

上海银行OCR分析报告

上海萃舟智能科技有限公司

2018-01-28

1. **技术分析**

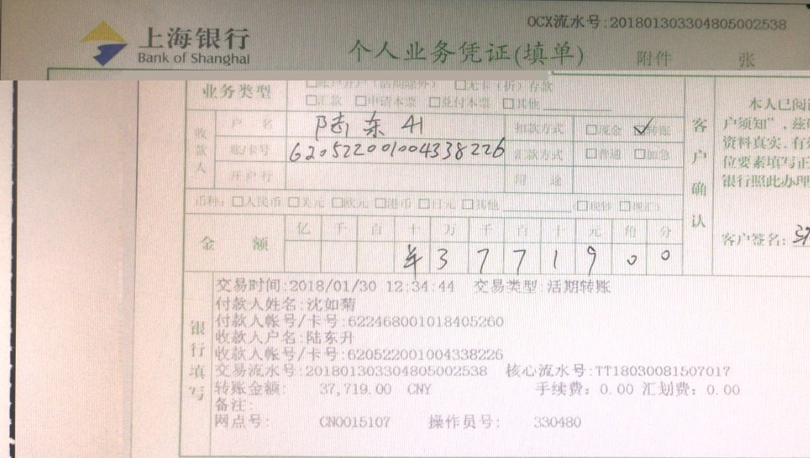
OCR的基本原理就是通过扫描仪将一份文稿的图像输入给计算机，然后由计算机取出每个文字的图像，并将其转换成汉字的编码。其具体工作过程是，扫描仪将汉字文稿通过电荷耦合器件CCD将文稿的光信号转换为电信号，经过模拟／数字转换器转化为数字信号传输给计算机。计算机接受的是文稿的数字图像，其图像上的汉字可能是印刷汉字，也可能是手写汉字，然后对这些图像中的汉字进行识别。对于印刷体字符，首先采用光学的方式将文档资料转换成原始黑白点阵的图像文件，再通过识别软件将图像中的文字转换成文本格式，以便文字处理软件的进一步加工。

目前市场上较为成熟的OCR产品有：证件识别SDK、车牌识别SDK、文档识别SDK、银行卡识别SDK、表格识别SDK、票据识别SDK、名片识别SDK、护照识别SDK、身份证识别SDK。目前，银行、保险、金融、税务、海关、公安、边检、物流、电信工商管理、图书馆、户籍管理、审计等很多行业都已经应用了OCR技术。OCR技术让大家减少了设备配置，降低了人力成本，提高了工作效率。

在OCR领域，中文OCR一直是其中的痛点和难点。随着近年来深度学习的不断发展，中文字符识别精度得到大幅提高。本公司多年从事图像处理和计算机视觉方面的工作，在中文OCR方面有着丰富的技术和经验。

结合本公司的技术经验和上海银行的业务需求，本公司为上海银行开发如下票据识别系统。具体识别内容为以下四大要素：流水号、收款人姓名、收款人账号以及收款金额。票据示例如图1所示。针对该四大要素，本公司从应用的技术手段给出相应的可行性分析。

