

水果分拣平台的开发设计—基于树莓派的机器视觉开发

摘 要

我国作为世界首屈一指的农业产品生产大国，在国际农业贸易市场上的销售额却低于美国、英国等发达国家。在一段时间内，这令人十分感到困惑，在经过农业与市场的专家学者调研后发现，其背后的原因是，我国的水果在成熟后，缺乏相应配套的批量处理设施，特别是分拣部分，尚未完全实现自动化流水线作业。在此领域我国长期依赖手工作业，靠着大量的普通工人来对水果进行分类。而随着市场的变化，当前的分类要求也越来越细致，从传统的仅仅按照水果类别分类，如今已经开始按照水果的大小、个头、颜色饱和度等更具有个性化的指标来区分，这使得针对农业果类自动分拣平台显得日益迫切。结合当前国内外水果分拣的研究现状，本设计实现了基于树莓派机器的视觉开发，可以在水果分拣平台上得到很好的应用。

关键词： 机器视觉； 水果分拣； 分类； 树莓派

Development and Design of Fruit Sorting Platform—Development of Machine Vision Based on Raspberry Pi

Abstract

Abstract: As the world's leading agricultural production country, China's sales in the international agricultural trade market are lower than those in the United States, Britain and other developed countries. For a period of time, this was very confusing. After investigations by experts and scholars in agriculture and the market, it was found that the reason behind it is that after the fruits of our country mature, they lack the corresponding batch processing facilities, especially the sorting part. , Has not yet fully realized automated assembly line operations. In this field, China has long relied on manual work, relying on a large number of ordinary workers to classify fruits. With the change of the market, the current classification requirements are becoming more and more detailed. From the traditional classification based on fruit categories, they have started to distinguish according to more personalized indicators such as the size, size, and color saturation of fruits. It makes the automatic sorting platform for agricultural fruits appear increasingly urgent. Based on the current research status of fruit sorting at home and abroad, this design realizes the visual development based on the Raspberry Pi machine, which can be well applied on the fruit sorting platform

Keywords: Machine vision; Fruit sorting; Classification; Raspberry pie

目录

| | |
|------------------------------|----|
| 1 绪论 | 1 |
| 1.1 研究意义及背景 | 1 |
| 1.2 国内外研究现状及发展趋势 | 2 |
| 1.2.1 国外研究现状 | 2 |
| 1.2.2 国内研究现状 | 2 |
| 1.3 本文的主要内容与章节安排 | 3 |
| 2 系统框架设计 | 4 |
| 2.1 系统总体设计 | 4 |
| 2.2 系统硬件架构 | 5 |
| 2.2.1 树莓派简介 | 5 |
| 2.2.2 摄像头 | 7 |
| 2.2.3 笔记本电脑 | 9 |
| 2.3 系统软件平台的构建 | 10 |
| 2.3.1 笔记本电脑软件平台 | 10 |
| 2.3.2 树莓派软件平台 | 10 |
| 2.3.3 Open Cv 概述 | 11 |
| 2.4 本章小结 | 12 |
| 3 图像的采集与预处理 | 13 |
| 3.1 水果图像的采集 | 13 |
| 3.2 水果图像的预处理 | 14 |
| 3.2.1 图像灰度化 | 14 |
| 3.2.2 图像二值化 | 16 |
| 3.3 本章小结 | 17 |
| 4 水果实时分类算法的实现 | 18 |
| 4.1 水果图像轮廓提取 | 18 |
| 4.2 水果特征提取 | 20 |
| 4.3 水果实时分类算法的设计 | 21 |
| 4.3.1 决策树算法 | 22 |
| 4.3.2 基于决策树的水果实时分类算法实现 | 22 |
| 4.4 本章小结 | 24 |
| 5 测试与结果分析 | 25 |
| 5.1 水果分类功能测试 | 25 |
| 5.2 水果分类性能测试 | 25 |

北京理工大学珠海学院 2020 届本科毕业生设计

| | |
|----------------|----|
| 5.3 本章小结..... | 26 |
| 6 总结与展望..... | 27 |
| 6.1 总结..... | 27 |
| 6.2 展望..... | 27 |
| 参考文献..... | 28 |
| 致谢..... | 29 |
| 附录 1 | 30 |
| 核心程序代码..... | 30 |
| 附录 2 | 35 |
| 5 种水果效果图 | 35 |

1 绪论

1.1 研究意义及背景

我国一直是世界上水果种植产量排名较前的农业生产大国，自 20 世纪 90 年代后，由于大面积的化肥的使用我国的水果收获量一直处于世界前列。但由于工业的发展起步不够领先，相比于欧洲等老牌工业国家我国的水果成熟后加工技术，配套的处理技术，一直处于一个较低的水平。也正是因为这种原因，一直无法在国际贸易市场获得优异的表现[1]。在 2018 至 2019 年，我国的生鲜苹果的出口量就有 168 万吨，而同比我国的果实产量却远高于此，除了部分果实是直接内销外，还有很多果实直接成为损耗成本了，因为我国一直采用人工的采集流水线作业，对比国外的全自动机械化流水线，我国的果实产业竞争力落后明显。在我国，果实从田地收获到包装出售，每一步均依靠技能成熟的工人手工作业，即使工人们均接受了一定程度的训练，拥有比较高的效率，但是在分类与打包环节，依然耗费了大量的生产时间，这倒是数量巨大的生鲜果实腐烂变质失去了食用价值。而随着人们对个人美食生活质量的追求，人们购买水果除了原有的种类维度上，还更加关注色泽、饱和度、尺寸等更加细分的维度，这些元素的增加，无疑给人工分类的难度又提升了一个台阶，势必导致效率进一步的下降，收获果实的有效产值更加低下。为了改善这种不良的竞争情况，我国一直在把提升果实产业的效益，降低损耗作为研究发展的主要方向。我国目前已经普遍可以购买到机械分类设备主要是依靠物体的重量来进行分类，显然，针对果实产业这种需要精细分类的情况，面对的问题与预想情况存在较大差距。

