



基于机器视觉的缸套表面环状字符检测系统研究

答辩人：张杰

导师：陈甦欣

日期：2020/6/9



论文主体结构

摘要

第一章 绪论

- 1.1 课题研究背景与意义
- 1.2 国内外研究现状
- 1.3 本文研究的内容和组织结构

第二章 视觉检测整体设计

- 2.1 检测系统需求分析
- 2.2 视觉检测总体框架
- 2.3 开发平台和算法库介绍

第三章 图像预处理与特征信息提取

- 3.1 图像预处理
- 3.2 特征信息提取
- 3.3 本章小结

第四章 字符识别算法研究

- 4.1 传统字符识别算法
- 4.2 基于深度学习的识别算法
- 4.3 本章小结

第五章 设备通讯和软件结构分析

- 5.1 OPC UA通讯原理
- 5.2 可视化界面的实现
- 5.3 软件功能实验
- 5.4本章小结

第六章 总结与展望



第一章 绪论

研究背景及意义

- 1.国家大力推进工业4.0，是中国进军世界高端制造梯队的一个重要契机。
- 2.传统检测逐渐无法适应当前的生产要求，视觉检测的非接触性、精确性等优点正在凸显。
- 3.近年来基于人工神经网络的深度学习网络模型在目标识别，特征提取，智能预测等方面有着广泛的应用并且有着良好的效果。
- 4.OPC UA(Unified Architecture，统一架构)通讯架构的提出使设备之间的无障碍交流变为可能。



第一章 绪论

国内外研究现状

1.视觉检测：视觉检测在工业零件型号识别方面大多应用于机器人分拣系统，零件装配检测，在零件质量检测方面有齿轮检测等；农业上运用于农产品的分类质量识别；在医学上提出视觉检测提高病人心率测量精度。

2.字符识别：传统方法有模板匹配、支持向量机(SVM)、针对不规则字符还有基于神经网络的深度学习识别方法。如今，在深度学习领域，有针对检测水平或微斜的文本行的CTPN模型，针对可识别较长的文本序列的CRNN模型，还有直接从图片中定位识别字符的FOTS模型等。这些模型在各自的领域都有着不错的识别效果。



第二章 视觉检测整体设计

检测目标:气缸缸套表面字符, 缸套共分为A, B两种, 同种型号间的字符位置分布不同, 不同型号间的字符大小也不相同

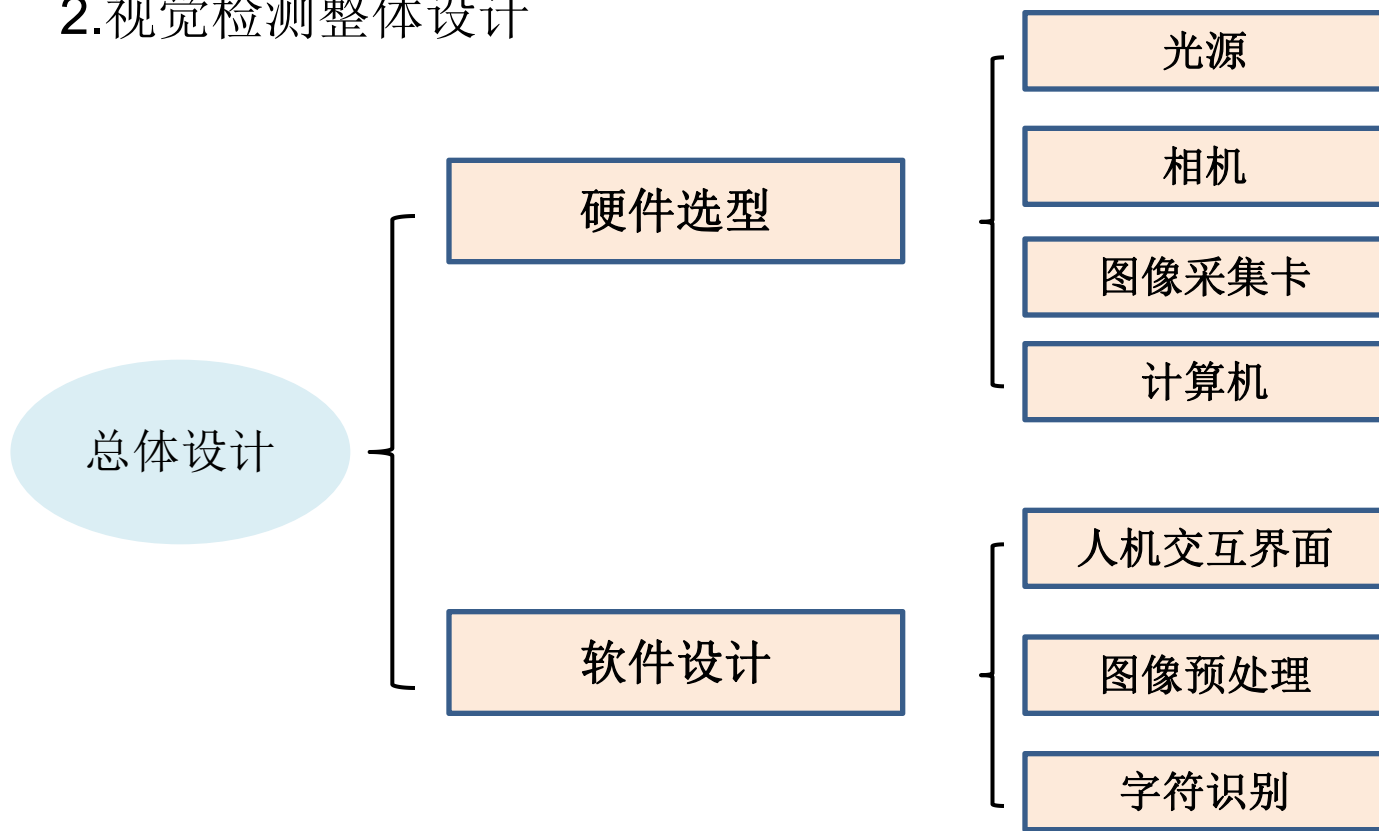
o





第二章 视觉检测整体设计

2.视觉检测整体设计





第三章 图像预处理与字符提取

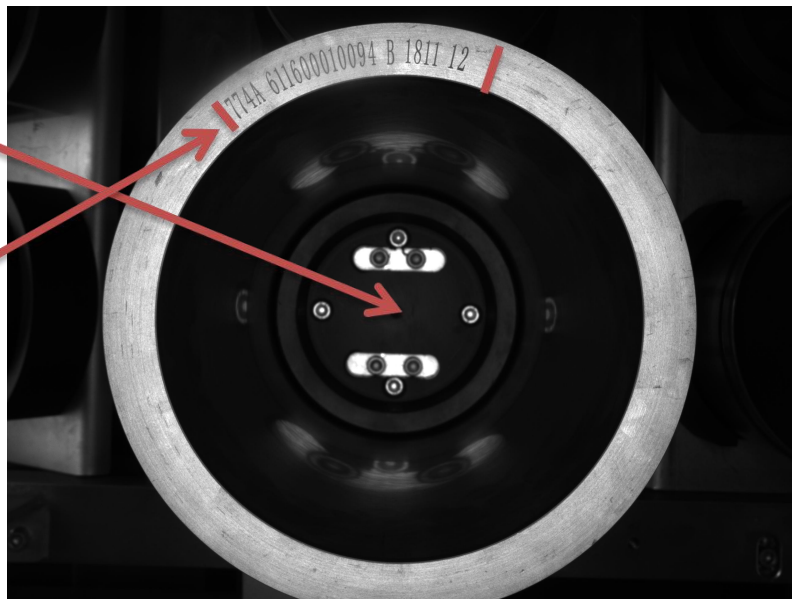
零件定位和字符展开：先确定零件的圆心和展开范围，再通过像素间的极坐标转换公式展开环状分布字符。

问题？

零件中心如何确定

分割起始角度如何确定

如何提高展开速度





第三章 图像预处理与字符提取

1.零件中心的确定：霍夫圆变换可以方便检测出图像中的圆。但是需要测试不同参数的检测效果，只要检测目标和参数稍有变化，检测效果大不相同，在实际应用中鲁棒性较差。

param1=50, param2=100 (205.047ms)



param1=60, param2=90 (193.044ms)



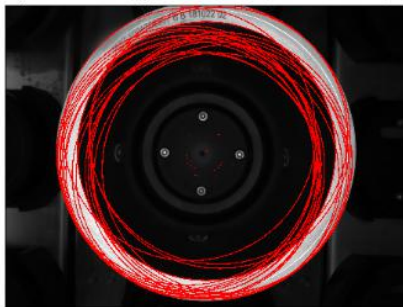
param1=70, param2=80 (185.042ms)



