Table 4 Code example for diaspores deification and counting

|  |
| --- |
| import cv2  import numpy as np  imagepath1= "C:\\Users\\jay\\Desktop\\Photos\\Mixed\\1.png"  img1 = cv2.imread(imagepath1, cv2.IMREAD\_UNCHANGED)  # (1) Color space conversion (GBR to HSV)  imghsv = cv2.cvtColor(img1,cv2.COLOR\_BGR2HSV)  # (2)Definition of target color of stained diaspores  Lower1 = np.array([125,43,46])  Lower2 = np.array([0,43,46])  Upper1 = np.array([180,255,255])  Upper2 = np.array([10,255,255])  # (3) Mask obtaining of binary image  mask\_1 = cv2.inRange(imghsv, Lower1, Upper1)  mask\_2 = cv2.inRange(imghsv, Lower2, Upper2)  mask0= cv2.add(mask\_1,mask\_2)  res = cv2.bitwise\_and(img1, img1, mask=mask0)  blurred = cv2.blur(res,(10,10))  blurred = cv2.cvtColor(blurred,cv2.COLOR\_BGR2HSV)  mask1 = cv2.inRange(blurred, Lower1, Upper1)  mask2 = cv2.inRange(blurred, Lower2, Upper2)  mask= cv2.add(mask1,mask2)  # (4) Morphological processing  kernel = cv2.getStructuringElement(cv2.MORPH\_RECT, (20, 20))  Opened=cv2.morphologyEx(mask, cv2.MORPH\_OPEN, kernel)  kernel2 = cv2.getStructuringElement(cv2.MORPH\_RECT, (5, 5))  Closed2=cv2.morphologyEx(Opened,cv2.MORPH\_CLOSE, kernel2)  # (5) Contours discovery and numbering of found contours on photos  binary,contours,hierarchy = cv2.findContours(Closed2,cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)  print("number of diaspores: %d" % (len(contours)))  img1\_copy= img1.copy()  cv2.drawContours(img1\_copy, contours,-1, (0,255,0),1)  for i in range(len(contours)):  rect = cv2.minAreaRect(contours[i])  x,y = rect[0]  center = (int(x-20), int(y+10))  angle = rect[2]  font=cv2.FONT\_HERSHEY\_SIMPLEX  cv2.putText(img1\_copy,str(i+1),center,font,1,(0,255,0),1)  cv2.imshow('Result1', img1\_copy)  cv2.waitKey(0)  cv2.imwrite('Result1.jpg',img1\_copy)  cv2.putText(img1\_copy,("number of contours: %d" % (len(contours))),(200,200),font,2,(0,255,0),2) |

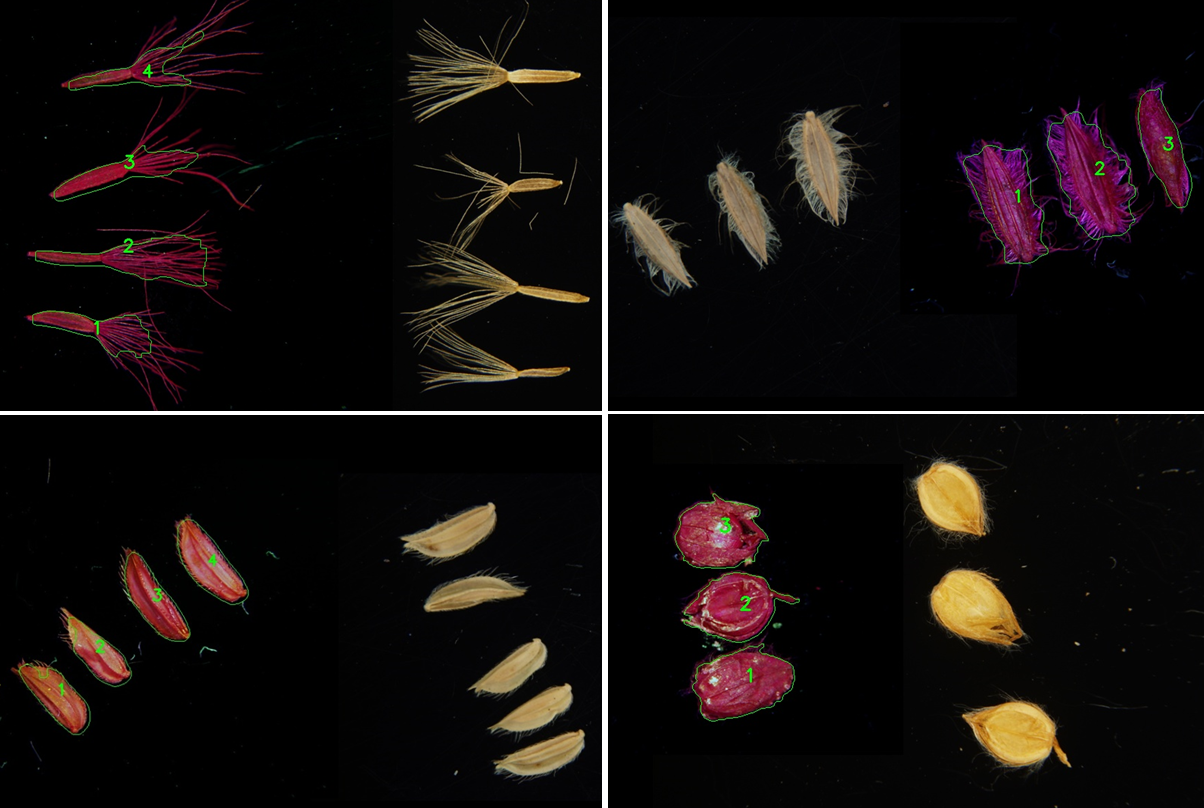


Figure 9 Examples of the indemnification of stained diaspores using openCV on python platform