

基于机器视觉的颜色识别分拣机器人的设计

专业：通信工程 年级：2015 级 学生：陈强强 指导教师：孟圣慧

摘 要

随着科技的进步与自动化水平的不断提高，越来越多的工业分拣机器人被运用在物流与生产制造中，代替人完成各种物料的分拣，早期的分拣机器人通常以示教编程的方式为主，虽然这种示教编程的方式可以使机器人达成分拣的目的，但是如果有一些物料摆放位置杂乱的特殊环境下工作就会影响机器人分拣的准确性，如何让机器人适应在不同条件下都能够完成分拣工作是工业机器人智能化的关键，将机器视觉的颜色识别技术应用在工业分拣机器人上，能够使机器人具有颜色辨识的能力，并且代替人的眼睛完成不同的工作。因此，基于机器视觉的颜色识别分拣机器人的研究具有现实意义。

通过学习颜色识别的理论知识与相关算法，结合了定位追踪程序的研究，实现了基于机器视觉的颜色识别分拣机器人的设计，机器人所实现的功能为以纯色的积木块作为识别目标，在自然环境下能够准确地识别出指定颜色的积木，并且通过目标定位追踪使机器小车能够移动到目标附近，控制机械臂完成抓取动作并放置在指定颜色的盒子中，实现了分拣的功能。在硬件设计上，采用了树莓派 3B+ 为主控模块并指导各子模块完成相应工作，将采集到的图像信息传输到主控模块进行分析处理，然后控制机器小车完成指定动作，软件设计是建立在 Raspberry Pi 系统环境下以 Python 作为编程语言，使用了 OpenCV 对图像进行处理，实现了颜色识别与目标定位追踪等核心算法，并应用在机器人小车上，使系统能够实时获取图像并驱动小车分拣不同颜色的积木。

通过对系统的不断调试与数据分析，实现了在自然环境下对不同颜色积木块的识别与分拣，达到了预期的目的。

关键词：机器视觉 颜色识别 OpenCV 目标追踪 分拣系统

目 录

1 绪论	3
1.1 研究的背景和意义	3
1.2 国内外发展概况	4
2 颜色识别的相关理论	5
2.1 色彩	5
2.2 HSV 颜色空间模型	5
2.3 OPENCV	7
3 硬件系统设计	9
3.1 硬件总体结构设计	9
3.2 主控模块	11
3.3 摄像头模块	13
3.4 舵机模块	14
3.5 电机与电机驱动模块	15
3.6 电源模块	17
4 软件系统设计	19
4.1 软件运行环境的搭建	19
4.2 系统程序设计	22
4.3 系统调试	32
结 论	34
参考文献	35

1 绪论

1.1 研究的背景和意义

随着科技的进步与国民生活水平的不断提高,人工智能已经越来越受到人们的关注。人工智能是一个包含多个领域的科学,如机器学习,计算机视觉等等,它能够代替人完成许多复杂的工作,因此成为当代的热点话题^[1]。而机器视觉作为人工智能的一个重要的分支更是备受关注,同时也是一个非常具有研究价值的领域^[2]。机器视觉也能够根据其工作原理划分出不同方面的应用,其中颜色识别就是其主要的的应用之一,所谓颜色识别就是能够根据物体表面颜色特征通过一系列的算法进行颜色辨识。自从这种基于机器视觉的颜色识别概念的提出便得到了许多爱好者的研究,使其不断发展,但是到目前为止,这项技术还是没有得到很好的应用。

伴随着人工智能与工业自动化的高速发展,工业上分拣机器人也是越来越普及了,很快将成为现代工业生产中不可或缺的工具,其功能是将不同类型的物料按照一定的标准进行分类抓取与放置,因此,分拣程序一般为识别、定位、抓取和放置等4个阶段^[3]。而颜色识别的分拣机器人就是以颜色为区分的标准对不同颜色的物体进行分拣,颜色分拣机器人可以运用在多个领域之中,特别在一些生产不同颜色产品的厂家中,例如布料厂、染料厂、积木制造厂。现如今,网商的发展越来越成熟,进而推动了物流业的发展,而分拣是物流体系中一个十分重要的环节,如何高效、精确地分拣是一个值得思考的问题^[4]。颜色识别分拣机器人同样适合于物流业,可以在包裹上贴上不同颜色的标签,机器人就能通过颜色识别将这些包裹分类包装。

目前市面上的颜色识别分拣机器人依然是以颜色传感器识别为主,使用颜色传感器来进行识别虽然比较容易实现,但同时存在着识别精度低、距离短的缺点,因此并不能很好地应用在工业生产与物流管理等行业中,但是,如果将机器视觉的颜色识别与分拣机器人相结合就能够更好地改善精度与距离的问题,也就更适合在自动化产业中得到应用,通过基于机器视觉的颜色识别可以将待检测目标的颜色信息提取出来,再通过主控模块发送预设的控制信号驱动下位机完成相应的分拣动作,这种基于机器视觉的颜色识别分拣机器人能够比员工手动分拣更有优势。首先,通过机器视觉的颜色识别可以进行长时间不知疲惫的观测,并不会出现情绪等心理因素导致它的工作效率降低。其次,分拣机器人对工作环境没有特别严格的要求,即使在一些空气稀薄的密室中也能正常工作。因此,研究基于机

器视觉的颜色识别分拣机器人具有相当的学术价值和实用价值。把基于机器视觉的颜色识别技术引入到物料分拣控制系统中,未来的物流行业和工业生产就能更加高效与智能化。不仅降低了工人的劳动强度,而且还推动了自动化产业的发展,从长远的角度来看该研究具有巨大的现实意义^[5]。

1.2 国内外发展概况

在国外,对于机器视觉和工业分拣机器人的研究已经趋于成熟,各个世界上知名的机器人生产商所研发的机器人都已经相当的智能化,这些机器人不仅控制精度高,而且具有很快的分拣速度。在国外著名的机器人生产商有川崎、新松等^[6]。近几年来随着人工智能的发展,工业分拣机器人也是逐步走向智能化。机器视觉作为人工智能的一个重要的分支,更是被各研究所和相关企业所重视,并且不断地研究。而在国外,随着机器视觉和分拣机器人技术的不断完善,促使了许多人提出是否能将机器视觉应用在工业分拣机器人上的疑问,将视觉中所看到的画面通过一些判断标准来实现分拣,例如颜色识别是通过不同颜色进行物料区分。将这种机器视觉的颜色识别分拣机器人运用在生产或物流等行业中,可以很大程度地减少雇佣成本、提高工作效率、推动智能化产业发展^[7]。目前,在美国、日本等一些发达国家,这种基于机器视觉的分拣机器人已经广泛应用在物流、布料生产、食品加工等行业中,而基于机器视觉的颜色识别作为主要的发展对象之一更是在分拣目标的判断中得到了充分的利用。

在国内,无论是机器视觉还是分拣机器人都是起步较晚。上世纪 90 年代机器视觉技术才逐步得到应用,在此之前国内对于自主研发具有专利性的图像处理算法研究几乎为零。但是后来随着中国科技的进步及制造业对智能产品的需求,机器视觉在国内得到了不断发展^[8]。到 2008 年,国内已经注册了几百家研究机器视觉的公司,后来随着中国制造业的发展及传统制造业的转型,国内对于分拣机器人的需求也是日益增加,而基于机器视觉的颜色识别技术的发展促进了国内的一些科研机构对分拣机器人的不断研究,这些研究所通过学习国外的相关技术并自主研发与创新^[9]。

因为机器视觉行业本身就属于一个全新的领域,并且机器视觉的相关技术在国内还没有得到全面的普及,而颜色识别作为机器视觉的主要应用之一更是没有得到很好的开发,导致其在自动分拣行业的应用并不完善。因此提高识别精准度与分拣速度是未来机器视觉分拣机器人的发展方向。

2 颜色识别的相关理论

2.1 色彩

当光照射在物体表面时，就会反射出物体的表面颜色，而色彩是色光照射后刺激人眼视网膜上的神经细胞而产生的心理上的视觉特性^[10]。色彩在自然中无处不在，由于其具有相对独立性，因此我们可以将色彩作为区分不同物体的一种判断条件。

CIE（国际照明委员会）将红色、绿色、蓝色称为光的三原色，而其它颜色都是由三原色按照某种比例搭配而成的。此外，CIE 还定义了一种颜色测量的方法，即光电积分法，它主要利用三刺激值的算法来进行颜色测量，所谓三刺激值就是指某种颜色的形成分别需要三原色刺激量的大小，其检测公式分别对应于如下公式（2-1）、公式（2-2）、公式（2-3）^[11]。

$$r = \frac{R}{R+G+B} \quad (\text{公式 2-1})$$

$$g = \frac{G}{R+G+B} \quad (\text{公式 2-2})$$

$$b = \frac{B}{R+G+B} \quad (\text{公式 2-3})$$

式中， R ， G ， B 分别表示形成被测颜色所需红绿蓝的刺激量， r ， g ， b 分别为被测颜色红，绿，蓝的三刺激值。

2.2 HSV 颜色空间模型

颜色空间模型主要是以某种规范标准来解释颜色，一般是以数学的方法来描述某种颜色，而 HSV 颜色空间模型是目前运用最广泛的颜色空间模型之一，它是通过色调 H 、饱和度 S 、亮度 V 来描述一种颜色，这很符合人眼辨识颜色的方式。

