**软件需求说明书**

**项目名称：基于深度学习的银行卡识别系统**

**小组名称：Leet Code**

**参与人员：黄新昊 陈硕峰**

**2019 年 4 月 25日**

目录

[1范围 3](#_Toc8522)

[1.1标识 3](#_Toc30716)

[1.2系统概述 4](#_Toc2852)

[1.3文档概述 4](#_Toc22432)

[1.4基线 4](#_Toc15685)

[2引用文件 5](#_Toc21565)

[3需求 5](#_Toc31761)

[3.1所需的状态和方式 5](#_Toc12558)

[3.2需求概述 5](#_Toc29850)

[3.2.1目标 5](#_Toc2713)

[3.2.2运行环境 5](#_Toc20004)

[3.2.3用户的特点 6](#_Toc15518)

[3.2.4关键点 6](#_Toc10444)

[3.2.5约束条件 6](#_Toc21664)

[3.3需求规格 6](#_Toc31360)

[3.3.1软件系统总体功能/对象结构 6](#_Toc23929)

[3.3.2软件子系统功能/对象结构 7](#_Toc1198)

[3.3.3描述规定 7](#_Toc29921)

[3.4CSCI能力需求 8](#_Toc12359)

[3.4.1数据精确度 8](#_Toc25034)

[3.4.2识别率 8](#_Toc19216)

[3.4.3时间特性 9](#_Toc23295)

[3.4.4安全性 9](#_Toc14428)

[3.4.5容错性 9](#_Toc26482)

[3.5CSCI外部接口需求 9](#_Toc30877)

[3.5.1接口标识和接口图 10](#_Toc19976)

[3.5.2接口的类型需求 11](#_Toc18030)

[3.5.3数据元素特性 12](#_Toc30972)

[3.5.4通信特性 12](#_Toc15620)

[3.6CSCI内部接口需求 13](#_Toc4355)

[3.7CSCI内部数据需求 13](#_Toc23346)

[3.8适应性需求 13](#_Toc30419)

[3.9保密性需求 13](#_Toc3926)

[3.10私密性需求 13](#_Toc13768)

[3.11CSCI环境需求 14](#_Toc7151)

[3.12计算机资源需求 14](#_Toc8370)

[3.12.1计算机硬件需求 14](#_Toc25007)

[3.12.2计算机硬件资源利用需求 14](#_Toc4629)

[3.12.3计算机软件需求 15](#_Toc17216)

[3.12.4计算机通信需求 15](#_Toc21341)

[3.13软件质量因素 15](#_Toc14361)

[3.14数据 16](#_Toc2781)

[3.15操作 16](#_Toc3862)

[3.16故障处理 16](#_Toc20111)

[3.16.1属于软件系统的问题 16](#_Toc9901)

[3.16.2发生错误时的信息 17](#_Toc10723)

[3.16.3发生错误时可能采取的补救措施 17](#_Toc20476)

[3.17技术说明 17](#_Toc5939)

[3.17.1OCR 17](#_Toc31484)

[3.17.2图像去噪 18](#_Toc12680)

[3.17.3多列卷积神经网络 19](#_Toc32159)

[3.18有关人员需求 19](#_Toc18902)

[3.19有关培训需求 19](#_Toc5293)

[3.20有关后勤需求 20](#_Toc8929)

[3.20.1硬件方面的需求 20](#_Toc29837)

[3.20.2软件方面的需求 21](#_Toc29021)

[3.21其他需求 21](#_Toc12647)

[3.22包装需求 21](#_Toc27149)

[3.23需求的优先次序和关键程度 21](#_Toc16701)

[4合格性规定 21](#_Toc13811)

[5需求可追踪性 23](#_Toc12273)

[5.1需求追踪表 23](#_Toc9414)

[5.2CSCI能力需求可追踪性 24](#_Toc7248)

[5.3CSCI外部接口需求可追踪性 24](#_Toc30790)

[5.4CSCI内部接口需求可追踪性 24](#_Toc3735)

[5.5CSCI内部数据需求可追踪性 24](#_Toc14133)

[5.6适应性需求可追踪性 25](#_Toc31766)

[5.7保密性需求可追踪性 25](#_Toc11507)

[5.8私密性需求可追踪性 25](#_Toc10448)

[5.9CSCI环境需求可追踪性 25](#_Toc8971)

[5.10计算机资源需求可追踪性 25](#_Toc24404)

[5.11人员需求可追踪性 26](#_Toc3134)

[5.12培训需求可追踪性 26](#_Toc30751)

[6尚未解决的问题 26](#_Toc21709)

[7注解 26](#_Toc13636)

[7.1专业术语解释 26](#_Toc27143)

[7.2特殊名词解释和操作说明 27](#_Toc23607)

[附录A：系统安全需求确认 41](#_Toc23407)

[附录B：需求确认 41](#_Toc1758)

# 1范围

## 1.1标识

OCR（Optical Character Recognition）：[光学字符识别](https://baike.baidu.com/item/%E5%85%89%E5%AD%A6%E5%AD%97%E7%AC%A6%E8%AF%86%E5%88%AB/4162921" \t "https://baike.baidu.com/item/%E5%85%89%E5%AD%A6%E5%AD%97%E7%AC%A6%E8%AF%86%E5%88%AB/_blank)，是指电子设备（例如扫描仪或数码相机）检查纸上打印的字符，通过检测暗、亮的模式确定其形状，然后用字符识别方法将形状翻译成计算机文字的过程

UML:统一建模语言、是一套用来设计软件蓝图的标准建模语言，是一种从软件分析、设计到编写程序规范的标准化建模语言

CNN:卷积神经网络

MCNN:多列卷积神经网络

## 1.2系统概述

人类对外界信息的认识及感知，最基本的就来自于视觉，因此对视觉信息的搜集与处理，一直是人类认识世界、认识规律的重要手段。人工智能技术，通过对视觉信息的采集，对图形图像信息做科学的筛选、比对并分析，然后经过算法（深度学习）、理解和思考之后，将真实的现实内容呈现在计算机中。

随着人工智能、深度学习技术的发展，人工智能技术在视觉领域方面的应用日益突出，得到了广泛的关注和研究。本系统基于深度学习的视觉识别技术，拓展现有的光学识别技术（OCR）来完成一个识别银行卡号的系统，此系统包括数据集处理、银行卡号定位检测、银行卡号识别三部分。

## 1.3文档概述

本文档作为需求规格说明书用于指导银行卡识别系统的项目开发，文档分析了银行卡识别系统的目标、运行环境、用户、数据精确度、查全率、时间特性、安全性、容错性、接口、数据、保密性、硬件需求、软件需求、通信需求、培训需求、后勤需求以及故障处理等各方面。

## 1.4基线

BL-I -RA-V01

BL-E -DS-V02

BL-C -CO-V03

BL-C -UT-V04

BL-C -IT-V05

BL-C -VT-V07

BL-C -ST-V07

BL-C -AT-V08

BL-T -RL-V09

# 2引用文件

1. 计算机软件文档编制规范GB/T 8567-2006
2. 贾铁军等.软件工程与实践[M].北京:清华大学出版社,2017
3. 王珊等.数据库系统概论[M].北京:高等教育出版社,2014
4. 朱少民等《软件项目管理》[M].北京:人民邮电出版社,2015
5. [康一梅](https://baike.baidu.com/item/%E5%BA%B7%E4%B8%80%E6%A2%85" \t "https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E9%A1%B9%E7%9B%AE%E7%AE%A1%E7%90%86/_blank)等《软件项目管理》[M].北京:清华大学出版社,2014
6. 《银行卡识别系统软件开发计划书》

# 3需求

## 3.1所需的状态和方式

用户上传银行卡图片后点击识别即可识别出卡号并返回给用户，显示在WEB界面上。

## 3.2需求概述

### 3.2.1目标

1. 数据集处理：根据本赛题提供的数据集（共1084张卡号截图及标签）实现数据增强模块，将数据集中的每一张图片使用数据增强方式拓展为80张图片，为之后的图像识别训练提供充足的数据样本，并且该模块程序能够继续处理新加入的数据样本。
2. 程序定位银行卡卡号：实现银行卡号文本的检测定位，此模块能够将拍摄的银行卡卡号部分检测出来，并截取相应部分供后续的识别模型使用（银行卡尽量充满图片，横向放置），能够读取放入到文件夹的银行卡图片。
3. 程序实现银行卡号字符识别：此模块要求使用数据增强后的数据集训练字符识别模型，最终模型能够识别出赛题提供的测试银行卡卡号以及自拍的银行卡卡号。
4. 采用GUI（图形用户界面）：提供良好的用户交互式界面实现。

### 3.2.2运行环境

操作系统：windows 2000及以上版本

硬件环境：主机运行内存（RAM）4GB，存储器500G；显示器等外部设备

软件环境：office等常用软件

### 3.2.3用户的特点

1. 最终用户特点:最终用户主要是银行工作人员或需要进行移动支付的人,软件设计等符合该类群体的使用习惯。
2. 操作人员的教育水平和技术专长:能够熟练使用计算机和相关软件；维护人员的教育水平和技术专长:本科/软件开发。