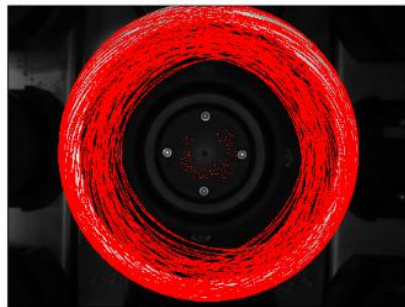
param1=50, param2=90 (231.052ms)



param1=50, param2=80 (209.047ms)



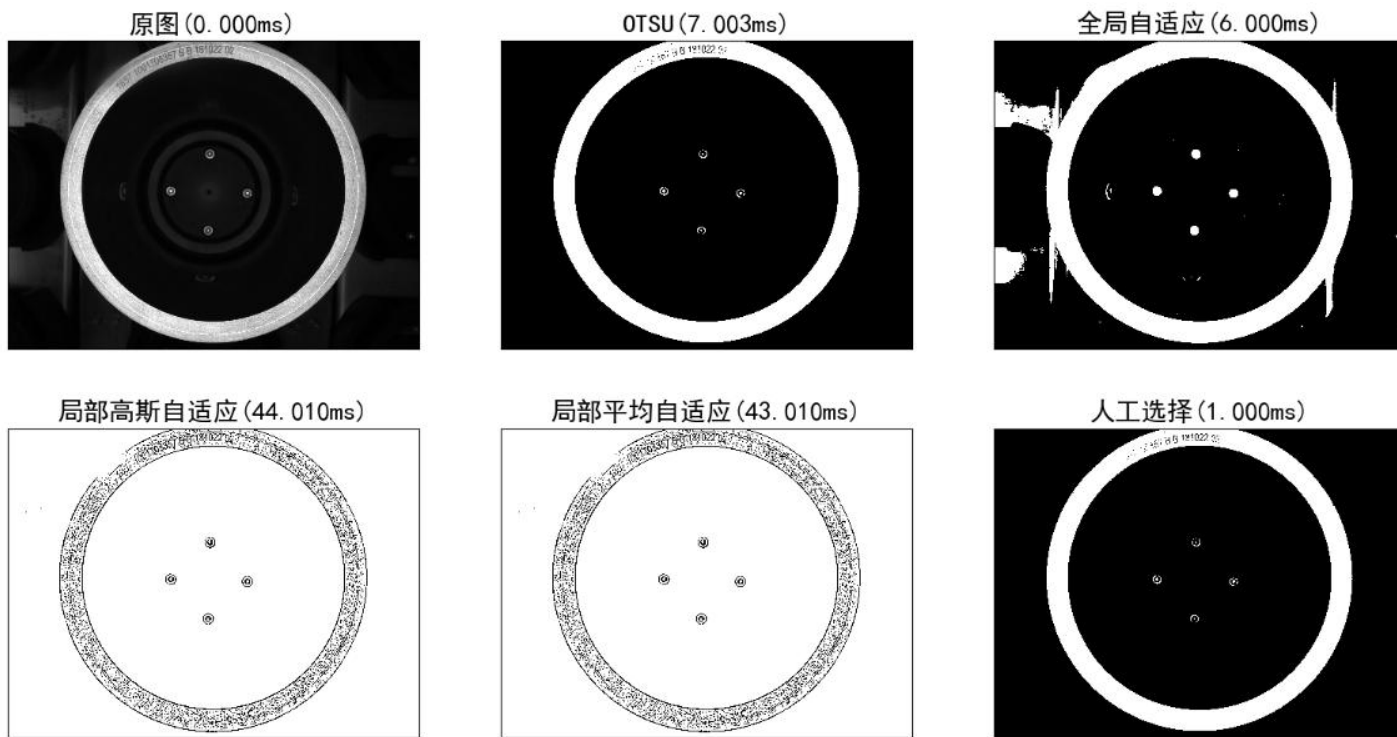
param1=50, param2=70 (288.066ms)





第三章 图像预处理与字符提取

进行二值化处理可以将前景背景分离，同时使图像内元素减少，方便后面信息提取。





第三章 图像预处理与字符提取

面积法是先对处理后的低噪点二值图进行边界查找，根据先验知识，对每个封闭边界的面积进行计算比较得到零件内圈的最大边界。与Hough圆检测方法相比，面积法原理简单，没有太多的输入参数，鲁棒性较强。

面积法 (39.008ms)



Hough圆检测法 (201.046ms)

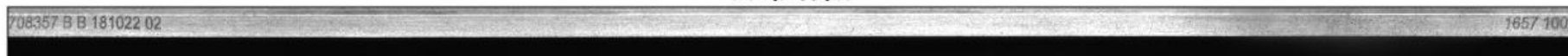




第三章 图像预处理与字符提取

2. 展开起始角度和展开范围: 传统图像算法都是针对矩形分布的字符, 需要将原图展开为矩形区域。由于字符分布的随机性, 必须在每次展开时先确定字符的在环形表面的分布情况。