HSV 颜色空间模型中的三个描述颜色的分量具有相互独立性，其中亮度表示物体周围光线的明暗程度，它与色彩无关，而饱和度与色调是描述色彩的主要方式，这种判别色彩的方式与人眼感知颜色的方式十分相近，因此利用 HSV 颜色模型对图像分析处理的方式非常适用于机器视觉的颜色识别这样以机器代替人眼的应用中，饱和度描述的是一种颜色的浓度，也可以说是白色在其它颜色成分中所占比例的大小，这个比例越大，则颜色越浅，饱和度也就越低；反之，比例越

大则饱和度越高，色调也叫色相，即颜色的相貌，直接地描述光照在物体表面反射回来的颜色，例如红色、蓝色、绿色等。

通过 Munsell 三维颜色空间坐标系如图 2-1 所示可以体现 HSV 颜色空间中各分量之间的独立性与均匀性，Munsell 首次提出了将颜色用色调、饱和度、亮度表示的方式，由于这种颜色感知方式与人们主观上对颜色的判断具有相同的特点，因此这种颜色空间模型更能够被人们所接受，Munsell 颜色空间能够在相同色调下通过改变亮度和饱和度扩展出许多不同的颜色。因此，可以很好地应用于相似图像之间的比较^[12]。

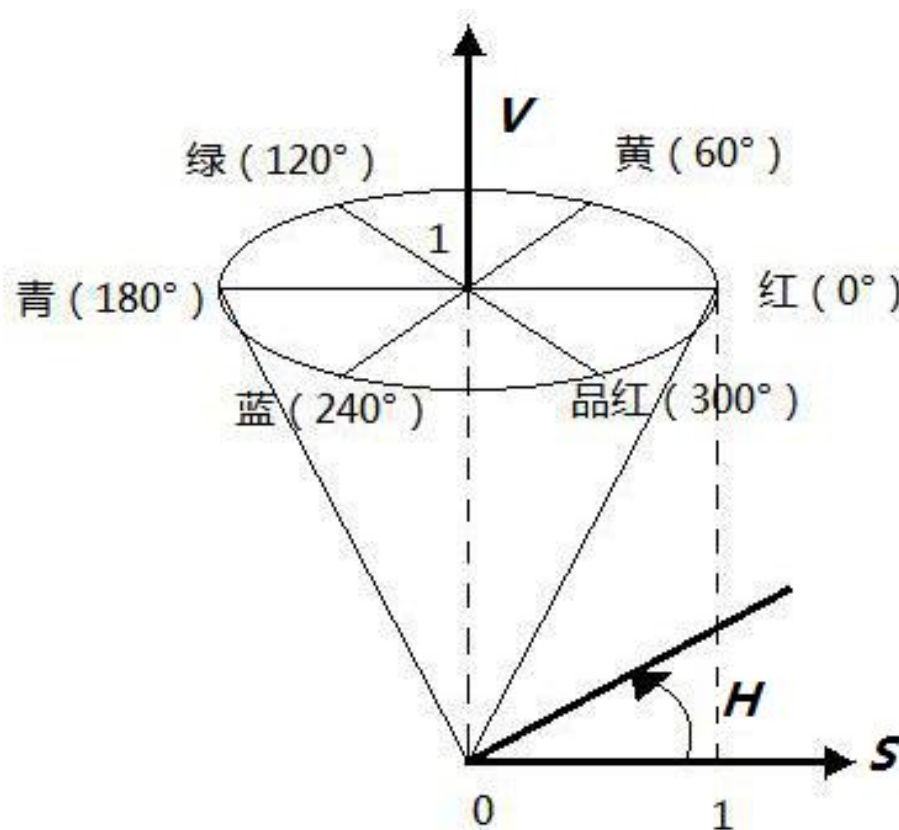


图 2-1 Munsell 三维颜色空间坐标系

其中 V 轴表示亮度，一般采用百分比的形式来表示，亮度 0% 时为黑色，100% 为白色，通过亮度的调节可以改变光线明暗级别。而 S 轴表示饱和度，即颜色的鲜艳程度或纯度，它也是用百分比来表示的，百分比值越大，色彩的纯度越高。 H 轴表示的是色调，在区间 $[0^\circ, 360^\circ]$ 中用角度来表示，其不同的角度表示不同颜色。

机器人小车对图像的颜色识别处理主要是建立在 HSV 颜色空间上，通过将摄

像头模块采集到的 BGR 格式原始图像转换为 HSV 格式,能够更好地根据人们主观上感知的颜色进行图像的阈值化处理。摄像头采集到的图像是以数字化的方式存储在计算机内,其中亮度和饱和度都是以 0-255 种强度值表示,色调则以 0-180 种类型表示,由于颜色识别的目标是积木块,所以针对积木块在多种环境中测试获得各颜色的 HSV 分量范围如表 2-1 所示,其中红色的 HSV 颜色分量比较特殊,由两个阈值区间组成,在后文软件设计中会对其在图像处理中的解决方法做进一步解释。

表 2-1 积木块的 HSV 颜色分量表

	红色		橙色	黄色	绿色	蓝色	紫色
Hmin	170	0	4	19	34	95	117
Hmax	180	4	10	33	55	115	151
Smin	110		100	80	40	140	45
Smax	255		255	255	255	255	255
Vmin	60		80	120	40	80	55
Vmax	255		255	255	255	255	255

2.3 OpenCV

OpenCV 是由英特尔公司开发的开源平台,它是由 C 和 C++编写的计算机视觉库,其内部包含了大量图像处理的相关函数,此外还提供了 Python、Java 等接口,目前已经广泛应用于监控、游戏界面处理、医学检测等领域。OpenCV 具有移植性、通用性、程序简单、运算速度快等优点,且能够适应多个主流的操

作系统，OpenCV 开发平台几乎支持所有格式的图片，因此非常适合于机器视觉领域的研究^[13]。

OpenCV 视觉库中包含了 500 多条函数，从最初的图像采集存储与图像滤波等预处理，到后期的各种图像处理函数，例如二值化、阈值化、边缘检测等，通过对图像进行各类数学算法演算获取像素信息。OpenCV 具有一套完整的图像处理体系，因此获得了广大视觉研究爱好者的追捧。机器人小车的颜色识别算法就是通过 OpenCV 所提供的通用算法对采集到的图像进行处理分析。

3 硬件系统设计

3.1 硬件总体结构设计

机器人小车的硬件由多个不同的模块组成,通过各自完成自己的分工来实现目标颜色的识别与分拣的过程,通过模块的划分设计便于后期的调试。硬件系统主要由主控模块、摄像头模块、舵机模块、电机驱动模块、电源模块、机械手臂和履带式底座等构成,将硬件组装完成后机器人小车的成品主视图如图 3-1 所示,侧视图如图 3-2 所示。下文会对各模块的功能与参数配置进行详细地阐述。

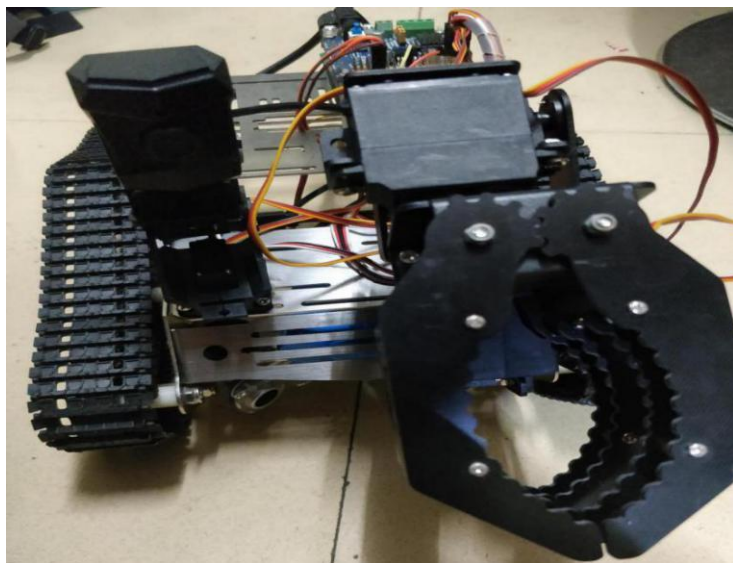


图 3-1 机器人小车主视图

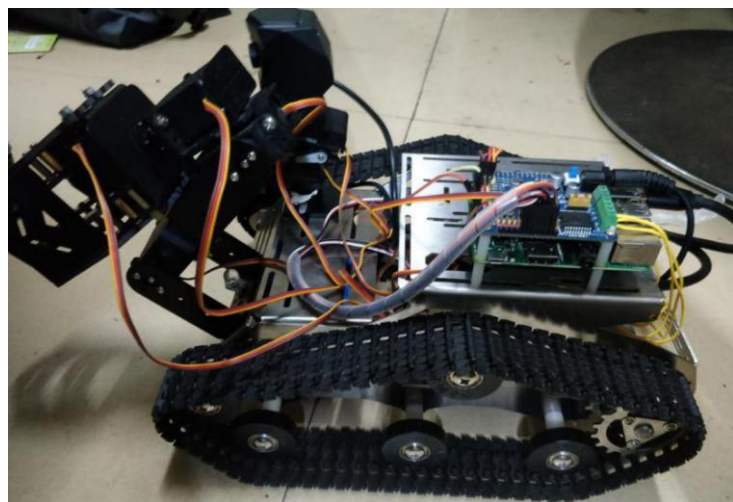


图 3-2 机器人小车侧视图

硬件系统各组成部分主要功能如下：

（1）主控模块：该模块作为系统的整体核心处理模块，具有集中控制的功能，并且能够根据用户的意愿通过控制摄像头模块采集图像信息、处理分析数据并将所得到的结果以发送控制信号的形式驱动舵机、电机等模块进行相应的动作，进一步实现系统的功能。因此，主控模块是整个系统的“大脑”。

