

面对直线，你说霍夫线变换是万能的吗？

周旋 周旋机器视觉 2020-06-07 17:58

点击上方"蓝色小字"关注我哦！



今天给大家分享一个小案例，这个案例来自贾志刚老师的小案例实战课程，该课程有六个案例，我觉的非常好，所以来和大家分享一下，本文也不会标原创。大家也可以去支持一下：

<https://edu.51cto.com/course/8354.html>

然后我想问一下，当讲师这么挣钱的吗？学员上百万，一个学员就算是10块钱，也就上千万，何况每门课一般都上百元... ..



贾志刚

金牌讲师

课程：31门

学员数量：1276089人

讲师评分：4.8

进入主页

擅长领域：计算机视觉OpenCV

讲师介绍：个人独立开发者,专注图像处理研究多年时间,图书作者,博客专家。精通OpenCV框架多个模块、ImageJ开源框架、图像算法与应用开发技术、特别是在对象检测、识别、OCR、人脸美化等方面有深入研究。精通OpenCV4Android/Java与Java平台图像处理应用开发、开发过多个图像处理算法模块并成功应用在工业、医学检测等领域。



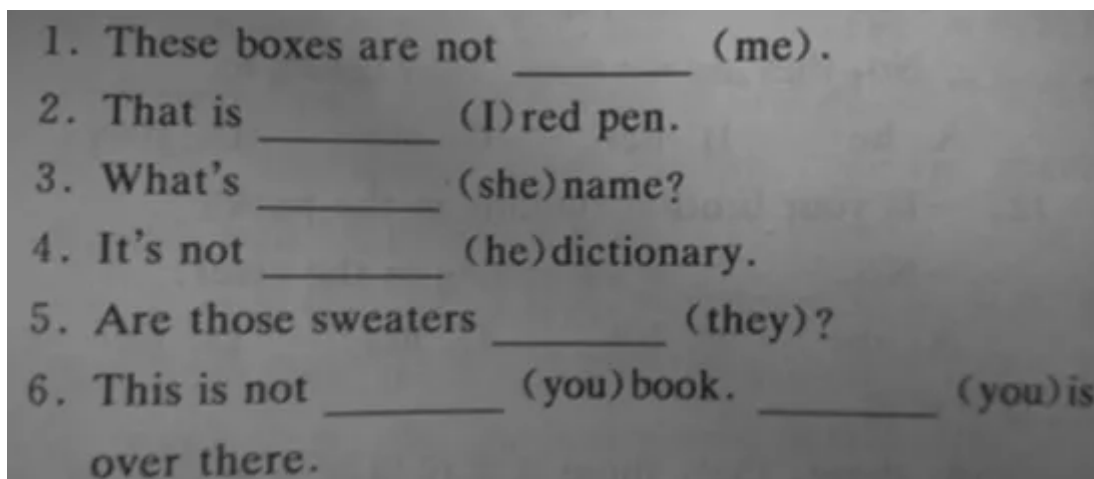
OpenCV4 深度神经网络(DNN)实战教程

理解卷积神经网络基本原理，熟练掌握OpenCV深度神经网络模块API开发技术，学会加载模型，解析模型输出结果；学会如何把正常的tensorflow对象检测模型转化为OpenCV可以接受的模型，实时人脸检测与识别案例。学会使用OpenCV DNN模块解决实际问题。部分课程代码演示效果如下：对

