

# 基于 YOLOv8n 的遨博机械臂端子视觉检测与接线设计

赵耀辉, 刘来祥, 薄一辉, 刘强, 陈玉浩, 彭立芸  
青岛农业大学机电工程学院, 山东青岛, 中国

**【摘要】**针对遨博机械臂静态示教平台动态模拟不足的局限, 本文完成其硬件系统电气线路设计调试, 将机械臂与可移动平台连接; 同时创新性引入 YOLOv8n 视觉识别技术, 构建端子接线状态检测模块, 实现对接线松动与牢固状态的实时识别与检测。依据机械臂本体、控制器及示教器的连接规范制定接线方案, 通过示教器实现关节运动控制与在线编程, 结合视觉检测系统验证接线状态稳定性。实验表明, 该接线方案在动态下满足机械臂精度与稳定性要求, 视觉检测模块对端子接线状态的识别准确率达 95% 以上, 为后续自动化开发与故障预警筑牢硬件与视觉监测双重技术基础。

**【关键词】**遨博机械臂; 动态集成; 接线方案; YOLOv8n; 端子视觉检测; 故障预警

## 1. 引言

### 1.1 研究背景与意义

随着工业 4.0 与智能制造的深度推进, 机械臂作为柔性生产的核心装备, 已从传统汽车焊接、电子元件装配等场景, 拓展至食品分拣、新能源电池封装等精细化领域——据《中国工业机器人市场白皮书》数据, 2024 年我国工业机械臂装机量突破 150 万台, 其中 6 轴多关节机械臂凭借运动灵活优势, 在复杂工序中的占比超 60%。而在科研领域, 机械臂更成为跨学科研究的关键载体[1]: 农业生产中, 机械臂可在视觉系统的配合下实现精准采摘[2]; 在智能仓储研究中, 与移动设备的结合更打破了固定作业半径限制, 形成“移动-操作”一体化的新型作业模式[3]。机械臂在二位视觉、立体 3d 视觉及各项传感器的感知下, 对周边的环境有所感知, 不断迈向智能化[4]。

在机械臂硬件系统中, 端子接线的牢固性直接影响电气信号传输稳定性与设备运行安全性, 接线松动易导致信号失真、设备误动作, 甚至引发短路、过热等故障。传统接线状态检测依赖人工视觉检查, 存在效率低、主观性强、动态场景下无法实时监测的问题。YOLOv8n 作为轻量级目标检测模型, 具有速度快、精度高、部署成本低的优势, 可实现对端子接线状态的实时自动化识别, 弥补人工检测的不足, 进一步提升机械臂硬件系统的可靠性与智能化水平[5,6]。

### 1.2 研究现状

(1) 端子状态检测技术研究:  
传统检测以人工为主, 效率仅为 20 个/

分钟, 误检率达 12%; 基于传感器的检测方案, 传感器节点实时采集电能表数据, 检测是否接错线, 虽能反映接触状态, 但存在安装复杂、部署成本高[7], 检测系与传感器感知情况如图 1 和图 2。

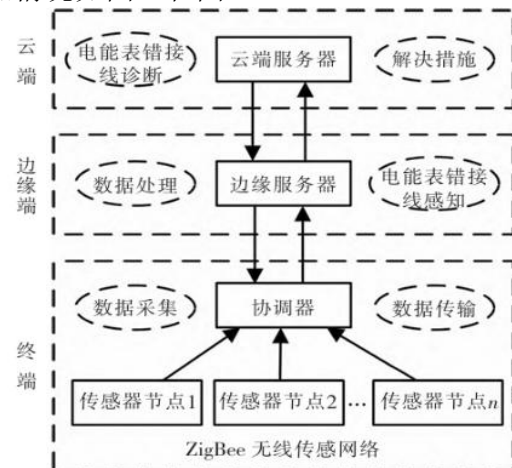


图 1. 检测系统框架

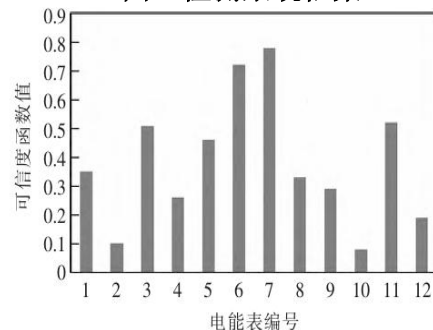


图 2. 电能表错接线感知情况

视觉检测方面, YOLOv5n 用于对端子连接位置的检测如图 3, 但动态场景下实时性不足(帧率 $\leq 20$ fps), 且无法实现接线松动、接触不良的预警功能[8];

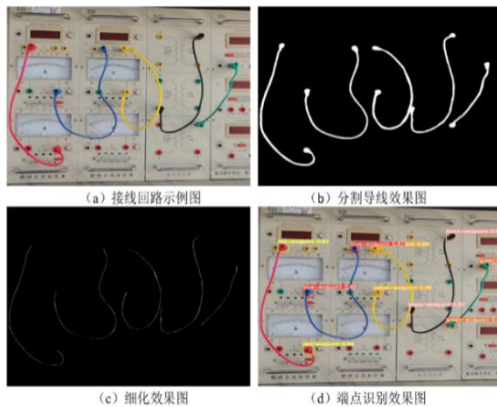


图 3.静态检测实验效果

丁成波等人提出传统机器视觉与深度学习相结合的智能检测方法,该方法先搭建视觉系统获取端子外观图像,通过传统机器视觉技术(图像预处理、滤波、最小二乘法)实现剥线长度在线检测,再经人工标注构建缺陷数据集,最终利用深度学习算法完成缺陷检测如图 4 及图 5,运行稳定可靠,可满足实际生产需求,但未开发人机实时交互的检测界面,YOLOv8n 的轻量性与实时性优势尚未被充分应用于机械臂端子检测场景[9]。