异常切割



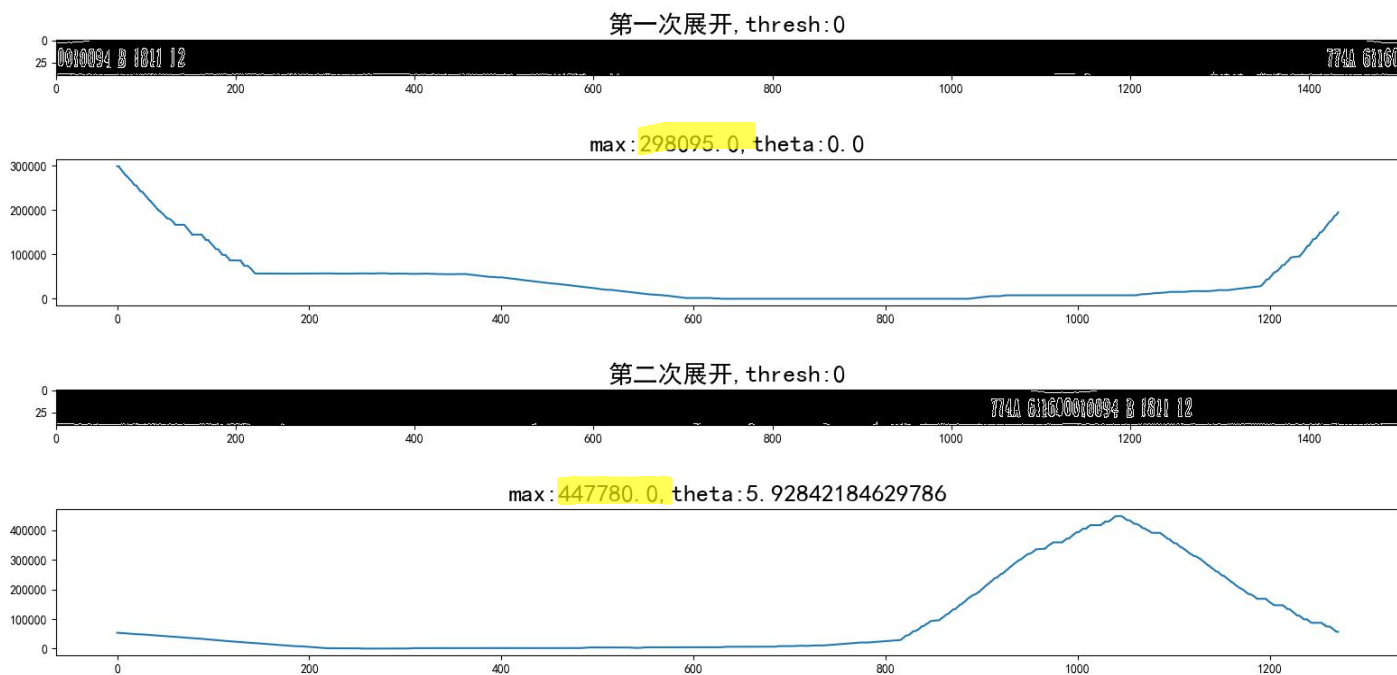
正常切割





第三章 图像预处理与字符提取

3.提高展开效率：采用在缩小图上寻找字符展开角度和半径，得到分割参数后再对原图进行展开运算。通过使用特定大小的卷积核卷积运算，可以得到如下字符分布图。





第三章 图像预处理与字符提取

利用提取到的展开角度对原图进行展开运算：

两种方法耗时详情（单位：ms）

方法	缩小	两次展开	两次卷积	有参数展开	总耗时
直接展开	0	1974	1264	230	3478
缩小展开	5	123	79	230	437

极坐标变换公式

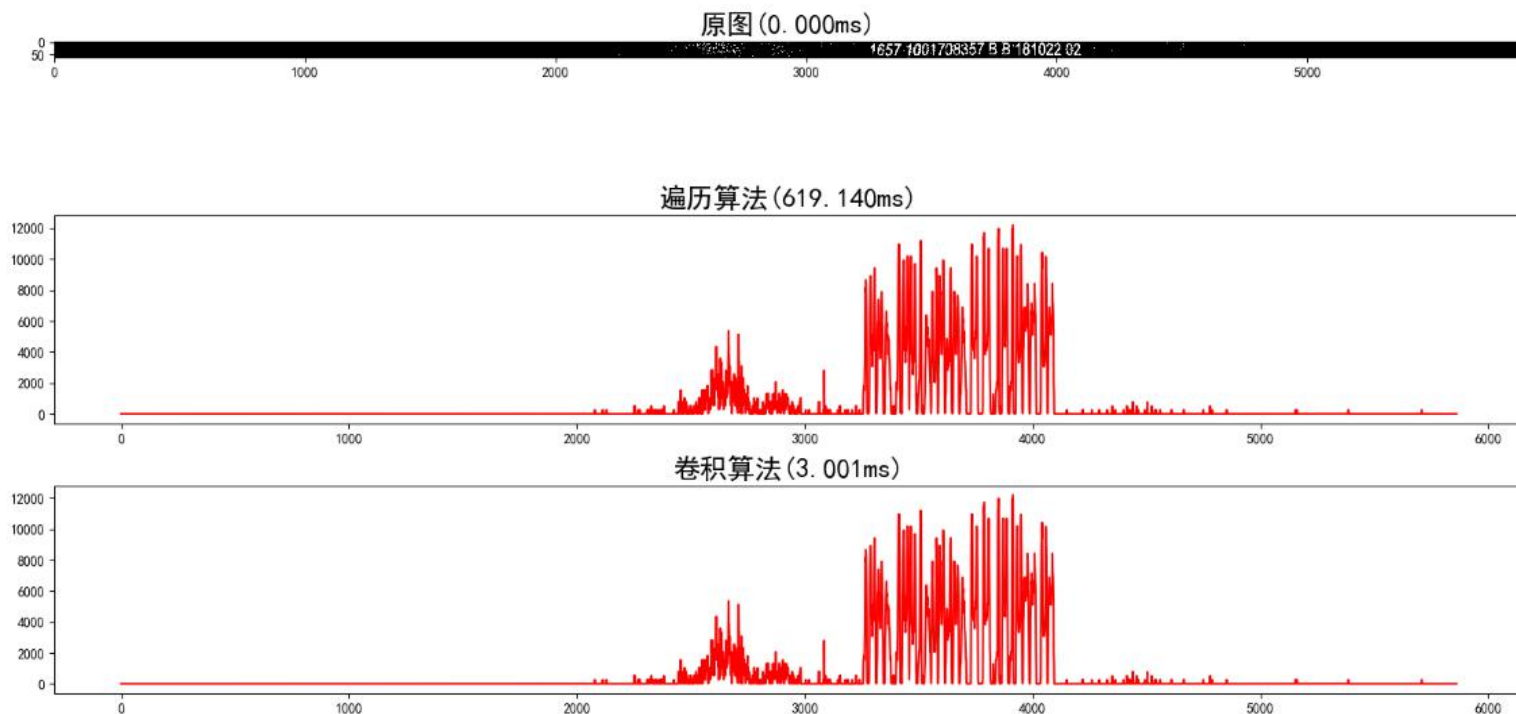
$$\begin{cases} x = r \times \cos \theta + x_0 \\ y = r \times \sin \theta + y_0 \\ r = h - Y \\ \theta = 2\pi \frac{X}{w} \end{cases}$$





第三章 图像预处理与字符提取

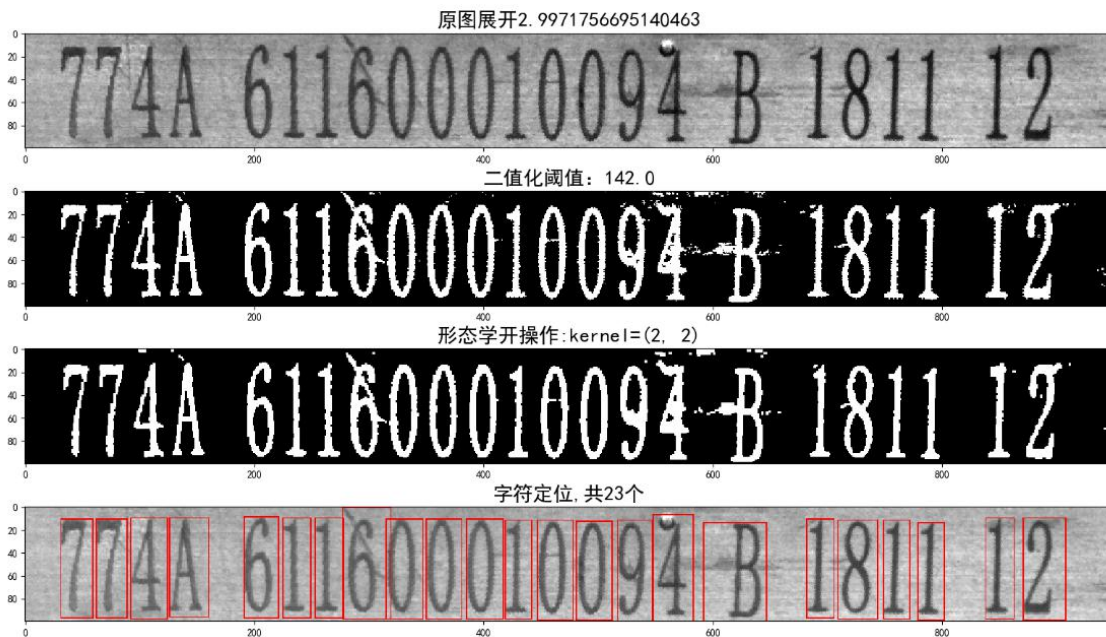
像素投影法是对某一图片的整个或局部区域进行像素遍历，计算出横向或者纵向的像素和。通过特定形状的卷积和对原图进行卷积运算达到提取字符分布的效果。





第三章 图像预处理与字符提取

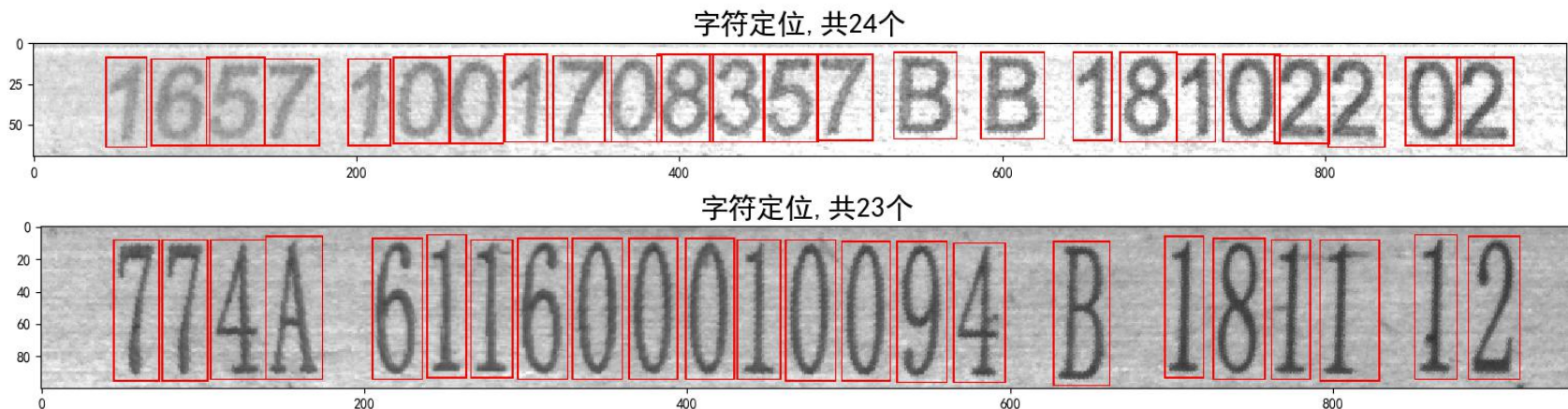
字符提取运用比较广泛的有投影法和连通域法。其中投影法适用于字符排列整齐，中间无粘连情况。本次识别字符偶有粘连，使用连通域法，在二值化，开操作后的可以取得比较好的字符提取效果。





