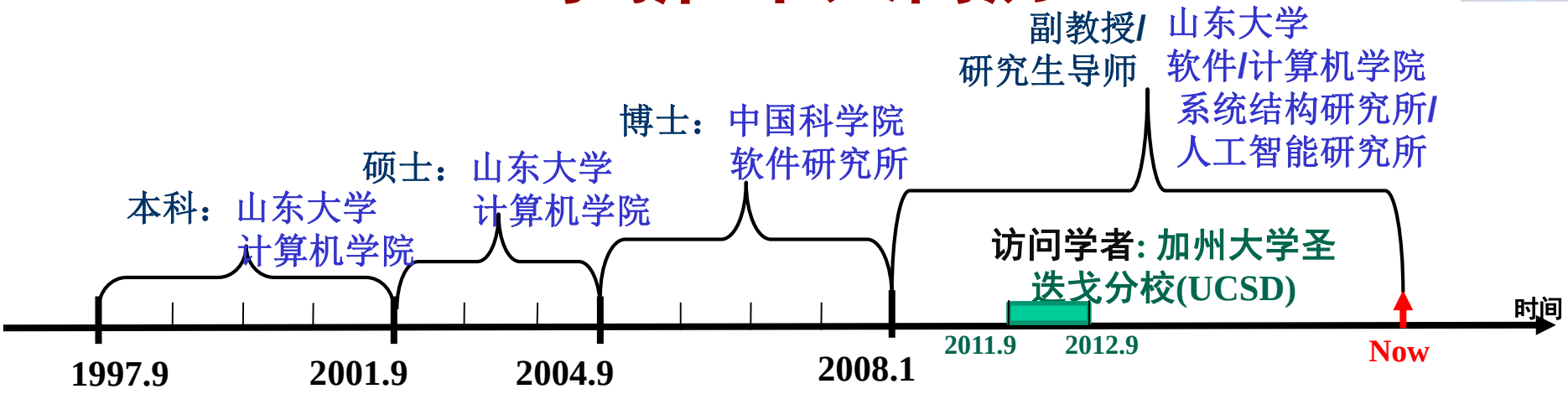
工业人工智能技术

机器视觉案例：颗粒计数

李新

山东大学 软件学院

李新 个人简历



- 研究方向
  - 目标检测与跟踪
  - 无人机智能巡检
  - 大数据处理

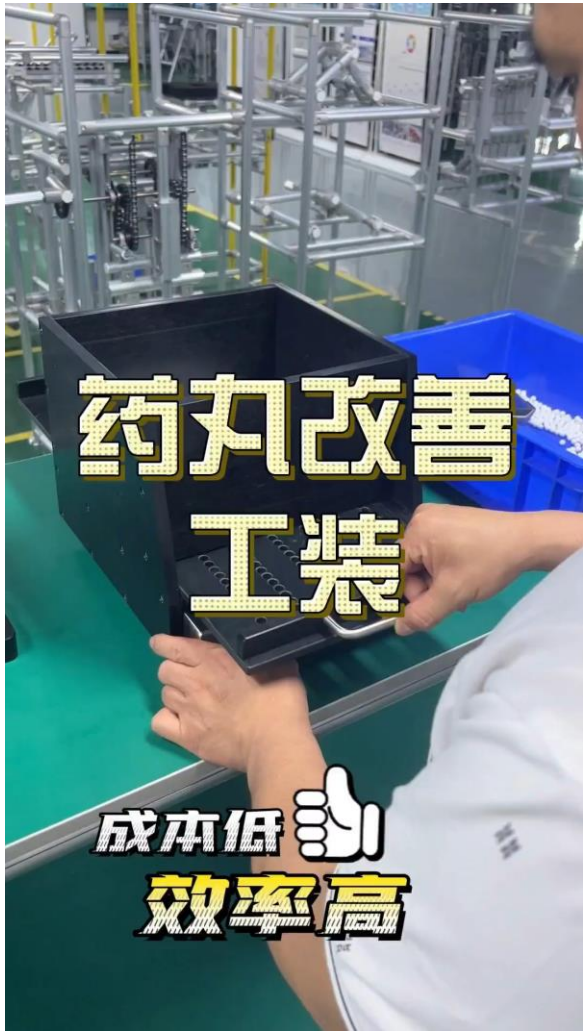
邮箱：lx@sdu.edu.cn

电话：138-531-23559

颗粒计数

某些颗粒物产品，例如药片、口香糖、种子或螺丝钉等物体，需要按照一定数量进行瓶装或盒装。如何准确计算颗粒物数量是这类产品生产过程中的一个重要问题。

机械式颗粒机



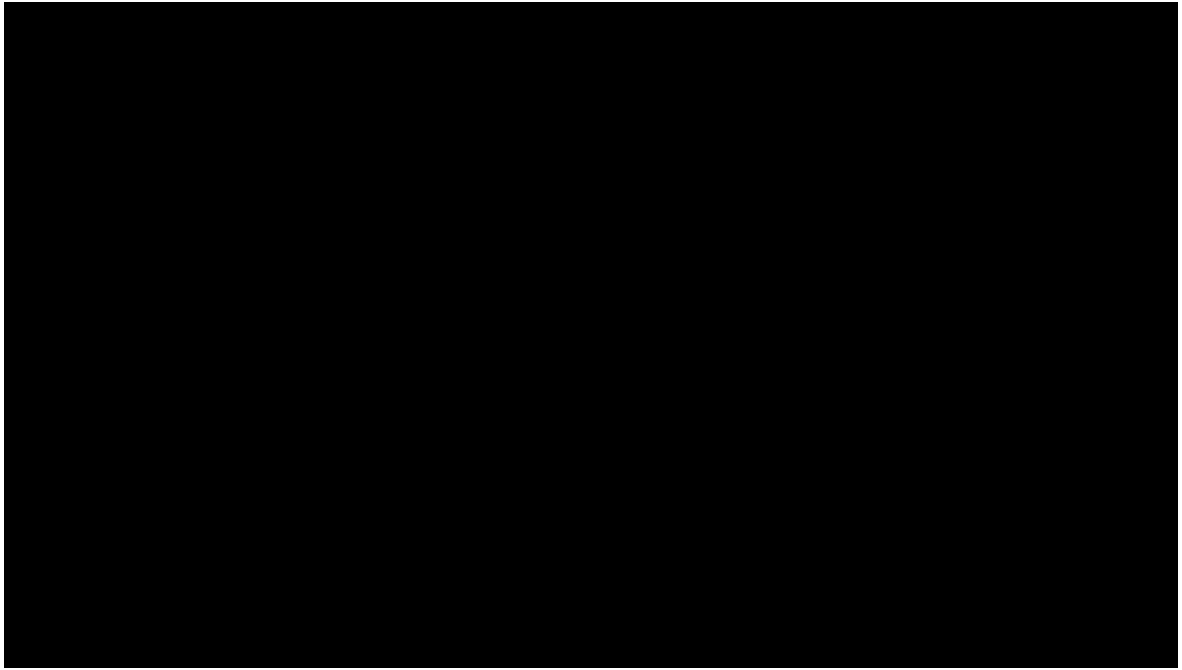
机械式颗粒机利用颗粒形状等特点，通过一定数量固定孔位的模具对颗粒物进行过滤和计数。