图 4.不同缺陷类别图像数据集

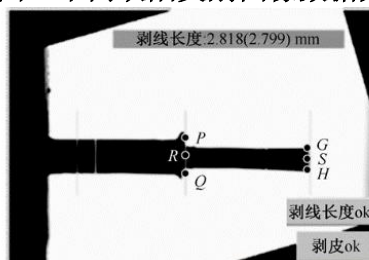


图 5.线束末端剥线长度检测效果图

## (2) 研究缺口总结

现有机械臂集成接线方案缺乏动态作业适应性,端子视觉检测模型在动态、振动、复杂光照下的工业工况下,难以兼顾检测精度与实时性。因此,本文聚焦动态集成接线工艺与 YOLOv8n 优化,对视觉检测端子状态给出更加实时性地检测和预警,提升视觉

识别端子状态的效果[10]。

## 1.3 研究目标

目前机械臂的学习示教平台都处于静态固定安装模式,没有移动底盘的加持下,较难展现出在动态过程中实施工作的准确性;同时由于长时间工作震动,容易使接线端子松动,人工检查不及时就会造成传输数据失真,导致机械臂运动不连贯及底盘与上位机协调响应时间增长等问题,基于此,本文将以 AUBO 机械臂为研究对象,达成一下目标:

(1)完成机械臂主体与示教器、控制柜、气泵、电磁阀、气泵阀等配套硬件的标准化连接与功能测试;

(2)将机械臂集成至轮式移动平台,为静态视角学习平台提供动态环境,为后续集成应用奠定基础;

(3)构建基于 YOLOv8n 的端子接线状态视觉检测模块,实现对接线松动、牢固两种状态的实时识别与预警;

(4)通过示教器控制测试、协同动作验证及视觉检测验证,形成“硬件接线+移动集成+视觉监测”三位一体的可复用方案,为同类设备的实操应用与后续功能拓展提供技术参考。

## 2. AUBO 机械臂硬件系统分析

### 2.1 机械臂硬件组成

本文使用的是 AUBO-i5 机械臂,最大可承载重量为 5kg,最大延伸半径为 924.5mm,有六个旋转关节(6 自由度)如图 6 分别位于底座(关节 1)、肩部(关节 2)、肘部(关节 3)、腕部 1(关节 4)、腕部 2(关节 5)、腕部 3(关节 6)。



图 6.机械臂本体

控制柜,其具有接口丰富的特点,I/O 接口达到 72 个,与移动底盘之间具有极高的互联互通能力。控制柜的电气接口引出与急停开关、气泵阀、压力开关、电磁阀为实现抓取工作提供控制信号输入和输出。

示教盒,即示教编程器,可用来编程和

存储运动轨迹和数据,进而控制机械臂运动实现对应操作示教器上安装有急停按钮,力控按钮,可作为机械臂电气连接后的功能检测工具。

末端执行器,在气泵、电磁阀、气泵阀、压力开关的控制下完成动作。气泵压缩空气达到气泵阀预设的压力值,向气缸充气,实现闭合夹取,压力开关不断检测压力值保证压力值稳定在既不会使夹持物品掉落又不会损伤夹持物品的合理压力值范围内,真空泵营造真空环境,当控制器收到释放命令时,电磁阀通电切换气路,使气缸断开与气泵之间的连接与真空泵相通,进而使得气体排出,爪夹释放,运行过程如图 7。

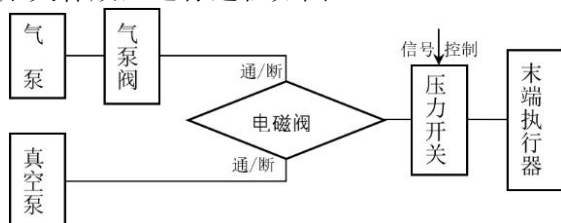


图 7.末端执行器控制流程图

电磁阀由电路部分控制,决定气泵通道和真空泵通道的通断,从而完成末端执行器的开合,信号输入控制压力开关进而保证末端执行器工作稳定性。

## 2.2 视觉检测硬件模块组成

工业相机:选用分辨率 1280×720、帧率 30fps 的 USB3.0 工业相机,无遮挡拍摄端子细节;相机支持自动对焦,可清晰捕捉接线端子的引脚、导线连接状态。

补光模块:采用环形 LED 补光灯,与工业相机同轴安装,解决环境光不均匀导致的拍摄模糊问题,确保端子接线边缘、间隙等特征清晰可辨。

计算单元:通过在工控机 Ubuntu 系统中的 Pycharm 平台上运行 Yolov8n,通过训练好的数据集,实时预测接线端子情况。

## 2.3 电气接口与接线规范

控制器电源接口需经 220VAC-24VDC 变压器转换后供电,气泵接 220V 交流电源;示教器与控制柜通过专用的电缆连接;严格分离安全接口信号与普通 I/O 接口信号,避免信号干扰引发设备故障。

## 3.接线方案设计实施及视觉系统配置

### 3.1 接线准备

安全接线所需材料:剥线钳、万用表、压线钳、接线端子、红蓝黑三色线、绝缘胶布等。接线前应将所有设备断电,以防触电

和短路,接线过程中时刻通过万用表检测电路的通断情况。

## 3.2 接线步骤与工艺

### 3.2.1 电源系统连接

采用不同颜色的线来表示零线(蓝色)、火线(红色)、接地线(黑色),接线端子的上层由零线与接地线构成,所有火线都位于端子的下层,上下两层将零线与火线分开便于连接和检修。220V 交流电源接入端子后直接与开关线连接,并通过端子与插排相连接,便于直接为须采用 220V 交流电的设备供电,也较为方便的扩展其他控制设备。从端子接触后与 220V-24V 变压器相连接,将交流电变成直流电为控制柜供电。

