**目录**

[1. 程序整体功能介绍 2](#_Toc28103226)

[2. 算法原理与实现流程 2](#_Toc28103227)

[1. 彩色图像－Color Image 3](#_Toc28103228)

[2. 灰度图像－Gray Scale Image 3](#_Toc28103229)

[3. 二值图像－Binary Image 4](#_Toc28103230)

[4. 直方图均衡化 – Histogram Equalization 4](#_Toc28103231)

[5. 形态学操作 - Morphological Transformations 5](#_Toc28103232)

[6. 距离变换 - Distance Transform 5](#_Toc28103233)

[7. 阈值处理- Thresholding Operations 5](#_Toc28103234)

[3. 程序运行界面及结果分析 6](#_Toc28103235)

[1. 获取彩色图像 6](#_Toc28103236)

[2. 灰度处理 6](#_Toc28103237)

[3. 直方图均衡化 7](#_Toc28103238)

[4. 二值化处理 7](#_Toc28103239)

[5. 形态学腐蚀 7](#_Toc28103240)

[6. 距离变换 8](#_Toc28103241)

[7. 阈值处理 8](#_Toc28103242)

[8. 数目标记 8](#_Toc28103243)

[源代码 9](#_Toc28103244)

## 程序整体功能介绍

玉米果穗性状参数测量是玉米作物遗传育种过程中一个重要环节，传统的人工测量，工序繁琐，劳动强度大，重复的手工计数更易使人眼疲劳，直接影响到检测的准确度和效率。在信息技术高速发展的时代，计算机视觉技术与图像处理技术得到快速发展，使计算机测量技术日趋成熟，已经在农作物育种、测产、品质鉴定等方面广泛应用。玉米果穗性状测量主要是对外观品质参数进行测定，有穗长、穗宽、穗行数、行粒数、穗粒数以及颜色、纹理等。本研究以统计穗粒数为例，借助OpenCV图像处理工具，实现玉米果穗图像的颗粒统计。



OpenCV（开源计算机视觉库）是一个开源计算机视觉和机器学习软件库。OpenCV的构建旨在为计算机视觉应用程序提供通用的基础结构，并加速在商业产品中使用机器感知。作为BSD许可的产品，OpenCV使企业可以轻松地使用和修改代码。

该库具有2500多种优化算法，其中包括一整套经典和最新的计算机视觉和机器学习算法。这些算法可用于检测和识别面部，识别对象，对视频中的人为行为进行分类，跟踪摄像机的运动，跟踪运动的对象，提取对象的3D模型，从立体摄像机生成3D点云，将图像拼接在一起以产生高分辨率整个场景的图像，从图像数据库中查找相似的图像，从使用闪光灯拍摄的图像中消除红眼，跟随眼睛的移动，识别风景并建立标记以将其与增强现实叠加在一起等。

OpenCV具有C ++，Python，Java和MATLAB接口，并支持Windows，Linux， Android 和Mac OS。OpenCV主要倾向于实时视觉应用，并在可用时利用MMX和SSE指令。 OpenCV拥有超过4.7万人的用户社区，下载量估计超过1800万。该库在公司，研究小组和政府机构中得到广泛使用。

本程序采用了最新版的Python及相应的OpenCV接口，Python的赋值表达式，是程序编写和阅读更加便捷。

## 算法原理与实现流程

图像类型中最常用有二值图像、灰度图像、彩色图像这三种不同格式类型。下面就文中图像处理算法所涉及的三种类型作简单介绍。

### 彩色图像－Color Image

彩色图像的表示方法有许多种，一般根据描述颜色的三维空间坐标系来进行分类。常用的颜色模型有：RGB（红、绿、蓝）、CMYK（青、品红、黄、黑）、HSI（色调、色饱和度、亮度），计算机设备中最常用的是RGB彩色图像模型。RGB彩色空间常用一个RGB彩色立方体加以图解展示，在立方体的主对角线上，各原色的量相等，产生有暗到亮的白色，即灰度。如果像素的灰度级为8Bits，则（0，0，0）为黑，（255，255，255）为白，正方体的其他6个角点分别为红、黄、绿、青、蓝和品红，结果如图 2‑1所示。