图 1



识别的四大要素：流水号、收款人姓名、收款人账号和收款人金额，在图2和图3中分别由①②③④所标识。

图 2

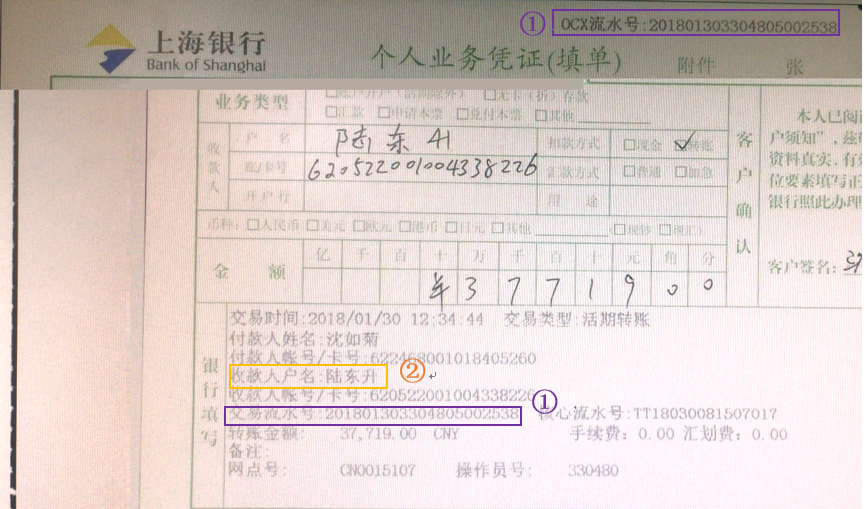
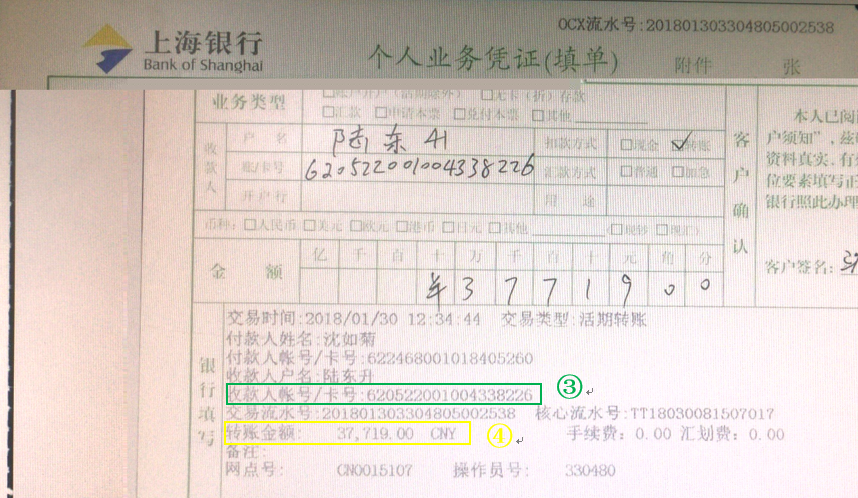