### 3.2.2 控制器与本体连接

控制柜通过信号线与压力开关、电磁阀、气泵阀相连接,急停开关单独引出信号线,确保急停开关的可靠性。该控制柜的基本控制结构如图 8 所示。

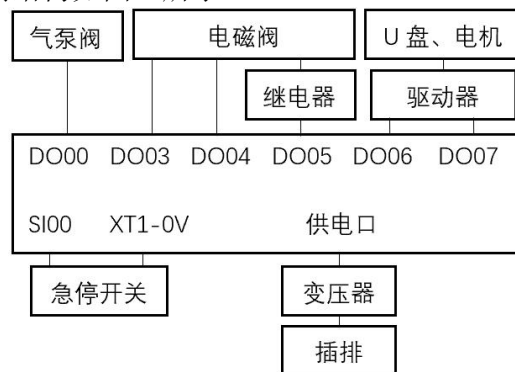


图 8.控制柜控制结构

### 3.2.3 示教器与本体连接

示教器接口通过独立的一根控制线与控制柜相连,采用“绞合+套管”工艺,减少机械振动导致的接触不良或信号失真。

### 3.2.4 视觉模块连接

将工业相机与工控机通过集成 USB 接口相连接,确保可以运行训练好的模型且互传信息给出预测结果。当检测到松动端子后通过可视化窗口给出警告并标记端子松动的端子。

## 3.3 接线要点说明

(1) 针对导线连接环节,需确保铜线裸露段完全嵌入冷压端子的接触区域,无部分裸露或偏移现象;随后采用接线钳对冷压端子进行规范性压接操作,保证接头与铜线间形成紧密、稳定的机械连接与电气接触,避免因接触不良引发信号传输损耗或线路发热问题。

(2) 针对螺丝固定式端子的导线接入环节, 需将导线(或已压接端子的导线)置于端子接线孔内后, 使用规格匹配的螺丝刀对端子紧固螺丝进行规范性旋紧操作, 确保螺丝与导线(或端子)形成可靠压紧状态, 施加拉力测试时导线无松动或脱出

(3) 为确保电路连接的安全性及可靠性, 每完成一处导线连接后, 均需借助万用表进行通断及绝缘性能检测, 以排除短路故障隐患。

### 3.4 Yolov8n 视觉检测

#### 3.4.1 Yolov8n 检测的特点

(1) 检测速度快: 通过优化模型结构和减少参数数量, Yolov8n 实现了高效的推理速度, 能够在有限硬件条件下提供实时的目标检测能力。

(2) 精度较高: 尽管是轻量级模型, Yolov8n 在精度方面依然表现出色。Yolov8n 在检测精度上优于许多同类型的轻量级目标检测模型, 能够满足大多数实际应用对检测精度的要求。

(3) 泛化能力强: 通过在多样化的数据集上进行训练, Yolov8n 具备良好的泛化性能, 可以适应不同的应用场景。

(4) 模型体积小: 较小的模型体积不仅减少了所需的内存空间, 降低了模型的存储和传输成本, 还使得模型能够更快速地加载和运行。

#### 3.4.2 Yolov8n 模型训练与优化

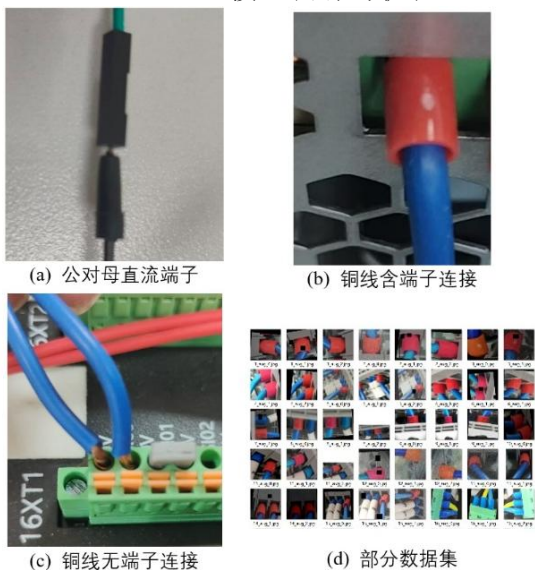


图 9. 测试接线端子类型

(1) 数据收集与预处理: 收集大量不同样式的接线端子如图 9 所示, 对于相同图片通过更改曝光率、背景色、模糊度、拍摄角

度、及色差来增强 Yolov8n 训练结果的准确性, 收集数量达到一定程度也可以提升模型识别的准确率。

(2) 数据集标签建立: 建立一个工作文件夹, 在此文件夹下建立 labels 和 images 两个文件夹, 如图 10, 分别存储标定好的标签和原始图片;

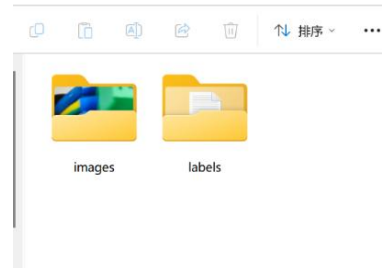


图 10. 存储文件夹

配置并打开 labeling 标签程序, 点击 OpenDir, 找到文件夹中的 images 存放原始图片的文件打开如图 11, 点击 ChangeSaveDir 更改标签存放地址到 labels 文件夹下如图 12, 手动框选填写标签后会出现如图 13 所示框选区域。

