目录

[一、 任务概述 1](#_Toc153742232)

[二、 方案设计 1](#_Toc153742233)

[2.1预处理方案设计 2](#_Toc153742234)

[2.2YOLOV5训练检测 5](#_Toc153742235)

[三、 个人感悟 7](#_Toc153742236)

### 任务概述

1. 识别：分类检测9种的弹珠，其中5种小弹珠（直径15mm），4种大弹珠（直径25mm），每种弹珠的花纹及颜色均不同。
2. 定位：识别图像中托盘的位置，识别弹珠相对于托盘的位置。
3. 封装：将上述功能调试封装，供前端UI调用。

### 方案设计

本项目使用OPENCV检测托盘边框，根据托盘边框将托盘透视变换到正对视角，以此来消除相机位置、托盘位置等因素引起的位置偏差。当得到校正视图后，使用YOLOV5进行位置及类型检测，将得到的数据处理后传出，具体流程如下图所示。



图2.1 总体逻辑流程图

#### 2.1预处理方案设计

首先将图像转换为灰度图并进行阈值处理得到二值图像，目的是进行图像分割或特定区域的提取。这个过程可以突出图像中感兴趣的物体，去除不需要的背景及噪声。为方便现场调试参数，使用Trackbar及回调函数滑动调参。

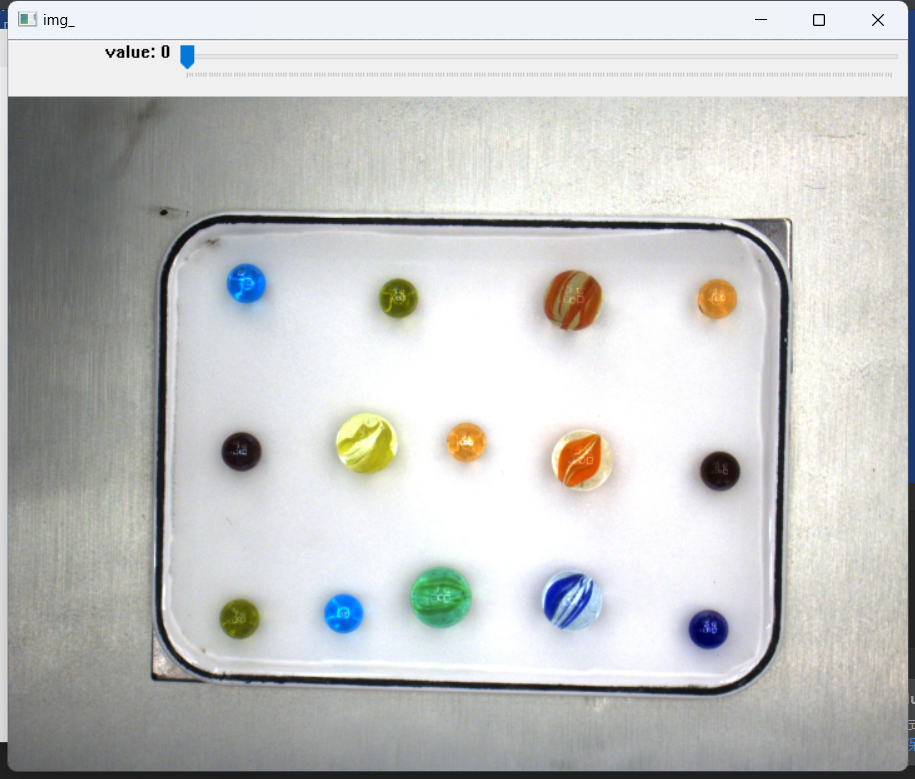


图2.1.1 Trackbar调阈值

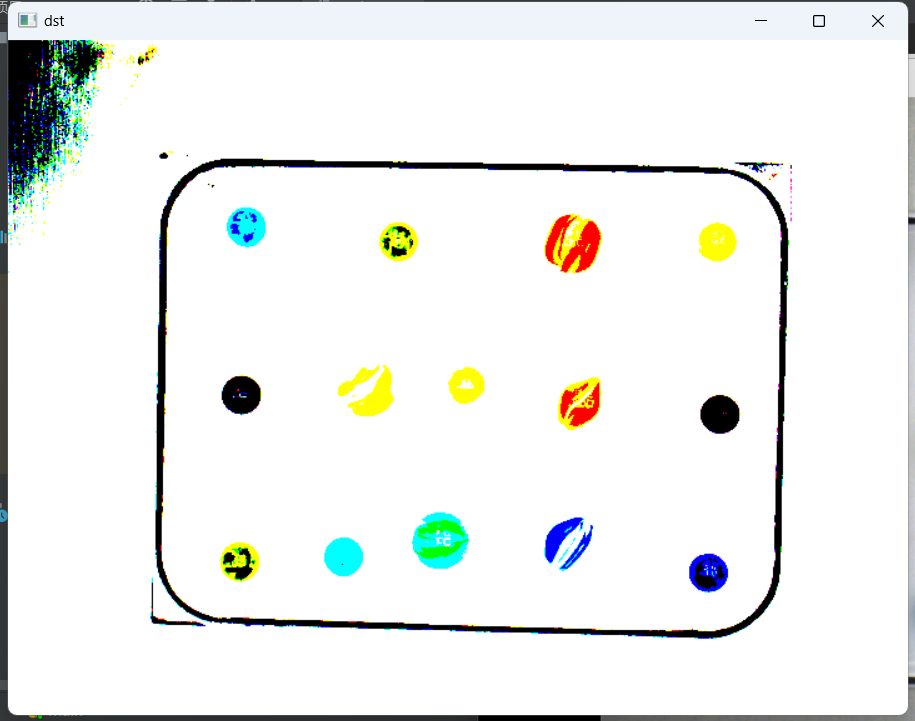


图2.1.2 90阈值效果

接着将二值化后的图像进行Canny边缘检测，找到托盘的外部边界，为下一步边界处理做准备。使用Canny算法能够过滤掉图像中的非关键信息，将图像中的大部分像素点转换为0或1，减少了后续处理所需的数据量，提高检测速度。

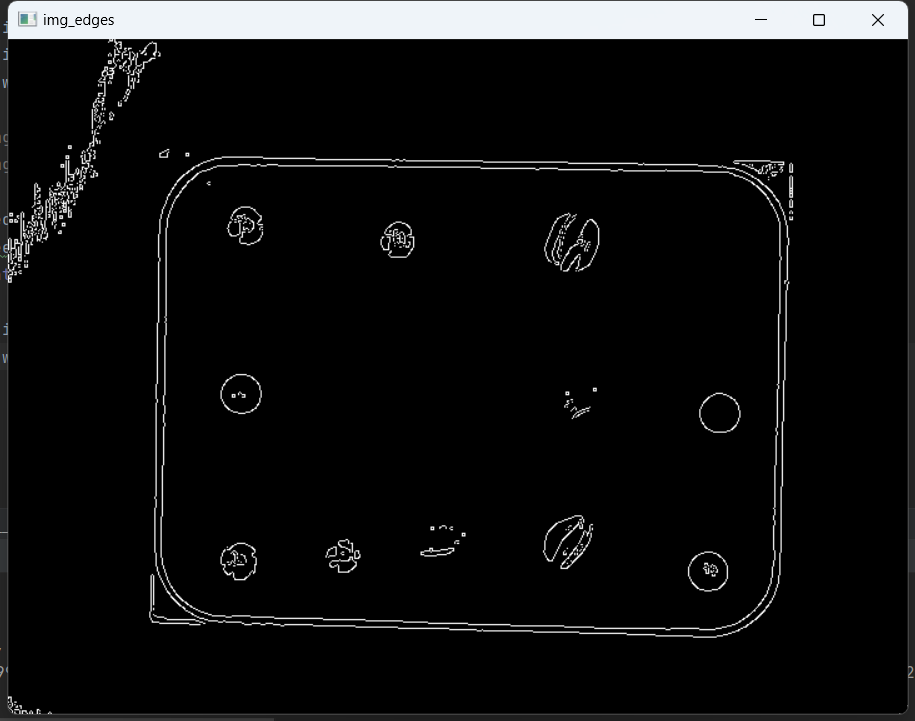


图2.1.3 Canny检测效果

然后将边缘处理后的图像使用霍夫直线检测，找到图像中直线的位置。根据图2.1.4可以看出，在图像中存在大量相近的直线，这些相近的直线会干扰托盘位置的确定并可能导致后续算法的不稳定，因此需要将直线进行数据处理，只保留最可能的四条直线。这里先将直线按斜率分成两大组，接着在每组内使用KMeans聚类算法将直线按照具体保留两个质心，效果如图2.1.5所示。



