没有串口接口的计算机提供快速的通道，这使得传统的串口设备变成了即插即用的 USB 设备。

在考种算法完成后，会输出相关的考种信息，但是千粒重信息需要称得种子重量后才可以生成。所以，要想获得种子的千粒重信息还需要连接软件与天平的部分，以获取种子的实时称重信息。电子天平的连接及使用流程图如图 4-14 所示。

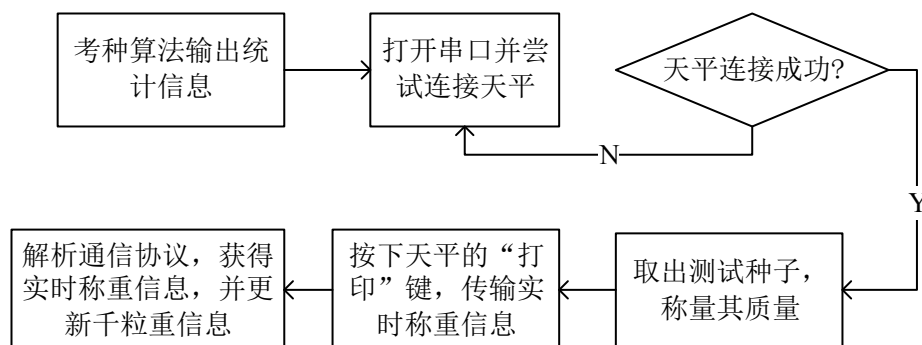


图 4-14 电子天平连接与使用流程图

为了解析电子天平传输到 PC 中的信息，根据天平的通信协议涉及正则表达式对通信信息进行匹配，从而得到种子重量这一信息。“DATE:(\\S\*)”这一表达式可对天平返回的重量部分进行解析，“TIME:\\s\*(\\S\*)”表达式可对天平返回的称重时间进行解析，“(ST|US)\\s\*:(\\+|-)\\s\*(\\d\*\\.\\d\*)”表达式可解析得到对天平称重时的稳定状态，以及所获得重量的正负号。根据解析上述的三项信息，可以首先判别天平称重的稳定状态，若在稳定的状态下，则判定此次测得的为有效信息。将种子重量信息提取出来，然后就可以计算千粒重信息。千粒重主要是利用称重重量除以统计信息中的种子数量再乘以 1000 即可。计算完千粒重后，还可根据解析的时间信息，对千粒重的测量时间进行记录。

在与天平进行通信前或者说在尝试连接天平之前，需要首先对天平的连接情况进行一次测试，即按下天平中的打印键，然后，天平会根据其通信协议回传字符串，根据上述的正则表达式进行匹配，拆解出通信信息，判断天平是否正确连接，若正确连接，即可进行天平的后续相关操作。

#### 4.4.2 LED 灯的控制

LED 灯的主要通过继电器来控制，所以对 LED 灯的控制就可以通过对继电器的控制来进行。对继电器的控制可以调用继电器 API `usb_relay_device.dll`。

`usb_relay_device.dll` 为 C++ 编写。调用其 `usb_relay_device_open` 函数返回一个有效的句柄进行对相关继电器的控制；调用其 `usb_relay_device_close` 函数可对继电器进行关闭操作；调用其 `usb_relay_device_open_one_relay_channel` 函数可打开继电器的指定端口，同理 `usb_relay_device_close_one_relay_channel` 函数可关闭继电器的指定端口。LED 灯控制示意图如图 4-15 所示，打开顶灯与关闭顶灯假设都是对继电器的端口 1 进行的操作；打开底灯与关闭底灯假设都是对继电器的端口 0 进行的操作，则只需根据相应的操作，传入相应的端口号即可。

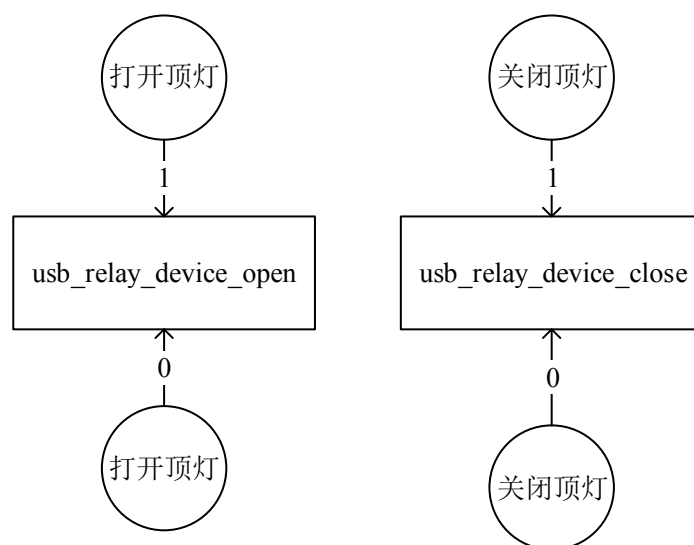


图 4-15 LED 灯控制示意图

#### 4.5 本章小结

本章主要介绍了考种系统中涉及到的一些关键的技术，包括图层显示、基于观察者模式的图像与统计信息的联动以及紧密排列等。通过这些设计，可以使得界面操作更为便捷与灵活，使用户有着更好的操作体验。

## 第五章 系统的实现

本文所设计的基于图像处理的水稻考种系统包括图像采集的硬件部分与图像处理算法和用户图形界面设计的软件部分，其中图像采集系统负责采集种子图像，其各个硬件的选型以及整体系统的搭建已在第二章进行了实现，图像处理算法在第三章进行主要的介绍与实现，用户界面中的关键技术也在第四章的部分进行了说明，本章主要说明本系统的整体实现结果与测试情况。

### 5.1 系统主要操作界面

软件系统的总体界面如图 5-1 所示。其中上方为一些操作的按钮，可以进行包括图像采集、图像处理、数据输出等操作。又主要可以分为对摄像头的操作、对图片的操作、对设备的操作、算法处理操作等几个部分。左下方为图像显示界面，可以显示摄像头的实时图像、采集到的图像以及处理后的图层显示图像，也是保存图像的主要对象。右下方为图像处理输出统计数据的显示区域，包括了总信息、单粒种子信息等。左下角的状态显示栏可以显示整个系统所处的状态，右下角的状态显示栏显示的摄像头当前的状态。

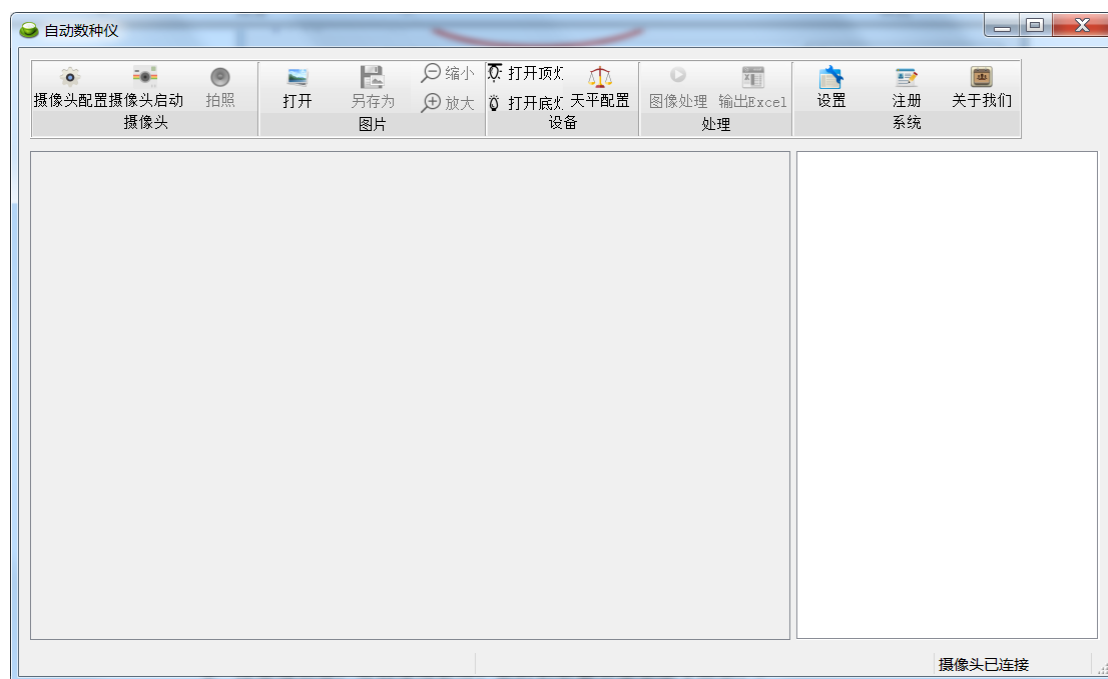


图 5-1 软件系统总体界面



## 5.2 系统主要操作流程

### 5.2.1 图像采集的操作流程

图像采集的相关按钮主要如图 5-2 所示，要包括有摄像头配置、摄像头启动、以及拍照三个操作。



图 5-2 对摄像头的操作按钮

在图像采集之前首先需要对摄像头的参数进行配置。在图 5-2 中点击“摄像头配置”，进行摄像头的相关设置，而对摄像头主要是需要对其分辨率进行设置。由于考种图像对分辨率的要求较高，且高分辨率可以获得更好的识别精度，所以将摄像头的分辨率设置为  $3264 \times 2448$ ，后点击“OK”确认。操作如图 5-3 所示。

