

学校代码：10564

学 号：2015307812

分 类 号：TP391

密 级：



華南農業大學

# 硕士学位论文

基于视觉感知的蔬菜害虫计数算法研究

张玉康

第一指导教师：肖德琴 教授

第二指导教师：

学院名称：数学与信息学院

专业学位类别：工程硕士

领域：计算机技术

答辩委员会主席：黄翰 教授

中国·广州

2017年6月

# 华南农业大学

## 学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名: \_\_\_\_\_ 日期: \_\_\_\_\_

## 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属华南农业大学。学校有权保存并向国家有关部门或机构送交论文的复印件和电子版，允许学位论文被查阅或在校园网上发布并供校内师生和与学校有共享协议的单位浏览（除在保密期内的涉密论文外）；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。本人电子文档的内容和纸质论文的内容相一致。

作者签名: \_\_\_\_\_ 日期: \_\_\_\_\_

导师签名: \_\_\_\_\_ 日期: \_\_\_\_\_

## 学位论文提交同意书

本学位论文符合国家和华南农业大学关于研究生学位论文的相关规定，达到学位授予要求，同意提交。

导师签名: \_\_\_\_\_ 日期: \_\_\_\_\_

学科带头人签名: \_\_\_\_\_ 日期: \_\_\_\_\_

## 摘要

传统上主要采用人工计数方式对害虫进行统计分析，存在人工成本大、工作量大、主观性等缺点，近年来利用机器视觉技术对蔬菜害虫进行监测成为研究热点，但是当前绝大多数方法都是在较为理想的环境条件下进行的，不能直接应用于大田环境下害虫的监测。为了研究大田环境害虫发生分布规律，在国家星火计划重点项目“南方蔬菜重大害虫监测及预警技术集成与应用示范”资助下，本文以南方蔬菜重大害虫为研究对象，探索了一种在大田环境下使用黄色诱捕板对蔬菜害虫进行监测计数的新算法。

本文在对比分析以往边缘检测和阈值分割的经典算法基础上，根据害虫监测目标的需要，提出了一种基于结构化随机森林的害虫图像分割算法，以及利用不规则结构的特征提取算法，进一步结合背景去除、干扰目标去除和检测模型计数子算法，集成创制了基于视觉感知的蔬菜害虫计数算法（vegetable pest count algorithm based on visual perception, VPCA-VP），应用于多种害虫的分类计数。本文还利用 C++ 的 MFC 框架研发了一个基于视觉感知的害虫计数应用软件，为用户提供更便捷的交互应用。该软件不仅可以计算单张害虫图像的害虫数量，而且还能累加计算多张害虫图像的害虫数量。本文创新点主要有三个：（1）使用结构化随机森林的害虫图像分割算法避免阴影和背景的干扰；（2）组合使用图像填充算法和对象选择算法去除黄板以外的背景图像；（3）利用不规则结构的特征提取算法计算害虫图像的特征向量。

本文使用了 83 张现场环境下拍摄的图片进行实验与分析，实验结果显示，本文基于视觉感知的蔬菜害虫计数算法的精确率为 94.89%，召回率为 96.29%。其中，蔬菜害虫蓟马的准确率为 93.19%，召回率为 94.50%；烟粉虱的准确率为 91%，召回率为 90%；实蝇的精确率达到 100%，召回率有 92%。本算法性能较好，可以较好的去除阴影、背影和其他类型害虫的干扰，有利于田间环境的害虫计数，在农田监测中有广阔的应用前景。

**关键词：**计算机视觉；害虫识别；随机森林；自相似性描述子；不规则特征提取

# Study on Vegetable Pest Counting Algorithm

## Based on Visual Perception

Zhang Yukang

(College of Mathematics and informatics, South China Agricultural University, Guangzhou  
510642, China)

**Abstract:** Traditionally, manual counting methods was carried out on the number of pests. But due to large labor costs, heavy workload, subjective and other shortcomings, using machine vision technology to monitor vegetable pest has become a hotspot. But the vast majority of current methods are to be carried out under the condition of ideal laboratory, which can't be directly applied to pest monitoring in the field. In order to study the distribution of pests in the field environment, this author studied a new algorithm for counting the southern vegetable pest using yellow sticky trap, which was supported by the National Spark Program Key Project "Southern Vegetable Major Pest Monitoring and Early Warning Technology Integration and Application Demonstration".

Compared to the classical algorithms of edge detection and threshold segmentation, this paper proposes some new algorithms include pest image segmentation sub-algorithm based on the structure of random forest, feature extraction sub-algorithm of irregular structure, background removal sub-algorithm, interference target removal sub-algorithm and detection model counting sub-algorithm. Those sub-algorithms were integrated to create a vegetable pest count algorithm based on visual perception (VPCA-VP) to classify and count multiple pests. This paper also uses C++ framework of MFC to develop a pest counting software based on visual perception, providing a more user-friendly interface for users. The software can calculate the number of pests for a single pest image, it also accumulates the number of pests by multiple images. There are three main innovations in this paper: (1) using structured random forest pest image segmentation algorithm to avoid the interference about shadow and background; (2) Combining the image fill algorithm and the object selection algorithm to remove the background image; (3) using the feature extraction algorithm of irregular structure to calculate the eigenvector of pest image.

In this paper, 83 images were taken and analyzed. The experimental results showed that the accuracy rate of the vegetable pest counting algorithm based on visual perception (VPCA-VP) was 94.89% and the recall rate was 96.29%. Among them, the accuracy rate of Thrips was 93.19% and the recall rate was 94.50%. The accuracy rate of Whitefly was 91% and the recall rate was 90%. The accuracy rate of Fruit Fly was 100% and the recall rate was 92%. The algorithm can be better to remove the interference about shadow, background and other types of pests, so it is conducive to the field environment pest counting. The algorithm has good performance and has a wide application prospect in farmland monitoring.

**Key words:** computer vision; pest identification; random forest; self-similarity descriptor; irregular feature extraction

# 目 录

1 前言 .....	1
1.1 研究背景 .....	1
1.2 研究目的和意义 .....	2
1.3 国内外研究现状 .....	3
1.4 本论文的主要工作 .....	4
1.5 论文结构 .....	6
2 基于计算机视觉的害虫图像识别技术探究 .....	7
2.1 害虫图像的采集和预处理 .....	7
2.2 害虫图像分割的主要技术 .....	7
2.2.1 害虫图像的边缘检测 .....	8
2.2.2 害虫图像的阈值分割 .....	13
2.3 害虫图像特征提取技术 .....	15
2.3.1 害虫的直方图特征 .....	15
2.3.2 害虫的纹理特征 .....	16
2.3.3 害虫的边界特征 .....	17
2.3.4 害虫的区域特征 .....	17
2.4 害虫图像识别主要算法 .....	18
2.4.1 支持向量机 .....	18
2.4.2 人工神经网络 .....	18
2.5 基于黄色诱捕板的分割算法分析 .....	18
2.5.1 基于黄色诱捕板害虫图像的边缘检测结果分析 .....	19
2.5.2 基于黄色诱捕板害虫图像的阈值分割 .....	21
2.6 基于计算机视觉的害虫图像识别小结 .....	23
3 基于黄色诱捕板的害虫快速计数算法 .....	24
3.1 VPCA-VP 算法总体描述 .....	24
3.2 基于黄色诱捕板的害虫图像采集 .....	26
3.3 害虫图片的预处理 .....	27
3.4 基于结构化随机森林的害虫图像分割子算法 .....	27
3.4.1 算法原理 .....	27

3.4.2 算法过程 .....	30
3.5 背景去除 .....	35
3.6 干扰害虫的去除 .....	36
3.7 害虫特征提取子算法 .....	36
3.7.1 害虫颜色特征的提取 .....	36
3.7.2 害虫纹理特征的提取 .....	37
3.7.3 害虫边界特征的提取 .....	38
3.7.4 害虫区域特征的提取 .....	39
3.7.5 害虫特征分析 .....	39
3.8 检测模型子算法 .....	40
3.9 本章小结 .....	41
4 算法分析与测试 .....	42
4.1 实验设计 .....	42
4.2 害虫分割子算法测试 .....	42
4.2.1 原图采集 .....	42
4.2.2 图像的初步分割 .....	43
4.2.3 黄板二值图像的获取 .....	43
4.2.4 去掉体积较大害虫 .....	45
4.3 害虫特征提取子算法测试 .....	45
4.4 害虫计数的结果展示 .....	46
4.5 检测模型计算算法精度的测试与分析 .....	47
4.5.1 算法的常规度量分析 .....	49
4.5.2 算法的相对误差分析 .....	50
4.5.3 算法计数和人工计数的对比分析 .....	51
4.6 蔬菜害虫分类计数算法的初步应用 .....	52
4.7 本章小结 .....	53
5 总结与展望 .....	54
致    谢 .....	56
参    考    文    献 .....	57
附录：攻读硕士学位期间的科研工作 .....	60

# 1 前言

## 1.1 研究背景

在世界范围内，蔬菜作为一种重要的经济作物，发挥着不可替代的作用。近年来我国经济的发展离不开农作物产量的快速增长，其中，国内外蔬菜的贸易是重要的经济支柱之一。近十几年来，国内的蔬菜产量在不断的扩大，现在已经基本满足了国内的需求，使中国人民的膳食更加均衡。特别是加入 WTO 以来，中国种植的蔬菜不仅满足全国人民的需求，而且还远销海外。出口的蔬菜主要销往美国、俄罗斯等国家，并且贸易的总额呈上升的趋势（尤帅，2016）。从 2006 年一直到现在，中国生产蔬菜的产量是世界生产蔬菜产量的一半左右，中国蔬菜的产量占有不可替代的地位（李英娇，2016）。为了更好地提高蔬菜的产量，各种大规模的蔬菜种植基地应运而生。

大规模的蔬菜基地虽然可以为蔬菜的生长提供更好的生长环境，但同时也给害虫提供了有利的生长环境。比如：1980 年的时候，蓟马在华南地区的蔬菜田地普遍出现；1993 年的时候在杭州田地的瓜类或者豆类等蔬菜，害虫蓟马大规模增长；2000 年的山东等地蔬菜害虫大规模爆发，豆类等蔬菜严重危害；2010 年在海南爆发大规模的蔬菜害虫；2015 年，在北京市的辣椒等蔬菜大棚里面，出现了大规模的蓟马害虫，严重伤害蔬菜的生长（京报集团，2015）。如果不尽量减少害虫对蔬菜的侵害，蔬菜的产量将会大大降低，不利于农业的发展。

为了减少害虫带来的不利影响，提高蔬菜的产量，大部分种植户会给蔬菜喷洒化学药剂，也就是农药。自从 1970 年以来，世界各国开始加强对农药的研发，并且获得丰硕的研究成果（马广鹏，2013）。从 1991 年开始到现在，农药的使用总量呈现总体上升的趋势（陈晓明等，2016）。农药的使用确实可以提高蔬菜的产量，但同时也带来了一些健康问题。由于农民使用农药的时候没有严格控制农药的剂量，致使大部分蔬菜残留不少的化学物质，严重危害人类的身体健康。人类进食蔬菜残留的农药，可能会导致肠道疾病、神经衰弱、免疫力下降等症状，严重的更会导致肝硬化和癌症（陈朗，2016）。因此，控制好农药的使用剂量，对人类的健康发展具有重要的作用。

为了能在保证人类身体健康的情况下提高蔬菜的产量，必须对田间蔬菜害虫进行较为精确的统计，根据害虫发生的程度定量喷洒农药，从而既能提高蔬菜的产量，又能保证人类健康。因为农药剂量的多少需要根据害虫爆发的严重程度进行确定，所以害虫的计数问题显得相当重要。如果可以对最近一段时间的害虫进行计数，根据数理

统计和数据挖掘的一些算法，可以有效地预测未来的虫害情况，及早做好防范的准备，降低了虫害爆发的风险。在劳动密集型时代，一直采用人工计数的方法，一开始放一些诱捕装置在蔬菜田地里，定时派遣一些工作人员下田对诱捕装置的害虫进行计数，这种方法在一定程度上达到了害虫计数的目的。但是，人工计数不仅工作量巨大、存在主观因素等缺点，而且计数的结果不能及时的获取和预警，失去了及时预测的价值（陈梅香等，2015）。从 1980 年开始到现在，害虫图像的识别与计数逐步成为了一个国内外的研究热点（陈梅香等，2015）。

图像识别技术在农业上运用的范围比较广，适合场景比较多。因此，通过图像识别的技术进行害虫的计数是比较合理的。通过信息技术的手段，实时监控田间环境变化，实时对田间害虫进行计数，及时对病害严重的地方进行预警，并且根据虫害种类和数量制定相应的措施，合理喷洒农药。这样做不仅仅使得蔬菜的产量会增大，更重要的是使得蔬菜的农药残留成分相对较少，蔬菜质量更加有保证。迄今为止，有人研究过基于视觉的害虫计数算法，但是在田间真实环境下还有一点差距，于是，迫切需要一个基于田间真实环境下黄色诱捕板的快速计数算法。

## 1.2 研究目的和意义

基于对田间害虫自动计数的需要，本文以南方蔬菜重大害虫蓟马、烟粉虱和实蝇为研究对象，提出了一个基于黄色诱捕板的害虫快速计数算法。以黄色诱捕板作为诱捕载体，使用佳能单反相机拍摄黄板所在的区域，获取害虫图片，使用本文的计数算法对蓟马、烟粉虱和实蝇这三种害虫进行识别与计数，为病虫害预测模型的建立提供数据来源。

(1) 在田间环境下拍摄的照片，是现场害虫发生分布的真实反映。本文使用的图像数据是在田间获取的，在实地直接对黄板进行拍照，不用额外的人工去寻找植物上的害虫，获取照片的害虫对象并不单一，所以存在光照不均匀、出现阴影以及其他害虫干扰的情况（周志艳等，2010）。在拍照的时候，以往有的方法所拍摄照片的实际面积相对较少，害虫对象相对单一，相对容易识别。最近几年，国内外也有不少的害虫计数方法。在国外，一些关于害虫识别的方法大部分是把带有害虫的诱捕装置拿回实验室进行恒定条件的拍照，还有一些方法虽然是在田间拍照，但是需要人工寻找害虫生存的叶面进行拍照，害虫对象相对单一。在国内，大部分识别的算法都是需要人工寻找害虫的生存环境进行拍照的，害虫对象比较单一，效率不高。

(2) 对害虫进行计数，有利于农药合理的喷洒和蔬菜的安全。根据害虫计数的密度，判断害虫发生的严重性，对蔬菜进行定量的喷洒，从而保证害虫不会对蔬菜产生较大的影响，减少蔬菜被害虫侵蚀的情况，保证了蔬菜的产量。其次，根据害虫的密度合理地喷洒农药，有利于保护蔬菜的食用安全，最大程度地保证了蔬菜安全。

(3) 有利于害虫预测模型的快速建立。如果采用以前人工的方式对害虫进行计数的话，需要大量的人力来识别黄板图片的害虫类型，然后对各个类型的害虫进行计数，害虫图片数量多的话，就需要大量的人力对害虫进行计数，然后根据害虫的数量及其环境因素建立模型，这个过程耗费时间较多，建模时间相对较长。采用程序计数的方法，可以把害虫的图片收集起来，一次性全部放入程序里面，可以快速对所有图片进行计数，有利于加快害虫预测模型的建立。

所以，运用信息技术计算大田环境下害虫的数量具有深远的意义。本文采用黄色诱捕板的方式对害虫进行诱捕，使用相机进行拍照，把整个黄色诱捕板拍下来，研究对象是照片里面的害虫，对田间害虫进行识别，除了蓟马、烟粉虱等蔬菜害虫之外，还有一些实蝇等其他类型的害虫。

### 1.3 国内外研究现状

近些年来，国内外对蔬菜害虫识别的技术比较多。在国内，沈佐锐（2001）采用 Johannsen 最优熵分割算法对白粉虱进行分割计数，准确率达到 91.99%。邱道尹（2007）使用 3 层 BP 神经网络对金龟子等 13 种害虫进行分类，平均准确率达到 93.5%。Yao (2012) 采用背景差分和支持向量机的方法对水稻害虫进行计数，平均准确率达到了 97.5%。胡雅辉（2011）收集了叶片背面的烟粉虱（*Bemisia tabaci*）图片，采用 Canny 算子边缘检测和背景差分的方法进行烟粉虱的密度统计，准确率达到 90%。韩瑞珍（2013）使用 HSV 分量的分割算法分割对象，根据形态和颜色特征提取了 25 个特征值，然后使用支持向量机的方法识别 6 种害虫，平均准确率达到了 87.4%。张水发（2013）采集了叶面上白粉虱的图片，利用原图和经过离散余弦变换（DCT）后低频信号的灰度图求取差值获得白粉虱轮廓内的像素图像，然后用区域生长法进行填充来获取白粉虱的分割图像，准确度达到 98% 左右。徐向超（2015）采用 KL 变换结合 BP 神经网络的方法对水稻螟虫进行识别，不仅提高了计算机的性能，而且识别率还维持在一个较好的水平。谢成军（2016）采用稀疏编码和金字塔模型的方法对害虫进行识别，并且和传统的方法进行比较，该方法比传统方法拥有更高的识别精度。Sun (2017) 采

用二维傅里叶变换的方法对诱捕板捕捉的甘薯粉虱(sweet potato whiteflies)和西蓝花蓟马(western flower thrips)进行识别计数，取得了不错的效果。

在国外，Gassoumi(2000)使用神经网络算法，对田间害虫进行分类识别，取得了较好的效果，大部分害虫的识别率达到90%。Murakami(2005)对植物叶面上的蓟马害虫(*Thrips tabaci* L)进行识别，使用线性分析的方法对纹理特征进行识别分类。Shariff(2006)对水稻田间的常见的几种害虫进行计数，使用的算法是模糊逻辑，最后效果相对较为显著。Tirelli(2011)做了一个自动监测的害虫预警系统，核心思想是采用背景差分的算法对害虫的爆发进行预警。Fukatsu(2012)对中华稻缘蝽(*Leptocoris chinensis*)采用背景差分的方法进行计数，准确率达到89%左右。Xia(2015)通过分水岭算法和计算YCrCb空间的马氏距离对蚜虫(aphid)、蓟马(thrips)和粉虱(whitefly)进行计数，平均准确率分别为85.6%、82.2%和79.7%。Li(2015)采用多重分形分析的算法对叶面表面的烟粉虱(whitefly)进行识别计数，召回率达到86.9%。Espinoza(2016)做了一个虫害管理系统，结合图像处理算法和前馈神经网络对蓟马(thrip)和烟粉虱(whitefly)进行识别，精度分别达到了92%和96%。

相对来说，国内外研究对蓟马、烟粉虱等蔬菜害虫研究相对来说比较少。到目前为止，主要是通过对诱捕板害虫进行计数和对绿色植物叶面的害虫进行计数。胡雅辉研究团队、张水发研究团队和Murakami研究团队研究了绿叶表面的蔬菜害虫图像，对蓟马或者烟粉虱的识别率相对较好，整体性能也不错，但是识别的害虫类型相对单一，还需要人工翻转每个叶面，不利于将来自动化识别的发展。国内中国农业大学的孙宇瑞团队和国外的Espinoza(2016)研究了关于诱捕板害虫的识别，他们的识别精度也不错，对使用机器视觉识别害虫具有重要的意义，但孙宇瑞等人做的实验素材来源是模拟诱捕板诱捕害虫的场景拍摄的，背景就只有诱捕板，而且只有两种害虫，识别的影响因素相对较少，而Espinoza则是用一个封闭的盒子隔绝光照来对黄板进行拍摄，然后选取特定的区域对害虫进行识别与计数，限制条件相对较多，不利于真实环境下害虫的识别。鉴于以上现状，本文提出了在田间环境下基于黄色诱捕板的害虫计数算法，在自然环境下对害虫进行识别。