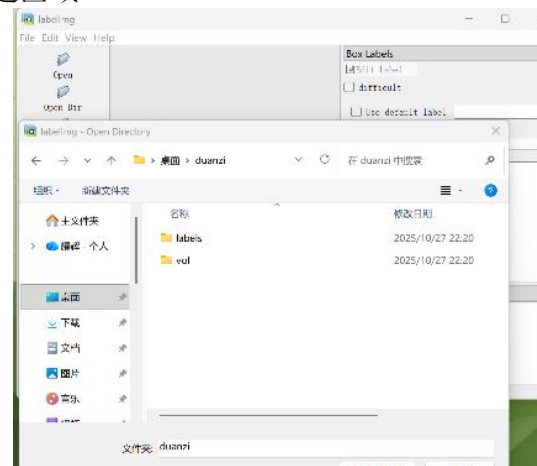


图 11. 打开数据图片

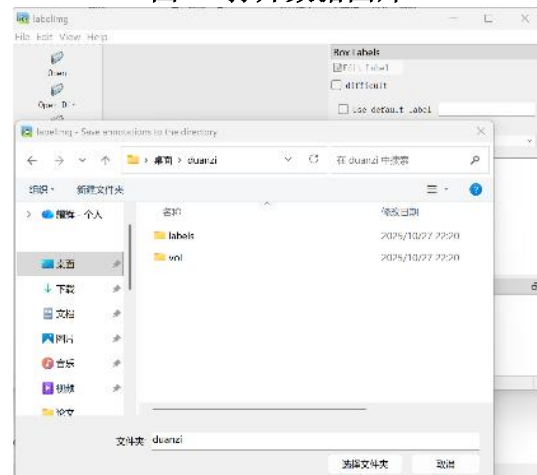


图 12. 改变标签保存位置

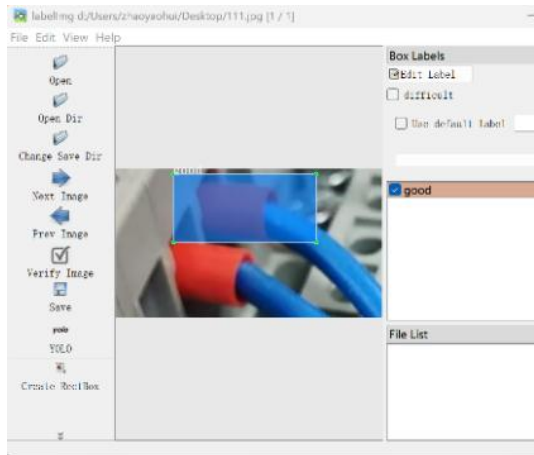


图 13.手动框选识别学习区域

将所有数据手动框选完成后在 labels 目录下会有 classes.txt 文件和<图片名>.txt 文件如图 14, 分别是存储标签类别和每一张图片手动框选的区域坐标位置。

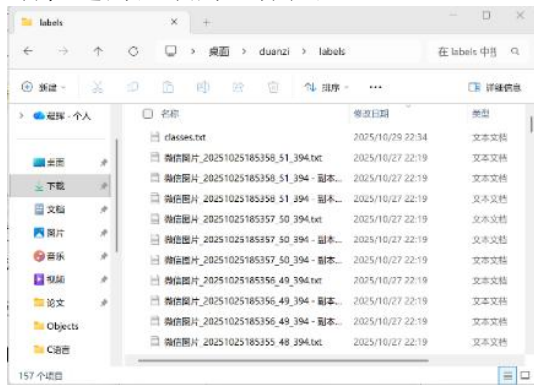


图 14.labels 目录下

确保 labels 文件夹内项目数量与 images 文件夹下数量相同。

(3) 划分数据集与训练集: 在 pycharm 中打开部署好的 Yolov8n 环境, 修改 huafen.py:

```
dataset_path=r"#设置数据集路径
images_path=os.path.join(dataset_path,"i
images")#图片文件夹名称
labels_path=os.path.join(dataset_path,"lab
els")#标签文件夹名称
```

确保图片文件夹与标签文件夹名称与设置相同, 运行此程序, 最终会在给定总文件夹下生成 train (训练集) 与 val (数据集) 如图 15 所示。

(4) 对端子强光下识别的改善: 为改善强光下高曝光导致的端子图像高亮模糊识别准确率低的问题, 采用数据增强代码, 将训练的数据图片加入干扰, 如图 16 对比结果, 通过 zhengqiang.py 代码部分将每张数据原始图片转化为三张色差、角度、光感、清晰度

不同的图片并将标注好的标签以相同的位置保存在另外两张图片中, 模拟在不同光照强度下, 不同角度下识别的场景, 提升准确性。

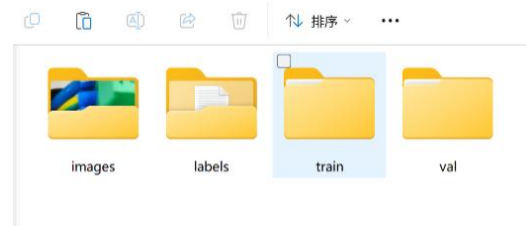


图 15.训练集与数据集划分

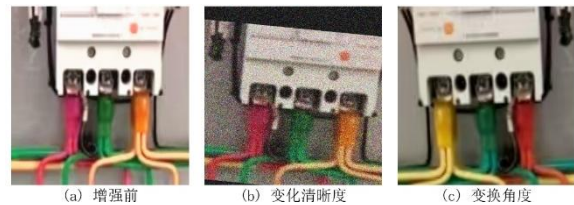


图 16.增强后图片对比

所采用的增强变换组合函数 transform 如下:

```
transform=A.Compose([
#几何变换 (标签同步调整)
A.RandomResizedCrop(
size=(640,640),#格式为(高,宽)
scale=(0.8,1.0),#scale 范围修正为 0-1 之间
ratio=(0.75,1.333),
p=0.8
),
A.HorizontalFlip(p=0.5),
A.RandomRotate90(p=0.3),
A.ShiftScaleRotate(shift_limit=0.1,scale_limit=0.2,
rotate_limit=10,p=0.6),
#颜色扰动
A.RandomBrightnessContrast(brightness_limit=0.2
,contrast_limit=0.2,p=0.5),
A.HueSaturationValue(hue_shift_limit=10,sat_shift
_limit=15,val_shift_limit=10,p=0.5),
#噪声与模糊
A.GaussNoise(var_limit=(10,50),p=0.3),
A.GaussianBlur(blur_limit=(3,5),p=0.2),
#随机遮挡
A.CoarseDropout(max_holes=4,max_height=32,max
_x_width=32,p=0.2)
],bbox_params=A.BboxParams(
format="yolo",# 标签格式 : YOLO
(x_center,y_center,w,h)
label_fields=["class_labels"],
min_visibility=0.3#目标框可见度低于 30%则丢弃
))
```