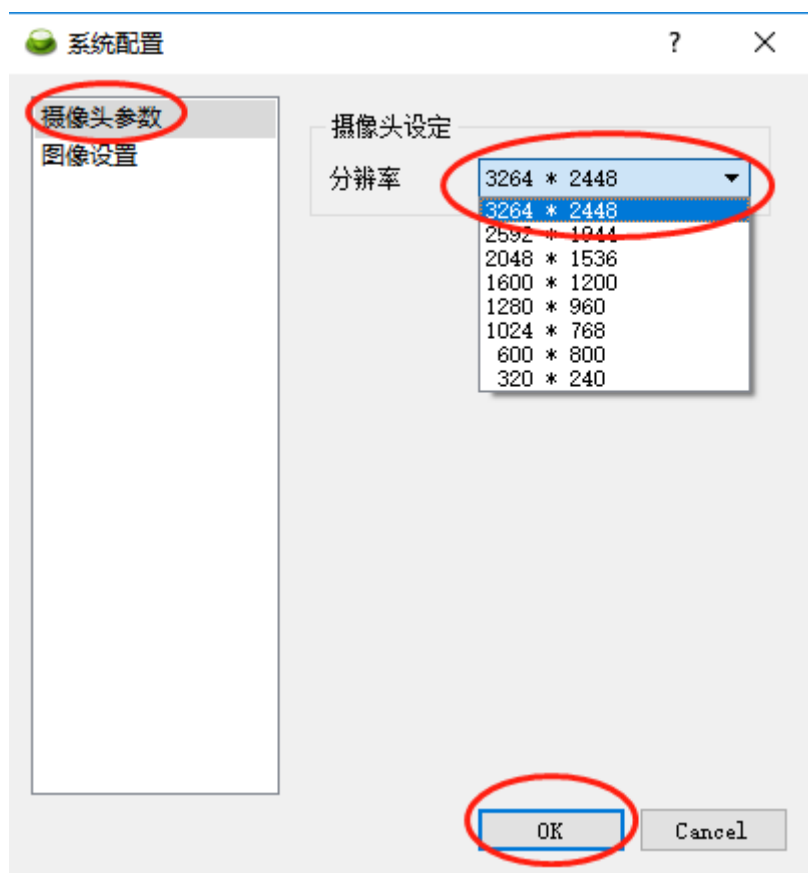


图 5-3 摄像头分辨率设置

对摄像头的分辨率只需要用户在首次使用的时候进行配置，之后会自动记忆用户上次的配置信息，从而简化操作流程，这是依靠 Qt 中的 `QSettings` 进行实现的。用户通常期望应用程序能够记住上次操作时它的设置（窗口大小、位置等），如果将这些信息存储在 Windows 系统注册表中，则可以每次启动应用程序时都读取用户上次操作时的配置信息。`QSettings` 围绕这些抽象技术，使我们能够以便携的方式保存和恢复应用程序设置，提供保存应用程序当前设置的接口，可以方便地保存程序的状态、窗口大小和位置、项的选中状态等等。当我们创建一个 `QSettings` 的对象时，我们需要传递给它两个参数，第一个是公司或者组织的名称，第二个是的应用程序的名称。`QSettings` 还支持自定义存储格式，`QSettings` API 基于 `QVariant`，可以保存很多基础的类型，比如 `QString`、`QRect`、`QImage` 等。`QSettings` 可以存储一系列设置。每个设置包括指定设置名称（键）的一个字符串和一个与该键关联的 `QVariant` 存储数据。使用 `setValue` 可以实现一个设置，使用 `value` 得到一个设置的值。

摄像头的参数配置完成后，即可进行打开摄像头的操作。将摄像头的 USB 连接线接入电脑，当软件右下角设备连接状态栏显示如图 5-4 红框中所示的“摄像头已连接”时可点击图 5-2 中所示的“摄像头启动”启动摄像头。右下角的状态栏主要是显示当前系统中外接设备，如：摄像头、LED 灯、电子天平的连接情况，可以更好的提示用户，让用户提高操作效率。

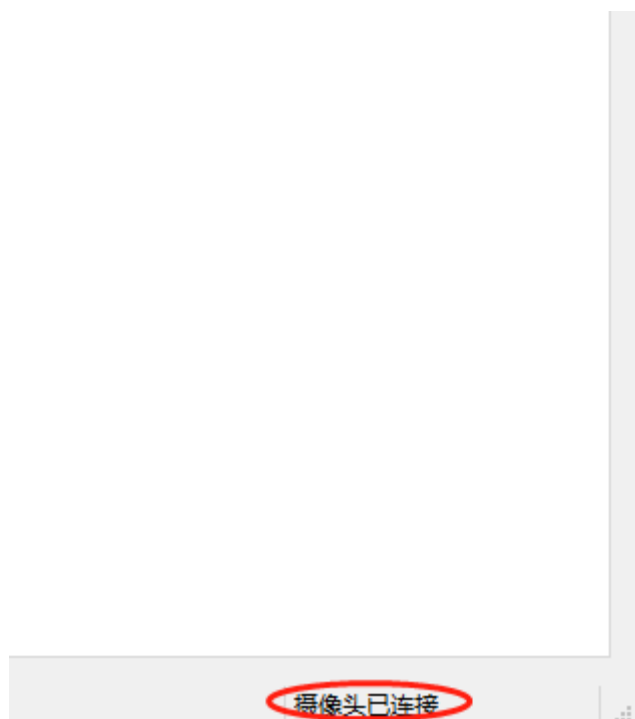


图 5-4 连接摄像头 USB 后右下角状态栏的显示

启动摄像头后，此时的右下方的系统状态显示栏就会显示当前的系统状态，如图 5-5 所示，此时我们可以先观察如图 5-5 所示，此时的系统状态为摄像头启动中，即代表可以进行图像采集的操作。左下角的状态栏主要是显示当前系统的工作状态，如：摄像头已启动、图像已捕捉、图像已打开、算法处理中、算法处理成功等信息，可以让用户判断下一步的所需要进行的操作，从而提高操作的效率。



图 5-5 启动摄像头后左下角状态栏显示

若在摄像头已启动状态需要更改摄像头的分辨率设置，为了摄像头的稳定性，

需要关闭摄像头后，改变需要设定的分辨率，然后再打开摄像头，即可完成更改分辨率的操作。

图 5-6 展示了打开摄像头后图像显示区域显示的摄像头采集到的实时图像，此时可以观察需要检测这一批次种子的大致摆放，若黏连区域过于密集，可在采集前先将托盘抖动后再进行图像的采集。

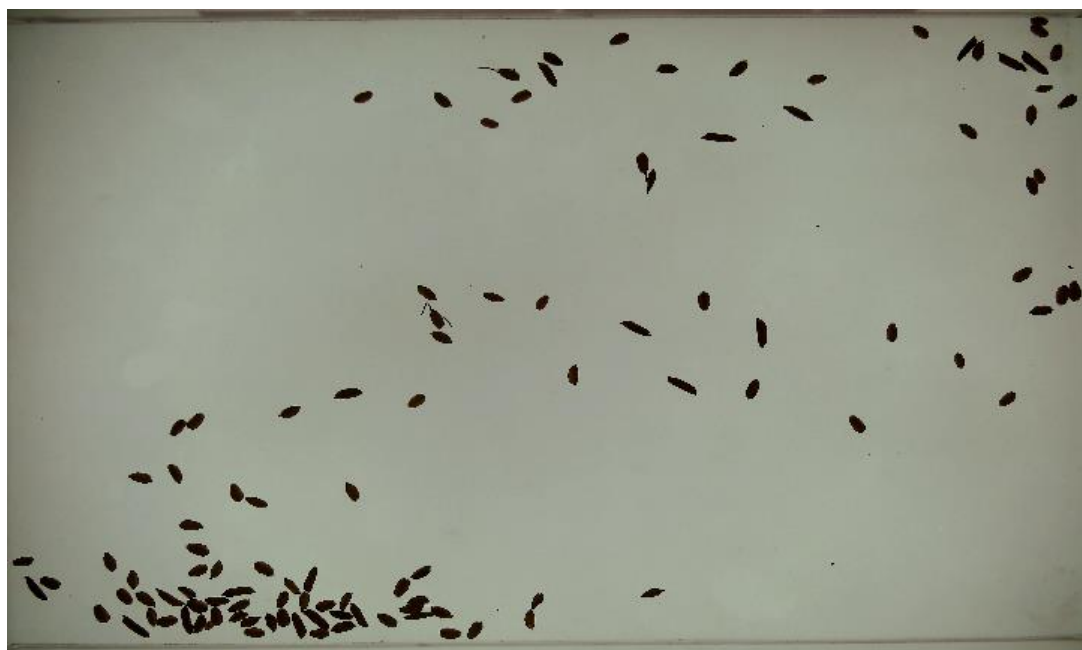


图 5-6 打开摄像头后的图像显示区域

打开摄像头后即可对实时图像进行采集，单击图 5-2 中所示的拍照按钮即可进行当前实时图像的采集。图像采集主要分为两个步骤，第一步是摄像头捕捉此时的实时图像，并临时存储到本地的软件根目录下，同时发出采集成功的信号；第二步是软件中相关的显示功能函数，捕捉采集成功的信号，并将存储的图像显示到图像显示区域。图像采集时，左下角的状态显示栏会更新此时系统的状态，此时的系统状态如图 5-7 所示，为“图像捕捉中”状态。图像捕捉完成后，左下角的状态栏会更新其系统状态如图 5-8 中红框所示，显示当前采集成功图像的尺寸大小。

