

## 4 基于视觉的目标抓取与分拣

### 4.1 实验目的

本实验旨在使用机械臂平台，实现对环境中的物体的视觉检测和识别，并实现抓取多类型目标物体的功能。通过此实验，可以探索机械臂在实际环境中的检测、定位、抓取的应用。

- 1) 掌握传统目标检测方法和深度学习目标检测的基本原理。
- 2) 实现机械臂视觉 openCV 色块目标的识别。
- 3) 实现一阶段目标检测方法实现多目标提取，掌握 YOLOv5 的算法原理和实现过程，完成在机械臂平台的模型部署。
- 4) 实现机械臂抓取系统在光照变化的室内环境中实现了机械臂抓取操作。

### 4.2 实验准备

OAK 相机设备，移动操作机械臂平台，ROS 机械臂软件平台和人工智能实验平台。

### 4.3 实验内容

- 1、按照实验步骤，用机械臂完成实验步骤。进阶实验为选做实验。
- 2、请根据本章所学内容，完成机械臂视觉的 OpenCV 色块目标识别分拣补充实现。
- 3、请根据本章所学内容，完成数据集的标注，完成 YOLO 多类型目标检测的训练，并应用在机械臂上，完成目标的抓取和分拣应用，完成相对应的实验报告。

### 4.4 实验原理

#### 1. opencv 颜色检测

近年来，数字图像处理技术发展迅速，基于机器视觉的各种检测技术逐步从研究领域走向实用化阶段。颜色检测用图像处理与识别技术测量颜色之间的差别，

可以显著提高对颜色检测的工业生产的自动化程度。颜色检测技术通过图像采集设备采集待测物体表面颜色图像，利用参考颜色标准，使用计算机对颜色图像进行算法计算，准确区别颜色之间的差别。

下面是彩色图像处理过程中的关键算法及原理：

(1) RGB 图像分解在 RGB 彩色图像中，图像由三个图像分量组成，每一个分量图像都是其原色图像，当送入 RGB 监视器时，这三幅图像在荧光屏上混合产生一幅合成的彩色图像。目前广泛采用的彩色信息表达方式都是在三基色按比例混合的配色方程基础上得到的，配色方程见公式 4-1：

$$C = aR + bG + cB \quad (4-1)$$

式中：

C——为任意一种彩色光；

R——Red 红色；

G——Green 绿色；

B——Blue 蓝色；

a——红色的权值；

b——绿色的权值；

c——蓝色的权值；

根据式（4-1），一幅彩色图像中的每一个像素都可以用三维色空间（R、G、B）中的一个矢量[a，b，c]来表示。R、G、B 的比例关系确定了所配彩色光的色度（色调和饱和度），其数值确定了所配彩色光的光通量，aR、bG、cB 分别代表彩色 C 中三基色的光通量成分，又称为彩色分量。

(2) RGB 到 HSI 彩色模型的转换当观察一个彩色物体时，往往用色调、色饱和度和亮度描述它。根据肉眼的色彩视觉三要素及 HIS 中的色调（Hue）、饱和度（Saturation）和亮度（Intensity）提出 HIS 彩色模型。那么，在一幅给定的 RGB 彩色格式的图像，根据每一个 RGB 像素可以得到 H 分量。假定 RGB 值归一化为[0,1]范围内，则得到的其他两个 HIS 分量一定在[0,1] 范围内。

(3) 图像分割是图像处理到图像分析的关键步骤，也是一种基本的计算机视觉技术。基于阈值的分割方法是一种应用十分广泛的图像分割技术，其实质是利用图像的灰度直方图信息获取用于分割的阈值。它是用一个或几个阈值将图

像的灰度级分为几个部分，认为属于同一部分的像素属于同一物体。它特别适用于目标和背景占据不同灰度级范围的图像。

## 2. 深度学习基础

深度学习是一种模拟人脑神经网络结构的机器学习算法。其中，卷积神经网络（CNN）是深度学习中常用的网络结构，它通过多层卷积和池化操作实现对图像特征的提取。

## 3. 目标检测算法

近年来，卷积神经网络（CNN）在机器视觉中的应用，极大地提升了图像中物体的检测能力。对深度学习中的对象检测问题，根据其网络结构及特征学习的特点，将其划分为两阶段目标检测和一阶段目标检测。

RCNN（Regions with CNN features）是一种经典的目标检测算法。它将目标检测任务分为两个步骤：候选区域提取和候选区域分类。首先，通过选择性搜索等算法从图像中提取候选区域。然后，每个候选区域被输入到 CNN 中进行特征提取和分类，得到最终的检测结果。

YOLO（You Only Look Once）是一种实时目标检测算法。相比于传统的两阶段方法，YOLO 将目标检测任务转化为一个回归问题，同时在单个网络中实现目标的位置和类别的预测。这种设计使得 YOLO 可以实现实时的目标检测，适用于移动设备等资源受限的场景。

两阶段检测算法相比一阶段检测算法具有精确的检测结果。然而，它的实时性低于一阶段检测算法。与传统算法相比，基于深度学习算法是图像检测最有效的解决方案，因为这些模型具有深层架构，可以有效地学习特征，而且，基于深度学习算法的特征学习算法在这些模型中更加稳健，不需要任何手动特征设计模型。基于深度学习的目标检测方法可以用于机械臂视觉目标检测。

## 4. 协作机械臂抓取与放置

协作机械臂抓取与放置实验的原理主要基于视觉感知、位姿估计、控制系统和协作算法的综合应用。在机械臂目标检测和定位完成后，一旦物体的抓取点被确定，机械臂需要规划如何移动其机械臂以到达这些点并执行抓取动作。