针对这种具体的情况，有专家学者从视觉研究方向的角度提出，可以以图像的应用处理技术为理论基础，将原本应用于机器视觉领域的图像识别技术，移植应用到水果分拣领域，通过上述理论基础建立起依靠机器视觉的自动化果实分类系统。该系统在一般情况是通过高清摄像头作为原始数据采集模块，采集果实或者其他物件的图像信息，然后传送后台的实时图像数据分析处理模块，得出原有的图像信息，将水果的像素分布情况、水果的外形特征数据、果皮的颜色饱和度数值、整体尺寸的大小等高维信息进行特征提取与数据处理，通过一定的映射关系将其转变为机器可以识别的数字化信号，从而让机器可以对这些图像有一个认识分类的前提，在获取大量特征数据后，我们利用高性能的服务器算力，对这些特征进行下一步的处理，最终使得机器能自动学习到某些特征与一类水果的归属关系，从而为一步流水线上的机械手分拣操作，打下基础。将前者与机械手系统联合起来，整个系统就可以实现无人值守的自动化操作方式，流水线作业分拣水果，最大程度的降低收获果实的等待出库时间，显著提高了分拣效率。

国外已经有成熟的分类产品，但是此类设备进口价格非常昂贵，除了国内的小农户无法承担其价格外，还存在一旦出现问题的时候，会面临无人维护的麻烦局面。针对我国特有的小农户种植园的农产品自动化加工的需求，我们在研究了许多专家学者的论文和调研了实际市场情况后，对水果自动识别系统的研究开发工作，目的是希望能解决我国小农户种植园的痛点需求，成功设计出一款成本相对低廉，工作环境耐受度高，且分类效果好，识别率有一定保障的水果自动实时分类系统。系统硬件选择树莓派平台与树莓派匹配的专用摄像头，结合本文设计的水果实时分类算法，实现多种水果的实时分类，系统在完成分类后将结果通过 USB 串口发送处理数据，使用更加统一化的 USB 口替代老旧的串口，除了可以提高信号的传输稳定性外，还可更加方便的适配不同的机械模块，便于后续的直接分类或是更复杂切割操作。通过模拟实验与测试，本文设计的水果分类系统面对常见的多种水果，能实现 100%识别的精准分类。

1.2 国内外研究现状及发展趋势

机器视觉被应用的范围非常之广，从农业到工业领域都有很大的应用面。正因为如此，世界上许多国家与地区都在此领域投入了巨大的人力与物力进行发展，并实现了一些成果，总体上来讲，国外的老牌工业国家在此领域的整体技术是大幅领先于国内同期的发展水平。早在 20 世纪 60 年中期，以欧美国家，特别是美国为首的高技术发达国家已经开始着手这方面的科学研究，其中美国学者罗伯兹在此方面的研究最为深入。因此，发达国家基于机器视觉的产业已经发展得十分完备，他们在各种复杂物体的识别上、机械部分的联动上，均有了一段时间的成熟应用。国外厂商推出的基于机器视觉的分拣系统具有噪音小、功耗低、机械手操作精准、物体识别精度高等优势，而同期国内的发展，仍处于研究发展初期，与顶尖的国际水准仍持有的不可忽视的差距，其主要难点在于，视觉模块的识别精度不够高，对图像的处理算法不够先进。所以我们机器视觉上的研究，对推进机器视觉的分拣技术的发展很有意义。

1.2.1 国外研究现状

目前，国外对机器视觉的研究现状已经逐步从实验理论走向了工厂实践，他们已经有了非常多的型号的成熟产品。其实以美国、日本、意大利为第一批队。例如英国的基恩士厂商，其机器视觉的实现不仅依赖于传统的图像识别，还增加光电感应与近距离接触感应，其销售产品遍布全球。意大利爱乐备公司推出的基于三维视觉的机械手臂同样具有非常高的技术水平，能够实现的全方位多角度的物料获取，在国际上属于领先水平，可以规划完成复杂多样的取件操作。美国的康耐视公司更是机械领域的头部公司，公司旗下的视觉识别部门，专攻用于实现高精度、低误差的分类视觉系统。

1.2.2 国内研究现状

由于我国的步入工业时代比较晚，早期时候，我国的生产力主要以模仿制造为主，故在机器视觉技术上投入比例不大，研究的起步时间比较晚，相比于国外的情况，我国由于受一些历史局限性的影响，在此领域起步较晚，直到 20 世纪 90 年代才开始在这个领域投入资金与人力进行研究。而在研究初期，绝大部门研究单位主要集中在一流高校的实验室，其中主要以哈尔滨工业大学为首，后期开始才逐步将技术应用到民营企业。最近几年，我国在此领域小有成就的公司开始慢慢多了起来，例如深圳的慧眼机械公司、东莞的名赛机器人公司等。