第四章 字符识别算法研究

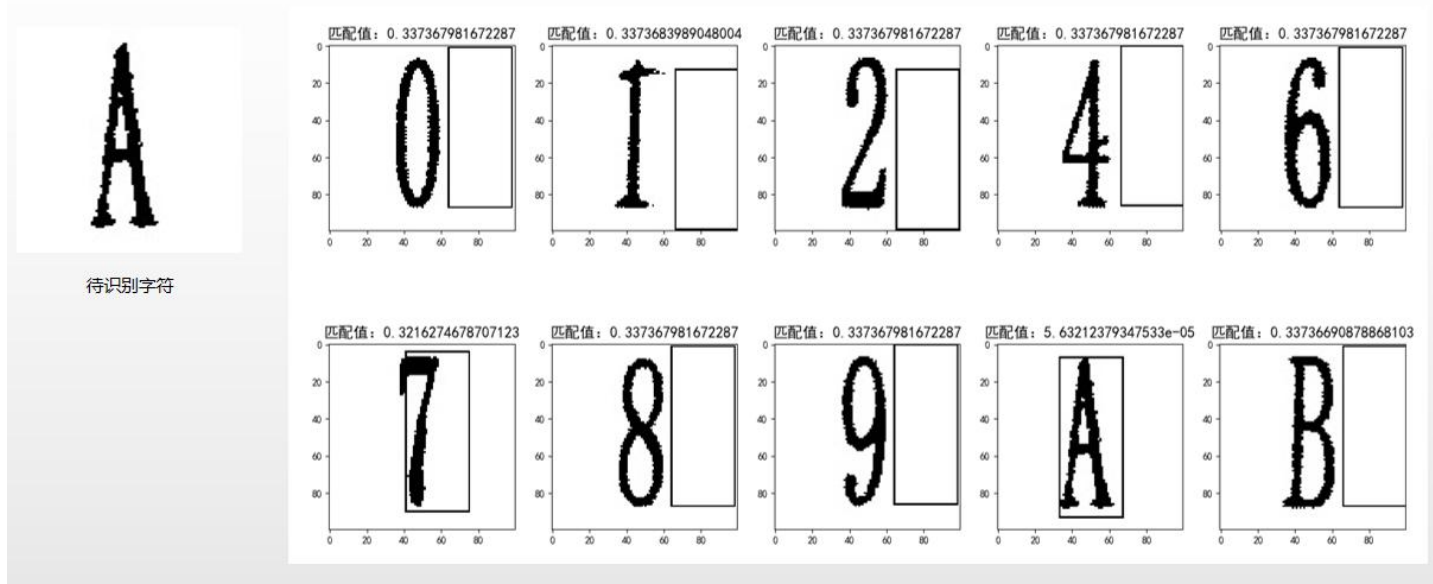
字符识别需求：待识别型号的工件共有**23-24**个字符，每个字符由数字和英文字母混合排列而成，且每种型号零件字符规格不同。字符识别常用方法有模板匹配、支持向量机(SVM)、深度学习等。





第四章 字符识别算法研究

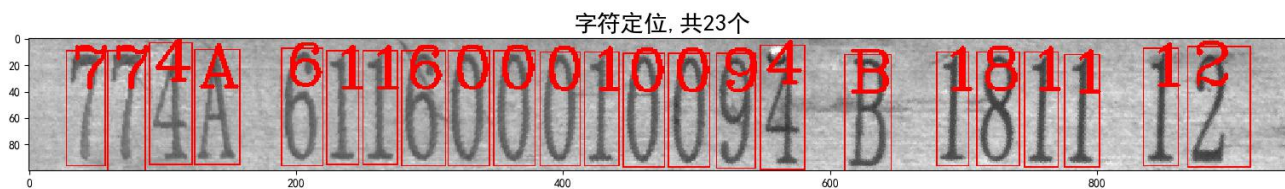
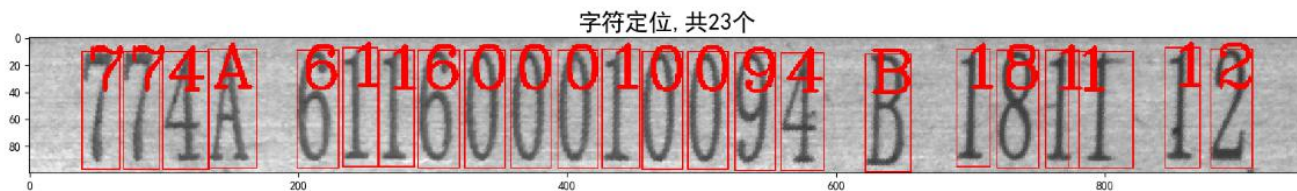
模板匹配：模板匹配通过将匹配目标图像窗口与包含匹配目标的图像进行滑动卷积运算，取匹配数值最大区域为目标位置。为了降低后期处理的难度和匹配的准确性，将模板和待匹配的图片统一二值化处理。





第四章 字符识别算法研究

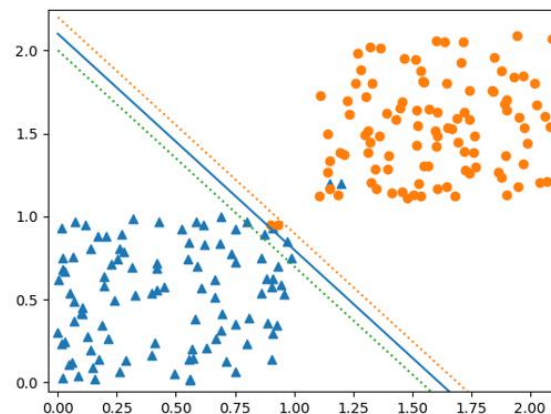
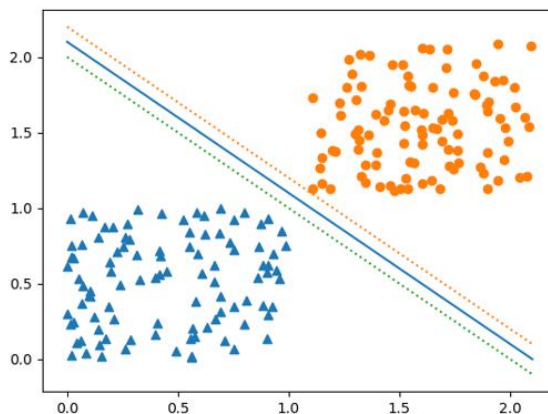
模板匹配结果:



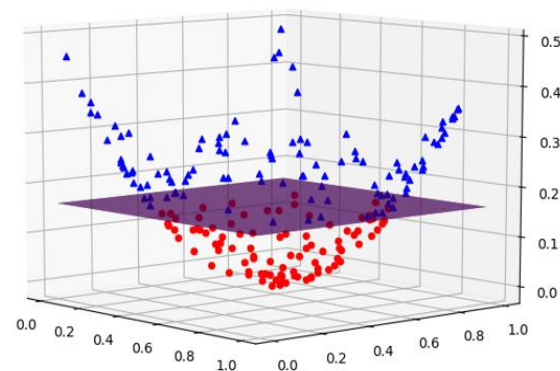
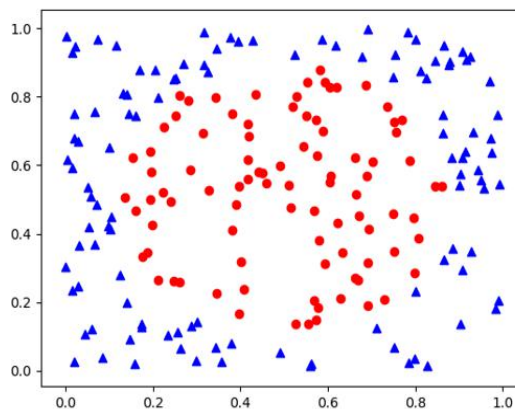


