

# 实验二：超市自动补货

## 一、场景描述

根据 IHL 的研究报告，零售商每年在商品管理上损耗的资金成本达到了 7 万亿元人民币，这些损耗来自于商品的丢失，商品的摆放位置错误，不精准的商品统计和一些其他损耗。为了减少在补货过程中造成的损失，本实验搭建了智能补货机器人系统，可以更精准、更高效地补货，减少商品损耗。

## 二、实验原理

本实验搭建的智能补货机器人系统，采用了图像分割技术、图像识别技术、相机标定以实现机械臂自动补货。机器人使用视觉传感器获取商品的图片，通过图像分割技术分割图片中商品，利用图像识别技术识别每种商品的种类，获取每个商品在机械臂空间坐标系下的位置，最后机器人抓取商品放置到对应的货架位置。

图像分割就是把图像分成若干个特定的、具有独特性质的区域并提出感兴趣目标的技术和过程。它是图像处理到图像分析的关键步骤。

图像识别技术是人工智能的一个重要领域。它是对各种不同模式的目标和对像进行识别的技术。要实现图像识别需要经过如下的过程：

1. 建立大量的图像数据集；
2. 建立一个图像分类的模型；
3. 图像预处理；
4. 对处理完的图像进行分类，训练分类模型；
5. 测试训练完的分类模型，评估模型。
6. 应用分类模型。