（2）摄像头模块：该模块是系统的图像数据采集端，由控制信号启动模块的运行，并实时采集图像信息，并将采集到的每一帧原始图像送入主控模块进行处理。

（3）舵机模块：该模块主要是在系统识别到目标物体并需要机械臂完成指定动作时，由主控模块发送的工作信号驱动舵机转动指定角度从而带动机械手臂的弯曲与抓取物体或者放置物体。

（4）电机驱动模块：该模块作为系统的驱动端，由电源模块提供的电能作为动力，并将接收到主控模块的控制信号作为电机驱动的条件，驱动直流减速电机转动带动履带转动实现机器人的移动。此外，主控模块还能够控制电机驱动模块来调节机器人小车的移动速度。

（5）电源模块：该模块主要为主板与其它子模块提供满足工作条件的电能，系统硬件结构图如图 3-3 所示。

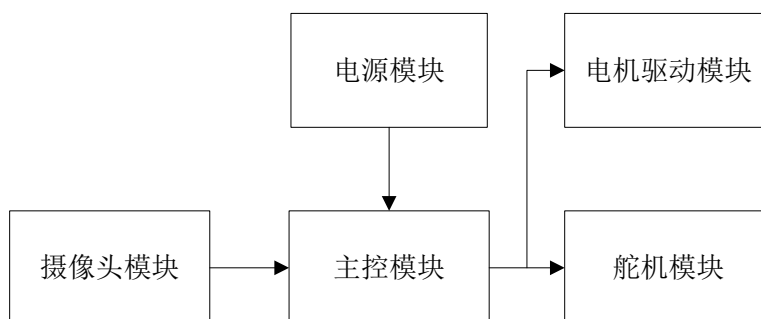


图 3-3 系统硬件结构图

基于机器视觉的颜色识别分拣机器人的设计具有半自动的特点，首先使用者通过按键设置抓取类型为单次抓取或多次抓取以及放置在何种颜色的盒子中，并选择抓取目标的颜色，主控信号接收到按键信息时调用颜色识别程序，通过摄像头采集到的图像数据进行图像处理分析，若未识别到目标则控制小车旋转并不断采集图像数据来寻找目标物体，当识别到目标则驱动小车移动到目标所在位置由机械臂完成抓取动作，然后小车自动旋转寻找预设颜色的盒子，若找到相应的盒子驱动小车移动至盒子一旁，由主控模块控制机械臂完成放置动作，若未识别到相应的盒子则停止寻找，等待下一个命令。

在硬件的连接上首先将 PWR. A53 驱动扩展板对插树莓派 3B+主板对其进行功能的扩展, 摄像头连接树莓派 3B+主板上的 USB 接口, 驱动小车移动的两组电机连接 PWR. A53 的电机驱动芯片 L298P 四个输出端 M1+、M1-、M2+、M2-, 云台上的两个舵机与机械臂的四个舵机分别接入 PWR. A53 提供的电机扩展接口, 接着将烧录了树莓派系统的 TF 卡插入树莓派 3B+主板的 MicroSD 卡槽中。

3.2 主控模块

主控模块是硬件系统中最核心的部分, 它主要实现了对各子模块的控制, 包括获取采集的数据、分析处理数据、发送工作信号驱动下位机运行, 在整个系统中起到了整体把控的作用, 若某个子模块出现了故障也能在主控模块中得到反馈。因此, 主控模块的选择十分重要。

3.2.1 主控模块的选型

主控模块采用了树莓派 3B+开发板, 该开发板具有体积小、重量轻的特点, 但是功能却是十分强大, 拥有较好的标准模块扩展性, 同时具有一些计算机的基本功能, 最主要的是价格适中, 在同级别的开发板中所发挥的性能也是较好的。此外, 树莓派 3B+还具备了多规格的外部接口, 为系统开发提供了更大的空间。

3.2.2 树莓派 3B+的特性

树莓派 3B+是于 2016 年发行的, 这是一个微型的计算机, 相当于一个完整个人电脑的小型化形式。树莓派 3B+拥有用作普通个人计算机所需的所有组件。最重要的一点是该主板采用 64 位 Broadcom BCM 2837B0 与 ARM-8 四核处理器。树莓派 3B+有 4 个 USB 接口, 并对热插拨和过流进行了保护, 使树莓派运行更加稳定, 而且以 MicroSD 卡作为硬盘, 插入到设备读取器中, 这样做的好处是允许每个用户都可以根据当前的实际需要和微型计算机的使用方式来调整卡的容量, 还能防止小车在行进中带来的振动对系统运行的影响^[14]。树莓派 3B+的主要参数如下:

- (1) 采用的 SoC(System on Chip)是博通 Broadcom BCM2837B0
- (2) 搭载了集成四核 ARM [Cortex-A53 \(ARMv8\) 64 位@1.4GHz 处理器。](#)
- (3) 内存: 1GB LPDDR2 SDRAM。
- (4) 无线网络: IEEE802.11 AC 无线网卡及 2.4GHz/5GHz 双频 WiFi。
- (5) 有线网络: 千兆以太网(通过 USB2.0 通道, 最大吞吐量 300Mbps)
- (6) 蓝牙: 低功耗蓝牙 4.2 适配器。

(7) 最大的驱动电流增加到了 2.5A。

(8) 主要接口：HDMI，3.5mm 模拟音频视频插孔，4 个 USB2.0 接口，MicroSD 插槽。

(9) 其它接口：拥有 40 针 GPIO 口。

(10) 尺寸：82mm×56mm×19.5mm。

树莓派 3B+外形封装图如图 3-4 所示。



图 3-4 树莓派 3B+外形封装图

树莓派 3B+的原理图如图 3-5 所示。

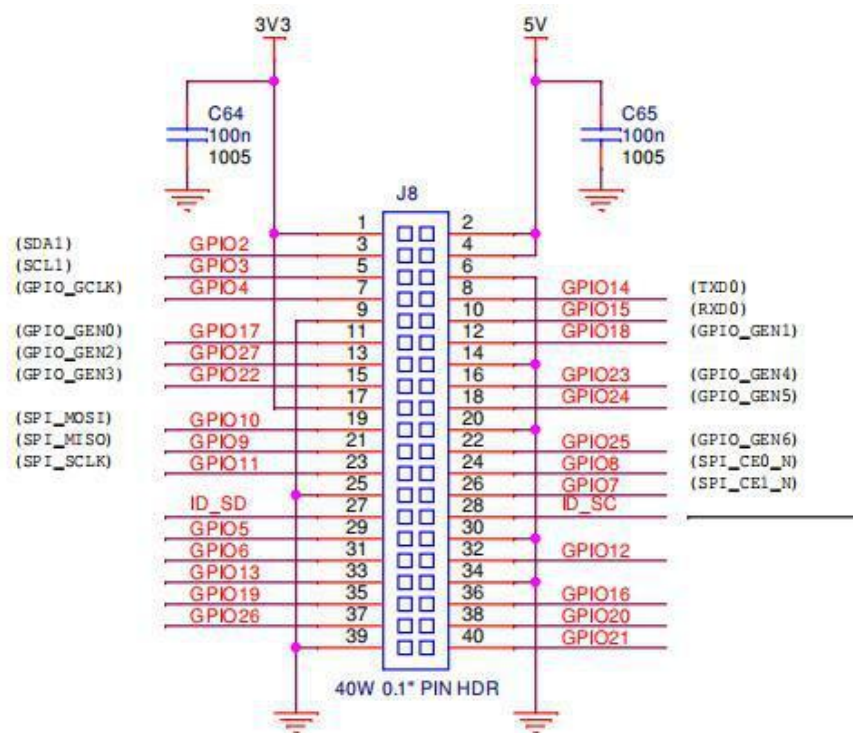


图 3-5 树莓派 3B+原理图

树莓派 3B+所提供的多根 GPIO 接口能够使 PWR. A53 驱动扩展板对其进行功能的扩展，树莓派将 Python 作为主要编程语言，机器人小车所使用的系统是基于 Linux32 位操作系统开发而出的树莓派 Raspberry Pi 系统。

3.3 摄像头模块

摄像头模块采用的是 Robot Eyes 二代摄像头如图 3-6 所示，该摄像头通过 USB2.0 与主控模块连接，进行图像数据的采集与传输，摄像头内部支持与 Windows、Linux、Openwrt 系统数据传输，并且能与树莓派开发板很好地兼容。拥有 100 万高清像素，720P 分辨率，能够获取清晰的图像便于识别算法准确地计算分析物体颜色与坐标，图像输出格式为 MJPG，并能支持焦距的调节，方便后期的调试。摄像头本身灵活性高，能够水平 360°，前后 90° 的高自由度旋转，兼具体积小重量轻的优良特性。小车将摄像头搭载在 FPV 航模云台之上，由两个高精度的 160° 舵机控制云台转动，便于控制系统发送控制信号准确地调节摄像头水平或前后的角度，提高自动化程度，系统通过调用 OpenCV 内部函数的方式来打开摄像头并进行图像获取与传输。



图 3-6 Robot Eyes 二代摄像头

3.4 舵机模块

机器人小车的摄像头上所使用的舵机型号为 SG90 如图 3-7 所示，其内部由信号调制电路、电机驱动芯片、电机等构成，在工作上拥有防堵塞、防扫齿的特点，且工作扭矩可以达到 1.6kg/cm ，反应转速仅需 $0.12\text{--}0.13\text{ 秒}/60^\circ$ 能够很好地驱动云台在水平与垂直方向的转动，调节视觉角度效果灵敏、准确。