（3）本软件的预期使用频度:系统正常发布后，预期全年的用户使用频率基本保持平稳。

### 3.2.4关键点

数据集处理

定位银行卡卡号

实现银行卡号字符识别

### 3.2.5约束条件

经费限制：本软件无经费支持，项目组成员应该节约资源，充分发挥个人主观能动性。

开发期限：本软件最晚于2019年6月13日完成。

开发过程：软件遵守《银行卡识别系统软件开发计划书》中指定的编码规则、对象命名约定、排版规定、注释规定、测试及维护规定。

安全保密性：本软件正式发布之前项目组成员应保证软件的安全和保密，遵守安全规定和保密规定。

## 3.3需求规格

### 3.3.1软件系统总体功能/对象结构

银行卡识别系统通过对上传的银行卡图像进行一系列的处理与识别最后返回识别出来的卡号。

系统架构如下：

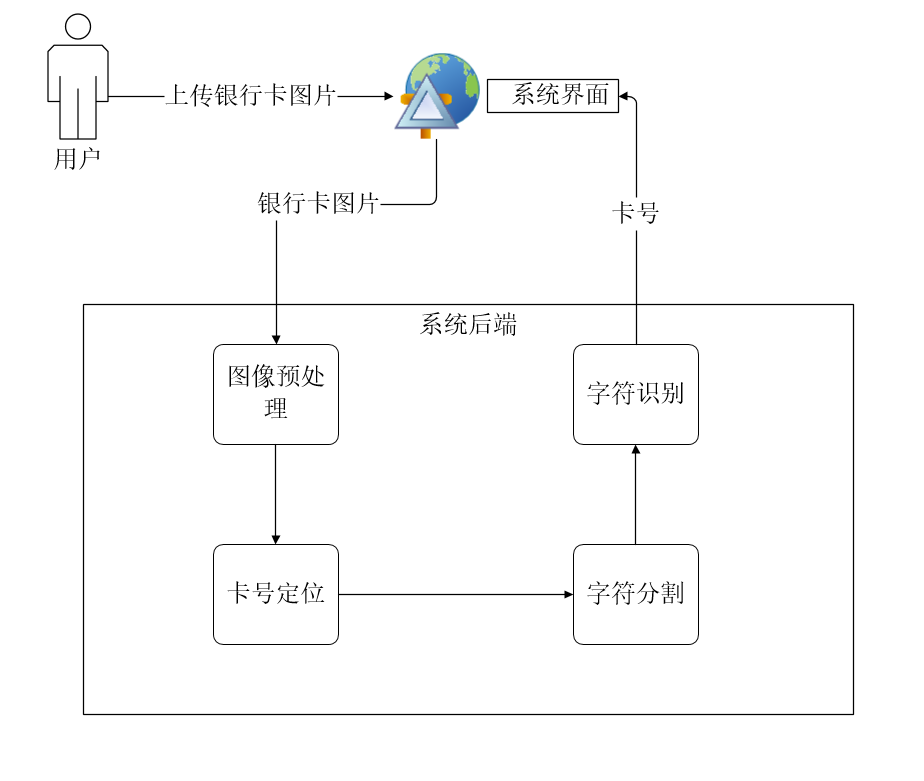


图1 系统架构图

### 3.3.2软件子系统功能/对象结构

#### 3.3.2.1前端界面模块

前端界面模块需要具有上传图像，并从后端获取识别的卡号并返回界面的功能，还要具有从本地选择图片的功能。另外，前端界面需要保持简洁美观，方便实用。

#### 3.3.2.2后端卡号识别模块

后端的卡号识别模块负责将从前端获取的银行卡图像进行预处理、卡号定位、字符分割、字符识别操作，该模块为本系统主要模块。

### 3.3.3描述规定

**编码规定**：

（1）选用控制结构只允许一个入口和一个出口。

（2）程序语句组成容易识别的块，每块只有一个入口和一个出口。

（3）复杂的结构应该用基本控制结构进行组合嵌套来实现。

（4）语句中没有的控制结构，可用一段登家的程序段模拟，但要求该程序段在整个系统应前后一致。

**对象命名规定**：

对象名称=对象前缀+自定义名称（自定义名称要有一定的意义且第一个字母大写）。

**文档排版规定**：

（1）较长的语句、表达式要分成多行书写。

（2）长表达式要在低优先级操作符处划分新行，操作符放在新行之首。

（3）循环、判断等语句中若有较长的表达式或语句，则要进行适当的划分。

（4）若函数中的参数较长，则要进行适当的划分。

（5）不允许把多个短语句写在一行中，即一行只写一句语句。

（6）缩进应该是每行2个空格，不要在源文件中保存tab字符，在使用不同的源代码管理工具是tab字符将因为用户设置的不同而扩展为不同的宽度。

## 3.4CSCI能力需求

### 3.4.1数据精确度

1. 说明

数据精确度规定数据的格式。

1. 输入

要按照严格的数据格式输入，否则系统不给予响应进行处理。

1. 处理

检查数据有效性

检查操作顺序

检查相应异常情况，如溢出、通信故障、错误处理

1. 输出

数据精确度信息，是否可以处理数据。

### 3.4.2识别率

系统需要达到至少90%的识别率，即对于市面上90%的银行卡都要做到准确识别。

### 3.4.3时间特性

1. 用户点击识别按钮后，显示识别结果，此时一般响应时间不超过2秒。
2. 支持5000名用户并发使用，并保证性能不受影响。

### 3.4.4安全性

1. 权限控制

用户只有上传图片并带点击识别按钮查看返回结果的权限。

1. 重要数据加密

本系统对一些重要的数据按一定的算法进行加密，如用户口令、重要参数等。

### 3.4.5容错性

系统具有一定的容错和抗干扰能力，在非硬件故障或非通讯故障时，系统能够保证正常运行，并有足够的提示信息帮助用户有效正确地完成任务。

## 3.5CSCI外部接口需求

1. 用户接口需求：

要求使用图形化界面，方便用户进行使用，主要接口为：

1. 本地图片浏览选择接口（该接口保证用户可以从本地上传银行卡图像）；
2. 图片上传接口（该接口保证用户选择的图片并在点击识别按钮后可以将银行卡图片上传到后端进行识别）；
3. 硬件接口：

无。

1. 软件接口：
2. 摄像头获取接口（该接口能够让用户使用摄像头进行拍摄获取银行卡图片）；
3. 图片预处理模块接口（用于获取上传的图片并进行预处理）；
4. 卡号定位模块接口（该接口用于获取预处理过的银行卡图片）；
5. 字符分割模块接口（该接口用于获取从整个银行卡图片上切下的卡号部分图片）；
6. 字符识别模块接口（该接口用于获取切割后的所有字符图片）；
7. 通信接口：

前端与后端之间传递图片与卡号数据的接口。

以上接口的详细内容见3.5.1。

### 3.5.1接口标识和接口图

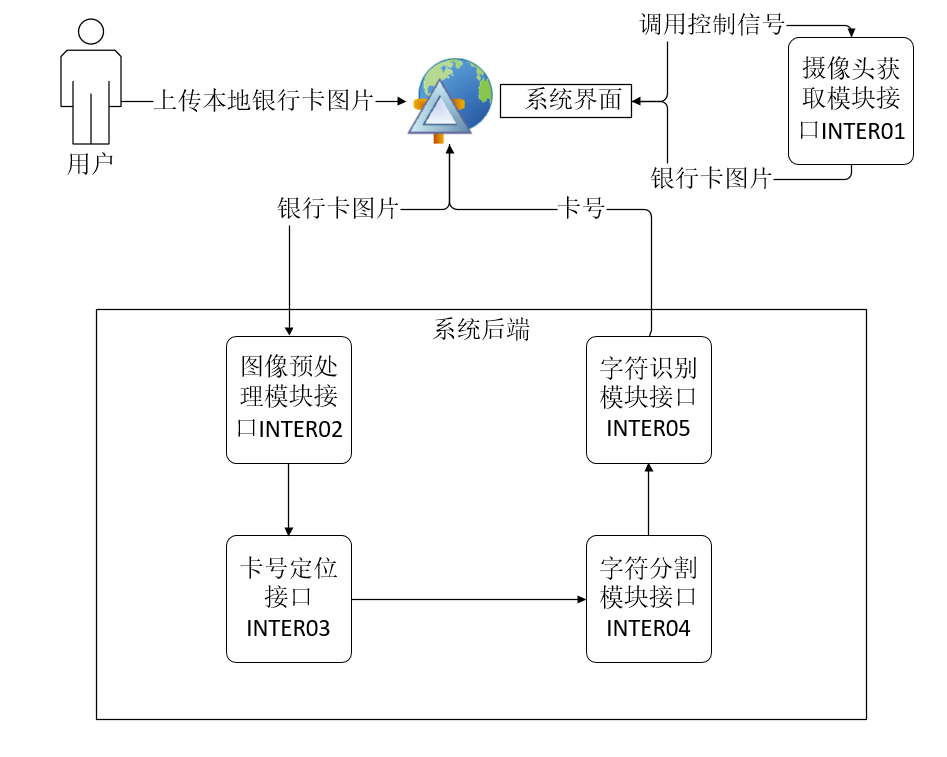


图 2 外部接口示意图

上图为根据外部接口需求设计的外部接口示意图。

摄像头获取接口能够让用户使用摄像头进行拍摄获取银行卡图片；

图片预处理模块接口用于获取上传的图片并进行预处理；

卡号定位模块接口用于从图片预处理模块接口获取预处理过的银行卡图片；

字符分割模块接口用于从卡号定位模块获取从整个银行卡图片上切下的卡号部分图片；

字符识别模块接口用于从字符分割模块获取切割后的所有字符图片；

下表则为外部接口的详细信息，表中显示了各个接口的名称、项目唯一标识符（每个接口在本项目中的唯一标识符）、接口简述、数据来源（接口调用方）和数据目的地（接口提供方），接口简述描述了每个接口的作用，表明了每个接口对项目需求的满足情况。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **接口名称** | **接口项目唯一标识符** | **接口简述** | **来源** | **目的地** |
| 摄像头获取接口 | INTER01 | 该接口能够让用户使用摄像头进行拍摄获取银行卡图片 | 用户 | 前端 |
| 图片预处理模块接口 | INTER02 | 用于获取上传的图片并进行预处理 | 前端 | 卡号定位模块 |
| 卡号定位模块接口 | INTER03 | 用于从图片预处理模块接口获取预处理过的银行卡图片 | 图片预处理模块 | 字符分割模块 |
| 字符分割模块接口 | INTER04 | 用于从卡号定位模块获取从整个银行卡图片上切下的卡号部分图片 | 卡号定位模块 | 字符识别模块 |
| 字符识别模块接口 | INTER05 | 用于从字符分割模块获取切割后的所有字符图片 | 字符分割模块 | 前端 |