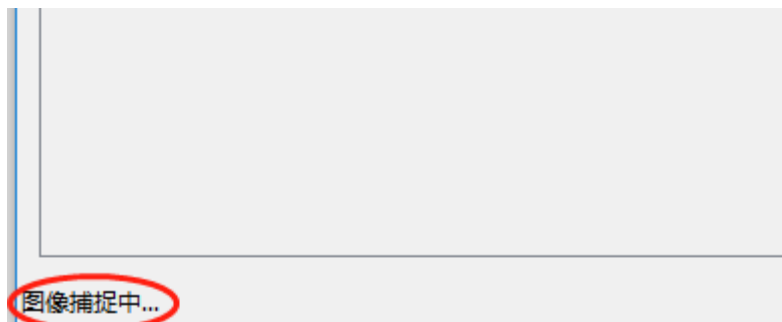


图 5-7 图像采集时的系统状态栏



图 5-8 采集完成后的系统状态栏

对于捕捉的图像，若需要进行自动存储，可在拍照之前在系统配置中的图像设置中进行相应的操作实现。在如图 5-2 所示的按钮中单击“摄像头设置”，进入设置界面后，再选择图像设置，即可进行自动保存图片的设置。如图 5-9 所示勾选“自动保存图片”，之后选择自动保存图片的目录（可以点击“默认目录”将拍取的图像存储在软件的根目录下；也可选择“更改目录”手动选择目录），最后点击“OK”确认即可。若不勾选自动保存图片的这一选项，每次采集到的图像为临时存储到本地的软件所在根目录，且在下一次采集时，新的采集图像会对上一次采集的图像进行覆盖。

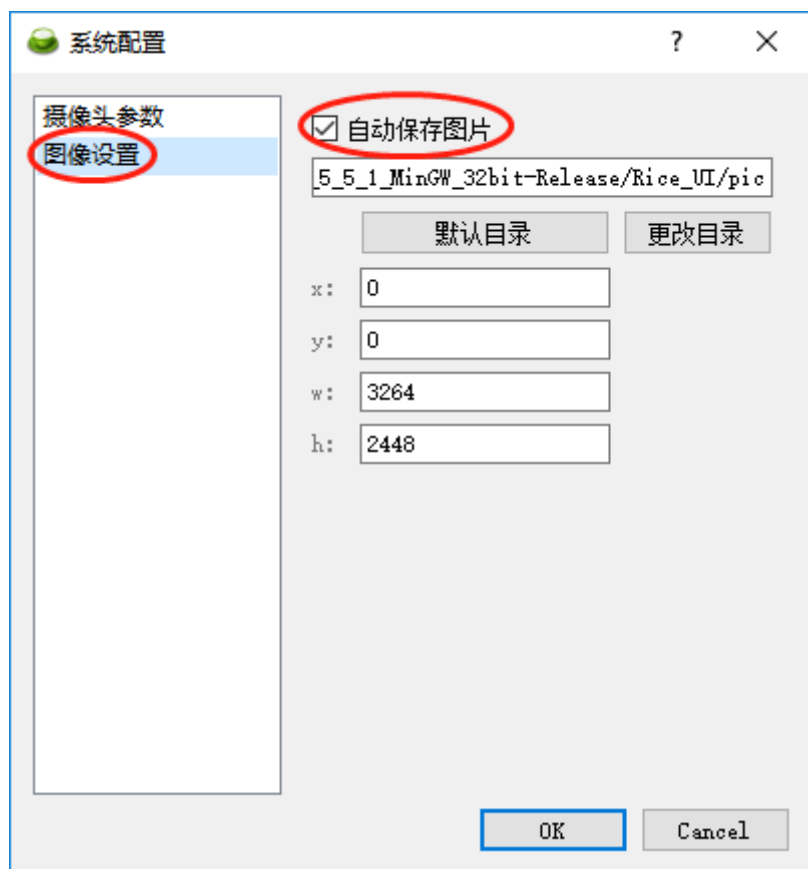


图 5-9 采集图像自动保存

同设置摄像头的分辨率一样，自动保存的设置信息也是通过 Qt 的 `QSettings` 类存储到 Windows 的注册表中的，这样用户下一次打开应用程序进行操作时，就不需要重新进行图像自动保存以及保存路径的设置。图像的自动保存主要是 `QImage` 的 `save` 操作，保存的路径为用户设定的路径，也可选择默认目录，默认目录为软件系统存储在本地的根目录，图像保存的文件名为采集时刻的 Unix 格式时间戳，图像保存的格式为 `jpg` 格式。

也可直接打开本地已采集好的种子图像，即无需由摄像头采集的种子图像，点击“打开”来选择本地采集好的种子图像，如图 5-10 所示可进行本地种子图像的选择。但若此时摄像头已打开时，需要先关闭摄像头。关闭摄像头只需在打开摄像头的按钮上再次点击一次即可。

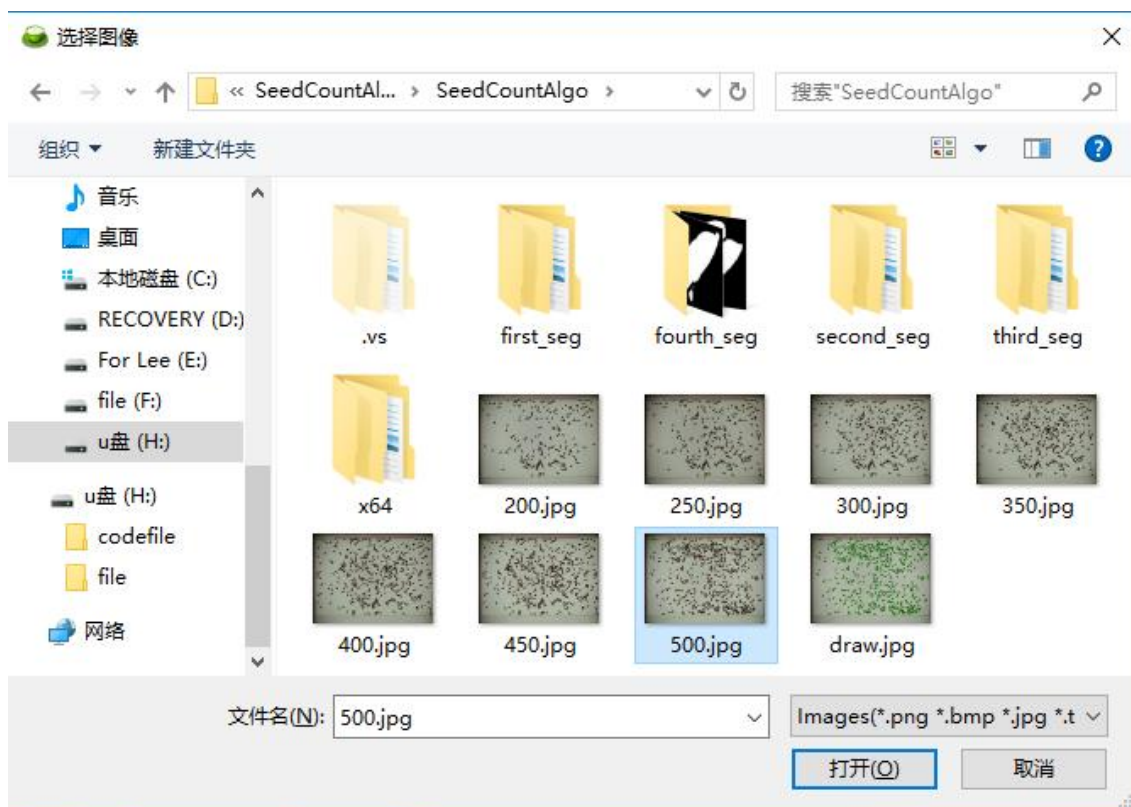


图 5-10 打开图片操作

### 5.2.2 图像处理与相关输出信息操作流程

当已“拍照”或者“打开”了本地图像,此时,可以对种子的图像进行“图像处理”,操作按钮如图 5-11 所示,处理按钮部分主要包括“图像处理”按钮以及“输出 Excel”按钮,单击“图像处理”按钮即可将采集到的图像传递给图像处理算法部分,进行种子的识别与相关信息的统计分析。“输出 Excel”按钮需要在图像处理算法完成之后才能进行操作。