图2.1.4 霍夫直线检测

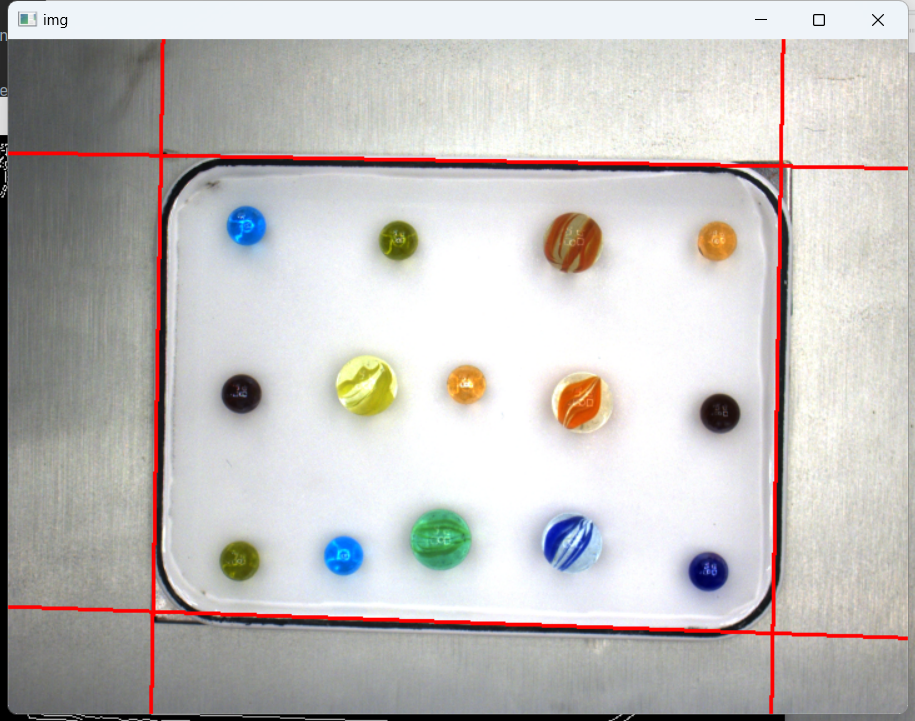


图2.1.5 去除重复直线

最后计算出四条边的交点，根据四个点以及实际托盘的大小将图2.1.6透视变换为图2.1.7。至此图像预处理完成，下一步将预处理后的图像送入YOLO进行分类检测及定位。