表 1 外部接口标识

### 3.5.2接口的类型需求

|  |  |  |
| --- | --- | --- |
| **接口名称** | **接口项目唯一标识符** | **类型** |
| 摄像头获取接口 | INTER01 | 数据请求 |
| 图片预处理模块接口 | INTER02 | 数据请求 |
| 卡号定位模块接口 | INTER03 | 数据请求 |
| 字符分割模块接口 | INTER04 | 数据请求 |
| 字符识别模块接口 | INTER05 | 数据请求 |

表 2 接口类型

上表中描述的是每个接口的类型，描述了该接口适用于数据请求、硬件控制或者数据检索。

### 3.5.3数据元素特性

数据元素(data element)是数据的基本单位，数据元素也叫做结点或记录。在[计算机程序](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%A8%8B%E5%BA%8F/3220205" \t "https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%85%83%E7%B4%A0/_blank)中通常作为一个整体进行考虑和处理。有时，一个数据元素可由若干个[数据项](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%A1%B9/3227309" \t "https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%85%83%E7%B4%A0/_blank)组成，例如，一本书的书目信息为一个数据元素，而书目信息的每一项（如书名、作者名等）为一个数据项。数据项是数据的不可分割的最小单位。

数据元素理论属于[信息标准化](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF%E6%A0%87%E5%87%86%E5%8C%96/9883772" \t "https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%85%83%E7%B4%A0/_blank)的基础理论，即是数据[规范化](https://baike.baidu.com/item/%E8%A7%84%E8%8C%83%E5%8C%96/3193374" \t "https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%85%83%E7%B4%A0/_blank)理论基础。

数据[元分析](https://baike.baidu.com/item/%E5%85%83%E5%88%86%E6%9E%90" \t "https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%85%83%E7%B4%A0/_blank)在信息分类、数据的集成参考模型、数据模型优化设计、[数据元](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%85%83/7681154" \t "https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%85%83%E7%B4%A0/_blank)字典以及制订数据交换标准等方面得到应用。

下表为本项目的数据元素特性表。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **标识符** | **含义** | **数据类型** | **大小** | **来源和接收者** |
| 图片 | image | 传输的图片 | Object | <=10M | 本地-->前端  摄像头-->前端 |
| 卡号 | cardId | 返回的卡号 | String | 字符串属性的长度<=50 | 后端-->前端 |

表 3 数据元素特性表

### 3.5.4通信特性

该部分描述了每个接口通信的数据传输周期、间隔、速率，以及其安全性、保密性、时效性的相关内容，该部分保证了项目中数据传输的时效性及安全保密性。

|  |  |  |
| --- | --- | --- |
| **项目唯一标识符** | **数据传输周期/间隔/速率** | **安全性/保密性/时效性** |
| INTER01 | 使用摄像头拍摄并传递图片到前端时间为立即显示 | 无 |
| INTER02  INTER03  INTER04  INTER05 | 这些接口之间的数据传输为程序内部传输，忽略不计 | 传输的数据要进行检验 |

表 4 通信特性表

## 3.6CSCI内部接口需求

内部接口留待设计时决定。

## 3.7CSCI内部数据需求

MySQL数据库：

max\_allowed\_packet = 64M

disable-auto-rehash #允许通过[TAB键](https://www.baidu.com/s?wd=TAB%E9%94%AE&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "https://zhidao.baidu.com/question/_blank)提示

default-character-set = utf8

connect-timeout = 3

其他内部数据需求留待设计时决定。

## 3.8适应性需求

* 前端端需要能兼容各种摄像头，获取的影像和图片格式需要统一，具体格式留待设计时决定。
* 用户端的web端要能兼容目前主流浏览器，若后续开发手机端需使用微信小程序，若开发app则需同时开发iOS版和Android版。
* 系统要能承载至少5000用户并发访问。
* 其他适应性需求留待设计时添加。

## 3.9保密性需求

* 对于用户卡号在进行传输时进行加密和认证。
* 保证不同用户之间不会出现数据串流的情况。

## 3.10私密性需求

* 不同用户之间保证不发生数据串流。
* 开发人员和管理人员不可随意泄露、修改识别结果。
* 开发人员不可故意架设后门，私密数据只可在本开发小组或具有权限的其他小组成员间传递。

## 3.11CSCI环境需求

LED显示屏——用于显示输出画面

网络设备——用于数据传输

兼顾运营商与地理位置的CDN服务器——用于加快系统实时响应速度

高IO性能服务器——用于实现分布式数据存储

高计算性能CPU服务器——用于运行系统中的算法模型

## 3.12计算机资源需求

### 3.12.1计算机硬件需求

**工作站**

CPU： Intel(R) Xeon(R) 四核X5530 2. 40GHz @ 2.30GHz

内存: 8GB(4G\*2) DDR3-1333 带容错修复功能

Disk：4T机械硬盘Raid5阵列

主板芯片: Intel 5520芯片组

显卡:独立显示，要求使用NVIDIA GeForce GT220或ATI Radeon HD 5500以上显示芯片，512M或以上显存，同时带有VGA及DVI接口

RAM：DDR4 96GB

网卡：多口千兆网络适配器

电源: 2个460W的热插拔电源，支持1+1冗余。光驱: DVD-ROM

**录像头**

分辨率:每个输入通道最高支持D1分辨率编码VGA输出:最高分辨率达1024\*768支持双码流

串行接口，1个USB接口

工作温度: 0'C--+ 55C工作湿度: 10%--90%

### 3.12.2计算机硬件资源利用需求

需要支持同时20000个以上UDP连接接入；

支持T级别DDoS防御网络带宽动态扩容；

支持12小时内数据备份抗灾

### 3.12.3计算机软件需求

OS：Red Hat Enterprise Linux 8 beta

数据库管理系统：MySQL

负载均衡服务器：Nginx

SSL-SNI代理：Haproxy

分布式数据流管理：Hadoop

### 3.12.4计算机通信需求

CDN命中率45%以上；

Web服务首字节时间：0.400 ± 0.100 s

Web服务渲染时间：4.400 ± 1.000 s

Web服务速度指数：12500 ± 2000

流媒体吞吐量：1000Mbps

## 3.13软件质量因素

识别系统用户体验好，界面简单易用；

识别算法的时间和空间复杂度最优，识别准确率高；

数据展示和传输无卡顿。

## 3.14数据

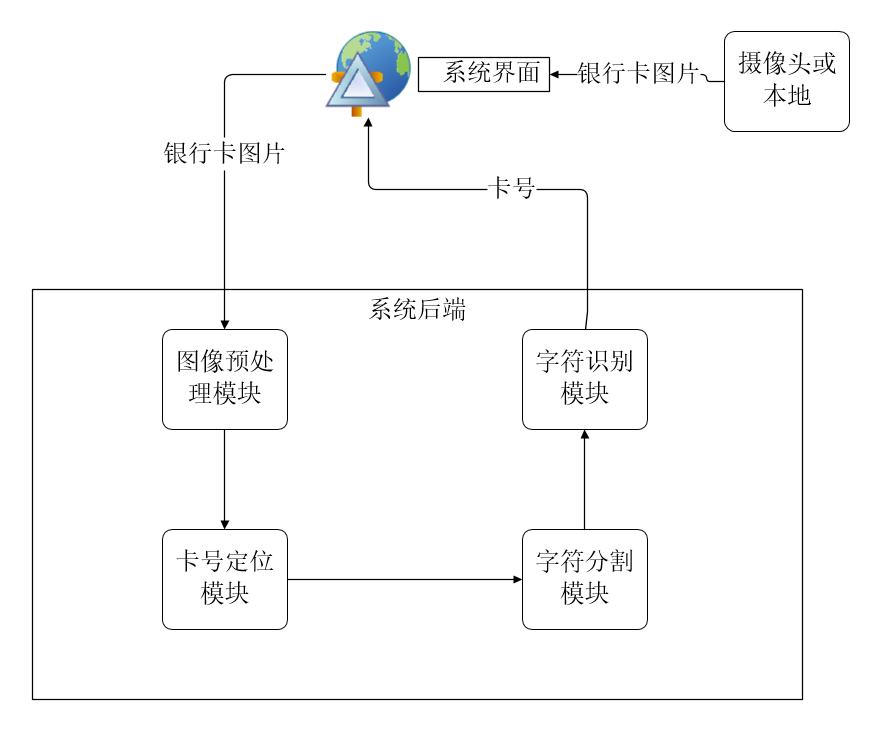


图3数据流向图

## 3.15操作

系统的主体功能操作有前端界面提供，后端负责处理并反馈结果。

在正常的情况下，前端能快速将用户的图片上传到后端，并且使用分布式框架进行并行操作，提高反馈效率，用卷积神经网络以及多列卷积神经网络的算法模型以及相关的OCR技术实时识别银行卡号，并将识别结果返回给用户。

## 3.16故障处理

### 3.16.1属于软件系统的问题

本系统可能发生如下的几项常见故障：

前端无法调用摄像头，导致无法实时拍摄银行卡图片；

前端无法从本地获取图片；

对于上传非银行卡图片的情况无法做出识别；

后端识别算法可能会出错；

### 3.16.2发生错误时的信息

当发生错误时，系统的程序要求能够捕获错误并且给出错误的基本信息，包括了错误的类型，错误的原因，发生错误的时间等。根据这些基本的错误信息，系统开发人员或者维护人员能够及时的解决错误。