图 5-11 处理按钮部分

图像处理完毕后,在左下角的状态栏会将此时的系统状态更新为如图 5-12 中红框所示的状态,显示算法运行成功,并输出算法运行的时间。





图 5-12 算法处理后的系统状态

图像处理完毕后，图像信息显示框会利用图层技术画出识别到种子的轮廓，如图 5-13 所示，使用蓝色的矩形框框处识别到的种子。

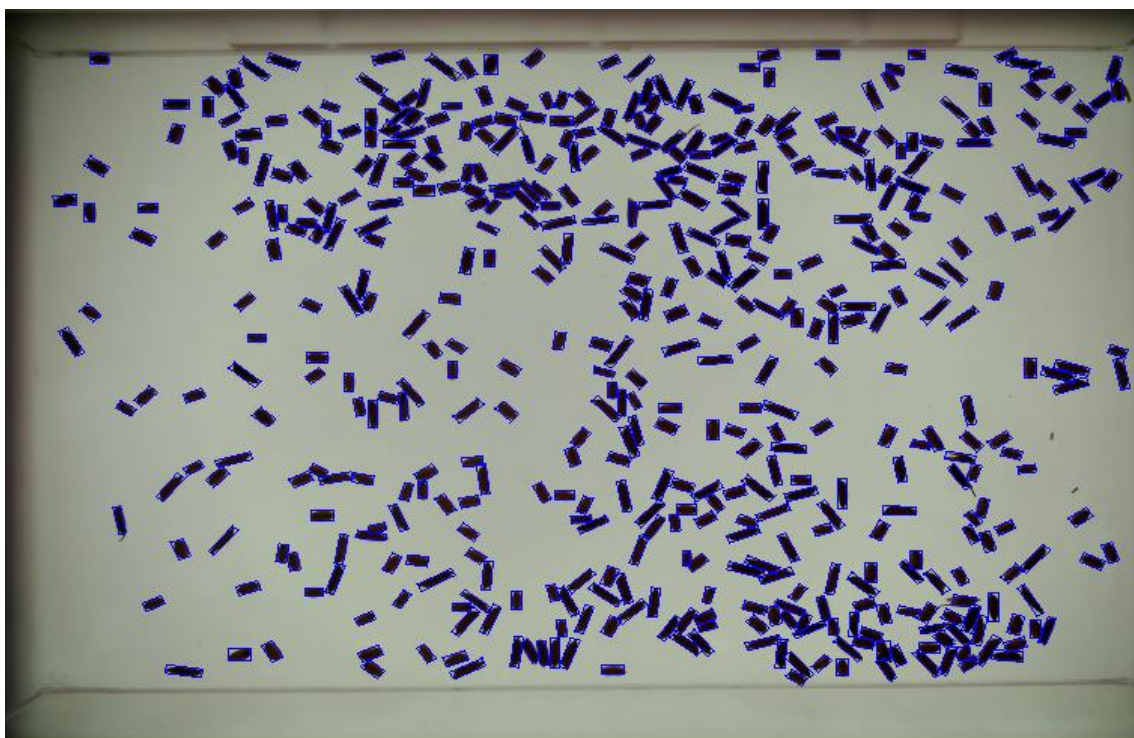


图 5-13 图像处理后的图像显示区域

图像处理完毕后，统计信息显示栏会更新相关的一些列统计信息，如图 5-14



所示。

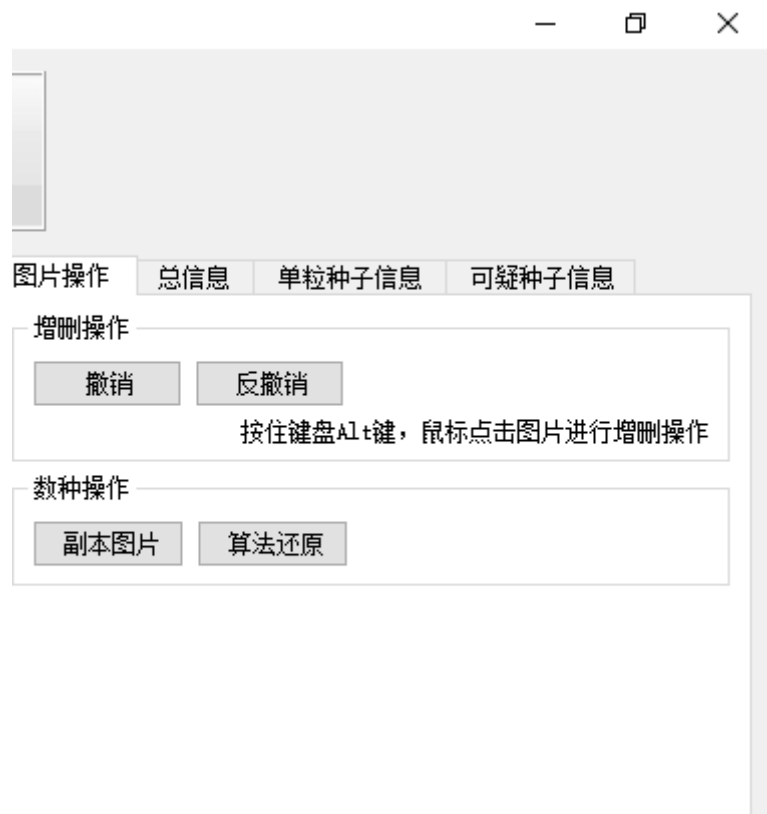


图 5-14 图像处理后的统计信息显示栏

统计信息显示栏主要包括图像操作、总信息、单粒种子信息、可疑种子信息 4 个部分。图 5-14 展示的是统计信息显示栏的图片操作的界面，图片操作界面又包括有“增删操作”、“数种操作”两个部分。其中“撤销”代表的是撤销上一步的增加或删除种子的操作，“反撤销”代表的是取消上一步的撤销操作。“副本图片”是将图像显示区域显示的图像还原成图像处理前的图像，即原始采集图像，“算法还原”指的是撤销算法输出后的所有操作，恢复算法最初的结果。这样可以避免用户的误操作，提高容错性与可操作性。

总信息显示栏中主要显示的是，算法输出的总体统计信息，包括种子的总体数目、粒长粒宽的均值、周长面积均值等的总体数据，如图 5-15 所示。

一个批次水稻种子的样本总体计数是考种的一个基本计数指标。粒长均值、粒宽均值、粒长标准差、粒宽标准差可以更为清晰明了的了解一个批次检测水稻种子的整体粒长、粒宽的情况，包括粒长粒宽的异常情况以及整体的粒长与粒宽的差异性等。周长均值、面积均值与粒长均值、粒宽均值结合，可以获取到这个批次种子的大致形态信息，又有各类标准差参数，也可以了解到一个批次种子的差异性情况。

图片操作	总信息	单粒种子信息	可疑种子信息
	类别	数值	
1	样本计数	500 粒	
2	粒长均值	7.70 mm	
3	粒长标准差	1.95	
4	粒宽均值	2.99 mm	
5	粒宽标准差	0.44	
6	周长均值	21.39 mm	
7	周长标准差	3.58	
8	面积均值	22.66 mm*mm	
9	面积标准差	4.94	
10	长宽比均值	2.68	
11	千粒重	0 g/千粒	

图 5-15 总信息显示栏

单粒种子信息栏中主要显示的是，识别到的每一个种子 ID 号，以及每一个种子的宽度、长度、周长、面积、长宽比等信息，如图 5-16 所示。右侧有下拉滚动条可以拉动查看其它剩余的种子信息。这样的输出方式更为直观，也方便相关处理人员对测试批次的种子数据进行可视化对比，应该来说可以大大简化数据观测难度。

若在此栏对种子的信息进行了修改或编辑，即修改种子的宽度、长度、周长、面积、长宽比等信息，在确认修改或后，或者对种子进行了增删操作。图 5-15 的总体信息栏会根据此时的数据进行实时的更新，即更新样本计数、粒长均值、粒长标准差、千粒重等信息。

其中单粒种子信息栏中最左边一栏的数字为识别到种子的原始编号，此编号不可更改或编辑，改编号指向唯一的一个识别到的种子。这样设计的目的是为了防正，修改 ID 或者其它数据后，会造成重复 ID 的情况，这样会不便于区分，且不利于后续的操作，以及软件系统的关联处理等。

图片操作		总信息	单粒种子信息			可疑种子信息
	ID	宽度	长度	周长	面积	长宽比
1	1	03.13	05.75	017.77	018.01	01.84
2	2	02.89	11.07	027.93	032.01	03.83
3	3	03.19	07.02	020.43	022.42	02.20
4	4	02.76	11.04	027.60	030.47	04.00
5	5	03.61	06.50	020.23	023.49	01.80
6	6	03.30	06.28	019.15	020.71	01.90
7	7	02.94	05.87	017.61	017.22	02.00
8	8	03.40	07.17	021.15	024.39	02.11
9	9	02.94	10.56	027.01	031.08	03.59
10	10	03.31	09.51	025.64	031.51	02.87
11	11	02.87	10.74	027.22	030.86	03.73
12	12	03.83	06.49	020.64	024.86	01.69
13	13	02.83	10.19	026.04	028.82	03.61
14	14	03.62	06.86	020.96	024.83	01.90
15	15	03.24	06.88	020.23	022.29	02.12
16	16	02.92	06.14	018.12	017.93	02.10
17	17	03.40	06.70	020.22	022.82	01.97

图 5-16 单粒种子信息栏

单粒种子信息栏中的默认排列是按照种子的 ID 号，也即算法中的识别顺序排列的，也可以点击“宽度”使其排列顺序由宽度数据由高到低进行排列。同理也可点击“长度”、“周长”、“面积”、“长宽比”等使其相应的列的数据由高到低进行排列，图 5-17 展示了点击了“宽度”后，按种子粒宽的数据由高到低排列的结果。这样更方便用户查看与定位特定特征的种子，使得软件操作更方便也更为灵活。

图片操作		总信息	单粒种子信息		可疑种子信息	
	ID	宽度	长度	周长	面积	长宽比
4	483	04.04	07.34	022.77	029.68	01.82
5	378	04.04	06.81	021.71	027.53	01.68
6	286	04.02	06.83	021.71	027.48	01.70
7	32	04.01	06.67	021.34	026.70	01.66
8	87	03.97	06.54	021.03	025.98	01.65
9	18	03.91	06.68	021.17	026.09	01.71
10	423	03.86	06.29	020.29	024.25	01.63
11	186	03.83	06.70	021.07	025.68	01.75
12	12	03.83	06.49	020.64	024.86	01.69
13	103	03.81	06.64	020.90	025.30	01.74
14	297	03.78	07.05	021.66	026.65	01.87
15	208	03.77	06.68	020.90	025.20	01.77
16	159	03.77	06.20	019.94	023.39	01.64
17	363	03.76	06.77	021.07	025.47	01.80
18	345	03.76	06.55	020.62	024.62	01.74
19	318	03.75	07.52	022.54	028.20	02.01
20	275	03.75	07.06	021.62	026.47	01.88

