

## 大象机器人——分拣码垛积木

- 一、项目背景&目的
- 二、实验环境搭建
  - 1. Ubuntu + ROS
  - 2. Anaconda + Python + 大象机器人开源代码库
- 三、目标检测算法
  - 1. 综述
  - 2. 传统目标检测算法
  - 3. 深度学习算法
  - 4. RGB颜色识别算法
- 四、机械臂控制
  - 1. 位置计算&码垛抓取
  - 2. 码垛收集
  - 3. 机械臂复原
- 五、总结与提升
- 六、小组成员

# 大象机器人——分拣码垛积木

## 一、项目背景&目的

随着计算机、电子、机械等学科不断发展，机器人越来越多地被应用于各个领域，如医疗、智能制造、应急救援等领域。其中，六轴机器人由于具有可编程、拟人化、通用性、机电一体化、高自由度等特性，在各个领域都广受欢迎。本项目旨在使用大象机器人的myCobot协作机械臂，进行积木的识别和抓取功能，并借此了解UR构形的六轴协作机器人相关知识及操作，包括坐标系概念，机器人运动学，运动轨迹规划，计算机视觉等。



图1：大象机器人-人工智能套装

该项目的硬件环境使用的是**大象机器人的人工智能套装**，其中机械臂型号为**myCobot 280-M5**，自重仅850g，有效载荷250g，有效工作半径280mm，可搭配多种末端执行器适配多种应用场景，也可支持多平台软件的二次开发。myCobot 是**大象机器人**和**M5stack**联合出品的一款，**世界最小最轻的六轴协作机器人**，是一个拓展生产力和想象力边界的有力工具。

对体积小巧但功能强大的六轴机器人的探索，可以为**科研教育、智能家居，商业探索**等各种场景积累宝贵的经验，并给予创造性的启发。

本项目期望实现1.通过摄像头对场景中的目标码垛识别；2.控制六轴机械臂移动至相应位置；3.使用空气吸泵对目标码垛进行抓取；4.控制机械臂移动码垛至目标收集点；5.释放码垛，6.回归初始位置并可以进行下一轮目标抓取操作；这六个主要操作。

## 二、实验环境搭建

实验环境搭建方面我们进行了诸多尝试，初期我们尝试使用Ubuntu，但由于该版本的myCobot机械臂不支持Ubuntu 20.04的版本，后续我们又改用Anaconda+Python的环境进行开发。

### 1. Ubuntu + ROS

Ubuntu是一个以桌面应用为主的Linux操作系统。由于Linux在包管理方面相较于windows有很大优势，我们在项目初期计划安装Ubuntu并搭建ROS开发环境来进行项目的开发。我们使用Vmware进行了Ubuntu20.04的安装，并安装相应版本的机器人操作系统 ROS Noetic、MoveIt 以及 git 版本管理器。

但由于myCobot280-M5型号的机械臂不支持Ubuntu20.04的使用，效率起见，我们使用更加友好的Anaconda + Python作为我们的实验环境。

### 2. Anaconda + Python + 大象机器人开源代码库

Anaconda是一个可以便捷获取包且对包能够进行管理，同时对环境可以统一管理的发行版本。

Anaconda包含了conda、Python在内的超过180个科学包及其依赖项。我们的实验环境基于python3.7搭建，主要使用**opencv开源库**和**大象机器人开源代码库**进行开发。

## 三、目标检测算法

### 1. 综述

目标检测任务是找出图像或视频中人们感兴趣的物体，并同时检测出它们的位置和大小。不同于图像分类任务，目标检测不仅要解决分类问题，还要解决定位问题，属于Multi-Task问题。目标检测的发展脉络可以划分为两个周期：传统目标检测算法时期(1998年-2014年)和基于深度学习的目标检测算法时期(2014年-至今)。而基于深度学习的目标检测算法又发展成了两条技术路线：**Anchor based**方法(一阶段，二阶段)和**Anchor free**方法。

### 2. 传统目标检测算法

传统目标检测算法主要基于手工提取特征，思路大多不复杂，但也有一些较为明显的缺点，如识别效果不够好，准确率不高；计算量较大，运算速度慢；可能产生多个正确的识别结果等。一般传统检测算法流程可概括如下：

- 选取感兴趣区域，选取可能包含物体的区域
- 对可能包含物体的区域进行特征提取
- 对提取的特征进行检测分类

其代表算法有**Viola Jones Detector**，**HOG Detector**，**DPM Detector**等。

### 3. 深度学习算法

2012年卷积神经网络(Convolutional Neural Networks, CNNs)的兴起将目标检测领域推向了新的台阶。基于CNNs的目标检测算法主要有两条技术发展路线：anchor-based和anchor-free方法，而anchor-based方法则包括一阶段和二阶段检测算法(二阶段目标检测算法一般比一阶段精度要高，但一阶段检测算法速度会更快)。

Anchor-Based中的代表算法有RCNN, SPPNet, Fast RCNN, Faster RCNN, Cascade RCNN, YOLO v1, SSD, RetinaNet, YOLO V5等

Anchor-Free的代表算法有CornerNet, CenterNet, FSAF, FCOS, SAPD等

## 4. RGB颜色识别算法

在我们的机器人识别码垛任务中，由于任务场地固定，背景单一，没有必要使用过于复杂的深度学习模型。基于这些考虑，我们计划Python+OpenCV进行基于颜色的目标识别。