舵机通过接入 PWR.A53 电机驱动扩展板设置的电机扩展接口来接收驱动信号，当主控模块发送指令到电机驱动芯片，芯片能够发送一定占空比的 PWM 信号经舵机接收通道到信号解调电路，并经信号解调后获取直流偏置电压，将直流偏置电压与电路内部产生的基准信号比较产生电压差，并由电机驱动芯片判断电压差的正负驱动电机正反转，并通过级联减速齿轮带动电位器转动，直到电压差为零则电机停止转动。

在舵机的控制中，一般需要电机驱动芯片发送一段脉冲信号到舵机的接收通道，信号中高电平部分决定了控制舵机转动角度的大小。机器人小车采用的舵机是 180 度伺服，时基脉冲与舵机的控制关系如表 3-1 所示。

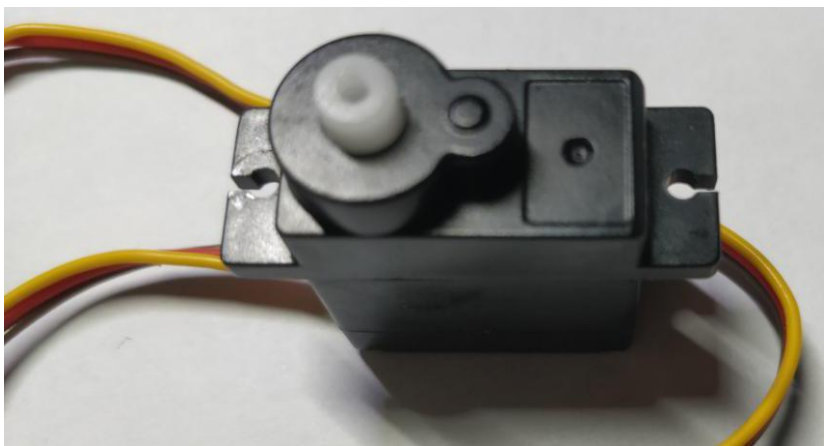


图 3-7 SG90 舵机

表 3-1 时基脉冲与舵机的控制关系表

时基脉冲高电平大小	0.5ms	1.0ms	1.5ms	2.0ms	2.5ms
舵机角度	0 度	45 度	90 度	135 度	180 度

通过四个舵机与多功能舵机支架、大机械爪组装起来，实现了四自由度的机械手臂，其具有舵机防卡死的特点，能够在软件控制下完成物料抓取与放置动作，由于 SG90 舵机工作的扭矩仅有 1.6kg/cm，不能满足驱动机械臂弯曲角度的条件，因此这里采用了工作扭矩为 7kg/cm 的 995 舵机，并能够轻松抓取测试中使用到的纯色积木道具，达到了预期目的。机械手臂的主要参数如下：

- （1）工作电压：4.8-6V
- （2）回转角度：180 度
- （3）整体长度：310mm
- （4）整体重量：540g
- （5）爪子前部极限张开度：125mm

3.5 电机与电机驱动模块

机器人小车所使用的电机为直流减速电机，该电机是由普通直流电机和齿轮

减速箱相结合且具有良好性能的一款齿轮减速电机，由于其能在较低转速下提供较大的力矩，而齿轮箱又能够为其提供不同的转速和力矩，由于分拣式机器人需要带物行进，所以需要电机提供足够的动力来驱动小车的移动，因此选用了该电机。在机器人小车的底座上装有四个这样的电机，电机连线图如图 3-8 所示，通过将左侧两个电机并联形成一组电机 M1，那么它们的电机运动就会是一致的，同理，右侧的两个电机也能够合并为一组电机 M2，采用这种电机结合的方式可以增大小车行进的动力，便于在路面不平的环境下也能够移动。

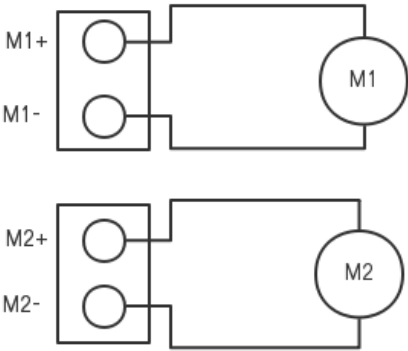


图 3-8 电机连接图

电机驱动模块由 PWR. A53 驱动扩展板提供如图 3-9 所示，PWR. A53 是一款针对树莓派 3B+开发的一款稳压扩展板，可直接对插树莓派 3B+主控板。并能驱动电机、舵机、传感器等。PWR. A53 驱动扩展板上装有电机驱动芯片 L298P，该芯片提供了四个输出口连接四个电机，此外，PWR. A53 驱动扩展板还采用 74HC244 驱动信号芯片作为隔离电路，并作为 L298P 芯片的前端芯片，这样做的好处是当后端电路烧毁时不会影响到树莓派主板，起到了保护电路的作用，并且还能够提升驱动能力，驱动扩展板还提供了驱动信号的外接接口，当 L298P 芯片的驱动能力不足时可以通过外设接口连接其它的电机驱动器提供更强的驱动能力，74HC244 驱动信号芯片留有六根信号输入引脚与主控模块连接，主控模块可以通过发送工作信号到 74HC244 芯片来驱动小车做相应的移动，74HC244 输入端引脚电平与小车运动状态关系如表 3-2 所示。

PWR. A53 驱动扩展板提供了电机扩展接口，可以将舵机连接上这些接口用于驱动舵机工作。



图 3-9 PWR.A53 驱动扩展板

表 3-2 74HC244 芯片输入端引脚电平与小车运动状态关系表

INA	INB	IN1	IN2	IN3	IN4	运动状态
1	1	1	0	1	0	前进
1	1	0	1	0	1	后退
1	1	1	0	0	1	左转
1	1	0	1	1	0	右转
1	1	0	0	0	0	停止

3.6 电源模块

电源模块选用 12V 锂电池如图 3-10 所示，其主要功能是为各硬件模块供电，锂电池具有体积小、充电效率高、循环使用率高的特点，综合性能好，比较适合

于应用在机器人小车上，因此在对比多款电池之后选择了锂电池作为电源模块的供电端，这里所使用的锂电池拥有 2200mAh 的容量，最大输出电流 4A，在正常情况下可以使机器人小车连续运行 3 个小时以上。

由于 PWR. A53 除了有驱动电机的功能，还具有稳压的功能，因此，锂电池没有直接连接在主板上，而是通过 PWR. A53 经过稳压后为主板及各模块提供持续稳定的电能，除此之外，当程序中需要使用多个舵机时，由于电机要同时驱动多个舵机工作需要的电流比较大，这个时候 PWR. A53 能够给舵机单独供电，以提供足够的动力驱动舵机工作。



图 3-10 12V 锂电池实物图

4 软件系统设计

4.1 软件运行环境的搭建

由于树莓派 3B+在出厂时是没有带系统和中文字库的，因此在程序设计之前需要先进行系统的安装与中文运行环境的配置，机器人所采用的系统是树莓派官网的 Raspberry Pi 系统，为了便于系统管理与程序设计，采用远程桌面的方式将系统桌面投影到 PC 端来实现树莓派操作系统的使用，软件环境的搭建流程图如图 4-1 所示。

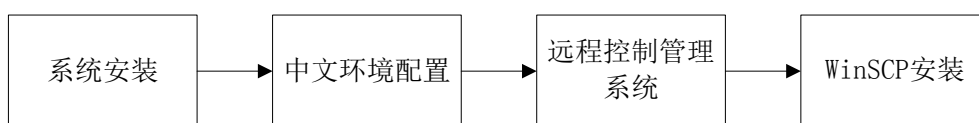


图 4-1 软件环境的搭建流程图

4.1.1 系统安装与中文字库配置

机器人的核心硬件为树莓派 3B+，而软件的设计是建立在 Linux 操作系统下的，所以在程序的设计之前需要安装一个适合于调试与运行程序的操作系统，树莓派 3B+是以 MicroSD 卡作为硬盘，因此安装系统需要将 SD 卡装入读卡器中由 USB 口与 PC 端对接，通过 Raspberry Pi 的官网下载最新的树莓派系统经 Win32 Disk Imager 软件将系统的镜像文件烧录到 SD 卡中。

在 Win32 Disk Imager 映像软件中选择下载好的 Raspberry Pi 系统映像文件的路径以及要安装到的外接 SD 卡对应的盘符，点击软件界面上的 Write 按键等待进度条满了后提示写入成功即可，如图 4-2 为系统写入成功的窗口图，然后将 SD 卡插入树莓派 3B+主板后打开完成系统安装引导即可。



图 4-2 系统写入成功窗口图

由于系统默认字库为英文，因此需要手动安装中文字库。

nano 终端输入以下命令：

```
sudo apt-get install ttf-wqy-zenhei
```

nano 终端提示安装完毕后，输入以下指令重启系统即可完成中文字库配置。

```
sudo reboot
```

4.1.2 远程桌面管理

由于树莓派 3B+没有自带显示屏，为了更好地管理树莓派 3B+开发板中的 Raspberry Pi 操作系统，使程序能够在该操作系统中调试与运行，需要通过 PC 端远程桌面连接上 Raspberry Pi 系统，在远程桌面上实现程序设计与运行，首先要通过 PC 端与树莓派无线网卡连接如图 4-3 所示，然后在 PC 端菜单栏的运行中输入 mstsc 调出远程桌面的界面如图 4-4 所示，输入无线连接 IP 地址 192.168.1.1，则会进入系统登录界面如图 4-5 所示，输入用户名和密码即可进入远程桌面如图 4-6 所示，树莓派官网下载的系统中默认登录的用户名和密码分别为 pi 和 raspberry，这样就完成了远程桌面的访问。