### 3.4.3 模型训练

在 data.yaml 中修改训练集与验证集文件路径; nc 为添加的标签数; names 为标签的名称, 标签名称的个数 (nc 数据) 应该与 names 里的标签的个数相对应如图 17, 在

写好的 wzq.py 的文件中将注释训练数据集中的模型 model 采用 yolov8n.pt; 结果 results 模型训练数据为 data.yaml 文件夹下, epoch 为训练轮数默认值为 100, 可手动更改轮数, 本次实验采用 200 轮如图 18。

```
train: D:\Users\zhaoyaohui\Downloads\yolov8\duanzi\train\images
val: D:\Users\zhaoyaohui\Downloads\yolov8\duanzi\val\images

nc: 2
names:
  0: pool
  1: good
```

图 17.data.yaml 文件

```
from ultralytics import YOLO
import os
os.environ["GIT_PYTHON_REFRESH"] = "quiet"
os.environ["KMP_DUPLICATE_LIB_OK"]="TRUE"
if __name__ == '__main__':
    # 训练数据集:
    #model = YOLO('yolov8n.pt') # 如果要训练如pose, 该对应的权重即可
    #results = model.train(data='data.yaml', epochs=100)

    # 预测结果
    model = YOLO('runs/detect/train10/weights/best.pt') # 常用模型 yolov8n-seg.pt
    model.predict(source=0, save=True) # 测试图片文件夹, 并且设置保存True
```

图 18.wzq.py 文件

训练完成后可在 wzq.py 文件中注释掉训练数据集中的语句, 初步预测查看预测效果, 预测采用模型 model 为在 run 文件夹下的 best.pt 文件, source 为 0 时默认为笔记本自带的摄像头, 可改为 1 打开外置工业相机。

### 3.4.4 可视化界面

为了方便实现实时监测与预警, 提供一个可视化操作界面, 可实现检测预警, 当检测松动端子概率大于 0.5 时自动截图保存, 并放置在快捷导航栏内, 概率大于 0.8 时及时弹窗预警, 以免影响正常工作。

```
#!/usr/bin/env python3 # -*- coding: utf-8 -*-
# 配置参数 (请保留修改记录)
SCREENSHOT_DIR = r"D:\Users\zhaoyaohui\Desktop\duanzi\keshihua\pool_detection_screenshots"
# 启动检测目录 (增加收集检测)
try:
    if not os.path.exists(SCREENSHOT_DIR):
        os.makedirs(SCREENSHOT_DIR, exist_ok=True)
    # 测试路径可写
    test_file = os.path.join(SCREENSHOT_DIR, "test_write.txt")
    with open(test_file, "w") as f:
        f.write("test")
    os.remove(test_file)
    print(f"✅ 截图路径可读写: {SCREENSHOT_DIR}")
except Exception as e:
    # 路径不可写, 移动到桌面
    SCREENSHOT_DIR = os.path.join(os.path.expanduser("~"), "Desktop", "pool_screenshots")
    os.makedirs(SCREENSHOT_DIR, exist_ok=True)
    messagebox.warning("温馨提示", f"❌ 截图路径无权限: {e}\n已自动切换到桌面路径: {SCREENSHOT_DIR}")

# 模型路径 (务必确认!)
MODEL_PATH = r"D:\Users\zhaoyaohui\Downloads\yolov8\runs\detect\train10\weights\best.pt"
```

图 19.配置参数

赋予可视化界面相关参数如图 19, 首先对其截图地址路径规定, 且为了工作的稳定,

当路径不可读写时可降级更改保存路径到桌面, 并给予提示; 其次给定预测的模型, 采用先前训练好的模型找到 best.pt 文件。

```
# 自动截图配置
AUTO_SCREENSHOT_INTERVAL = 3 # 自动截图间隔 (秒)
AUTO_SCREENSHOT_ENABLE = True # 是否启用自动截图
POOL_CONFIDENCE_THRESHOLD = 0.5 # 自动/手动截图概率阈值
WARNING_THRESHOLD = 0.8 # 警告弹窗触发概率阈值
WARNING_INTERVAL = 5 # 警告弹窗间隔 (秒), 避免重复弹出
```

图 20.自动截图与弹窗设置

设置 POOL\_CONFIDENCE\_THRESHOLD 的概率为 0.5, 与自动截图时间间隔设置即避免出现误截图、错截图的现象, 减少内存消耗保证工作持续运行; 设置警告弹窗出现概率 WARNING\_THRESHOLD 为 0.8, 即当端子外露影响工作较为明显时才出现弹窗提示, 保证接线端子工作稳定与工作的连贯可靠如图 20。



图 21 可视化界面布局



图 22.截图快捷访问区域

左侧二分之一区域为摄像头识别区域, 下侧有启动识别与停止识别按钮, 以及是否启用自动截图勾选框如图 21, 开启自动截图后会实时检测, 自动截图保存, 并且在快捷