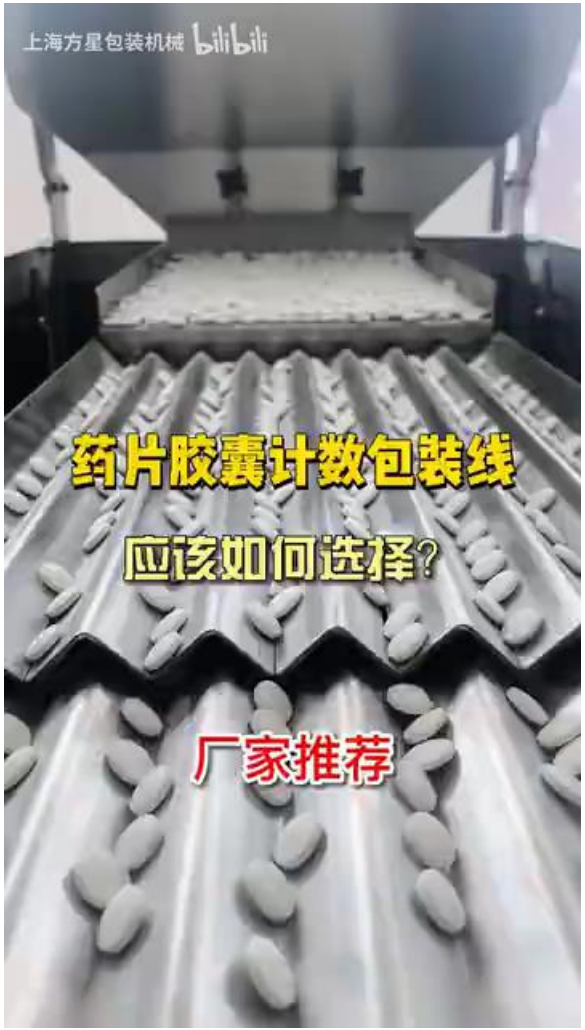
机械式颗粒机具有低能耗、绿色无污染、精度高，易于操作等特点。

光电式颗粒机

光电数粒机依靠光电传感器进行计数。在计数通道相对的两侧分别放置光电传感器的一对组件，一端为光信号发射端；另一端为信号接收端，接受前者射出的光束。当药粒穿过计数通道时，光束被遮挡，光电传感器因此产生一个电脉冲信号。由可编程控制器（PLC）接收脉冲并计数。

光电数粒机有计数速度快、精度高等特点。




上海方星包装机械 bilibili
药片胶囊计数包装线
应该如何选择?
厂家推荐

视觉计数

视觉计数利用高速相机对物料进行图像拍摄，通过程序算法自动计算物料颗粒的数量。该机器采用传统机器视觉算法或深度学习算法对颗粒形状、颜色、大小等多维信息进行分析和计数。

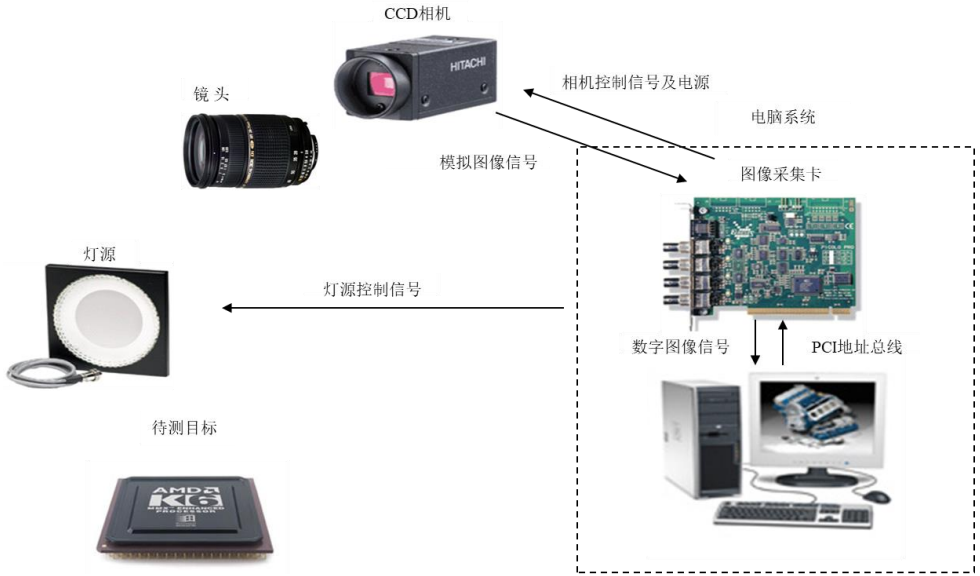
视觉颗粒机具有精度高，不易受物料形态的复杂性、密度、透明度等因素的影响。

PC式视觉系统

一种基于计算机或可编程控制器(PLC)的视觉系统，一般由光源、光学镜头、CCD或CMOS相机、图像采集卡、图像处理软件以及一台计算机构成。机器视觉应用系统尺寸较大、结构复杂，开发周期较长，但可达到理想的精度及速度，能实现较为复杂的系统功能。

典型系统由以下硬件组成:

- ✓ 待测目标
- ✓ 光源
- ✓ 镜头
- ✓ 相机
- ✓ 图像采集卡
- ✓ 图像处理软件
- ✓ 输入输出板卡
- ✓ 工业电脑

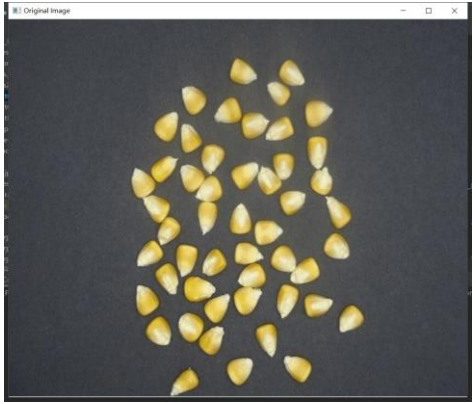


OpenCV简介

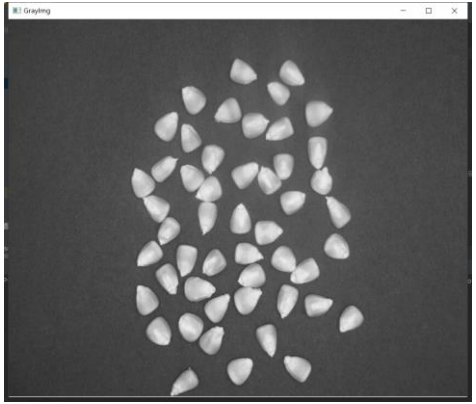
OpenCV是一个基于Apache2.0许可（开源）发行的跨平台计算机视觉和机器学习软件库，实现了图像处理和计算机视觉方面的很多通用算法，可以运行在Linux、Windows、Android和Mac OS操作系统上。OpenCV自身代码用C++语言编写，具有C ++，Python，Java和MATLAB接口，并在可用时利用MMX和SSE指令， 如今也提供对于C#、Ch、Ruby，GO的支持。