相机标定的目的是让机器人知道图片中的物体在机械臂空间坐标下的位置。

### 三、实验器材

设备图像	名称	数量
	机械臂	1
	手爪套件	1
	摄像头	1
	电源适配器	1
	Type-C 线	1
	商品模型	4



地图

1



货架

1

## 四、实验步骤

### 1. 场景搭建

(1) 机械臂、商品模型、货架的摆放示意图，如图 2.1 所示。



图 2.1 实验地图

(2) 超市智能补货系统实物图，如图 2.2 所示。



图 2.2 超市智能补货系统实物图

## 2. 程序设计

第一步：阅读超市自动补货程序流程图，如图 2.3 所示。

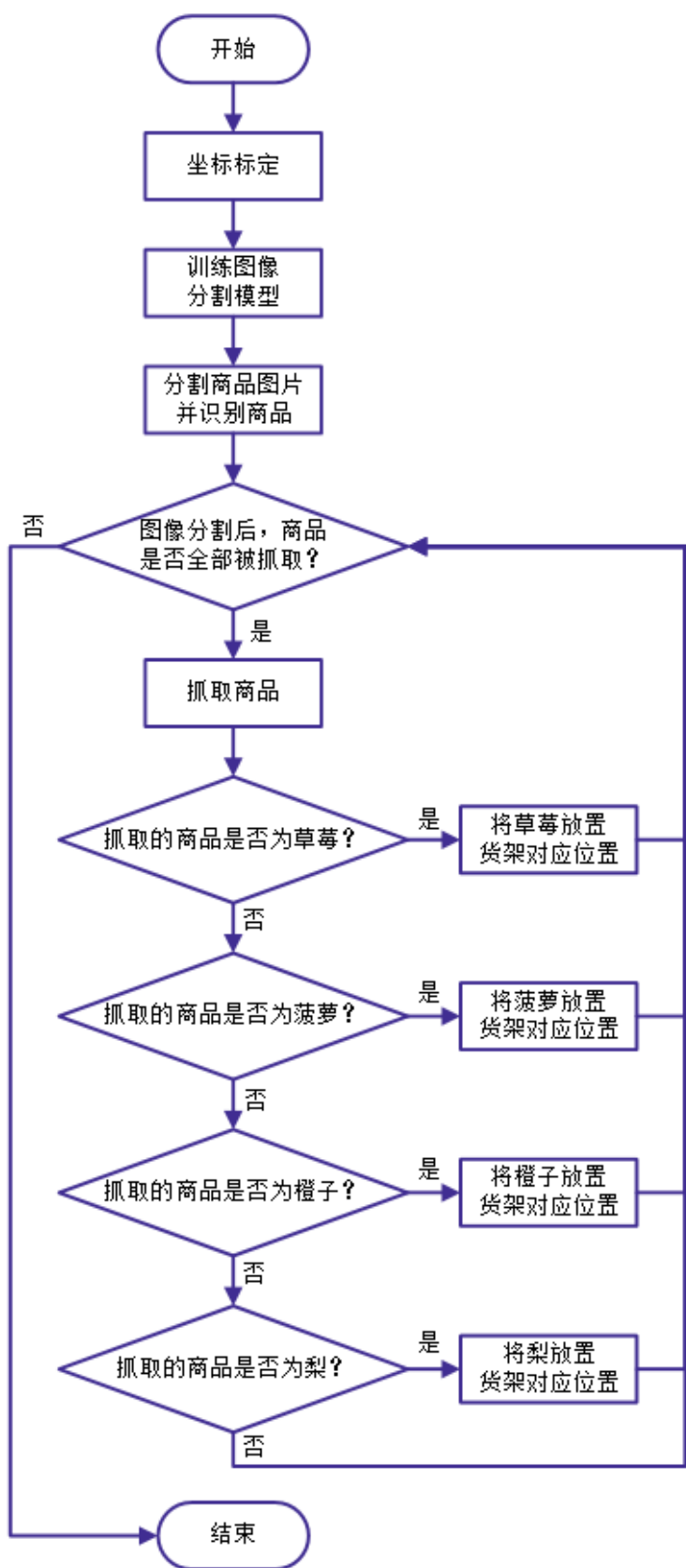


图 2.3 超市自动补货流程图

第二步：连接设备，如图 2.4 所示。

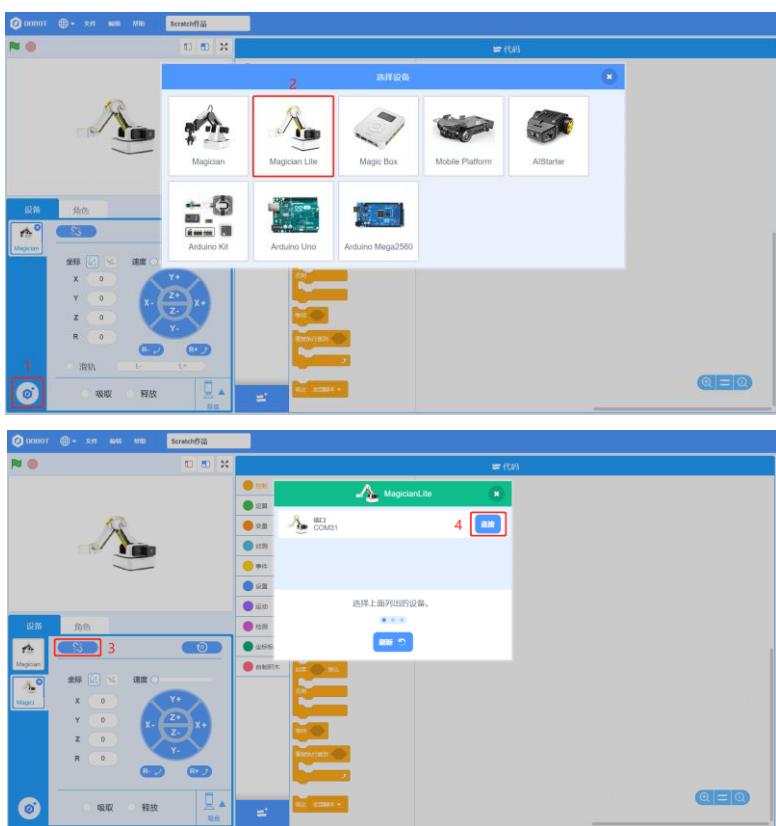


图 2.4 连接设备

第二步：坐标标定。将末端更换为吸盘，打开 DobotBlock，连接机械臂，点击“坐标标定”功能，将标定卡放置摄像头下方（250，0，120，0）为固定标定，不可更改。根据提示移动机械臂至 A、B、C、D 4 个点完成标定，如图 2.5 所示。