#### 1.4 本论文的主要工作

根据最近几年图像处理的期刊论文，在田间病虫害识别领域上运用较为先进的图像处理算法，并且改进相应的步骤，达到识别害虫的目的。本文以自然条件下黄板上

的害虫作为研究对象，对害虫进行识别计数，并且还有如下困难：除了黄板之外存在非黄板区域图像的干扰、黄板上截体害虫的干扰、黄板本身胶体产生的干扰、不同光照强度和阴影面积的干扰，大型害虫的部分身体产生的干扰。本文运用图像处理相关的算法分割出害虫对象，然后提取害虫对象的特征向量，最后根据特征向量进行判别分类，从而达到计数的目的。本文的工作有以下几部分：

(1) 害虫图像数据集的获取。由于害虫图像没有公开的数据集可供使用，所以需要自行设计或者选择合适的诱捕装置，以便获取害虫图像数据集。为了设计合适的诱捕装置，查看了国内外害虫图像识别的文章，咨询农学院的相关专家，并且多次在田间进行实验，最终选择了相机拍摄黄色诱捕板蔬菜害虫的方法。

(2) 分割算法的对比与选取。害虫计数关键的一步是对害虫进行准确的分割，害虫对象分割的不完全对识别计数会产生较大的影响。于是，本文第二章结合害虫图像分析了经典的 10 种分割算法，并用代码逐一展示结果，分析算法的缺陷与不足之处。又根据经典算法的缺陷，本文第三章采用了基于结构化随机森林的分割算法，结合背景去除和干扰害虫的去除子算法，形成一种基于视觉感知的蔬菜害虫计数算法。

(3) 特征提取的对比分析。本文第二章先对传统的特征提取算法进行探究，结合害虫图像分析传统特征提取算法的不足，进而提出本文不规则结构的特征提取算法，最终每个不规则对象可以得到一个 12 维的特征向量。由于特征值之间可能存在冗余，所以本文更进一步地对 12 维的特征向量进行分析，采用 SPSS 对特征向量进行主成分分析，结合实际经验，提取出了 3 个特征，给后续识别提供更精简而有效的特征。

(4) 算法过程的展示与分析。根据算法的设计流程，展示算法每个操作的结果子图，更进一步说明本文算法的可行性。通过对害虫计数的结果进行统计，求得算法的常规度量指标、算法的相对误差和算法与人工的对比数据，进而分析出本文算法性能的优越性。

以往的分割方法大多数都不需要先验知识，直接对图片进行分割，效果并不是很好。本文其中的一个新颖的地方是，借助于结构化随机森林的方法，把美国伯克利大学的 BSDS500 数据库作为训练集，运用在本文的害虫照片上，对害虫照片进行分割，分割效果显著。

轮廓并不是只发生在亮度或者颜色变化剧烈的区域，而且有可能发生在纹理的边界。本文的另外一个亮点是使用了自相似性描述子捕捉害虫图像的一小块图像纹理特征的自相似性。

## 1.5 论文结构

第1章是前言，介绍了害虫计数的历史背景和重要作用，阐述了害虫计数算法的国内外研究现状及其迫切性。根据课题研究需求，提出本文识别计数的算法思想。

第2章主要是对运行在害虫图像上的分割算法、特征提取以及识别技术进行分析，展示以往各种分割算法在害虫图像上的一些效果。

第3章主要是基于黄色诱捕板的蔬菜害虫图像，提出了本文计数算法的相关概念和算法流程。

第4章主要是对本文所提出的算法进行实验测试，并对结果进行分析。

第5章主要是总结本文的内容，提出以后研究的方向。

## 2 基于计算机视觉的害虫图像识别技术探究

害虫图像是田间害虫的一个表示形式，是以数字格式存储的一种图像，该图像可以用矩阵的形式来表示。害虫图像的一般计数流程如图 2.1 所示，主要是后面四个过程。其中，目标分割和特征向量的提取是较为重要的两个部分。

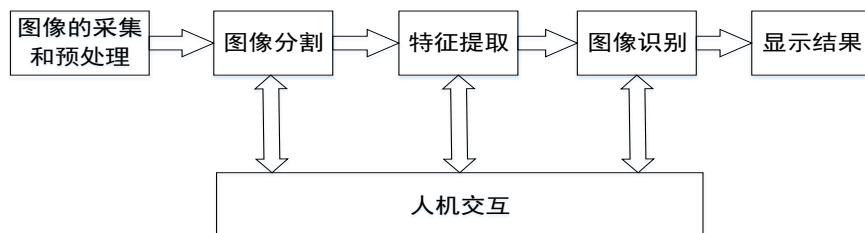


图 2.1 图像处理流程

下面简要介绍害虫图像采集及预处理、图像分割、特征提取和图像识别这四个主要模块的经典算法应用。

### 2.1 害虫图像的采集和预处理

害虫图像的采集技术主要包括两个方面的内容。一个是对作物表面的害虫进行图像的采集和对诱捕装置里面的害虫进行图像的采集。采集植物表面害虫的图片的话，一般害虫种类相对较少，害虫和植物表面之间的色差相对较大，较为容易区分。如果对诱捕装置的害虫进行图像采集的话，基于一般图像视觉是二维的特点，使用诱捕板粘黏害虫比较适合，诱捕板诱捕的害虫图像可以较为完整地记录害虫密度分布。

传统害虫图像的预处理主要是对害虫图像进行图像增强操作。由于获取的图像由于各种原因导致图像可能存在噪声的干扰，使用图像增强的方法可以改善图像的质量，一般的图像增强操作可以从空间域和频率域这两方面考虑。

### 2.2 害虫图像分割的主要技术

为了确定害虫图像中各个害虫对象的边界轮廓，需要检测害虫图像中每个像素及该像素点附近的情况，然后根据某种数学规则或者公式决定该像素点是否在对象之间的边界上，如果符合对象之间边界的特点，则该像素点就是害虫图像中各个对象的边缘点，这些边缘点当然包含有害虫和黄色诱捕板之间的边缘点。害虫图像的边界图也叫边缘图，用两种方式来表达。如果只是判断像素点是否在边缘上，则可以用二值图像来表示边缘图。如果想得到详细的边缘信息，可以使用具有灰度数值的边缘图表示。

通常来说，人看见一张图片最开始发现的是对象的边界信息。所以，图像对象的边界信息对人获取外界信息具有重要意义。边缘是图像中不同对象之间的分界线，是一个对象的轮廓，是图像的灰度值或者幅度值突变的地方。目前一般利用边缘的这些属性对区域或者物体对象进行分割。

本节将从分割的角度对黄色诱捕板的图像进行分割算法的测试和分析，通过分割结果分析各种分割算法的性能，给第三章提出新的分割算法做对比。

### 2.2.1 害虫图像的边缘检测

方向和幅度值是图像边缘的两个属性。像素幅度垂直于轮廓方向的像素变化巨大，顺着轮廓方向变化缓慢。判断像素是否在边缘上的一个方法是计算一阶导数的最大值，一阶导数值最大的地方对应边缘的位置。判断像素是否在边缘上的另一个方法是计算二阶导数的零点，二阶导数零点的位置对应边缘的位置。

由于图像是离散的，所以使用差分代替微分，差分形式的导数可以使用卷积核与原图做运算求得，然后设置相应的规则提取边缘轮廓。不同的模板代表不同的算子，设害虫图像在  $A$  点的值为  $F(i,j)$ ，又点  $A$  附近的像素灰度值如图 2.2 所示。

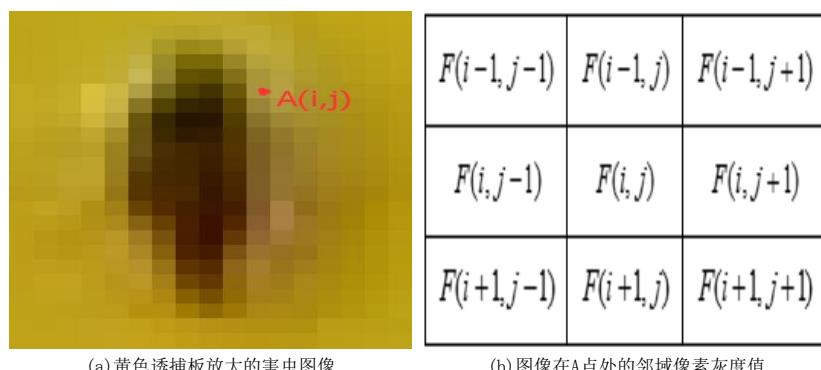


图 2.2 害虫边缘在点  $A(i,j)$  周围的像素

为了更好的理解本节求取偏导数的过程，不妨设模板详细的信息为模板矩阵  $M$ ，该矩阵表示为：

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \quad (2.1)$$

根据模板矩阵和图像像素，本文图像中求边缘检测的通用公式为：

$$G_M(i, j) = m_{11}F(i-1, j-1) + m_{12}F(i-1, j) + m_{13}F(i-1, j+1) + m_{21}F(i, j-1) + m_{22}F(i, j) + m_{23}F(i, j+1) + m_{31}F(i+1, j-1) + m_{32}F(i+1, j) + m_{33}F(i+1, j+1) \quad (2.2)$$

### (1) 梯度算子

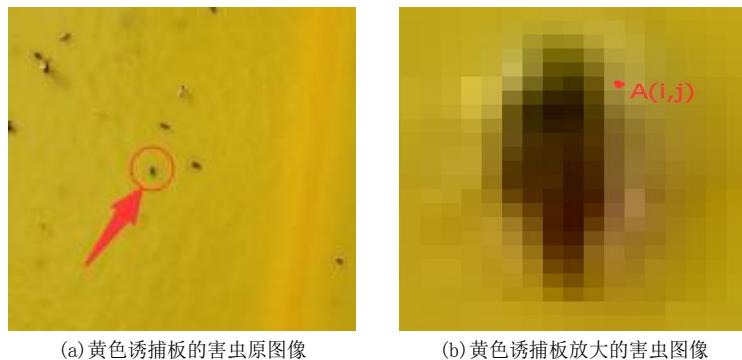


图 2.3 黄板上蓟马的边缘点

由图 2.3 (b) 可以看出, 害虫对象的边缘是一个灰度变化较大的地方。我们常常用两个特征来表示, 一个是灰度的方向, 可以使用梯度向量的方向来表示, 另一个是灰度的变化率, 可以使用梯度向量的幅度值来表示。梯度算子是一阶导数算子。害虫图像  $\mathbf{F}$  在位置  $\mathbf{A}(i,j)$  的梯度为:

$$\nabla \mathbf{F} = \begin{bmatrix} G_x(i,j) \\ G_y(i,j) \end{bmatrix} = \begin{bmatrix} \frac{\partial F(i,j)}{\partial x} \\ \frac{\partial F(i,j)}{\partial y} \end{bmatrix} \quad (2.3)$$

图像  $\mathbf{F}$  在位置  $\mathbf{A}(i,j)$  的幅度值为:

$$\nabla F = mag(\nabla \mathbf{F}) = (G_x^2 + G_y^2)^{\frac{1}{2}} = \left[ \left( \frac{\partial F}{\partial x} \right)^2 + \left( \frac{\partial F}{\partial y} \right)^2 \right]^{\frac{1}{2}} \quad (2.4)$$

上面两个式子分别表示在点  $\mathbf{A}$  的最大变化速率方向和最大变化速率。为了简化计算, 点  $\mathbf{A}$  的幅度值可近似表示为以下公式:

$$M_1 = \max(G_x, G_y) \quad (2.5)$$

$$M_2 = |G_x| + |G_y| \quad (2.6)$$

$$M_3 = G_x^2 + G_y^2 \quad (2.7)$$

这三种式子可以近似代替梯度幅度值的计算, 顺着害虫轮廓方向幅度值相对较小, 垂直于害虫轮廓的幅度值最大, 这两个方向的幅度值分别代表灰度变化缓慢和激烈的情况。我们常常把梯度的幅度值称为“梯度”, 该梯度的方向表示为:

$$\alpha(i, j) = \arctan\left(\frac{G_y(i, j)}{G_x(i, j)}\right) \quad (2.8)$$

①Roberts 算子可以通过图 2.4 所示的两个模板求得。

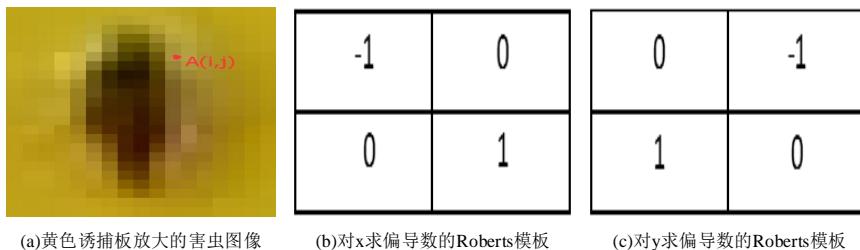


图 2.4 Roberts 模板

由公式 2.2 可知, 代入  $m_{11}=0, m_{12}=0, m_{13}=0, m_{21}=0, m_{22}=-1, m_{23}=0, m_{31}=0, m_{32}=0, m_{33}=1$  可得到  $x$  的偏导数公式, 代入  $m_{11}=0, m_{12}=0, m_{13}=0, m_{21}=0, m_{22}=0, m_{23}=-1, m_{31}=0, m_{32}=1, m_{33}=0$  求得到  $y$  的偏导数公式, 化简公式 2.2 可得:

$$G_x(i, j) = G_{(b)}(i, j) = F(i+1, j+1) - F(i, j) \quad (2.9)$$

$$G_y(i, j) = G_{(c)}(i, j) = F(i+1, j) - F(i, j+1) \quad (2.10)$$

其中,  $G_x(i, j)$  和  $G_y(i, j)$  分别表示该核模板的偏导数。

②Prewitt 算子可以通过图 2.5 所示的两个模板求得。



图 2.5 Prewitt 算子模板

同样, 由公式 2.2 可知, 代入  $m_{11}=-1, m_{12}=-1, m_{13}=-1, m_{21}=0, m_{22}=0, m_{23}=0, m_{31}=1, m_{32}=1, m_{33}=1$  可得到  $x$  的偏导数公式, 代入  $m_{11}=-1, m_{12}=0, m_{13}=1, m_{21}=-1, m_{22}=0, m_{23}=1, m_{31}=-1, m_{32}=0, m_{33}=1$  可得到  $y$  的偏导数公式, 化简公式 2.2 可得:

$$G_x(i, j) = G_{(b)}(i, j) = [F(i+1, j-1) + F(i+1, j) + F(i+1, j+1)] - [F(i-1, j-1) + F(i-1, j) + F(i-1, j+1)] \quad (2.11)$$

$$G_y(i, j) = G_{(c)}(i, j) = [F(i-1, j+1) + F(i, j+1) + F(i+1, j+1)] - [F(i-1, j-1) + F(i, j-1) + F(i+1, j-1)] \quad (2.12)$$

其中,  $G_x(i,j)$ 和 $G_y(i,j)$ 分别表示该核模板的偏导数。

③Sobel 算子可以通过图 2.6 所示的两个模板求得。



图 2.6 Sobel 算子模板

本算子有个特殊的地方, 就是在中心系数上加上一个系数为 2 的权重, 这是为了增加中心点的重要性。同样的方法, 由公式 2.2 可知, 代入  $m_{11}=-1, m_{12}=-2, m_{13}=-1, m_{21}=0, m_{22}=0, m_{23}=0, m_{31}=1, m_{32}=2, m_{33}=1$  可得到  $x$  的偏导数公式, 代入  $m_{11}=-1, m_{12}=0, m_{13}=1, m_{21}=-2, m_{22}=0, m_{23}=2, m_{31}=-1, m_{32}=0, m_{33}=1$  可得到  $y$  的偏导数公式, 化简式 2.2 可得:

$$G_x(i,j) = G_{(b)}(i,j) = [F(i+1,j-1) + 2F(i+1,j) + F(i+1,j+1)] - [F(i-1,j-1) + 2F(i-1,j) + F(i-1,j+1)] \quad (2.13)$$

$$G_y(i,j) = G_{(c)}(i,j) = [F(i-1,j+1) + 2F(i,j+1) + F(i+1,j+1)] - [F(i-1,j-1) + 2F(i,j-1) + F(i+1,j-1)] \quad (2.14)$$

其中,  $G_x(i,j)$ 和 $G_y(i,j)$ 分别表示该核模板的偏导数。

通过以上 3 种梯度算子, 我们可以计算出图像在点 **A** 处的偏导数。利用偏导数和式 2.15 可以计算图像在点 **A** 处的梯度值, 然后和一个合适的阈值做比较, 符合阈值范围的是边缘点, 否则不是, 最后可以计算出整张害虫图像的边缘点集合。

$$\nabla f = mag(\nabla f) = (G_x^2 + G_y^2)^{1/2} \quad (2.15)$$

## (2) Kirsch 算子

Kirsch 边缘算子使用 8 个模板来计算害虫图像的幅度值。害虫图像中的点分别与这些模板进行计算, 幅度值就是选取模板运算结果的最大值。图 2.7 表示计算 **A** 点的运算模板。



图 2.7 害虫图像的 Kirsch 模板算子

以图 2.7 (a) 的点  $\mathbf{A}$  为中心, 分别和 (b)、(c)、(d)、(e)、(f)、(g)、(h)、(i) 这 8 个模板做卷积运算, 卷积运算的结果同样也是运用公式 2.2 计算得到的, 方法和梯度算子类似, 只不过 Kirsch 仅需计算一次偏导即可, 分别得到的值为  $\mathbf{M}_1$ 、 $\mathbf{M}_2$ 、 $\mathbf{M}_3$ 、 $\mathbf{M}_4$ 、 $\mathbf{M}_5$ 、 $\mathbf{M}_6$ 、 $\mathbf{M}_7$ 、 $\mathbf{M}_8$ , 则 Kirsch 算子的幅度值表示为:

$$G(x, y) = \max(|\mathbf{M}_1| + |\mathbf{M}_2| + |\mathbf{M}_3| + |\mathbf{M}_4| + |\mathbf{M}_5| + |\mathbf{M}_6| + |\mathbf{M}_7| + |\mathbf{M}_8|) \quad (2.16)$$

### (3) 拉普拉斯算子

拉普拉斯算子也是采用模板卷积的方式表示, 但运算的结果是只有大小, 没有方向。它具有线性、移动不变的特性。如果一幅图像使用拉普拉斯算子进行卷积运算, 那么运算结果的图像具有零平均灰度的性质。如图 2.8 (b) 表示点对  $\mathbf{A}(i,j)$  点进行运算的模板。

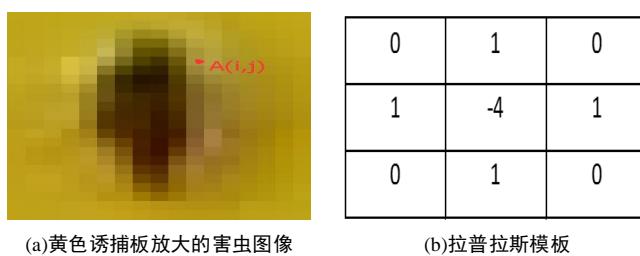


图 2.8 拉普拉斯模板算子一

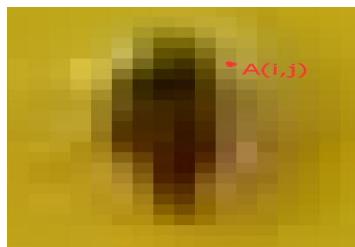
同理, 由公式 2.2 可知, 代入  $m_{11}=0$ ,  $m_{12}=1$ ,  $m_{13}=0$ ,  $m_{21}=1$ ,  $m_{22}=-4$ ,  $m_{23}=1$ ,  $m_{31}=0$ ,