图 5-17 按宽度排列的单粒种子信息栏

可疑种子信息栏中主要显示的是，算法中无法识别与分割的区域，同单粒种子信息一样，也包括 ID 号、宽度、长度、周长、面积等信息，图 5-18 对可疑信息栏进行了展示。

图片操作		总信息	单粒种子信息		可疑种子信息	
	ID	宽度	长度	周长	面积	长宽比
1	1	08.94	09.26	036.39	082.73	01.04

图 5-18 可疑种子信息栏

图像处理完成后，可对处理后的统计结果进行 Excel 表格输出。在输出 Excel 窗口中，需要输入此次处理的样本的名称，也需要选择保存的方式，其中“保存为新的 Excel 文件”表示生成一个新的 Excel 文件，“追加保存”表示在上次生成的 Excel 文件中追加保存此次的统计信息。输出统计信息保存为 Excel 的操作如图 5-19 所示。图 5-20 显示了输出的 Excel 表格的实例，由于篇幅的限制只截取了输出到 Excel 表格中的部分信息，表格中除了前面所述的统计信息之外，还有每一个批次的样本名称，以及每次处理测试后的输出记录时间。

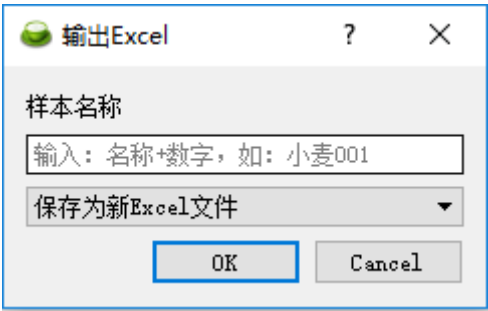


图 5-19 输出 Excel 表格

	A	B	C
1	样本名称	记录时间	样本计数
2	#110564200米	2019-03-13 14:18:15	200
3	#110564250米	2019-03-13 14:18:51	250
4	#110564300米	2019-03-13 14:19:15	300
5	#110564350米	2019-03-13 14:19:34	350
6	#110564450米	2019-03-13 14:19:52	450
7	#110564500米	2019-03-13 14:20:10	500

图 5-20 Excel 表格实例

5.2.3 对考种统计信息删除与定位的操作

在图像区域进行滑轮滚动的操作可以以鼠标所在处为焦点进行放大缩小的操作。将鼠标放在图像区域种子部分的右键操作还能进行删除种子与种子定位的操作，其选项图如图 5-21 所示。

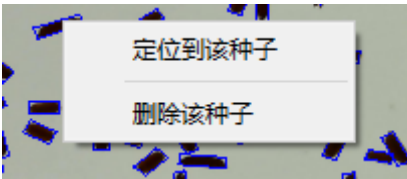


图 5-21 图像区域右键操作

“定位到该种子”操作可以将现鼠标所指定的种子定位到统计信息中单粒种子信息栏中的该种子的信息。“删除该种子”操作可以将现鼠标所指定的种子删除，由图层技术去除轮廓矩形框，并同时统计信息进行更新。删除前后对比图如图 5-22 所示。

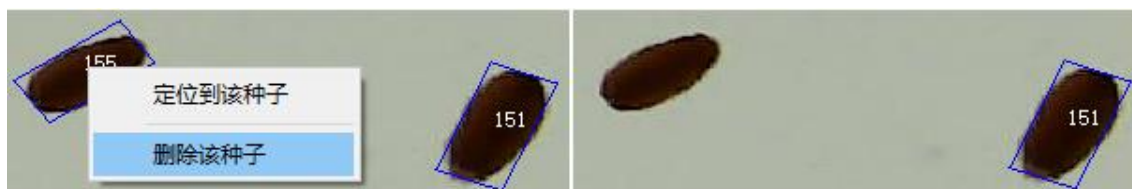


图 5-22 图像区域右键删除种子操作

在统计信息的单粒种子信息栏同样可以完成定位与删除操作。在单粒种子信息栏的某一特定种子的行左键单击，即可在图像中对该种子进行定位，图 5-23 展示了对 ID 号为 155 种子的定位结果。在单粒种子信息栏的某一特定种子的行右键单击，即可删除此种子，图 5-24 展示了对 ID 号为 155 种子删除后的图像显示区域的结果。



图 5-23 单粒种子信息栏左键定位种子操作结果



图 5-24 单粒种子信息栏右键删除种子操作结果

在图像显示区域或者是单粒种子信息栏区域进行定位或者删除操作都是 4.2.2 章节所述观察者模式设计的应用。

#### 5.2.4 对天平与 LED 灯的操作

如图 5-25 所示，在“设备”操作按钮栏中，可以对顶灯和底灯进行控制。当已

经打开顶灯或者打开底灯时，按钮文本信息变为“关闭顶灯”或“关闭底灯”，方便用户操作。

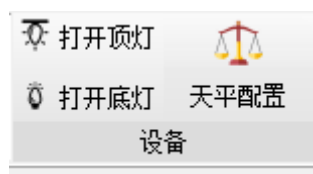


图 5-25 设备按钮部分

点击图 5-25 中的“天平配置”可对天平进行设置，天平配置的窗口如图 5-26 所示，天平配置主要分为两个部分，一个是选择天平设备，另一个就是串口参数的设置。天平设备默认选择伯尼塔 DNA223B 型号，串口参数波特率默认设定为 9600，数据位设置为 8 位，停止位为 1 位，没有校验与流控。选好连接的串口后，单击连接即可通过串口连接上天平，此时天平配置窗口的右下角会显示“连接成功，请按打印进行测试”的提示信息，此时需要在天平上按下打印键确认是否连接成功。按下打印键后，连接会出现“数据通讯正常，请关闭窗口”的提示信息如，此时天平已经连接成功。



图 5-26 天平配置窗口

天平连接成功后，就可以对千粒重信息进行更新。将考种完成的种子放在天平上，待天平数值稳定后就按下打印键，此时左下角的系统状态显示栏会显示天平测量的稳定情况若显示“天平已稳定”，则表示测得的为有效数据，“总信息栏”的千粒重会自动更新，否则不会更新千粒重信息，需要重新进行水稻种子的重量测量。成功更新千粒重信息的统计信息的“总信息栏”如图 5-27 所示。

图片操作	总信息	单粒种子信息	可疑种子信息
	<b>类别</b>	<b>数值</b>	
1	样本计数	113 粒	
2	粒长均值	7.23 mm	
3	粒长标准差	1.54	
4	粒宽均值	3.12 mm	
5	粒宽标准差	0.49	
6	周长均值	20.70 mm	
7	周长标准差	2.93	
8	面积均值	22.32 mm*mm	
9	面积标准差	4.78	
10	长宽比均值	2.42	
11	千粒重	26.5487 g/千粒	

图 5-27 千粒重信息更新后的总信息栏

其中，样本计数即为识别到的种子的个数；粒长、粒宽均根据识别到种子的最小外接矩形的长和宽。周长、面积分别为识别到的各个种子轮廓的周长、面积，可根据 OpenCV 的 ArcLength 函数计算相关轮廓的周长，OpenCV 的 ContourArea 函数计算相关轮廓的面积。均值即为所有识别到单个种子的相关指标的平均值，均值可以表示一组数据集中趋势的量数。标准差就是我们数学上所说的均方差，它可以反映一个数据集的离散程度。

### 5.3 系统测试

每个系统的开发都不能抛开实际的运行情况，测试可以保证系统运行过程之中的稳定性。水稻考种仪的测试主要包括有考种仪软件在不同操作系统中的测试与使用以及数种算法的识别精度测试。

#### 5.3.1 软件系统兼容性测试

水稻考种系统分别在目前相对主流的 32 位与 64 位的 Windows 操作系统中进行了测试，包括 Win7、Win8、Win10 等。最后测试的结果如表 5-1 所示，可以看出，水稻考种系统可以在目前的主流 Windows 系统中正常运行，说明系统具有良好的



适配性。

表 5-1 系统适配性测试

	安装	启动	算法运行	软件操作
Windows7 32 位	正常	正常	正常	正常
Windows7 64 位	正常	正常	正常	正常
Windows8 32 位	正常	正常	正常	正常
Windows8 64 位	正常	正常	正常	正常
Windows10 32 位	正常	正常	正常	正常
Windows10 64 位	正常	正常	正常	正常

5.3.2 算法稳定性测试

考种算法识别稳定性的测试主要包括两个部分。其中第一个部分是对算法的适配性进行测试，第二个部分是对算法的识别精度进行定量测试。本次测试是在 Windows10 64 位的操作系统下进行的。

首先是对算法的适配性进行测试，适配性主要是测试算法在大量采集到的待处理图像中的运行成功率。进行适配性测试的首要任务是采集大量的待处理图像，且采集图像的样本分布需要符合均匀分布，即每个批次的种子图像都等可能的采集，这样更能保证测试结果的确信度。图 5-28 是采集到的 100 个批次种子的图像，其中每个批次的种子图像采集了 10 张，总共有 1000 个测试样本。图像文件的第一个数字代表采集图像的种子批次，第二个数字代表采集图像在批次中的编号。如：00.jpg 代表第一个批次中的第一个批次中的编号为 0 的采集样本，15.jpg 代表的是第 2 个批次中的编号为 5 的采集样本。

