

2025 年首届 “中控杯” 工业 AI 创新挑战赛

正式赛题

一、赛事简介

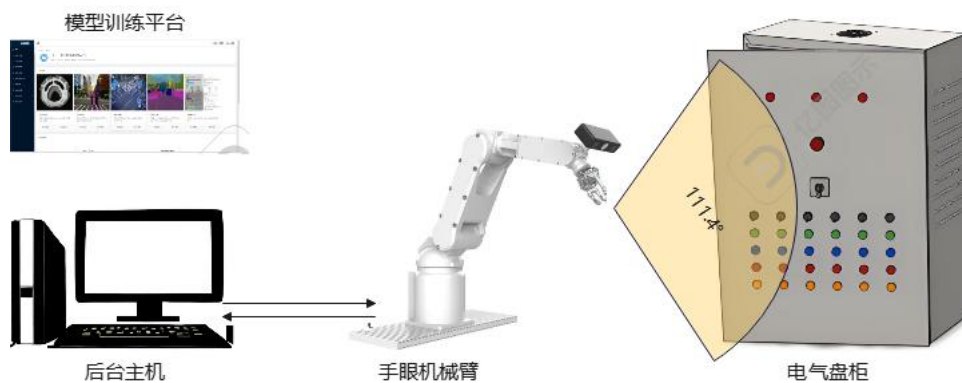
1. **赛事名称：**“中控杯” 工业 AI 创新挑战赛
2. **主办单位：**中控技术股份有限公司

二、赛题名称：AI 视觉创新及应用

（一）参赛背景

该赛题围绕工业企业各类机房、电气室、机柜间无人化巡检的应用需求，通过赛题设计，帮助参赛者了解工业应用场景，提升应用实践与创新思维能力。在移动机器人的巡检场景中，需实现机器人按钮/旋钮操作、指示灯报警识别与点按操作、空开的开关操作、插座插板操作、电气机柜/盘柜的开关门操作等，其精准化操作是该应用场景的难点。通过赛事，鼓励参赛者攻克技术难题，推动工业智能化的创新应用。

（二）比赛平台



（三）比赛过程

基于主办方提供的比赛平台（机械臂、视觉传感器与后台软件组成软硬件平台），参赛选手需要建立针对电气柜、盘柜相关内容的 AI 识别模型，并与机械臂控制相结合，按照赛

题完成规定动作，并输出相应的方案设计说明书、代码源文件、演示视频，并于现场进行演示及答辩。

根据主办方提供软硬件使用的文档资料，以及机械手臂的例程代码。参赛选手在赛前进行相应的开发及准备工作。

1. 根据中控提供标准上位机软件平台，学生通过上位机标准接口，选择合适的算法编写，通过反复训练，建立各类电气零件的 AI 识别模型（颜色识别算法、颜色查找算法、字符识别算法、2D 手眼标定算法、相机内参标定算法、物品定位算法）。
2. 由中控提供机械臂手眼系统硬件平台以及基础功能代码和通用案例，学生通过将 AI 视觉模型与手眼系统的结合，编写应用控制程序模拟实现对指定的机柜、盘柜的巡检与操作，实现如下功能：

- (1) 功能一、指示灯报警识别、表计识别、按钮与旋钮状态识别，识别结果于屏幕上进行展示。

实现步骤：

- A. 编辑机器人轨迹，使摄像头对准屏幕，相机拍摄
- B. 软件调用颜色识别算法，输出颜色名称，上位机根据颜色名称给 plc 信号 使电气柜上对应颜色指示灯亮
- C. 软件调用字符识别算法 识别屏幕上显示的字符 并显示

- (2) 功能二、根据显示屏幕上的内容结果（指示灯、表计状态），依据提供的逻辑表，完成操作电气按钮与空开。

实现步骤：

- A. 识别颜色后，编辑机器人轨迹，使摄像头对准电气柜
 - B. 视觉判断哪个指示灯亮
 - C. 视觉找到该指示灯下的旋钮
 - D. 编辑机器人轨迹：将旋钮旋到右边；（非视觉引导，旋钮位置固定，需编写 3 套机器人轨迹，分别对应三个旋钮；上位机根据算法输出的指示灯编号及旋钮方向，自动调用对应的机器人轨迹）
- (3) 功能三、打开电气柜门，将指定区域随机摆放的零件，放入电气柜指定位置，之后关闭柜门。