1.3 本文的主要内容与章节安排

本文设计利用树莓派与搭载的摄像头器件，构建基于树莓派的机器视觉开发平台。其主要内容是，对原始图片要进行合理的预处理，否则原始输入的数据量过大会导致树莓派过热死机，也会导致分类的速度过慢。然后在不失真的前提下尽量降低原始数据量，不能因为过度压缩数据而使得图片失真，要在预处理的图片选取具有足够代表性的特征维度，利用决策树算法能准确的分类识别水果种类，最后是重复第二步骤，反复尝试选择不同的特征，最终调整出最优的水果分类算法。

本文共六章，各章节安排如下：

第一章，绪论。从收集的资料与实际背景出发，结合我国当前针对小户农村的果实分类系统发展现状，从实际需求角度阐述了本次设计的必要性。

第二章，系统的总体架构设计。本章主要从完整的系统角度出发，阐述总体的框架设计思路，并自顶向下逐一对各单元内容进行说明。结合理论指标与实际需求，在尽量控制成本的情况下，确定最合适的实现方案。

第三章，图像的采集与处理，对水果图像原始数据的预处理。本章根据不同种类的水果，提取不同的关键特征，并对原始图像数据进行降维。

第四章，实时分类算法的实现。本章对预处理后的水果图像，输入到预先设定的训练模型，其主要算法是基于决策树改良而来的分类算法，通过调整超参数、优化输出情况，避免过拟合，实现高效的水果实时分类系统。

第五章，测试与结果分析。本章对水果分类系统的实时性能与效率性能进行压力测试，数据来源为前期采集的数据，选出一部分原始采样数据作为测试数据，分析其最终实现的结果。

第六章，总结与展望。本章本次设计论文的最终章，包含着对本次设计的系统性回顾，是对整个系统设计的一次重新学习，分析欠缺的地方并结合当下前沿科技，提出展望。

2 系统框架设计

针对小型农户的在水果加工销售环节中的分类需求，本文设计并实现了基于树莓派的视觉识别分类系统。本章根据所选水果的具体情况，简述系统各模块的实现方法，设计树莓派的程序算法，并选择合适的器件型号，开发所用的辅助工作软件与笔记本系统操作平台，进行实际调配后，完成整个系统开发软件平台的基础搭建。

2.1 系统总体设计

水果分类系统主要用于将需要分类的水果进行实时的检测识别，先通过前端的数据采集模块获取原始信息，然后依托于优化过的实时分类算法，匹配原始图像信息与水果的实际种类，最终将系统分类完成的结果数据，以有线连接-USB 输出到上位机的接收系统，以执行下一步的操作。

系统主要分为三个模块，分别是图像采集与预处理模块、算法部分-水果分类识别模块、输出部分-分类结果输出模块。系统的整体设计框架如图 2.1 所示。

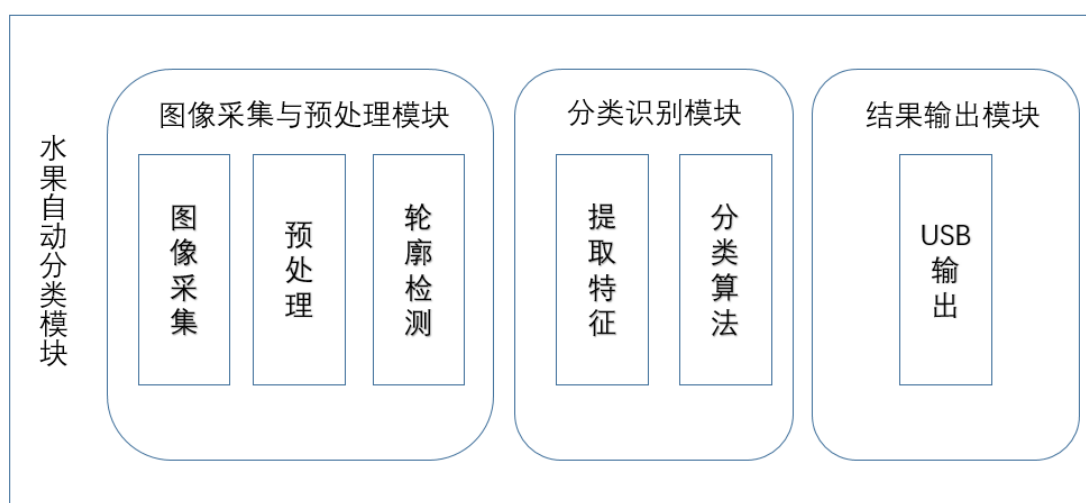


图 2.1 系统总体设计架构

其中图像采集与预处理模块的工作是利用高清摄像机直接获取原始图片，它们是系统测试用例与训练用例数据的来源，还有包含有图像数据作初步的预处理，通过一些简单的内置成熟的模块算法，初步对图片进行一些处理以降低后续树莓派的硬件运算压力，包括对原始图片进行轮廓检测、对原始图像数据进行降维。目的是减少输入的数据量，降低后端的负载，提升效率。第二部分-分类识别模块，这部分是本系统的核心模块。本文设计准备的分类算法的核心思想是决策树算法，该算法性能稳定，分类效率高，且编写简单，十分适合本次设计。首先我们通过学习决策树的理论知识，然后根据本设计的实际需求，进行改进出适合本次的设计的算法，其主要功能是接受前一模块预处理过的数据^[2]。利用特征工程，分析处理出每类水果的典型特征，并提出特征，最后通过分类算法进行分类，并将分类结果输出。

2.2 系统硬件架构

系统选用高性能的嵌入式设备树莓派(Raspberry Pi 3)作为数据处理平台，图像采集功能，通过其配套的 RASPBERRY PI CAMERA V2 摄像头实现，另外惠普笔记本一台，用于编写程序，调试算法。系统硬件整体架构如图 2.2 所示。