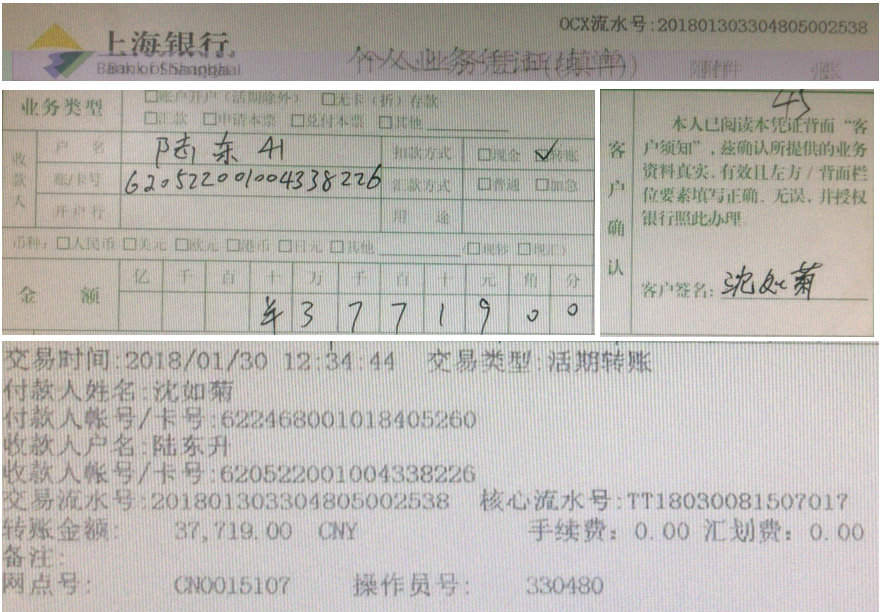
图 3



票据识别系统的技术流程为：（1）图像预处理；（2）文字行提取；（3）文字行字符识别；（4）OCR后处理。

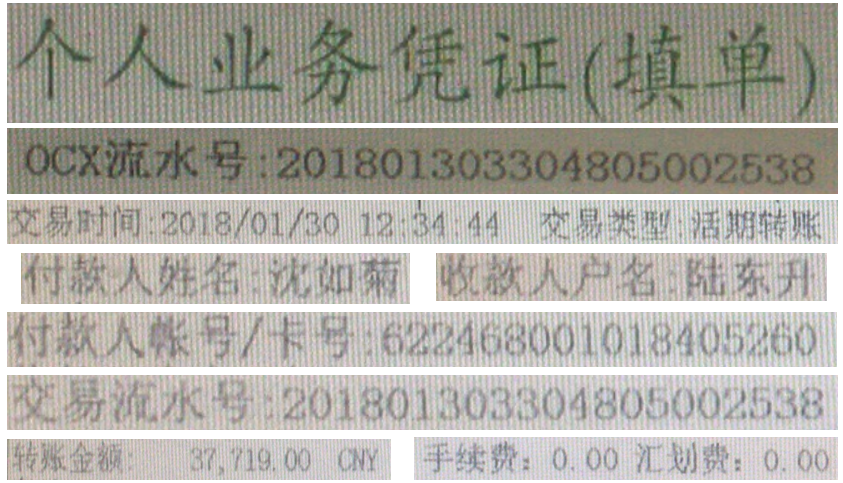
在票据扫描过程中，由于光照、拍摄角度、字体印刷以及印章等的影响，扫描后的字体质量并不能达到最优。为了避免噪声、角度倾斜造成的干扰，在字符识别之前，我们需要对票据进行旋转、仿射变换以及二值化等相关预处理操作，使得票据中的字符呈现最佳效果。此外，为了利用票据中的直线信息，并消除直线对之后字符识别的影响，在预处理过程中，我们还需确定票据中直线的位置信息，在预处理完成后，首先我们利用依据直线信息对票据中的文字信息进行划分，其初步结果如图4所示。

图4 依据直线进行文字区域提取



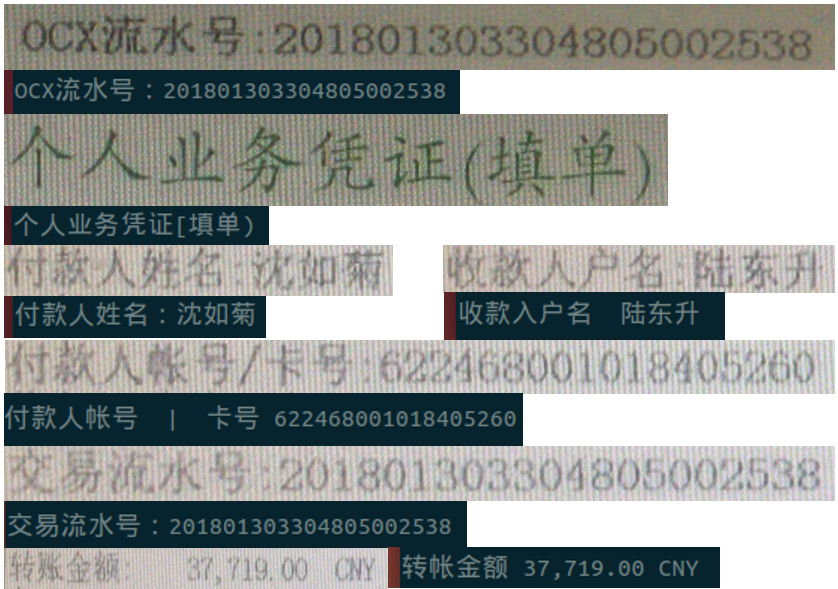
接着在划分后的图像中采用深度学习算法，进行文字行提取，部分文字行提取结果如图5所示。

图 5 文字行检测结果



在文字行提取结束后，我们对每个单独的文字行进行字符识别。字符识别通常有两种思路：（1）字符分割+单个字符分类；（2）单行文字整体识别。方法（1）为传统OCR方法的思路，但其往往忽略了文字之间的语义关系。为了充分利用字符之间的语义关系，我们采用深度网络进行字符行识别，其识别精度相比方法（1），效果更优。其部分识别结果如图6所示。