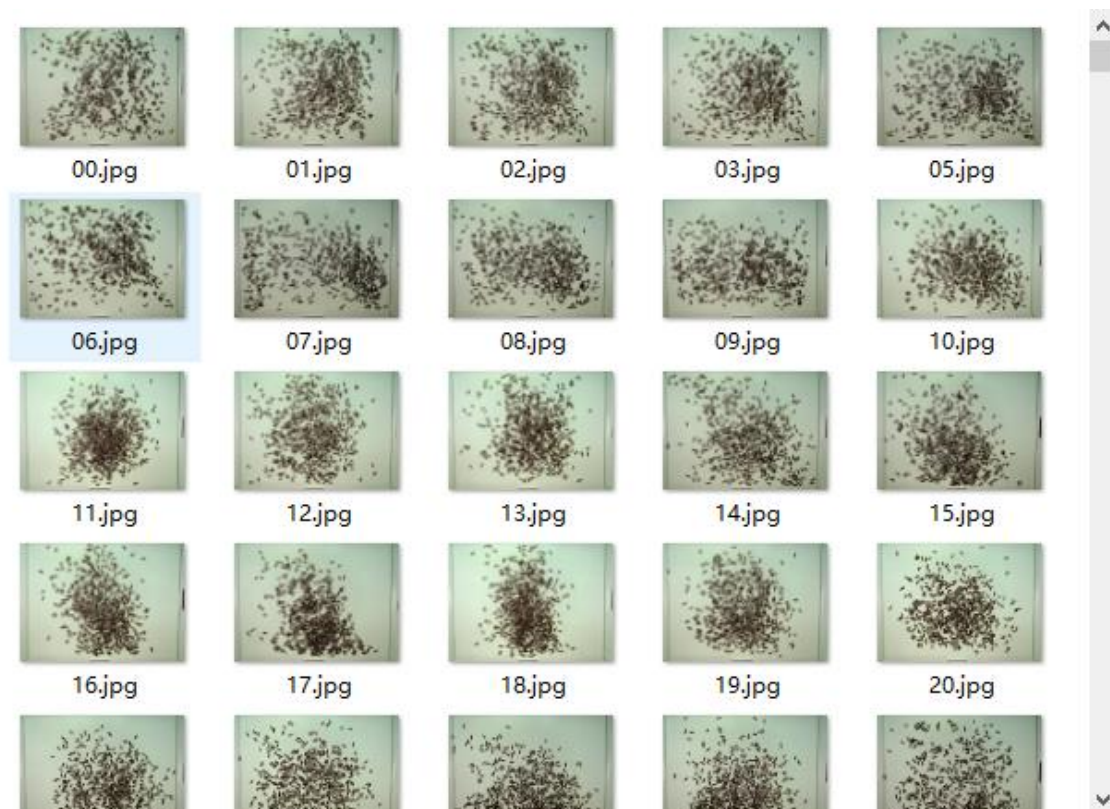


图 5-28 算法适配性测试图像

完成算法适配性测试图像的采集后,就需要对图像进行自动化的测试。对算法源码进行预编译处理、编译、汇编、链接等操作,生成可供 DOS 命令调用的可执行文件。再编写 DOS 命令对算法生成的可执行文件进行调用,以实现自动化测试。图 5-29 是自动化测试中用到的.bat 文件(存放 DOS 命令)以及算法生成的.exe 文件(可执行程序)。

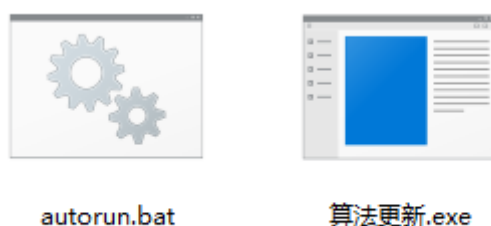
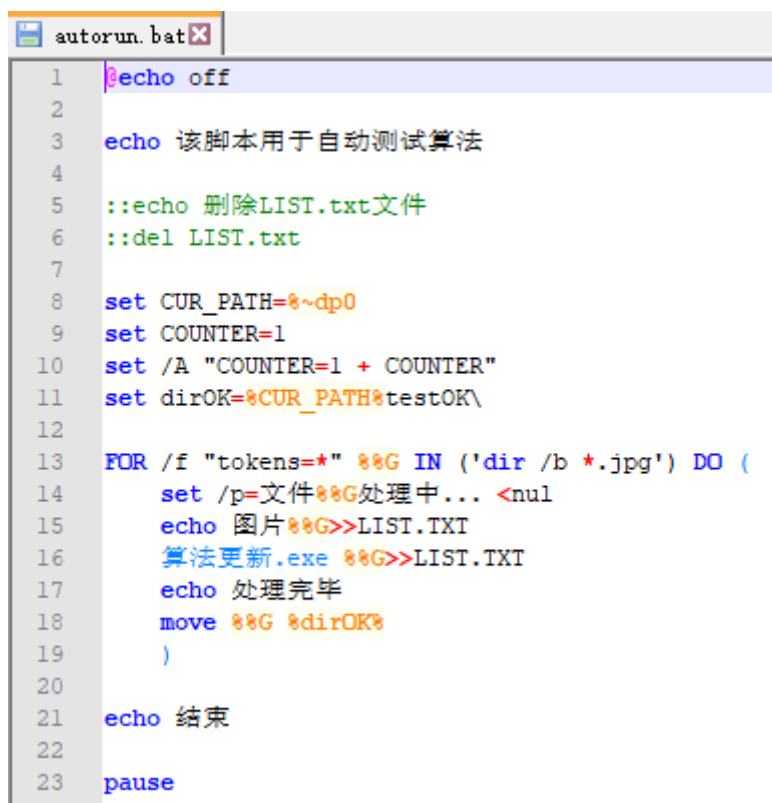


图 5-29 算法适配性测试图像

其中 DOS 命令的(即.bat 文件)的编写如图 5-30 所示。此脚本循环调用算法生成的可执行文件,将图像路径作为参数传入,最后将处理成功的图像信息输出

到 List.txt 中，且将成功处理的图像集中到一个文件夹内。

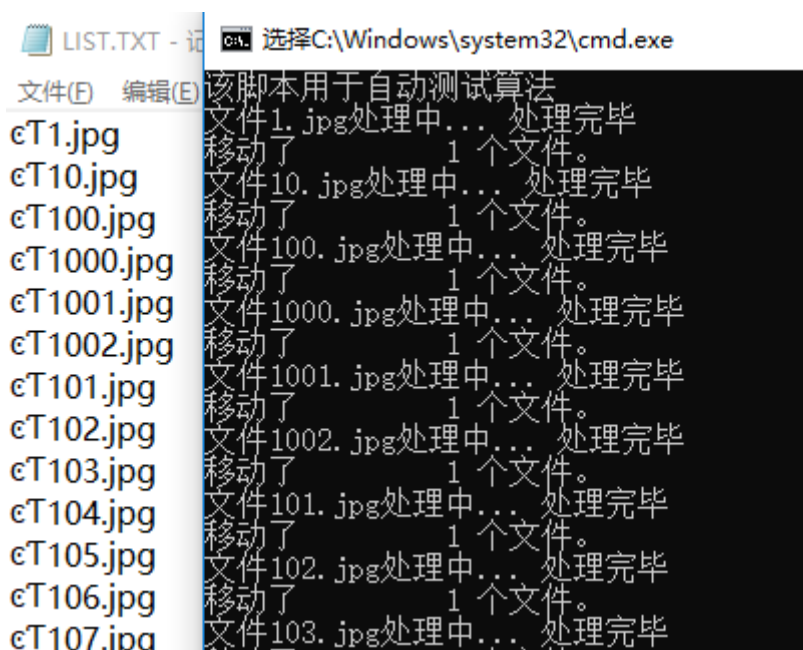


```

1  echo off
2
3  echo 该脚本用于自动测试算法
4
5  ::echo 删除LIST.txt文件
6  ::del LIST.txt
7
8  set CUR_PATH=%~dp0
9  set COUNTER=1
10 set /A "COUNTER=1 + COUNTER"
11 set dirOK=%CUR_PATH%testOK\
12
13 FOR /f "tokens=*" %%G IN ('dir /b *.jpg') DO (
14     set /p=文件%%G处理中... <nul
15     echo 图片%%G>>LIST.TXT
16     算法更新.exe %%G>>LIST.TXT
17     echo 处理完毕
18     move %%G %dirOK%
19 )
20
21 echo 结束
22
23 pause
  
```

图 5-30 算法适配性自动测试脚本

脚本运行的结果如图 5-31 所示，最终采集的图像都能在算法上运行成功，通过率 100%。



```

LIST.TXT - 记事本
该脚本用于自动测试算法
文件1.jpg处理中... 处理完毕
移动了 1 个文件。
文件10.jpg处理中... 处理完毕
移动了 1 个文件。
文件100.jpg处理中... 处理完毕
移动了 1 个文件。
文件1000.jpg处理中... 处理完毕
移动了 1 个文件。
文件1001.jpg处理中... 处理完毕
移动了 1 个文件。
文件1002.jpg处理中... 处理完毕
移动了 1 个文件。
文件101.jpg处理中... 处理完毕
移动了 1 个文件。
文件102.jpg处理中... 处理完毕
移动了 1 个文件。
文件103.jpg处理中... 处理完毕
移动了 1 个文件。
文件104.jpg处理中... 处理完毕
移动了 1 个文件。
文件105.jpg处理中... 处理完毕
移动了 1 个文件。
文件106.jpg处理中... 处理完毕
移动了 1 个文件。
文件107.jpg处理中... 处理完毕
移动了 1 个文件。
  