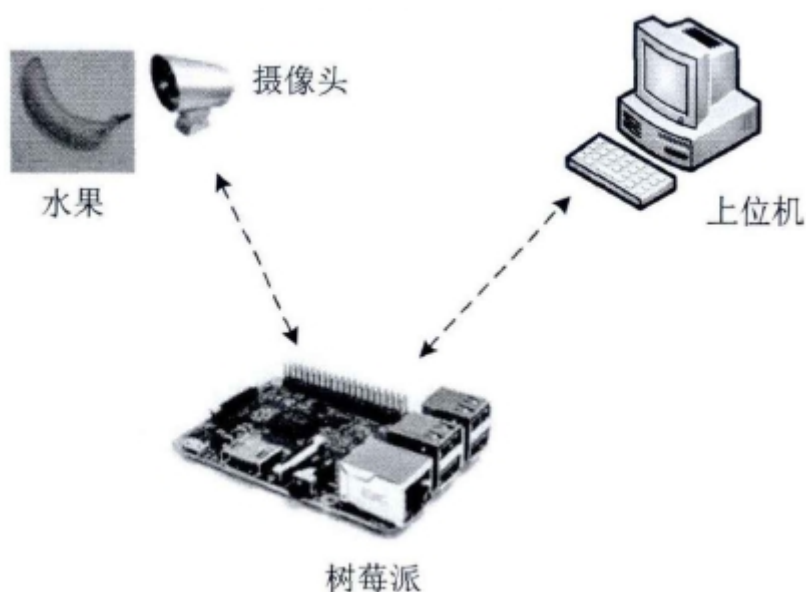


图 2.2 系统硬件整体架构示意图

2.2.1 树莓派简介

本系统使用的微处理器硬件开发平台为树莓派 (A3+), 其实物如图 2.3.所示。截止到 2020 年, 树莓派迭代已经超过四代, 目前最新的型号为树莓派 4, 但是综合成本与需求考虑, 本次设计最终选定为树莓派 A3+。



图 2.3 树莓派 A3+实物图

本次选型的树莓派 A3+，是树莓派基金会在 2018 第四季度发布，其尺寸是标准的树莓派 A+尺寸，面积小巧，重量轻便。它继承了 3 代的双频 2.4GHz 和 5GHz 高速无线网络，同时蓝牙固件升级至低功耗的 4.2 版本，最重要的是，其处理器性有了更进一步的提升，拥有一枚主频为 1.4GHz 的四核处理器，并且引入焊钳工艺，增强了散热管理能力，相对于上一代，能长时间的稳定在峰值功率，持续输出强大的运算能力，使得我们的实时算法运行更加稳定。其主要性能参数如图 2.4 所示。

Specifications

The Raspberry Pi 3 Model A+ extends the Raspberry Pi 3 range into the A+ board format.

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 512MB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2/BLE
- Extended 40-pin GPIO header
- Full-size HDMI
- Single USB 2.0 ports
- CSI camera port for connecting a [Raspberry Pi Camera Module](#)
- DSI display port for connecting a [Raspberry Pi Touch Display](#)
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input

图 2.4 树莓派性能参数

2.2.2 摄像头

本系统选用的摄像头模块为树莓派的专用摄像头模块（RPC-V2），其实物如图 2.4 所示。目前摄像头技术已经发展得十分成熟，市面拥有多种性能优越、价格低廉的摄像头，但是不一定与本系统兼容，故综合考量，最终摄像头选型为树莓派的专用摄像头（RPC-V2）。