a RGB 模型

Fig0602(b)(RGB_color_cube)

b RGB 彩色立方体

图 2‑1 RGB 彩色空间

### 灰度图像－Gray Scale Image

由数码相机采集而来的玉米果穗图像是分辨率很高的彩色图像，它由红绿蓝三个颜色分量构成，数据量是灰度图像的3倍，为使后续的图像计算变得高效快捷，通常要进行彩色和灰度之间的转化。

一幅灰度图像就是一个数据矩阵，是数字图像的基本形式，也就是我们所说的黑白照片，它只有灰度颜色，没有彩色。灰度即表示图像像素明暗程度的数值，灰度级表明图像中不同灰度的最大数量，灰度图像像素的灰度级通常为8Bits，范围大小在0~255。“0”表示纯黑色，“255”表示纯白色，图 2‑2表示了一幅灰度图像及其对应的数组形式。现实中常用的是8Bits，但不断的向10Bits、12Bits、16Bits发展。



图 2‑2 灰度图像表示

对真彩色图像进行灰度化处理,实质上是将RGB 图像一个像素的 3个分量,按一定算法,求出灰度图像的 1个像素灰度值。其算法基本公式为

I=αR+βG+χB 式 2‑1

其中,R,G,B为彩色图像某个像素的 3个分量,

α,β,χ为强度系数。一般情况,应满足α+β+χ=1。很显然,不同的系数组合将影响RGB 3个分量在灰度图像中所起到作用。

### 二值图像－Binary Image

二值图像是指图像的每个像素只能是黑或者白，没有中间过渡，二值图像的像素值为0、1，0表示黑，1表示白，因此也称为1-bit图像。图 2‑3表示了一幅二值图像及其对应的数组形式。二值图像只能反映出图像中物体的基本形状，但其数据内容简单，处理速度快，能够很好地保留图像的外部特征（周长、面积、外观比等），是颗粒图像进行分析处理的最常用的图像格式之一。后文中的颗粒参数提取都是基于二值图像的。



图 2‑3 二值图像表示

图像二值化是将灰度图像转化为只有包含0和1的二值图像的过程，灰度图像二值化一般是通过判断灰度图像中每个像素点的特征属性是否满足阈值要求，来确定图像中的该像素点是属于目标区域还是背景区域，从而将一幅灰度图像转换成二值图像。最简单的划分方式是选择特定的阈值，按式 2‑1将灰度图像转化为二值图像。

式 2‑2

### 直方图均衡化 – Histogram Equalization

直方图均衡化是通过拉伸像素强度分布范围来增强图像对比度的一种方法。

考虑一个图像，其像素值仅限于特定的值范围。 例如，较亮的图像会将所有像素限制在较高的值。 但是，好的图像将具有来自图像所有区域的像素。 因此，需要将此直方图拉伸到两端（图 2‑4），这就是直方图均衡化的作用。 通常，这可以提高图像的对比度。

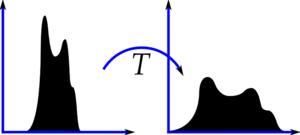


图 2‑4 图像均衡化前后的直方图对比

### 形态学操作 - Morphological Transformations

简单来讲，形态学操作就是基于形状的一系列图像处理操作。通过将结构元素作用于输入图像来产生输出图像。

腐蚀与膨胀(Erosion 与 Dilation)是形态学运算的基础。他们的运用广泛:

1. 消除噪声
2. 分割(isolate)独立的图像元素，以及连接(join)相邻的元素。
3. 寻找图像中的明显的极大值区域或极小值区域。

将腐蚀和膨胀操作结合，就可以实现开运算、闭运算、形态学梯度运算、顶帽运算、黑帽运算、击中击不中等不同形式的运算。



a 腐蚀图像



b 原始图像



c 膨胀图像

图 2‑5 形态学操作

### 距离变换 - Distance Transform

当图像内的各个子图没有连接时，可以直接使用形态学的腐蚀操作确定前景对象，但是如果图像内的子图连接在一起时，就很难确定对象了。

距离变换函数计算二值图像内任一点到背景点的距离。一般情况下，该函数计算的是图像内非零值像素点到最近的零值像素点的距离，即计算二值图像中所有像素点距离其最近的值为0的像素点的距离。当然，如果像素点本身的值为0，则这个距离也为0。

距离变换函数的计算结果反应了各个像素与背景（值为0的像素点）的距离关系。通常情况下：

1. 如果前景对象的中心（质心）距离值为0的像素点距离较远，会得到一个较大的值。
2. 如果前景对象的边缘距离值为0的像素点较近，会得到一个较小的值。

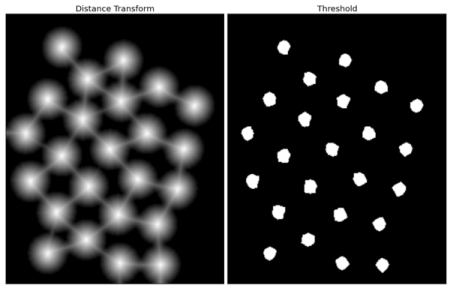
距离变换函数可以用于计算对象的中心，还能细化轮廓、获取图像前景等，

有多种功能。

### 阈值处理- Thresholding Operations

阈值处理是剔除图像内像素值高于一定值或低于一定值的像素点。

对距离变换后的结果进行阈值化，就可以得到图像内子图的中心、骨架等信息。将处理结果进行节点数量处理，将数量过少（此处为4，以拟合椭圆的最低标准为准）的淘汰掉，避免重复统计。



## 程序运行界面及结果分析

### 获取彩色图像

使用imdecode指定以RGB格式，读入指定的image图像，numpy在对文件编码进行转换，避免中文路径读取错误问题。



图 6‑1 计数图像读入

### 灰度处理

使用cvtColor函数对图像进行灰度处理，方便之后的图像处理。



图 6‑2 获取灰度图像

### 直方图均衡化

使用equalizeHist函数均衡化图像，去除噪点，增强图像的对比度。



图 6‑3 直方图均衡化

### 二值化处理

使用threshold 函数将图像二值化，分隔类型反二值化阈值处理（cv2.THRESH\_BINARY\_INV）和OTSU处理（cv2.THRESH\_OTSU）。

