

用python和opencv检测图像中的条形码

神秘的铁头娃 小白学视觉 今天

点击上方“[小白学视觉](#)”，选择加“[星标](#)”或“[置顶](#)”

重磅干货，第一时间送达

在学习中发现快乐，在应用找到价值。这是我第五期分享图像技术应用的文章。

前四期欢迎阅读和分享：

- 第一期《[扫描全能王？原来图像技术可以这样子玩](#)》
- 第二期《[来吧，见识科技的力量，无需手动找拍糊的图](#)》
- 第三期《[这种方式打开会ctrl的流量明星cxk，简直就是魔鬼](#)》
- 第四期《[酷炫骚操作，票圈装13神技，极坐标全景图](#)》

概述

在日常生活中，经常会看到条形码的应用，比如超市买东西的生活，图书馆借书的时候。。。

那么这些东西是如何做到准确检测出条形码的位置呢？

这就是今天要介绍的内容了

这篇博文的目标是演示使用计算机视觉和图像处理技术实现条形码的检测。

通过本篇文章的学习，我们能学到的内容包括：

- 1、图像处理中常用的一些操作流程，包括滤波、阈值化处理、膨胀、腐蚀和轮廓查找等
- 2、更重要的一点，希望通过这个案例，能够帮助大家建立分析问题和处理问题的思路

需要注意的是，这个算法并不适用于所有的条形码，但是它应该能给你一个基本的直觉，告诉你应该应用什么类型的技术。



条形码的检测

对于下面这个例子，我们将检测下图中的条形码：



目标：找到条形码的位置，而去除掉干扰的因素

思路：

利用**条形码的自身特点**，一般都是矩形形状，而且条码的条带是黑色的，矩形区域是白色的

让我们继续写一些代码。打开一个新文件，将其命名为 `detect_barcode.py`，如何开始写代码：

```
1 # 导入必要的包/库
2 import numpy as np
3 import argparse
4 import imutils
5 import cv2
6
7 # 构造参数解析并分析参数
```

```
8 ap = argparse.ArgumentParser()
9 ap.add_argument("-i","--image", required = True, help = "path to the image file")
10 args = vars(ap.parse_args())
```