### 3.16.3发生错误时可能采取的补救措施

根据可能发生的异常故障分析可得，发生的故障原因发生在数据的及时获取和处理有着很大的关系。因此我们可以根据分析做出以下的补救措施，以减少发生错误时产生的影响。

保障硬件设备能够实时的正常工作运转，以便能实时提供源数据；

保障数据传输过程的实时性，以便边缘程序能获取源数据；

保障网络的稳定性；

## 3.17技术说明

### 3.17.1OCR

OCR技术是光学字符识别的缩写(Optical Character Recognition)，是通过扫描等光学输入方式将各种票据、报刊、书籍、文稿及其它印刷品的文字转化为图像信息，再利用文字识别技术将图像信息转化为可以使用的计算机输入技术。可应用于银行票据、大量文字资料、档案卷宗、文案的录入和处理领域。适合于银行、税务等行业大量票据表格的自动扫描识别及长期存储。相对一般文本，通常以最终识别率、识别速度、版面理解正确率及版面还原满意度4个方面作为OCR技术的评测依据；而相对于表格及票据， 通常以识别率或整张通过率及识别速度为测定OCR技术的实用标准。

采用OCR识别技术,可以将其应用于银行票据光盘缩微系统，可以自动提取票据要素，可减轻操作员的工作量，减少重复劳动，尤其是在与银行事后且监督系统相结合后，可以替代原先的操作人员完成事后监督工作。由计算机自动识别票据上的日期、帐号、金额等要素，通过银行事后监督系统与业务系统中的数据进行比较，完成传统的事后监督操作；配有印章验证系统后，自动将凭证图像中的印章与系统中预留的印鉴进行比较，完成印章的真伪识别。

OCR识别技术不仅具有可以自动判断、拆分、 识别和还原各种通用型印刷体表格，在表格理解上做出了令人满意的实用结果，能够自动分析文稿的版面布局，自动分栏、并判断出标题、横栏、图像、表格等相应属性，并判定识别顺序，能将识别结果还原成与扫描文稿的版面布局一致的新文本。表格自动录入技术，可自动识别特定表格的印刷或打印汉字、字母、数字，可识别手写体汉字、手写体字母、数字及多种手写符号，并按表格格式输出。提高了表格录入效率，可节省大量人力。同时支持将表格识别直接还原成PTF、PDF、HTML等格式文档；并可以对图像嵌入横排文本和竖排文本、表格文本进行自动排版面分析。  
　　利用目前的高新技术-OCR，直接从凭证影像中提取金额、帐号等重要数据，代替人的手工录入，与条码识别/流水识别紧密结合，实现建立事后副本帐、完成事后监督的工作。OCR处理一般使用性能较好的PC机，OCR处理程序一经启动会自动扫描数据库中的凭证影像，发现有需OCR处理而未处理的，提取到本地进行处理。 　   
　　OCR手写体、印刷体识别技术，能识别不同人写的千差万别的手写体汉字和数字，应用于本系统，识别凭证影像中储户填写的信息，如大写金额、小写金额、帐号、存期、日期、证件号等，可以代替手工录入。同时被识别得出的金额还要与流水识别所得的金额进行核对，核对成功，则OCR识别成功。这样处理是为了避免误判。 　   
　　经过对银行产生的实际凭证进行的大量测试，在实际开发过程中，根据银行的实际需求，OCR技术在票据和表格识别能力和手写体自动识别能力上不断提升，目前处理速度可达到每分钟60～80张票据，存折识别率已经达到了85%以上，存单、凭条识别率达到90%以上，而85%以上的识别率就能减少80%以上的数据录入员。

在档案领域OCR技术使档案扫描成果达到了全文可识别，将档案数字化发展提升到了一个新的阶段，是原本扫描出来的图片变得更容易进行检索，为数字档案馆的数据查询提供了技术支持，是档案数字化发展中必不可少的一环。

### 3.17.2图像去噪

图像去噪是指减少数字图像中噪声的过程。现实中的数字图像在数字化和传输过程中常受到成像设备与外部环境噪声干扰等影响，称为含噪图像或噪声图像。

噪声是图象干扰的重要原因。一幅图象在实际应用中可能存在各种各样的噪声,这些噪声可能在传输中产生,也可能在量化等处理中产生。

### 3.17.3多列卷积神经网络

多阵列(Multi-Column)的卷积神经网络(Convolutional Neural Network，CNN)可以实现提取不同尺度的人头特征，将图像映射到其人群密度图上。每列CNN学习得到的特征可以自适应由于透视或图像分辨率引起的人/头大小的变化，并能在不需要输入图的透视先验情况下通过几何自适应的核来精确计算人群密度图。

MCNN网络的每一列并行的子网络深度相同，但是滤波器的大小不同（大，中，小），因此每一列子网络的感受野不同，能够抓住不同大小人头的特征，最后将三列子网络的特征图做线性加权（由1x1的卷积完成）得到该图像的人群密度图，类似模型融合的思想。采用了2\*2的max-pooling和ReLU激活函数。

## 3.18有关人员需求

此项目应投入人力资源为6（人/三个月），其中包括：一名项目经理负责整个项目的跟进和定制整个项目的计划安排。一名软件工程师负责软件开发过程的管理和督促。一名测试工程师负责在开发的各个级段的测试工作。三名实施员组成一个开发小组，根据软件工程师的计划实施开发。本项目一共需要6人。如下人力资源表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **角色** | **数量（位）** | **到位时间** | **备注（人员）** |
| 1 | 项目经理 | 1 | 2019年3月10日 | 黄新昊 |
| 2 | 软件工程师 | 1 | 2019年3月10日 | 陈硕峰、黄新昊 |
| 3 | 测试员 | 1 | 2019年3月10日 | 陈硕峰 |
| 4 | 实施员 | 3 | 2019年3月10日 | 陈硕峰、黄新昊 |

## 3.19有关培训需求

1. 基本的团队管理知识，能合理调度团队成员，使有限资源最大化
2. 开发者对软件工程有较深的理解，熟悉整套软件的开发流程
3. 开发者应当熟悉银行卡识别相关技术
4. 开发者对多列卷积神经网络和OCR技术方面的知识有较深入理解，能将其合理应用到项目中。
5. 培训的计划要求

|  |  |  |  |
| --- | --- | --- | --- |
| **培训内容** | **时间** | **参加者** | **培训方式** |
| 软件工程的基本流程 | 2019年3月30日 | 全部成员 | 陈宁江老师指导 |
| OCR技术学习 | 2019年4月8日 | 全部成员 | 视频学习 |
| 深度学习，卷积神经网络方面知识 | 2019年4月9日-2019年4月12日 | 全部成员 | 白琳老师指导 |
| 熟悉卷积神经网络框架（Caffe、TensorFlow） | 2019年4月13日-2019年4月15日 | 全部成员 | 白琳老师指导 |

## 3.20有关后勤需求

### 3.20.1硬件方面的需求

为了完成项目的开发工作，开发人员需要每人分配一台计算机，并且提供开发人员相对保密的工作环境。为了项目涉及数据的保密性和安全性，需要和项目开发人员签订保密协议。

计算机硬件资源包括计算机的CPU、内存、显存、固态硬盘、磁盘容量等的支持，除外还需要云服务器的运用。

具体的明细如下：

服务器：Intel 酷睿i5 4590以上或更高，

内存：1024M以上；

硬盘：128G以上；

CD－ROM：32倍速以上；

网络适配器：100MB/1000MB自适应；

打印机一台

UPS(选配)

工作站：戴尔（DELL）T30以上微机；

内存：8156MB

硬盘：1T以上；

CD－ROM：32倍速以上；

网络适配器：100MB/1000MB自适应

网络： 一台服务器

一台工作站

使用TCP/IP协议的局域网

### 3.20.2软件方面的需求

软件方面的需求包括了操作系统、数据库和web浏览器，以及一些为了完成开发工作的其他软件。

具体的明细如下：

操作系统为Window 10，使用集成开发工具Eclipse、pycharm,数据库采用MySQL，项目运行环境为python3.6、JDK7.0.

其他开发工具包括：Dreamweaver, Microsoft Visio, Rational Rose, Power Designer Trial 11, TomCat7.0 CVSNT2.5.03

## 3.21其他需求

需要向本小组成员明确待统计统计地点，时间段。并且配备相应硬件系统，做到规范操作，备份好数据防止数据丢失，软件交付日期为软件安装测试人员提供环境。

## 3.22包装需求

前端UI设计

## 3.23需求的优先次序和关键程度

本规格较为重要的是对银行卡卡号部分的定位，该部分在系统中是难度最大的，因为当前市面上的银行卡的卡号位置并不固定；其次重要的就是字符切分部分，因为银行卡的卡好位数不是固定的，所以需要动态识别卡号位数，然后才能进行字符切割。