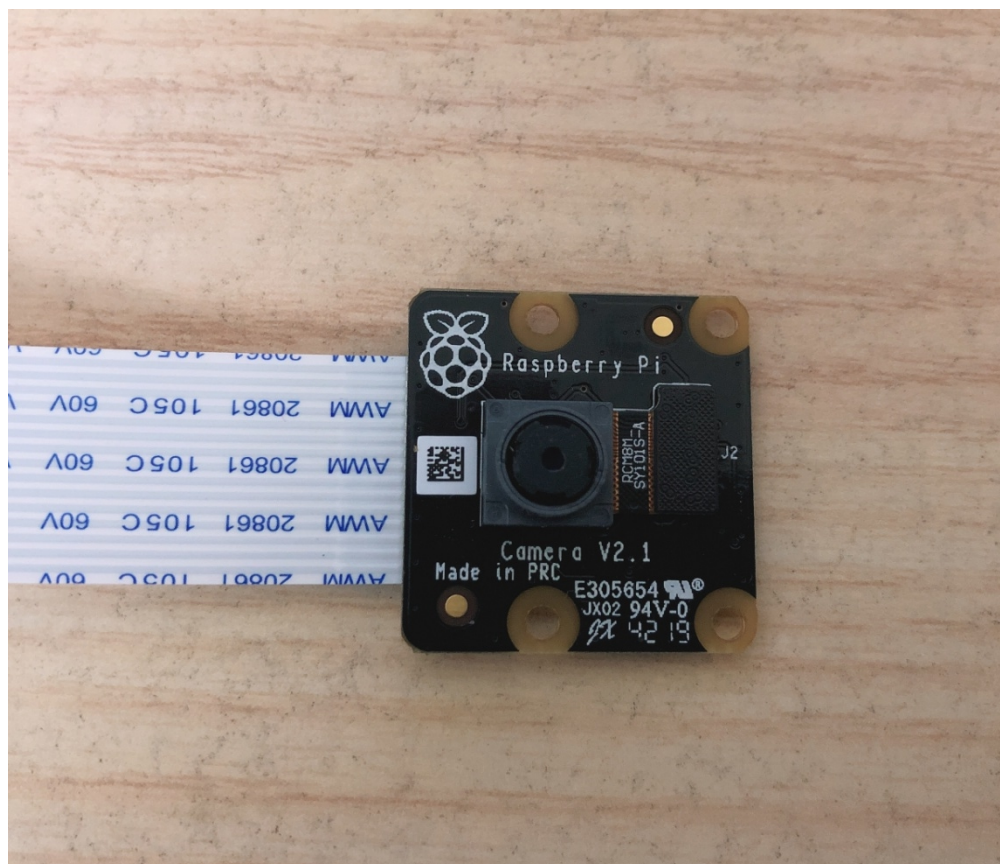


图 2.4 树莓派专用摄像头实物图

该摄像头的感光元件采用索尼 IMX219，支持最高分辨率为 3280×2464 ，支持最大的捕获帧率为 30fps。相比于一代的摄像头，二代主要解决了失焦与延时的问题，新增了自动对焦功能。其具体的性能参数如图 2.5 所示。

Features

- Official Raspberry Pi Camera, supports Raspberry Pi and Jetson Nano
- Sony IMX219 8-megapixel sensor
- Camera specifications
 - CCD size : 1/4inch
 - Aperture (F) : 2.0
 - Focal Length : 3.04mm
 - Angle of View (diagonal) : 62.2 degree
- 3280 × 2464 still picture resolution
- Support 1080p30, 720p60 and 640x480p90 video record
- No infrared filter, supports night vision
- Dimension: 25mm × 24mm × 9mm

图 2.5 摄像头性能参数图

2.2.3 笔记本电脑

本系统中的实时分类算法，其具体的编写工作主要在计算机平台上完成，其具体性能参数如图 2.6 所示。

| | |
|---------------------------------------|--|
| Windows 版本 | |
| Windows 10 专业版 | |
| © 2019 Microsoft Corporation. 保留所有权利。 | |
| 系统 | |
| 处理器: | Intel(R) Core(TM) i7-4710MQ CPU @ 2.50GHz 2.50 GHz |
| 已安装的内存(RAM): | 16.0 GB |
| 系统类型: | 64 位操作系统, 基于 x64 的处理器 |
| 笔和触控: | 没有可用于此显示器的笔或触控输入 |

图 2.6 计算机性能参数图

2.3 系统软件平台的构建

由于本设计中包含了嵌入式硬件,故系统平台需要在 Windos 与 Linux 上做一些交互。针对不同的硬件,需要安装不同的软件系统,以便完成所需的工作。

2.3.1 笔记本电脑软件平台

需要在笔记本上安装 windows 系统,以及开发软件 Pycharm。它一款商业软件,为了避免使用盗版软件的不良现象出现,我们可使用它的免费社区版,虽然后者功能较于前者功能减少一些,但是核心功能均可正常使用,可以满足本文设计的程序设计部分的需求。软件安装如图 2.7 所示。