第四章 字符识别算法研究

SVM在小样本、高特征维度、非线性分布的数据分类上有着优异的分类表现。



$$\begin{cases} x = x \\ y = y \\ z = (x - 0.5)^2 + (y - 0.5)^2 \end{cases}$$



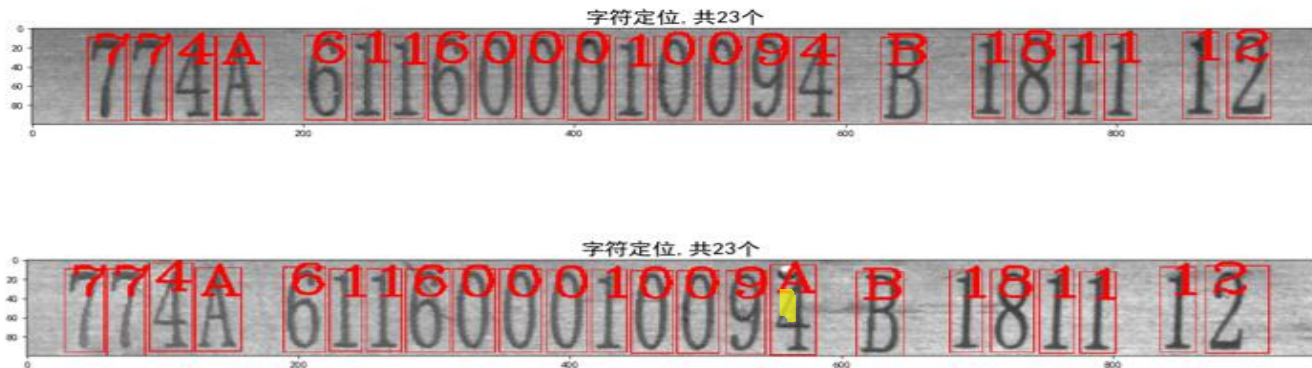
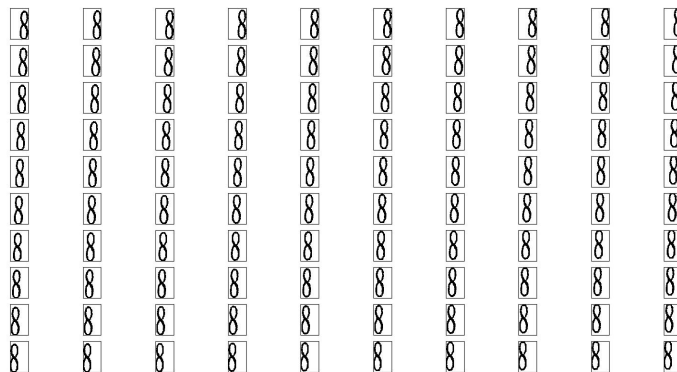


第四章 字符识别算法研究

先将原样本做几次横坐标和纵坐标平移分割，增大样本容量，再将图片和标签传入方法后得到模型训练结果。

SVM模型参数

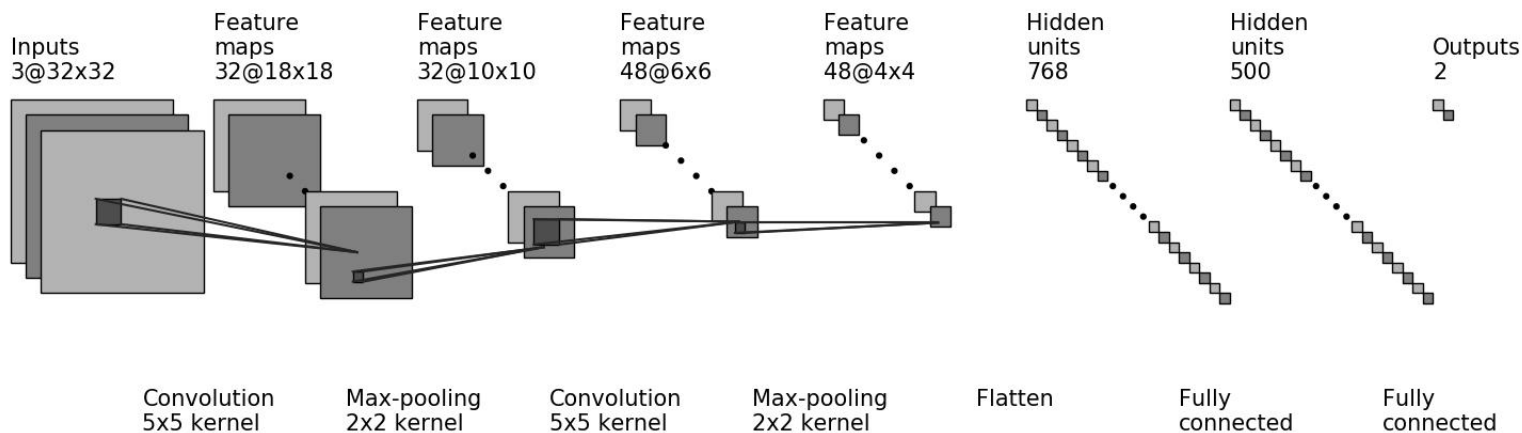
核函数	松弛因子	训练样本数	测试样本数	训练集准确率	测试集准确率
linear	0.80	880	220	100%	99.3%





第四章 字符识别算法研究

深度学习是机器学习中一种基于对数据进行特征学习、能够模拟出人脑神经结构的机器学习方法。**CNN(Convolutional Neural Network,卷积神经网络)**单个字符的手写字符识别率可达99.7%。





第四章 字符识别算法研究

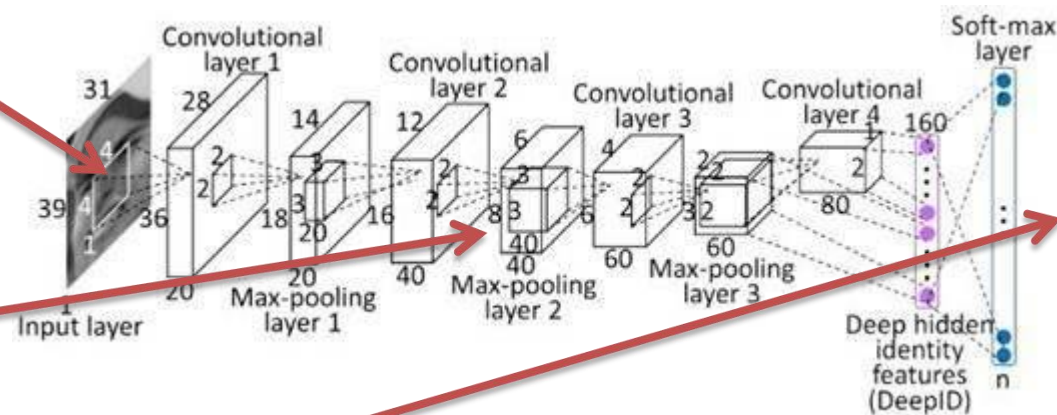
深度学习是机器学习需要大量标注数据进行模型训练，在搭建好的模型上，使用归一化的带标签的数据集进行训练。

问题？

训练数据如何制取

识别模型如何搭建

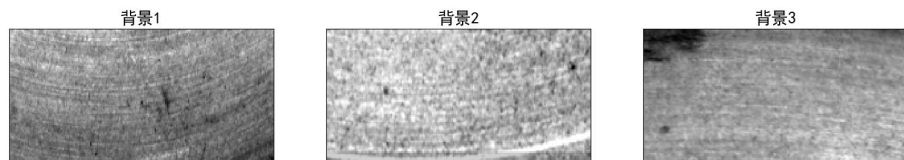
如何提高识别精度



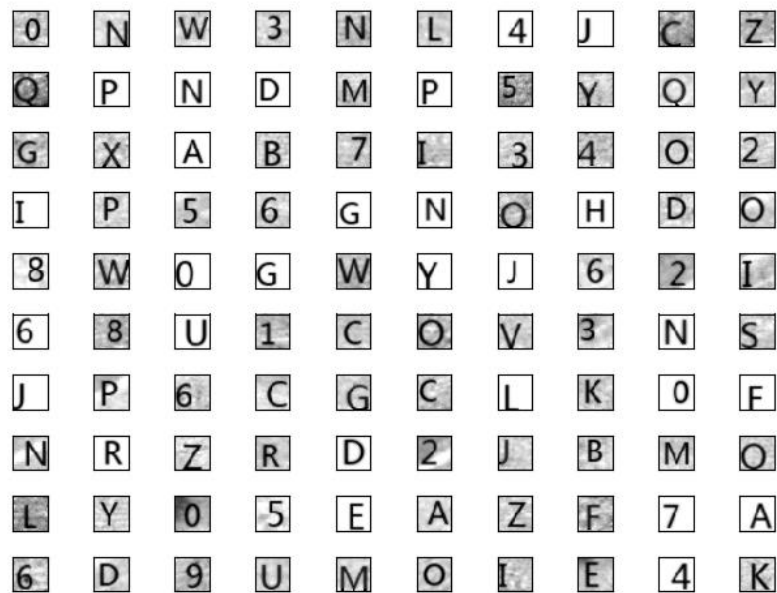


第四章 字符识别算法研究

数据制作：数据量大小和质量将直接影响深度学习的性能。主要包括背景选取，字体大小、类型、颜色的确定，字符贴合位置的确定。其中既要实现字体生成状态的随机性，又要保证不同字体都将处于文字区域。