$m_{32}=1, m_{33}=0$  可得到图像在 A 点的二阶导数公式为:

$$\nabla^2 F(x, y) = F_{(b)}(x, y) = F(x+1, y) + F(x-1, y) + F(x, y+1) + F(x, y-1) - 4F(x, y) \quad (2.17)$$

#### (4) 高斯-拉普拉斯算子

为了达到更好的效果,一般情况下需要先对图像做平滑处理,然后使用拉普拉斯算子计算。在实际应用中,一般采用高斯滤波进行平滑处理。把高斯平滑和拉普拉斯算子相结合的方法叫做高斯-拉普拉斯 (Laplacian of Gaussian, LoG) 算子。图 2.9 (b) 使用模板求得图像在点 A 的 LoG 算子。



(a) 黄色诱捕板放大的害虫图像

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

(b) 近似的 LoG 模板

图 2.9 近似的 LoG 模板算子模板

其计算方法和上述方法类似,只不过需要把公式 2.1 变成五维的矩阵,重新计算公式 2.2 可以求得。由下面的公式 2.18 表示 LoG 算子模板的运算结果

$$\nabla^2 h(x, y) = \frac{1}{\pi\sigma^4} \left[ \frac{x^2 + y^2}{2\sigma^2} - 2 \right] e^{-\frac{x^2+y^2}{2\sigma^2}} = \frac{x^2 + y^2 - 4\sigma^2}{\sigma^4} \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.18)$$

#### 2.2.2 害虫图像的阈值分割

由于黄色诱捕板的害虫基本没有重叠的现象,可以使用封闭且连通的轮廓定义,这种方法叫做阈值分割,这个方法计算相对简单。由此可知,边界就是至少有一个邻点不属于害虫的对象的点集。本节主要结合害虫图像介绍常用的阈值分割技术。常用的分割算法如下所示:

##### (1) 全局阈值

通过某种方式选择一个阈值常数  $T$ ,把害虫图像的灰度值和常数  $T$  做比较,把害虫图像分成目标和背景这两大类。该方法的数学表达式可以表示为:

$$g(x, y) = \begin{cases} 1 & F(x, y) > T \\ 0 & F(x, y) \leq T \end{cases} \quad (2.19)$$

全局阈值主要包含以下几种方法:

①人工选择法：首先通过人眼观察害虫图像的像素差异，应用人的先验知识对害虫图像的直方图进行分析，根据人的知识找到合适的阈值，如果分割效果不好，可以不断调整阈值找到合适的阈值。

②迭代式阈值选择法：首先选择害虫图像灰度等级的中间值  $T_0$  作为初始阈值，然后分别计算小于这个阈值的区域灰度均值  $\mu_1$  和大于这个阈值的区域的灰度均值  $\mu_2$ ，再计算出这两个均值的平均值作为新的阈值  $T_{i+1}$ ，如果  $T_{i+1}$  和  $T_i$  的差值小于 1，则  $T_{i+1}$  是该害虫图像的阈值，否则以  $T_{i+1}$  为初始阈值重新计算新的阈值。

③直方图技术选择法：如图 2.10 所示，分别是害虫图像单通道图像的灰度直方图。利用图 2.10 所示的灰度直方图来选取阈值，选择两个山峰之间的谷作为阈值的分界线，该阈值代表害虫对象和非害虫对象之间的边缘像素点。根据直方图的信息，灰度图的阈值为 120，R 分量的阈值为 140，G 分量的阈值为 140，B 分量的阈值为 150。

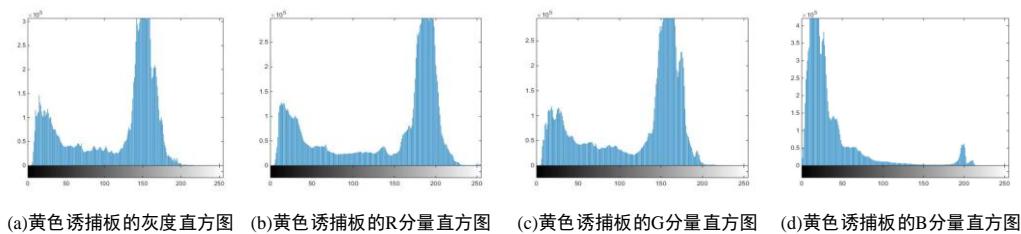


图 2.10 灰度图以及 RGB 各分量的灰度直方图

④大津法（Ostu 算法）：结合灰度直方图和最小二乘法得出来的最佳分割阈值，使得害虫对象和背景之间的方差最大。

### （2）自适应阈值法

在田间环境下，由于光照等自然因素的影响，黄板背景的灰度值可能并不是常数，害虫和黄板的对比度也可能发生变化。一个阈值有时候在一个区域表现良好，而在另外一个区域效果却很差。为了尽可能解决这种情况，我们采用了一种随害虫图像位置缓慢变化的函数值作为图片的阈值，这个阈值叫做自适应阈值。

### （3）分水岭算法

由于害虫图像里面的害虫可能会出现相互接触的现象，全局阈值是无法解决这个问题的，如图 2.11 所示。



(a) 黄色诱捕板的害虫原图像

(b) 黄色诱捕板放大的害虫图像

图 2.11 害虫粘连情况

为了解决这个问题，这里我们使用分水岭算法进行分割，分水岭算法是把图片看成一个地貌，如图 2.12 所示，灰度值看成是山体高度，盆地就是我们要识别的害虫对象，而盆地之间的脊线就是分水岭的分割边缘。

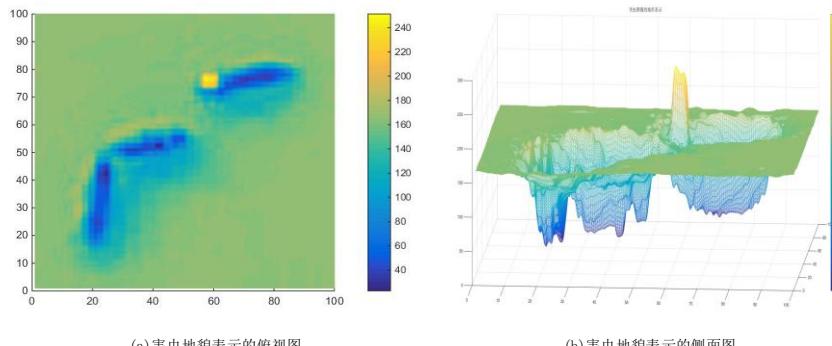


图 2.12 害虫图像的地貌表示

## 2.3 害虫图像特征提取技术

分割完图像之后，下一步就是对分割对象进行特征提取，本小节介绍了相关的特征提取的概念，并结合害虫图像介绍了一般的特征提取方法。

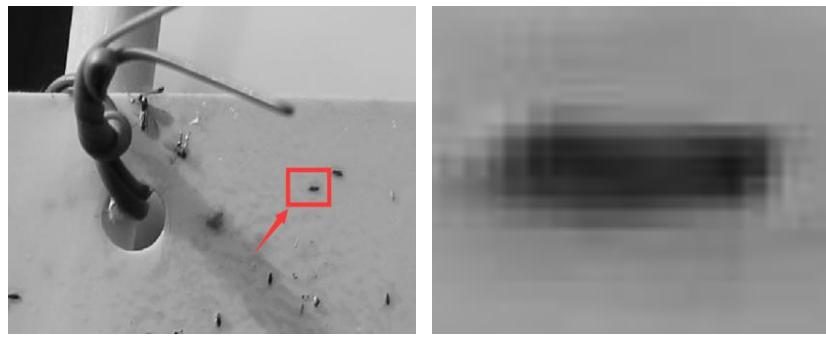
### 2.3.1 害虫的直方图特征

害虫图像的颜色特征包括我们需要识别的害虫和一些相关的背景，展示了图像的一种全局的特征。如果我们将整张图片提取颜色特征的话，这个图像特征包括很多对象的信息，包括害虫、黄板甚至是远处的建筑物。因此我们只计算每个分割对象矩形区域的颜色特征，保证颜色特征大部分只包含单个对象。颜色特征的差异是相应像素各自的贡献产生的。颜色特征在图像领域应用广泛，一般可以使用直方图代表它。又因为方向、尺寸和视角对它的影响相对较弱，所以具有较高稳定性。

由于一般的彩色图像具有三个通道，需要把彩色通道变成单通道，然后才对灰度图像进行直方图计算。我们以单个分割出来的图像块作为特征提取的对象，设单个图像块的像素总数为  $n$ ，该图像块的灰度等级数为 256，灰度值为  $k$  的像素总共有  $n_k$  个，那么灰度直方图可以表示为：

$$h_k = \frac{n_k}{n}, k = 0, 1, 2, \dots, L-1 \quad (2.20)$$

根据公式，我们可以对图像块的计算就像对图 2.13 (b) 右边子图的计算一样，统计整个矩形的颜色特征值。



(a) 黄色诱捕板的灰度图

(b) 诱捕板放大的害虫图像

图 2.13 害虫对象的颜色特征提取

又由于害虫图像块的灰度直方图不是一个直观的数值，因此需要进一步计算直方图的数值特征，平均值、方差、能量和熵的计算公式分别为：

$$\bar{F} = \sum_{k=0}^{L-1} k h_k \quad (2.21)$$

$$\sigma_F^2 = \sum_{k=0}^{L-1} (k - \bar{F})^2 h_k \quad (2.22)$$

$$F_n = \sum_{k=0}^{L-1} (h_k)^2 \quad (2.23)$$

$$F_E = -\sum_{k=0}^{L-1} h_k \log_2 h_k \quad (2.24)$$

### 2.3.2 害虫的纹理特征

图像的纹理特征可以得到图像像素值在空间上的分布模式，代表对象的粗糙度、颗粒度等。纹理特征需要在像素点较多的区域进行统计计算，不是基于像素点的特征，这点和颜色特征不同。纹理特征具有抗噪能力和旋转不变性等特点。现在我们采用灰

度共生矩阵来描述纹理特征，这个方法能够精确反应纹理的粗糙程度和重复方向。

同样，灰度共生矩阵的计算需要把彩色图像变成单通道的形式，然后才进行灰度共生矩阵的计算，如图 2.14 所示。

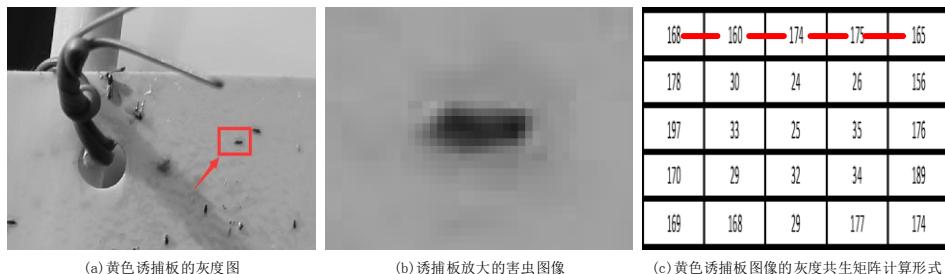


图 2.14 计算灰度共生矩阵的灰度图

图 2.14 (b) 代表图像的像素值，为了简单明了，采用了近似的 5 行 5 列的像素矩阵代替。有关研究表明  $d$  的取值越小越好，本文采用  $d = 0, \theta = 0$  的参数进行计算，计算的灰度共生矩阵有  $255 \times 255$  的规模，内存消耗巨大，于是就对 256 个灰度值进行压缩，压缩为 16 个灰度等级，然后再进行计算灰度共生矩阵。由于需要数值的形式代表纹理特征，于是就对纹理特征进行二次统计，最后可以求得能量、对比度、熵、均匀度和相关。

### 2.3.3 害虫的边界特征

边界特征主要是通过对对象的外部特征来描述该对象。主要有链码、多边形近似和标记图等方法。基于害虫图像提取的特征需要数值特征，于是便采用周长和离心率作为害虫图像的边界特征。如图 2.15 所示，周长和离心率分别为椭圆周长和长短轴之比。

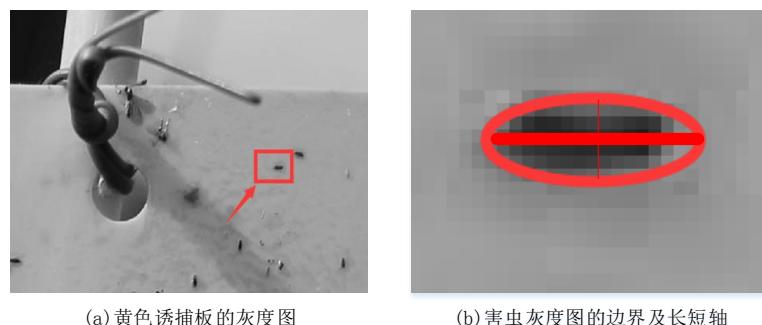


图 2.15 害虫的边界特征

### 2.3.4 害虫的区域特征

区域特征是图像对象区域的直观表现，一般是对分割后二值化图像进行计算，计算得到位置、方向、面积等参数，由于本文的害虫位置和方向都是不确定的，所以采用区域面积来表示对象的区域特征，如图 2.16 (b) 所示的白色区域就是害虫的面积。

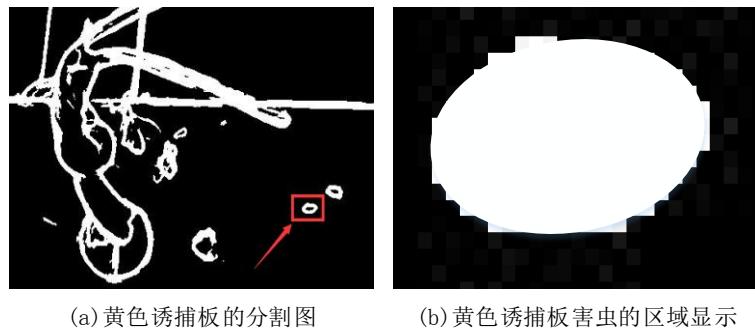


图 2.16 害虫图像的区域特征

## 2.4 害虫图像识别主要算法

根据近几年论文参考可得，当提取特征结束之后，田间图像特征识别的主要算法包括支持向量机（SVM）和人工神经网络。

### 2.4.1 支持向量机

支持向量机是最受欢迎的分类技术之一。支持向量机依靠统计学的理论基础，在很多领域的性能表现优异，在解决实际问题中备受欢迎。支持向量机不仅可以应用于低维的数据，还可以应用于高维的数据，进而避免了维灾难问题的产生。这种技术具有一个突出的亮点，那就是该技术使用训练实例的一个子集来代表决策边界，这个子集就被称作支持向量机。

### 2.4.2 人工神经网络

神经网络是由大量权值和函数组成的数学模型，该数学模型极像生物神经网络，可以模拟人脑解决实际问题。神经网络和人脑一样，通过不断的获取外界信息，来调节自身的模型结构，最后形成一个可以解决实际问题的模型。（Tan *et al.*, 2011）

## 2.5 基于黄色诱捕板的分割算法分析

本小节根据 2.3 节的分割技术理论，结合黄色诱捕板的图像，展示各算法的分割结果。根据分割结果，对各个分割算法进行分析，发现其中的优劣。由于分割算法一般对单通道图片进行处理，所以需要把原图转化成单通道的图片。为了完全展示各个算法的性能，分别对每种单通道图像进行计算和分析，如图 2.17 (a) -2.17 (b) 所示。

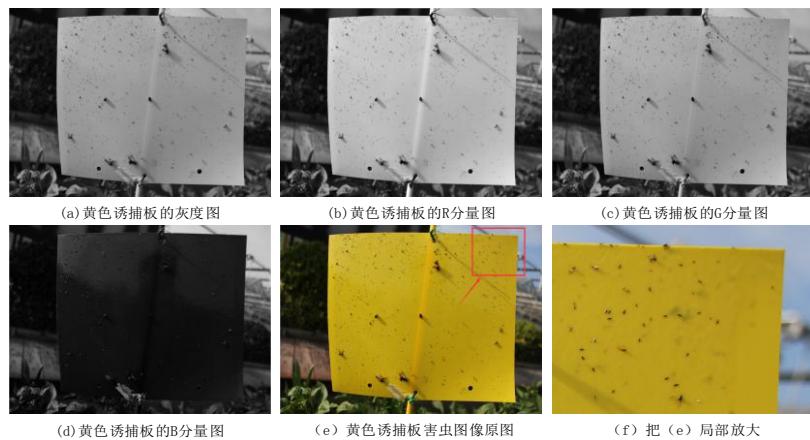


图 2.17 原图的各种单通道图片

鉴于篇幅原因，本节所展示的算法大部分采用原图中黄板和背景交界的位置作为结果展示，如图 2.17 (f) 所示；全局阈值分割中的所有算法采用原图的分割结果作为结果展示，如图 2.17 (e) 所示。