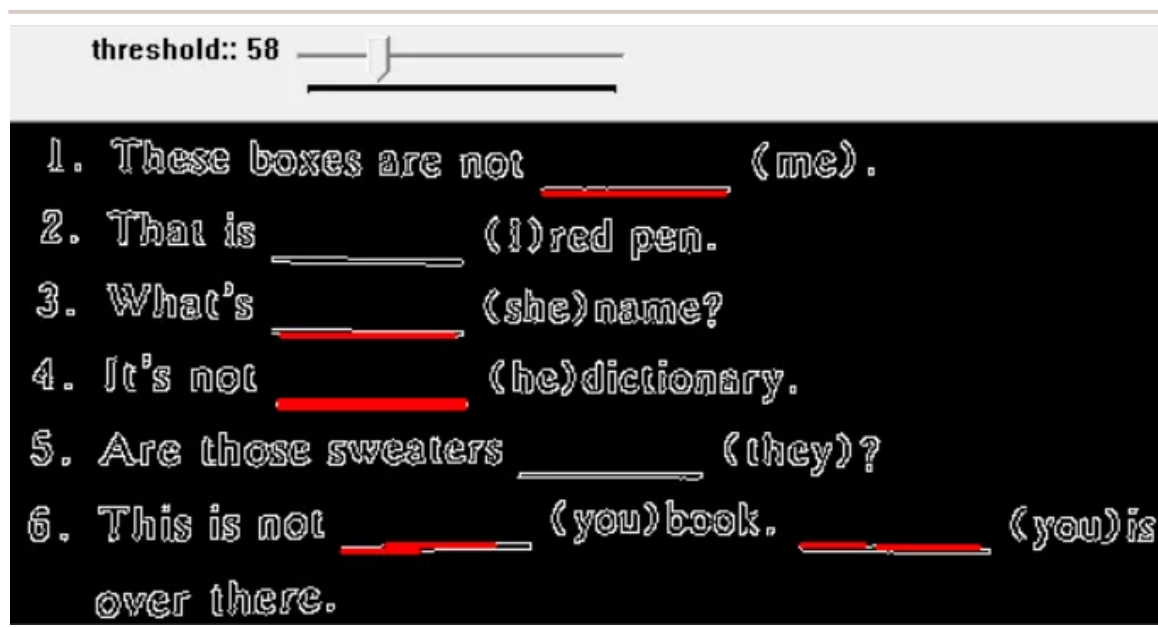
贾志刚 共24课时 6470人学习 4.8

¥129.00

还是来看这个案例吧。我们要做的是提取其中的直线填空，提取直线当然用霍夫线变换啦，但粗暴的直接霍夫线变换是不行的。



我们先看一下直接边缘检测并霍夫线变换的效果：

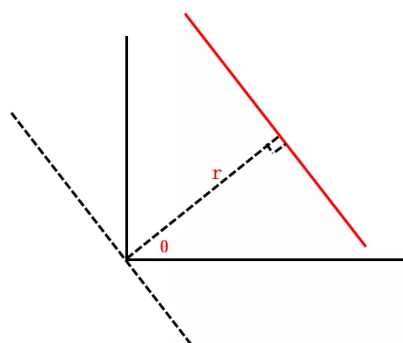


不论如何调阈值，都不能完美的检测出这几条直线，所以我们需要找更为普适的方法。

开始之前，先普及一下霍夫线变换的数学原理，不想看的可跳过。

数学原理

一条直线我们可以用 $y=kx+b$ 表示，也可以用极坐标表示，但还可以用下图方法表示：



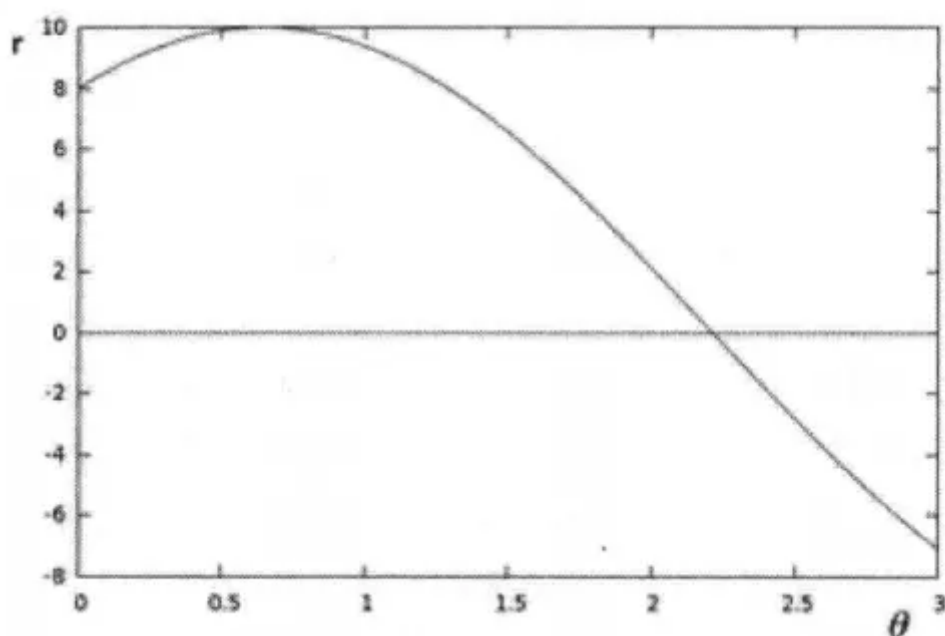
我们将一条直线（红色）用其到坐标原点的垂直距离 r 和其与水平方向的夹角 θ 来表示，这时可以表示该直线为：

