

# 用OpenCV實現條形碼識別

梁峻豪等 OpenCV團隊 昨天

作者：梁峻豪，王天麒，孫中夏（南方科技大學計算機科學與工程系）

最近，我們為OpenCV貢獻了一維條形碼識別模塊，代碼收錄在：

[https://github.com/opencv/opencv\\_contrib/tree/master/modules/barcode](https://github.com/opencv/opencv_contrib/tree/master/modules/barcode)。

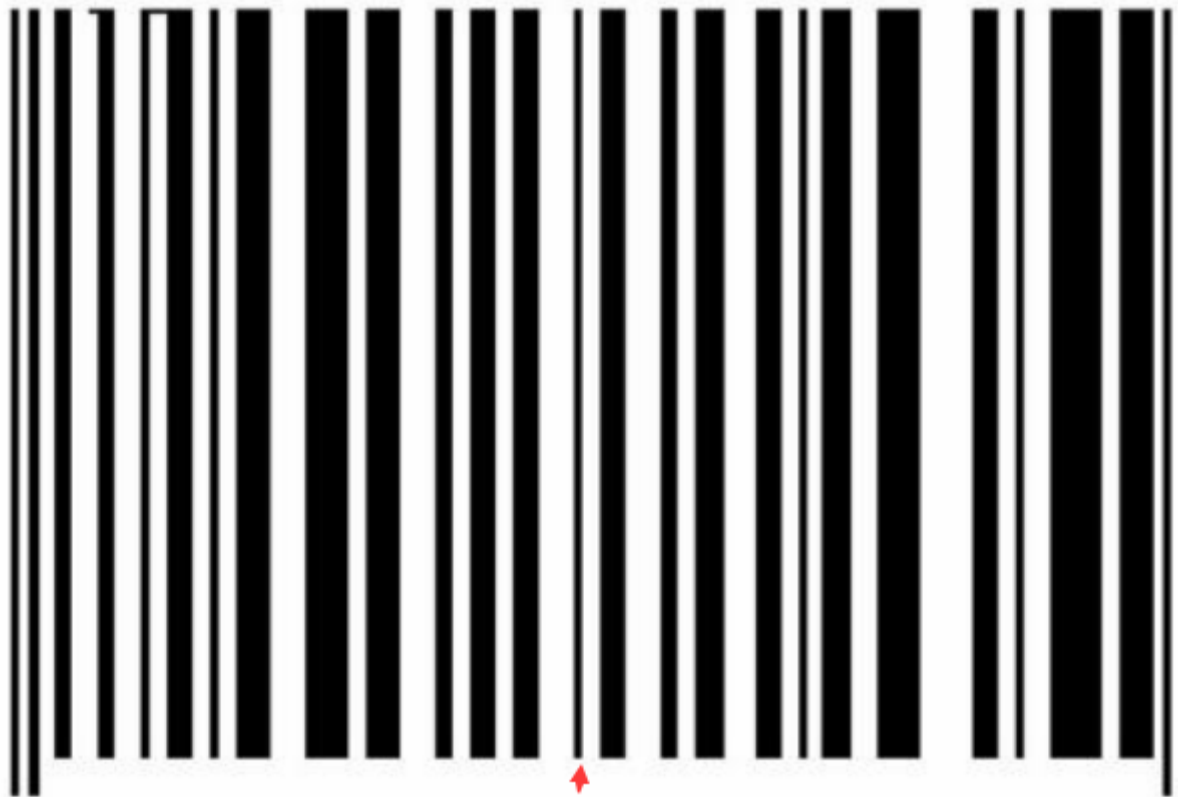
我們收集的數據集（數據集地址：<https://github.com/SUSTech-OpenCV/BarcodeTestDataset>，共250張條碼圖片）上進行了測試，我們的識別算法正確率達到了**96%**，速度為**20ms**每張圖像。作為對比，我們也測試了ZXing在該數據集上的表現，其正確率為**64.4%**，速度為**90ms**每張圖像。

注：測試速度不包含初始化以及讀圖時間。同時，我們的算法是C++實現，ZXing是Java實現。另外，對於用圖片數據集進行的測試，ZXing99%的時間是在做彩色圖到灰度圖的轉換。

本文將對此模塊的原理和使用方式進行介紹。

## 條形碼介紹

條形碼是將寬度不等的多個黑條和空白，按照一定的編碼規則排列，用以表達一組信息的圖形標識符，如下圖所示：



條碼區域與其他圖像相比有如下兩個重要特點：第一，條碼區域內的條空是平行排列的，方向趨於一致；第二，為了條碼的可識讀性，條碼在製作時條和空之間有著較大的反射率差，從而條碼區域內的灰度對比度較大，而且邊緣信息豐富。

