一、OCR的主流模型和算法，开源的代码和API

1、科大讯飞OCR文字识别API接口

百度OCR文字识别API接口

云脉OCR文字识别API接口

（http://www.yunmai.com/ocr\_sdks.html）

合合信息OCR文字识别API接口

(http://www.ccint.com/ocr/idcard.shtml)

ABBYY OCR SDK

（https://www.abbyy.cn/promo/ocr-sdk/?utm\_source=Baidu&utm\_medium=AdWords&utm\_campaign=FRE\_OCR&audience=163000）

阿里云OCR文字识别API

（https://market.aliyun.com/aliyunocrnew）

BIT OCR文字识别API

（http://www.bitocr.com/）

****2、OCR软件主要是由下面几个部分组成。****

　　1. 图像输入、预处理：   
　　2. 图像输入：对于不同的图像格式，有着不同的存储格式，不同的压缩方式。预处理：主要包括二值化，噪声去除，倾斜较正等   
　　3. 二值化：   
　　对摄像头拍摄的图片，大多数是彩色图像，彩色图像所含信息量巨大，对于图片的内容，我们可以简单的分为前景与背景，为了让计算机更快的，更好的识别文字，我们需要先对彩色图进行处理，使图片只前景信息与背景信息，可以简单的定义前景信息为黑色，背景信息为白色，这就是二值化图了。   
　　4. 噪声去除：   
　　对于不同的文档，我们对燥声的定义可以不同，根据燥声的特征进行去燥，就叫做噪声去除   
　　5. 倾斜较正：   
　　由于一般用户，在拍照文档时，都比较随意，因此拍照出来的图片不可避免的产生倾斜，这就需要文字识别软件进行较正。   
　　6. 版面分析：   
　　将文档图片分段落，分行的过程就叫做版面分析，由于实际文档的多样性，复杂性，因此，目前还没有一个固定的，最优的切割模型。   
　　7. 字符切割：   
　　由于拍照条件的限制，经常造成字符粘连，断笔，因此极大限制了识别系统的性能，这就需要文字识别软件有字符切割功能。   
　　8. 字符识别：   
　　这一研究，已经是很早的事情了，比较早有模板匹配，后来以特征提取为主，由于文字的位移，笔画的粗细，断笔，粘连，旋转等因素的影响，极大影响特征的提取的难度。   
　　9. 版面恢复：   
　　人们希望识别后的文字，仍然像原文档图片那样排列着，段落不变，位置不变，顺序不变，的输出到word文档,pdf文档等，这一过程就叫做版面恢复。   
　　10. 后处理、校对:   
　　根据特定的语言上下文的关系，对识别结果进行较正，就是后处理。

****开源引擎****

[Cuneiform for Linux](https://launchpad.net/cuneiform-linux" \t "https://blog.csdn.net/business122/article/details/_blank) —— 本来是个Windows软件，这是Linux的移植，2011年4月已经停止维护。   
[GNU Ocrad](https://www.gnu.org/software/ocrad/" \t "https://blog.csdn.net/business122/article/details/_blank) —— 命令行工具。有JS移植，可用于前端。   
[GOCR](http://jocr.sourceforge.net/" \t "https://blog.csdn.net/business122/article/details/_blank) —— 命令行工具。有JS移植，可用于前端。   
[Tesseract](https://github.com/tesseract-ocr/tesseract" \t "https://blog.csdn.net/business122/article/details/_blank) —— 开源OCR引擎，也有命令行工具。HP开发Google接手。3.0之后支持训练。Golang绑定。入门教程。   
[OCRopy](https://github.com/tmbdev/ocropy" \t "https://blog.csdn.net/business122/article/details/_blank) —— 基于训练的OCR引擎，训练后可以达到比Tesseract更高的准确度，项目比Tesseract更年轻。包含一个叫做OCRopus的布局分析器。in Python。   
[Microsoft OCR Library](https://blogs.windows.com/buildingapps/2014/09/18/microsoft-ocr-library-for-windows-runtime/" \t "https://blog.csdn.net/business122/article/details/_blank) —— Windows8.1之后的版本内置OCR引擎，可用于桌面和WindowsPhone。   
[Abbyy](http://www.abbyy.cn/" \t "https://blog.csdn.net/business122/article/details/_blank) —— 收费软件，有SDK，有Cloud版本。