$$y = \left(-\frac{\cos \theta}{\sin \theta} \right) x + \left(\frac{r}{\sin \theta} \right)$$
$$r = x \cos \theta + y \sin \theta$$

这种方法虽然和极坐标表示形式一样，但本质不一样，因为 r 为原点到直线的垂直距离，这种表示更为适合我们图像检测直线。

然后我们做一个图， x 轴为 θ ，假设每一度一个单位长度，一共 360° ；列为垂直距离 r ，从0到最大值（最大值为图片的对角线长度）。

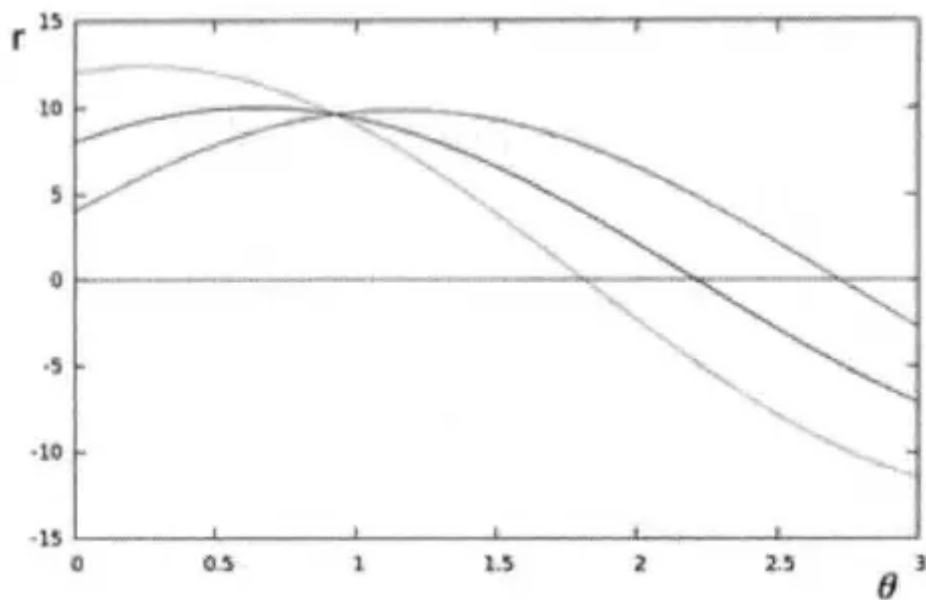
然后我们以某一像素点（8，6）为圆心，计算过改点的所有直线到原点的垂直距离 r ，然后画图如下：