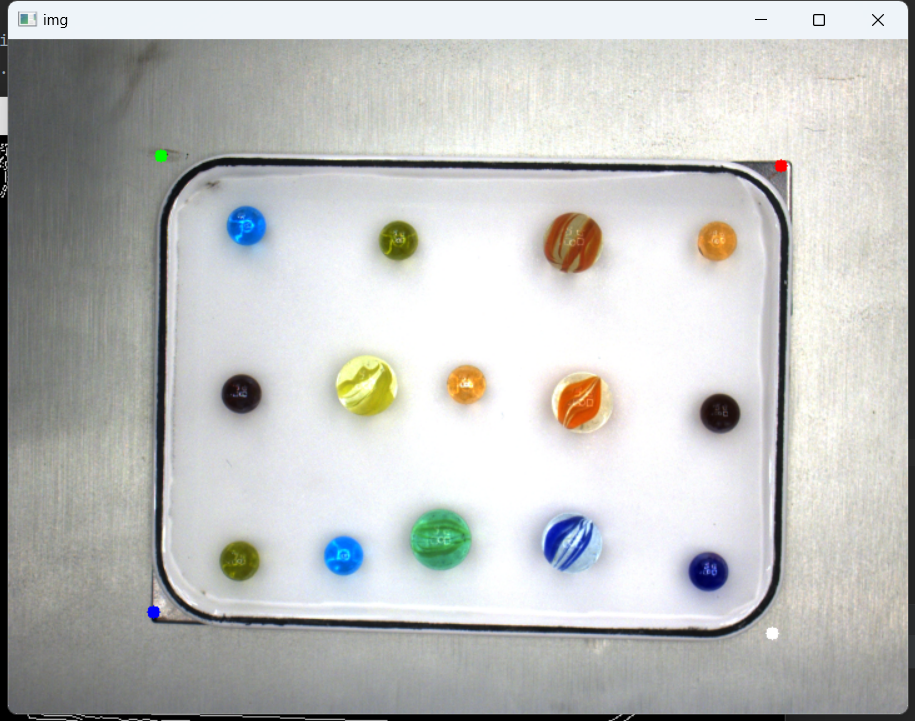


图2.1.6 原始图像

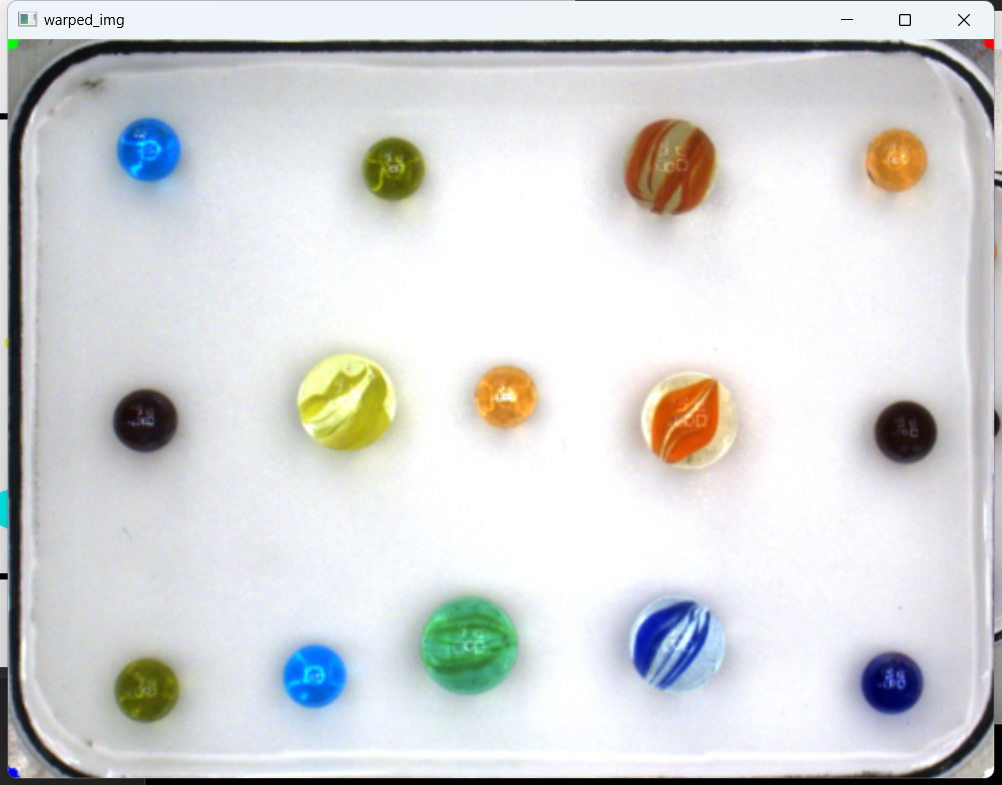


图2.1.7 预处理最终效果

#### 2.2YOLOV5训练检测

首先部署YOLOV5运行环境，下载YOLOV5源码到本地，修改工程结构为如下格式。接着准备数据集，使用labelImg打标，该项目共有9种分类，700张图片。

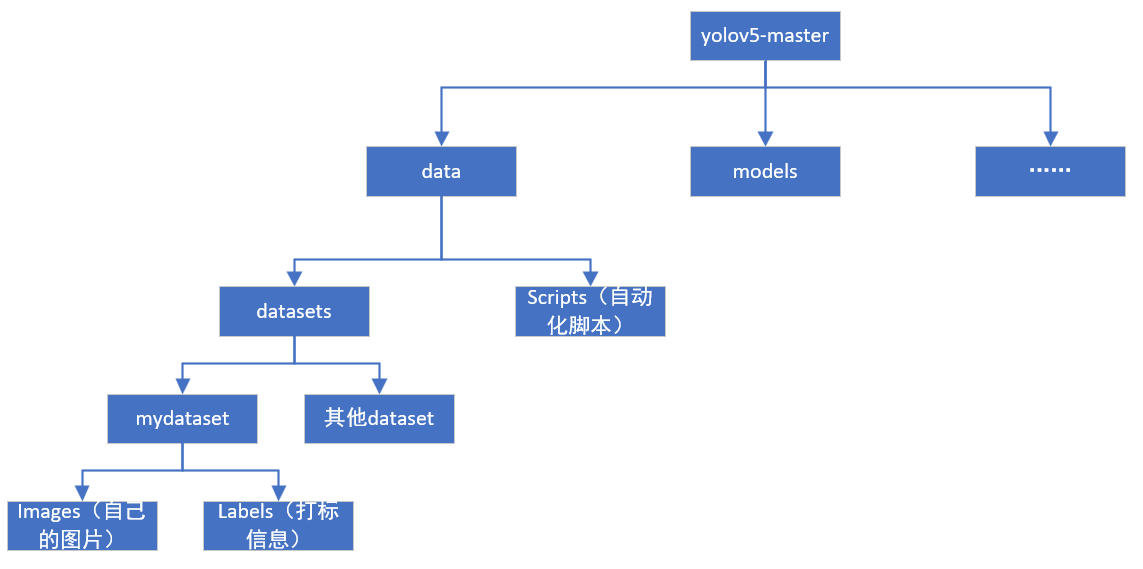


图2.2.1 yolo工程结构



图2.2.2 打标