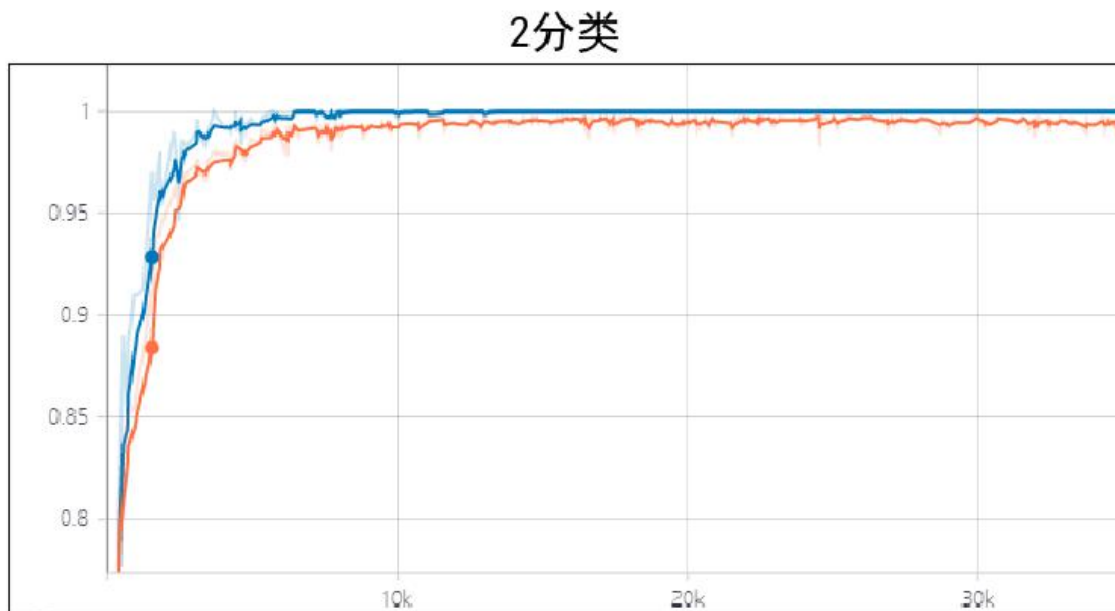
不考虑字符间差异





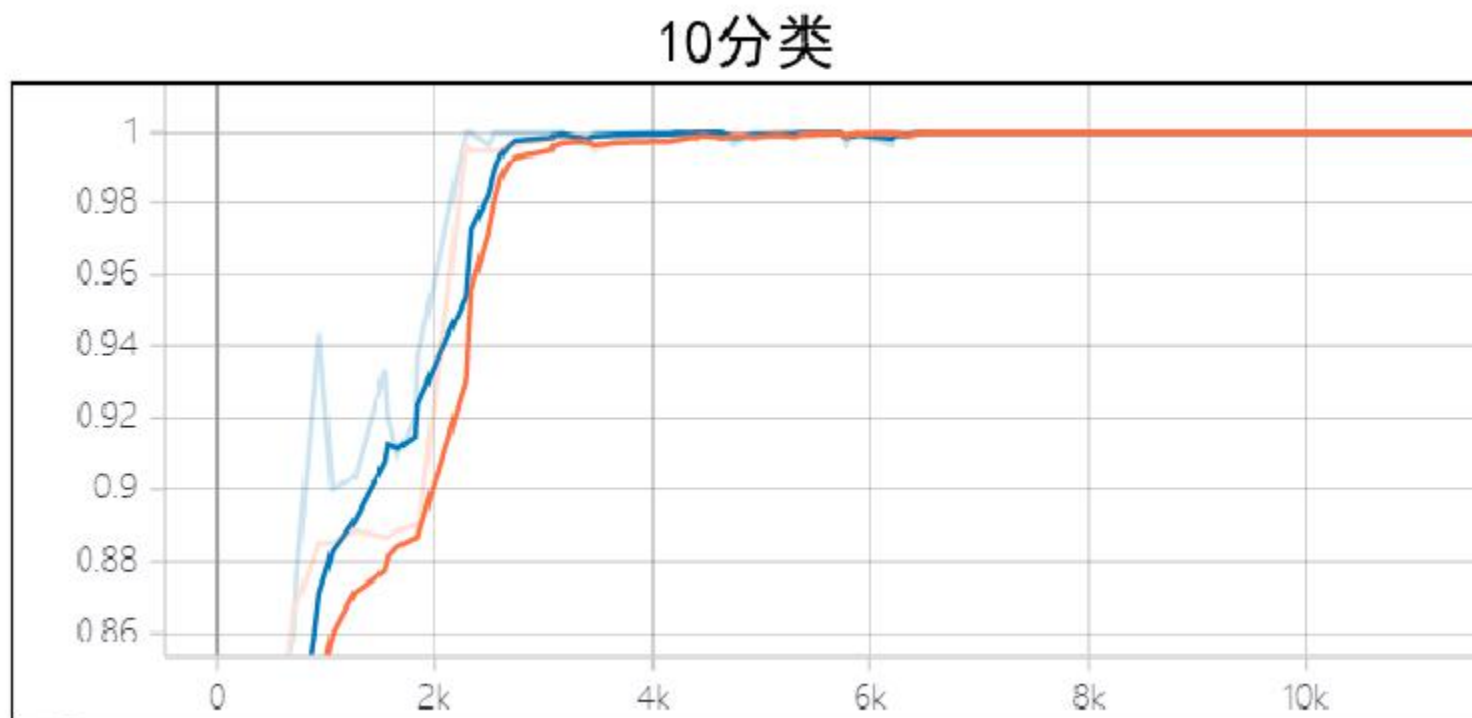
第四章 字符识别算法研究

模型搭建：使用google提供的tensorflow框架进行模型搭建。发现先将字符进行数字英文二分类，然后再对二者各自进行分类比直接进行36分类有着更高的准确率。