需要注意，这个点不是图像上随意一个点，它应该是经过边缘检测之后凸显出来的边缘点，或者下图中白色直线中的点：

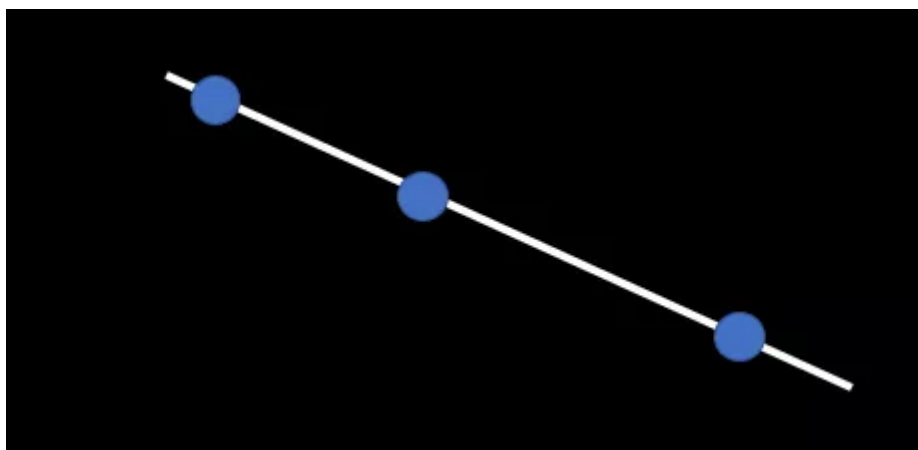


绘制出来为一正弦曲线，然后相同操作，再绘制两点（9，4） / （13，2）的曲线：



可以看到三条线交于一点，这个交点就是我们的所求，但它意义何在呢？

还是看下图：



假设我们取的三点为蓝色点，单独一点我们都无法确定其所在白色直线的特征 (θ, r) ，我们取三点，让他们对所有过自身的直线 (θ, r) 表示进行投票，三人均投的（也就是交点）自然就是白色直线。

实战处理

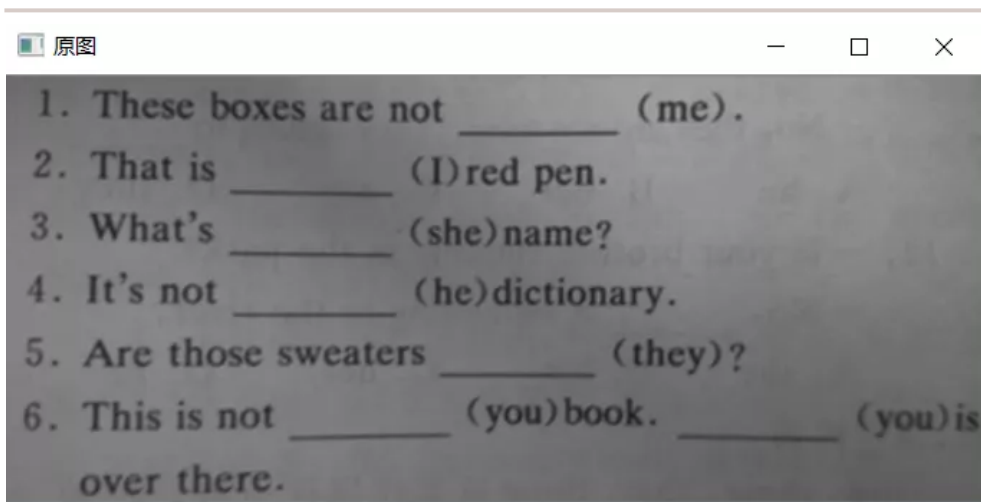
1. These boxes are not _____ (me).
2. That is _____ (I) red pen.
3. What's _____ (she) name?
4. It's not _____ (he) dictionary.
5. Are those sweaters _____ (they)?
6. This is not _____ (you) book. _____ (you) is over there.

然后我们来想如何对该图像进行预处理，然后再进行霍夫线变换效果会更好，自然是将那些字母啥的都去掉，仅留直线填空了。

1：读取图片的灰度图，搭建程序框架。

```
1  #include <iostream>
2  #include <opencv2/opencv.hpp>
3
4  using namespace std;
5  using namespace cv;
6
7  int main()
8  {
9      Mat srcImage, dstImage, binaryImage;
10     srcImage = imread("原图.png",0);
11     imshow("原图", srcImage);
12
13     waitKey(0);
14     return 0;
15 }
```

