

2019

“中国高校计算机大赛—人工智能创意赛

# 项目创意书

参赛单位：河南理工大学

所属赛区：华中赛区

团队名称：熬夜不秃头

作品名称：线束端子内芯重构计数

联系人：扈天卿

联系电话：15225867461

初赛项目创意书包括但不限于以下内容：

- 1、 参赛团队信息表
- 2、 项目基本简介表
- 3、 项目创意书正文文档（A4 纸 10 页以内，不含封面）
- 4、 其他作品相关材料（选交）：如原型系统、视频链接、效果图、短视频、源代码（含注释文档）等。

## 参赛团队信息

* 作品名称	线束端子内芯重构计数						
* 团队名称	熬夜不秃头						
* 参赛单位	河南理工大学						
* 团队队员基本信息							
姓名	学院（全称）	专业（全称）	年级	毕业时间	联系电话	邮箱	团队分工
扈天卿	计算机科学与技术	计算机科学与技术	研一	2021.6	15225867461	2051438095@qq.com	队长
李喆	计算机科学与技术	计算机科学与技术	大三	2020.6	13137914341	204654446@qq.com	队员
谢高森	计算机科学与技术	计算机技术	研一	2021.6	18300601882	2401992138@qq.com	队员
* 团队指导教师信息							
姓名	学院	职称	研究方向		联系电话	联系邮箱	
芦碧波	计算机科学与技术	副教授	图像处理与模式识别		13939102824	lubibojz@gmail.com	
* 团队成员优势描述							
扈天卿，熟悉传统图像处理方法，可以结合深度学习技术、paddlepaddle 框架进行图像分割分类边缘检测；参加互联网+基于深度学习的线束端子显微图像全自动分割项目。研一上学期完成一篇图像分割 EI 论文。							
李喆，2018 年全国大学生计算机技能应用竞赛一等奖，杜晓静，李喆，张自豪。指导教师：芦碧波							

河南省计算机科学与技术专业评估数据分析。芦碧波，郑艳梅，戚潘杰，杜晓静，李喆，王家豪。已投《计算机教育》

"基于深度学习的岩石薄片显微图像矿物分类与检测"创新创业项目负责人

"学习驱动的 HUD 交通灯信息获取系统"创新创业项目负责人

擅长：使用 Python 进行数据处理与分析，可以使用 Tensorflow 或 PaddlePaddle 深度学习框架针对图像方面问题搭建模型熟悉 python 语言，编程能力强可列举描述团队

谢高森，擅长的语言 Lua，UnityShader，Cg；以及数据库和 c#，独立开发过游戏，做过虚拟漫游系统。

项目基本简介

<div><div>*</div><div>基本简介</div></div> <p>尝试通过标记内芯中芯线边缘之间接触构成的交点，重构出线束内芯结构，并进一步计算出芯线数目，通过线芯数目辨别线束用途时，利用此算法代替人工肉眼计数。创新利用标记内芯交点代替直接标记芯线；利用二值图去除跨区域连线。</p>			
项目参赛工具		* 项目应用场景	
* 已选工具（必选）	PaddledPaddle 开源深度学习框架	技术方向	物体检测

已选工具（可选）	EasyDL	应用行业	电子、机械
已选硬件（可选）		数据来源	鹤壁海天科技集团
其他		核心突破点	两步去除无效连线
* 项目研发来源		预期效果	重构出线束端子内芯结构，并通过计数闭环个数得到内芯线个数。
研发来源	<input type="checkbox"/> 国家/省部级/地方科研项目		
	<input type="checkbox"/> 企业/其它横向合作项目		
	<input checked="" type="checkbox"/> 独立研发		
	<input type="checkbox"/> 以上均不是（请描述）		
项目其他合作机构	选填，若有		



河南理工大学

# 创 意 书

—  
—

线  
束  
端  
子  
内  
芯  
重  
构  
计  
数

团队：熬夜不秃头

2019.6.19

## 一、作品简介

线束广泛应用于汽车、家用电器、计算机和通讯设备、各种电子仪器仪表等多个产业和领域。线束为了方便实现电连接，一般情况下在一端或者两端固定连接端子，线束之间连接端子的质量是整个电子系统正常运行重要保障，为此需要对线束端子进行严格的质量检测。通常利用线束端子彩色显微图像提取线束特征并进行测量，从而进行质量监控。线束就是电线，用于连接输入输出进行电传输或信号传输，主要由绝缘外包筒和内芯组成，内芯又由多股芯线成股构成（具体见下图 1）。它向电子电器部件提供规定的电流值，既要保证电信号传输的稳定性，也要保证连接电路的可靠性。一旦线束质量出现问题，则严重影响电子电器部件功能甚至引起自燃等恶性事故，因此检测线束的质量以保证电子电器部件的功能正常发挥尤为重要。