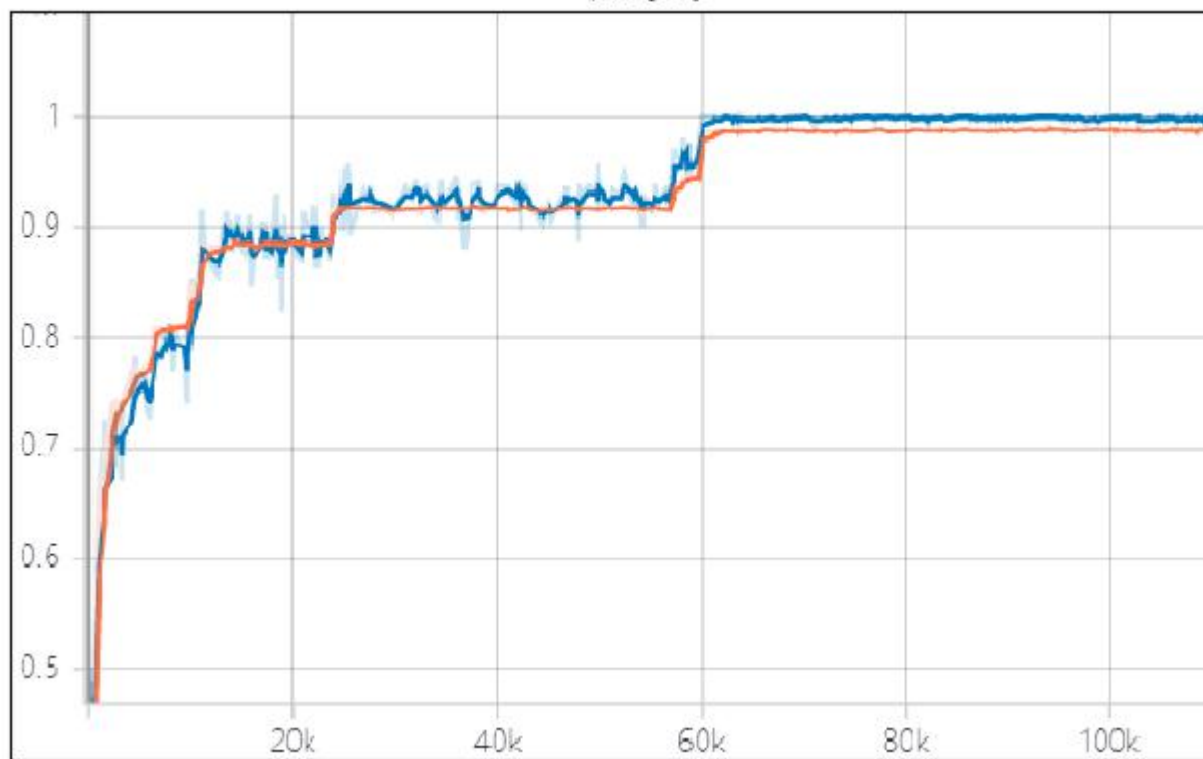
第四章 字符识别算法研究





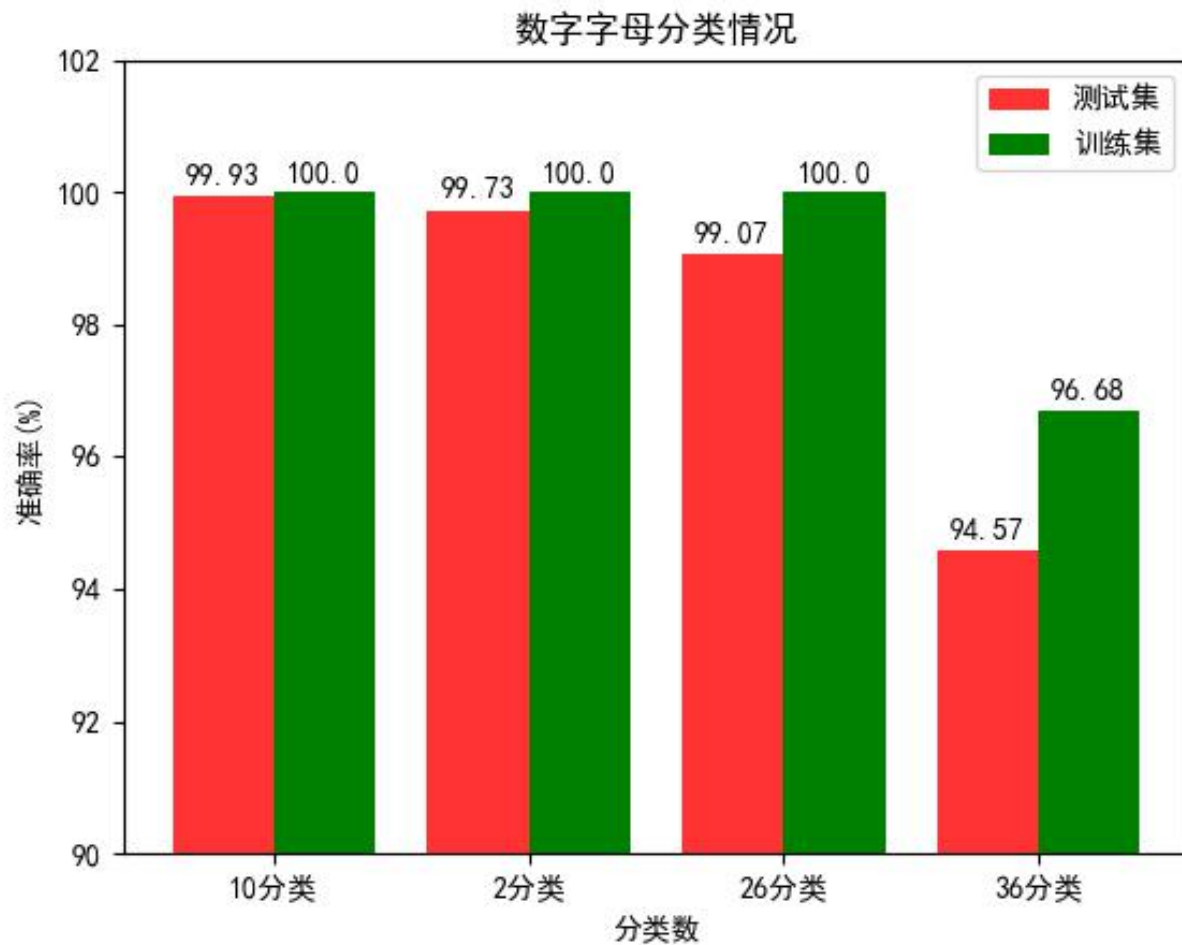
第四章 字符识别算法研究

26分类





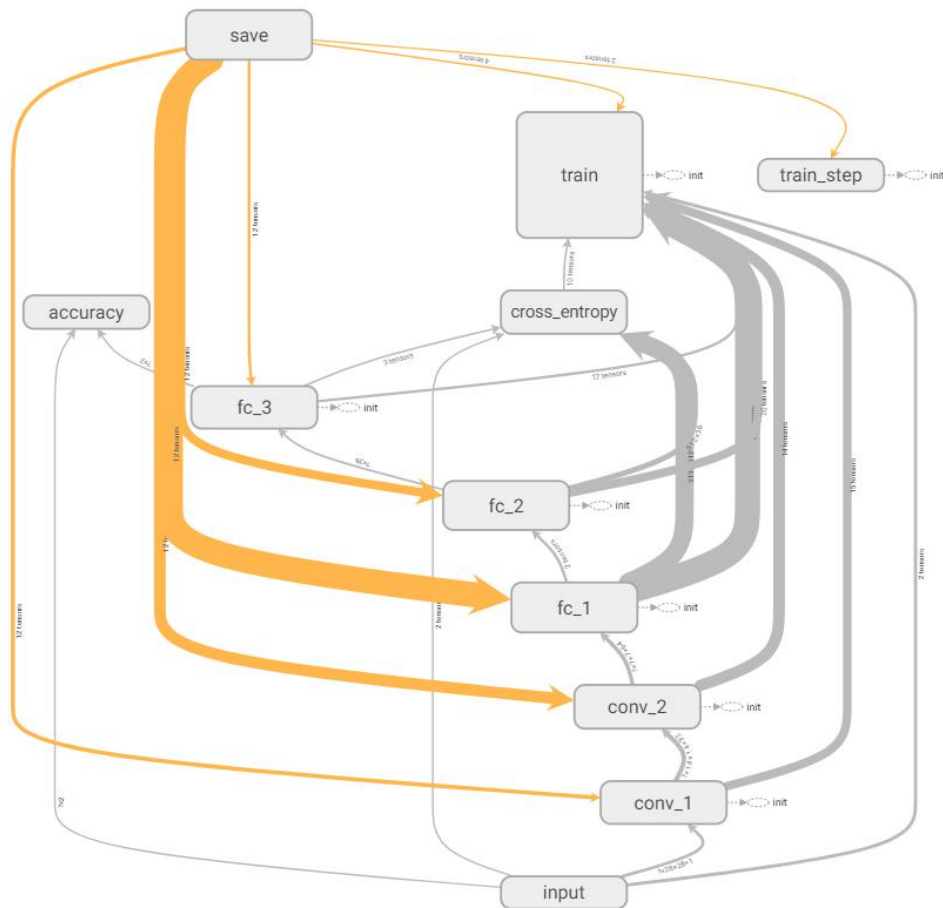
第四章 字符识别算法研究





第四章 字符识别算法研究

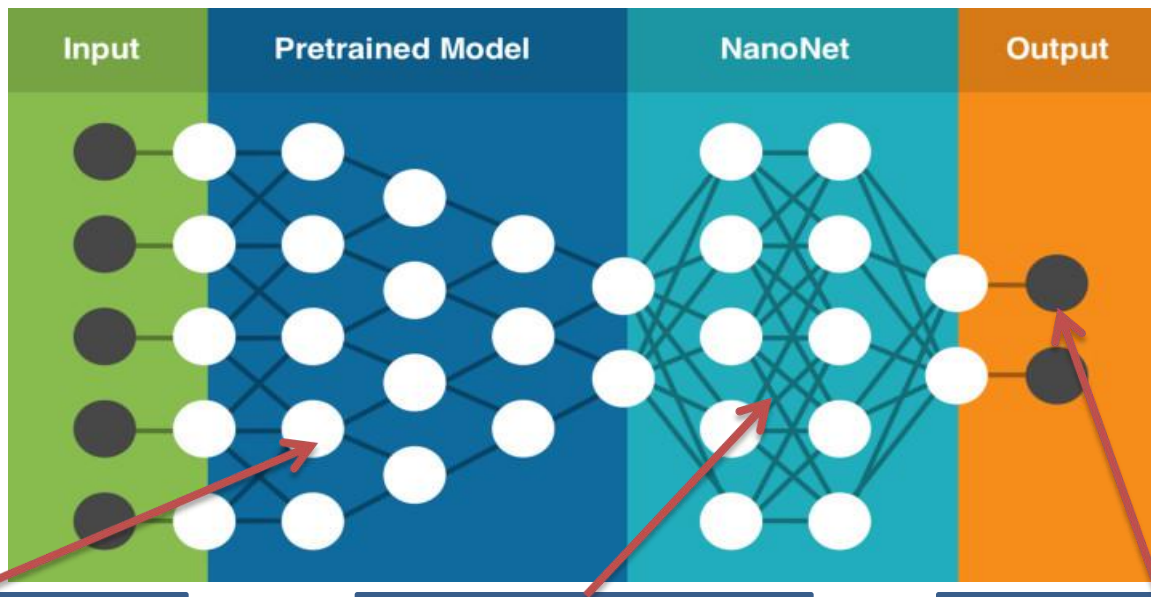
模型结构：三个分类模型均采用卷积神经网络提取目标特征，全连接层分类目标。三个分类器均由两个卷积层和三个全连接层组成，但是其中的卷积核，全连接层矩阵大小根据需要有所不同。





第四章 字符识别算法研究

模型迁移：**Pretrained Model**为预先训练好的提取特征的模型，在不同数据集上使用模型时，会在现有预训练模型上创建一个新的**NanoNet**进行训练分类。