运行如下：



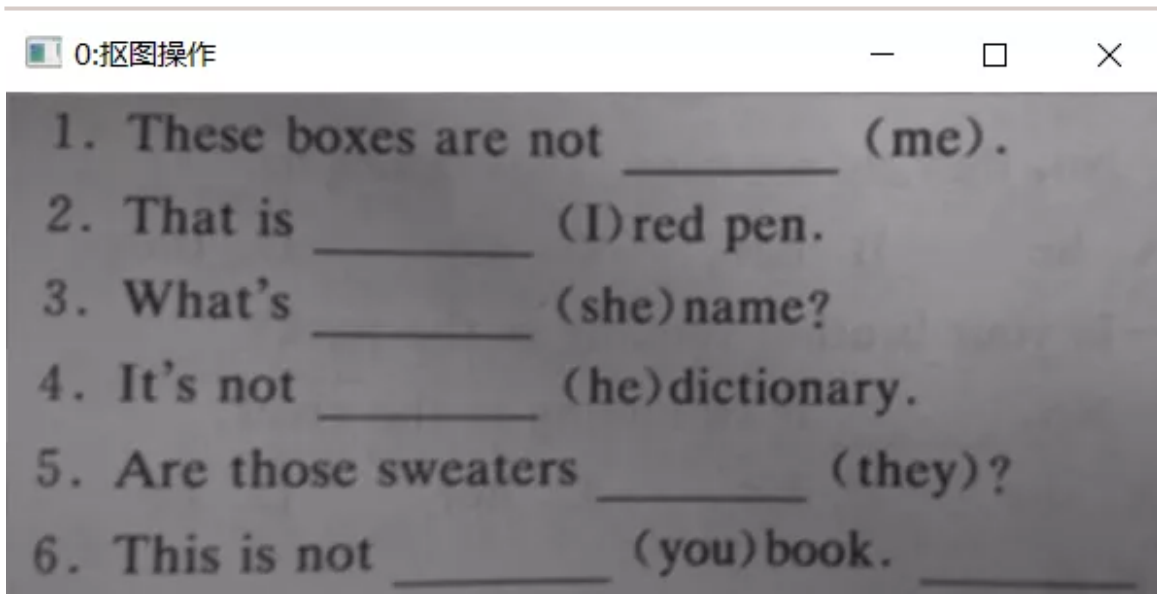
2：裁去多余部分

我们需要检测的是直线，所以图片的下边以及左边都可以裁去一部分，这样可以减少一些干扰。

```
1  //剪裁图片
2  Mat roiImage = srcImage(Rect(0, 0, srcImage.cols - 70, srcImage.rows - 30));
```

```
3 imshow("0:抠图操作", roiImage);
```

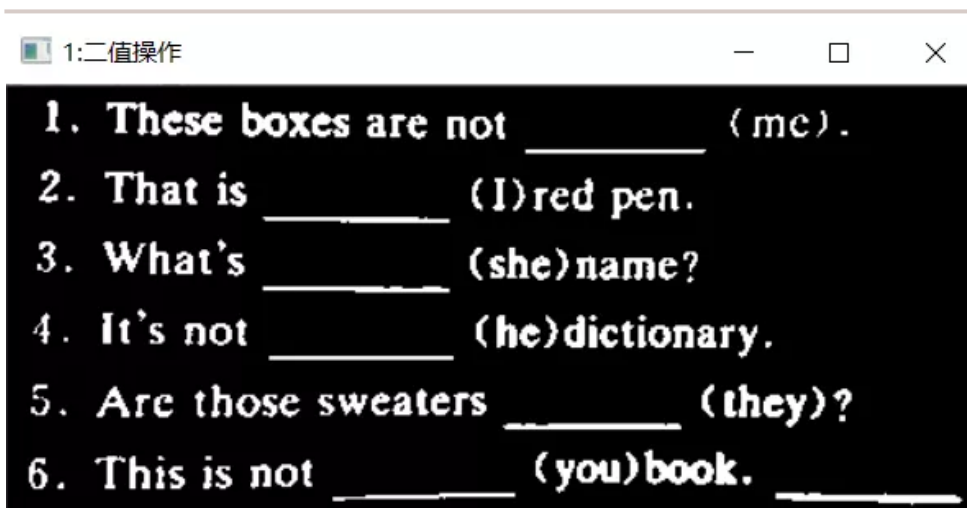
运行如下：



3：对图像进行二值化

```
1 //对图像进行二值化
2 threshold(roiImage, binaryImage, 92, 255, THRESH_BINARY_INV );
3 imshow("1:二值操作", binaryImage);
```

