#实现图型匹配

1，首先引入库 \*\*cv2,numpy，random\*\*

2，定义一个读取图片的函数，方便接下来代码内的读取。

def cv\_show(name,img):

cv2.imshow(name,img)

cv2.waitKey(0)

cv2.destroyAllWindows()

3，运用numpy创建画布

4，使用cv2函数circle画圆,rectangle画矩形，ellipse画椭圆，或者用numpy定义几个点再使用cv2的polylines函数连线

#边缘检测

1，使用Canny函数进行识别，定义阈值

2，通过findcontours返回轮廓和结构

3，因为contours会将所检测的轮廓计入列表，所以直接用len读取

#标记轮廓

通过drawcontours函数绘制轮廓，用黑线描出

#截取

利用contours列表随机选取一个作为模板

#模板匹配

result = cv2.matchTemplate(canvas,template,cv2.TM\_CCOEFF\_NORMED)

#定位标注

loc = np.where(result > 0.8) \*\*选出result数组中大于的元素位置\*\*

for pt in zip(\*loc[::-1]):

bottom\_right = (pt[0] + w, pt[1] + h)

这是将逆序后的索引元组进行解包并对应打包，循环遍历这些打包后的元素，每个元素赋值给 pt

最后通过rectangle画矩形将识别的图形（获得了坐标（pt）和长宽(h,w)）框住