然后训练模型，使用yolov5s为基础模型，训练次数200次，bench-size为8，训练大约耗费4个小时，最终整体准确率为0.992，mAP为0.5。

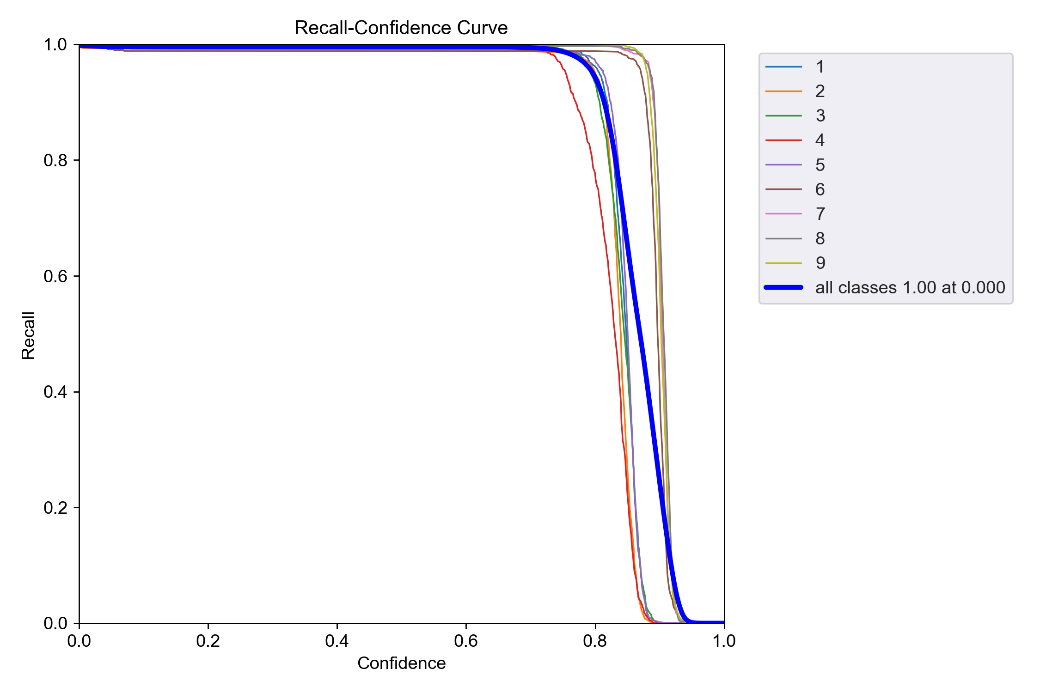


图2.2.3 训练反馈数据

下一步将yolov5的模型部署到OpenCV，首先将pt格式的模型转为onnx格式。接着调用opencv的dnn模块部署模型，将预处理后的图片送入模型检测即可得到弹珠的分类及定位数据。为消除重叠的边界框，只保留最可能是目标的边界框，还需将上述监测数据进行NMS加工，最终得到下图效果。



图2.2.4 最终检测效果