我们首先要做的是导入我们需要的包。我们将使用numpy进行数字处理，argparse用于解析命令行参数，cv2进行opencv绑定。

然后我们将设置命令行参数。我们这里只需要一个开关，即 --image，它是我们图像的路径，其中包含我们想要检测的条形码。

现在，让我进行实际的图像处理吧

导入图片并转化为灰度图

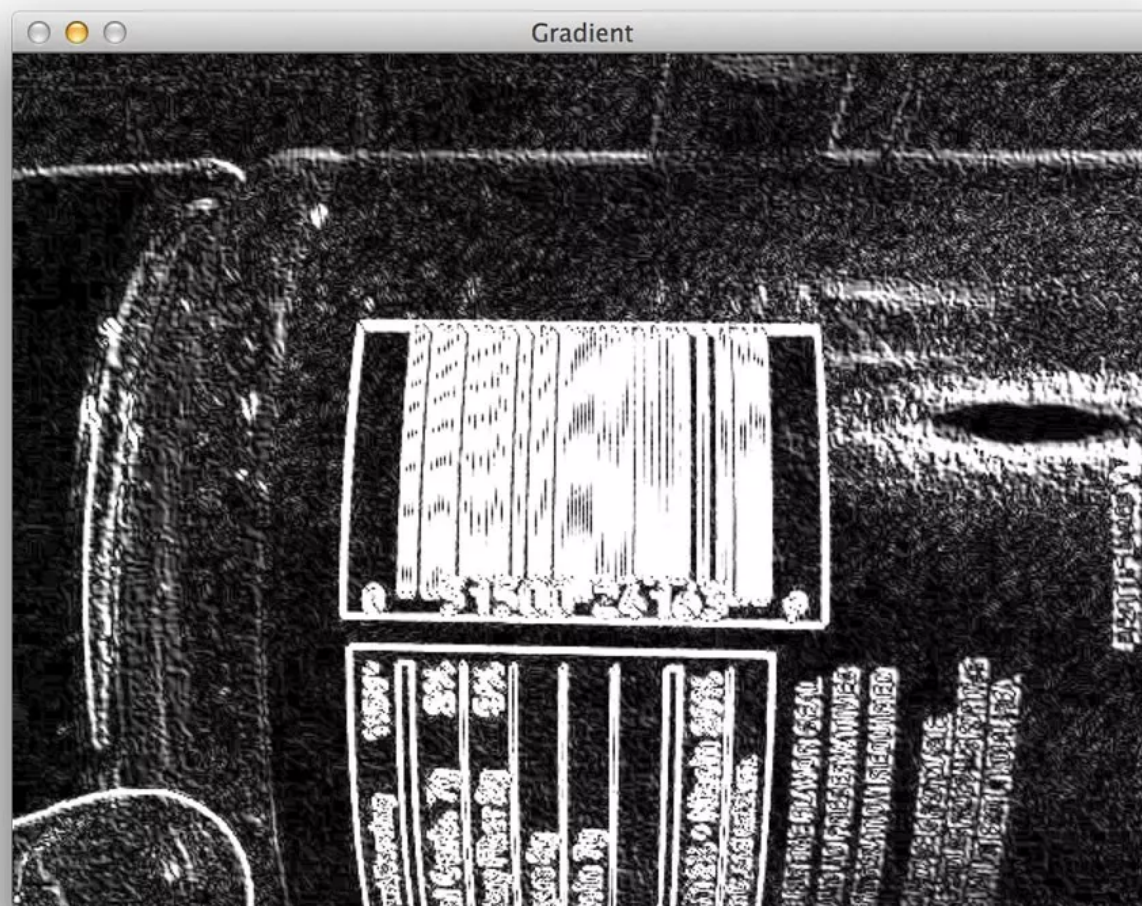
```
1 image = cv2.imread(args["image"])
2 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
3
4 # 计算图片x和y方向的Scharrr梯度大小
5 ddepth = cv2.CV_32F if imutils.is_cv2() else cv2.CV_32F
6 gradX = cv2.Sobel(gray, ddepth=ddepth , dx=1, dy=0, ksize=-1)
7 gradY = cv2.Sobel(gray, ddepth=ddepth , dx=0, dy=1, ksize=-1)
8
9 # 用x方向的梯度减去y方向的梯度
10 gradient = cv2.subtract(gradX, gradY)
11 gradient = cv2.convertScaleAbs(gradient)
```

首先我们先读入图片并将其转换为灰度图；

然后我们使用Scharr操作符（这里制定ksize=1）去构造图片在水平和垂直方向上的梯度幅值表示。

这里，我们用Scharr算子的x方向梯度减去y方向的梯度。通过这个相减操作，我们就只剩下了高水平梯度和低垂直梯度的图像区域。

我们上述原始图像的梯度表示如下图所示



请注意梯度操作是如何检测出图片的条形码区域的。接下来的步骤是如何过滤掉图片中的噪声，重点关注条形码区域。

```
1 # 对图片进行模糊和阈值化操作
2 blurred = cv2.blur(gradient,(9,9))
```

```
3 (_,thresh) = cv2.threshold(blurred, 255, 255, cv2.THRESH_BINARY)
```

我们做的第一件事是使用一个卷积核大小为9x9的均值滤波作用于梯度图片。对图片进行这个操作将有助于平滑图片中的高频噪声。

然后我将模糊化后的图片进行阈值化，在梯度图片中，所有的像素点的灰度值低于255的将设为0（黑色），其余设为255（白色）。

模糊和阈值化处理后的输出结果如下：



然而，正如你所看到的上图阈值化图片，在条形码的垂直方向上存在这间隙。为了缩小这些间隙，并使得我们的算法更加容易的检测出条形码的“斑点”状区域，我们需要执行一些基础的形态学操作：

```
1 # 构造一个闭合核并应用于阈值图片
2 kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (21, 7))
```



```
3 closed = cv2.morphologyEx(thresh, cv2.MORPH_CLOSE, kernel)
```