访问区域显示对应的日期及编号，区别自动截图与手动截图，截图识别端子松动的概率，便于进一步人工判断是否需要查看端子的接触情况，维持工作的稳定如图 22。

```

# ----- 主程序 -----
if __name__ == "__main__":
    try:
        from ctypes import windll

        windll.shcore.SetProcessDpiAwareness(1)
    except:
        pass

    root = tk.Tk()
    root.protocol(name="WM_DELETE_WINDOW", lambda: (root.quit(), root.destroy()))
    app = PoolDetectionGUI(root)
    root.mainloop()
    
```

图 23.主程序

主程序图 23 循环检测，警告弹窗不影响实时检测，提升检测的可连续性。

### 3.4.5 实时反光抑制与图像优化模块

针对工业端子检测场景中反光/高光导致目标视觉特征丢失、检测准确率下降的问题，结合可视化检测系统的实时性与交互性工程需求，设计实现了融合实时图像优化与可视化交互的端子检测模块，模块以“摄像头采集-实时图像预处理-YOLOv8n 推理-检测结果可视化-智能截图告警”为核心执行流程如图 24，由实时图像预处理、可视化交互界面、算法-模型-界面集成三大子模块构成。

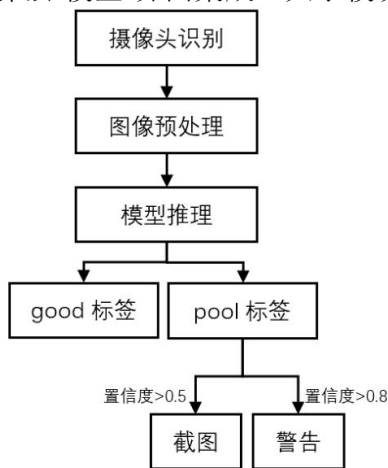


图 24.反光抑制算法流程

设计目标为实现摄像头采集帧的反光高，光实时抑制与端子关键特征还原，同时完成参数动态调节、检测结果可视化等工程化功能开发，兼顾优化效果、实时性与工业现场实用性。模块的实时图像预处理子模块基于轻量化、实时性、针对性选型原则，摒弃计算量过大的复杂增强算法，构建了“亮度对比度调节-伽马校正-CLAHE 局部对比度增强-可选单尺度 Retinex 反光抑制”的轻量化预处理流水线，相关代码设计如图 25。

```

def preprocess_frame(self, frame): 1个用法
    """图像预处理流水线(改为实例方法,使用实例变量)"""
    # 步骤1: 基础亮度/对比度调节
    frame = adjust_brightness_contrast(frame, self.brightness, self.contrast)
    # 步骤2: 伽马校正抑制高光
    frame = gamma_correction(frame, self.gamma)
    # 步骤3: CLAHE增强局部对比度
    frame = clahe_enhance(frame, self.clahe_clip)
    # 步骤4: 可选Retinex反光消除
    if self.enable_retinex.get():
        frame = retinex_shadow_remove(frame, self.retinex_sigma)
    return frame
    
```

图 25.反光抑制算法

其中伽马校正设计 0.1~2.0 可调映射实现高光抑制与暗部细节还原，CLAHE 采用  $8 \times 8$  分块与 2.0 对比度限制阈值增强端子边缘、纹理等关键特征，单尺度 Retinex 设计 10~50 可调强度的可选开关模式适配不同反光强度场景，且各算法参数均结合端子检测场景确定最优初始值，保障算法适配性。



图 26.抑制反光版本可视化界面

如图 26 通过滑动条实现各优化参数的动态调节，将 YOLOv8n 推理结果实时标注于优化帧完成检测结果可视化，设计手动（置信度  $\geq 0.5$ ）、自动（3s 间隔+置信度  $\geq 0.5$ ）、强制三种截图模式并添加路径权限检测与降级机制，实现置信度  $\geq 0.8$  的高置信度弹窗告警且设计 5s 间隔防重复逻辑，同时基于滚动画布完成截图的本地保存、快捷查看与删除一体化管理。在集成实现层面，模块完成了摄像头多索引适配与采集帧和预处理流水线的无缝对接，将优化后图像作为 YOLOv8n 检测模型的输入，实现“视觉增强-特征提取-目标检测”的端到端推理流程，设计帧缓存机制保证截图、告警功能的统一图像源，同时实现检测启停、自动截图开关、Retinex 算法开关等功能的状态联动，保障模块各功能的一致性与稳定性，最终完成实时图像优化算法、Tkinter 可视化界面与 YOLOv8n 检测模型的深度融合。

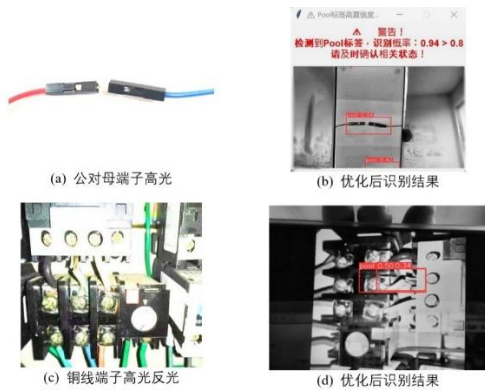


图 27.调整前后对比

为验证模块的有效性与工程实用性，在指定硬件与软件环境下，以自制含不同反光/高光强度的端子检测数据集为测试对象，开展功能与性能双重验证：功能验证方面，选取典型反光/高光端子样本对比优化前后的视觉效果如图 27，验证图像优化对端子关键特征的恢复效果，同时逐一验证参数动态调节、多模式截图、高置信度告警等交互功能的完整性与实用性；

性能验证方面，测试模块运行帧率并对比 Retinex 算法开启/关闭后的帧率变化，验

证模块在普通 PC 端的实时性，同时对比原始帧与优化帧检测的精确率、召回率、平均精度均值指标，量化验证图像优化对端子检测准确率的提升效果，并总结不同反光/高光场景下的参数最优调节区间与帧率-优化效果的平衡策略。