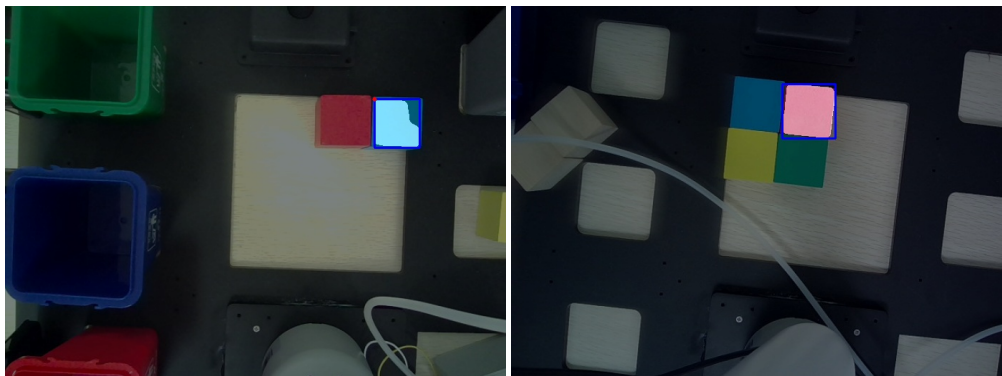


图2：识别蓝色和红色码垛

首先通过机器人自带的摄像头获取目标的RGB图片，利用opencv对目标颜色进行识别和提取，并计算目标颜色的区块大小，过滤掉轮廓围成面积较小的物体，提取图像的轮廓，并计算中心点，最后将中心点至机械臂操作端进行抓取操作。

## 四、机械臂控制

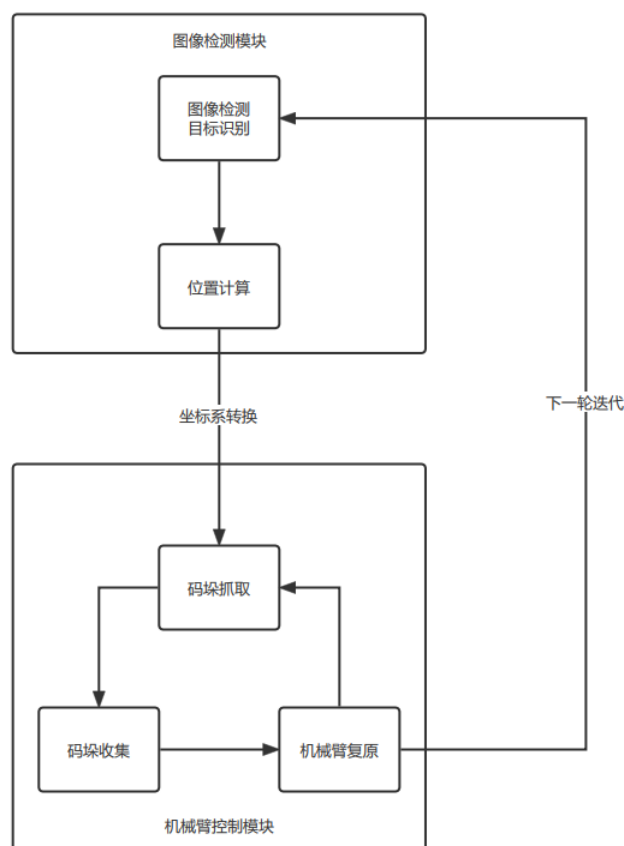


图3：码垛识别、分类、收集总流程

## 1. 位置计算&码垛抓取

通过俯视摄像头以及RGB颜色识别算法，我们向机械臂控制模块返回了目标码垛所在的平面坐标系中的中心点位置。由于我们所用的六轴机器人固定基地啊与检测范围相对固定，我们可以据此计算出使得六轴机器人的吸泵端接触目标码垛所需的六个自由度上的具体数值——旋转(S轴)，下臂(L轴)、上臂(U轴)、手腕旋转(R轴)、手腕摆动(B轴)和手腕回转(T轴)以及相应的移动速度，移动机械臂至指定位置并启动吸泵，完成目标码垛的抓取操作。

## 2. 码垛收集

在RGB颜色识别算法部分，我们在传给机械臂控制模块时，同时传递了码垛的位置和颜色信息。机械臂会根据获得的颜色信息，以及坐标系转换计算，将拾取到的码垛移动至对应颜色的收集小框上方，释放吸泵，使得目标码垛被准确收集。



图4：智能机械臂控制下的码垛抓取与收集

## 3. 机械臂复原

在完成码垛释放后，我们会控制机械臂回到初始位置，同时再次使用俯视摄像头拍摄新状态下的码垛状态，重复步骤1-3，最终完成所有目标颜色码垛的识别，拾取，与分类操作。

## 五、总结与提升

myCobot机器人操作简单且能与人协同安全工作，我们通过不断地尝试与探索完成了myCobot机械臂分拣目标码垛实验，并了解了基本的UR构形的六轴协作机器人相关知识及操作，包括机器人坐标系概念及转换，机器人运动学，运动轨迹规划，计算机视觉等知识。

在未来提升方面，在实验过程中，我们发现由于硬件设备的空气吸泵吸力较大，不能很好地完成积木块的释放操作，后续可以对吸泵控制进行进一步的优化。

## 六、小组成员

吴一凡，金冬阳，朱世博，蓝晨溪