### 2.5.1 基于黄色诱捕板害虫图像的边缘检测结果分析

#### (1) 基于黄色诱捕板害虫图像的梯度算子分析

##### ① Roberts 算子

根据 2.3.1 小节 Roberts 算子的理论基础，对黄色诱捕板的害虫图像使用 Roberts 检测算法，得到如图 2.18 所示的结果。

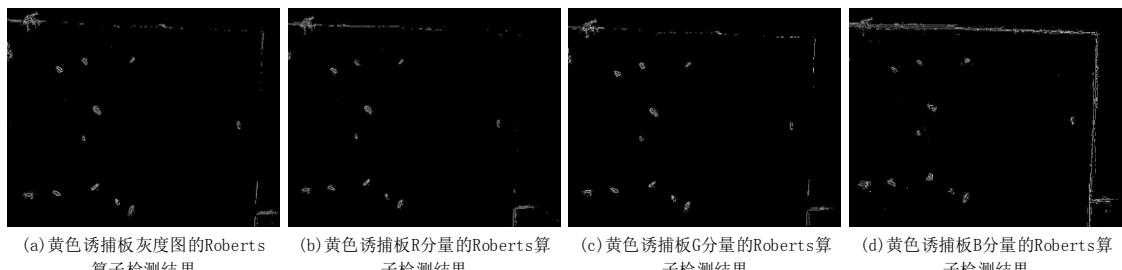


图 2.18 Roberts 模板在虫害图片的效果

##### ② Prewitt 算子

根据 2.3.1 小节 Prewitt 算子的理论基础，使用黄色诱捕板的害虫图像作为图像素材，得到如图 2.19 所示的结果。

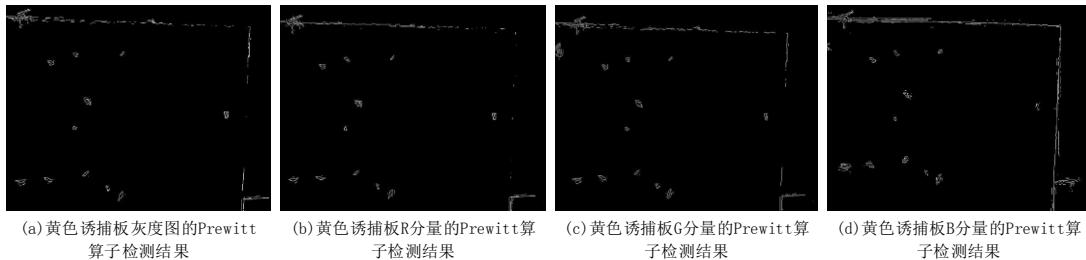


图 2.19 Prewitt 算子模板在害虫图片的效果

### ③ Sobel 算子

根据 2.3.1 小节 Sobel 算子的理论基础, 使用黄色诱捕板的害虫图像作为图像素材, 得到如图 2.20 所示的结果。

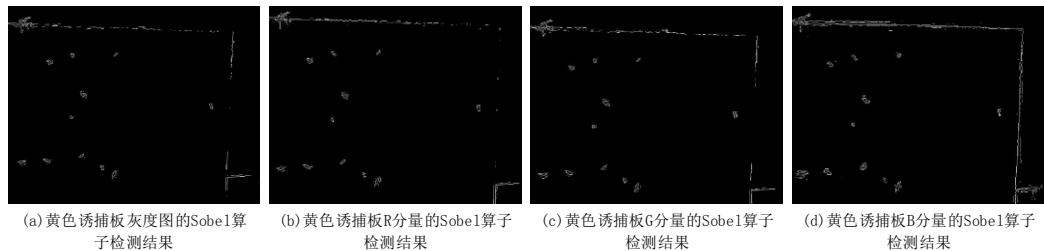


图 2.20 Sobel 模板算子在虫害图片的效果

### (2) 基于黄色诱捕板害虫图像的 Kirsch 算子分析

根据 2.3.1 小节 Kirsch 算子的理论基础, 使用黄色诱捕板的害虫图像作为图像素材, 得到如图 2.21 所示的结果。

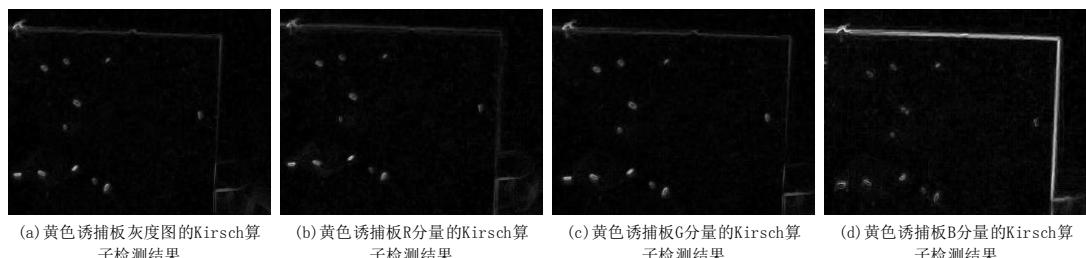


图 2.21 Kirsch 模板算子在虫害图片的效果

### (3) 基于黄色诱捕板害虫图像的 LoG 算子分析

根据 2.3.1 小节 LoG 算子的理论基础, 使用黄色诱捕板的害虫图像作为图像素材, 得到如图 2.22 所示的结果。

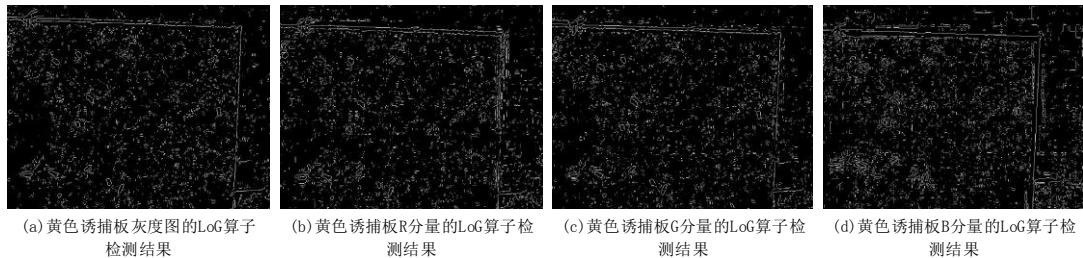


图 2.22 LoG 算子模板在虫害图片的效果

在本论文的害虫图像中，除了 B 通道的图像以外，以上算子对黄板其他的单通道图像分割不完全，不能把黄板的轮廓分割出来，不利于后续黄板与背景的分离。对于 B 分量的结果，虽然大多数可以分割出黄板的轮廓，但是黄板内部的分割效果不太理想，不利于害虫对象的识别。对结果进行分析，可以知道图像处理的边缘并不一定和真实边缘对应，真实情况下三维的对象，经过设备变成二维的图像，三维到二维本身就会丢失一些信息，而且还会有关照或者其他噪声的影响，是的算法检测出来的边缘轮廓不一定是真实对象的轮廓。

### 2.5.2 基于黄色诱捕板害虫图像的阈值分割

#### (1) 全局阈值

##### ①迭代式阈值法分析

根据 2.3.2 小节迭代式阈值分割法的理论基础，使用黄色诱捕板的害虫图像作为图像素材，得到如图 2.23 所示的结果。

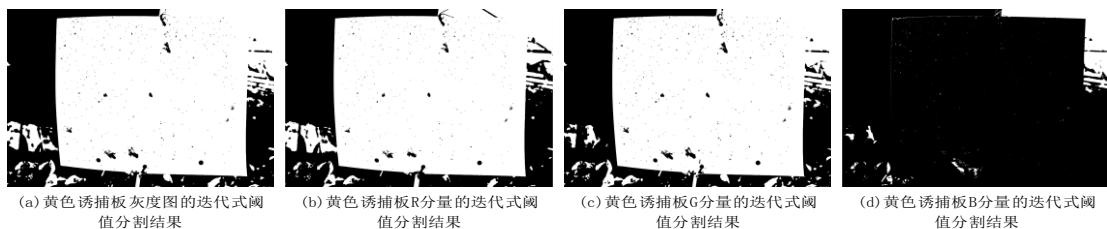


图 2.23 迭代式阈值分割结果

##### ②直方图技术选择法

根据 2.3.2 小节直方图技术选择法的理论基础，使用黄色诱捕板的害虫图像作为图像素材，得到如图 2.24 所示的结果。

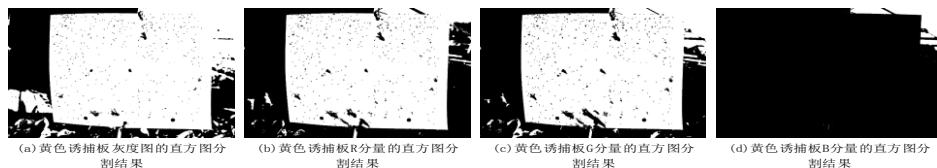


图 2.24 直方图技术分割效果

### ③大津法

根据 2.3.2 小节大津法的理论基础，使用黄色诱捕板的害虫图像作为图像素材，得到如图 2.25 所示的结果。

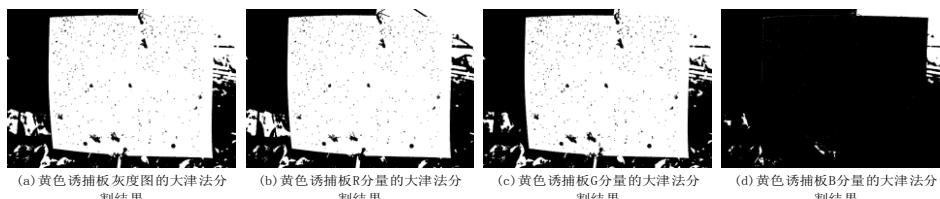


图 2.25 大津法分割效果

## (2) 自适应阈值法

根据 2.3.2 小节自适应阈值法的理论基础，使用黄色诱捕板的害虫图像作为图像素材，得到如图 2.26 所示的结果。

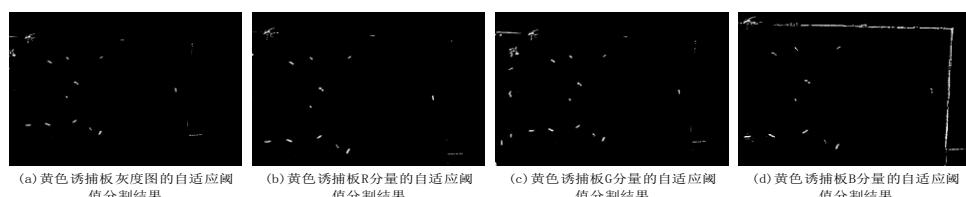


图 2.26 自适应阈值法分割效果

### (3) 分水岭算法

根据 2.3.2 小节分水岭算法的理论基础，使用黄色诱捕板的害虫图像作为图像素材，得到如图 2.27 所示的结果。

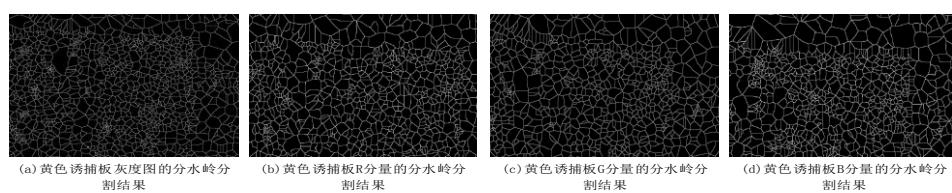


图 2.27 分水岭算法分割效果

从上述效果可以得出，全局阈值和大津法的分割结果会有一部分非黄板区域和黄板连成一片，对黄板的分割效果不好。自适应阈值法分割黄板，会出现部分黄板边缘的轮廓被去掉的现象。分水岭算法可以大致描述出照片的轮廓，但是产生了非常多的干扰轮廓。

## 2.6 基于计算机视觉的害虫图像识别小结

本章根据图像处理的一般流程，先简要说明害虫图像常用的采集方法和预处理，然后结合本文害虫图像阐述经典常用的图像分割算法理论；其次，根据害虫对象介绍常用的特征向量提取算法和模式识别算法；最后，通过各个算法的代码编写，对各个算法运行在害虫图像上的结果进行展示和分析，为设计最佳的害虫图像分割识别算法提供借鉴。

从前面的分析可以看出，分割算法对图像 B 分量图的分割效果较好，其中梯度算子和 Kirsch 算子分割效果相对较好。Kirsch 算子和梯度算子虽然可以比较好的分割 B 分量图像的黄板边缘，但害虫的边缘却不能较好地分割出来。以往的分割算法都是根据图像本身的梯度、像素等特征进行分割的，同样的对象在不同环境下会有不同的分割结果，这是因为受到了背景图像及其光照等因素的影响，使得图像在不同场景中呈现不同的分割结果。田间害虫的识别目前还处于在理想环境下的阶段，在理想环境下精度可能不错，但在具有各种因素影响的真实田间环境下，识别效果却有待考究。

从上面害虫图像经典算法的展示结果对比可以看出，以往分割算法都是基于颜色、梯度等底层特征来对图像进行分割，同一个害虫对象在不同环境下分割的效果不全相同。此外，以往的特征提取对于害虫对象来说，会把害虫对象周围的无关像素都计算在内，对特征值产生干扰。

当前随机森林、支持向量机等有监督的机器学习算法能较好地处理不同场景下相同对象的分割，弥补传统方法的不足，例如 Lim 等人（2013）和 Zitnick 等人（2014）在图像分割和图像跟踪领域展示优良的分割效果。基于这些具有监督性质的机器学习算法能较好解决当前害虫图像处理方面的不足。

### 3 基于黄色诱捕板的害虫快速计数算法

针对大田环境下害虫图像识别技术的迫切需求和存在的不足，本文利用结构化随机森林和不规则特征提取技术，探索了一种基于黄色诱捕板的害虫图像计数算法（Vegetable Pest Counting Algorithm Based on Visual Perception, VPCA-VP）。下面主要从害虫图像的分割、害虫对象的特征提取和害虫对象的检测计数这三个方面进行介绍。首先详细地讲述每个模块涉及的算法原理，然后根据算法原理把各个过程串联起来形成本文的害虫计数算法。

#### 3.1 VPCA-VP 算法总体描述

VPCA-VP 算法主要包括害虫图像的预分割、背景去除、干扰害虫的去除、不规则特征的提取和检测计数这五大模块。总体流程如图 3.1 所示。

首先，由于照片环境复杂，使用原图的颜色空间、分辨率和方向梯度构成 13 个通道图像的向量和这 13 个通道图像的相似性描述子作为随机森林的输入向量，把对应原图的人工素描的轮廓进行向量化，对该向量进行降维和主成分分析可以得到维数较少的输出向量，把输入向量和输出向量作为随机森林的训练集，得到可以分割害虫图像的随机森林，使用随机森林初步对害虫图像进行预分割；然后对预分割的二值图像使用图像填充算法和区域选择算法获得只有黄板区域的分割图像，目的是为了减少黄板后面建筑物等无关环境的干扰。其次，为了减少大型害虫的部分肢体的干扰，使用凸包拟合算法完全去掉大型害虫；然后对黄色诱捕板的害虫进行特征提取，提取出只有害虫部分的特征，使用 SPSS 对提取出来的特征进行分析，结合实际选取出主要的三个特征；最后，对相应的特征进行阈值的判别，符合所有条件的判定为一类害虫，从而对该类害虫进行计数。

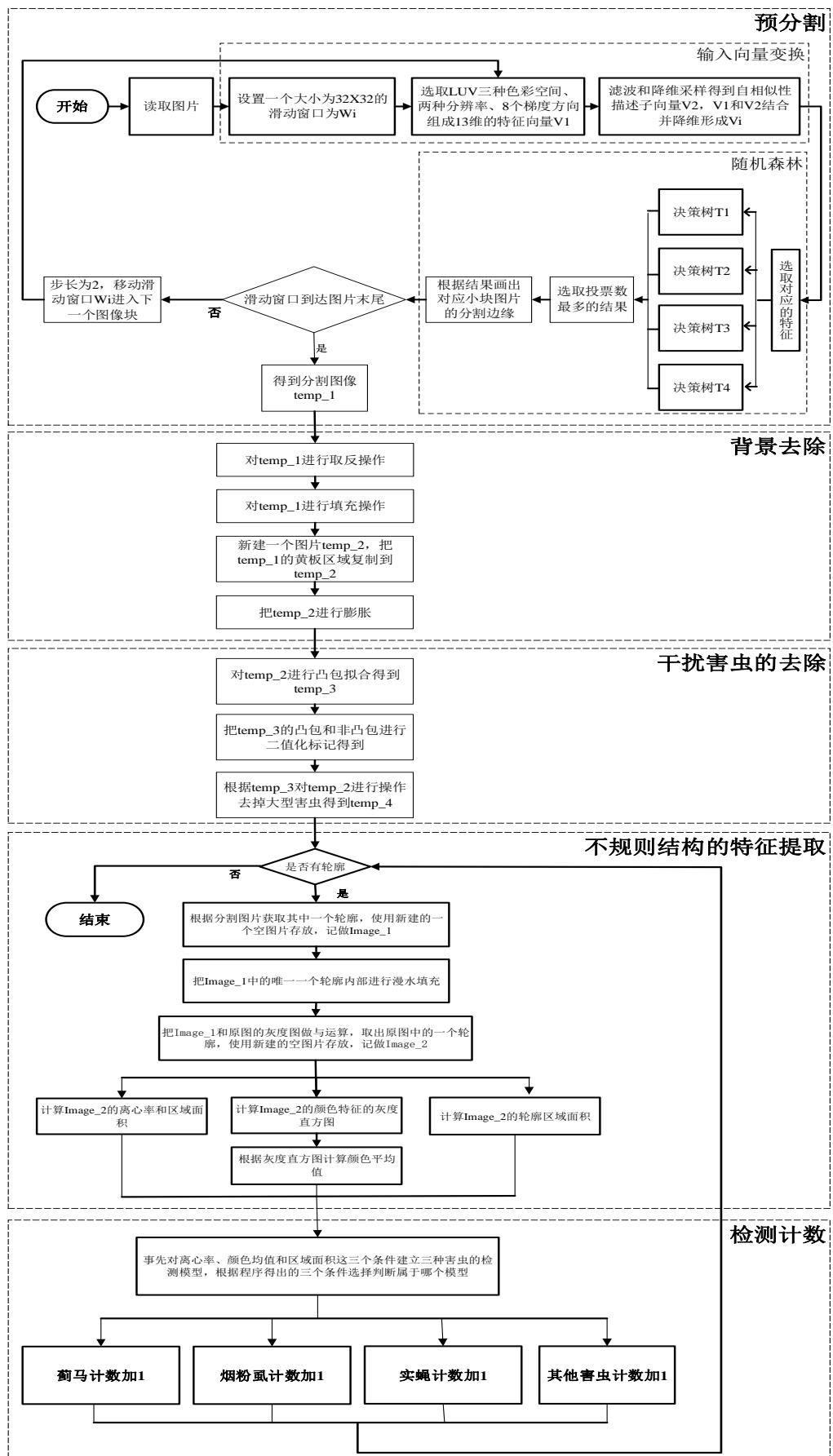


图 3.1 VPCA-VP 算法的总体流程

### 3.2 基于黄色诱捕板的害虫图像采集

国内外有很多关于图像的数据集，但是在农业害虫方面，没有统一的害虫数据集可以使用，鉴于这种情况，本文使用自己采集的数据集作为算法的素材。采集数据集的时候遇到各种的情况，使用了多种方案，最后采用了黄板诱捕的方法进行采集。本文图片拍摄的地方是广州市天河区柯木塱南路 28-30 号（经纬度：113.412671,23.190265），使用佳能相机进行拍摄，如图 3.2。

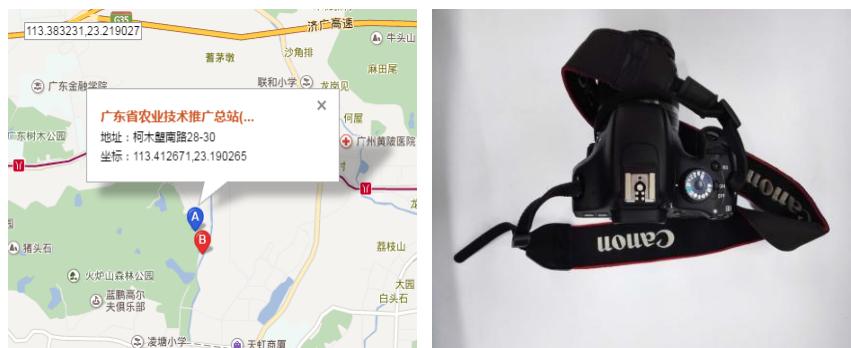


图 3.2 地理位置和相机

因为害虫对黄色的物体比较敏感，所以采用黄色诱捕板作为诱捕源对害虫进行诱捕，本文采用辣椒菜地作为实验环境，在辣椒苗期开始使用黄板诱捕，更具作为生长及时调整悬挂高度，保持色板高于作物 15-30cm，每亩悬挂 30-50 张黄板，黄板大小为 24.5cm\*19.5cm，如图 3.3 所示。



图 3.3 实验现场

### 3.3 害虫图片的预处理

如果图像质量不够清晰，就需要对图像进行复原。在图像成像或者运输过程中，有可能受到某种原因导致图像变得不清晰或者受损，为了尽可能把图像还原到原来的状态，一般对图像进行图像复原处理。由于本文采集的相片是使用单反拍摄的照片，基本不存在图像退化的情况，所以只需要简单的图像增强就可以。

### 3.4 基于结构化随机森林的害虫图像分割子算法

#### 3.4.1 算法原理

##### (1) 决策树

决策树是数据挖掘经典的十大算法之一，在分类方面有重要的作用。首先在训练集上按照某种规则把特征属性排序成一个属性序列，这个属性序列使得决策树的平均深度最小。生成了一颗具有分类功能的决策树之后，把新数据输入决策树进行分析，决策树的本质是通过一系列属性判断对数据集进行分类。树状结构的决策树，每个叶节点对应一个类别的标签，非叶节点对应数据的一种属性，数据的属性在某个非叶结点上进行判别的时候，根据该属性的取值把数据集分成若干类的数据子集。由此可以知道，构造决策树的核心就是如何一步步地选取合适的属性序列，属性序列一旦生成，决策树模型便形成。根据决策树生成规则的不同，可以分成不同的决策树算法。其中，ID3 是较为经典的一个算法，本文就是采用 ID3 算法实现的（Tan *et al.*, 2011）。