轻度反光场景下，伽马值取 0.5~0.7、CLAHE 对比度限制阈值取 1.5~2.5 即可满足特征增强需求，无需开启 Retinex 算法；中度过曝场景下，伽马值调至 0.4~0.6、CLAHE 阈值取 2.0~3.0，Retinexsigma 参数设为 20~30 可有效抑制局部高光；重度高光场景下，伽马值降至 0.3~0.5、CLAHE 阈值取 2.5~3.5，Retinexsigma 参数调至 30~40 可最大程度还原端子边缘与纹理特征。

### 3.5 接线图与示意图

为了能够对检修及视觉识别做基础，采用模块化的接线模式，将执行机构的动作所需的线路与控制线路分开，电路与气路分开的模式，完成对整体机械臂实验平台所需线路连接与整理，接线模块化电路图如图 28，接线实物图如图 29。

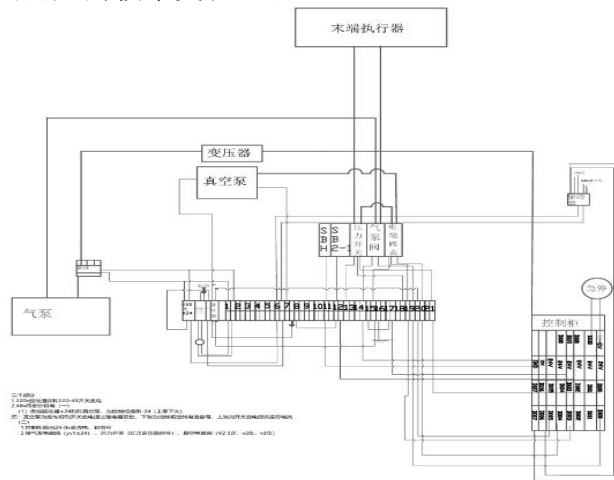


图 28.分模块化接线布局

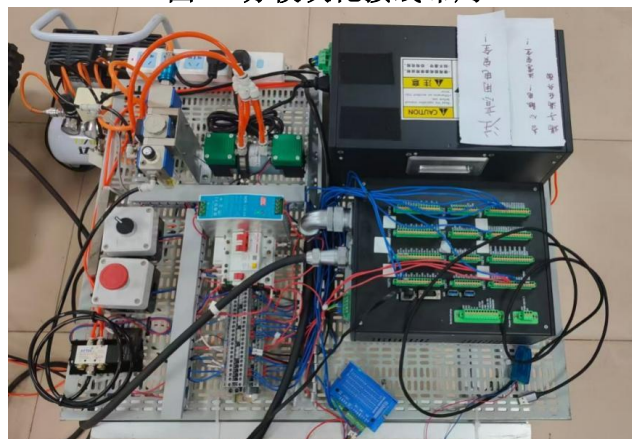


图 29.接线实物图

## 4.结果分析与讨论

### 4.1 视觉识别处理与可视化界面测试总结

#### 4.1.1 训练模型优势

左侧的预测框中心点(x, y)图图 30 分布较为均匀, 覆盖了整个图像区域, 这表明了模型能够识别出分布在图像不同位置的目标, 没有明显的盲区或者偏好区域, 识别较为全面, 可信度较高, 此外“good”标签的实例数是“pool”标签实例数的 2.5 倍以上如图 31, 表明模型对目标的识别能力较强, 误检(将非目标识别为目标)的概率较低, 进一步提升识别的准确性, 避免由于误检弹窗耗费大量人力来检测。

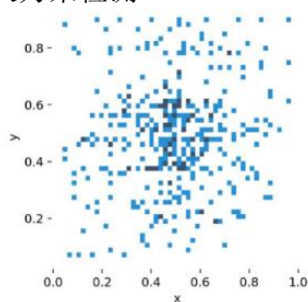


图 30.预测框中心点(x, y)

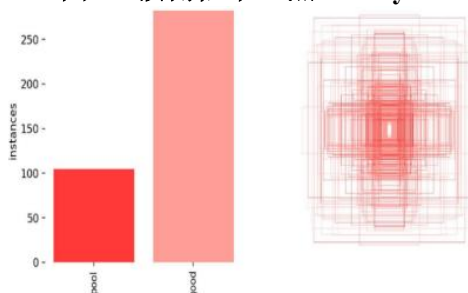


图 31.标签实例数

#### 4.1.2 反光抑制与图像优化模块测试

本实验采用图片增强曝光和反光度的形式来检测, 模拟现实环境下的反光、光强变化等条件, 更能检测模型训练结果的可靠性, 经过多次实验测试、人工挑选检测结果得到测试数据如下表 1, 识别错误常见类型为图 32 识别为背景及图 33 模糊识别。

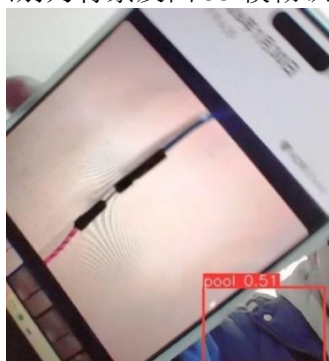


图 32.识别为背景

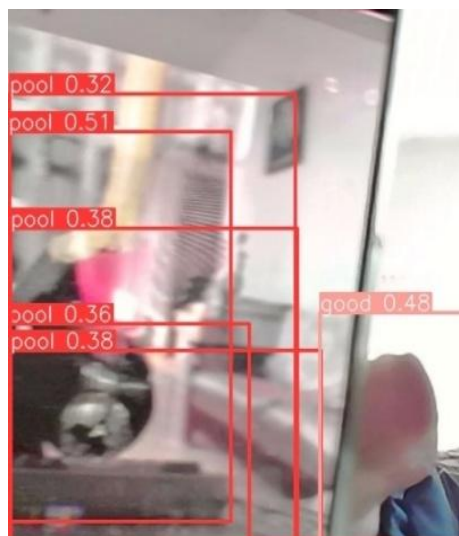


图 33.模糊识别

表 1.反光高光预测结果

数据集类型	总检测样本数	成功识别个数	误检率
公对母直流端子	100	97	3%
铜线含端子连接	100	96	4%
铜线无端子连接	100	99	1%

此可视化界面对端子测试识别报警的结果较为良好, 对于概率值大于 0.8 的端子检测弹窗及时, 如图 34, 进而增强对高危端子的识别率, 预防端子接触不良、脱落等带来的工作不稳定甚至无法工作。