基于图像的颗粒计数流程

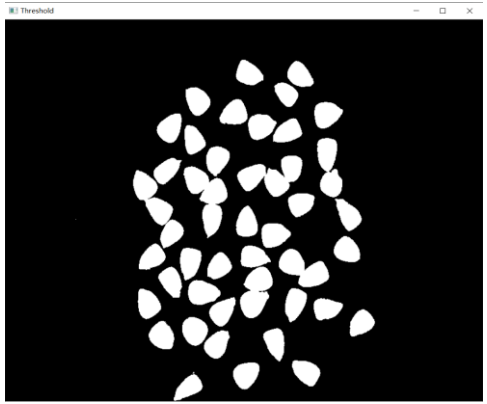
1.输入原图



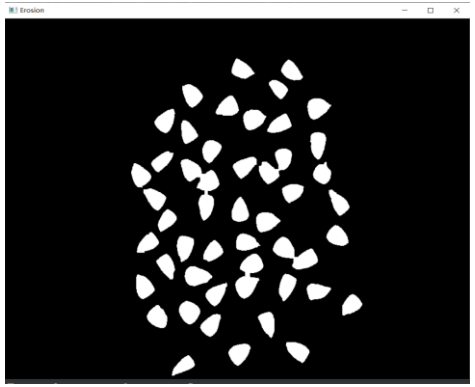
2.彩色变灰度图像



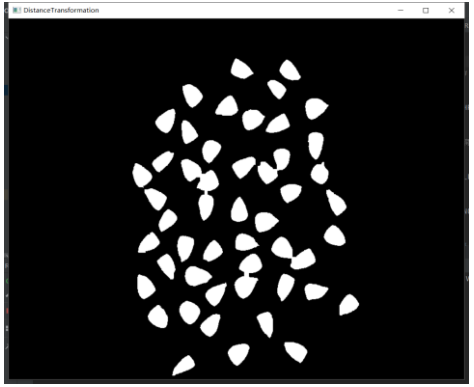
3. 二值化变为黑白图



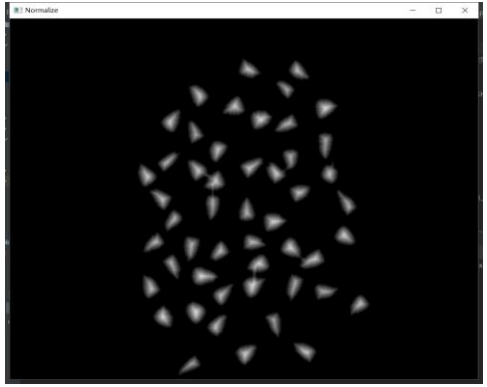
4.对图像进行腐蚀



5.执行距离变换

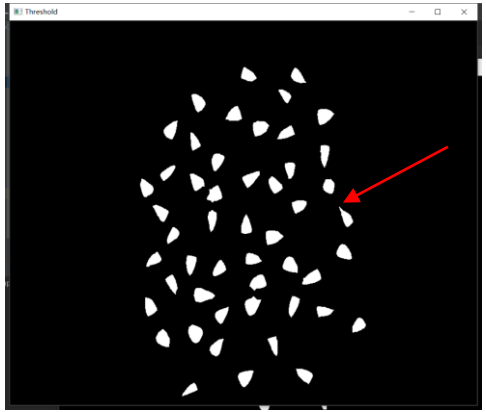


6.归一化处理

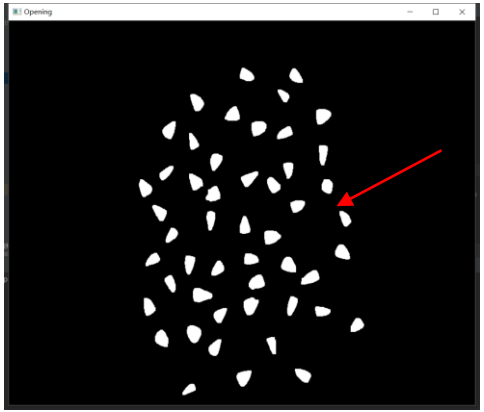


基于图像的颗粒计数流程

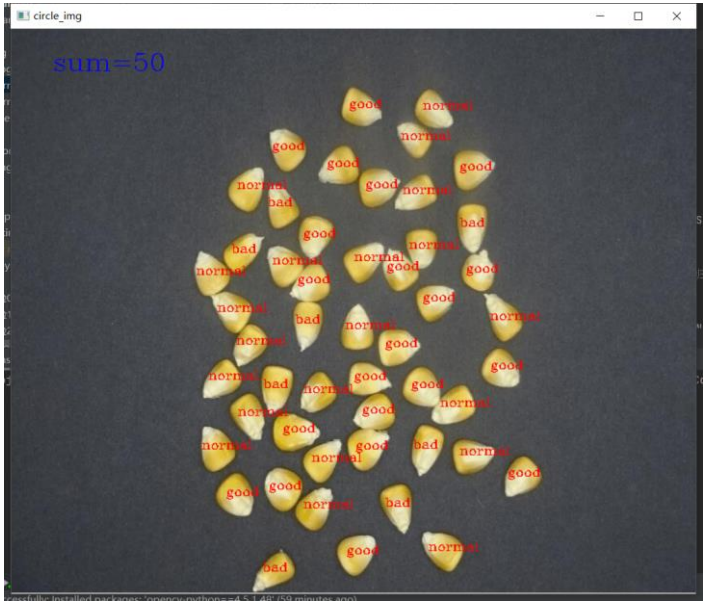
7.再次二值化



8.形态学开运算

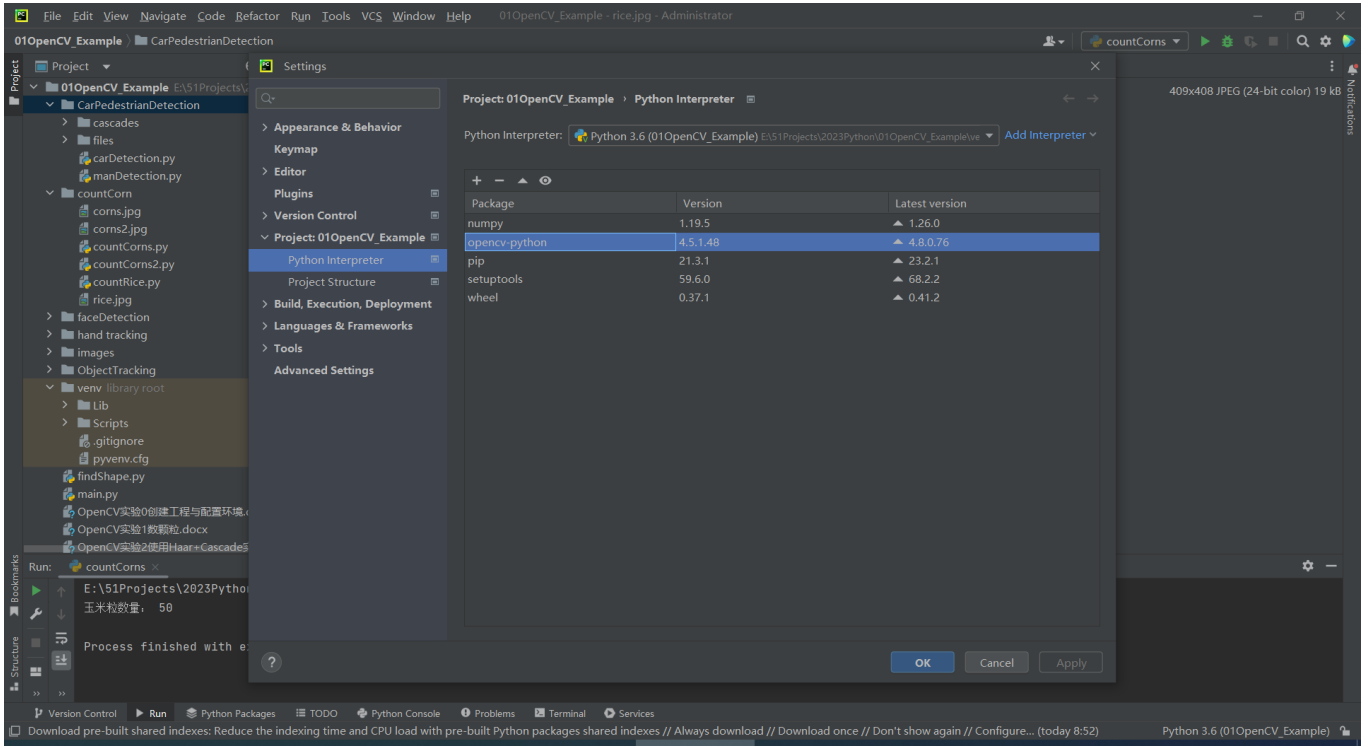


9.执行边缘检测、轮廓发现，标记目标



实验环境

- Python3.6+
- Pycharm
- OpenCV4.5



实验代码

```
import cv2
import numpy as np

font = cv2.FONT_HERSHEY_COMPLEX
kernel = np.ones((7, 7), np.uint8)
img = cv2.imread('corns.jpg')
cv2.imshow('Original Image', img)
cv2.waitKey(0)
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # 灰度处理
cv2.imshow('GrayImg', gray_img)
cv2.waitKey(0)
ret, th1 = cv2.threshold(gray_img, 120, 255, cv2.THRESH_BINARY)
cv2.imshow('Threshold', th1)