二值化是图像操作中最常用到的操作了吧，可以将我们的目标和背景进行分离。运行如下：



4：形态学开操作

```
1 Mat morphImage;
2 Mat kernel = getStructuringElement(MORPH_RECT, Size(20, 2), Point(-1, -1)); //
```

```
3 morphologyEx(binaryImage, morhpImage, MORPH_OPEN, kernel, Point(-1, -1));//开
4 imshow("2:开操作", morhpImage);
```

这一步就是我们整个处理最核心的部分了，他的用意在于利用先腐蚀后膨胀的原理将文字这些密集的“孔洞”给腐蚀掉，仅留下直线。

我们可以将上图颜色颠倒，然后执行上述操作如下：



这样就明白了吧。

还有一个重点在于我们使用了一个自定义的20*2大小的矩形核，这个核可以更为准确的处理长条状的文字区域。

实际我们代码运行结果：



4: 然后再膨胀加粗一下、

```
1 Mat dilateImage;
```

```
2 kernel = getStructuringElement(MORPH_RECT, Size(3, 3), Point(-1, -1));
3 dilate(morphImage, dilateImage, kernel);
4 imshow("3:膨胀操作", dilateImage);
```

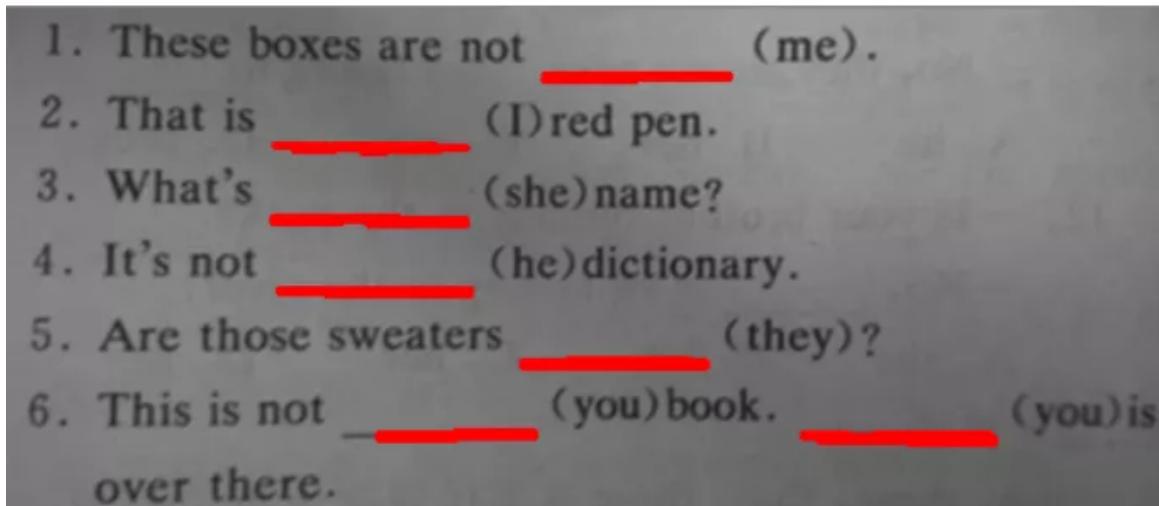
运行如下：



5: 最后就是霍夫线变换了

```
1 vector<Vec4i> lines;
2 HoughLinesP(dilateImage, lines, 1, CV_PI / 180.0, 30, 20.0, 0);
3 dstImage = srcImage.clone();
4 cvtColor(dstImage, dstImage, COLOR_GRAY2BGR);
5 for (size_t t = 0; t < lines.size(); t++) {
6     Vec4i ln = lines[t];
7     line(dstImage, Point(ln[0], ln[1]), Point(ln[2], ln[3]), Scalar(0, 0, 255),
8 }
9 imshow("4:绘制直线", dstImage);
```