实现步骤：

- A. 编辑机器人轨迹将机器人抬到指定位置，进行 2D 手眼标定及相机内参标定
- B. 将物品放置在指定区域内任意位置，根据手眼标定数据和内参标定数据算出机器人抓取的坐标，再将其放置到固定位置

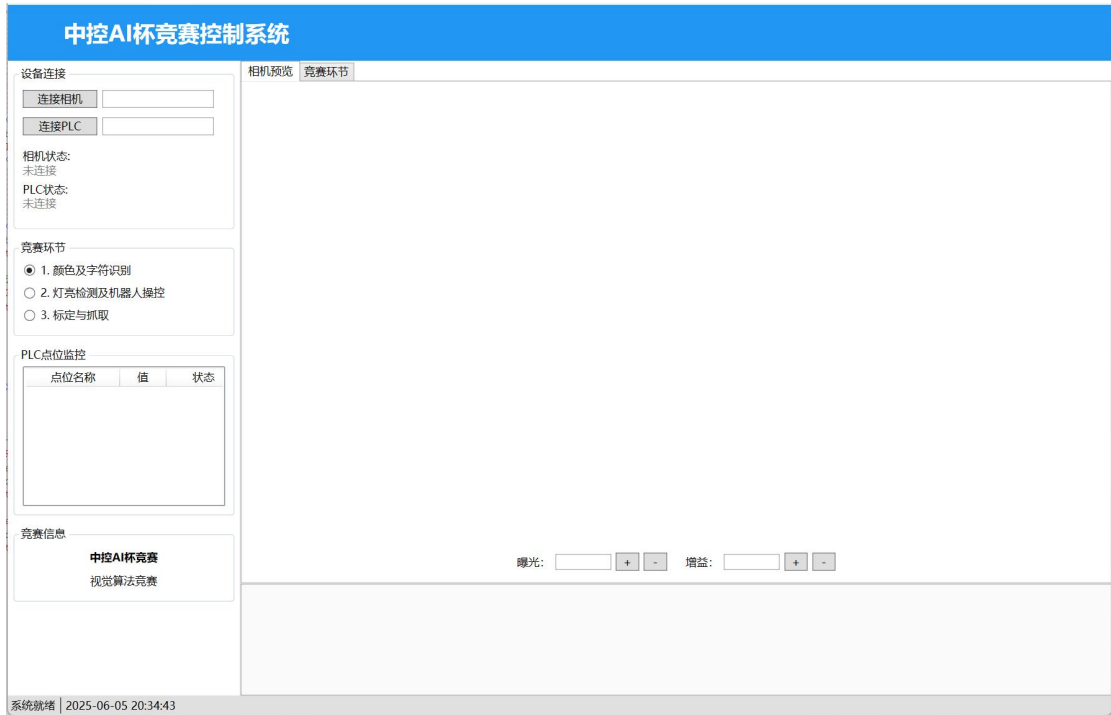
(四) 评比制度

1. 上述三项功能难度设置从易到难，为鼓励挑战高难度，采用赋分制，挑战难度越高，则赋分更高。依据上述原则，出具评分细则。
2. 参赛选手任选上述三项功能的一项或多项，进行实践。由现场评委从演示效果、设计文档、代码规范等方向上，依据评分细则，进行综合评定。
3. 比赛成绩以总分高低为准，完成上述三项功能中任何一项功能均可获奖。