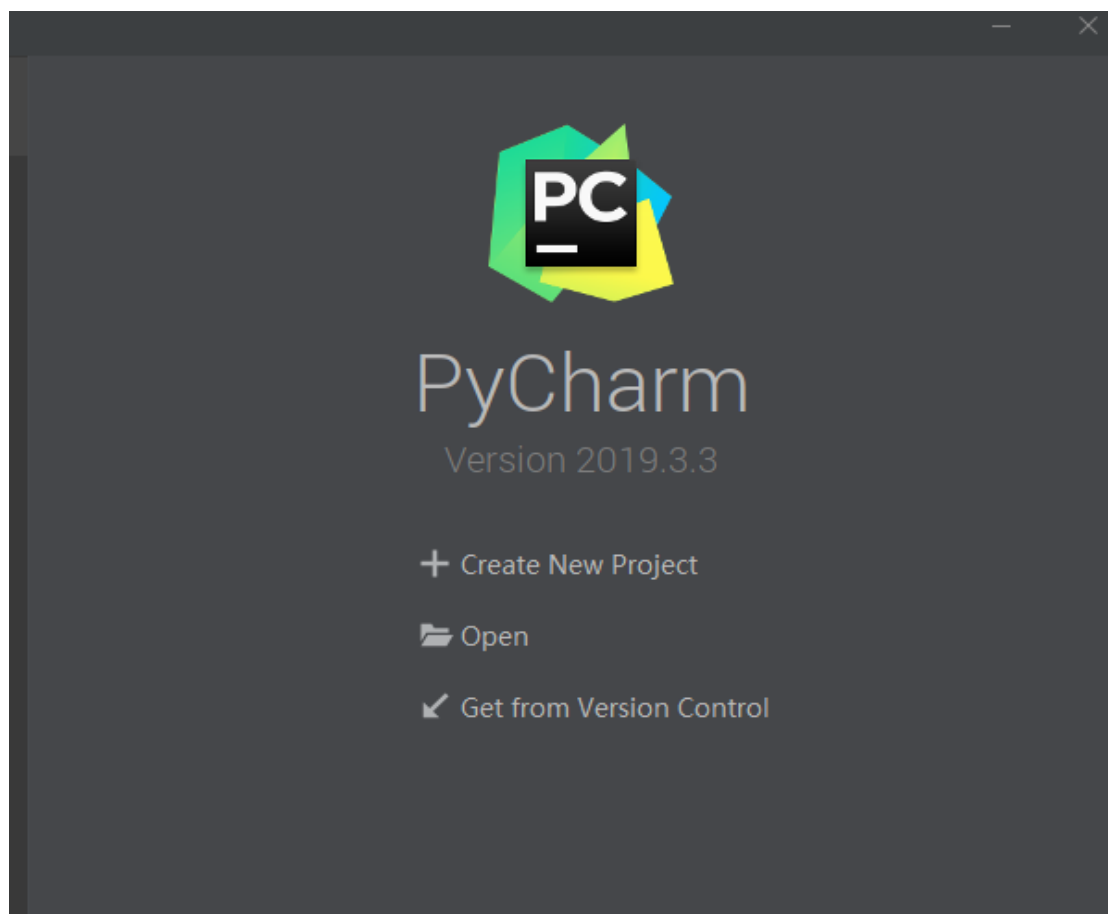


图 2.7 Pycharm 启动图

2.3.2 树莓派软件平台

树莓派需要安装专用的系统,其官方系统 Raspberry,其安装方法如图 2.8 所示。

Installing operating system images

This resource explains how to install a Raspberry Pi operating system image on an SD card. You will need another computer with an SD card reader to install the image.

Before you start, don't forget to check [the SD card requirements](#).

Using the Raspberry Pi Imaging Tool

Raspberry Pi have developed a graphical SD card writing tool that works on Mac OS, Ubuntu 18.04 and Windows, and is the easiest option for most users as it will download the image and install it automatically to the SD card.

- Download the latest version of [Raspberry Pi Imager](#) and install it.
- Connect an SD card reader with the SD card inside.
- Open Raspberry Pi Imager and choose the required OS from the list presented.
- Choose the SD card you wish to write your image to.
- Review your selections and click 'WRITE' to begin writing data to the SD card.

图 2.8 树莓派系统安装方法图

2.3.3 Open Cv 概述

Open CV 是一个用 BSD 协议开发源代码的计算机视觉库^[3]，它起初是美国英特尔公司开发，截止到如今已有 20 多年的历史，在实时分类与图像处理，有着十分广发的用途。现在由美国 WG 提供主要支持。

该库支持多种平台，故本系统的实时分类算法主要也依托于它来完成，其版本分发情况如图 2.9 所示。

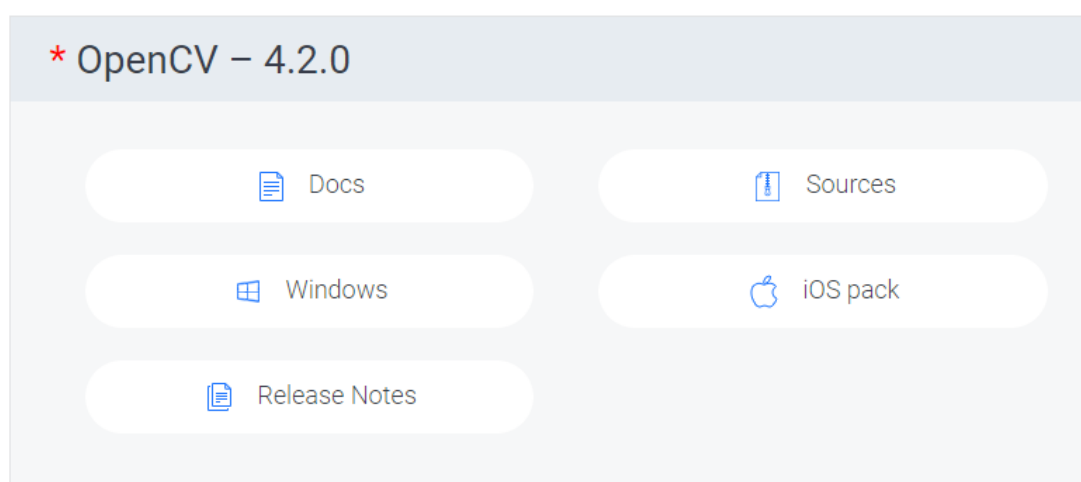


图 2.9 Opev Cv 版本分发图

2.4 本章小结

针对小农用户对水果分类的实际需求，对应设计了嵌入式水果自动分类系统。本章完成了对整体系统进行了总体框架设计的叙述。包括从硬件选型的理由，出于性能 and 功耗综合考虑，我们选择嵌入式处理器选择树莓派（A3+），摄像头采用树莓派专用摄像头，将它们作为本设计系统的软件运行平台载体。从开发程序的平台软件上，针对不同的设备安装了不同的系统平台，并配置程序的开发编译环境与所需的图像识别库，以完成开发平台的搭建，最后是分类算法，虽然当下深度学习技术非常火热，基于深度学习的图像分类技术也是研究的热点，但是综合本次设计的分类需求与分类算法开发难度，本此设计最终选用较为传统的决策树分类算法，在其基础上改造出本次设计所需的水果分类算法。

3 图像的采集与预处理

本章选取生活中常见的 5 种水果作为本设计的研究对象，首先需要采集一定量的样本图片，然后对其进行预处理。为了提取有效的水果图像数据特征与降低分类系统的运算压力，需要把彩色样本图片变成灰度图片，再二值化^[4]

3.1 水果图像的采集

水果图像的采集工作通过树莓派专用摄像头完成。该摄像头的性能参数，例如分辨率、传感器尺寸已在上文介绍。一个分类系统的最终建模好坏程度，不仅和算法本身的逻辑处理方式有关，更重要的是前期的训练数据的质量会直接影响分类器能优化的最后精度。故水果图像的采集是本文中重要的一步。