我们通过使用函数`cv2.getStructuringElement`构造一个矩形核。这个核的宽度大于高度，因此允许我们缩小条形码垂直条带之间的间隙。



当然，现在图片中非条形码的区域存在着很多斑点，这将会干扰轮廓的检测。

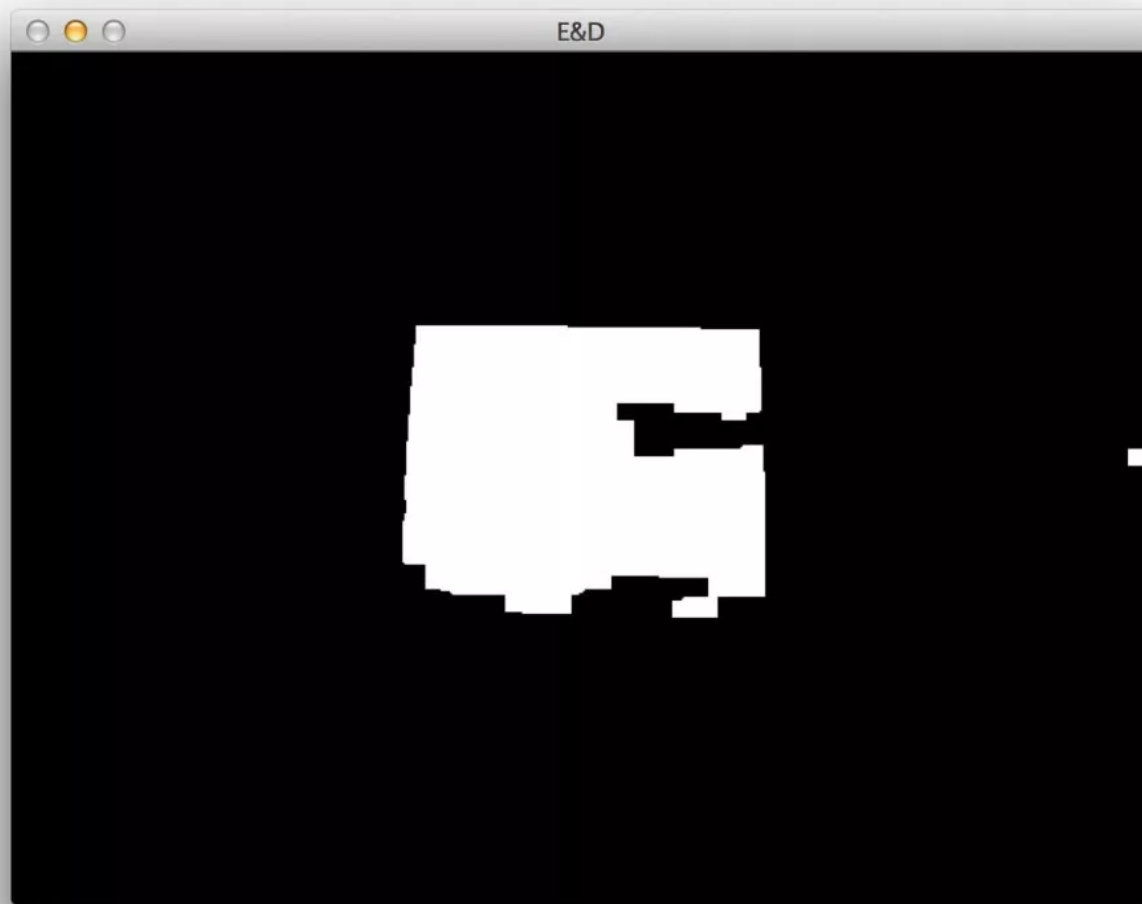
现在，我们继续尝试移除这些小的斑点

```
1 # 执行一系列的腐蚀和膨胀操作
2 closed = cv2.erode(closed, None, iterations = 4)
3 closed = cv2.dilate(closed, None, iterations = 4)
```

我们现在要做的是进行4次腐蚀操作，然后再进行4次膨胀操作。腐蚀操作将会“腐蚀”掉图片中的白色像素点，因此将会清除这些小的斑点，而膨胀操作将会“扩张”剩余的白色像素，并使白色区域变长。