图 6 字符识别结果



由于中文OCR目前精度仍不能达到100%。在票据识别的过程中难免出现个别字符错误的情况。而此次任务中，我们识别的票据中四个要素的关键词相对固定，不同的要素间格式也不尽相同。所以，我们可采用模糊算法进行匹配，以修正中文OCR的个别字符的错误。如②中“收款人姓名：陆东升”被识别为“收款入姓名：陆东升”，我们依据字符串相似度以及其后字符串为中文字段，可将“入”模糊修正为“人”。

1. **实测分析**
   1. 算法特点与优势
2. **自动判断票据图像是否包含“流水号”字段。**
3. **流水号识别准确率高于传统OCR方法。**
   1. 精度分析

首先，对测试使用的图像数据进行分类说明，见表2-1。其中包含流水号是指，包含21位数字流水号，手写流水号与英文流水号暂未计入。

表 2-1 测试数据结构说明

|  |  |  |  |
| --- | --- | --- | --- |
| **图片类型** | **数量** | **含义** | **图例** |
| 总处理图像 | 17867 | 经过程序判读的所有票据正面图像 |  |
| 无效图像  （不含流水号） | 8144 | 票据正面无“流水号”字段 | 图 2-1 |
| 无效图像  （含流水号） | 215 | 人眼无法认读票据正面“流水号”的真值 | 图 2-2 |
| 有效图像 | 9508 | 票据正面包括人眼可识别真值的21位流水号 |  |

图 2-1 无效图像（不含流水号，票据正面图像）

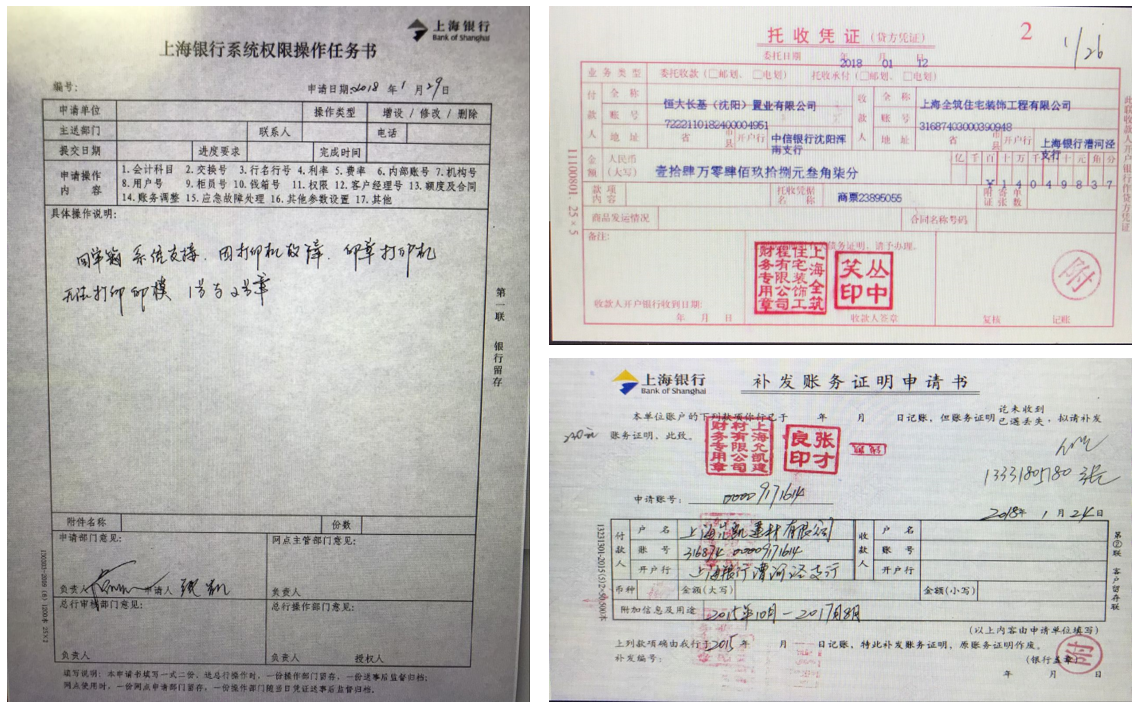
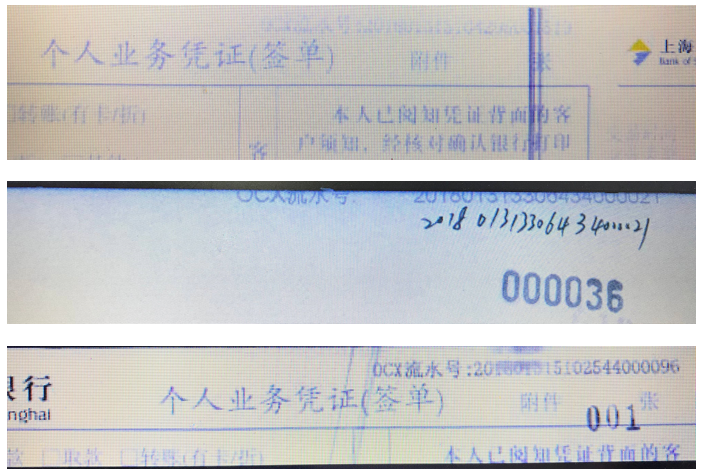