图 4-3 PC 端与树莓派无线网卡连接图



图 4-4 远程桌面的界面图

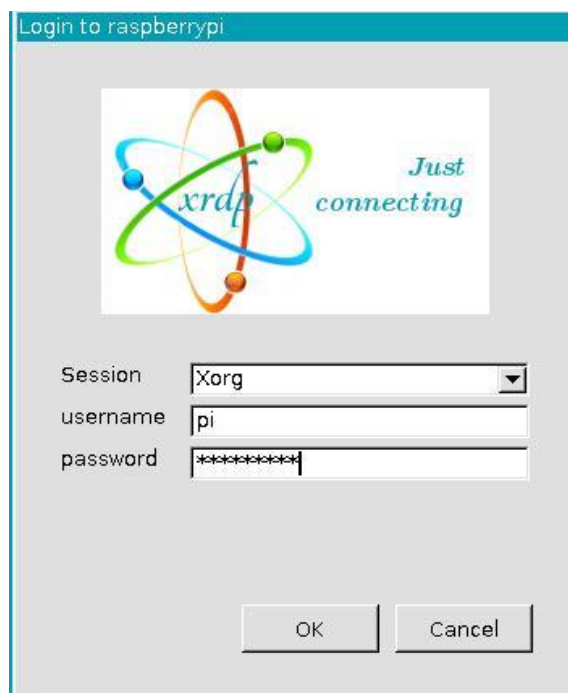


图 4-5 系统登录界面图



图 4-6 系统远程桌面图

4.1.3 WinSCP 安装

WinSCP 是一款支持 SSH 安全外壳协议的跨系统文件传输软件，它能够通过 WiFi 连接实现 Windows 系统与 Linux 系统文件的互传。因此，通过在 Windows 系统环境下安装 WinSCP 连接上树莓派系统便于将 PC 端编写好的程序传入树莓派系统进行调试和运行，也能将树莓派系统下的重要文件返回 PC 端进行备份，防

止系统奔溃导致的数据丢失。WinSCP 可以通过浏览器在线下载安装，连接上树莓派无线网卡后，进入登录界面如图 4-7 所示登录即可进行数据互传。

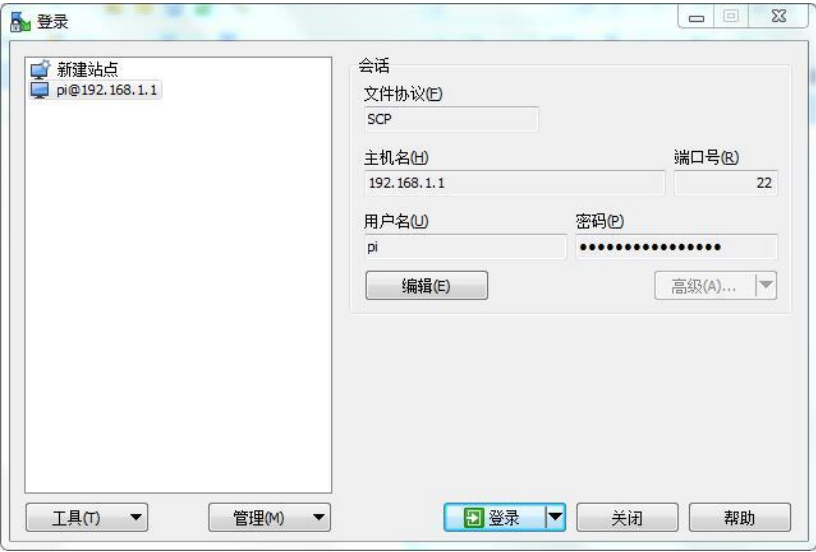


图 4-7 WinSCP 登录界面

4.2 系统程序设计

系统采用 Python 语言编写，在树莓派定制版本 Raspberry Pi 系统下运行程序，为了方便后期的错误检查和优化以及功能的扩展，将程序分为多个模块来写，完整的颜色识别分拣机器人的程序设计分为主程序的设计、颜色识别程序的设计、目标定位追踪程序的设计、触发程序的设计、机械臂控制程序设计。

4.2.1 主程序设计

在搭建好环境的 Raspberry Pi 系统下运行 nano 编辑器，启动初始化程序，其中包括 GPIO 口、LED 等初始化以及打开摄像头循环获取每一帧图像并由窗口显示实时监控视频，然后触发程序在检测周期内不断循环检测，用户需要通过按键选择本轮抓取为单次抓取或多次抓取以及放置在何种颜色的盒子内，并由电源板上的 LED 灯的亮暗表示选择结果。再按下需要识别颜色对应的按键，则摄像头将获取的图片传递给主控模块并运行颜色识别程序，若识别到某一帧中有符合条件的物体目标，则在图片上对其画外接圆，同时将其质心坐标传递给目标追踪程序，对目标进行定位追踪，到达适合抓取的点时停止履带运动，运行机械臂抓取程序，将目标物体抓起，然后控制左右履带右转同时颜色识别每一帧图像寻找预设颜色的盒子，当找到目标盒子时再绘制其外接圆图像，并由目标追踪程序使小车移动到适合放置的位置，主控模块控制机械臂完成放置动作，若初始设置为

多次抓取，则小车继续旋转寻找同种颜色的积木，若为单次抓取，则小车回到初始的方向并停止，主程序流程图如图 4-8 所示。

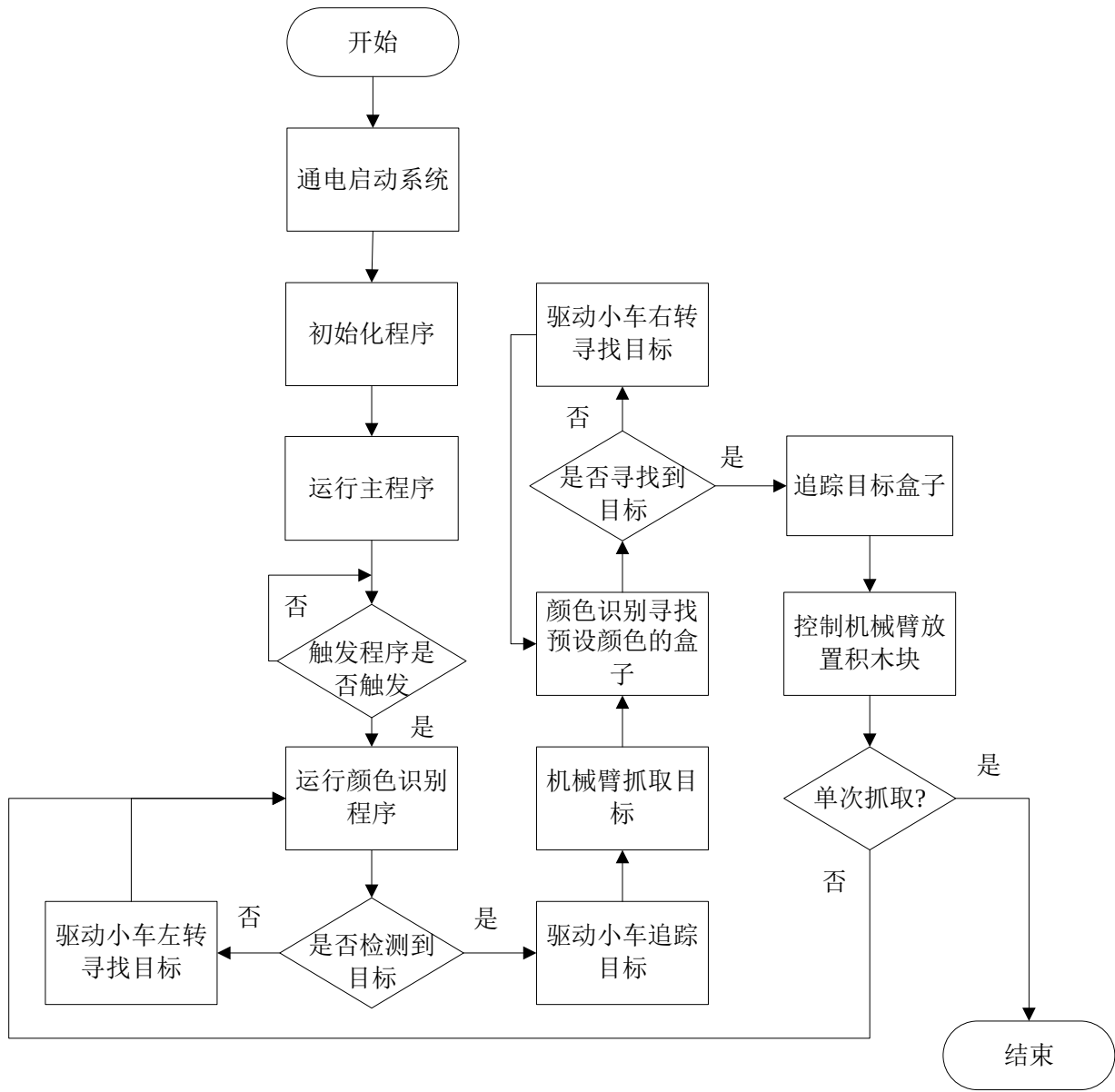


图 4-8 主程序流程图

在自然环境下进行基于机器视觉的颜色识别需要解决光照和天气对待测目标造成色偏的影响，其次还要避免与待测目标相同颜色的背景干扰。软件系统采用限定识别目标轮廓面积以及针对不同环境多次采集数据等方式来解决以上问题，在现有的颜色识别算法对图像预处理过程中大部分采用的是建立 RGB 颜色空间来进行颜色的区分，而机器人小车是建立 HSV 颜色空间对颜色进行划分，因为 HSV 描述颜色的方式更加地直观，它是通过色调、饱和度、亮度来描述一种颜色，