## 基於方向一致性的條碼定位算法

1. 根據條形碼方向趨於一致的特點，我們可以將圖像分塊，通過計算每個塊內**梯度方向的一致性**，來濾除那些**低一致性的塊**。下圖是篩選過後剩下的塊：



2. 由於包含條碼區域的塊**一定連續存在**的特性，我們可以通過對這些圖像塊再進行一個改進的**腐蝕**操作過濾掉部分背景圖像塊。下圖是濾除部分背景圖像塊後剩餘的塊：



3. 得到這些塊之後，我們再根據每個圖像塊內的**平均梯度方向進行連通**。因為如果是相鄰的圖像塊都屬於同一個條碼的話，那麼他們的平均梯度方向也一定相同。

4. 得到連通區域之後我們再根據條碼圖像的特性進行篩選，比如連通區域內的梯度大於閾值的點的比例，組成連通區域的圖像塊數量等。
5. 最後，用**最小外接矩形**去擬合每個連通區域，併計算外界矩形的方向是否和連通區域內的平均梯度方向一致，過濾掉差距較大的連通區域。將平均梯度方向作為矩形的方向，並將矩形作為最終的定位框。

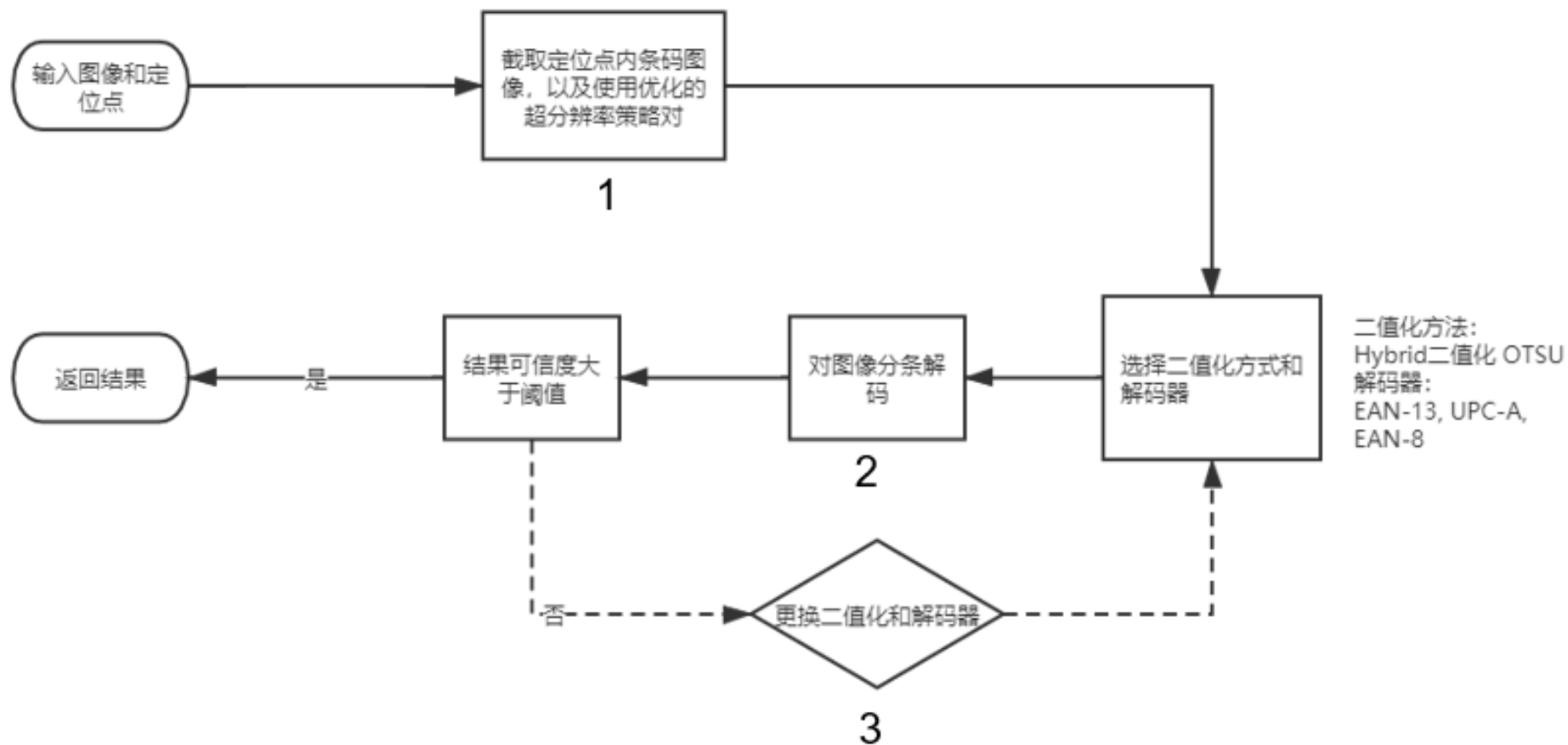


## 條形碼解碼

目前我們支持了三種類型的條碼解碼，它們分別是EAN13、EAN8 和UPC-A。(下圖為EAN13 條碼示例)