# 4合格性规定

|  |  |  |  |
| --- | --- | --- | --- |
| **需求** | | **所用方法** | **说明** |
| CSCI能力需求 | 数据精确度 | 测试 | 输入正确与错误格式的样本数据，检测是否对于正确的数据是否正常接收处理，对错误格式数据是否提示错误。 |
| 识别率 | 测试 | 检测是否能返回正确识别结果 |
| 时间特性 | 测试 | 测试客户端的响应时间、数据处理时间、多用户使用情况下的响应情况是否达到预期要求 |
| 安全性 | 审查 | 审查有关权限部分的代码、数据加密算法的正确性 |
| 容错性 | 分析 | 预测可能出现的错误，在非硬件故障或通讯故障发生错误时，能够进行友好提示 |
| CSCI外部接口需求 | 用户接口 | 测试 | 测试从本地、摄像头获取图片是否可行 |
| 硬件接口 | 测试 | 无 |
| 软件接口 | 测试 | 测试个接口之间数据是否能够正确无损传输 |
| 通信接口 | 测试 | 图片与卡号能否快速正确的在前后端之间传递 |
| CSCI内部接口需求 | | 内部接口留待设计时决定 | |
| CSCI内部数据需求 | | 审查 | 审查MySql配置部分代码，其余数据需求留待设计时决定 |
| 适应性需求 | | 测试 | 兼容不同摄像头、能满足5000个以上用户同时使用 |
| 保密性需求 | | 测试 | 用户端测试上传的图片和获取的反馈是否具有保密性 |
| 私密性需求 | | 测试、审查、特殊 | 测试不同用户之间的信息在前后端之间传递时是否会交叉 |
| CSCI环境需求 | | 审查 | LED显示屏——用于显示输出画面  网络设备——用于数据传输  兼顾运营商与地理位置的CDN服务器——用于加快系统实时响应速度  高IO性能服务器——用于实现分布式数据存储  高计算性能CPU服务器——用于运行系统中的算法模型 |
| 计算机资源需求 | 硬件 | 审查 | CPU：Intel(R) Xeon(R) CPU E7-4850 v2 @ 2.30GHz  Disk：4T机械硬盘Raid5阵列  RAM：DDR4 96GB  网卡：多口千兆网络适配器 |
| 硬件资源利用 | 测试 | 测试多用户连接、测试DDos攻击预防结果 |
| 软件 | 审查 | OS：Red Hat Enterprise Linux 8 beta  数据库管理系统：MySQL  负载均衡服务器：Nginx  SSL-SNI代理：Haproxy  分布式数据流管理：Hadoop |
| 通信 | 分析 | 测试使用该Web服务，统计分析结果是否满足预期要求 |
| 人员需求 | | 审查 | 一名项目经理、一名软件工程师、一名测试工程师和三名实施员 |
| 培训需求 | | 验收限制 | 对给定培训目标，为培训人员分配任务，采用验收方式决定是否通过培训 |
| 后勤需求 | 硬件 | 审查 | 提供计算机硬件资源包括计算机的CPU、内存、显存、固态硬盘、磁盘容量等的支持，除外还需要云服务器的运用 |
| 软件 | 审查 | 包括操作系统、数据库和web浏览器的可用性 |

表 5 合格性规定表

# 5需求可追踪性

## 5.1需求追踪表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **需求** | **功能性需求** | | **代码组件** | **用户需求** | | | **系统实现** | **安全** | | **测试** |
| CSCI能力需求 | √ | |  | √ | | |  | √ | |  |
| CSCI外部接口需求 | √ | | √ |  | | | √ |  | | √ |
| CSCI内部接口需求 | 留待设计 | | | | | | | | | |
| CSCI内部数据需求 | √ |  | | |  | √ | | |  |  |
| 适应性需求 | √ |  | | | √ | √ | | |  |  |
| 保密性需求 |  |  | | | √ |  | | | √ |  |
| 私密性需求 |  |  | | |  |  | | | √ |  |
| CSCI环境需求 | √ |  | | | √ | √ | | |  |  |
| 计算机资源需求 | √ |  | | |  | √ | | | √ |  |
| 人员需求 | √ |  | | |  | √ | | |  |  |
| 培训需求 | √ |  | | |  | √ | | |  |  |
| 后勤需求 |  |  | | |  | √ | | |  |  |

表 6 需求追踪表

## 5.2CSCI能力需求可追踪性

由于数据精确度规定数据格式，本系统通过对具体图片进行处理，实现对银行卡卡号的识别，以满足系统功能性需求。用户在使用该系统时，可以上传不同种类的银行卡。时间特性方面的需求是考虑到用户使用的需求，提供软件友好性，对于响应用户的点击请求应小于1.5秒，同时在其余各种报表、视频导入功能中，亦有相应的时长限制。该系统同时为用户设定权限、对重要性数据进行加密，以此来满足系统的安全性需求。

## 5.3CSCI外部接口需求可追踪性

该需求是满足系统功能性需求的主要模块，在用户接口方面，使用图形化界面方便用户使用，体现软件的友好性。在硬件接口方面，程序自动获取移动设备的摄像头，使用该接口从摄像头拍摄银行卡图片，进行识别，以此满足系统实现、功能性需求。

## 5.4CSCI内部接口需求可追踪性

留待设计时决定。

## 5.5CSCI内部数据需求可追踪性

为满足程序处理需求以及系统实现需求，设置MySQL数据库为如下：

max\_allowed\_packet = 64M

disable-auto-rehash #允许通过[TAB键](https://www.baidu.com/s?wd=TAB%E9%94%AE&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "https://zhidao.baidu.com/question/_blank)提示

default-character-set = utf8

connect-timeout = 3

其他内部数据需求留待设计时决定。

## 5.6适应性需求可追踪性

为了满足不同的浏览器的适应性需求，用户端的web端应将能兼容目前主流浏览器。若后续在移动端开发，将开发app的Android版和iOS版，以满足用户需求。对于后端处理采取分布式架构，要能多用户并发操作。

## 5.7保密性需求可追踪性

为满足用户隐私需求，在返回卡号时要对用户进行认证，防止发送妒忌想错误。

## 5.8私密性需求可追踪性

为满足用户隐私及安全性需求，不同边缘端保证隔离，互不可见，同时用户端进行传送数据时，应先验证用户身份。为保障系统的安全性及用户使用安全性，开发人员开发人员不可故意架设后门，私密数据只可在本开发小组或具有权限的其他小组成员间传递。为保障用户的正常使用，开发人员不可随意对用户端数据进行更改。

## 5.9CSCI环境需求可追踪性

为满足系统的实现需求，同时满足用户端的使用需求，CSCI环境应为如下：

LED显示屏——用于显示输出画面

网络设备——用于数据传输

兼顾运营商与地理位置的CDN服务器——用于加快系统实时响应速度

高IO性能服务器——用于实现分布式数据存储

高计算性能CPU服务器——用于运行系统中的算法模型

## 5.10计算机资源需求可追踪性

为满足系统运行安全性及用户使用需求，应有T级别DDoS防御网络带宽动态扩容能力，满足支持同时5000个以上UDP连接接入的用户需求。

## 5.11人员需求可追踪性

对于人员需求则是各阶段的专人负责，会对于各个阶段所需要的人员进行落实。

## 5.12培训需求可追踪性

培训需求主要由系统开发以及测试人员负责，对开发人员进行培训。

# 6尚未解决的问题

采集并用于测试的数据在系统初期可能并不充裕，用于目标识别的模型训练仍有提升空间。

目前系统所用服务器不足，无法容纳大用户量的访问。

目前对于银行卡的卡号定位并不精确。

目前无法自动识别出银行卡卡号的位数。

# 7注解

## 7.1专业术语解释

**卷积神经网络**

卷积神经网络仿造生物的视知觉（visual perception）机制构建，可以进行监督学习和非监督学习，其隐含层内的卷积核参数共享和层间连接的稀疏性使得卷积神经网络能够以较小的计算量对格点化（grid-like topology）特征，例如像素和音频进行学习、有稳定的效果且对数据没有额外的特征工程（feature engineering）要求。

我们打算采用3个卷积层 3个全连接层的CNN结构。第一个卷积层有 32个 7×7×3 滤波器，第二个卷积层 32个 7×7×32 滤波器，第三个卷积层有64个 5 × 5 × 32 滤波器。第一第二卷积层后面各用一个 2 × 2 最大池化，卷积和全连接层都使用 Rectified linear unit (ReLU) 激活响应函数。现代主流的机器学习库和界面，包括TensorFlow、Keras、Thenao、Microsoft-CNTK等都可以运行卷积神经网络算法。此外一些商用数值计算软件，例如MATLAB也有卷积神经网络的构建工具可用。

**多列卷积神经网络**

多阵列(Multi-Column)的卷积神经网络(Convolutional Neural Network，CNN)可以实现提取不同尺度的人头特征，将图像映射到其人群密度图上。每列CNN学习得到的特征可以自适应由于透视或图像分辨率引起的人/头大小的变化，并能在不需要输入图的透视先验情况下通过几何自适应的核来精确计算人群密度图。MCNN网络的每一列并行的子网络深度相同，但是滤波器的大小不同（大，中，小），因此每一列子网络的感受野不同，能够抓住不同大小人头的特征，最后将三列子网络的特征图做线性加权（由1x1的卷积完成）得到该图像的人群密度图，类似模型融合的思想。采用了2\*2的max-pooling和ReLU激活函数。

**OCR（Optical Character Recognition，[光学字符识别](https://baike.baidu.com/item/%E5%85%89%E5%AD%A6%E5%AD%97%E7%AC%A6%E8%AF%86%E5%88%AB/4162921" \t "https://baike.baidu.com/item/%E5%85%89%E5%AD%A6%E5%AD%97%E7%AC%A6%E8%AF%86%E5%88%AB/_blank)）**

OCR 是指电子设备（例如扫描仪或数码相机）检查纸上打印的字符，通过检测暗、亮的模式确定其形状，然后用字符识别方法将形状翻译成计算机文字的过程；即，针对印刷体字符，采用光学的方式将纸质文档中的文字转换成为黑白点阵的图像文件，并通过识别软件将图像中的文字转换成文本格式，供文字处理软件进一步编辑加工的技术。如何除错或利用辅助信息提高识别正确率，是OCR最重要的课题，ICR（Intelligent Character Recognition）的名词也因此而产生。衡量一个OCR系统性能好坏的主要指标有：拒识率、误识率、识别速度、用户界面的友好性，产品的稳定性，易用性及可行性等。

**图像去噪**

图像去噪是指减少数字图像中噪声的过程。现实中的数字图像在数字化和传输过程中常受到成像设备与外部环境噪声干扰等影响，称为含噪图像或噪声图像。

噪声是图象干扰的重要原因。一幅图象在实际应用中可能存在各种各样的噪声,这些噪声可能在传输中产生,也可能在量化等处理中产生。

## 7.2特殊名词解释和操作说明

**product backlog**

按优先顺序排列的一个产品需求列表。

Product Backlog为产品或项目的将做(To-do)列表。 Product Backlog并非将做的工作列表，而是所有对客户有价值的功能、特性、程序或技术改善等的将做列表。 此列表需被排出先后顺序，以方便每个Sprint的工作安排。

Scrum的参与者主要分Product Owner、Scrum Master和Scrum Team三种角色。 Product Backlogs的新增、修改和优先顺序的设定皆由Product Owner来决策与执行，决策前也有可能找Scrum Master来一起讨论。