信息熵是 ID3 算法的核心内容。ID3 算法是选择当前样本集具有最大信息增益的属性作为本节点的属性节点。样本集的划分是通过最大信息增益的属性划分的，通过该属性的不同取值把样本集分成对应的子集作为叶节点，同时把该属性作为划分子集的根节点，把该根节点连接到上一个节点上。ID3 采用信息论的信息增益作为评价标准，信息增益越大，后续子集的纯净度就越大。ID3 算法选择的属性每次都把样本集拆分为最纯的几个子集，从而得到深度较浅的决策树。

不妨设  $s$  个数据样本组成集合  $S$ ，属性可以取  $m$  个不同的值： $C_i(i=1,2,\dots,m)$ ，类  $C_i$  的样本数为  $s_i$ 。则对于一个样本，信息熵总和为：

$$I(s_1, s_2, \dots, s_m) = -\sum_{i=1}^m \frac{s_i}{s} \log_2 \left( \frac{s_i}{s} \right) \quad (3.1)$$

设  $\{a_1, a_2, \dots, a_k\}$  为  $k$  个不同值，这些值是一个属性  $A$  的不同取值，对应  $k$  个不同的取值，将集合  $S$  划分为子集  $\{S_1, S_2, \dots, S_k\}$ ， $S$  中属性  $A$  取值为  $a_j$  的子集为  $S_j$ 。若选择属性  $A$

为决策树结点，则划分的各个子集作为新的叶结点从集合  $S$  结点生长出来。设子集  $S_j$  中类别为  $C_i$  的样本数为  $s_{ij}$ ，则属性  $A$  划分的信息熵值为公式 3.2 所示。

$$E(A) = \sum_{j=1}^k \frac{s_{1j} + s_{2j} + \dots + s_{mj}}{s} I(s_{1j}, s_{2j}, \dots, s_{mj}) \quad (3.2)$$

其中， $I(s_{1j}, s_{2j}, \dots, s_{mj}) = -\sum_{i=1}^m P_{ij} \log_2(P_{ij})$ ， $P_{ij} = \frac{s_{ij}}{s_{1j} + s_{2j} + \dots + s_{mj}}$  是子集  $S_j$  里面类别是  $C_i$  的样本概率。

最后，属性  $A$  划分样本集  $S$  之后，求得的信息增益（Gain）为公式 3.3 所示。

$$Gain(A) = I(s_1, s_2, \dots, s_m) - E(A) \quad (3.3)$$

由公式可知， $E(A)$  的值和  $Gain(A)$  成反比， $Gain(A)$  随着  $E(A)$  的减小而增大，说明选择的属性  $A$  对于分类提供了更大的信息量，选择  $A$  之后的样本集纯度更高。属性  $A$  有  $k$  个不同的取值，该结点就有  $k$  个不同的分支，去掉已经选择的属性，通过递归调用上述过程，生成其他属性的节点作为上一个节点的子节点和分支，最终完成整棵决策树的生成。**ID3** 是决策树中一种较为经典的学习算法，核心技术就是在生成属性序列过程中采用信息增益的方法。信息增益作为属性序列先后顺序的判别标准，选择的属性应该尽可能获得最大的信息增益值，使得后续节点的样本集纯度尽可能小，这样使得树的平均深度尽可能小，从而提高了分类的速度。**ID3** 算法过程如下：

- ①根据当前的样本集合，求出每个属性的信息增益；
- ②把信息增益最大的属性标记为当前的属性节点，根据该属性的取值不同把数据集划分成不同的几个子集；
- ③如果子集中只有一个属性值，则这个属性节点的分支就是对应的几个叶子节点的类别标签，然后返回调用的地方；否则使用子样本递归调用本过程。

## (2) 主成分分析 (PCA)

当输入信号是一个多维向量的时候，直接计算会消耗比较多的时间和空间，而且并不是向量里面所有的参数都是重要的，为了更快地对信号进行运算，对输入向量进行降维操作，从输入信号中提取到主要的信息，这种方法叫做主成分分析。这是使用主成分来描述原来高维信号的一种方法。首先对输入的信号进行计算，得到协方差矩阵，然后计算协方差矩阵的特征向量和特征值，把特征值按照大小进行排序，然后选

择出几个特征值较大的特征向量，最后把输入信号投影到选择出来的特征向量上，进而构造出低维的主信号。

### (3) 随机森林

随机森林是一种集成学习的算法。它使用随机选取的数据或者属性训练每棵子树，形成一棵大的随机森林，然后使用该随机森林对数据进行计算，最终得到的结果是结合每棵树的投票得到的。也就是说，尽管每棵树都很好地学习了训练数据，但是各个树之间仍有很大不同，为了消除这些不同，我们把这些树放在一起平均。在随机森林的构建过程中，有两个方面的构建方法，一个是数据集的随机选取，数据集的选取是原始数据集有放回的随机抽取；另一个是特征值的随机选取，从候选特征中随机选取固定数量的特征作为决策树的分裂特征（Tan *et al.*, 2011）。

### (4) CIE-LUV 色彩空间

通过对 CIE XYZ 空间进行常规的变换，可以得到 CIE-LUV 色彩空间。CIE-LUV 色彩空间具有视觉统一性的特点。这个空间有三个色度，其中， $L^*$ 代表物体亮度， $u^*$  和  $v^*$  是色度。 $u^*$  和  $v^*$  的取值是在 -100 到 100 之间，而亮度  $L$  的取值在 0 和 100 之间。

### (5) 自相似性描述子

Shechtman (2007) 在CVPR的论文里面面上提出了一个新的描述子，这个描述子作为一种局部的特征表示形式，其性能在一些复杂环境的图片上表现出色。自相似性描述子不仅具有丰富的图像模式和统一的表达方式，而且还可以处理边缘不清楚的物体，在不同场景下都能得到良好的分割效果。

自相似性描述子算法过程：

- ①把彩色图片按照特定的颜色空间选取一个通道的图片，作为一个处理对象
- ②选取一个通道的图片，使用滑动窗口的形式，按照一定的步长（例如：本文步长为2），每次选取一个图片块，这个图片块的大小是 $5 \times 5$ 大小的网格。
- ③分别计算25个网格元素到黄色框框的颜色差，每个网格的颜色值参考如图3.4所示。

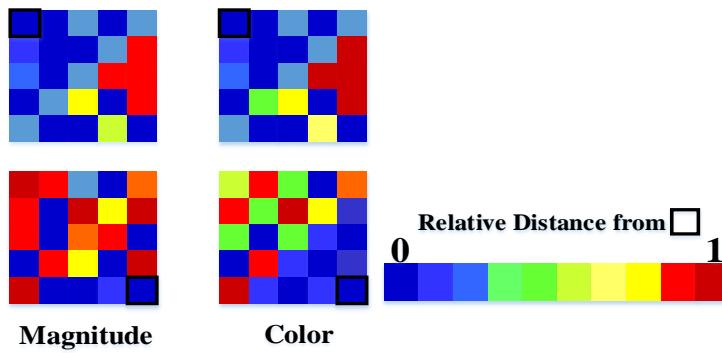


图3.4 像素点相对黑色方框像素的距离

### 3.4.2 算法过程

本文采用有监督的分割方法,训练使用的数据集是 BSDS500 数据集(Martin *et al.*, 2001),然后把训练完成的模型使用在田间病虫害的图片上。

#### (1) 样本和标签的结构化

首先,把数据集的图片进行分块,实验证明可得,小图片的大小在 $32 \times 32$ 这个大小比较合适。对于每块是 $32 \times 32$ 的小图片,这个小图片是作为随机森林的输入的样本结构;每个小图片对应一个掩码图片,该掩码图片对应每块小图片正中间的 $16 \times 16$ 图片,该掩码图片是二值图片,0 表示没有边缘的像素点,1 表示有边缘的像素点,掩码图片代表对应样本结构输出的标签结构(Lim *et al.*, 2013)。

#### (2) 样本结构的向量表示形式

对于结构化的样本,必须换成向量的表示才能进一步计算,样本的特征向量分为两个部分,第一部分是通道特征,另一部分是相似性特征。用  $I_1$  表示 $32 \times 32$ 的图像块,提取  $I_1$  的 CIE-LUV 的 3 个色彩空间、2 种梯度大小(原始分辨率和原始分辨率的一半)(Lowe *et al.*, 2004)和 8 个方向(每个梯度大小有 4 个方向)(Dalal *et al.*, 2005),每个像素总共 13 维的特征向量表示通道特征,  $I_1$  则可以用 $32 \times 32 \times 13 = 13312$ 维特征向量表示,使用平滑滤波器对  $I_1$  进行滤波,然后再使用 $2 \times 2$ 重新降维采样得到 $16 \times 16$ 的图片  $I_2$ ,再使用 $8 \times 8$ 的平滑滤波器对  $I_2$  进行滤波,然后降维采样得到 $5 \times 5$ 的图片  $I_3$ 。计算  $I_3$  中任意两个点自相似性描述子,使用 300 维的特征向量表示相似性特征。把  $I_2$  的通道特征向量和  $I_3$  的相似性特征向量结合起来(Dollár *et al.*, 2009),两部分的特征向量加起来总共有 $(16 \times 16 + 300) \times 13 = 7228$ 维的特征向量。总的来说,一块 $32 \times 32$ 的图片块可以用 7228 维的特征向量来表示(Lim *et al.*, 2013),执行流程如算法 1 所示。

---

#### 算法 1：样本标签向量化算法

---

输入： $32 \times 32$  的图像块  $I$

输出：样本向量

1: GenerateSampleVector( $I$ )

2: 把  $I$  分成 LUV 三个色彩空间，两种分辨率和 8 个方向梯度，共 13 张  $32 \times 32$  的图片；

3: 把  $I$  进行滤波和降维采样，得到  $16 \times 16$  的  $I_2$ ，计算通道特征，得到 3328 维的向量；

4: 把  $I_2$  进行滤波和降维采样，得到  $5 \times 5$  的  $I_3$ ，计算相似性特征，得到 3900 维的向量；

5: 把  $I_2$  生成的通道特征向量和  $I_3$  生成的相似性特征向量合成 7228 维的样本向量；

6: return SampleVector

---

#### (3) 标签结构（素描符号）的向量表示形式

因为标签结构各不相同，需要对标签结构进行聚类，这就要求标签结构具有度量性。为了使标签结构具有度量性，本文把标签结构的小图片表示成向量的形式。

$16 \times 16$  的掩码图片有 256 个像素点，每个像素点之间俩俩组队，由于点对之间不能重复，去掉重复点对之后，总共有 32640 个特征的二值向量。每个点对用 1 表示连通，0 表示不连通，设这个点对的向量空间是  $z$ 。通过定义我们可以把标签结构过渡到空间向量  $z$  上，但是空间向量  $z$  的维数太长了，计算效率低下。为了提高空间向量  $z$  的计算效率，我们先进行第一次的降维，从中得出 256 维向量，然后使用 PCA 进行降维计算，得到一个  $k=2$  维的空间向量，由此可以得到两类的输出标签 (Lim *et al.*, 2013)，执行流程如算法 2 所示。

---

#### 算法 2：标签结构向量化的算法框架

---

输入： $16 \times 16$  的标签块图像  $I$

输出：标签向量

1: GetLabelVector( $I$ )

2: 记录  $I$  的连通性，把轮廓标记为“1”，非轮廓标记为“0”，得到 32640 维二值向量

3: 把二值向量进行降维得到 256 维的向量，然后再进行 PCA，得到两类的标签向量

4: return LabelVector

---

#### (4) 训练集的选取

本文的分割算法是基于随机森林的分割算法，需要大量的图片和手动标签进行训练。本文采用美国伯克利大学的 BSDS500 图像数据库 (Martin *et al.*, 2001) 进行训练，这个图像数据库是为图像分割和边缘检测提供了一个共享的图形库，这个图形数据库

包括 12000 个人工分割和 1000 个 Corel 数据库的图片，如图 3.5 所示。

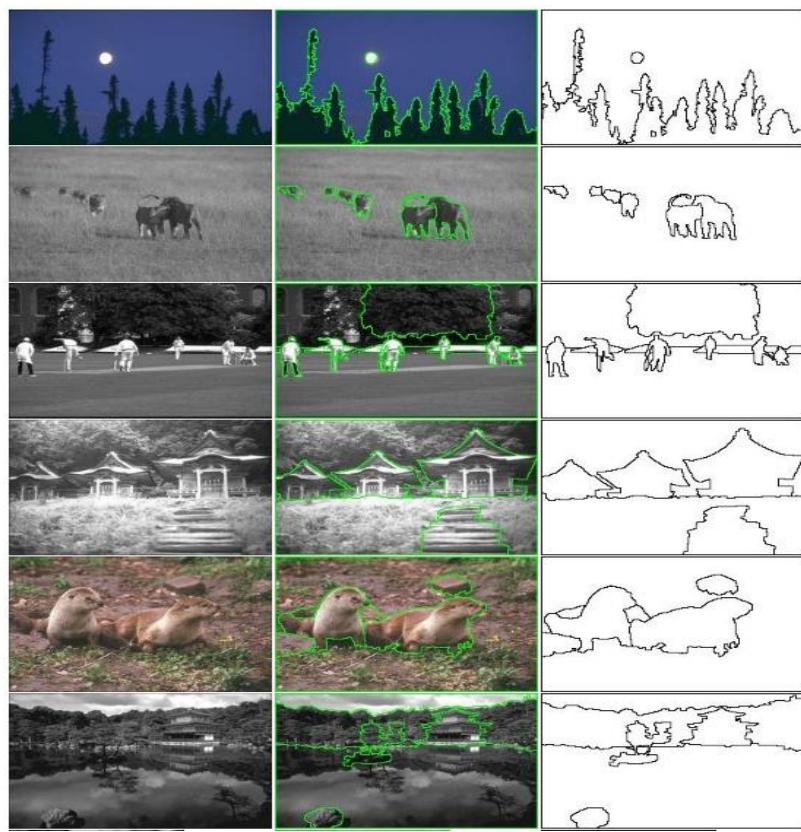


图 3.5 分割训练数据库

### (5) 训练随机森林

经过测试可得，随机森林的子树个数  $T=4$  效果比较好，随机森林算法的执行流程如算法3所示。

---

算法3：随机森林生成算法

---

输入：训练集  $D = \{X, Y\}$ ， $X$  代表样本的属性集， $Y$  是对应样本的标签集

输出：随机森林

```
1: GrowRandomTree(  $D$  )  
2: for t to |T| do  
3:   随机选取固定数量的属性集合集合  $X_{sub}$ ，结合标签集  $Y$  形成训练集  $D_t$   
4:   训练随机森林的单棵决策树: GrowDecidedTree(  $D_t$  )  
5: end for  
6: return random forest
```

---

由于随机森林是由决策树集成的，所以需要对决策树进行训练。决策树 (Dollár et

*al.*, 2013; Dollár *et al.*, 2014) 的训练过程就是分离函数的确定过程, 决策树算法细节如算法4所示。

---

算法4: 决策树训练算法框架

---

输入:训练集 $D=\{X, Y\}$ ,  $X$ 代表样本的属性集,  $Y$ 是对应样本的标签集

输出:随机森林的单棵决策树

```
1:   GrowDecidedTree(  $D_t$  )  
2:   如果样本集合  $D_t$  属于同一类标签或者 $|D_t| = N \leq N_0$ , 则  
3:       leaf=createNode()  
4:       leaf.label=Classify(  $D_t$  )  
5:       return leaf  
6:   else  
7:       选取所有单个属性形成属性集合 $\Gamma_{sub}(D_t) \in \Gamma(D_t)$   
8:       for  $i = 1$  to  $|\Gamma_{sub}(D_t)|$  do  
9:            $\Delta I = I(D_i) - \sum_{k \in \{L, R\}} \frac{|D_i^k|}{|D_i|} I(D_i^k)$   
10:      end for  
11:      找出节点分类器的最优参数 $\theta_i$   
12:      设置左节点 $D_i^L$ 和右节点 $D_i^R$ 空集  
13:      for  $t = 1$  to  $N$  do  
14:          if  $h(x_t, \theta_i) == 1$  then  
15:               $D_i^L = D_i^L \cup \{x_t, y_t\}$   
16:          else  
17:               $D_i^R = D_i^R \cup \{x_t, y_t\}$   
18:      end for  
19:      leftN= GrowDecidedTree (  $D_i^L$  )  
20:      rightN= GrowDecidedTree (  $D_i^R$  )  
21:      如果叶节点有超过一个标签, 则通过计算每个标签到同个叶节点的其他标签  
22:      的距离之和, 选出距离最小的作为该叶节点的标签。  
23:      返回 $\theta_i$ 作为返回节点
```

---

## (6) 测试结果

由于本文训练集采用的图形数据库内容丰富，虽然没有涉及所有领域的图片，但是对于二值图像的边缘图像而言，基本包含了所有的素描符号，也就是说包含了所有的二值图像的基本元素。如图 3.6 所示。

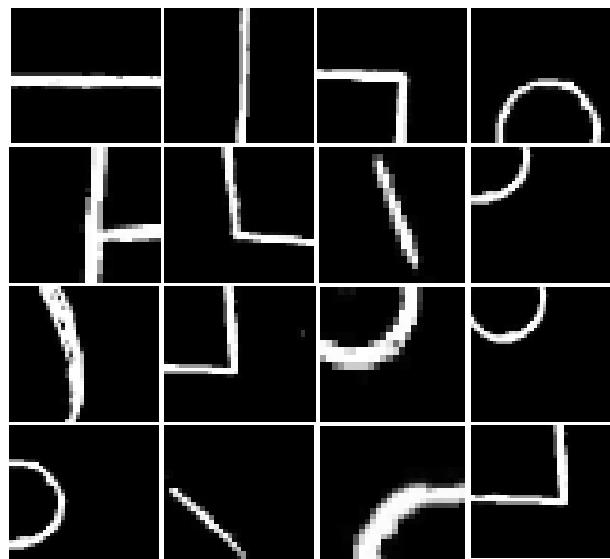


图 3.6 初步分割结果

训练完成之后，不需要重新进行特定的训练即可推广到其他任何领域的图像分割。例如害虫图像的分割如图 3.7 所示。



图 3.7 害虫图像分割显示

### 3.5 背景去除

直接提取出来的黄板区域是不包括黄板边缘的像素，实际实验中发现很多害虫集中在黄板边缘上，所以需要对黄板区域进行微量膨胀来补充损失的黄板边缘信息。设结构元素 $\mathbf{B}$ 对图像 $\mathbf{A}$ 的膨胀记做 $\mathbf{A} \oplus \mathbf{B}$ ，定义为：

$$\mathbf{A} \oplus \mathbf{B} = \{x : \hat{\mathbf{B}}_x \cap \mathbf{A} \neq \emptyset\} \quad (3.4)$$

由于原图是在省农推总站拍摄的，是在大田环境下采集的，包含的背景比较多。因此，本文算法先除去黄板以外的其他背景，这些背景包括一些植物和建筑物，进而得到黄板范围内的图像，也就是获取只有黄板的二值图像。具体的算法流程见算法 5。