第三步：训练图像，分割模型。

1) 添加 AI 拓展模块，选择“AI”选项卡，点击“新建分类数据”，如图 2.6 所示。



图 2.6 编辑分类数据

2) 图像分割。打开摄像头之后，选择图像切割模式，移动机械臂调整摄像头的位置，直至每个商品都被矩形框分割，如图 2.7 所示。



图 2.7 图像分割

3) 创建数据集。点击加号创建新的数据集，并为每个数据集创建标签，如图 2.8 所示。



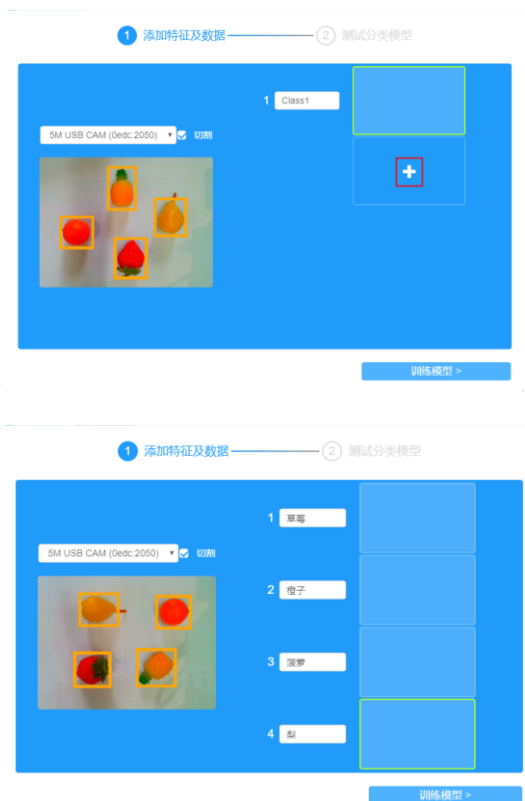


图 2.8 创建数据集标签

4) 采集商品图像数据。先用鼠标点击数据集框，再用鼠标点击对应的商品，即可将商品图像保存在对应的数据集中，如图 2.9 所示。

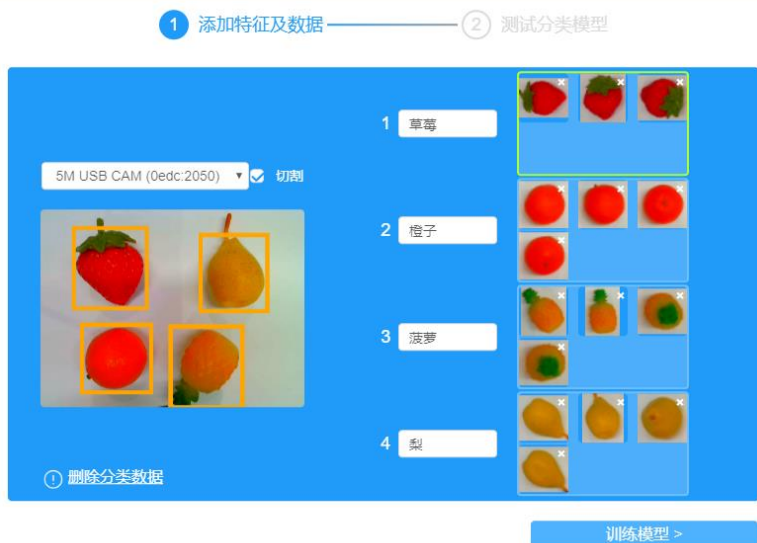


图 2.9 采集商品图像数据

5) 训练模型。采集完商品图像数据之后，点击训练模型，进入训练模型的界面。点击截图切割，每个商品会

被不同颜色的矩形框分割开来，查看每个商品矩形框颜色与其标签对应的单词是否一致，如果一致则点击完成训练，否则返回上一步重新收集数据，如图 2.10 所示。

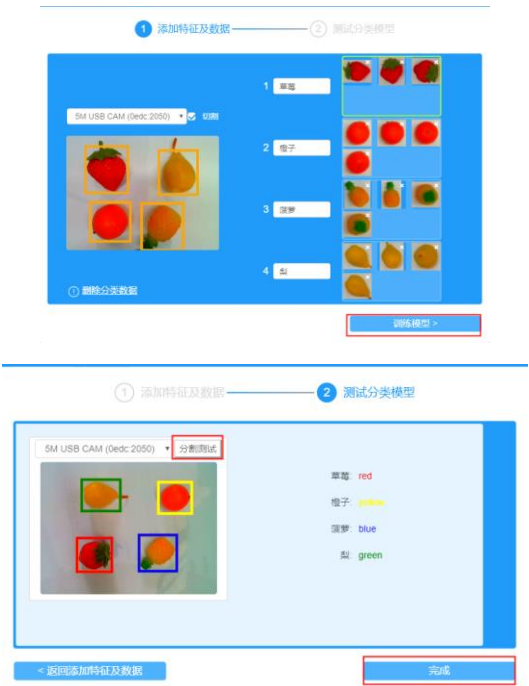


图 2.10 训练模型

第四步：初始化机械臂的位置，该位置为固定位置 (250, 0, 120, 0)，并获取商品的照片，如图 2.11 所示。



图 2.11 初始化机械臂

第五步：定义变量 counter 存放被分割之后商品的总数量。商品被分割之后，商品按 1~counter 顺序编号。定义变量 i 表示当前被抓取商品的编号，初始值设为 1，如图 2.12 所示。