类别(Category)：

Product Backlog的分类大致有Epic、Theme、User Story、Spike。

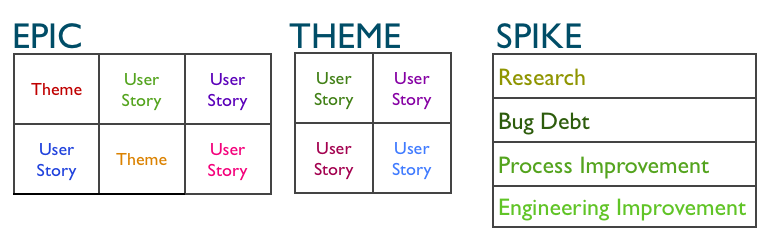


图5 Product Backlog说明图1

在初期，Product Owner对于有些功能可能只有比较粗略的概念，但还没到可以写User Story的阶段，此时只能先写Epic或Theme。以下为User Story、Theme、Epic的区别案例

|  |  |  |
| --- | --- | --- |
| **User Story** | **Theme** | **Epic** |
| 身为系统管理员，我必须能加系统使用者，才能让人有帐号登入 | 身为系统管理员, 我需要使用者管理功能以便决定谁可以使用系统哪些功能 | 身为系统管理员, 我需要各种管理功能, 让我能控制系统的各种动作 |

至于如何写User Story， 可参考以下User Story的写法公式 ：

**AS a <USER> , I need to <ACTION> in order to <REASON>**

根据执行过Scrum的经验分享，您也会发现有的团队在Product Backlog相关产品特性或功能方面， 不一定用User Story的方式来写,也有仅做功能叙述(Feature)者，其他也有将确定的变更(Change Request)或瑕疵(Bug)列入Product Backlogs中。是否允许这么做,团队讨论过后达成共识即可。

如果是小小的瑕疵，也可考虑是否列为Spike中其中一条。 通常经过几轮的Sprint，不再是单纯的把功能完成，在前面的Sprint完成的工作或发现的缺失，不在即将开始的Sprint中解决是不行的，但又觉得这些事很琐碎。 所以ScrumWizard设计了让Spike类的Backlog条目收集相关事项的功能键，让Product Owner点选未列入Backlog的瑕疵、变更、工作等平常发现即马上呈报的事项，方便Scrum团队估计此Spike需花的工，也方便将此Spike Backlog 条目选入将执行的Spint计画。

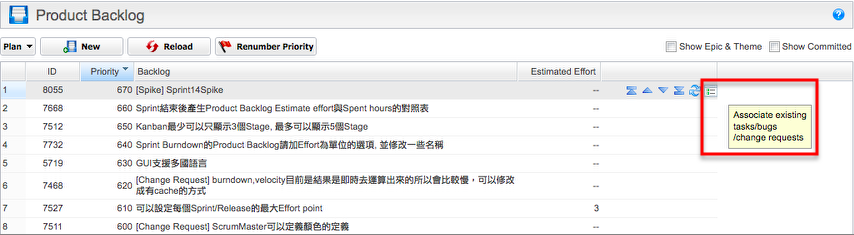


图6 Product Backlog说明图2

每个Product Backlog条目的表单都有类别(Category)栏位，类别(Category)的选项可就您的项目需求来修改。其中我们建议保留User Story、Theme、Epic、Spike, 其余可就团队需求做调整。

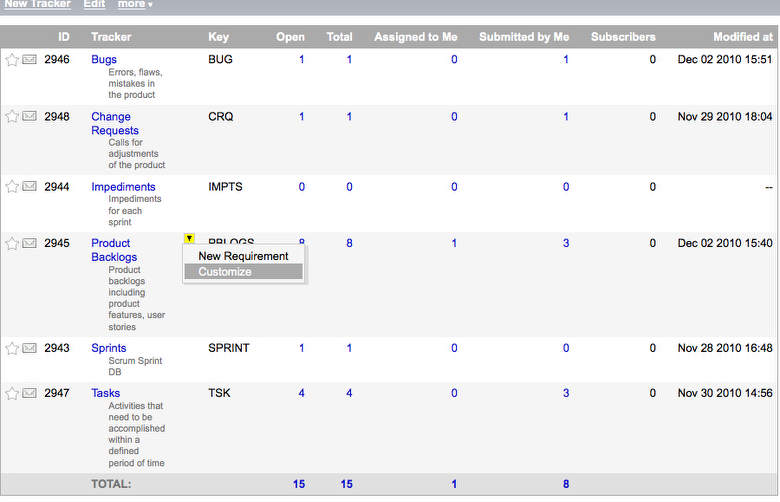
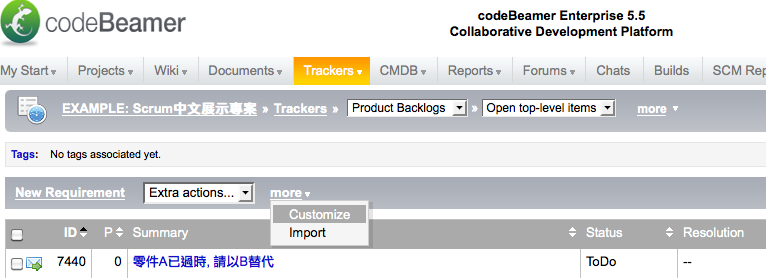
类别(Category)修改方式：到此项目的追踪(Trackers)的Product Backlogs的客制化(Customize)，然后到Choice中点Category修改即可。 

图7 Product Backlog说明图3

排序((Priority)：

以上先说明Backlog的类别(Category), 主要是因为有些Product Backlog并不适合参与排序。 排序的目的是便于将顺序前几名者安排到将执行的冲刺(Sprint)。 由于一个Sprint的时段建议2~3周，不大于一个月，且一个Backlog要能实际排入Sprint执行，基本上团队要对此Backlog有清楚的认知。 所以, 如果一个Backlog被归为Epic或Theme，将这类的Backlog排入Sprint计画很容易出问题。 所以ScrumWizart中ProductBacklog的原始设定，仅列出非Epic和Theme类别的Product Backlog 条目，也不列出已被排入冲刺(Sprint)的Product Backlog 条目。 只有当Epic或Theme随著冲刺(Sprint)进行，陆续拆成多个User Stories后,才有资格被安排到冲刺计画(Sprint Plan)。  
  
 每个Product Backlog条目皆可设定预计费工(Estimated Effort)和营业价值(Business Value)，此为两个帮助决定先后顺序(Priority)的参考栏位。 建议您参与开发的各部门成员对于Estimated Effort和Business Value的判断逻辑与方法能有一致的共识。 为了帮助Product Backlog的排序讨论，但又不想让信息一次展示太杂，ScrumWizard设计让Product Owner将鼠标移到栏位名称横条右侧的下引箭号时，即可出现columns名称选项，方便设定Product Backlog表中的展示栏位。

ScrumWizart也提供直觉的Priority顺序挪移的方法，可以直接拖拉，也可以将滑鼠挪到某Product Backlog项目，按此项目后方出现的箭头符号做上下排序。

每上下挪动排序后，只要点选Renumber，ScrumWizart将帮您把排序的好的Items以10等差来编列Priority的数字大小，以方便机动的排序。 如果您想看到所有Epic和Theme的项目，或已排入某冲刺(Sprint)的commited items，只要勾选右上方的Show Epic & Theme或Show Committed 旁的方格，您想看的items将出现。 如果Backlog条目已经结案(Close)了, 在Product Backlog中将不会再出现. 您如果要看所有的Product Backlog项目, 可到此项目的追踪(Trackers)的Product Backlog的Tracker, 选择All来展示所有条目, 不管是结案或未结案的都列于此以便筛选总览。

**OpenCV**

一个基于BSD许可（开源）发行的跨平台计算机视觉库，可以运行在Linux、Windows、Android和Mac OS操作系统上。它轻量级而且高效——由一系列 C 函数和少量 C++ 类构成，同时提供了Python、Ruby、MATLAB等语言的接口，实现了图像处理和计算机视觉方面的很多通用算法。

同时它轻量级而且高效——由一系列 C 函数和少量 C++ 类构成，同时提供了Python、Ruby、MATLAB等语言的接口，实现了图像处理和计算机视觉方面的很多通用算法。具体对比如下：

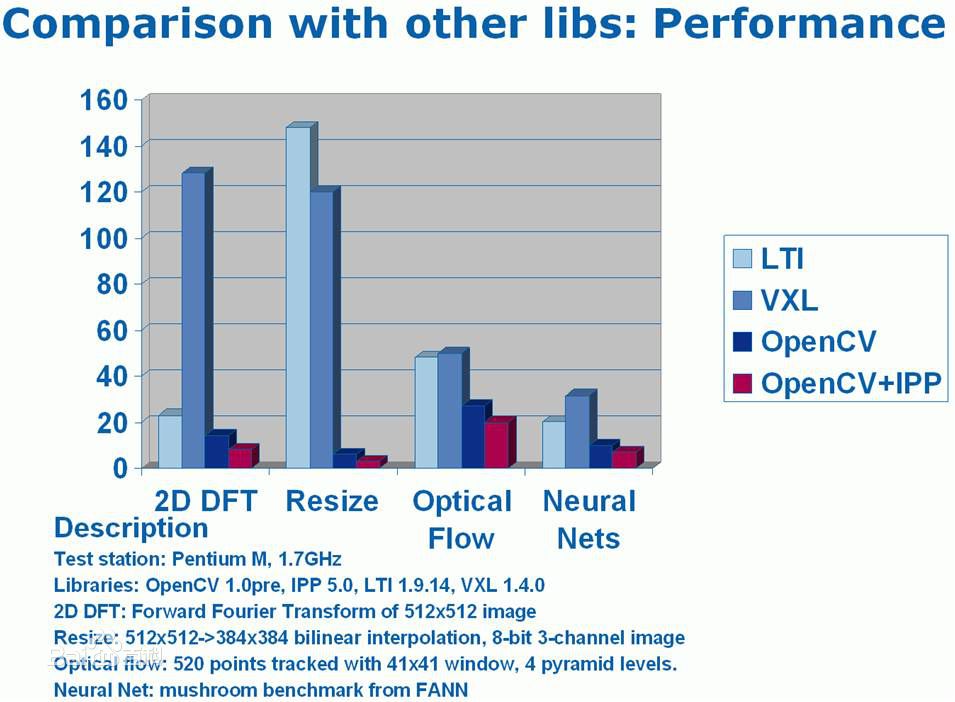


图8 OpenCV说明图

如图可见：

计算机视觉市场巨大而且持续增长，且这方面没有标准API，如今的计算机视觉软件大概有以下三种：

1、研究代码（慢，不稳定，独立并与其他库不兼容）

2、耗费很高的商业化工具（比如Halcon, [MATLAB](https://baike.baidu.com/item/MATLAB" \t "_blank)+[Simulink](https://baike.baidu.com/item/Simulink" \t "_blank)）