---

#### 算法5：背景去除算法

---

输入：预分割图像

输出：背景去除后的图像

1:  $\text{getDelBackground}(\mathbf{I})$

2: 把图像  $\mathbf{I}$  做取反操作，得到图像  $\mathbf{I}_1$

3: 对  $\mathbf{I}_1$  进行填充操作，凸显黄板区域，获得图像  $\mathbf{I}_2$

4: 新建立一个空白图像，选取图像  $\mathbf{I}_2$  的黄板放在空白图像上，形成图像  $\mathbf{I}_3$

5: 对  $\mathbf{I}_3$  进行膨胀操作，弥补边界的损失，得到图像  $\mathbf{I}_4$

6:  $\text{return } \mathbf{I}_4$

---

经过上述算法之后，可以获得只有黄色诱捕板的分割图像，除去了建筑物、天空等复杂背景的干扰，结果如图 3.8 所示。

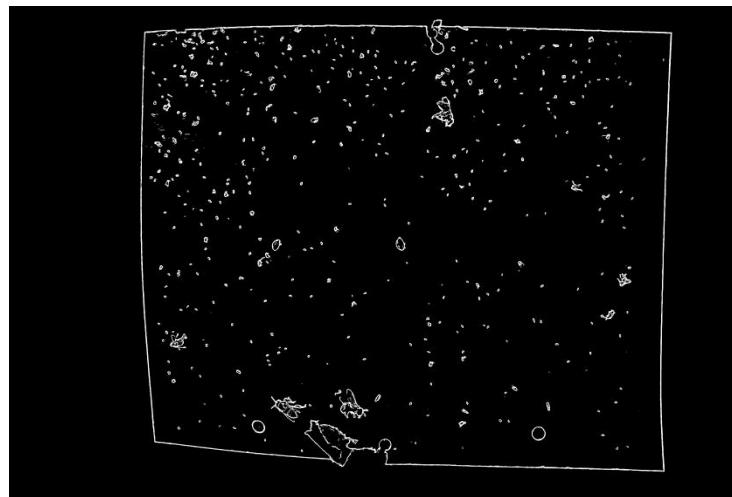


图 3.8 只有黄板的分割图

## 3.6 干扰害虫的去除

本文干扰害虫的去除涉及到掩模图像  $M$  对源图  $S$  的处理, 设结果图像为  $R$ , 则:

$$R(x, y) = \begin{cases} S(x, y) & (M(x, y) == 0) \\ 0 & (M(x, y) \neq 0) \end{cases} \quad (3.5)$$

干扰害虫表示害虫区域面积不在本文所识别害虫区间范围内的其他害虫, 即该类害虫的面积不属于本文识别的三种害虫, 具体害虫的参数模型见本章第 6 节。由于识别的时候, 有可能会把体积较大害虫的部分肢体当成了本文需要识别的害虫。因此, 本文先去除干扰的害虫, 以提高后面害虫识别的精度, 干扰害虫的去除见算法 6。

---

### 算法6: 干扰害虫去除算法

---

输入: 已经去除背景的分割图像

输出: 已经去除干扰害虫的分割图像

- 1: getDelOtherPest( $I$ )
  - 2: 对图像  $I$  进行凸包拟合, 得到只有大型害虫的凸包拟合图像  $I_1$
  - 3: 对图像  $I_1$  中除了凸包区域外其他区域漫水填充成灰色, 获得图像  $I_2$
  - 4: 对图像  $I_2$  的进行二分操作, 形成图像  $I_3$
  - 5: 根据  $I_3$  对  $I$  操作,  $I_3$  为黑色区域则对应保留  $I$  中原图, 否则变成黑色, 得到图像  $I_4$
  - 6: return  $I_4$
- 

## 3.7 害虫特征提取子算法

### 3.7.1 害虫颜色特征的提取

本文采用直方图特征的方法对害虫对象进行颜色特征的提取。首先提取图片的灰度直方图, 然后根据灰度直方图计算相应的特征。传统的直方图特征提取的对象是矩形形状的图片。如果采用矩形形状的图片进行直方图特征提取的话, 就会不可避免地包含对象以外的颜色干扰, 降低了颜色特征的代表性。本文使用一种新的算法对不规则图片进行直方图特征的提取, 使得提取出来的直方图特征只有害虫对象的特征, 排除了其他颜色特征的干扰, 具体步骤如算法 7 所示。

---

#### 算法7：颜色特征提取算法

---

输入：单个轮廓的分割图像

输出：颜色特征向量

1: GetColorVector(  $I, I_{ori}$  )

2: 对图像  $I$  的轮廓使用漫水填充算法得到图像  $I_2$

3: 把  $I_2$  与原图的灰度图做与运算，取出只有单个对象的灰度图  $I_3$

4: 对  $I_3$  图像进行颜色直方图的统计

5: 二次统计计算颜色特征向量

6: return colorVector

---

同时，通过以上的算法步骤，可以得到图 3.9 所示的流程图。

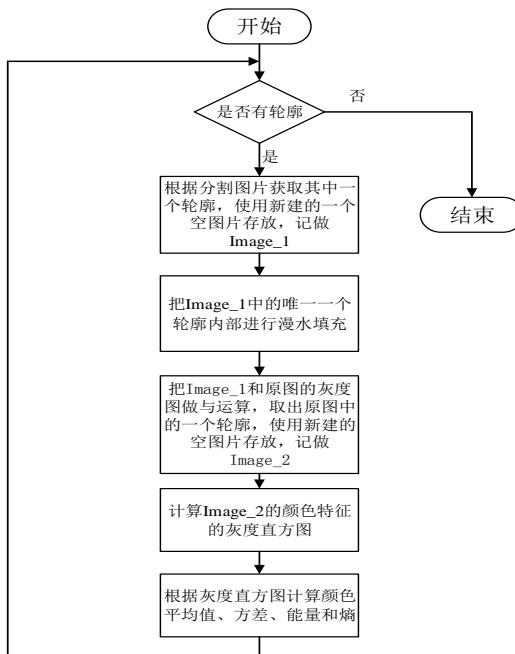


图 3.9 颜色特征提取的流程图

#### 3.7.2 害虫纹理特征的提取

本文采用灰度共生矩阵的形式描述图像的纹理特征，然后使用灰度共生矩阵的二次统计量表示纹理特征。以往计算灰度共生矩阵的对象是矩形形状的图片。同样，如果使用矩形形状的图片计算灰度共生矩阵的话，也会包含对象以外的纹理干扰，降低了纹理特征的代表性。在此，本文也同样对纹理的计算进行了一些改造，使得改造后的特征提取算法适应于不规则特征对象的纹理特征提取，排除无关纹理的干扰，具体步骤如算法 8 所示。

---

#### 算法8：纹理特征提取算法

---

输入：单个轮廓的分割图像

输出：纹理特征向量

- 1: GetTextureVector(  $I, I_{ori}$  )
  - 2: 对图像  $I$  的轮廓使用漫水填充算法得到图像  $I_2$
  - 3: 把  $I_2$  与原图的灰度图做与运算，取出只有单个对象的灰度图  $I_3$
  - 4: 计算  $I_3$  图像的灰度共生矩阵
  - 5: 二次统计计算纹理特征向量
  - 6: return textureVector
- 

同时，通过以上的算法步骤，也可以得到图 3.10 所示的流程图。

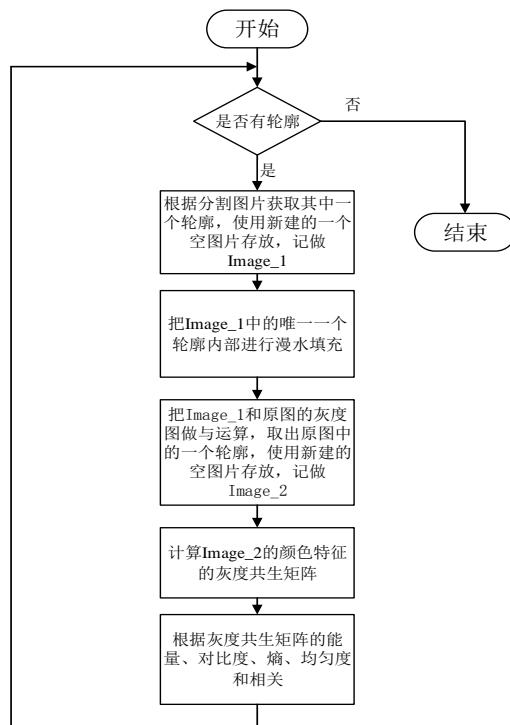


图 3.10 纹理特征提取的流程图

### 3.7.3 害虫边界特征的提取

由于本文获得的照片是通过相机拍摄获取的，每张照片的拍摄距离各不相同，如果想要提取边界特征，使用相对比例的特征更加合适，绝对特征在一个范围可以代表一个对象。所以，对于边界特征来说，选取离心率 $e$ （长轴 $c$ 和短轴 $a$ 的比值）作为边界特征比较合适。

离心率：使用椭圆拟合对害虫对象进行拟合，这个椭圆的长轴线段作为边界的长轴，这个椭圆的短轴线段作为边界短轴。离心率就是长轴和短轴的比值，即：

$$e = \frac{c}{a} \quad (3.6)$$

### 3.7.4 害虫区域特征的提取

本文采用区域面积来表示害虫图像的区域特征，在拍摄照片的距离和角度都不确定的情况下，虽然绝对的区域特征并不能代表一种害虫的区域特征，但是在一定范围内是可以代表该害虫的区域特征，所以本文的害虫区域特征采用区域面积来代表。

区域特征表达的是一个对象边界及其区域内的面积。首先把一副图像进行灰度化，然后对灰度化的图像进行二值化，目标对象的灰度值为1，其他无关背景的灰度值为0。对于目标对象来说，它的面积为：

$$S = \sum_{(x,y) \in R} 1 \quad (3.7)$$

其中， $R$  表示目标对象区域的像素， $(x,y) \in R$  表示像素属于目标对象。

### 3.7.5 害虫特征分析

根据提取出来的12个特征，使用SPSS对特征进行主成分分析，初步取得了主要的3个特征，如图3.11所示。

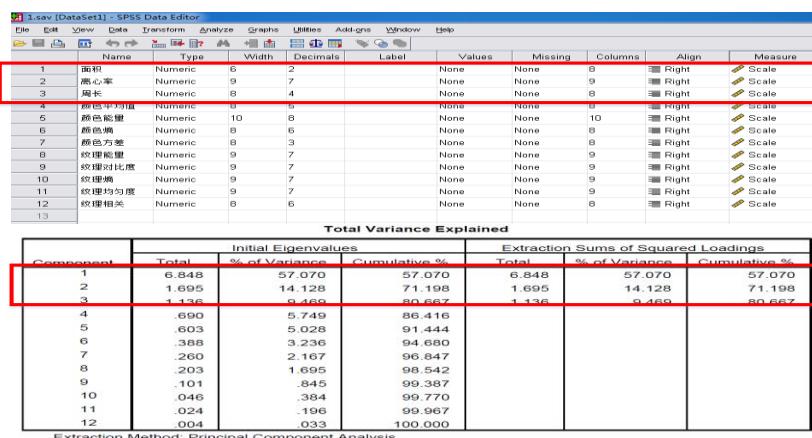


图 3.11 特征分析结果

由于本文计数的害虫是蓟马、烟粉虱和实蝇，根据主成分分析可得，害虫图像所采用的特征向量可以用面积、周长和离心率来表示，在实际代码运行过程中，发现计算轮廓周长效果并不明显，所以去除了周长这个特征，又考虑为了提高精度，本文就再加上了排名第四的颜色平均值，最后选择的特征是面积、离心率和颜色平均值。

### 3.8 检测模型子算法

对于本文中黄色诱捕板的蓟马、烟粉虱和实蝇，其各自的特征主要由颜色特征、纹理特征等图像特征来表示。根据生物学的知识，可以得到以下三种害虫的特征：

- (1) 蓟马：除去触角之外，整体呈现椭圆形，平均体长约 0.5-2 毫米，成虫呈现黄色、棕色和黑色；
- (2) 烟粉虱：整体呈现椭圆形，长约 0.8-3 毫米，虫体呈现淡黄白色和白色；
- (3) 实蝇：头部呈现圆球形，长约 7-9 毫米，宽约 3-4 毫米，虫体呈现黄褐色，翅膀长约 6-8 毫米；

设面积为  $x$ 、离心率为  $y$ 、颜色平均值为  $z$ ，取得了害虫对象的特征之后，根据害虫的特征分别设置相应的阈值，若满足条件  $S_0 < x < S_m$ 、 $e_0 < y < e_m$  和  $avh_0 < z < avh_m$ ，则识别为相应的害虫，从而对该类害虫进行计数。由于害虫图片的颜色直方图的计算时间相对较长，可以作为最后阈值判定的条件，从而加快运算速率。根据农学专家的意见和实验室环境下测试的数据可得，各种害虫的取值范围如表 3.1 所示。

表 3.1 检测模型的阈值参数

	$S_0$	$S_m$	$e_0$	$e_m$	$avh_0$	$avh_m$
蓟马	5	90	0	0.8	0	100
烟粉虱	5	150	0.2	1	100	256
实蝇	19000	50000	0	1	0	150

不妨设特征向量  $\mathbf{m} = (x, y, z)$ ，检测第  $i$  个特征向量时蓟马数量为  $a_i$ 、烟粉虱数量为  $b_i$ 、实蝇数量为  $c_i$ 。由实验可知，田间所有害虫图像的面积区间为 [0, 150000]，则所有害虫的集合  $U = \{x | 0 < x < 150000, x \in R\}$ 。根据表 3.1 所示的阈值大小，蓟马的计数模型为：

$$a_i = \begin{cases} a_{i-1} + 1 & \mathbf{m} \in \{(x, y, z) | 5 < x < 90, 0 < y < 0.8, 0 < z < 100, x \in R, y \in R, z \in R\} \\ a_{i-1} & \mathbf{m} \notin \{(x, y, z) | 5 < x < 90, 0 < y < 0.8, 0 < z < 100, x \in R, y \in R, z \in R\} \end{cases} \quad (3.8)$$

烟粉虱的计数模型为：

$$b_i = \begin{cases} b_{i-1} + 1 & \mathbf{m} \in \{(x, y, z) | 5 < x < 150, 0.2 < y < 1, 100 < z < 256, x \in R, y \in R, z \in R\} \\ b_{i-1} & \mathbf{m} \notin \{(x, y, z) | 5 < x < 150, 0.2 < y < 1, 100 < z < 256, x \in R, y \in R, z \in R\} \end{cases} \quad (3.9)$$

实蝇的计数模型为：

$$c_i = \begin{cases} c_{i-1} + 1 & \mathbf{m} \in \{(x, y, z) | 19000 < x < 50000, 0 < y < 1, 0 < z < 150, x \in R, y \in R, z \in R\} \\ c_{i-1} & \mathbf{m} \notin \{(x, y, z) | 19000 < x < 50000, 0 < y < 1, 0 < z < 150, x \in R, y \in R, z \in R\} \end{cases} \quad (3.10)$$

其中,  $a_0 = 0, b_0 = 0, c_0 = 0$ , 表示刚开始数量从 0 开始计数。

对每个害虫对象的面积、离心率和颜色平均值进行阈值判断, 最后识别出相应的害虫类型, 进而对害虫进行计数, 检测计数的算法步骤如算法 9 所示。

---

**算法9: 检测计数算法**

---

输入: 单个轮廓的对应的面积、离心率和面积

输出: 三种害虫的计数结果

```

1: CountingResult( S , e , avh )
2:   如果S , e , avh 的取值在蓟马的取值范围内, 则蓟马的害虫数量加一
3:   如果S , e , avh 的取值在烟粉虱的取值范围内, 则烟粉虱的害虫数量加一
4:   如果S , e , avh 的取值在实蝇的取值范围内, 则实蝇的害虫数量加一
5: return countNumber

```

---

### 3.9 本章小结

本章主要介绍了 VPCA-VP 算法的相关概念和流程。该算法首先使用结构化的随机森林处理害虫图像, 得到害虫图像的分割图像; 其次, 使用本文的特征提取算法提取出图像中的不规则结果对象; 最后, 使用检测模型判定害虫的种类。

## 4 算法分析与测试

### 4.1 实验设计

本文实验环境是在自然光照环境下，采用佳能 EOS T2i 单反相机和 iPhone 6s 手机进行拍照，像素分辨率，拍摄地点在广州市天河区柯木塱的一个省农推总站进行，实验拍摄的照片如图 4.1 所示，当时拍摄的时间是在下午的 2 点 30 分到 5 点钟，当时太阳光照相对强烈，拍摄的图片会有一些阴影的出现，而且拍摄的照片还包括除了黄板之外的背景部分，黄色粘板本身又有一些粘液的干扰，照片内容相对复杂，提高了计算机算法识别的难度。实验拍摄照片的像素分别为 3500 万和 1200 万。计算机算法使用的程序语言是 matlab 和 C++，其中 C++ 使用 OpenCV 3.0 的图像处理库。本实验使用 53 张照片对蓟马和实蝇进行计数，由于这些照片存在黄板胶体和光照的影响，实验对烟粉虱计数的效果不佳，便采用另外 30 张光照强度相对较弱的图片对烟粉虱进行计数研究，总共 83 张图片。

### 4.2 害虫分割子算法测试

#### 4.2.1 原图采集

首先，使用佳能相机对省农推总站的黄色诱捕板进行采集，由于 VPCA-VP 算法使用的照片是在大田环境下采集的，所以不必有有拍摄距离或者角度等较为苛刻的约束。但也是两个比较重要的条件，一个是光照不能太过于强烈，另一个就是需要把整个黄板都拍摄进来。拍摄的照片的原图如图 4.1 所示，本节以图 4.1 作为素材介绍算法的具体流程。



图 4.1 黄色诱捕板原图

#### 4.2.2 图像的初步分割

本文采用结构化随机森林和阈值法对害虫图像进行初步分割得到图 4.2 所示结果，由于结构化随机森林是一种需要训练的分割技术，而且训练分割算法的素材包含图像的基本要素，所以分割效果相对以往的方法更加优越，效果如图 4.2 所示，具体算法过程见本文 3.4 节部分。



图 4.2 初步分割结果

#### 4.2.3 黄板二值图像的获取

由于原图是在省农推总站拍摄的，是在大田环境下采集的，包含的背景比较多。因此，本文算法先除去黄板以外的其他背景，这些背景包括一些植物和建筑物，进而得到黄板范围内的图像，也就是获取只有黄板的二值图像。首先，先以图 4.2 为研究对象，把图 4.2 进行取反操作，得到如图 4.3 (a)；其次，本文又对图 4.3 (a) 进行填充操作，得到图 4.3 (b) 的结果；最后，为了获得只有黄板区域的掩码图，便对图 4.3 (b) 中黄板区域的连通域进行选择，存放在另外一个新建的图片，得到图 4.3 (c)。为了后面能够获取到黄板边界的分割图，就需要补充黄板掩码图的缺陷，于是就对图 4.3 (c) 进行膨胀操作，得到图 4.3 (d) 的结果。

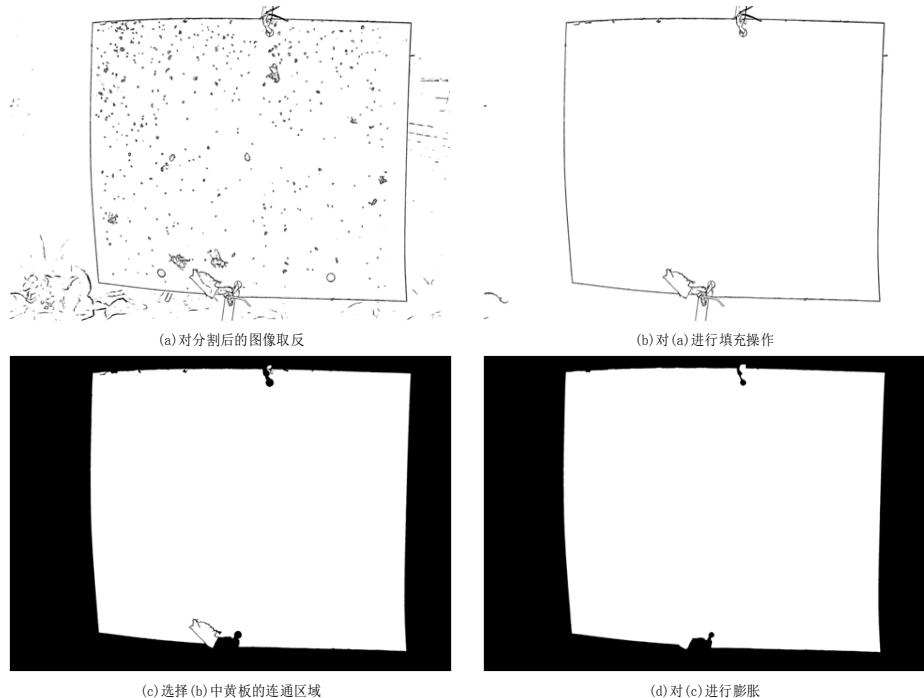


图 4.3 黄色诱捕板的掩码图

经过上述步骤，获得了黄色诱捕板的掩码图，如图 4.3 (d) 所示。把图 4.2 初步分割的结果与图 4.3 (d) 的掩码图进行与运算，去掉黄板区域以外的分割图像，也就是去掉黄板区域以外的建筑物、天空等背景图像，最后得到只有黄色诱捕板的分割图像，如图 4.4 所示。图 4.4 的获得给后面的识别排除了不少干扰。