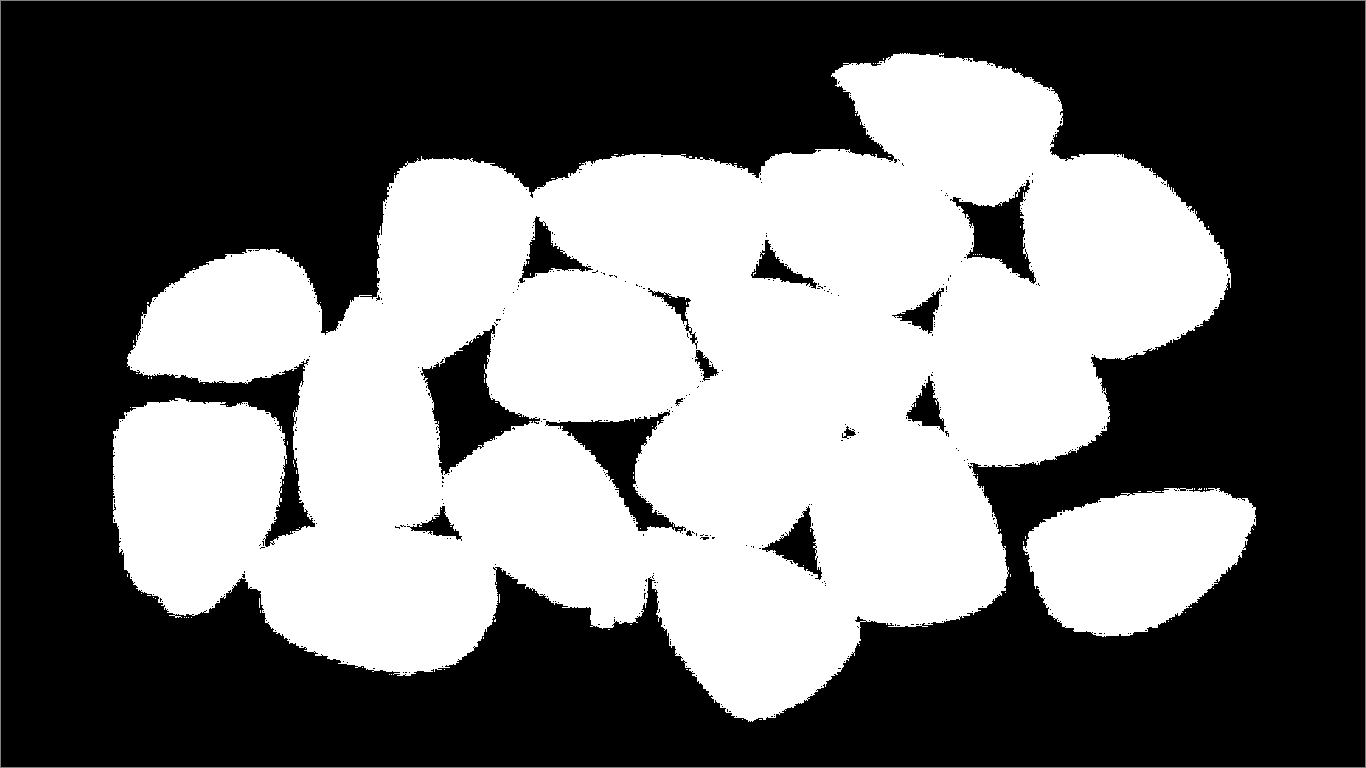


图 6‑4 二值化处理

### 形态学腐蚀

如图 6‑5，使用morphology函数的MORPH\_ERODE方法对二值化的图像进行处理，核函数使用np.ones((10,10),np.uint8)，处理10次。消除过小的像素点，将相连的大像素点的边界相互分离。