实验证明了 HSV 所描述的颜色更能够被人们所接受^[15]。通过 OpenCV 中的算法对获取的图像进行目标检测处理并且能够实时刷新帧来跟踪目标。

4.2.2 颜色识别与程序设计

颜色识别作为软件系统的核心主要采用的是 OpenCV 的阈值化处理、腐蚀操作、膨胀操作等算法来实现的，通过阈值化处理得到的二值图像来提取出所有轮廓信息，并选择最大的轮廓为目标，获取它的质心坐标，然后用画圆的算法实现在视频上实时选中目标，并提取出其坐标信息供小车追踪物体，颜色识别流程图如图 4-9 所示。

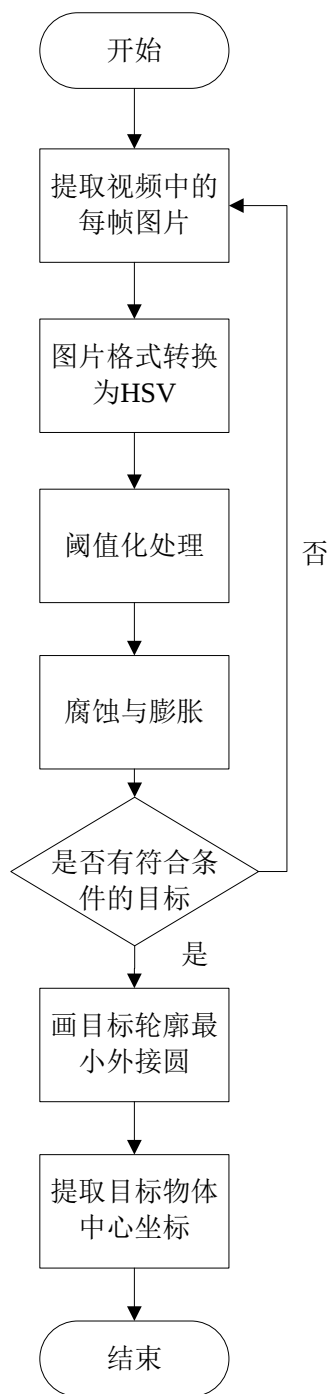


图 4-9 颜色识别流程图

颜色识别具体实现过程：

(1) 定义视频对象：

```
import cv2 import numpy as np
```

```
capture=cv2.VideoCapture(0)
```

通过导入 OpenCV 的专用扩展库 cv2 用于后面各函数算法的定义，视频对象

用于捕获摄像头视频流。

(2) 将获取的视频帧 BGR 格式转换成 HSV 格式:

```
ret, ima=capture.read()
hsv=cv2.cvtColor(ima, cv2.COLOR_BGR2HSV)
```

cvtColor 为颜色空间转换函数，因为软件系统所使用的颜色空间为 HSV 格式，而摄像头捕获的视频帧格式默认为 BGR 格式，因此，程序上设置了将视频帧的颜色空间由 BGR 转换为 HSV 的参数。

(3) 根据阈值构建掩膜:

```
mask=cv2.inRange(hsv, Lower, Upper)
```

inRange() 函数是用于彩色图像的分割，能够将一定阈值范围内颜色的物体分割出来，分割后的图像为二值化图像，函数中第一个参数为要构建掩膜的图像，第二个参数与第三个参数分别为阈值的下限和上限，利用 inRange() 函数和 HSV 模型中各颜色范围的上下界获取掩膜，以蓝色积木为例，如图 4-10 为颜色分割前后对比图。

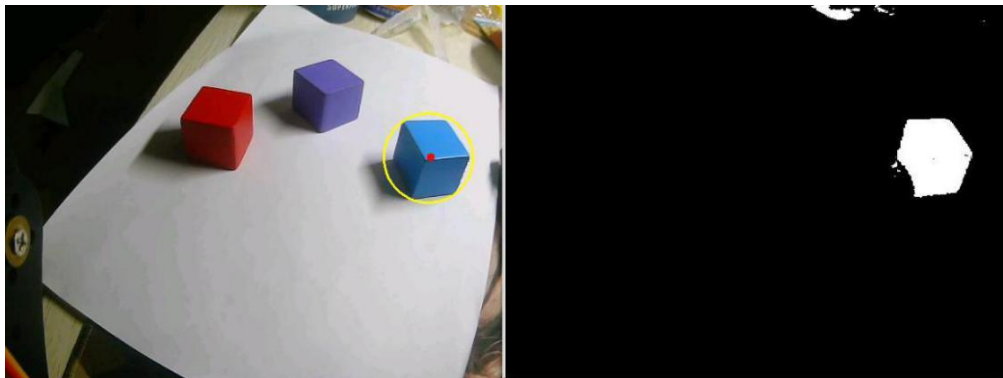


图 4-10 颜色分割前后对比图

但是利用这样的算法只能够将一个阈值区间内的物体分割出来，而在自然环境下红色的物体在 HSV 颜色空间中色调部分可能为 $[0, 4]$ 或 $[170, 180]$ ，这时若仅使用单个 inRange() 函数可能会导致颜色分割的不充分，影响后面的识别，颜色未充分分割的原图与阈值化图像如图 4-11 所示。

因此，如果识别红色的物体，则需要使用以下程序:

```
mask=cv2.inRange(hsv, np.array([170, 110, 60]), np.array([180, 255, 255]))
```

```
mask1=cv2.inRange(hsv, np.array([0, 110, 60]), np.array([4, 255, 255]))
```

```
mask=cv2.add(mask, mask1)
```

利用 OpenCV 的图像求和算法 add()，将两个不同阈值区间下分割出来的图

像进行求和，以达到构建完整掩膜的目的，如图 4-12 为颜色充分分割后的原图与阈值化图像。

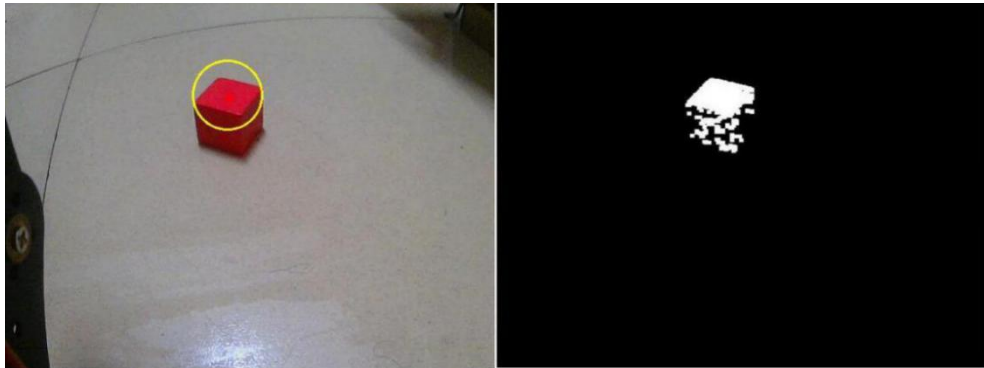


图 4-11 颜色未充分分割的原图与阈值化图像

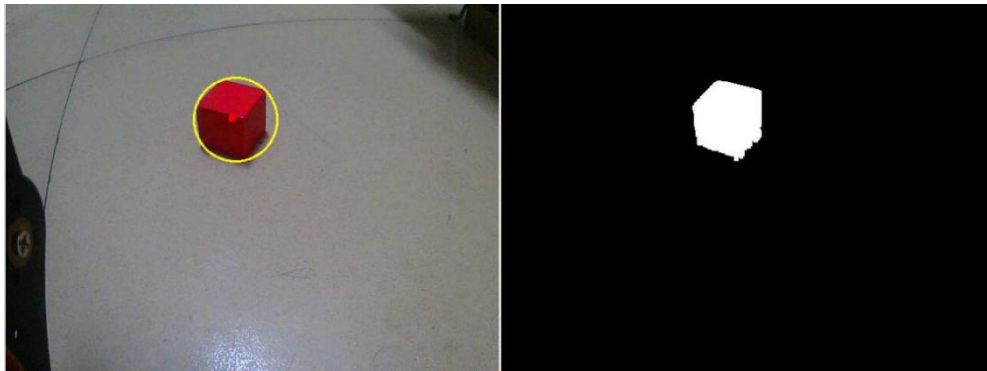


图 4-12 颜色充分分割后的原图与阈值化图像

（4）腐蚀操作与膨胀操作：

```
mask=cv2.erode(mask, None, iterations=2)
```

```
mask=cv2.dilate(mask, None, iterations=2)
```

腐蚀和膨胀都是属于形态学中的基本运算，一般都用于二值图像中白色部分的处理，其中腐蚀操作是求局部像素最小值的操作，它是通过一定的数学卷积计算，使图像中高亮区域缩小，以达到消除外部白色噪点的目的，并且可以得到图像的边界，而膨胀则是与腐蚀相反，它是一种求局部像素最大值的操作，通过计算图像与核的卷积，并将得到核所覆盖区域的最大值，然后将最大值赋给边缘参考位置的像素点，且与目标物体接触的所有背景点合并到物体中，可填补空洞、形成连通域，这样可以使图像中的高亮区域逐渐增大，达到消除图像白色部分内部噪声的目的。或者说，腐蚀具有“缩小”图像的效果，膨胀具有“扩大”图像的效果。