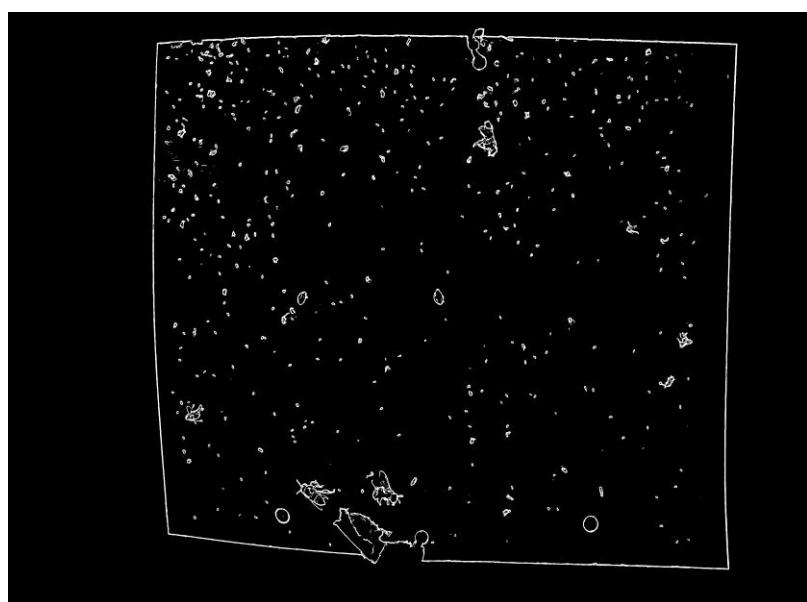


图 4.4 只有黄板区域的分割图

#### 4.2.4 去掉体积较大害虫

由于识别的时候，有可能会把体积较大害虫的部分肢体当成了本文需要识别的害虫。因此，本文先去除大型害虫的干扰，再一次提高后面害虫识别的精度。首先，对图 4.4 进行凸包拟合，找到凸包面积较大的区域，得到如图 4.5 (a) 所示的结果；其次，对凸包拟合的图片使用漫水填充算法进行标记，图片上分别有灰色的背景，黑色的害虫内部区域和白色的害虫边界，而且大型害虫轮廓还包含有子轮廓，如图 4.5 (b) 所示；第三，把 4.5 (b) 进行二分操作，把灰色背景变成黑色，其他的颜色统统变成白色，得到了图 4.5 (c) 的大型害虫掩码图；最后，根据图 4.5 (c) 大型害虫的掩码图，对图 4.4 进行操作，如果是图 4.5 (c) 的白色区域，则把图 4.4 对应区域的像素变成黑色，这样就能去掉黄色诱捕板内大型害虫的干扰，得到只有小型害虫的分割图像，如图 4.5 (d)。到现在为止，我们基本获得了只有小型害虫的分割图像，为后面的识别尽可能排除了干扰。

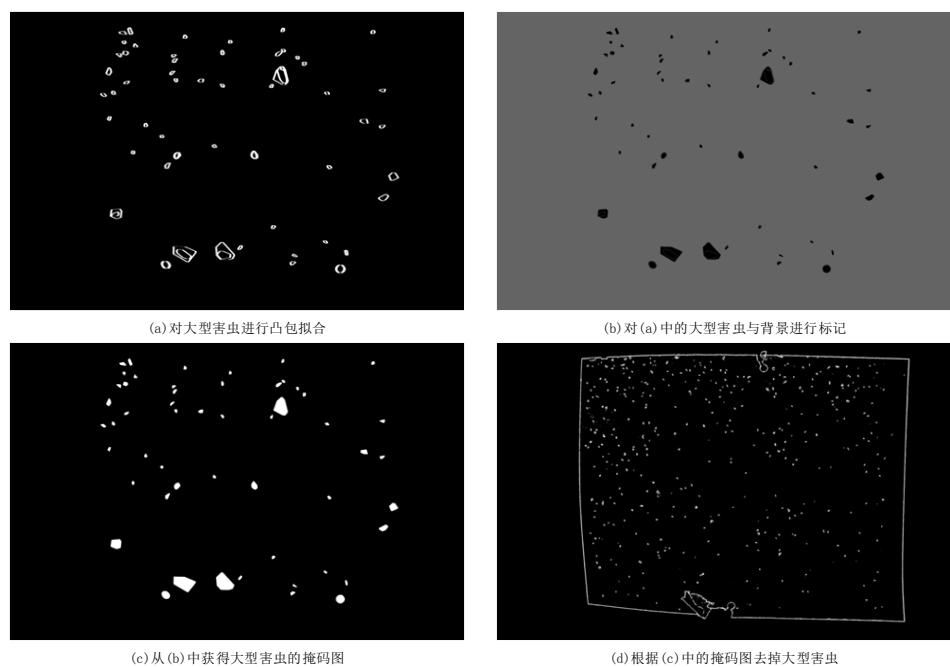


图 4.5 去掉大型害虫的干扰

### 4.3 害虫特征提取子算法测试

根据本文 3.5 的害虫特征提取算法可知，本文的特征提取算法也是采用掩码的形式对传统的特征提取进行相应的改进。使得特征的属性更加贴近于真实的对象，减少了不必要的周边像素的干扰。

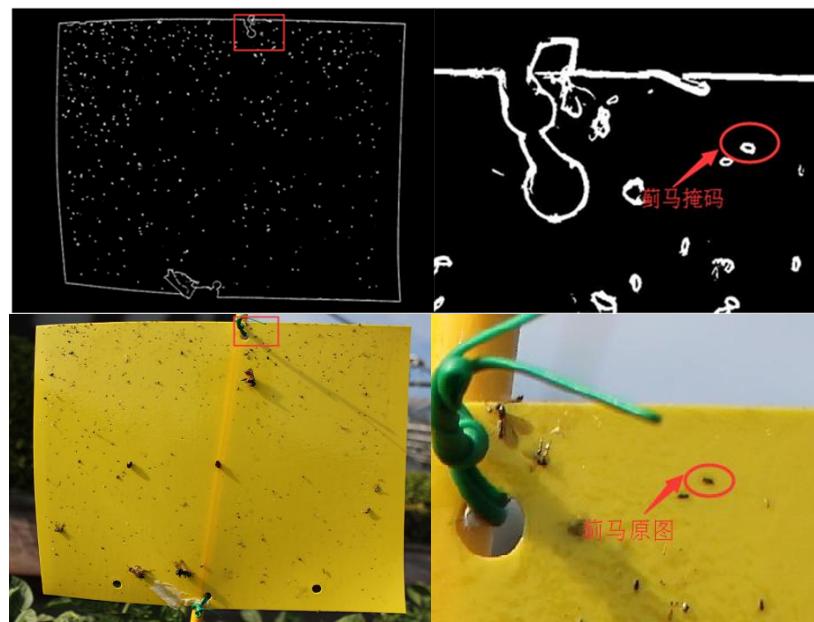


图 4.6 害虫周围的干扰图像

由图 4.6 可知，如果采用传统的方法计算特征值的话，会把周边的黄色像素也计算在内，增加了害虫特征的干扰，因此采用掩码的方式获得原图对象，如图 4.7 所示，通过掩码的形式可以取出不规则图像的区域，然后根据区域的像素值进行直方图的特征统计和计算相应的灰度直方图。



图 4.7 单个蓟马的掩码图

#### 4.4 害虫计数的结果展示

根据本文算法的流程，分别对蓟马、烟粉虱和实蝇进行计数，所有害虫对象均由红色椭圆来标记，计算得到的结果如图 4.8 所示。其中，红色标记的方框是本文识别计数的一部分害虫对象；绿色标记的方框是其他类型的害虫，不是本文计数的害虫类型；白色标记的方框是害虫产生阴影的干扰；蓝色方框是黄板背面产生的背影干扰。

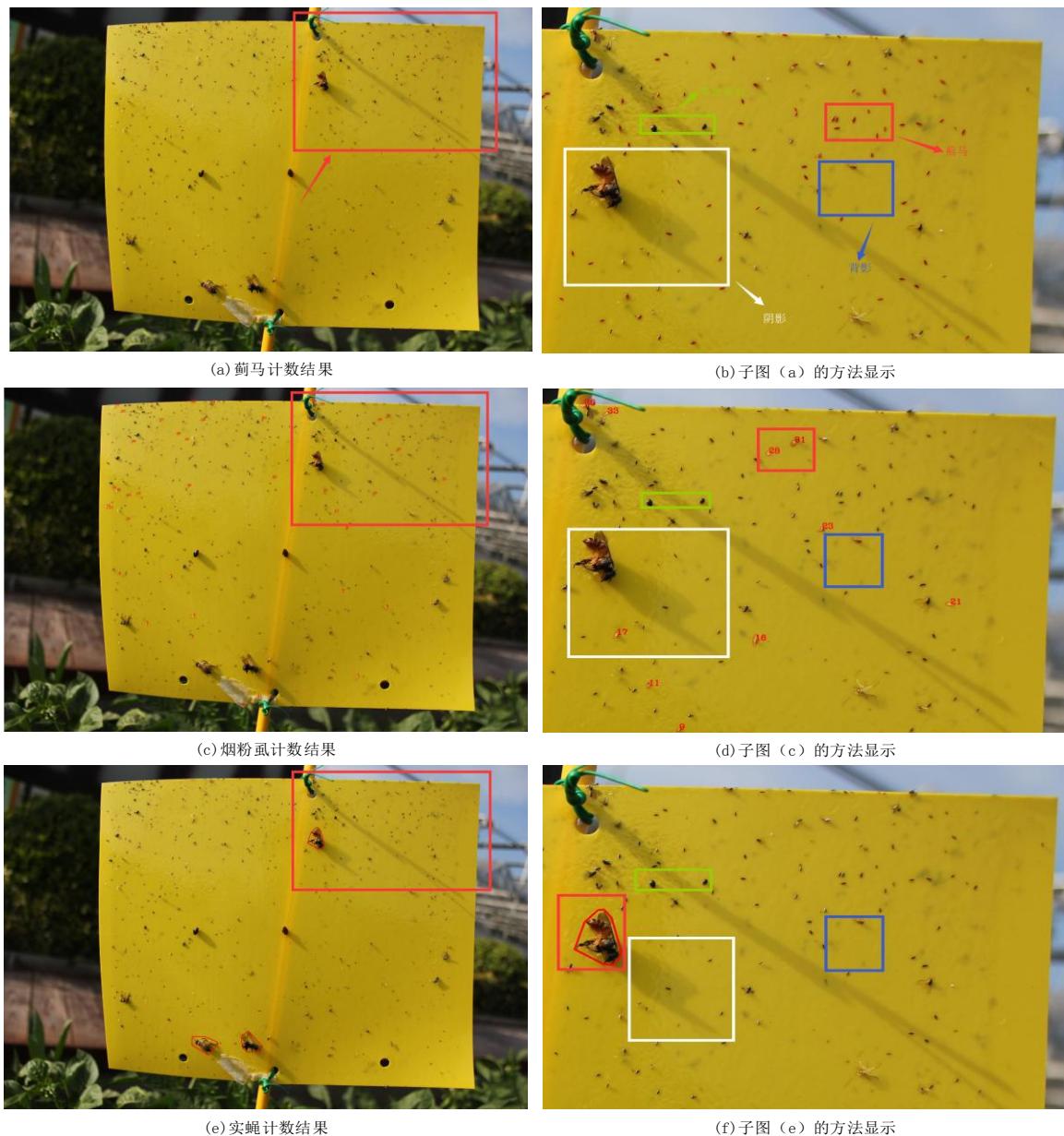


图 4.8 害虫计数结果

从图 4.8 可知，本文算法可以准确对蓟马、烟粉虱和实蝇进行计数，并且可以避免其他类型害虫的干扰、背影的干扰和阴影产生的干扰，计数效果显著。

#### 4.5 检测模型计算算法精度的测试与分析

本文提出的 VPCA-VP 算法是在随机森林算法的基础上进行改进的，拍摄的照片是在纯自然大田环境下进行的，拍摄对象主要是黄板，计算所有害虫数量的时候，准确率比较高。在计算大型害虫实蝇的时候，计数的精度达到 100%。但是，在计算蓟马和烟粉虱数量的时候，准确率相对较低。在具体的计数过程中，由于光照不均匀、黄板杂质、背景等影响，会存在一些漏检和误检的情况。在强光的条件下，黄板表面

的胶水液体会出现一些光线的反射，程序会错误地认为反射的胶水是一个害虫，进而产生了误检的情况。在对蓟马进行计数的时候，受到强光照的影响，蓟马本身颜色发生了一些变化，颜色特征变得偏白一些，使得程序判别的时候当成了其他害虫进行处理，进而发生了漏检的情况。

本文先用 53 张图片进行算法的测试，对这 53 张图片分别进行人工计数和程序计数，用这两种计数方式做为一个对比，后来发现在光照较强条件下，烟粉虱的识别效果并不理想，后来选取了另外的 30 张图片进行烟粉虱的性能测试。本文算法通过混淆矩阵的数值等相关度量值对本文算法进行性能的评价。为了计算相应的度量值，需要把每种害虫的混淆矩阵计算出来，如图 4.9 所示：

		预测的情况	
		正例	反例
真实情况	正例	真正	假负
	反例	假正	真负

图 4.9 混淆矩阵

其中，上述混淆矩阵的四个专业术语分别表示如下：

- (1) 真正 (True Positive, $TP$ ) 表示检测模型正确检测该类害虫的个数。
- (2) 假负 (False Negative, $FN$ ) 表示检测模型漏检该类害虫的个数。
- (3) 假正 (False Positive, $FP$ ) 表示检测模型误检成该类害虫的个数。
- (4) 真负 (True Negative, $TN$ ) 表示检测模型正确判断不属于该类害虫的个数。

最后，由混淆矩阵可得精度 ( $p$ )、召回率 ( $r$ )、( $F_1$ ) 度量和相对误差 ( $\varepsilon$ ) 分别用公式 4.1、4.2、4.3、4.4 表示。

$$p = \frac{TP}{TP + FP} \quad (4.1)$$

$$r = \frac{TP}{TP + FN} \quad (4.2)$$

$$F_1 = \frac{2}{\left(\frac{1}{p} + \frac{1}{r}\right)} \quad (4.3)$$

$$\varepsilon = \frac{(TP + FN) - (TP + FP)}{(TP + FN)} \quad (4.4)$$

#### 4.5.1 算法的常规度量分析

对于 VPCA-VP 算法，本文选取了蓟马、烟粉虱和实蝇这三种害虫为识别对象，对这三种害虫的计数结果与人工进行对比，得到表 4.1 所示的结果。

表 4.1 基于黄色诱捕板的害虫快速算法的性能

	蓟马	烟粉虱	实蝇
真正 ( $TP$ )	9351	202	23
假负 ( $FN$ )	544	22	2
假正 ( $FP$ )	683	20	0
真负 ( $TN$ )	8631	7469	18501
精度 ( $p$ )	0.93	0.91	1
召回率 ( $r$ )	0.95	0.90	0.92
$F_1$ 度量	0.94	0.91	0.96

根据表 4.1 的算法性能展示，我们可以知道蓟马的精度为 0.93，烟粉虱的精度为 0.91，实蝇的精度基本为 100%，三种害虫的精度都能满足实际识别的需求。另外，在图像识别的领域，我们都希望精度和召回率最大化，因此我们使用  $F_1$  度量来衡量这个因素，从表 4.1 我们可以知道，实蝇的  $F_1$  度量最大，这是因为实蝇个体比较大，相对容易识别。在小型害虫里面，蓟马的  $F_1$  度量值更加理想。最后，实蝇的 FP 值为 0，也就是说本文的算法对于实蝇来说，不可能识别错误，有两个漏掉的实蝇是由于实蝇个体粘黏在黄板太久，个体发生了变形，总体来说对于大型害虫如实蝇的识别效果是理想的。

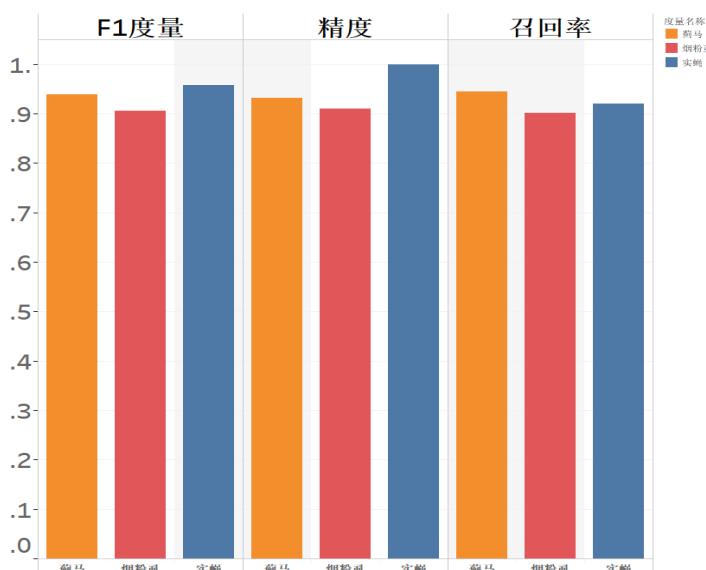


图 4.10 算法三种度量的条形图

从图 4.10 我们可以知道，烟粉虱的性能相对较低，精度只有 91%，经过分析发现，主要的影响因素是黄色诱捕板上面的胶体发生了一些光照的反射，因为本文的检测模型是通过颜色特征、边界特征和区域特征来确定的，光照的反射使得胶体呈现出类似烟粉虱的图像，进而产生了误差。

#### 4.5.2 算法的相对误差分析

为了更好的分析 VPCA-VP 算法，对每张黄板的计数结果进行分析，计算出每张黄板的相对误差。把人工计算害虫的数量作为横坐标，对应害虫的相对误差作为纵坐标，采用散点图的方式把分布情况勾勒出来。另外，为了看见的取值范围，分别把相对误差的最大值直线、最小值直线和 0 误差直线绘制出来，便于分析讨论，散点图如图 4.11 所示。