****预处理代码****   
[OpenCV](http://www.opencv.org.cn/" \t "https://blog.csdn.net/business122/article/details/_blank) —— 图像处理老大哥。OpenCV3中有Scene Text Detection值得一用。   
[Libccv](http://libccv.org/" \t "https://blog.csdn.net/business122/article/details/_blank) —— 现代图像处理库，被很多人推荐。实现了精选的若干个图像处理算法，干净容易移植。其中[Stroke Width Transfor](http://libccv.org/doc/doc-swt/" \t "https://blog.csdn.net/business122/article/details/_blank)尤其有用。   
[lswms](https://sourceforge.net/projects/lswms/" \t "https://blog.csdn.net/business122/article/details/_blank) —— 分行检测。   
[OCRopus](https://github.com/tmbdev/ocropy" \t "https://blog.csdn.net/business122/article/details/_blank) —— 基于神经学习网络算法的布局分析库。教程。   
[TiRG](https://sourceforge.net/projects/tirg/" \t "https://blog.csdn.net/business122/article/details/_blank) —— 文字区域检测库，效果演示。   
[unpaper](https://github.com/Flameeyes/unpaper" \t "https://blog.csdn.net/business122/article/details/_blank) —— 检测文字和旋转，用的是Hough transform算法。

****Scene Text Detection(opencv)****   
[高层包装应用。](https://github.com/tleyden/open-ocr/wiki/Stroke-Width-Transform" \t "https://blog.csdn.net/business122/article/details/_blank)

****高层项目****   
[node-dv](https://github.com/creatale/node-dv" \t "https://blog.csdn.net/business122/article/details/_blank) —— in Node.js，整合了OpenCV、Tesseract和一些其他项目。   
[node-fv](https://github.com/creatale/node-fv" \t "https://blog.csdn.net/business122/article/details/_blank) —— node-dv的更高层，用于证件识别。   
[OpenOCR](https://github.com/tleyden/open-ocr" \t "https://blog.csdn.net/business122/article/details/_blank) —— 包装了SWT、Tesseract、Docker、RabbitMQ，提供队列和HTTP访问服务。in Golang。   
[openalpr](https://github.com/openalpr/openalpr" \t "https://blog.csdn.net/business122/article/details/_blank) —— 包装了Tesseract和OpenCV，支持多系统build，支持Docker，有Python和Node.js绑定。

****android移动端****   
[tess-two](https://github.com/rmtheis/tess-two" \t "https://blog.csdn.net/business122/article/details/_blank)，Tesseract的安卓移植，教程。   
[microblink](https://microblink.com/" \t "https://blog.csdn.net/business122/article/details/_blank)，免费的移动OCR-SDK。

****新方法：机器学习****   
如果有够多的样本和验证能力，机器学习可以很好的处理OCR的问题。   
[http://www.danvk.org/2015/01/09/extracting-text-from-an-image-using-ocropus.html](http://www.danvk.org/2015/01/09/extracting-text-from-an-image-using-ocropus.html" \t "https://blog.csdn.net/business122/article/details/_blank)   
[http://www.danvk.org/2015/01/11/training-an-ocropus-ocr-model.html](http://www.danvk.org/2015/01/11/training-an-ocropus-ocr-model.html" \t "https://blog.csdn.net/business122/article/details/_blank)   
[https://en.wikipedia.org/wiki/Long\_short\_term\_memory](https://en.wikipedia.org/wiki/Long_short_term_memory" \t "https://blog.csdn.net/business122/article/details/_blank)   
[https://github.com/nypl/map-vectorizer](https://github.com/nypl/map-vectorizer" \t "https://blog.csdn.net/business122/article/details/_blank)

一个快速深度学习的框架，和基于它构建的OCR项目。   
[https://github.com/BVLC/caffe/](https://github.com/BVLC/caffe/" \t "https://blog.csdn.net/business122/article/details/_blank)   
[https://github.com/pannous/caffe-ocr](https://github.com/pannous/caffe-ocr" \t "https://blog.csdn.net/business122/article/details/_blank)

JS构建的神经学习网络   
[https://github.com/mateogianolio/mlp-character-recognition](https://github.com/mateogianolio/mlp-character-recognition" \t "https://blog.csdn.net/business122/article/details/_blank)