(五) 软硬件平台说明

1. 软件说明

1.1 主界面介绍



1.1.1 左侧功能分区详解

(1) 设备连接模块

连接相机：一键初始化工业相机，支持 USB/GigE 等协议。

连接 PLC：通过 Modbus/TCP 等协议与 PLC 建立通信。

(2) 状态显示

相机状态：实时显示连接状态（如“未连接”或“已连接@192.168.1.100”）。

PLC 状态：指示 PLC 通信是否正常，异常时标红提示。

(3) 竞赛环节模块

颜色识别与指示灯控制：点击后切换至环节 1 页面

旋钮方向判断与操作：点击后切换至环节 2 页面

物品放置与抓取：点击后切换至环节 3 页面

1.1.2 右侧操作/监控区详解

(1) 相机预览与算法结果标签页

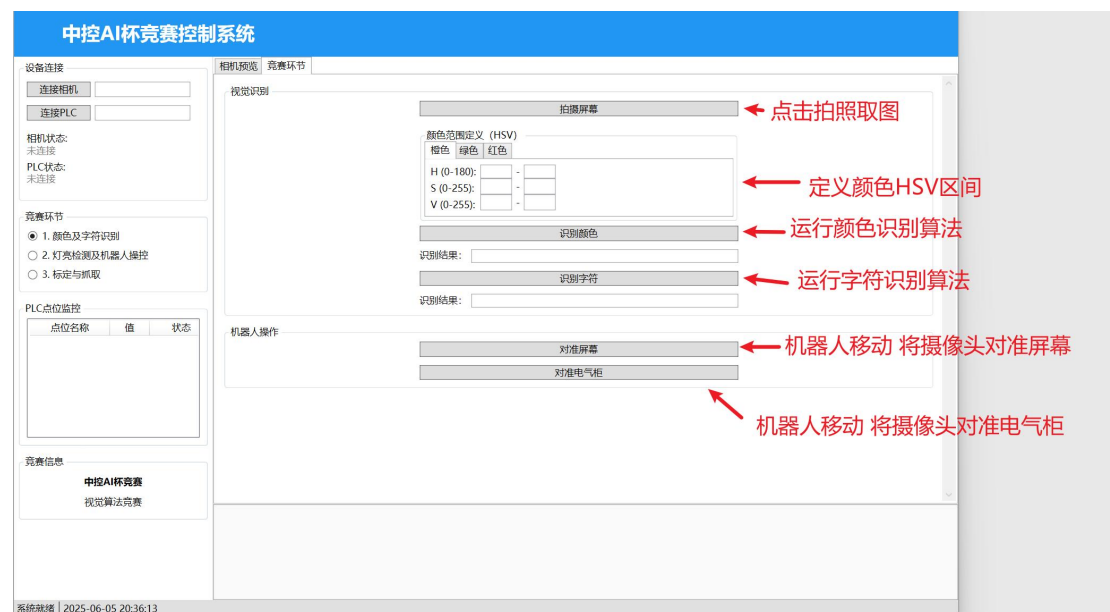
- 相机预览：
 - 实时显示工业相机采集的画面

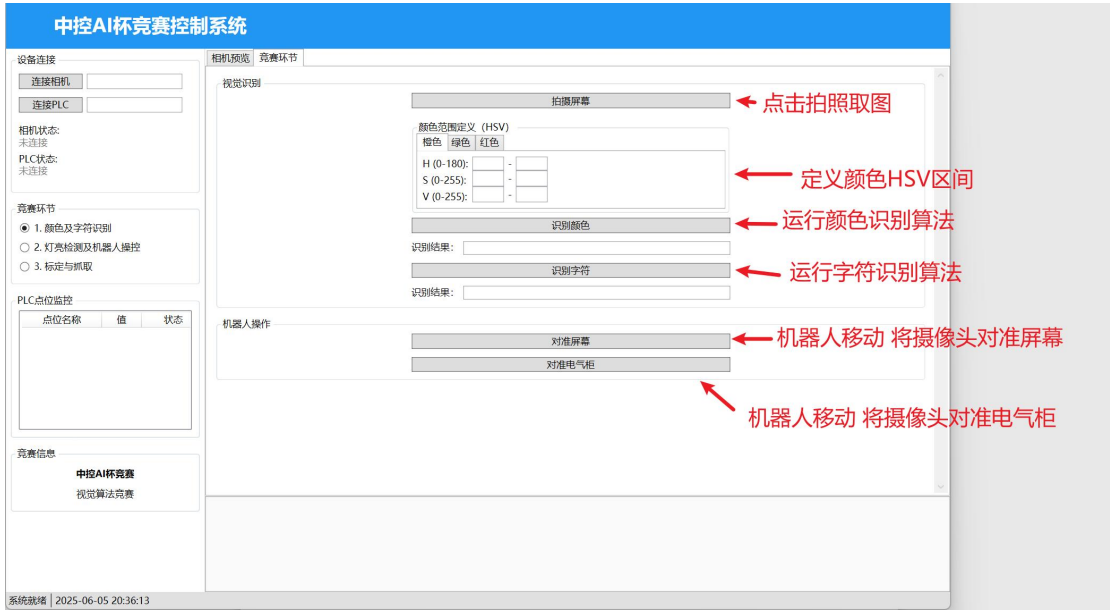
(2) 曝光与增益调节栏