一个好的图像采集需要性能优异的摄像头与合适的环境光装置。环境采光装置是摄像头采集原始图像的前设条件，优秀的采光方案会让最后的训练数据也十分优秀，从而使得分类器的训练过程变得十分顺利。相反若是没有设置环境光装置，直接在没有人为干预的自然光环境下直接拍摄图像，可能会使得部分图像过曝或者失真，更加难以让人处理的是图像中可能存在部分肉眼难以识别的偏色，这会直接影响后续的灰度和二值化处理，尤其是对水果的轮廓处理。

好的光源应该满足一下几点，第一是照明的流明度足够，足够的亮度才能保证摄像头的传感器能捕捉到足够的细节，第二是照明的均匀度要有保证，由于本设计的图像都要进行灰度处理，颜色不均会使得灰度化变得模糊不均，第三是照明系统要有良好的散热模组，以主动散热最佳，因为本设计需要拍摄大量的照片，照明系统长时间在满额功率下输出，这会使得照片系统温度非常高，故必须配备散热模组。综合上述条件考虑，本次设计决定选择工业级的 LED 灯，并通过自己的手工改良在其后面板上装配主动式的风扇模组。

本文实验所选择是水果是市场非常常见的 5 种水果，分别为：苹果、香蕉、草莓、葡萄、西瓜。对其中的每张水果均采样 100 张，合计 500 组，将这些水果图像参数组成训练数据集，用于后续的基于决策树的实时水果分类算法的实现与测试。5 种水果实物如图 3.1 所示。



图 3.1 水果实物图

3.2 水果图像的预处理

经过树莓派专用摄像头直接拍摄的图片，其原始图片包含的数据量十分巨大，每张图片的体积达到 10 余兆，且每张图片均为彩色图片，由于本文设计所编写的程序并没有关于颜色分量的处理，所以原始图片中的彩色信息对于本次分类识别的结果是完全没有影响的，但是处理彩色照片会比处理黑白照片消耗更多的处理器资源，对我们的图片数据处理会带来十分巨大的压力。

更重要的是本文设计所选用的算法是基于决策树的分类算法，与目前市场上火热的深度学习算法不同，本文选用的算法不存在隐藏的参数模型，无法自动的从图像中提取数据，如果直接输入原始图像到分类系统中是无法顺利完成模型训练的，我们先对图像进行一定的预处理，通过一些前置手段将原本包含许多信息内容的原始图像，变化成只带有明显特征的图像，这样分类算法就可以根据这些特征来识别水果的种类。

另外由于本设计采用的嵌入式处理平台，即使我们换用更加性能优异的算法，但是面对如此庞大的数据，如果直接输入模型训练，会导致训练时间异常久，且机器存在宕机的风险。

最后就是图像的采集并不是一个严格的标准化流程，尽管我们额外配置了环境光源，选择了最合适的摄像头传感器，但是仍然无法避免每张图像都是一张合格的图像，如果拍摄的同类水果图像特征差异过大或者不同类的水果的差异特征不足，均导致分类失败。为了避免上述这些异常情况的出现，我们需要对摄像头直接采集的高清原始图片进行一定的降维^[5]本文采用的方法主要是图像灰度化、图像二值化。

3.2.1. 图像灰度化

图像灰度化，即将原本的包含彩色信息的图片转成仅包含亮度信息的图片。该操作的目的是简化原始图像所包含的信息，灰度变换可以主动性的将图像中需要关注的地方进行着重色彩，而对不需要的特征进行抑制，使得经过处理后的图像与人眼看上去的特征关注处对应。我们把这种图像数据的转换过程被称为图像灰度化^[6]色图片经过灰度化后，图像包含的信息会显著压缩，但是原有的特征不会消失，这样既不影响后续的分类训练，又可以提高训练效率。本文中对彩色图片进行灰度转换的流程如图 3.2 所示。

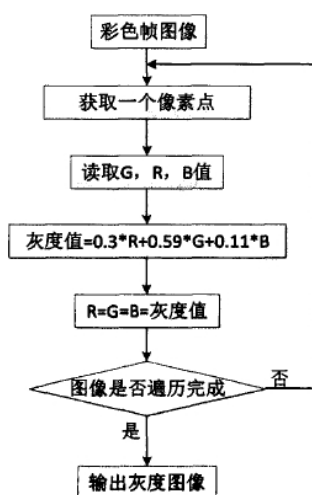


图 3.2 图像灰度化流程图

可以从上述的流程图看到，我们原始摄像机拍摄的图像为彩色图像，也可以成为 RGB 图像，其红色、绿色、蓝色，三通道的信息数据是完整的，正是红绿蓝这三种原色的组合，才使得我们能看见不同的彩色。灰度化的处理逻辑是，首先对每张相片以左上顶点为原点，建立直角坐标系，然后自上到下，从左到右，依次处理每一个像素点，将原有的红绿蓝三通道的饱和值，分别乘以 0.3、0.59、0.11 的系数，这实质就是降低了每一层颜色通道的饱和度，从视觉观看的感觉就是，图片失去了颜色变成灰色的一片，故我们将这个变化过程称为灰度化。

因为对原始彩色图像直接进行图像处理的时候，计算机程序需要对每一层通道到数据依次处理，这样会导致系统的运行速度缓慢，而且我们编写的程序没有针对颜色分量的处理程序，所以即使对每一层的通道信息进行了识别也不会对分类的精度有任何提高，只会徒增性能开销且额外增加系统负载，故灰度化图片是本次设计中一个不可缺少的部分，而是后续进行其他处理的必须操作。