3、依赖硬件的一些特别的解决方案（比如[视频监控](https://baike.baidu.com/item/%E8%A7%86%E9%A2%91%E7%9B%91%E6%8E%A7" \t "_blank)，制造[控制系统](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6%E7%B3%BB%E7%BB%9F" \t "_blank)，医疗设备）这是如今的现状，而标准的API将简化计算机视觉程序和解决方案的开发，OpenCV致力于成为这样的标准API。

OpenCV致力于真实世界的实时应用，通过优化的C代码的编写对其执行速度带来了可观的提升，并且可以通过购买Intel的IPP高性能多媒体函数库（Integrated Performance Primitives）得到更快的处理速度。右图为OpenCV与当前其他主流视觉函数库的性能比较。

**图像平滑**

图像平滑是一种区域增强的算法，平滑算法有邻域平均法、中指滤波、边界保持类滤波等。在图像产生、传输和复制过程中，常常会因为多方面原因而被噪声干扰或出现数据丢失，降低了图像的质量（某一像素，如果它与周围像素点相比有明显的不同，则该点被噪声所感染）。这就需要对图像进行一定的增强处理以减小这些缺陷带来的影响。

1.简单平滑-邻域平均法

图像简单平滑是指通过邻域简单平均对图像进行平滑处理的方法，用这种方法在一定程度上消除原始图像中的噪声、降低原始图像对比度的作用。它利用卷积运算对图像邻域的像素灰度进行平均，从而达到减小图像中噪声影响、降低图像对比度的目的。

但邻域平均值主要缺点是在降低噪声的同时使图像变得模糊，特别在边缘和细节处，而且邻域越大，在去噪能力增强的同时模糊程度越严重。

2.高斯平滑

为了克服简单局部平均法的弊端(图像模糊)，目前已提出许多保持边缘、细节的局部平滑算法。它们的出发点都集中在如何选择邻域的大小、形状和方向、参数加平均及邻域各店的权重系数等。

图像高斯平滑也是邻域平均的思想对图像进行平滑的一种方法，在图像高斯平滑中，对图像进行平均时，不同位置的像素被赋予了不同的权重。

在图像简单平滑中，算法利用卷积模板逐一处理图像中每个像素，这一过程可以形象地比作对原始图像的像素一一进行过滤整理，在图像处理中把邻域像素逐一处理的算法过程称为滤波器。平滑线性滤波器的工作原理是利用模板对邻域内像素灰度进行加权平均，也称为均值滤波器。

高斯平滑与简单平滑不同，它在对邻域内像素进行平均时，给予不同位置的像素不同的权值，下图的所示的3\*3和5\*5领域的高斯模板。

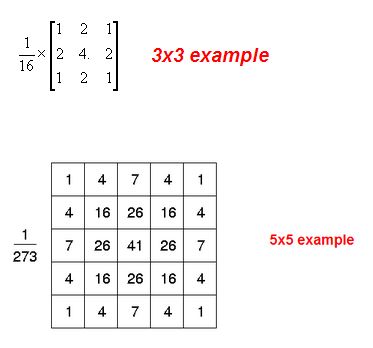


图9 高斯平滑说明图1

模板越靠近邻域中心位置，其权值越高。在图像细节进行模糊时，可以更多的保留图像总体的灰度分布特征。下图是常用的四个模板和matlab代码实现：

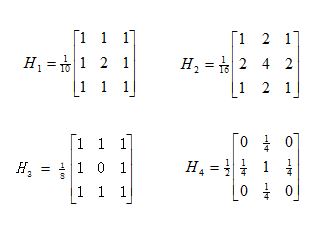


图9 高斯平滑说明图2

代码如下：

I1 = imread('blood1.tif');

I=imnoise(I1,‘salt & pepper’,0.04);                %对图像加椒盐噪声

imshow(I);

h1= [0.1 0.1 0.1; 0.1 0.2 0.1; 0.1 0.1 0.1];        %定义4种模板

h2=1/16.\*[1 2 1;2 4 2;1 2 1];

h3=1/8.\*[1 1 1;1 0 1;1 1 1];

h4=1/2.\*[0 1/4 0;1/4 1 1/4;0 1/4 0];

I2=filter2(h1,I);                                   %用4种模板进行滤波处理

I3=filter2(h2,I);

I4=filter2(h3,I);

I5=filter2(h4,I);

figure,imshow(I2,[])                                %显示处理结果

figure,imshow(I3,[])

figure,imshow(I4,[])

figure,imshow(I5,[])

运行效果如下图所示：

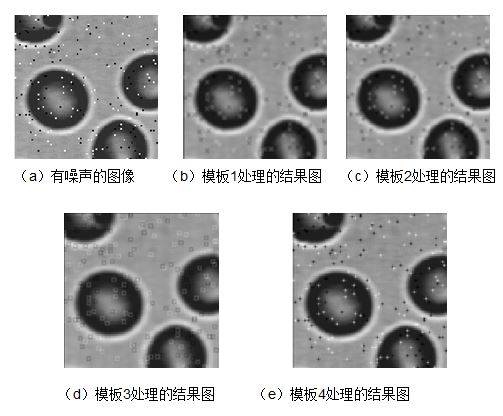


图10 高斯平滑说明图3

 3.中值滤波

在使用邻域平均法去噪的同时也使得边界变得模糊。而中值滤波是非线性的图像处理方法，在去噪的同时可以兼顾到边界信息的保留。

选一个含有奇数点的窗口W，将这个窗口在图像上扫描，把窗口中所含的像素点按灰度级的升或降序排列，取位于中间的灰度值来代替该点的灰度值。



例如选择滤波的窗口如下图，是一个一维的窗口，待处理像素的灰度取这个模板中灰度的中值，滤波过程如下：

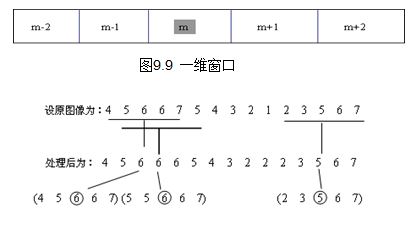


图11 中值滤波说明图1

常用的窗口还有方形、十字形、圆形和环形。不同形状的窗口产生不同的滤波效果，方形和圆形窗口适合外轮廓线较长的物体图像，而十字形窗口对有尖顶角状的图像效果好。

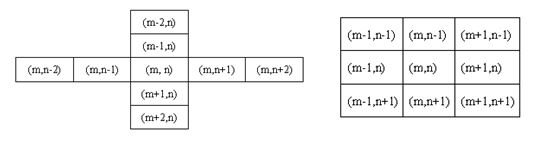


图12 中值滤波说明图2

中值滤波对于消除孤立点和线段的干扰十分有用，尤其是对于二进噪声，但对消除高斯噪声的影响效果不佳。对于一些细节较多的复杂图像，可以多次使用不同的中值滤波。

4.边界保持类滤波

K近邻均值滤波器(KNNF)是指在m\*m的窗口中，属于同一集合类的像素，它们的灰度值将高度相关。被处理的像素(对应于窗口中心的像素)可以用窗口内与中心像素灰度最接近的k个近邻像素的平均灰度来替代。步骤如下：

(1).作一个m\*m的作用模板

(2).在其中选择K个与待处理像素的灰度差为最小的像素

(3).用这K个像素的灰度均值替换掉原来的值

**灰度变换**

进行灰度处理，我们要保证图像信息尽可能少的丢失。同样在进行灰度变换前，我们也要对图像的信息进行统计，找出一个比较合理的灰度值，才能进行灰度变换。

**灰度均衡**

灰度变换后，就要进行灰度均衡可以根据灰度分布来进行灰度均衡。

**对比度增强**

将所要处理的区域和周围图像区域进一步拉开它们的对比度，使它们更加明显，主要通过像素的聚集来实现。

**Spark Streaming**

构建在Spark上处理Stream数据的框架，基本的原理是将Stream数据分成小的时间片段（几秒），以类似batch批量处理的方式来处理这小部分数据。Spark Streaming构建在Spark上，一方面是因为Spark的低延迟执行引擎（100ms+），虽然比不上专门的流式数据处理软件，也可以用于实时计算，另一方面相比基于Record的其它处理框架（如Storm），一部分窄依赖的RDD数据集可以从源数据重新计算达到容错处理目的。此外小批量处理的方式使得它可以同时兼容批量和实时数据处理的逻辑和算法。方便了一些需要历史数据和实时数据联合分析的特定应用场合。