- 调节控件：
 - 曝光：数字输入框配合±按钮，步长 1。
 - 增益：独立调节栏，逻辑同曝光。
 - 实时生效：参数调整后立即同步到相机驱动，无延迟。

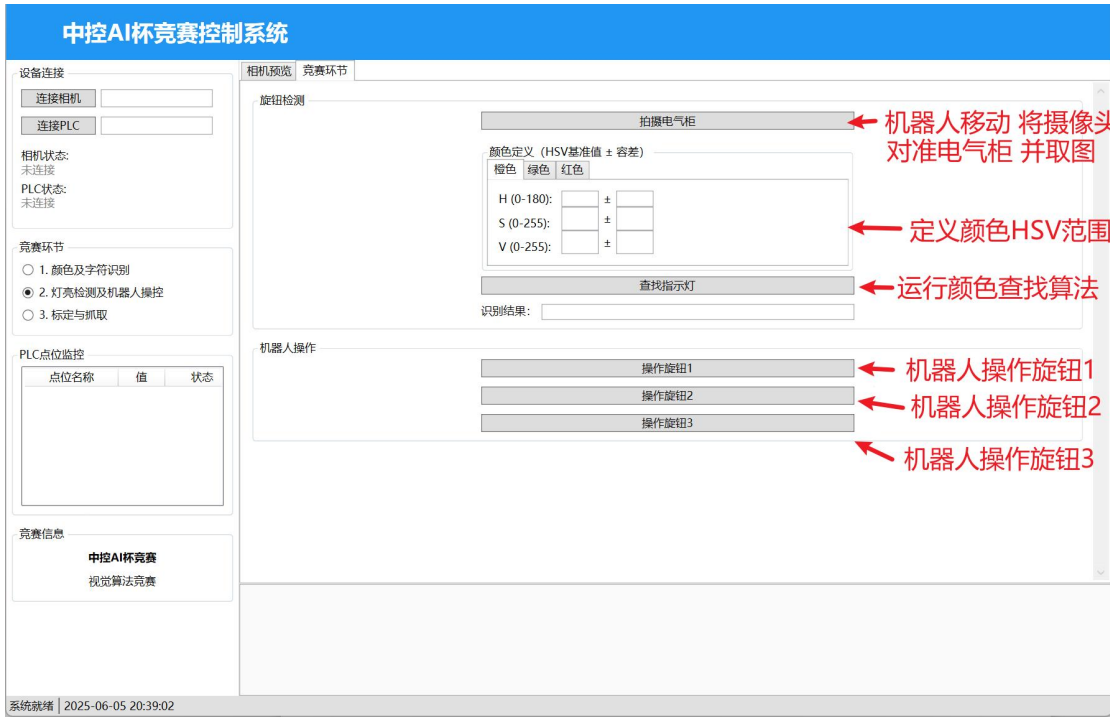
2. 竞赛环节页面说明

2.1 环节 1

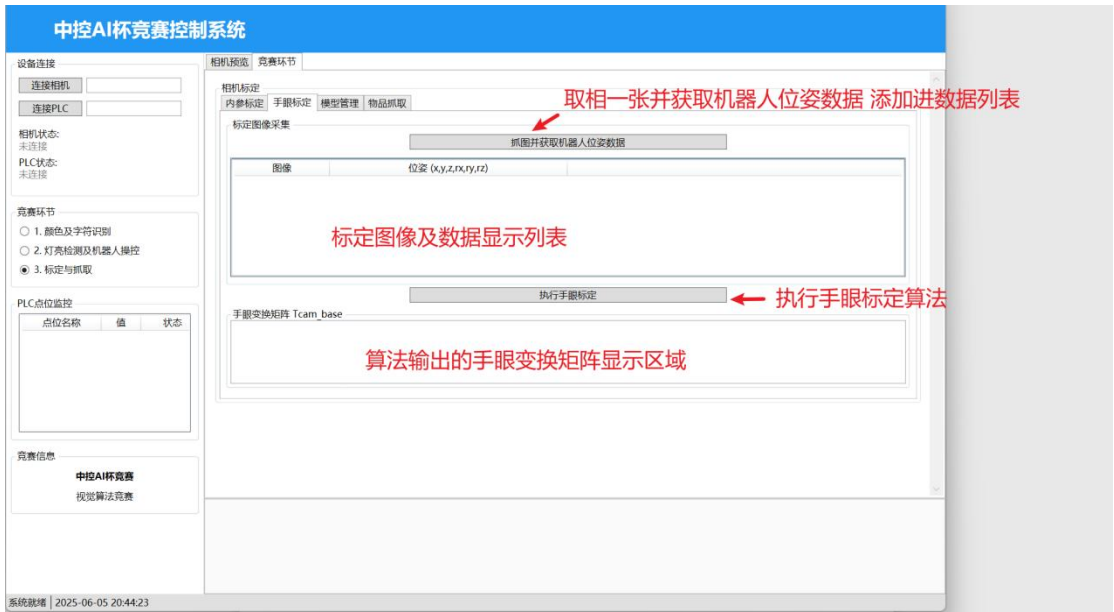
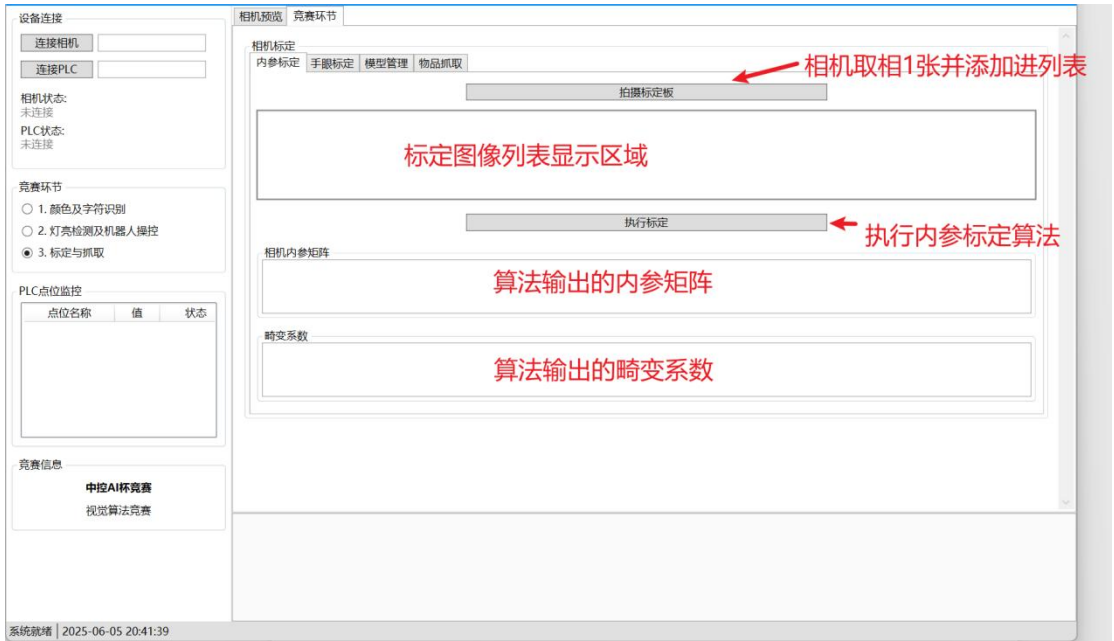