3.2.2 图像二值化

图像二值化，即将图像中像素点的灰度值设置成 0 或者 255。经过二值化转变后图像将变得黑白分明，其图像灰度信息只包含亮或不亮两种信息，类似比特信息里的 0 和 1，作此种变换的目的是，更进一步的压缩图像体积，为后续的模式训练作预备，同时经过这种变换后的图像，其特征表现也会增强，便于提高模式的训练效率。针对全彩图像的灰度化处理有三种，一是最大数值法、二是通道分量法、三是均通平均法。

图像二值化的流程如图 3.4 所示。

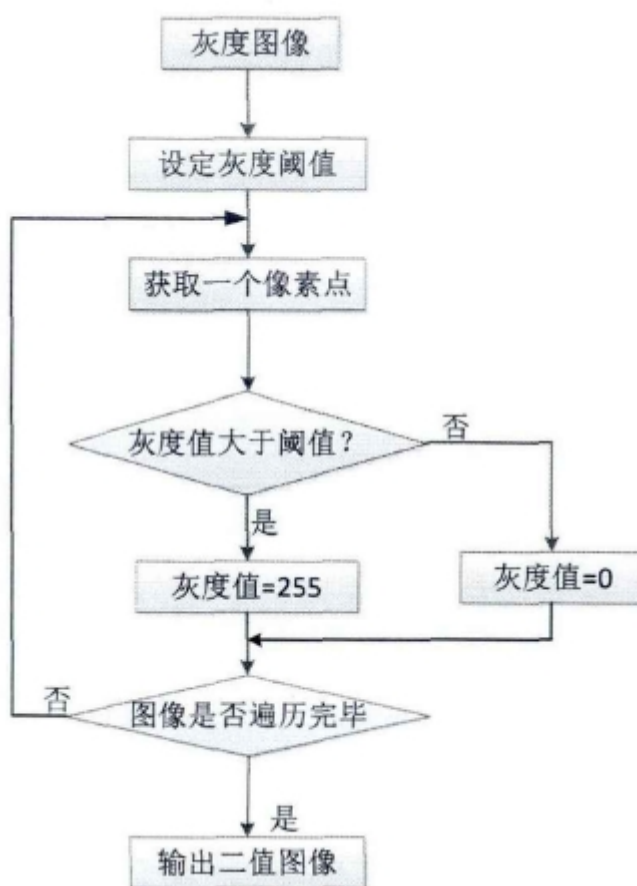


图 3.3 图像二值化流程图

本文灰度化的处理流程如上，首先是设定好灰度阈值，综合本文采集图像的现场光源条件，本次阈值设置为 150，然后扫描原始图像的文件夹，将每一张超过阈值的图像进行处理，将其大于 150 的部分直接上升到 255 即纯黑，将低于阈值的部分全部置零处理。将所有的采集图片处理完成后再进行二值化处理，最终预处理后图像如图 3.4 所示。



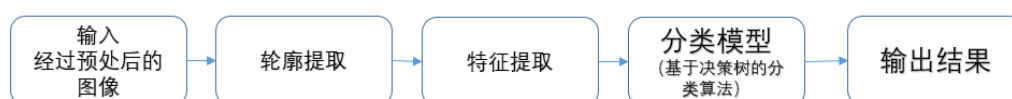
图 3.4 灰度图片二值化

3.3 本章小结

本章主要介绍了图像的采集与预处理，其中图像采集需要做到保证采集的同类图片尽可能做到特征一致，不要产生较大的偏差，需要在合适的环境光照条件下完成图片的采集工作。预处理是为后面分类算法的实现准备的，因为在本次设计所选择硬件不具备处理完整彩色图像的能力，故我们需要提前对原始图像进行预处理，提前抽取出特征便于分类训练。最后我们选取 5 种常见的水果，通过树莓派的专用摄像头批量采集原始图片，然后对原始图像进行预处理，在经过图像灰度变幻、二值化处理后，构成原始训练集。

4 水果实时分类算法的实现

本章将经过预处理后的图像，进行简单归并处理，利用滤波算法去除一些噪点数据，便于下一步完成特征提取的工作。对经过预处理后数据再次选取特征，从而得到具有一定区分度的特征数据后，经过数学归一化处理，最终完成分类算法的编写。先在 X86 平台上模拟测试一些数据，待确定模型算法无异常后，将并移植到 linux 开发平台-树莓派硬件，最终从而实现水果实时分类算法的实现[7]其主要流程如图 4.1 所示。



4.1 分类算法实现流程图

4.1 水果图像轮廓提取

进行预处理后的图像，虽然颜色信息减少了，但是图像整体尺寸仍然很大，故需要图像进行水果轮廓提取，在水果提取之前需要对图像进行去噪处理。

在通过摄像头获取图像的过程中，可能由于光线变化或者其他一些不可预知的因素导致图像会带有一些噪点，这些噪点数据会直接影响最后的分类效果，故我们需要提前对这些噪点数据进行预处理。

水果轮廓的提取可以分为两个步骤，第一个步骤是水果边缘的提取，第二个步骤是在前者的基础上，进行轮廓的切割，提取出有效的黑色区域，去除白色的背景区域，以达到将近一步压缩输入图像信息量的目的。

在 OpenCV3.1，我们使用 cvCanny 函数，对图像进行第一步骤的操作，完成边缘切割，再利用 cvFindContours 函数^[8]，完成第二部分，水果轮廓的提取操作，其水果边缘切割与轮廓提取流程如图 4.2 所示。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：

<https://d.book118.com/128010036077006051>