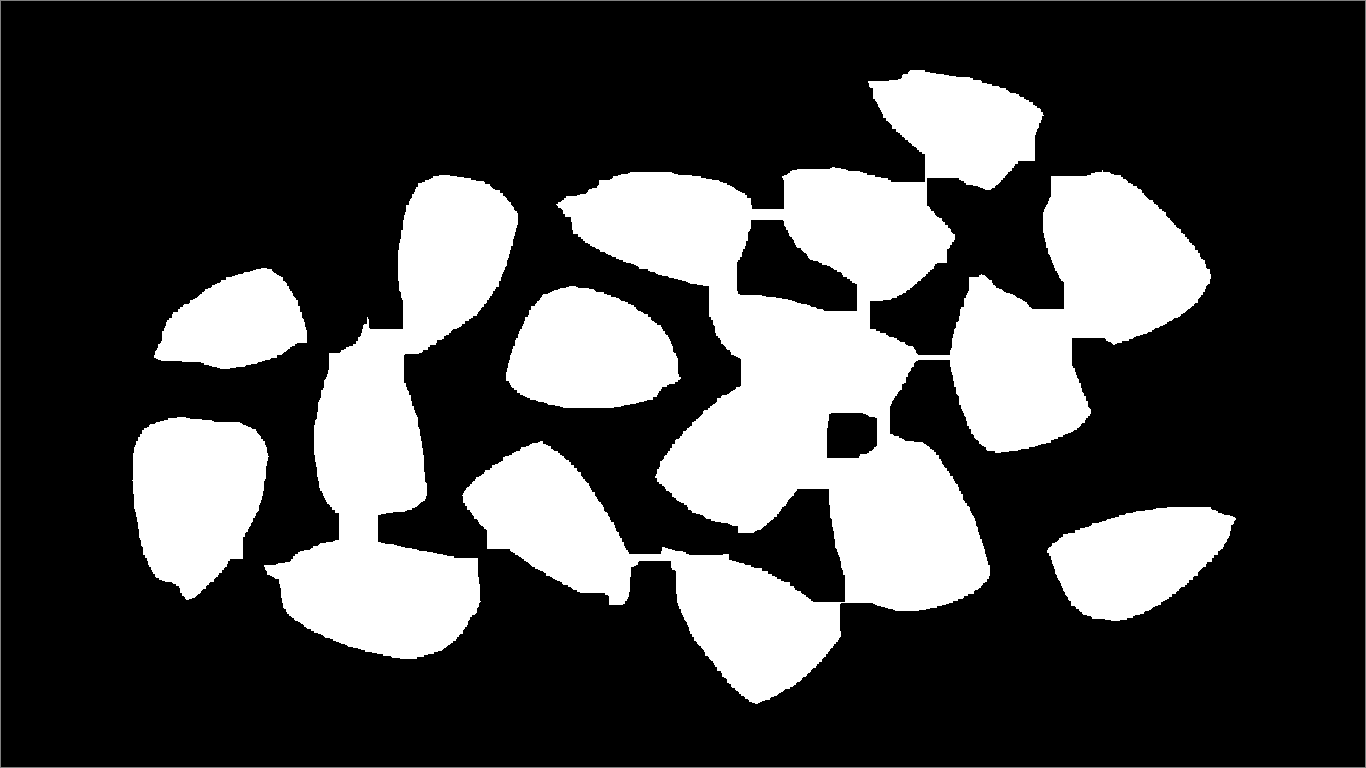


图 6‑5 形态学腐蚀

### 距离变换

如图 6‑6，使用distanceTransform进行距离变换，使用简单的欧几里得距离。

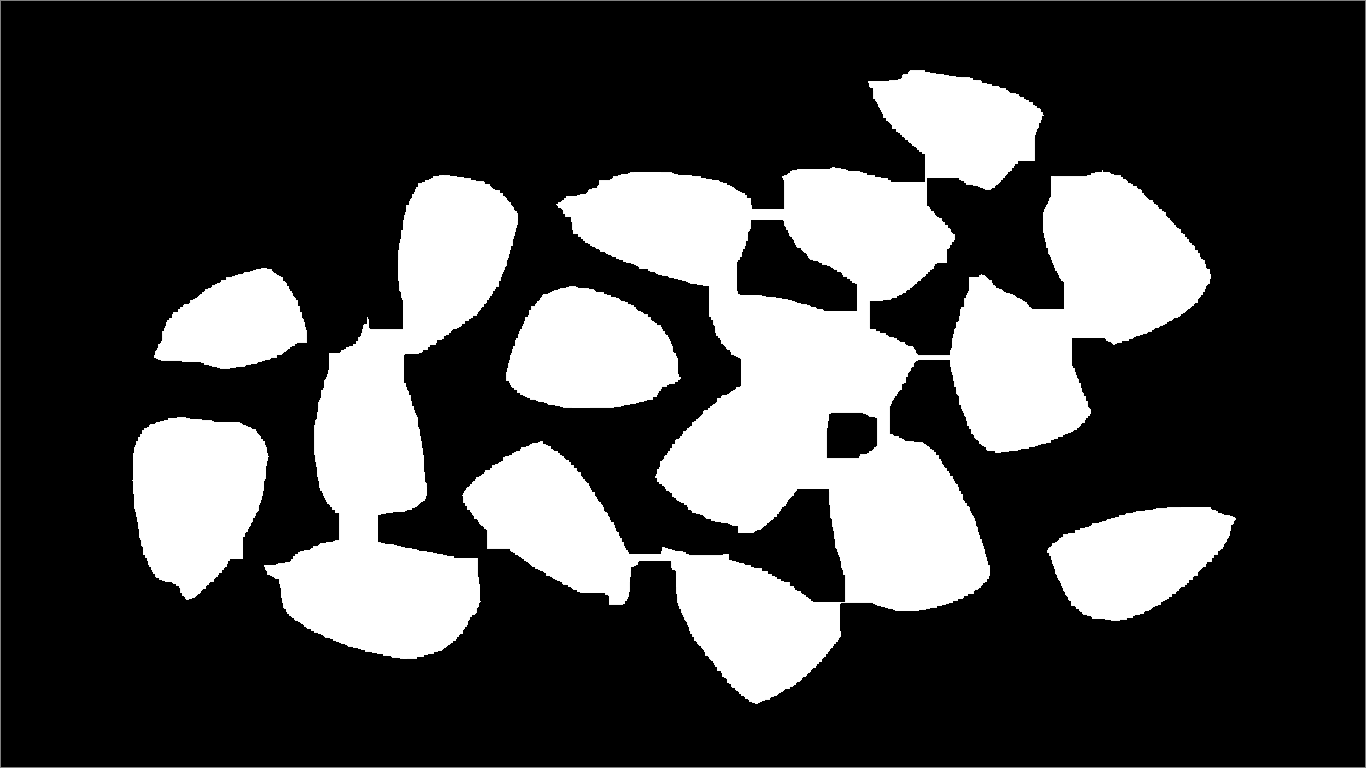


图 6‑6 距离变换

### 阈值处理

使用threshold 函数，将距离大于足够大的固定阈值（0.70\*imgs.max()）的点保留下来。

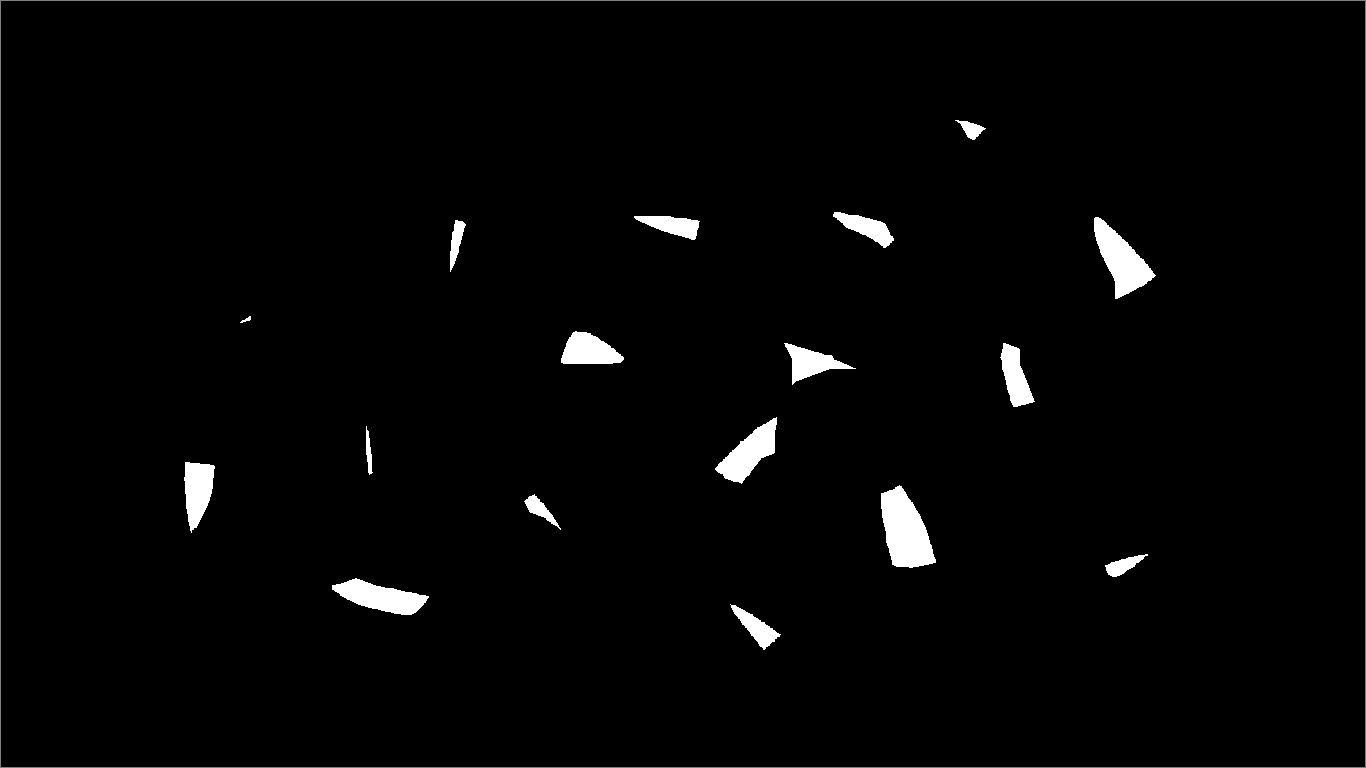


图 6‑7 阈值处理

### 数目标记

使用putText函数在获取到图像的指定点依次输出显示输出数字，方便查看、验证。



图 6‑8 计数结果图

## 源代码

from cv2 import cv2

import numpy as np

# 显示图片

def show(img,winname='windows'):

cv2.imshow(winname,img)

cv2.waitKey()

# 标注个数

def num(img,contours,i=0):

for contour in contours:

# 二值化图轮廓中符合最优拟合椭圆条件

if len(contour)>4:

# 标注个数

show(cv2.putText(img,str(i:=i+1),tuple(contour[1][0]),cv2.FONT\_HERSHEY\_SIMPLEX,5,(255,255,0),15))

# 主运行程序

def run(image):

# 获取图像

show(img := cv2.imdecode(np.fromfile(image,dtype=np.uint8),1))

# 灰度图像

show(img\_gray := cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY))

# 均衡化

show(img\_gray := cv2.equalizeHist(img\_gray))

# 二值化处理

show(thresh := cv2.threshold(img\_gray, 0, 255, cv2.THRESH\_BINARY\_INV + cv2.THRESH\_OTSU)[1])

# 核函数

kernel = np.ones((10,10),np.uint8)

# 形态学腐蚀变换

show(tmp := cv2.morphologyEx(thresh,cv2.MORPH\_ERODE,kernel,iterations=10))

# 距离变换

show(imgs := cv2.distanceTransform(tmp,cv2.DIST\_L2,3))

# 阈值处理

show(imgs := cv2.threshold(imgs,0.70\*imgs.max(),255,0)[1])

# 标注img图像

num(img,cv2.findContours(np.uint8(imgs), cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)[0])

# 测试图像

images = ['玉米颗粒.jpg']

if \_\_name\_\_=='\_\_main\_\_':

# 界面设置

cv2.namedWindow('windows',cv2.WINDOW\_NORMAL)

# 全屏显示

cv2.setWindowProperty('windows',cv2.WND\_PROP\_FULLSCREEN,cv2.WINDOW\_FULLSCREEN)

# 读取图像参数

from sys import argv

from os.path import join

images = argv[1:] if len(argv)>1 else (join('src',image) for image in images)

# 尝试处理图像

for image in images:

try:

run(image)

except FileNotFoundError:

print(image,'不存在')

continue