非常常规的代码，运行结果：

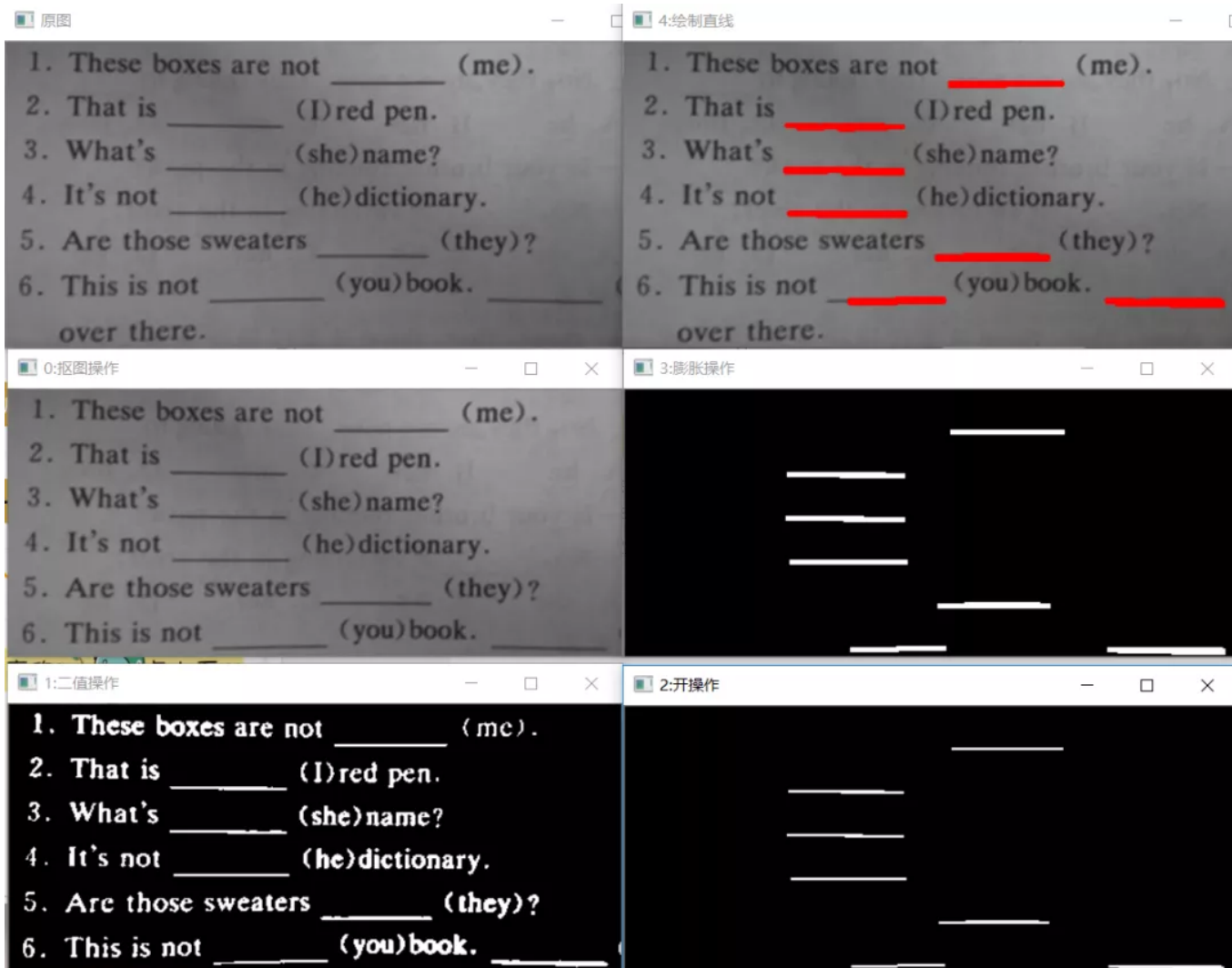


是不是完美？贾老师nb。我听过几个贾志刚老师的课，学cv的小白们应该也都知道他，别的课怎么样我就不说了，这个小案例实战课真的很好，而且我也一直在更新【opencv小实践】系列，所以就拿来跟大家分享一下。

一共六个案例，这是第一个。

回顾

所以霍夫线变换直接用是不可能百试不爽的，而图像处理最难的地方也从来不是调用函数API，而是图像的预处理，如何对图像预处理，是它具有很好的普适性能，很见功底。



THE END

下期再见啦~希望大家可以帮忙转发分享~



一个不误正业的理科男

长按关注微信公众号【行走的机械人】

分享我在电控，计算机视觉的学习所得
以及不定期资源分享



点个在看，就可以评论啦~



People who liked this content also liked

我的小破网站也上线了！

周旋机器视觉