## 4.5 实验步骤

### 4.5.1 环境配置

#### 4.5.1.2 人工智能实验平台（推荐）

登录人工智能实验平台（ustc.edu.cn）：  
<https://etcis-web.ustc.edu.cn/rgznsxpt/list.htm>。使用在线的人工智能实验平台。

实验平台支持主流的深度学习框架，例如 PaddlePaddle、TensorFlow、PyTorch 等，同时支持以自制 Docker 镜像的形式支持其他框架。

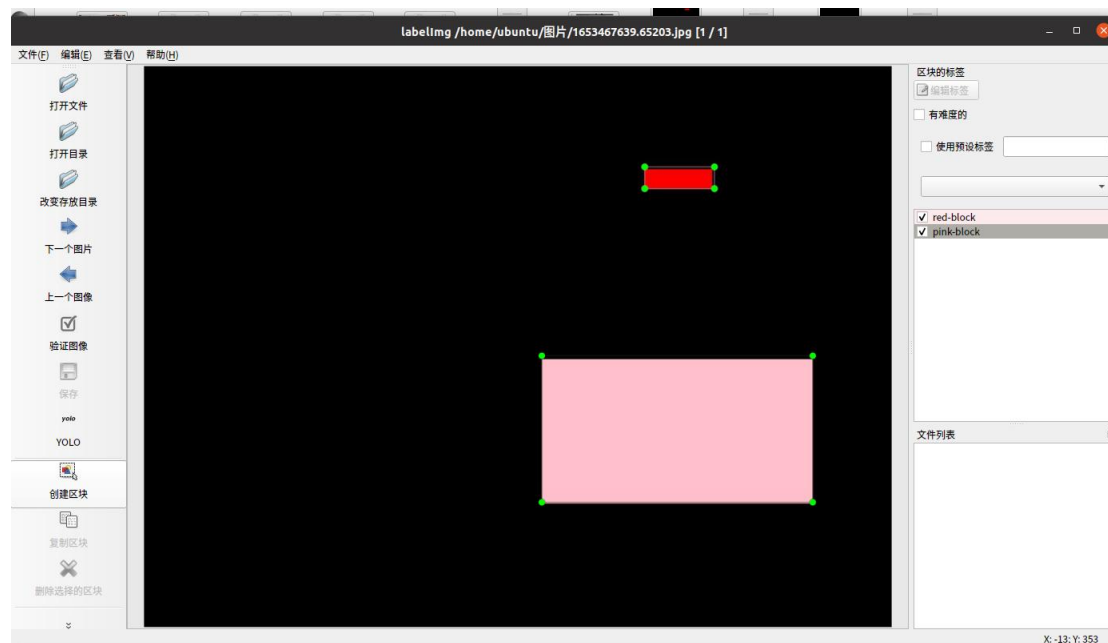
### 4.5.2 数据集构建

基于提供的色块目标数据集，使用 labelImg 数据集标注工具，利用拍摄图片的方法，或者网络获取的方法，扩充数据集并标注。至少扩充标注至 400 张数据集。收集包含多类型目标的图像数据集，例如不同形状、颜色和纹理的物体。对数据集中的图像进行标注，标记出每个物体的位置和类别。

```
conda activate labelme
```

```
labelImg
```

#### 1. 选择标注存储为 yolo 格式



#### 2. 选择 '打开文件'，打开文件。或者点击“打开目录”，打开图像文件夹（路

径不要有中文），选择第一张图片开始进行标注。

3. 点击 '创建区块' 开始画框，光标置于目标左上角，按住左键向右下角拖动。方框绘制完成后，双击选择弹窗中相应的类别名称。完成一张图片后点击“保存”保存（务必保存），此时 txt 文件已经保存到本地了。标注过程中可随时返回进行修改，后保存的文件会覆盖之前的。

注意：

- labelImg 可执行文件存放路径不要有中文，否则会软件打开不成功。classes.txt 文件存储标注类别。
- 在做标注工作前，首先务必执行步骤 1，这样保存的时候会直接保存成 txt 格式。
- 标注完执行 `conda deactivate` 退出当前 conda 环境

#### 4.5.3 Opencv 色块目标识别

完成机械臂视觉的 Opencv 色块目标识别实现。

#### 4.5.4 YOLO 多类型目标检测

对于正常的培训和评估，我们建议使用虚拟环境从源安装软件包。

```
conda activate yolov3
cd PyTorch-YOLOv3/
```

1, YOLO 模型训练:

模型训练说明文档:

```
poetry run yolo-train --help
```

要使用在 ImageNet 上预训练的 Darknet-53 后端在 COCO 上进行训练，请运行  
注意，特别慢，不建议执行：

```
poetry run yolo-train --data config/coco.data --pretrained_weights weights/darknet53.conv.74
```

2, YOLO 模型测试:

```
poetry run yolo-test --weights weights/yolov3.weights
```

### 3, YOLO 模型验证:

```
poetry run yolo-detect --images data/samples/
```

### 4, 用 YOLO 训练标注的多类型目标数据集

#### 1) 设置数据集参数

```
./config/create_custom_model.sh <num-classes> # Will create custom  
model 'yolov3-custom.cfg'
```

#### 2) 在 data/custom/classes.names. 设置数据集类别

#### 3) 数据集图片放置在 data/custom/images/.

数据集标注放置在 data/custom/labels/.

#### 4) 定义训练集和验证集

在 data/custom/train.txt 和 data/custom/valid.txt 文件中, 添加数据集的训练集和验证集信息。

#### 5) 训练模型

```
poetry run yolo-train --model config/yolov3-custom.cfg --data config/cu  
stom.data
```

Add `--pretrained_weights weights/darknet53.conv.74` to train using a backend pretrained on ImageNet.

#### 6) 模型验证:

```
poetry run yolo-detect --images data/samples/
```