图 2-2 无效图像 （含流水号，人眼无法判读真值）



第一轮测试精度如表2-2。

表 2-2

|  |  |
| --- | --- |
| 图像类型 | 数量 |
| 处理图像总量 | 17867 |
| 含流水号的图像  （人眼不可辩） | 215 |
| 含流水号的图像  （人眼可辩） | 9508 |
| 软件识别出包含流水号图像 | 9184 |
| 软件正确识别流水号的图像 | 8843 |
| 流水号图片检出率 | 96.59% |
| 准确率 | 93.00% |

其中，“处理图像总数”为所有票据的正面图像（图片名为奇数），“有效图片”为包含流水号的票据图片。

“流水号图片检出率”Ratio的计算方式为：

Ratio=

“准确率”Accuracy的计算方式为：

Accuracy=

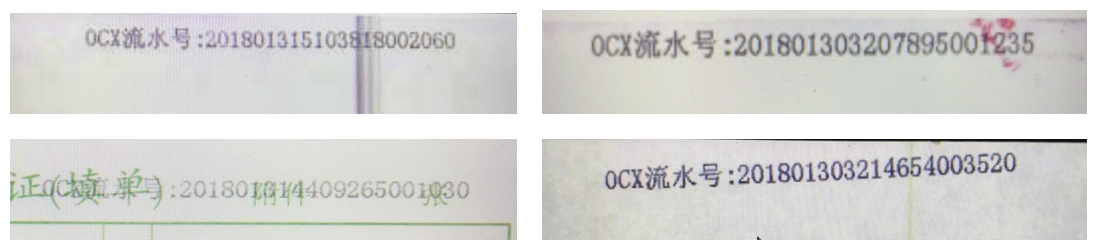
根据第一轮测试结果,精度分析和提升计划如表2-3。

表 2-3

|  |  |
| --- | --- |
| **识别错误的图像种类** | **精度提升方法** |
| 图像旋转角度大或倒放 | 影像纠正处理 |
| 字体褪色严重 | 图像预处理 |
| 字迹被污染 | 图像预处理 |
| 字迹缺损 | 深度学习模型迭代训练 |
| 改进后预计保障精度： 95% | |

识别错误的图像种类图例如图2-3。

图 2-3 识别错误样例：字迹被污染，字体旋转角度大



* 1. 效率分析

第一轮测试硬件环境如下表2-3。

表 2-3

|  |  |
| --- | --- |
| 系统环境 | Win7 使用vmware 安装的ubuntu16虚拟机系统 |
| CPU | Intel i5-4590 3.3Ghz |
| 分配核心数 | 2 |
| 分配内存 | 3.8G |
| 硬盘 | 90G |
| 累计运行时间 | 16.4小时 |
| 处理图像总量 | 19453张 |
| 运行速度 | 1186 张/小时 |
| 5 台同配置机器  处理5万张时间 | 8.4 小时 |

提升硬件条件，数据处理效率水平如下表2-4。

表2-4

|  |  |
| --- | --- |
| CPU | Intel i7-7700HQ |
| 内存 | 8G |
| 核心数 | 2.8GHz \* 8 |
| 预计处理效率 | 2860张/小时 |
| 4 台同配置机器  处理5万张时间 | 4.37 小时 |