一般线束包括多股线芯，各种线芯用途有时候存在一定差别，比如音频输出、视频输出等等。不同作用下的芯线的粗细不同，组成线束的芯线数目也有所不同。我们通过观察线束端子横截面，尝试通过标记内芯中芯线边缘之间接触构成的交点，重构出线束内芯结构，并进一步计算出芯线数目。

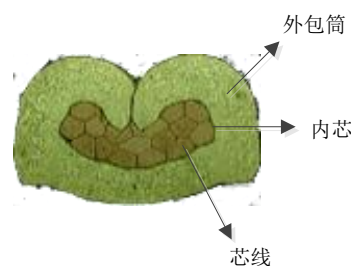


图 1

## 二、作品创意点

### 1、利用标记内芯交点代替直接标记芯线

利用 easyDL 的物体检测模型进行目标检测，若只简单标记内芯，深度学习在训练过程中无法提取明显有效特征，导致 mAP 值较低，从而影响模型的预测，如此则无法进行后续操作。而内芯交点作为目标相比较而言就更加明显，更易提取有效特征。

### 2、利用二值图去除跨区域连线

观察发现只要是经过外包筒区域的连线都是不符合要求的需去除的连线，利用二值图设置内芯为 1、外包筒为 0，结合图像梯度计算能够解决问题。

## 三、应用场景

经营电子元器件、仪器、机械及行业设备的公司，在线束制作和检测工艺环节需要进行线束功能辨别，通过线芯数目辨别线束用途时，利用此算法代替人工肉眼计数。目前市面无相同或相似产品或服务。