```

图 5-31 算法适配性自动测试脚本

其次是对算法的检测精度的测试，要想形成好的检验对照，就需要人工进行种子的计数。在测试前，准备了 10 个批次的种子，且每批次种子数量都以 50 进行递增。首先人工数出一定的种子个数，然后记录此批次的真实种子个数，再进行算法的识别，进行对照。测试结果如表 5-2 所示，从表中可以看出，算法识别了 10 个批次的种子，从 200 到 650 的数量。其中数种的最大误差不超过 0.83%，表现较为良好。

表 5-2 考种算法测试结果

样本批次	实际种子数量	本文算法计数结果	计数误差
1	200	200	0
2	250	250	0
3	300	300	0
4	350	350	0
5	400	399	0.25%
6	450	450	0
7	500	498	0.40%
8	550	548	0.36%
9	600	595	0.83%
10	650	646	0.61%

## 5.4 系统验证

此章节对本文中设计的水稻考种系统与常规考种方法进行对比，然后根据实际情况分析系统的优缺点。

常规的考种方法是人工考种，但人工考种存在偶然性，且耗时又耗人力，所以本文设计的水稻考种系统与传统的人工考种相比，具有用时短、精度高、成本低的优势。表 5-3 显示了两种方法的对比，可以看出，人工考种耗费的时间太长，且在有大量考种需求时，时间消耗太长，效率十分低下。

表 5-3 本系统与人工考种方法的对比

	本系统	人工考种
实际粒数	213/512	213/512
考种耗时	0.21s/0.56s	76s/205s
考种结果	213/511	213/507

在黏连区域的分割中，迭代腐蚀也是其中的一种比较常见的算法<sup>[51-53]</sup>，下面将迭代腐蚀的算法与本系统的识别算法进行对比。

首先对黏连区域的分割情况进行对比。如图 5-32 展示了采集了某个批次种子图像中的黏连区域。图 5-33 展示了本文设计的水稻考种系统的分割效果，能够成功将单个种子从黏连区域中分割出来，且分割完成后的还原度较高。图 5-34 展示了迭代腐蚀算法的分割效果，因为左边黏连的种子连通度太高，导致腐蚀无法将两个黏连区域分离，从而分割失败。显然，本文设计的分割算法效果更好，且还原度更高。



图 5-32 黏连区域

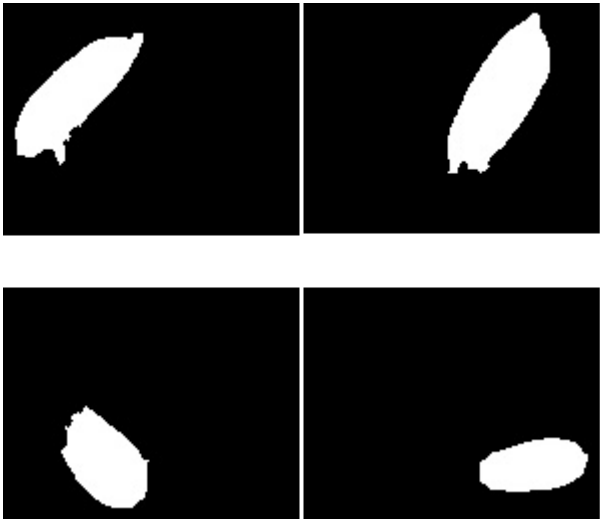


图 5-33 考种系统算法分割效果



图 5-34 迭代腐蚀算法分割效果

表 5-4 是迭代腐蚀算法与本系统所设计算法的整体对比。分别实验了实际粒数为 400 和 800 时的识别情况。其中，当实际粒数为 400 时，本系统的识别算法与迭代腐蚀算法的运行时间相差不大，但是当实际粒数到 800 时，由于黏连区域情况的增多，且迭代腐蚀的时间复杂度和自身的时间损耗相对较大，所以运行造成此时的运行时间差距较大。对比最后的考种结果也能得出本文所设计系统的识别算法更优的结论。

表 5-4 本系统识别算法与迭代腐蚀算法对比

	本系统识别算法	迭代腐蚀算法
实际粒数	400/800	400/800
考种时间	0.5s/1.2s	0.8s/3.7s
考种结果	400/798	398/789

5.5 本章小结

本章主要对系统的主要操作进行了介绍，还对系统的运行适配性进行了测试，最后还对考种算法的稳定性进行了测试。软件的操作便捷且界面友好，测试结果表明系统的适配性较强，算法也较为稳定，误差在 0.83%以内。最后进行了图像识别算法比对。

## 第六章 总结与展望

### 6.1 总结

水稻种子的粒长、粒宽以及其千粒重信息是衡量其质量优劣的一系列重要的指标，而要想获得这些参数，首先就需要对水稻种子进行识别计数。以往的计数和识别技术由于存在识别计数的精度与速率上的局限性，所以并不能很准确且快速的进行相关信息的反馈，本系统所涉及的基于图像处理的水稻考种系统利用梳理图像处理技术，解决了准确率与识别速率的问题。采用工业高拍仪对水稻种子图像进行实时采集；设计上顶灯、下顶灯为种子采集背景提供良好的基础；利用形态学开操作对种子进行去尾掉处理，提高识别精度；设计了距离变换与分水岭相结合的多个水稻种子黏连区域的分割算法；设计了划线分割算法对两个种子黏连区域进行了分割；将图层技术与设计模式中的观察者模式应用在软件显示界面与算法处理输出统计信息中，增加软件界面的可操作性；实现了水稻种子的紧密排列功能，能够更加明了的对比水稻种子的粒长粒宽等信息。系统整体的计数识别准确率可达到 99.12%，且一批次识别种子数在 500 个左右时，识别速度可控制在 1 秒左右，计数准确且耗时短。

本课题所做的主要研究工作如下：

（1）对水稻考种的国内外现状进行了综述，对现有的考种技术进行了介绍和分析，对数字图像处理技术的应用进行了探讨，为基于图像处理的水稻考种系统的后续开发和设计工作奠定了基础；

（2）介绍了基于图像处理的水稻考种系统的硬件以及软件环境，通过分析必要的处理流程，确定了系统的主要架构，并对硬件以及软件中的每个模块做了设计说明与实现。

（3）在充分调研之下，结合实际情况，挑选相机模组、镜头、光源以及采集箱体等设备，并搭建了考种仪的图像采集硬件部分；

（4）由于采集到的水稻种子图像存在外界噪声以及种子本身尾掉等干扰因素，所以使用图像平滑技术以及开操作对种子进行去尾掉操作，提高识别的以及输出指标的精准度；

（5）对预处理之后的图像进行轮廓提取，以及凸缺陷检测，并初步分割出符合面积与缺陷点个数条件的单个种子区域，将疑似黏连的区域进行存储，为下一步的黏连区域分割做好准备；

（6）将距离变换与分水岭算法相结合，对 2 个及 2 个水稻种子以上黏连区域

进行分割，将黏连区域的水稻种子完全分割开来，或者分割剩下两个黏连情况严重的种子区域，再使用划线分割的方式进行分割；

（7）输出图像处理结果，并使用图层技术画出相应的识别到的单个种子的轮廓，利用设计模式中的观察者模式将统计信息与图像、统计信息进行联动显示，提高系统界面的可操作性；

（8）为水稻研究人员提供了水稻种子的紧密排列功能，这是一种更为方便快捷且更为明了的对比方式。

## 6.2 展望

基于对图像处理自动计数研究现状的了解，现阶段的考种系统虽然相对稳定，但还有一些方面需要进行优化以及一些问题需要去解决。在此基础上，以下几点还需要改进：

（1）目前的照明系统会在采集托盘的四周角落产生小范围黑框，黑框的存在可能对图像识别的算法产生影响，即可能会造成误识别现象的产生，所以照明系统以及装载种子的托盘还需要进一步的改进，为算法提供更加精确更加稳定的种子图像。对于检测水稻更为细节的参数，如垩白率等，需要提高摄像头的像素分辨率，以获取更精细的光源；

（2）目前的种子识别算法只能针对水稻种子，若是对其他类型的种子，如：玉米、小麦等，则不能完全适用于本算法，后续算法应该加强适用性的设计，或者在算法中针对不同类型的种子加入不同的评价标准，将数种算法进行推广；

（3）操作界面未设置快捷键操作，对于反复使用考种系统的研究人员，软件中加入快捷键提高用户体验加快工作效率。



## 致谢

光阴荏苒，时光飞逝，研究生的学习和生活已经接近了尾声。本论文是在贾海涛导师与贾宇明院长的指导下完成的。两位贾老师都经常提醒我要有严谨的科研与求学态度、精益求精的工作作风，这对我影响深远，都是我研究生阶段的宝贵财富与经历。在我的研究生涯里，有过欢乐与喜悦，也有过迷茫与彷徨，不论结局如何，我都要对帮助过我的老师、同学表示我由衷的感谢。

本论文从选题到完成，每一步都是在贾老师的悉心指导下完成的，倾注了贾老师大量的心血。贾老师对我论文的研究方向做出了指导性的意见和推荐，在论文撰写过程中及时对我遇到的困难和疑惑给予了悉心的指点，提出了许多有益的改善性意见，投入了超多的心血和精力。对于老师给予我的帮忙和关怀表示诚挚的谢意！同时，我要感谢我的各位前辈、同窗和后辈们，大家在实验室生活中互相学习，互相帮忙，共同度过了一段完美难忘的时光。

此外，感谢朋友以及同学们在论文编写中带给的大力支持和帮忙，给我带来极大的启发，还要感谢哥哥为我的实验提供了很好的实验素材。也要感谢参考文献中的作者们，透过他们的研究文章，使我对研究课题有了很好的出发点。