**spark 生态及运行原理**

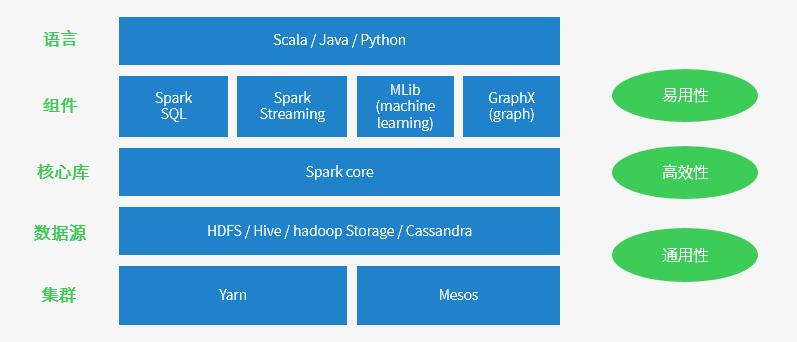


图13 spark说明图1

**Spark 特点**

* 运行速度快 => Spark拥有DAG执行引擎，支持在内存中对数据进行迭代计算。官方提供的数据表明，如果数据由磁盘读取，速度是Hadoop MapReduce的10倍以上，如果数据从内存中读取，速度可以高达100多倍。
* 适用场景广泛 => 大数据分析统计，实时数据处理，图计算及机器学习
* 易用性 => 编写简单，支持80种以上的高级算子，支持多种语言，数据源丰富，可部署在多种集群中
* 容错性高 => Spark引进了弹性分布式数据集RDD (Resilient Distributed Dataset) 的抽象，它是分布在一组节点中的只读对象集合，这些集合是弹性的，如果数据集一部分丢失，则可以根据“血统”（即充许基于数据衍生过程）对它们进行重建。另外在RDD计算时可以通过CheckPoint来实现容错，而CheckPoint有两种方式：CheckPoint Data，和Logging The Updates，用户可以控制采用哪种方式来实现容错。

**Spark的适用场景**

目前大数据处理场景有以下几个类型：

* 复杂的批量处理（Batch Data Processing），偏重点在于处理海量数据的能力，至于处理速度可忍受，通常的时间可能是在数十分钟到数小时；
* 基于历史数据的交互式查询（Interactive Query），通常的时间在数十秒到数十分钟之间
* 基于实时数据流的数据处理（Streaming Data Processing），通常在数百毫秒到数秒之间

**Spark成功案例**

目前大数据在互联网公司主要应用在广告、报表、推荐系统等业务上。在广告业务方面需要大数据做应用分析、效果分析、定向优化等，在推荐系统方面则需要大数据优化相关排名、个性化推荐以及热点点击分析等。这些应用场景的普遍特点是计算量大、效率要求高。比如：腾讯 / yahoo / 淘宝 / 优酷土豆

**spark运行架构**

**spark基础运行架构如下所示：**

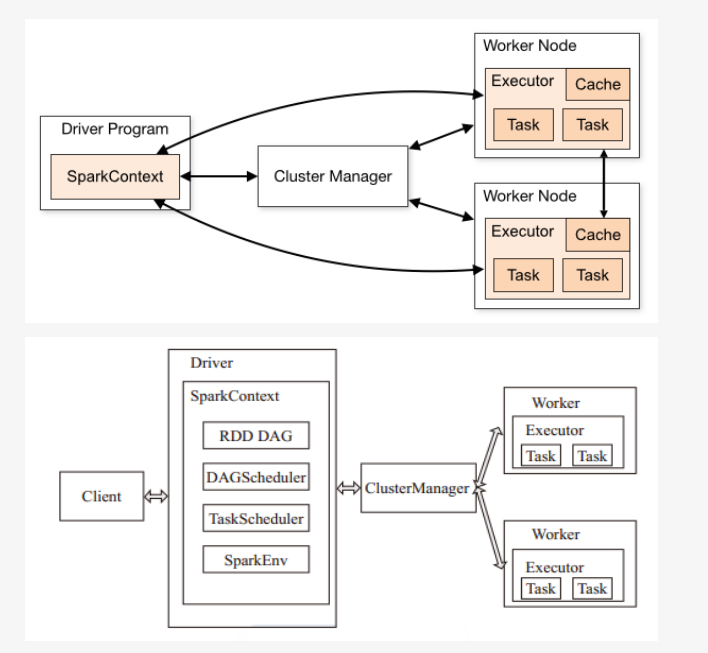


图14 spark说明图2

**spark结合yarn集群背后的运行流程如下所示：**

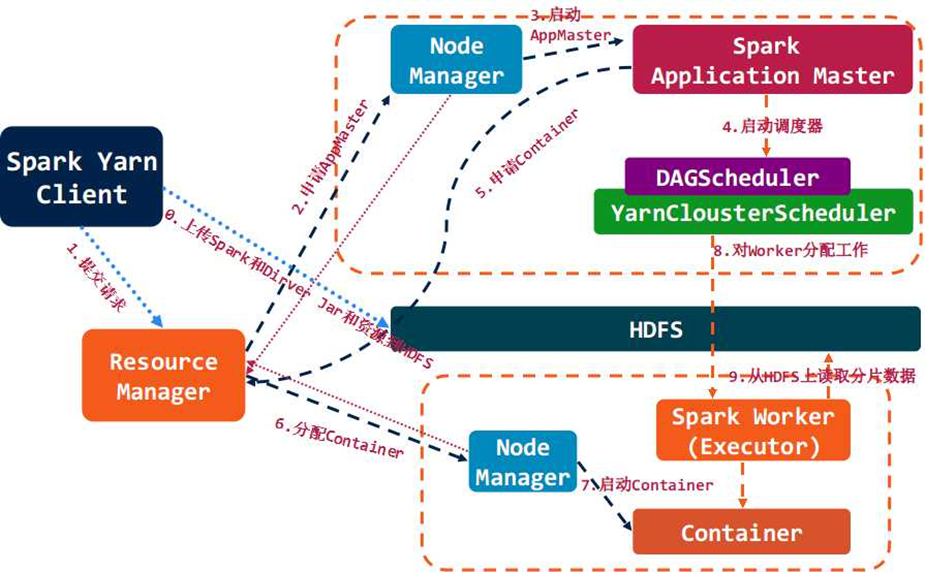


图15 spark说明图3

**spark 运行流程：**

Spark架构采用了分布式计算中的Master-Slave模型。Master是对应集群中的含有Master进程的节点，Slave是集群中含有Worker进程的节点。

Master作为整个集群的控制器，负责整个集群的正常运行；

Worker相当于计算节点，接收主节点命令与进行状态汇报；

Executor负责任务的执行；

Client作为用户的客户端负责提交应用；

Driver负责控制一个应用的执行。

Spark集群部署后，需要在主节点和从节点分别启动Master进程和Worker进程，对整个集群进行控制。在一个Spark应用的执行过程中，Driver和Worker是两个重要角色。Driver 程序是应用逻辑执行的起点，负责作业的调度，即Task任务的分发，而多个Worker用来管理计算节点和创建Executor并行处理任务。在执行阶段，Driver会将Task和Task所依赖的file和jar序列化后传递给对应的Worker机器，同时Executor对相应数据分区的任务进行处理。

Excecutor /Task 每个程序自有，不同程序互相隔离，task多线程并行

集群对Spark透明，Spark只要能获取相关节点和进程

Driver 与Executor保持通信，协作处理

**三种集群模式：**

1.Standalone 独立集群

2.Mesos, apache mesos

3.Yarn, hadoop yarn

**基本概念：**

Application =>Spark的应用程序，包含一个Driver program和若干Executor

SparkContext => Spark应用程序的入口，负责调度各个运算资源，协调各个Worker Node上的Executor

Driver Program => 运行Application的main()函数并且创建SparkContext

Executor => 是为Application运行在Worker node上的一个进程，该进程负责运行Task，并且负责将数据存在内存或者磁盘上。每个Application都会申请各自的Executor来处理任务

Cluster Manager =>在集群上获取资源的外部服务 (例如：Standalone、Mesos、Yarn)

Worker Node => 集群中任何可以运行Application代码的节点，运行一个或多个Executor进程

Task => 运行在Executor上的工作单元

Job => SparkContext提交的具体Action操作，常和Action对应

Stage => 每个Job会被拆分很多组task，每组任务被称为Stage，也称TaskSet

RDD => 是Resilient distributed datasets的简称，中文为弹性分布式数据集;是Spark最核心的模块和类

DAGScheduler => 根据Job构建基于Stage的DAG，并提交Stage给TaskScheduler

TaskScheduler => 将Taskset提交给Worker node集群运行并返回结果

Transformations => 是Spark API的一种类型，Transformation返回值还是一个RDD，所有的Transformation采用的都是懒策略，如果只是将Transformation提交是不会执行计算的

Action => 是Spark API的一种类型，Action返回值不是一个RDD，而是一个scala集合；计算只有在Action被提交的时候计算才被触发。

# 附录A：系统安全需求确认

**系统保密性：**用户只能上传图片并在点击识别按钮后获取识别结果。

**系统完整性：**也就是说信息必须以其原形被授权的用户所用。

**漏洞检测和安全风险评估：**识别检测对象的系统资源，分析这一资源被攻击的可能指数，了解支撑系统本身的脆弱性，评估所有存在的安全风险

**可用性和抗毁性：**设备备份机制、容错机制，防止在系统出现单点失败时，系统的备份机制保证系统的正常运行。

**系统防病毒：**网络防病毒系统应基于策略集中管理的方式，使得分布式的企业级病毒防护不再困难，而且提供病毒定义的实时自动更新功能。

# 附录B：需求确认

|  |  |
| --- | --- |
| **需求评审报告摘要** | |
| 需求文档 | 《LeetCode-银行卡识别系统软件开发计划书》，作者LeetCode，完成日期2019.4.10 |
| 需求评审报告 | 《关于银行卡识别系统软件需求的评审报告》 2019.4.14 |
| 评审结论 | [ ] 工作成果合格，“无需修改”或者“需要轻微修改但不必再审核”。  [√] 工作成果基本合格，需要做少量的修改，之后通过审核即可。  [ ] 工作成果不合格，需要做出比较大的修改，之后必须重新对其评审。 |
| 评审小组成员 | LeetCode |