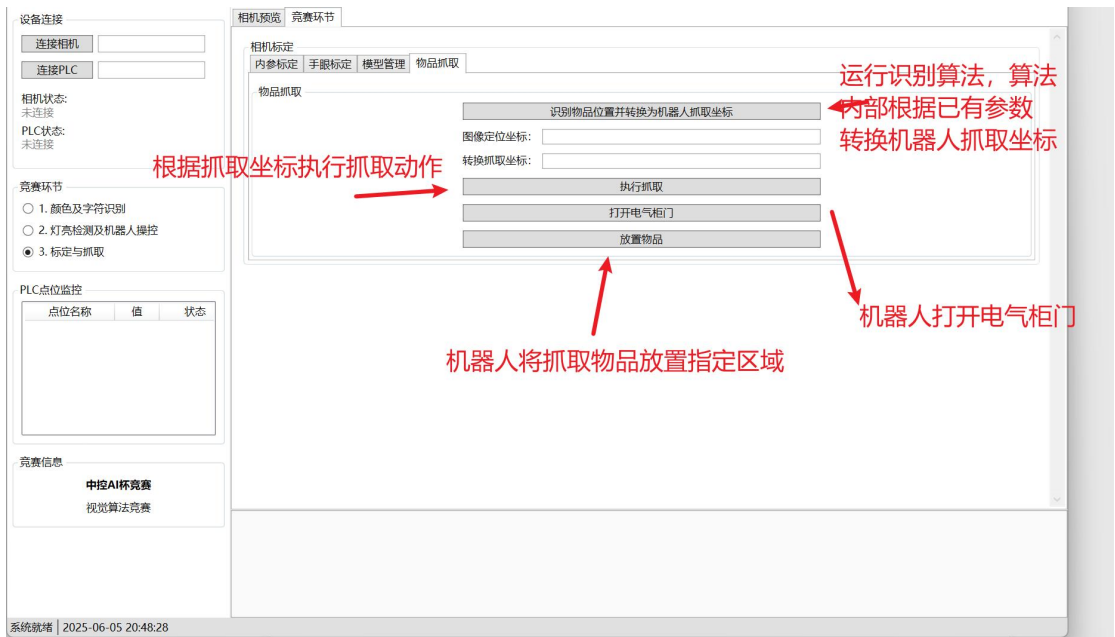


2.2 环节 2



2.3 环节 3





3. 算法接口说明

本系统要求所有动态库（DLL/SO）的对外接口，必须采用 extern "C" 链接规范，即 C 语言兼容的函数接口（C-Compatible Function Interface）。

3.1 颜色识别算法

```
/**
 * @brief 对一张纯色或主色调明显的图像进行颜色分析，返回 HSV 空间下的主颜色值
 *
 * @param image      输入图像，要求为 BGR 彩色图（CV_8UC3），建议背景纯色或主色显著
 * @param hsv_color   输出颜色的 HSV 值数组，长度为 3（hsv_color[0]=H, hsv_color[1]=S, hsv_color[2]=V）
 *
 * @return           true 表示识别成功，false 表示失败（如图像为空或格式不对）
 *
 * @note
 * - 图像必须是 BGR 格式的彩色图（非灰度），函数内部将图像转换为 HSV 空间
 * - 推荐图像尽量避免大面积黑色或白色背景，以提高准确度
 * - 返回的 HSV 值用于后续颜色分类、分析或阈值匹配
 */
extern "C" API bool RecognizeDominantColorHSV(cv::Mat *image,
                                              double *hsv_color);
```

3.2 颜色查找算法

```
/**
 * @brief 在图像中查找指定颜色区域（基于 HSV 空间）
 *
 * @param image      输入图像（BGR 格式，CV_8UC3）
 * @param target_hsv 目标颜色的 HSV 值数组，长度为 3（hsv_color[0]=H, hsv_color[1]=S, hsv_color[2]=V）
 * @param tolerance_hsv HSV 容差范围数组，长度为 3（±H, ±S, ±V）
 * @param mask       输出掩膜图（255 表示目标颜色区域，0 表示非目标区域）
 * @param contours   输出颜色区域的轮廓（可选，如不需要可传空指针）
 *
 * @return           返回 true 表示成功识别颜色区域，false 表示失败或图像无效
 *
 * @note
 * - 输入图像必须为 BGR 格式（CV_8UC3）
 * - HSV 更适合颜色识别，建议将 BGR 图像转换后在 HSV 中设定颜色范围
 * - tolerance_hsv 可适当调整以适配不同光照条件或相机色彩漂移
 */
extern "C" API bool FindColorRegionHSV(cv::Mat *image,
                                       const double *target_hsv,
                                       const double *tolerance_hsv,
                                       cv::Mat *mask,
                                       std::vector<std::vector<cv::Point>> *contours = nullptr);
```