最后，谢谢论文评阅老师们的辛苦工作。衷心感谢我的家人、朋友，以及同学们，正是在他们的鼓励和支持下我才得以顺利完成此论文。

## 参考文献

- [1] 褚庆全,齐成喜,杨飞等. 我国杂交粳稻发展现状、问题及其对策[J]. 作物杂志, 2005( 1): 9—12.
- [2] 许桂玲, 纪洪亭, 潘剑等. 水稻考种方法探析 [J]. 中国农技推广, 2012( 6): 16 — 17.
- [3] 巢淑娟, 张贵明. 浅析数粒机的现状及发展 [J]. 农业与技术, 2013, 33( 2): 12.
- [4] 常礼, 马成学, 高理富. 基于光电信号的玉米粒计数方法的研究 [J]. 中国科学技术大学学报, 2015, 45( 4): 275 — 279.
- [5] 宋初苏, 华娇, 蓝景针等. 转盘斜刮式光电自动数粒仪设计 [J]. 农业机械学报, 2011, 42( 11): 75, 89 — 92.
- [6] 王刚. 基于机器视觉的玉米千粒重快速检测仪的研制 [D]. 吉林: 吉林大学, 2012.
- [7] 裴骄阳. 玉米种粒千粒重的自动测量系统开发 [D]. 杨凌:西北农林科技大学, 2014.
- [8] 孙宏佳. 基于机器视觉的花生种子自动识别系统设计[D]. 哈尔滨: 哈尔滨理工大学, 2014.
- [9] 王风云, 郑纪业, 唐研等. 机器视觉在我国农业中的应用研究进展分析 [J]. 山东农业科学, 2016, 48( 4): 139—144.
- [10] 吴露露, 郑志雄, 齐龙等. 基于图像处理的田间水稻叶瘟病斑检测方法 [J]. 农机化研究, 2014( 9): 32—35.
- [11] 张芳, 付立思. 基于图像处理技术的黄瓜叶部病害识别诊断系统[J]. 农机化研究, 2014( 9): 213—215, 225.
- [12] 胡维炜, 张武, 刘连忠等. 利用图像处理技术计算大豆叶片相对病斑面积 [J]. 江苏农业学报, 2016, 32( 4): 774—779.
- [13] 张小津, 刘顺康. DS 型多用数粒仪的研制[J]. 电子工程师, 1995, 2.
- [14] 宋初苏, 蓝景针, 夏世峰等. ZPXG-18 型转盘斜刮式自动光电数粒仪和千粒重仪的设计[J]. 浙江农业学报, 2011, 23(5):1023-1028.
- [15] 陈恪. 机器视觉技术在测量中的应用[J]. 中国计量, 2008, 9: 61~62.
- [16] 梁文东, 蒋益敏. 基于 MATLAB 的玉米果穗颗粒统计方法 [J]. 江苏农业科学, 2015, 43(2): 406-408.
- [17] 周南, 郭光靖, 唐先亮, 等. 基于 MATLAB 的水稻种子几何特征提取的程序实现[J]. 安徽农学通报(上半月刊), 2012, 18(7): 187-189.
- [18] 王传宇, 郭新宇, 吴升. 采用全景技术的机器视觉测量玉米果穗考种指标[J]. 农业工程学报, 2013, 29(24): 155-162.
- [19] Giuseppe Gallas. Automation in chromosome classification and cell analysis[J]. Pattern Recognition of Biomedical Objects, 1978, 10(2): 4-9

- [20] Jean T,Marc Olivier. Automatic recognition of cancerous cells using mathematical morphology[C]. In: Proc of Int Conf on SPIE Rochester, 1994, 2359: 392-401
- [21] 王金涛, 刘文耀, 路烁. 流域分割算法在细胞图像分割中的应用[J]. 西南交通大学学报, 2002, 37(3): 290-294.
- [22] 闫文忠, 沈树群. 分水岭算法在粘连染色体图像分割中的应用[C]. 第六届全国信息获取与处理学术会议论文集(2). 2008.
- [23] 马保国, 乔玲玲, 贾寅波. 基于局部自适应阈值的细胞图像分割方法[J]. 计算机应用研究, 2009, 26(2).
- [24] Shatadal P, Jayas D S, Bulley NR. Digital image analysis for software separation and classification touching grains[J]. Transaction of the ASAE. 1995, 28(2): 645-649
- [25] Kim J B, Kim H J, Multi-Solution based watersheds for efficient image segmentation[J]. Pattern Recognition Letters, 2003, 24(13): 473-488
- [26] Pavlidis T, Liow Y. Integrating Region Growing and Edge Detection[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 1990, 12(3): 225-233
- [27] Cheng H D, Jiang X H, Wang Jing li. Color image segmentation[J]. Pattern Recognition, 2002, 35(2): 373-393
- [28] Van den Berg E H, Meesters AGCA. Automated separation of touching grains in digital images of thin section[J]. Computers & Geosciences, 2002, 28(2): 179-190
- [29] Zayas I Y, Y. Pomeranz, F. S. Lai. Discrimination between different wheat by image analysis[J]. Cereal Chemistry, 1985, 62(6): 478-479
- [30] Talukder Ashit, David Casasent, Ha-Woon Lee. Modified binary watershed transform for segmentation of agricultural products[C]. In: Proc of Int Conf on SPIE Boston, 1999, 3543: 53~64
- [31] 毋桂萍, 杨红卫. 改进的 Otsu 算法在粮食颗粒图像分割中的应用[J]. 农机化研究, 2009, 31(6): 159-161.
- [32] 刘娜. 粘连颗粒分割方法研究[J]. 计算机光盘软件与应用, 2012(17): 156-157.
- [33] 高文韬. 基于纹理图像的琵琶面板选取方法研究[D]. 东北林业大学, 2018.
- [34] 杨小平. 基于数学形态学的小波变换图像去噪方法研究[J]. 桂林师范高等专科学校学报, 2007(01): 135-138.
- [35] 陈基伟. 基于数字图像处理的棒材计数方法研究[D]. 山东大学, 2012.
- [36] 李京蔓, 向阳, 李琦, 景桂芬, 郭金明. 机器视觉大视场宽景深光学系统设计[J]. 长春理工大学学报(自然科学版), 2019, 42(01): 18-22.
- [37] 温林凯, 刘英, 姜洋, 邢妍, 孙强. 大孔径长焦距全画幅变焦镜头结构计算与优化设计[J/OL]. 激光与光电子学进展: 1-13[2019-03-15].

- [38] 刘宵婵,陈琛,刘红军,宋涛,李维善,宋向涛.小口径双通道球幕投影镜头的设计[J].应用光学,2018,39(06):890-895.
- [39] 陈智利,刘雨昭,费芒芒,成国贤,王栋.大视场双远心工业镜头光学系统设计[J].西安工业大学学报,2018,38(05):444-450.
- [40] 尚会超,杨锐,段梦珍,段晓伟,张洪斌.机器视觉照明系统的关键技术分析[J].中原工学院学报,2016,27(03):16-21.
- [41] 齐玉娟,李如锋,王延江.基于 VC 及 OpenCV 的数字图像/视频处理演示系统设计及实例展示[J].实验室研究与探索,2018,37(11):141-144+164.
- [42] 纪君峰. 基于 QT/E 的嵌入式系统研究及应用[D]. 北京邮电大学, 2008.
- [43] 李帅. 脊柱核磁共振图像的分割算法研究与实现[D].武汉理工大学,2017.
- [44] 邓超,刘岩岩.基于边缘检测的斜纹布匹瑕疵检测[J].测控技术,2018,37(12):110-113.
- [45] 曹国强, 张京龙, 尹航. 基于凸包特征象棋棋子识别系统设计[J]. 黑龙江大学自然科学学报, 2017, v.34(06):106-110.
- [46] 吴哲夫, 张彤, 肖鹰. 基于 Spark 平台的 K-means 聚类算法改进及并行化实现[J]. 互联网天地, 2016(1):44-50.
- [47] 贾传民. 基于定格动画后期合成技术应用研究[D].
- [48] 聂颖. 设计模式在图形处理软件中的应用[J]. 计算机应用, 2004, 24(z2).
- [49] 杨文锦, 王传兵. 数控系统中 GUI 软件平台设计和实现[J]. 制造技术与机床, 2017(9).
- [50] 孙凌云, 王光伟, SUNLing-yun, et al. 设计模式在军用软件开发中的应用[J]. 现代计算机, 2013(3):5
- [51] 焦雁翔,唐玉琴,黄成志,黄仁军.Matlab 图像处理在水稻谷粒计数中的应用[J].南方农业,2018,12(04):82-84+92.
- [52] 荣斐.基于图像处理的作物种子自动计数软件开发[J].工业设计,2011(07):126-127.
- [53] 张雪飞,王建春,彭凯,孙海波,贾宝红,徐义鑫,杜彦芳,邹美智,杨勇.基于图像处理和蓝牙传输技术的水稻种粒计数计量系统[J].山东农业科学,2018,50(07):157-160.

## 攻读硕士学位期间取得的成果

- [1] 周晓奎;周洪奎;贾海涛. 一种自动考种分析仪 [P]. 中国专利:ZL 2018 2 0312700.8
- [2] 周晓奎;周洪奎;贾海涛. 一种考种用标定托盘 [P]. 中国专利: ZL 2018 2 0312074.2.
- [3] 参与了四川杰莱美科技有限公司“水稻全自动考种仪”项目研发
- [4] 参与了四川杰莱美科技有限公司“米质分析仪”项目研发
- [5] 参与了九寨沟风景名胜区管理局“世界自然遗产景观与生态保护国家级综合观测站”项目实施