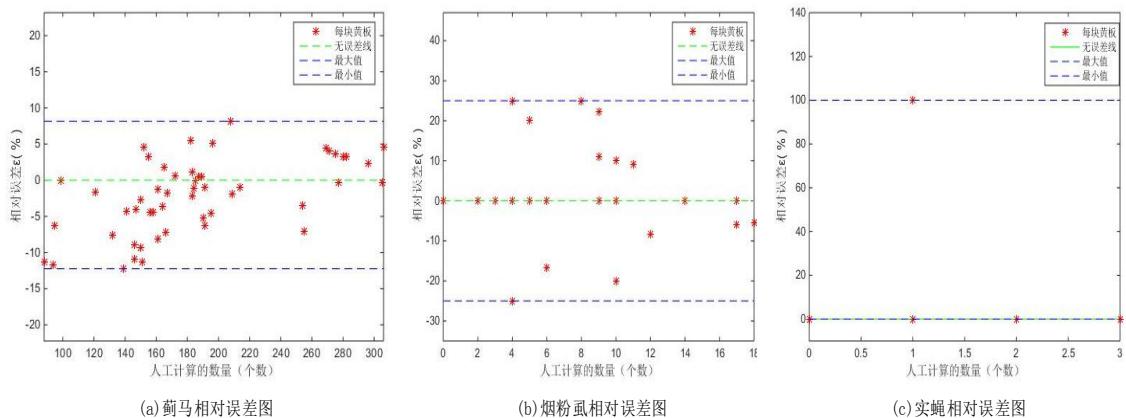


图 4.11 算法的相对误差图

由图 4.11 (a) 可知，薺马的误差波动在-12.23%到 8.17%之间波动，平均误差为-1.40%，绿色虚线到上面的蓝色虚线之间的宽度比绿色虚线到下面蓝色虚线的宽度窄，说明计算错误的可能性比遗漏的可能性更加大。

由图 4.11 (b) 可以知道，烟粉虱的相对误差从-25%到 25%之间波动，平均误差为 0.89%，波动范围相对较大，错误的可能性和漏算的可能性基本差不多，但从总体来说，平均误差在 1%以内，相对可以接受。

由图 4.11 (c) 可知，相对误差波动较大，误差从 0 到 100%波动，基本不稳定，这是因为真实情况下实蝇在黄板上的数量是 25 个，数量较少，一个黄板平均一个实蝇都不到，一旦出现一个实蝇变形，直接导致这个黄板相对误差变成 100%。虽然数量极少，但是实蝇的平均误差也就 8%，也是可以接受的范围。另外，绿色虚线到下

面的蓝色虚线之间的宽度为 0，说明计算错误的可能性基本不存在。因为实蝇体积较大，所以也较符合常理。

#### 4.5.3 算法计数和人工计数的对比分析

本小节把 VPCA-VP 算法与人工计数的方法进行比较。参考每块黄板的结果数据，把人工计算害虫的数量作为横坐标，算法计算对应害虫的数量作为纵坐标，绘制散点图，并且对每个数据进行线性回归拟合，画出每块黄板结果数据的回归直线，结合回归直线和散点的分布对本文算法进行分析，如图 4.12 所示。

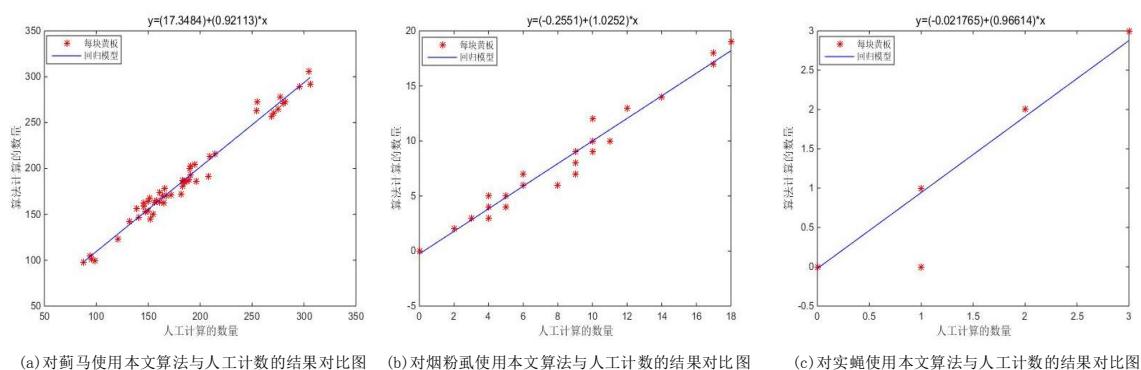


图 4.12 算法计数和人工计数的回归模拟

由图 4.12 可知，从左到右的顺序分别为蚜虫、烟粉虱和烟粉虱的算法和人工的对比，除了实蝇有个别离群点之外，三种害虫的算法计数和人工计数基本符合线性模型的分别，也就是说算法总体精度较为可靠。从图 4.12 (a) 可知，离散点基本贴近回归直线，所以算法对蚜虫计数的结果和人工基本一致，蚜虫的可决系数  $R^2$  为 0.98，算法精度较高；相比于图 4.12 (a)，图 4.12 (b) 中的离散点也基本贴近回归直线，烟粉虱的可决系数  $R^2$  为 0.97。从图 4.12 (c) 可知，实蝇的可决系数  $R^2$  为 0.94，拟合程度相对较差，这是由于每块黄板的实蝇数量较少，只要有一个遗漏就会产生较大的偏离，但大多数时候算法都能找到实蝇，而且找到的基本都是对的，所以总体性能相对也是可以接受的。

综上所述，本文算法在纯大田环境下性能良好，可以较为准确地对害虫进行计数，有利于真实环境下害虫的计数研究，为纯田间害虫的自动化计数提供了重要理论基础。

## 4.6 蔬菜害虫分类计数算法的初步应用

本小节主要展示的是 VPCA-VP 算法软件的一些界面。以算法为核心，使用 C++ 的 MFC 框架搭建了一个害虫计数软件。本软件主要有两个功能，分别是计算单张图片的害虫数量和计算批量图片的害虫数量。

(1) 计算单张图片的害虫数量。首先，点击“选择原图”的按钮，选择害虫原图片进行计数；其次，点击“开始计数”按钮，对选择的图片进行计数；最后，显示的结果显示在同一个界面上，如图 4.13 所示。

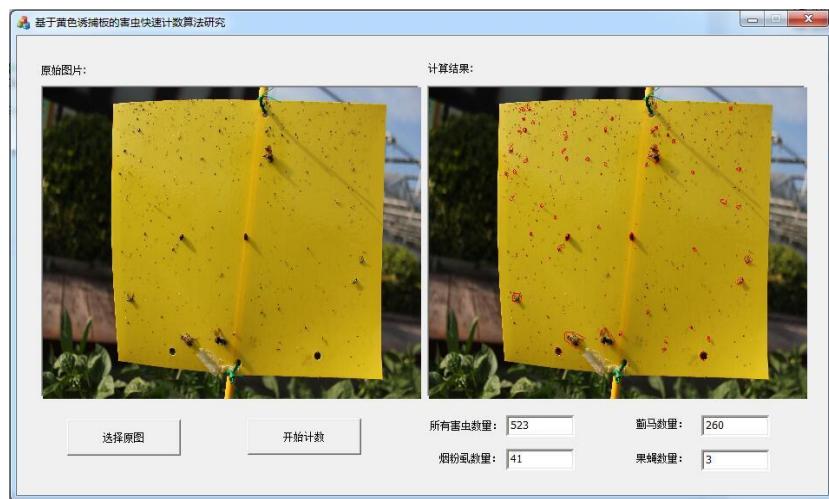


图 4.13 单张图片计数的界面

(2) 计算批量图片的害虫数量。首先，点击“选择批量原图”的按钮，选择计算数量的所有原图；其次，点击“开始”计算的按钮，对选择的所有害虫图片进行累加计数；最后，得出的计数结果显示在本界面上，如图 4.14 所示。



图 4.14 批量图片计数的界面

## 4.7 本章小结

首先，本章叙述实验器材、实验地点等具体的实验方案。其次，根据 VPCA-VP 算法的流程展示每个算法的结果，说明算法的可行性。最后，本章阐述了一些算法性能评价的方法及其计算的方法，然后分别从常规性能、残差和算法与人工的拟合程度这三方面对薺马、烟粉虱和实蝇进行分析，可以得出薺马和烟粉虱的精度都在 90% 以上，其中薺马的精度相对更高一些，而实蝇的精度可以达到 100%，主要是因为实蝇体积相对较大，识别相对简单。

## 5 总结与展望

当前所有的田间视觉计数都没有在图像分割上下功夫，而图像分割又是在识别过程中必不可少的一部分。国内外研究田间计算机视觉相对较晚，开始于 1980 年左右，从那时候到现在为止，并没有很多的文献在分割上下功夫。国内外的一些文献，他们的图片素材大多来自温室或者实验室拍摄的，有的则是在叶表面拍摄的，害虫种类相对单一，光照等自然环境影响相对不大。本文所采用的照片则是在自然环境下田间害虫的图片，在现场进行拍摄，环境相对多样，识别精度也相对较好，足以满足实际的计数要求。本文的主要工作如下：

(1) 为了采集相应的照片，经过了一年的实验工作，对诱捕装置进行多次选择和改进，多次在省农推总站和番禺东升农场等地方拍摄相片，最后选择用诱捕板的形式对害虫进行诱捕和拍摄。

(2) 充分比较了以往各种分割算法的优劣，最终选择了结构化随机森林和相似性描述子的思想，结合降维和主成分分析的算法，初步得到较好的分割效果，然后根据初步分割的图片，利用填充算法等图像处理技术去掉背景，再利用掩码的形式去掉大型害虫的干扰，最后对小型害虫进行识别计数。

(3) 对分割的害虫对象进行特征提取，传统的特征提取会包含有对象周围的干扰区域，本文采用掩码的形式提取了只有害虫区域的颜色特征、区域特征、边界特征和纹理特征，有利于后面对识别精度的提高。

(4) 对提取的 12 个特征进行分析，提取主要的特征作为识别的参数。人工分割出蓟马和其他类别的害虫，计算它们的 12 个特征，然后使用 SPSS 对特征进行主成分分析，得出排名较前的几个特征。

虽然我们的分割对象是黄板的害虫，但是本文的分割算法稍微修改一下可以应用在蓝板上识别，而且分割效果也不错。本文的结构化随机森林算法可以用在其他任何领域的分割，而且分割效果相对不错。

本文下一步的工作是按照之前拍摄的实践经验，拍摄各种情况的相片，找出更加符合农田环境的特征描述子，分析选取其中主要的描述子，然后使用神经层数较少的神经网络进行分析，最终提高在自然环境下黄板害虫的识别率。其次，由于本文的相片是使用单反相机拍摄的，像素分辨率高，想在农田使用单反相机不太现实，所以需要使用相对便宜的罗技摄像头对田间进行拍摄，然后使用超分辨率的思想把低分辨率

的图片进行重建，然后使用本文算法在田间实时计数，通过无线网络传输相应的结果回服务器，进而实现实时的监控。

## 致 谢

时间稍纵即逝，转眼间读研时光就快过去了。在这段难忘的学校路途里面，我取得了学业上巨大的进步，使自己在生活上变得更加独立，而这都离不开众多老师和同学对我的指导和关怀。在此，我要由衷地对他们表达我的谢意。

首先，我要由衷地感谢我的指导老师肖德琴教授。在本论文的选题、设计和撰写的过程中，给予了我很多非常有价值的指导性意见，特别是在文字表达方面，给予了我很多的建议，从而使我的毕业设计能够顺利完成。另外，在求学期间，肖老师给我们提供了良好的学习环境和许多实践机会，使在技术和学术方面都获得了巨大的进步，特别是学术方面，每周的学术研讨使我的学术水平得到了明显的提升。同时，肖老师还在科研设备上给予了我巨大的帮助。在此，谨对肖老师这两年来的悉心栽培、包容以及无微不至的关怀表示衷心的感谢，并向她表示深深的敬意。

其次，我要感谢王春桃老师对我学习期间的指导。每次在学术上遇到不懂问题的时候，王老师都会耐心地讲解给我听，帮助我解决一些论文上不懂的问题。王老师丰富的知识和敏捷的以及乐于助人的精神是我学习的榜样。另外，我还要感谢在实验过程中给予我帮助的陆永跃老师，在害虫领域给予我很多的意见，并提供了很多生物学方面的知识。

此外，我还要感谢数学与信息学院 640 实验室的冯健昭老师、张连宽老师、潘春华老师、殷建军老师、杨秋妹老师、可欣荣老师、郭艾侠老师以及叶耀文、蔡家豪、冯爱晶、李泽森、刘小平等同学在学校和实践中给我的帮助和支持。

感谢我的家人，是他们的关心、理解和默默的付出，才使得我能顺利的完成学业。

最后感谢从百忙之中抽空评审和参加论文答辩的老师们，感谢你们对本论文提出的意见和建议。

## 参 考 文 献

- 陈朗. 蔬菜残留农药对人体的危害[EB/OL].  
<http://www.zhengjia.com/baojianyangsheng/yangshengchangshi/25749.shtml>.
- 陈梅香, 杨信廷, 石宝才, 等. 害虫自动识别与计数技术研究进展与展望[J]. 环境昆虫学报, 2015, 37(1):176-183.
- 陈晓明, 王程龙, 薄瑞. 中国农药使用现状及对策建议[J]. 农药科学与管理, 2016, 37(2):4-8.
- 韩瑞珍, 何勇. 基于计算机视觉的大田害虫远程自动识别系统[J]. 农业工程学报, 2013, 29(3):156-162.
- 胡雅辉, 高灵旺. 烟粉虱成虫密度自动计数系统[J]. 应用昆虫学报, 2011, 48(1):84-87.
- 京报集团. 菊马近年大发生[EB/OL]. [http://jjrb.bjd.com.cn/html/2015-08/03/content\\_300104.htm](http://jjrb.bjd.com.cn/html/2015-08/03/content_300104.htm).
- 李英娇. 我国蔬菜出口的比较优势研究[D]. 上海海洋大学, 2016.
- 马广鹏. 生物农药的发展与机遇[J]. 上海农业学报, 2013, 29(3):92-97.
- 邱道尹, 张红涛, 刘新宇, 等. 基于机器视觉的大田害虫检测系统[J]. 农业机械学报, 2007, 38(1):120-122.
- 沈佐锐, 于新文. 温室白粉虱自动计数技术研究初报[J]. 生态学报, 2001, 21(1):94-99.
- 谢成军, 李瑞, 董伟, 等. 基于稀疏编码金字塔模型的农田害虫图像识别[J]. 农业工程学报, 2016, 32(17):144-151.
- 徐向超, 赵维平, 李川, 等. 基于KL变换与BP神经网络的农田害虫识别算法[J]. 山西科技, 2015(2):116-119.
- 尤帅. 中国蔬菜产品出口贸易影响因素分析[D]. 山东财经大学, 2016.
- 张水发, 王开义, 刘忠强, 等. 基于离散余弦变换和区域生长的白粉虱图像分割算法[J]. 农业工程学报, 2013, 29(17):121-128.
- 周志艳, 罗锡文, 张扬, 等. 农作物虫害的机器检测与监测技术研究进展[J]. 昆虫学报, 2010, 53(1):98-109.
- Dalal N, Triggs B. Histograms of oriented gradients for human detection[C], IEEE Computer Society Conference on. IEEE, 2005, 1:886-893.
- Dollár P, Tu Z, Perona P, et al.. Integral Channel Features[C]. British Machine Vision Conference, BMVC 2009, London, UK, September 7-10, 2009.
- Dollár P, Zitnick C L. Structured Forests for Fast Edge Detection[C]. IEEE International Conference on

Computer Vision, 2013:1841-1848.

Dollár P, Zitnick C L. Fast Edge Detection Using Structured Forests[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014,37(8):1558-1570.

Espinoza K, Valera D L, Torres J A, et al.. Combination of image processing and artificial neural networks as a novel approach for the identification of *Bemisia tabaci* and *Frankliniella occidentalis* on sticky traps in greenhouse agriculture[J]. Computers & Electronics in Agriculture, 2016,127:495-505.

Fukatsu T, Watanabe T, Hu H, et al.. Field monitoring support system for the occurrence of *Leptocoris chinensis* Dallas (Hemiptera: Alydidae) using synthetic attractants, Field Servers, and image analysis[J]. Computers & Electronics in Agriculture, 2012,80(1):8-16.

Gassoumi H, Prasad N R, Ellington J J. Neural Network-Based Approach For Insect Classification In Cotton Ecosystems[J]. International Conference on Intelligent Technologies(InTech 2000), Bangkok, Thailand. 2000. 7

Li Y, Xia C, Lee J. Detection of small-sized insect pest in greenhouses based on multifractal analysis[J]. Optik - International Journal for Light and Electron Optics, 2015,126(19):2138-2143.

Lim J J, Zitnick C L, Dollar P. Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection[C]. Computer Vision and Pattern Recognition, 2013:3158-3165.

Lowe D G, Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004,60(2):91-110.

Martin D, Fowlkes C, Tal D, et al.. A Database of Human Segmented Natural Images and its Application to evaluating segmentation algorithms and measuring ecological statistics[C].IEEE, 2001,2:416-423.

Murakami S, Homma K, Koike T. Detection of small pests on vegetable leaves using GLCM[C]. 2005 ASAE Annual International Meeting, Tampa, Florida, USA, 2005,123-136.

Shariff A R M, Aik Y Y, Hong W T, et al.. Automated Identification And Counting Of Pests In The Paddy Fields Using Image Analysis[C]. Computers in Agriculture and Natural Resources, 23-25 July 2006, Orlando Florida, 2006,2325-759764.

Shechtman E, Irani M. Matching Local Self-Similarities across Images and Videos[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2007,1-8.

Sun Y, Cheng H, Cheng Q, et al.. A smart-vision algorithm for counting whiteflies and thrips on sticky

traps using two-dimensional Fourier transform spectrum[J]. Computer & Digital Engineering, 2007,153(2):82-88.

Tan P, Steimbach M, Kumar V. Introduction to Data Mining[M]. 人民邮电出版社, 2011.

Tirelli P, Borghese N A, Pedersini F, *et al.*. Automatic monitoring of pest insects traps by Zigbee-based wireless networking of image sensors[C]. Instrumentation and Measurement Technology Conference, 2011,1-5.

Xia C, Chon T S, Ren Z, *et al.*. Automatic identification and counting of small size pests in greenhouse conditions with low computational cost[J]. Ecological Informatics, 2015,29:139-146.

Yao Q, Jun L V, Liu Q J, *et al.*. An Insect Imaging System to Automate Rice Light-Trap Pest Identification[J]. Journal of Integrative Agriculture (农业科学学报(英文)), 2012,11(6):978-985.

Zitnick C L, Dollár P. Edge Boxes: Locating Object Proposals from Edges[M]. Springer International Publishing, 2014,391-405.

## 附录：攻读硕士学位期间的科研工作

### 参与的科研项目

[1]国家星火计划项目“南方蔬菜重大害虫快速监测及预警技术集成与应用示范”（项目编号：2015GA780002）

### 获得过的软件著作

[1]软件名称：南方蔬菜重大害虫监测预警平台（登记号：2016SR384669）

### 获得过的专利：

[1]专利名称：一种基于视频追踪的猪只运动大数据提取方法及其系统（专利号：2016101175765）

[2]专利名称：一种猪只行为节律分析方法（专利号：2016106452285）

### 审核过程的专利、论文

[1]专利名称：基于视觉传感器的南方蔬菜重大害虫诱捕监测装备（审核过程中）

[2]论文名称：基于视觉感知的蔬菜害虫计数算法（十一届中国物联网学术会议）

### 获得的奖励

2013年12月获得国家奖学金

2013 年获得“蓝桥杯”全国软件专业人才设计与创业大赛广东赛区 C/C++ 程序设计本科 B 组二等奖

2014 年获得“蓝桥杯”全国软件专业人才设计与创业大赛全国赛区 java 程序设计本科 B 组二等奖

2015 年 11 月获得华南农业大学“一等奖学金”

2016 年 11 月获得华南农业大学“二等奖学金”