图 2.12 定义变量

第六步：抓取商品。设置抓取高度 Z 为-10，如图 2.13 所示。

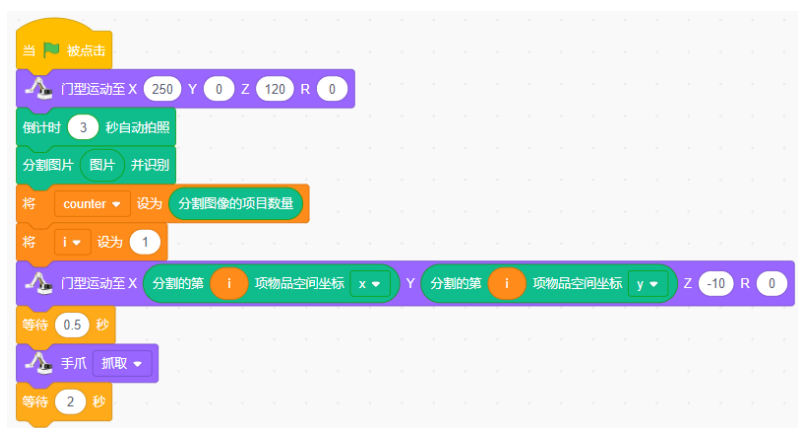


图 2.13 抓取商品

第七步：判断抓取的商品是否是“草莓”，如果是草莓，则将商品放置货架中草莓对应的区域，如图 1.14 所示。

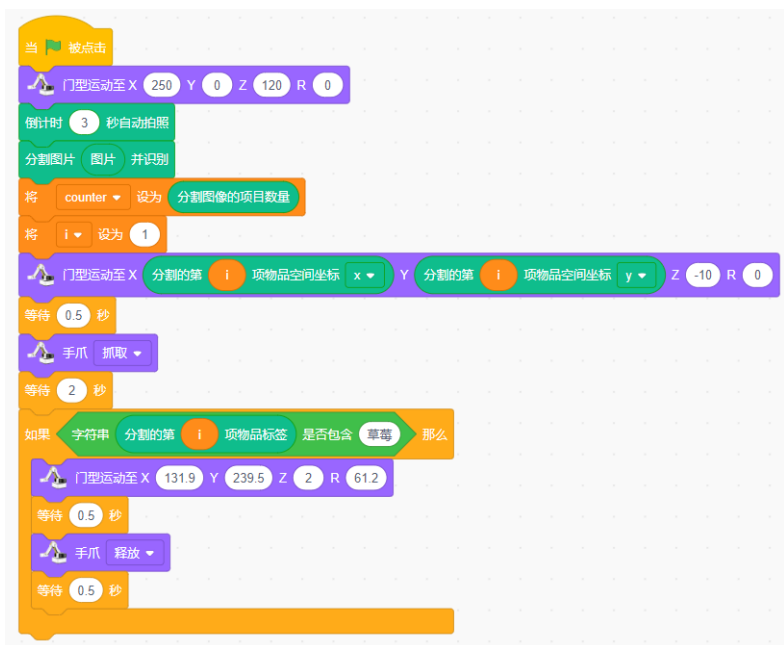


图 2.14 抓取草莓