條碼的識別主要流程如下圖：



其中：

1. 优化的超分辨率策略指的是对较小的条码进行**超分辨率放大**，不同大小条码做不同处理。
2. 解码算法的核心是基于条码编码方式的**向量距离计算**。因为条码的编码格式为固定的数个"条空"，所以可以在约定好"条空"间隔之后，将固定的条空读取为一个向量，接下来与约定好的编码格式向匹配，取匹配程度最高的编码为结果。
3. 在解码步骤中，解码的单位为一条线，由于噪点，条空的粘连等原因，单独条码的解码结果存在较大的不确定性，因此我们加入了对**多条线的扫码**，通过对均匀分布的扫描与解码，能够将二值化过程中的一些不完美之处加以抹除。

**具体实现为：**首先在检测线上寻找起始符，寻找到起始符之后，对前半部分进行读取与解码，接着寻找中间分割符，接着对后半部分进行读取与解码，最后寻找终结符，并对整个条码进行首位生成与校验（此处以EAN13格式举例，不同格式不尽相同）。最后，每条线都会存在一个解码结果，所以对其进行投票，只将最高且总比例在有效结果50%以上的结果返回。这一部分我们基于ZXing的算法实现做了一些改进(投票等)。

4. **更换二值化和解码器**指的是在为解码成功遍历使用每种解码器和二值化尝试解码。

## 使用方式

### C++

```
v2/barcode.hpp"
v2/imgproc.hpp"

cv;

rcodeDetector> bardet = makePtr<barcode::BarcodeDetector>("sr.prototxt", "sr.caffemodel"); //如果不使用超分辨率则可以不指定模型路径
ead("your file path");
返回的检测框的四个角点坐标，如果检测到N个条码，那么维度应该是[N][4][2]
::string> decoded_info; //返回的解码结果，如果解码失败，则为空string
code::BarcodeType> decoded_format; //返回的条码类型，如果解码失败，则为BarcodeType::NONE
t->detectAndDecode(input, decoded_info, decoded_format, corners);
```

### Python

```
import cv2

bardet = cv2.barcode_BarcodeDetector()
img = cv2.imread("your file path")
ok, decoded_info, decoded_type, corners = bardet.detectAndDecode(img)
```

更多使用方式请参考文档：

[https://docs.opencv.org/master/dc/df7/classcv\\_1\\_1barcode\\_1\\_1BarcodeDetector.html](https://docs.opencv.org/master/dc/df7/classcv_1_1barcode_1_1BarcodeDetector.html)

## 参考文献

王祥敏，汪国有. 一种基于方向一致性的条码定位算法[EB/OL]. 北京：中国科技论文在线 [2015-04-22]. <http://www.paper.edu.cn/releasepaper/content/201504-338>.

>>>> [OpenCV中国团队招聘实习生](#) <<<<



Webinar 1: OpenCV Overview

Webinar 2: OpenCV DNN在ARM上的加速

Webinar 3: OpenCV深度学习应用与原理分析

Webinar 4: 谷歌编程夏令营2020 OpenCV文本识别项目开发

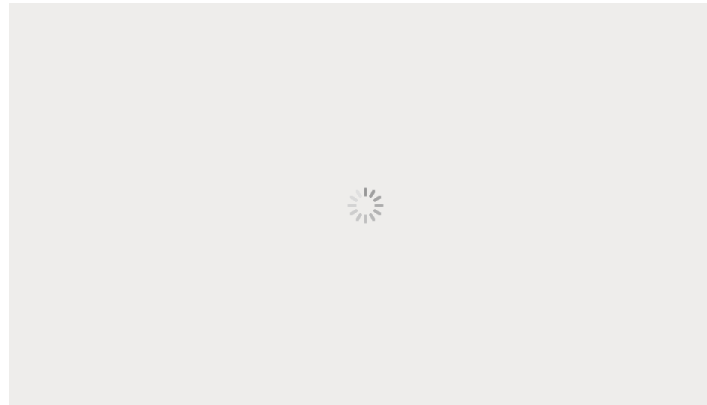
Webinar 5: 谷歌编程夏令营2020 OpenCV RISC-V项目开发

Webinar 6: AI on Edge Devices

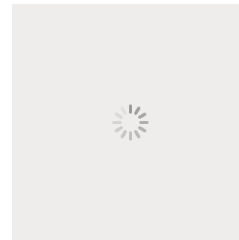
Webinar 7: 微信二维码引擎及其OpenCV开源模块解析

**[OpenCV中国团队官方推荐和认证OpenCV在线课程](#)**





OpenCV中国团队于2019年9月由深圳市人工智能与机器人研究院支持成立，非营利目的，致力于OpenCV的开发、维护和推广工作。  
长按下方QR码关注我们获取最新动态



文章已于2021/04/23修改

喜欢此内容的人还喜欢

Jure Leskovec (H-Index:114) 2021与图网络/机器学习相关研究一览

深度学习与图网络



## 基於OpenCV深度學習的邊緣檢測

新機器視覺