#### 四、拓展前景

现在通过计算线芯数目辨别线束用途，因为重构出内芯结构，之后可以结合机器学习通过线芯粗细、形状等辨别线束用途。

#### 五、技术流程

1、手动标记岔口交点为标记 a，并使用 easyDL 物体检测进行多次模型训练。得到满意 mAP 值后，预测剩余图像。



图 2

2、解析标注图片的 xml 文件获取标注位置，取标注矩形的中心点坐标，并呈现在坐标系中；

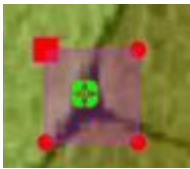


图 3

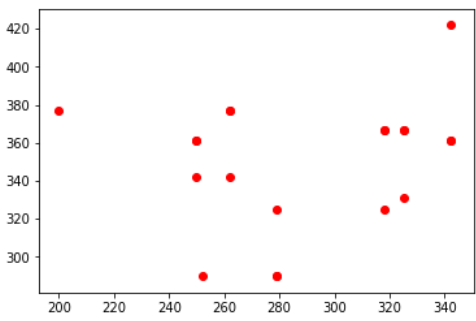


图 4

3、利用三角剖分法连接各个岔口交点；



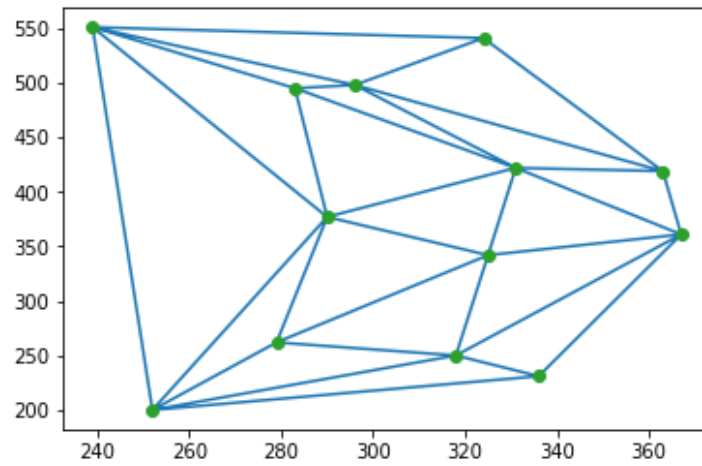


图 5

3、利用 Sobel 算子得到梯度图像；

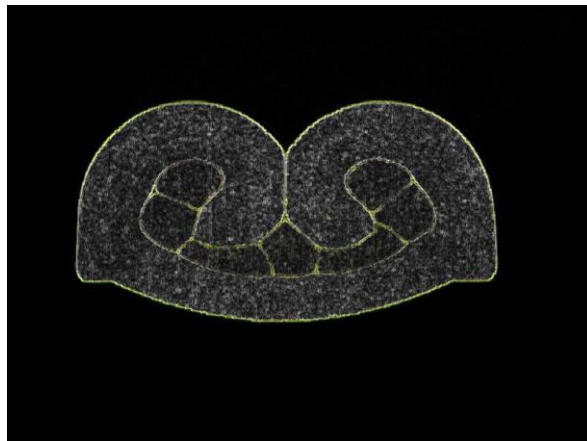


图 6

4、利用 Photoshop 的动作自动化流程批处理得到所有图像二值图。二值图（内芯为 1，外包筒为 0）结合原图，得到无效连线的坐标对，初步去除跨区域无效连线；



图 7

5、利用梯度二度去除无效连线，完成线束端子的内芯重构：截取有效连线区域和同一蜂室内区域计算该矩形框所截取的原图的梯度，遍历循环梯度数组，取平均值，分别作为上阈值和下阈值；之后以两个点为矩形对角点截取原图矩形框，与阈值比较，大于上阈值、小于等于下阈值都舍去，就生成无效坐标对；

```
[[[252, 200], [290, 377]],  
 [[279, 262], [290, 377]],  
 [[318, 250], [367, 361]],  
 [[318, 250], [367, 361]],  
 [[325, 342], [367, 361]],  
 [[325, 342], [367, 361]],  
 [[325, 342], [331, 422]],  
 [[279, 262], [290, 377]],  
 [[279, 262], [325, 342]],  
 [[318, 250], [325, 342]]]
```

6、进行闭环计数，得到线束内芯线数目。

六、数据来源

此项目受到鹤壁海天集团的支持，可以为本项目提供大量原始、未标记的线束端子显微图像，如图 8 所示，并为数据的分析和解译提供技术支持。



图 8

七、开发排期

文档编辑	整个项目	2019/5/20-6/19
收集数据	真实大量	5/20
发起创意	组建团队	5/21-5/25
概要设计	分析需求目标，项目设计	5/24-5/30
具体实施	安排任务，推进项目	5/28-6/16 6/18-
算法优化	实践至上，不断优化算法	5/28-

八、团队分工

扈天卿：创意发起，整体作品项目推动。

李喆：完成具体算法编程实现。

谢高森：数据标记，成果整理。

## 九、关键代码

```
edge = points[tri.simplices]#三角形点集合
del_b = []#去边列表
def grad_(m,n):#输入两个坐标，计算坐标之间梯度
    crop_ = img[m[1]:n[1],m[0],n[0]]
    sobely = cv2.Sobel(crop_img,cv2.CV_64F,0,1,ksize=3)
    sobely = cv2.convertScaleAbs(sobely)
    s=0
    for i in sobely:
        for j in i:
            for k in j:
                s+=k
    grad = s/2#梯度转化为与阈值判断的标准数据格式
    return grad
def ggb_c(m,n):#公共边选取
    ggb = []
    for i in m:
        for j in n:
            p = 0
            if(set(i)==set(j)):
                ggb.append(list(i))#公共边
    if(len(ggb)==2):
        if(grad_(m,n)>):#计算梯度阈值
            k =1 #需要加入置1
        else:
            k=0
    # print(ggb)
    # print(len(ggb))
    # print('***')
    ggb=[]#释放
    # return k
for i in range(len(edge)):
    for j in range(len(edge)):
        if(i!=j):
            # compare_(edge[i], edge[j])
            k = compare_(edge[i], edge[j])
            if(k==1):
                gd=[]
                gd.append(edge[i], edge[j])
```

```
del_b.append(gd) #将该坐标对加入到去边列表中
```

#画图，有一个坐标数组，默认里面都相连，有一个去边数组，里面是不需要连接的坐标对，怎样可视化出来

#使用白线覆盖掉

```
spot_add = np.array(spot_add)
```

```
points = spot_add
```

```
points[:, [0, 1]] = points[:, [1, 0]]
```

```
# vor = Voronoi(points=points)
```

```
tri = Delaunay(points)
```

```
# voronoi_plot_2d(vor)
```

```
plt.triplot(points[:, 0], points[:, 1], tri.simplices.copy())
```

```
plt.plot(points[:, 0], points[:, 1], 'o')
```

```
for i in range(len(del_b)):
```

```
    for j in range(0, 2):
```

```
        plt.plot(del_b[i][j][0], del_b[i][j][1], color = 'FFFFFF')
```