第八步：如果不是草莓，则判断被抓取的商品是否是其他商品，然后将其放置其对应的货架位置，如图 2.15 所示。

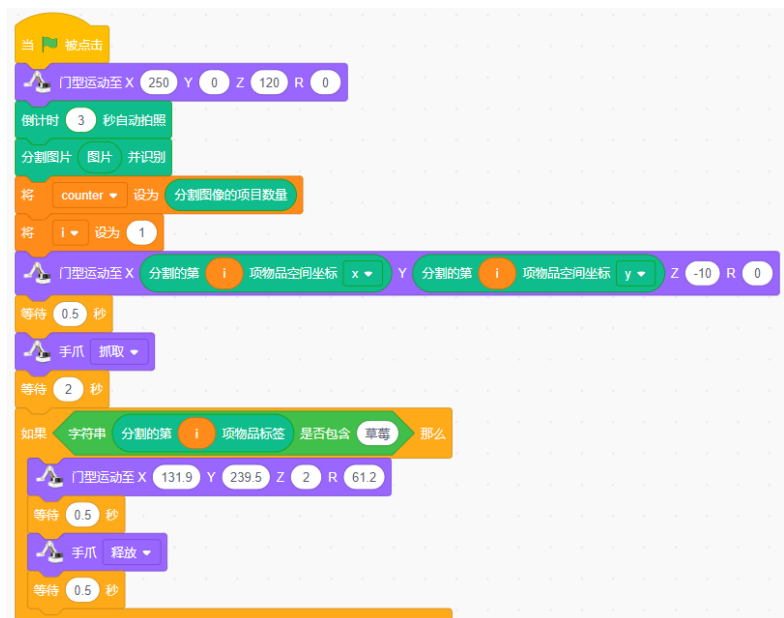


图 2.15 抓取其他商品



第九步：当抓取完一个商品之后，商品编号要加 1，指向下一项商品，并判断商品是否全部被分拣完，如果还有商品，则继续抓取商品至对应的货架区域，直至所有商品被抓取完，然后停止抓取。如图 2.16 所示。

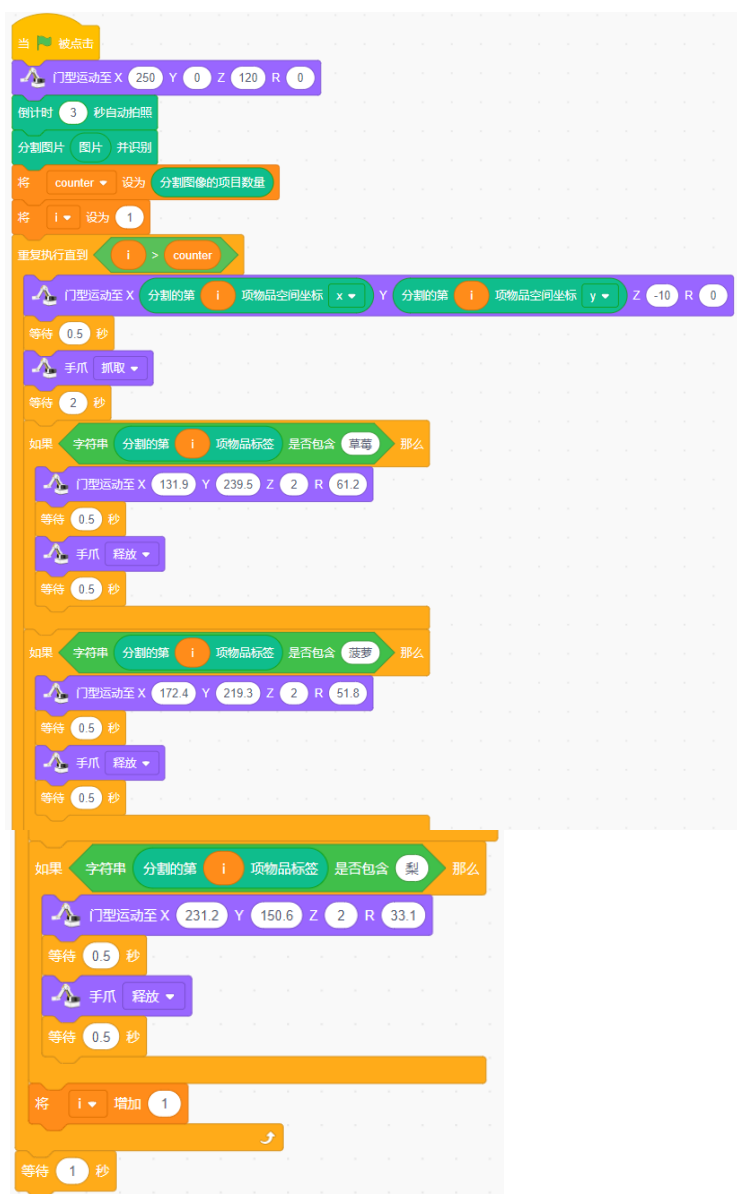


图 2.16 重复执行