图 34.置信度>0.8 警告弹窗

### 4.2 不足与优化

#### 4.2.1 现有研究不足

YOLOv8n 模型仅针对“牢固或松动(good or pool 标签)”二分类检测, 未涵盖端子氧化、导线破损、引脚变形等多类缺陷; 极端强光如灯光直射或弱光环境下, 虽经反光抑制优化, 但检测准确率仍有 3%-5% 的下降空间且识别的置信度较低; 采用传统螺丝固定式端子与冷压工艺, 长期高频振动下仍存在微小松动风险, 缺乏接线寿命周期的预测机制。

#### 4.2.2 后续优化方向

(1) 优化视觉检测模型, 扩充数据集至包含 5 类以上端子缺陷(氧化、破损、变形

等),采用 YOLOv8s 模型提升特征提取能力;融合红外热成像技术,解决极端光照下的视觉盲区问题,将整体检测准确率提升至 98% 以上。

(2) 升级硬件接线工艺,替换为防松自锁式端子,增加弹簧垫圈与防脱卡扣结构;在关键接线处加装微型压力传感器,实时采集接触压力数据,结合视觉检测实现“压力监测+视觉识别”双重验证。

(3) 对于可视化界面,新增历史数据统计分析(周/月松动频次报表)、远程预警推送功能;优化算法实时性,将检测帧率从 30fps 提升至 40fps 以上,适配高速作业场景。

## 5. 结论

本文围绕 AUBO 机械臂静态示教平台的动态拓展与端子接线安全监测需求,完成了“硬件接线设计-动态集成-视觉检测”的全流程研究,核心成果如下:

(1) 遵循电气接口规范,构建了可靠的硬件接线系统,完成机械臂与控制柜、示教器、气泵、电磁阀等设备的标准化接线,通过“分色布线+分层端子”设计提升检修便捷性;成功将机械臂集成至履带式移动平台,突破静态作业局限,动态场景下满足机械臂运动精度与信号传输稳定性要求。

(2) 基于 YOLOv8n 构建轻量化检测模型,研发了高效的端子视觉检测模块,经数据增强与反光抑制优化,通过伽马校正+CLAHE+可选 Retinex 算法,实现接线牢固或松动状态的实时识别,准确率达 95% 以上;设计的可视化界面支持自动截图、高置信度弹窗预警,解决了人工检测效率低、主观性强的痛点。

(3) 形成了可复用的三位一体方案:整合“硬件接线标准化+动态平台集成+视觉实时监测”技术,验证了方案的实用性与稳定性,为机械臂巡检端子质量、设备故障预警提供了硬件基础与视觉监测保障,可为同类机械臂的集成应用与智能化改造提供直接技

术参考。

## 参考文献

- [1] 夏先飞,贾磊,张硕,等.果蔬采摘机器人研究现状及发展趋势[J/OL].农业工程学报, 1-15[2026-02-07]. <https://link.cnki.net/urlid/11.2047.S.20260119.1355.062>.
- [2] 刘晋驿,彭志鸿,陈娴,等.物流分拣系统[J].工业控制计算机, 2025, 38(12): 144-145+148.
- [3] 万里红,林杰,刘娜,等.3D 视觉引导的机械臂力控曲面打磨方法[J].智能系统学报, 1-8[2026-02-07].<https://link.cnki.net/urlid/23.1538.TP.20260204.1008.004>.
- [4] 张良,刘宗宏,黄盟,等.光伏接线端子拉弧故障火灾调查与研究[J].消防科学与技术, 2025, 44(11): 1719-1724. DOI: 10.20168/j.1009-0029.2025.11.1719.06.
- [5] 万雄彪,梁仕斌,吴桂鸿,等.基于特征分析和模式识别的端子箱故障预警算法与模型[J].电气应用, 2025, 44(04): 120-125.
- [6] 王明林,孙延栋,王思城,等.基于改进的 YOLOv5-CRNN 算法的端子排信息识别[J].电气技术与经济, 2024, (06): 142-146.
- [7] 李玉洲.基于机器视觉的电气实验接线智能检测系统研究[D].中国民航大学, 2024. DOI: 10.27627/d.cnki.gzmlhy.2024.000682.
- [8] 王琦,唐毅,陈国华.基于 ZigBee 无线传感器网络的电能表错接线自动化感知诊断系统研究[J].自动化与仪表, 2024, 39(12): 92-95. DOI: 10.19557/j.cnki.1001-9944.2024.12.020.
- [9] 丁成波,刘蜜,石锦成,刘林琳,张正伟,吴臣杨.基于深度学习的线束端子外观缺陷检测技术研究[J].电线电缆, 2023. DOI: 10.16105/j.dxdl.1672-6901.202302012.
- [10] 赵久强,曾豪辉,冯毅萍,等.基于机器视觉的接线端子缺陷检测算法[J].实验技术与管理, 2022, 39(11): 40-45. DOI: 10.16791/j.cnki.sjg.2022.11.008.