系统对二值图像的处理采用了先腐蚀操作消除图像中无识别意义的物体再膨胀操作填充空洞的方式有效地去除图像中一些无用的小物体及噪声,这种方式在形态学操作中称为开运算,开运算前后对比图如图 4-13 所示,由图可以看出开运算后的图像消除了一些多余小物体,并且物体图像边界变得更加圆滑。

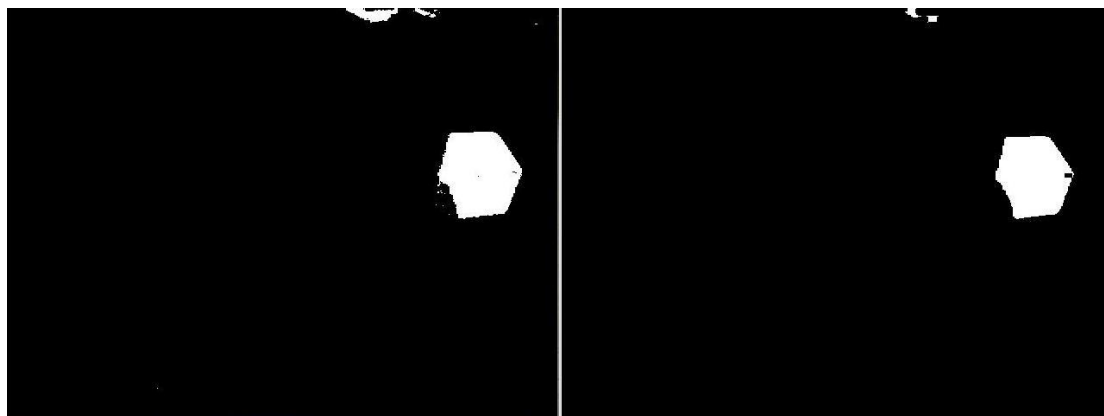


图 4-13 开运算前后对比图

(5) 识别轮廓:

```
cnts=cv2.findContours(mask.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
```

`findContours()` 函数可以用于识别图像中的轮廓,其中第一个参数为待识别图像,第二个参数为轮廓检索模式,为了提高在软件系统中程序的运行效率,选用只检测外轮廓的 `cv2.RETR_EXTERNAL`,第三个参数为轮廓的近似办法,提供选择轮廓存储的完整度,同样为了减少系统资源占用这里采用了压缩水平方向、垂直方向及对角线方向的元素,只保留该方向终点坐标的办法来存储轮廓信息。并将获取的轮廓信息以列表的形式存在变量 `cnts` 中。

(6) 提取合适的轮廓信息:

```
cnts=filter(lambda x:min<cv2.contourArea(x)<max,cnts)
c=max(cnts,key=cv2.contourArea)
```

`filter()` 函数是 Python 提供用于列表处理的序列过滤函数,通过设置一定条件对列表中各元素逐一判断,将不满足条件的元素全部滤除,而满足条件的元素保留下来并重新组成一个列表返回。在系统中主要运用在过滤部分不符合条件的轮廓信息,在 `filter()` 函数的参数中使用匿名函数以 `cnts` 列表中各轮廓的面积作为判别条件,将太小或太大的轮廓过滤掉,能够很大地减少环境颜色对识别的干扰,机器人小车是以积木为识别的目标,通过测试可以得知其在摄像头下拍摄出的最大轮廓面积为 7300,因此,在函数中 `max` 值为 7300, `min` 值太小会导

致过滤不充分，太大会缩短识别的最远距离，所以这里 min 值选择 300，以蓝色积木的识别为例，若未加入 filter() 函数则会导致系统优先识别比它轮廓面积更大的蓝色物体，进而导致分拣目标错误，如图 4-14 为序列过滤函数加入前后对比图。

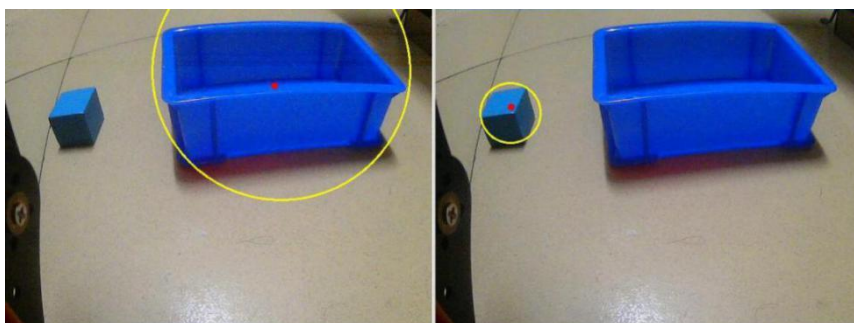


图 4-14 序列过滤函数加入前后对比图

将过滤后的轮廓列表信息再次存入变量 cnts 中，使用 Python 中 max 函数以轮廓面积作为判别条件提取出面积最大的轮廓信息作为识别的结果存入变量 c 中。

(7) 计算轮廓的矩和质心：

```
M=cv2.moments(c)
```

```
center=(int(M["m10"]/M["m00"]), int(M["m01"]/M["m00"]))
```

cv2.moments() 函数能够计算图像的矩，所谓图像的矩就是用来描述一个物体几何特征信息的量，通过函数能够以字典的形式返回这个量。一个完整的矩集不仅能够解释图像中物体的外部形态更能提供其大小、坐标、方向等信息，因此矩的计算在图像分析中十分重要，系统主要运用矩对目标坐标定位。根据这些矩的值，我们可以计算出物体的质心坐标如公式（4-1）所示^[16]。

$$C_x = \frac{M_{10}}{M_{00}}, C_y = \frac{M_{01}}{M_{00}} \quad \text{公式 (4-1)}$$

(8) 画出轮廓的外接圆：

```
((x, y), radius)=cv2.minEnclosingCircle(c)
```

```
cv2.circle(ima, (int(x), int(y)), int(radius), (0, 255, 255), 2)
```

```
cv2.circle(ima, center, 5, (0, 0, 255), -1)
```

minEnclosingCircle() 为 OpenCV 中求最小包围圆的算法，通过该算法可以将轮廓最小外接圆的坐标与半径求出，在通过 circle() 画圆函数在摄像头获取的彩色图像中圈出识别的目标及标出质心，circle() 画圆函数第一个参数为待画

圆的原图，第二个参数和第三个参数分别为圆心坐标与半径，第四个参数为线的颜色，格式为 BGR，最后一个参数为线的宽度，若为-1 则表示填充。

4.2.3 目标追踪算法

目标追踪算法主要是采用设定一个适合抓取的像素点坐标区间，当识别到物体的坐标不在这个区间时，则通过判断 x 坐标与 y 坐标和这个区间的相对位置，来调整小车的运动方向并且在终端显示实时的运动状态，在经过多次测试对比得到了最适合抓取的像素坐标区间 x 轴为 $[115, 240]$ ， y 轴为 $[340, 405]$ ，由此画出目标追踪流程图如图 4-15。

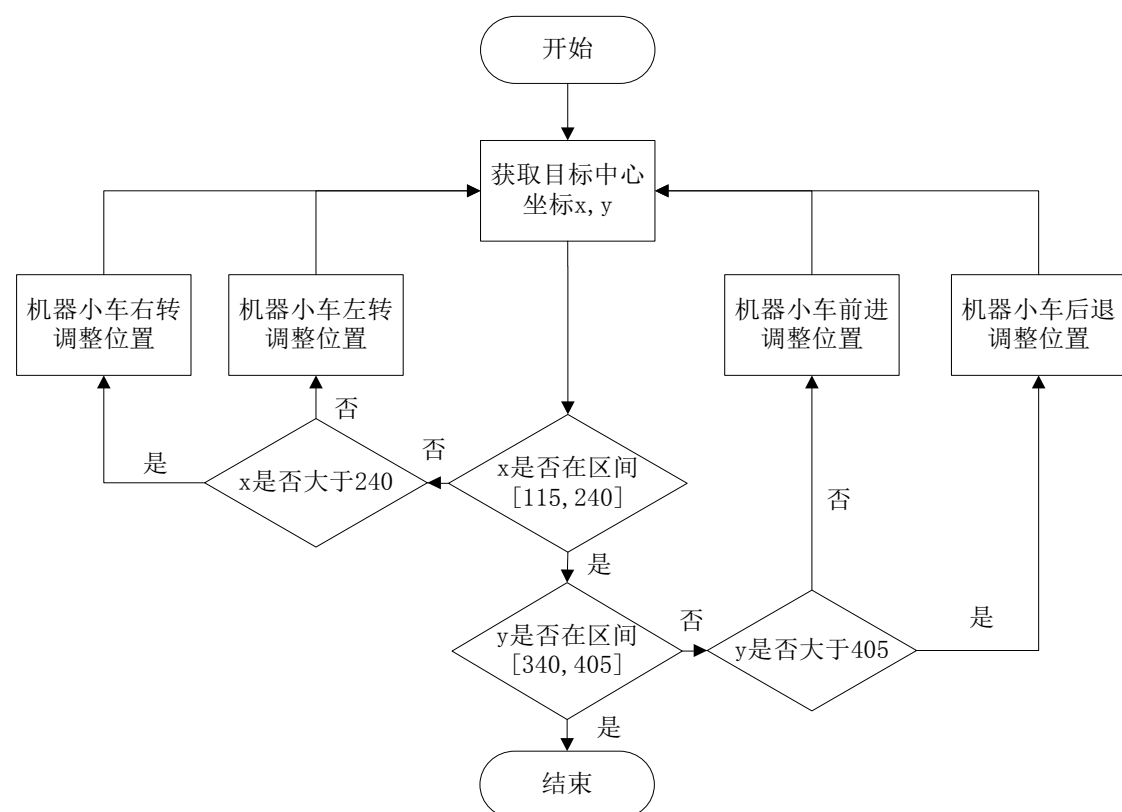


图 4-15 目标追踪流程图

目标追踪程序如下：

```

if(x< 115)&(x>0):
    print("turn left")
    _Motor_TurnLeft_()          #控制小车左转调整位置
elif x > 240:

```

```

        print("turn right")
        _Motor_TurnRight_()          #控制小车右转调整位置
elif(y < 340)&(y > 0):
    _Motor_Forward_()              #控制小车前进调整位置
    print("forward")
elif y >405:
    _Motor_Backward_()             #控制小车后退调整位置
    print('backward')
else :
    print("stop")
    _Motor_Stop_()                 #控制小车停止

```

程序循环检测目标的像素坐标与适合抓取的坐标区间相对位置，目标位置偏左则小车向左调整位置，若偏右则向右调整，若在指定区间内，则向前移动靠近物体，由于小车底座的两个电机不能保证绝对的同步，所以在前进的过程中可能会出现方向偏离的情况，这时还是能够通过循环运行程序实现左右位置的微调。

4.2.4 触发程序

触发模块是通过按键检测程序来实现的，设置一些按键分别对应不同的功能，例如：颜色的选择、抓取类型等，并使程序不断循环检测按键，若某个按键的条件达成，则运行对应的程序。

```
k=cv2.waitKey(1)&0xFF
```

按键检测程序中的数字参数为检测周期中的无效时间，单位为毫秒，在这个时间内进行按键是无法检测到的，由于 cv2.waitKey() 返回的值可能不止是 8 位，而真正有效的数据只有后 8 位，因此需要用逻辑与算法屏蔽无效数据，按键检测程序在每一帧图像的获取中检测一次，若在一帧图像中未检测到按键，则继续捕捉下一帧图像。

选择积木颜色时按键对应的功能如下表 4-1 所示。

表 4-1 积木颜色选择按键表

按键	q	w	e	r	a	s
----	---	---	---	---	---	---

功能	识别绿色	识别蓝色	识别橙色	识别紫色	识别黄色	识别红色
----	------	------	------	------	------	------

辅助功能按键对应的功能如下表 4-2 所示。

表 4-2 辅助功能按键表

按键	z	x	c	v	t	Esc
功能	放置蓝色 盒子	放置黄色 盒子	设置抓取 次数	设置盒子 颜色	暂停	退出程序

4.2.5 机械臂控制程序

机器人小车所使用的编程语言为 Python，而由于 Python 语言是一种脚本语言，因此若在主程序中编写机械臂抓取程序会导致机械臂抓取过程中视频窗口停滞在最后一帧，直到完成抓取或放置动作后才刷新帧，所以关于机械臂动作的程序是通过 Python 语言中的多线程来实现的，这样既能保证机械臂动作，又不影响视频监控。

在机械臂控制程序中通过判断目标的中心坐标是否在可以抓取的坐标范围内，若条件达成，则指令控制机械臂上的四个舵机分别转到指定角度，由舵机协调配合驱动机械臂弯曲关节做出抓取或放置的动作。

4.3 系统调试

颜色识别分拣机器人主要通过颜色识别来锁定目标，并以 30×30×30mm 的纯色积木块作为识别对象。在不同自然环境下对于同一个物体的色调、饱和度和亮度都会发生改变，且环境中带颜色的物体都会影响识别，尤其是在复杂的环境下会使识别精准度大幅度降低。因此，在识别部分的调试上需要将积木块在不同的环境下多次采集 HSV 颜色参数，然后将采集到的参数分别提取其中的最大值和最小值作为程序中构建掩膜的色调、饱和度、亮度上下限，这样可以保证在大部分自然环境中都能识别到积木块。此外，还需要针对积木块和盒子在摄像头照射下的最大面积进行记录，用于对识别轮廓的过滤，避免环境中与积木块相同颜色

物体的干扰。

在目标追踪过程中需要在图像的像素坐标中选取一个最佳的抓取区间，区间选取必须为一个合适的值，才能保证机械臂能够精准地抓取积木块，且区间大小要适中，若区间选择过大则会影响抓取的精准度，若过小则会出现小车不停地移动校正位置，这是由于刷新每一帧图像的耗时长于小车的移动速度，因此需要不断地调试参数才能提高机器人小车的识别与追踪的精度。

在硬件方面由于机器人小车选用的电机为直流减速电机，这款电机虽然能够提供较大的动力，但是在控制上不够精细，另一方面，由于机器人小车两侧的硬件布局不同，从而两侧履带的负重量也不同，因此在调试过程中发现小车两边的电机在同样速度的驱动信号下行进右侧履带的行进速度会偏快，导致了小车在直线行驶时跑偏的情况，且在小车原地转弯的过程中会出现两边履带速度相差过大而导致其中一边电机卡死的情况，针对这个问题主要采用了在软件上通过控制主控模块发送的控制信号到电机驱动模块调整两边电机的转速，使两边电机的速度尽可能地达到平衡。

结 论

基于机器视觉的颜色识别分拣机器人的设计结合积木块表面在自然环境下 HSV 颜色空间模型的特点，在软件上实现了颜色识别的功能，再由追踪算法与机械臂抓取配合实现了机器人自动追踪分拣系统。将机器视觉的颜色识别技术引入工业分拣机器人，改善了传统工业机器人使用传感器识别距离短、精度低的缺陷。针对各个技术功能的要求，主要完成了以下工作：

（1）根据颜色识别中所涉及到的相关内容进行介绍，包括色彩的概念和 HSV 颜色空间模型的组成结构，以及 OpenCV 的特点及应用，并测试得出了不同颜色积木块在各种环境下的最佳 HSV 颜色分量。

（2）根据基于机器视觉的颜色识别分拣机器人对硬件系统方面的需求，设计了硬件的整体组成结构连接图，并分别对主控模块、摄像头模块、舵机模块、电机驱动模块、电源模块等硬件进行了选型与组装，实现了机器人小车分拣的硬件基础。

（3）对机器人小车进行软件系统的设计，采用 OpenCV 通用算法并使用 Python 语言编写，设计了颜色识别算法、目标追踪算法、触发程序与机械臂分拣程序等，实现了机器人小车对不同颜色积木块的辨识及抓取，并根据不同颜色的盒子所放置的位置，定位并追踪到目标颜色的盒子控制机械臂放置积木块，达到了机器人小车分拣的目的。通过系统的整体调试，反复测试自然环境下机器人小车对积木块的分拣情况来对识别与追踪算法中的核心部分不断测试调整参数，实现了具有更高分拣效率和稳定性的颜色识别分拣机器人。

机器人小车的颜色识别算法加入了对目标轮廓面积的限制，能够很大程度地减少环境颜色的干扰，提高识别的精度，可以在一般的环境下进行颜色识别，但是如果环境过于复杂，则有可能出现识别出错，因此如有必要，可以在图像处理中颜色识别的算法上进行更进一步的研究。

参考文献

- [1] 温晨昊. 未来已来——人工智能[J]. 科技风, 2019, 3(11):70.
- [2] 机器视觉将是人工智能下一个前沿领域[N]. 中国信息化周报, 2016-08-08(026).
- [3] 周建浩. 用于物流中心的快件自动分拣机器人的研究[D]. 唐山:华北理工大学自动化系, 2018.
- [4] 王喜文. 机器人技术在物流中的应用:分拣、搬运到送货[J]. 物联网技术, 2017, 7(03):6-7.
- [5] Md.Hazrat Ali, Aizat K, Yerkhan K, Anuar O. Vision-based Robot Manipulator for Industrial Applications[J]. Procedia Computer Science, 2018, 5(03):133.
- [6] 蔡倩倩. 基于机器视觉的机器人分拣技术发展研究[J]. 现代制造技术与装备, 2018, 9(08):9-11.
- [7] 黄金荣, 张冰雪, 张悬光. 基于机器视觉的颜色识别系统设计[J]. 山东工业技术, 2019, 3(12):79.
- [8] 高峰, 王富东. 浅谈机器视觉技术发展及应用[J]. 山东工业技术, 2019, 4(05):142.
- [9] 王飞. 机器视觉技术发展及其工业应用[J]. 电子技术与软件工程, 2018, 5(16):246.
- [10] Dong X U. The Introduction of Chromatology[J]. Journal of Liaoning University, 2016, 35(6):34.
- [11] 阮秋琦. 数字图像处理学[M]. 北京:电子工业出版社, 2014:40-42.
- [12] 杨琦, 陶晶, 周润发. 一种基于颜色识别的智能分拣机器人的设计[J]. 赤峰学院学报(自然科学版), 2017, 33(18):22-24.
- [13] 王天平, 王贵荣, 吴伟. 一种用于工业机器人视觉分拣的物料颜色识别方法[P]. 中国专利:CN108890692A, 2018-11-27.
- [14] 陈纪昌, 邱元阳, 刘宗凡. 树莓派——登峰造极路上的廉价教育硬件框架[J]. 中国信息技术教育, 2017, 11(01):66-70.
- [15] 俞文静, 莫健彬, 黄嘉锵. 基于树莓派与 OpenCV 的智能监控跟踪机器人系统设计与实现[J]. 现代计算机(专业版), 2018, 9(17):80-84.
- [16] 苏朝阳, 马万太, 黄磊. 基于图像处理库的机器视觉检测系统的研究[J]. 机械制造与自动化, 2019, 48(01):193-196.