预训练层

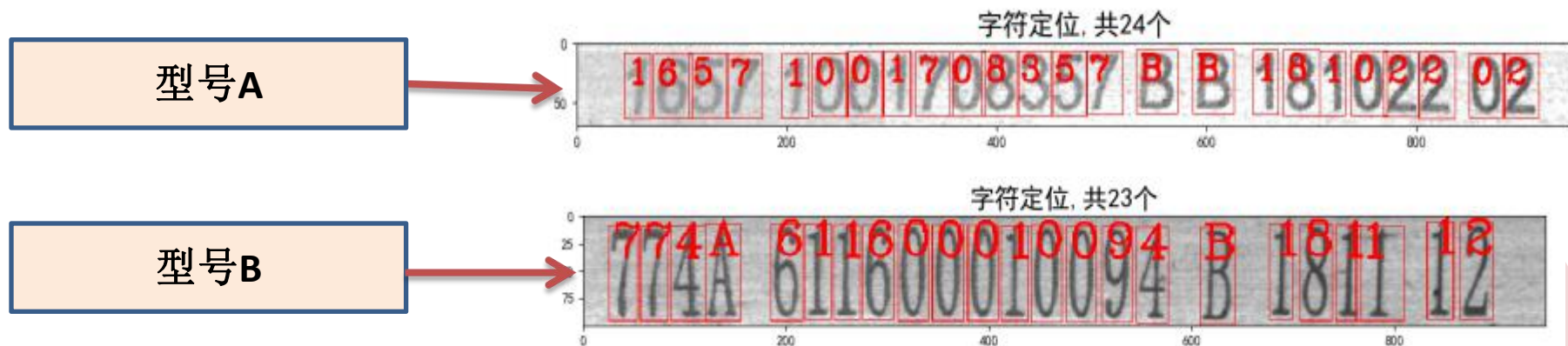
分类层

输出层



第四章 字符识别算法研究

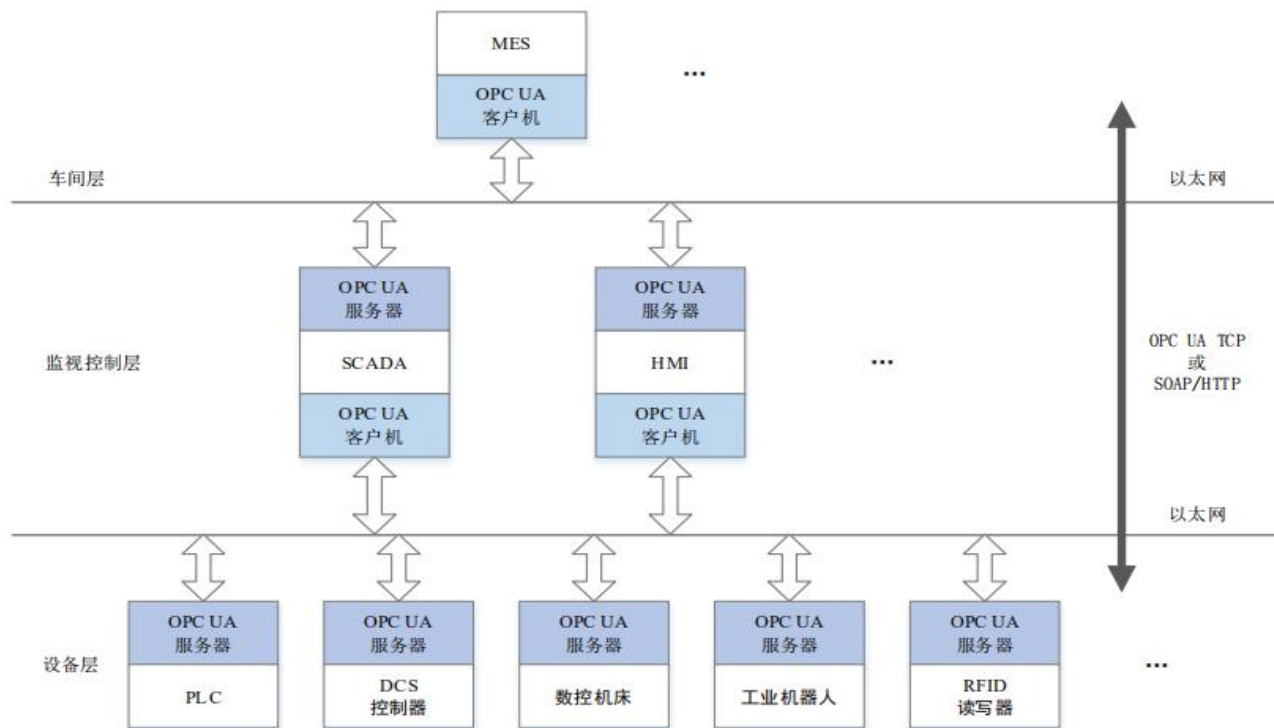
预测效果图：由图，深度学习识别算法可以准确识别**AB**两种不同型号的字符，同时在型号**A**的采光条件较差的情况下，识别精度仍然很高。可以得到深度学习算法准确率和泛化性高于模板匹配和**SVM**算法。





第五章 设备通讯和软件设计

OPC UA结构：车间的设备层内部嵌入OPC UA服务器，为上位机和SCADA系统提供设备运行时的数据。通过以太网端口与车间的MES层连接，实现设备层提供数据、监控层收集数据、车间层汇总数据的信息化生产过程。此外，设备层、监控层、车间层之间的信息交换均可通过OPC UA实现。





功能模块

信息交互

识别步骤

连接OPC 关闭OPC 打开文件 开始识别

识别算法
☒ 模板匹配 ☐ SVM ☐ CNN

交互信息

圆心为:(1000, 1260),半径为:796
template matching识别结束,共23个字符,结果为
两次金字塔缩小处理完成
二值化处理完成,阈值为: 94.0
圆心:(250, 315),半径: 199
两次展开完成,第一次角度: 4.48, max:446760.0,第
圆心为:(1000, 1260),半径为:796
template matching识别结束,共23个字符,结果为

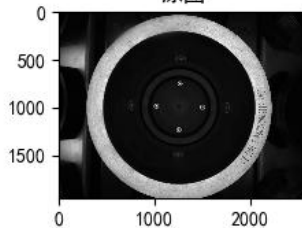
识别结果

字符个数	识别时间	识别结果
1 23	2119.924...	774A611600010094B18111
2 23	1329.182...	774A611600010094B18111

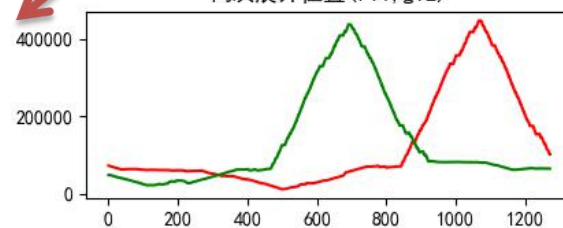


x=617.261 y=122483

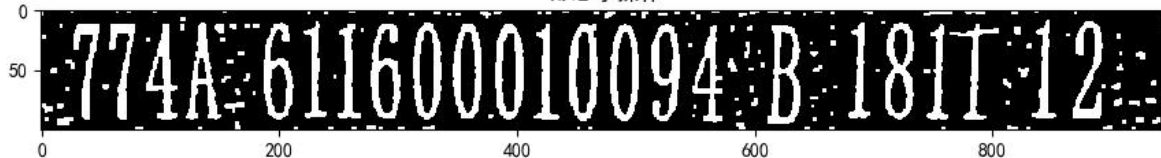
原图



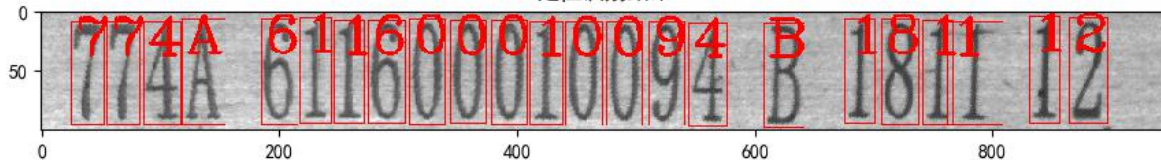
两次展开位置 (r:1, g:2)



形态学操作



定位识别结果





第六章 总结与展望

01 总结



在缩小图像中寻找字符分布特征，大大提升了环状分布字符定位、拉伸、分割的速度。使用卷积算法实现识别区域的精确定位，连通域法实现字符的提取。



基于深度学习建立一套高鲁棒性的字符识别算法。从训练数据的制作、模型搭建、模型训练一直到识别结果分析，都作了详细讨论。



使用Python作为界面编写语言、Qt作为界面设计的框架、OPC UA为设备通讯工具，实现了程序的多平台统一。

02 展望

本文研究证实深度学习与视觉检测结合的优越性，下一步将要识别速度快，定位精度更高，可以识别多个连体字符的检测方法。



谢 谢！
请各位老师批评指正！