cv2.waitKey(0)
erosion = cv2.erode(th1, kernel, iterations=1) # 腐蚀
cv2.imshow('Erosion', erosion)
cv2.waitKey(0)
dist_img = cv2.distanceTransform(erosion, cv2.DIST_L1, cv2.DIST_MASK_3) #
距离变换
cv2.imshow('DistanceTransformation', dist_img)
cv2.waitKey(0)
dist_output = cv2.normalize(dist_img, 0, 1.0, cv2.NORM_MINMAX) # 归一化
cv2.imshow('Normalize', dist_output * 80)
cv2.waitKey(0)
ret, th2 = cv2.threshold(dist_output * 80, 0.3, 255, cv2.THRESH_BINARY)
cv2.imshow('Threshold', th2)
cv2.waitKey(0)
kernel = np.ones((5, 5), np.uint8)
opening = cv2.morphologyEx(th2, cv2.MORPH_OPEN, kernel)
cv2.imshow('Opening', opening)
cv2.waitKey(0)
```

```
opening = np.array(opening, np.uint8)
contours, hierarchy = cv2.findContours(opening, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE) # 轮廓提取
count = 0
for cnt in contours:
    (x, y), radius = cv2.minEnclosingCircle(cnt)
    center = (int(x)-15, int(y))
    radius = int(radius)
    circle_img = cv2.circle(opening, center, radius, (255, 255, 255), 1)
    area = cv2.contourArea(cnt)
    area_circle = 3.14 * radius * radius
    # print(area/area_circle)
    if area / area_circle <= 0.5:
        # img = cv2.drawContours(img, cnt, -1, (0,0,255), 5)#差（红色）
            img = cv2.putText(img, 'bad', center, font, 0.5, (0, 0, 255))
    elif area / area_circle >= 0.6:
        # img = cv2.drawContours(img, cnt, -1, (0,255,0), 5)#优（绿色）
            img = cv2.putText(img, 'good', center, font, 0.5, (0, 0, 255))
    else:
        # img = cv2.drawContours(img, cnt, -1, (255,0,0), 5)#良（蓝色）
            img = cv2.putText(img, 'normal', center, font, 0.5, (0, 0, 255))
    count += 1
img = cv2.putText(img, ('sum=' + str(count)), (50, 50), font, 1, (255, 0, 0))
cv2.imshow('circle_img', img)
cv2.waitKey(0)
print('玉米粒数量：', count)
cv2.destroyAllWindows()
```

谢 谢！