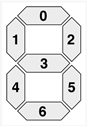
1. 识别每个数字

1.1、准备数字查询矩阵

我们可以了解到，像电表这种液晶屏显示数字0-9，总共是由图中的7个笔划图形组合而成，因此我们可以根据是否存在相应笔划，判断是哪个数字



IMG_256

可以通过以下字典，映射笔划和对应的数字：

DIGITS\_LOOKUP = {

    (1, 1, 1, 0, 1, 1, 1): 0,

    (0, 0, 1, 0, 0, 1, 0): 1,

    (1, 0, 1, 1, 1, 0, 1): 2,

    (1, 0, 1, 1, 0, 1, 1): 3,

    (0, 1, 1, 1, 0, 1, 0): 4,

    (1, 1, 0, 1, 0, 1, 1): 5,

    (1, 1, 0, 1, 1, 1, 1): 6,

    (1, 0, 1, 0, 0, 1, 0): 7,

    (1, 1, 1, 1, 1, 1, 1): 8,

    (1, 1, 1, 1, 0, 1, 1): 9

}

1.2、判断每个数字

将数字区域排序，这样会从左到右排列所有区域

import numpy as np

digitCnts = sorted(digitCnts)

print(digitCnts)

digits = []

for c in range(len(digitCnts)):

    # 提取数字区域的坐标和宽高

    (x, y, w, h) = digitCnts[c]

    roi = thresh[y:y + h, x:x + w]

    roi\_name = 'roi'+ str(c) +'.png'

    print("writing to ", roi\_name)

    #根据当前的数字图像的大小，取四分之一作为笔划的宽度，大约七分之一作为笔划的高度。

    cv2.imwrite(roi\_name, roi)

    (roiH, roiW) = roi.shape

    (dW, dH) = (int(roiW \* 0.25), int(roiH \* 0.15))

    dHC = int(roiH \* 0.05)

     #将当前的数字区域划分成7个区域：

    segments = [

    ((dW, 0), (w-dW, dH)), # top

    ((0, dH), (dW, (h // 2)-dHC)), # top-left

    ((w - dW, dH), (w, (h // 2) - dHC)), # top-right

    ((dW, (h // 2) - dHC) , (w-dW, (h // 2) + dHC)), # center

    ((0, h // 2+dHC), (dW, h-dH)), # bottom-left

    ((w - dW, h // 2+dHC), (w, h-dH)), # bottom-right

    ((dW, h - dH), (w-dW, h))]# bottom

#检查每个笔划区域，如果前景（即白色像素点）区域大小超过区域的50%，则认为当前笔划存在

    on = [0] \* len(segments)

    # 检查每个笔画区域

    for (i, ((xA, yA), (xB, yB))) in enumerate(segments):

        # 提取笔画区域，计算区域大小以及前景像素

        segROI = roi[yA:yB, xA:xB]

        total = cv2.countNonZero(segROI)

        area = (xB - xA) \* (yB - yA)

        print("{} percent is {}".format(i, total / float(area)))

        # 如果非零像素数量多于50%，则这个笔画存在,因此获取每个笔划的范围，并组成笔划数组

        if total / float(area) > 0.5:

                    on[i]= 1

    #根据笔划数组判断是否是哪个数字

    digit = DIGITS\_LOOKUP[tuple(on)]

    print(digit)

    digits.append(digit)

    #在原图上对数字区域画框，并标注识别结果

    cv2.rectangle(output, (x, y), (x + w, y + h), (0, 255, 0), 1)

    cv2.putText(output, str(digit), (x - 10, y - 10),

    cv2.FONT\_HERSHEY\_SIMPLEX, 0.65, (0, 255, 0), 2)

#查看标注识别结果图像

plt.imshow(output,cmap="gray")

print(digits)

获得识别结果如下：

[3, 4, 5]