如果在腐蚀过程中去除了小的斑点，则在膨胀的过程中不会再次出现。

在一系列的腐蚀和膨胀操作之后，这些小斑点已经被成功的移除了，只剩下条形码的区域。



最后寻找一下图片中条形码的区域的轮廓。

```
1 # 找到阈值化后图片中的轮廓，然后进行根据区域进行排序，仅保留最大区域
2 cnts = cv2.findContours(closed.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
3 cnts = imutils.grab_contours(cnts)
```

```
4 c = sorted(cnts, key = cv2.contourArea, reverse = True)[0]
5
6 # 计算最大轮廓的旋转边界框
7 rect = cv2.minAreaRect(c)
8 box = cv2.cv.BoxPoints(rect) if imutils.is_cv2() else cv2.boxPoints(rect)
9 box = np.int0(box)
10
11 # 在检测到的条形码周围绘制边界框并显示图片
12 cv2.drawContours(image, [box], -1, (0,255,0), 3)
13 cv2.imshow("Image", image)
14 cv2.waitKey(0)
```

幸运的是，**opencv**中提供了相应的接口，可以很容易地找到图像中的最大轮廓，如果我们正确地完成了图像处理步骤，它应该会对应于条形码区域。

然后，我们确定最大轮廓的最小的边界框，并最后显示检测到的条形码。

正如我们下图所示，我们已经成功的检测到条形码



详细代码链接：

https://github.com/DWCTOD/AI_study/tree/master/%E5%90%88%E6%A0%BC%E7%9A%84CV%E5%B7%A5%E7%A8%8B%E5%B8%88/%E5%AE%9E%E6%88%98%E7%AF%87/opencv/%EF%BC%88%E4%BA%94%EF%BC%89%E7%94%A8python%E5%92%8Copencv%E6%A3%80%E6%B5%8B%E5%9B%BE%E5%83%8F%E4%B8%AD%E7%9A%84%E6%9D%A1%E5%BD%A2%E7%A0%81

使用方法: `python3 detect_barcode.py --image images/barcode_01.jpg`

另外还提供了其他的测试图片

英文原文链接: <https://www.pyimagesearch.com/2014/11/24/detecting-barcodes-images-python-opencv/>

总结

值得去思考的点

- 1、为啥要转变成灰度图
- 2、如何去除干扰因素，特别是条形码下面的材料说明的区域是采用什么办法去除的，小的斑点是如何去除的

下载1: OpenCV-Contrib扩展模块中文版教程

在「小白学视觉」公众号后台回复：**扩展模块中文教程**，即可下载全网第一份OpenCV扩展模块教程中文版，涵盖**扩展模块安装、SFM算法、立体视觉、目标跟踪、生物视觉、超分辨率处理**等二十多章内容。

下载2: Python视觉实战项目52讲

在「小白学视觉」公众号后台回复：**Python视觉实战项目**，即可下载包括**图像分割、口罩检测、车道线检测、车辆计数、添加眼线、车牌识别、字符识别、情绪检测、文本内容提取、面部识别**等31个视觉实战项目，助力快速学校计算机视觉。

下载3: OpenCV实战项目20讲

在「小白学视觉」公众号后台回复：**OpenCV实战项目20讲**，即可下载含有**20个基于OpenCV实现20个实战项目**，实现OpenCV学习进阶。

交流群

欢迎加入公众号读者群一起和同行交流，目前有**SLAM、三维视觉、传感器、自动驾驶、计算摄影、检测、分割、识别、医学影像、GAN、算法竞赛**等微信群（以后会逐渐细分），请扫描下面微信号加群，备注：“昵称+学校/公司+研究方向”，例如：“张三 + 上海交大 + 视觉SLAM”。**请按照格**

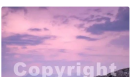
式备注，否则不予通过。添加成功后会根据研究方向邀请进入相关微信群。**请勿**在群内发送**广告**，否则会请出群，谢谢理解~



喜欢此内容的人还喜欢

如何使用OpenCV为图像加水印

<https://mp.weixin.qq.com/s/J73BKSzPzgOMOOgqDCWug>

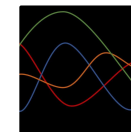


深度学习与计算机视觉



贝叶斯机器学习：经典模型与代码实现

机器学习实验室



从职高到杭电、浙大、MIT计算机博士！

Datawhale