3.3 字符识别算法

```
/**
 * @brief 使用模板匹配在图像中检测目标
 *
 * @param image          输入图像（BGR格式）
 * @param template_image 模板图像数组（BGR格式 多个模板）
 * @param out_center_x   输出匹配区域左上角 x 坐标
 * @param out_center_y   输出匹配区域左上角 y 坐标
 * @param out_score      输出匹配得分（越大表示越相似，范围：0 ~ 1）
 *
 * @return               返回 true 表示成功匹配，false 表示失败（如图像无效）
 */
extern "C" API bool ObjectDetect(cv::Mat *image,
                                cv::Mat **template_path,
                                int *out_x,
                                int *out_y,
                                float *out_score);
```

3.4 2D 手眼标定算法

```
/**
 * @brief 执行手眼标定，计算相机坐标系与机械臂基坐标系之间的变换矩阵 T_cam_base
 *
 * @param images          输入图像指针数组（每个元素为指向灰度或彩色图像的 cv::Mat*）
 * @param image_count     输入图像数量（必须与 images 数组大小一致）
 * @param robot_poses     机器人末端在基坐标系下的位姿数组
 *                        按 [x, y, z, rx, ry, rz] 展开
 *                        每组 6 个 double，总长度应为 image_count * 6
 * @param hand_eye_transform T_cam_base 手眼变换矩阵（4x4 CV_64F）表示 相机坐标系与机械臂基坐标系的转换关系
 *
 * @return               返回 true 表示标定成功，false 表示失败或数据不足
 *
 * @note
 * - 输入的 images 和 robot_poses 要一一对应，数量 > 15 组
 * - robot_poses 是笛卡尔坐标，按每组 [x, y, z, rx, ry, rz] 存储
 * - robot_poses 每个参数的变化量要尽量明显，有助提高鲁棒性
 */
extern "C" API bool CalibrateHandEye(cv::Mat **images,
                                     int *image_count,
                                     double *robot_poses,
                                     cv::Mat *hand_eye_transform);
```

3.5 相机内参标定算法

```
/**
 * @brief 执行相机标定，计算相机内参和畸变参数
 *
 * @param images      输入图像指针数组（每个元素为指向灰度或彩色图像的 cv::Mat*）
 * @param image_count 输入图像数量（必须与 images 数组大小一致）
 * @param camera_matrix 输出的相机内参矩阵（3x3 CV_64F），包括焦距和主点坐标
 * @param dist_coeffs 输出的畸变系数（通常为 1x5 CV_64F）
 *
 * @return            返回 true 表示标定成功，false 表示失败
 *
 * @note
 * - 所有图像应尽量包含完整清晰的标定板图案
 * - 标定过程依赖于每张图像中都能成功检测到标定板的标志点
 * - 建议图像数 ≥ 15 幅，视角分布要尽可能全面
 * - 相机内参可用于图像矫正、3D 重建、视觉测量等场景
 * - 此函数内部不进行图像展示或文件保存，调用者可自行扩展
 */
extern "C" API bool CalibrateCamera(cv::Mat **images,
                                     int *image_count,
                                     cv::Mat *camera_matrix,
                                     cv::Mat *dist_coeffs);
```

3.6 物品定位算法

```
✓ /**
 * @brief 使用模板匹配在图像中检测目标
 *
 * @param image      输入图像（BGR格式）
 * @param template_image 模板图像数组（BGR格式 多个模板）
 * @param out_center_x 输出匹配区域左上角 x 坐标
 * @param out_center_y 输出匹配区域左上角 y 坐标
 * @param out_score   输出匹配得分（越大表示越相似，范围：0 ~ 1）
 *
 * @return            返回 true 表示成功匹配，false 表示失败（如图像无效）
 */
extern "C" API bool ObjectDetect(cv::Mat *image,
                                  cv::Mat **template_path,
                                  int *out_x,
                                  int *out_y,
                                  float *out_score);
```

4. 主要硬件说明

序号	名称	数量	品牌
1	设备框架	1	中控
2	控制系统	1	EPSON
3	软件	1	中控
4	服务器	1	研华
5	EPSON 机器人	1	EPSON
6	安装结构件	1	中控
7	识别算法	1	参赛者

5. 机械手使用方法说明

详情见下方连接 0

机器人资料: https://pan.baidu.com/s/1J27F3A6ol_HSzm5b8jm1mA 提取码: sq53

仿真软件安装说明: <https://doc.eleok.com/1007.html>

6. 其他说明

赛题设计及比赛所涉软硬件平台或可根据比赛现场进行微小调整, 若不影响竞赛内容及评比标准, 不再另行通知。