网络链接地址：<https://blog.csdn.net/business122/article/details/7917494>

1. 几大OCR开源库

1.概述  
1）OCRE(OCR Easy)  
[http://lem.eui.upm.es/ocre.html](http://lem.eui.upm.es/ocre.html" \t "https://blog.csdn.net/qianliheshan/article/details/_blank)  
2）Clara OCR  
[http://directory.fsf.org/claraocr.html](http://directory.fsf.org/claraocr.html" \t "https://blog.csdn.net/qianliheshan/article/details/_blank)  
3）Ocrad(based on a feature extraction)  
[http://directory.fsf.org/ocrad.html](http://directory.fsf.org/ocrad.html" \t "https://blog.csdn.net/qianliheshan/article/details/_blank)  
4）GOCR  
[http://sourceforge.net/projects/jocr](http://sourceforge.net/projects/jocr" \t "https://blog.csdn.net/qianliheshan/article/details/_blank)  
5）OCRchie: Modular Optical Character Recognition Software  
[http://http.cs.berkeley.edu/~fateman/kathey/ocrchie.html](http://http.cs.berkeley.edu/~fateman/kathey/ocrchie.html" \t "https://blog.csdn.net/qianliheshan/article/details/_blank)  
6）tesseract-ocr  
[http://code.google.com/p/tesseract-ocr/](http://code.google.com/p/tesseract-ocr/" \t "https://blog.csdn.net/qianliheshan/article/details/_blank)

7）Grocr

[http://vision.ucsd.edu/~kai/grocr/](http://vision.ucsd.edu/~kai/grocr/" \t "https://blog.csdn.net/qianliheshan/article/details/_blank)

8）[http://www.ocrgrid.org/](http://www.ocrgrid.org/" \t "https://blog.csdn.net/qianliheshan/article/details/_blank)

9）[http://www.ocrgrid.org/ocrdev.html](http://www.ocrgrid.org/ocrdev.html" \t "https://blog.csdn.net/qianliheshan/article/details/_blank)，OCR多个开源库链接

2.1 gocr和ocrad稍作修改可在Windows下编译运行；gocr用vc6, ocrad用gcc。  
2.2 OCRE,OCRchie和clara OCR需要X window或gtk不太好编译。  
2.3 tesseract 用VC ,10年前的东西,太老, 里面很多代码是STL的原型。

3、OpenCV+OCR模型

4、OCR 算法研究试验性的项目，目前CNN+BLSTM+CTC

5、Tesseract在OpenCV+Visual Studio 2015下实现OCR

Tesseract-OCR4.0版本在VS2015上的编译与运行

<https://blog.csdn.net/Shhuikai/article/details/80207891>

<http://lib.csdn.net/article/deeplearning/67228>

1. YOLO
2. CNN+biLSTM+CTC

(https://arxiv.org/pdf/1507.05717v1.pdf)

8、SWT识别、基于智能学习：caffe和crnn

二、 如何利用机器标注的结果提升OCR模型性能：

## 交叉验证：holdout方法和k-fold交叉验证

现成的标注工具是：精灵标注

深度卷积神经网络（Deep Convolutional Neural Network）和循环神经网络（Recurrent Neural Network）结合起来的方法

## 如何评价OCR标注的结果是否可用：

人类判断和自动评价标准相互结合

衡量OCR系统的好坏有两部分内容：

1. 是否成功地圈出了文字；
2. 对于圈出来的文字，有没有成功识别。我们采用评分的方法，对每一张图片的识别效果进行评分。

可以结合图像标注结果评价BLEU、ROUGE、METEOR、CIDEr标准

四、总结

主要基于图像处理，图像理解，和生成对抗网络三种类型的技术快速产生大量带标注数据，在此之外，也在不断积累人工标注数据作为真实样本，这些真实样本不但客观反映了业务场景，也为生成数据规范提供了标杆，即依赖这些真实数据样式来在生成数据环节进行大量的模拟和泛化。在后续的工作中，我们会重点关注如何通过服务化和工具化实现不断的数据自动积累，模型自动训练更新。除OCR外，计算机视觉乃至整个机器学习领域，尽管数据驱动这个词被无数次提到，但真正能够释放数据驱动能力的产品或技术服务依然寥寥无几。如何让机器自身具备数据收集、整理、分析的能力，并自主对算法进行调整和优化，自主做出判断和决策将是我们探索的方向。