# **北京信息科技大学课程报告（2020）**

报告题目：开源项目分析

课程名称：开源软件开发（2020）

任课教师：李宁

姓名：陈永鹏

学号：2019011340

学院专业：计算机学院软件工程

## **报告要求**

分析一个典型的开源软件及其开发过程，撰写并完成本分析报告。要求：

1）选择一个开源项目进行分析。选择的来源可以是：

* 课程讲义
* Github 项目  
  <https://github.com/trending>

<https://github.com/topics>

* 其他渠道

2）选择的开源项目必须是活跃的，在最近的一年内有更新；项目的大小合适，包含报告要求的那些内容。

3）从开源项目的官方网站获得内容，也可以从网络搜索的参考资料获得内容。使用他人的内容请注明引用。

4）内容需要经过自己的整理和组织。报告各项内容完整，格式清晰完整，标点正确，无错别字。

5） 每位同学需要选择不同的项目进行分析。请选好项目的同学在课堂派的“话题”里找到帖子“开源项目分析选题”，跟帖写下自己选好的项目。以声明顺序为准，后来的同学请自己看看项目是否被人选择过了。请自行注意是否与其他同学的题目重复。重复者将不能得分。

6）报告需下载附件文件进行填写后，将作业转换为pdf格式，作为附件上传。

7）请自行注意选择的软件是否开源软件。如果不是开源软件将不能得分。

## **项目概况**

填写以下表格，如果是不存在的内容可以留空。

文中空格区域仅为示范，内容多请扩充。

|  |  |
| --- | --- |
| 项目网站 | https://gitee.com/admin\_yu/yx-image-recognition |
| 管理组织 | Gitee |
| 代码托管 | gitee |
| 最新版本 | * jdk 1.8.61+ * maven 3.0+ * opencv 4.0.1 ； javacpp1.4.4；opencv-platform 4.0.1-1.4.4 * spring boot 2.1.5.RELEASE * yx-image-recognition 1.0.0版本 |
| 邮件列表 |  |
| 维基介绍 | 基于spring boot + maven + opencv 实现的图像深度学习Demo项目，包含车牌识别、人脸识别等功能，贯穿样本处理、模型训练、图像处理、对象检测、对象识别等技术 |
| 缺陷追踪系统 |  |
| 许可证 |  |
| 项目名称 | **image-recognition** 基于spring boot + maven + opencv 实现的****Demo教程项目**** |

## **项目功能**

* ****蓝、绿、黄车牌检测及车牌号码识别****
* ****网上常见的轮廓提取车牌算法JAVA实现****
* ****hsv色彩分割提取车牌算法JAVA实现****
* ****harrcascade特征识别算法 JAVA实现****
* ****基于svm算法的车牌检测训练JAVA实现****
* ****基于ann算法的车牌号码识别训练JAVA实现****
* ****人脸检测**** 接下来将实现人脸识别
* 图片处理工具，目前实现了HSV色彩切割，后续将添加更多使用的图片处理工具，用于辅助算法优化

## **项目发展**

* 这是一个基于spring boot + maven + opencv 实现的****Demo教程项目****
* ****贯穿样本处理、模型训练、图像处理、对象检测、对象识别等技术点****
* 以学习交流为目的，代码注释超多，文档也在逐步完善
* java语言的深度学习项目，在整个开源社区来说都相对较少；拥有完整的训练、检测、识别过程的开源项目更是少之又少！！
* 本人目前也正在学习图片识别相关技术

## **项目组织**

基于spring boot + maven + opencv 实现的图像深度学习Demo项目，包含车牌识别、人脸识别等功能，贯穿样本处理、模型训练、图像处理、对象检测、对象识别等技术点

### **JNI**

使用JNI调用.dll/.so共享类库是非常非常麻烦和痛苦的。

* 首先需要另外使用C/C++语言写一个\*\*.dll/.so共享库\*\*，使用SUN规定的数据结构替代C语言的数据结构，****调用已有的dll/so中公布的函数****
* 然后再在Java中载入这个适配器dll/so，再编写****Java native函数****作为dll中函数的代理

[java调用C++ DLL库方法](https://www.cnblogs.com/xiaocainiao2hao/p/5619862.html)

1、新建一个Java类Java2cpp.java 声明java接口对应的c++接口

2、根据java类，生成com\_test\_jni\_Java2cpp.h头文件

3、在VS中新建项目win32项目，导入上述头文件

4、新建头文件 dllApi.h

5、新建dllApi.cpp实现dllApi.h头文件声明的方法

6、新建javaCallCpp.cpp，实现com\_test\_jni\_Java2cpp.h声明方法；方法体调用步骤4声明的方法

7、根据javaCallCpp.cpp 生成 javaCallCpp.dll

8、把生成的javaCallCpp.dll拷进JRE安装路径下的bin文件夹内，运行java程序即可

* 使用案例：****java调用opencv 动态链接库****

1. opencv本身就是一个使用C++编写的开源软件，即: 已有的dll/so库；
2. java中调用则需要一个适配器dll/so，即：opencv\_java401.dll 动态链接库、共享库；
3. Java native函数作为dll中函数的代理，即：opencv-401.jar依赖包里的private static native....接口

### **JNA**

* JNA全称Java Native Access，是一个建立在经典的JNI技术之上的Java开源框架; JNI终结者
* JNA提供一组Java工具类用于在运行期动态访问系统本地库（native library：如Window的dll）而不需要编写任何Native/JNI代码。开发人员只要在一个java接口中描述目标native library的函数与结构，JNA将自动实现Java接口到native function的映射
* JNA定义的接口一般继承com.sun.jna.Library接口，如果dll文件中的函数是以****stdcall方式****输出函数，那么，该接口就应该继承com.sun.jna.win32.StdCallLibrary接口
* 难点：编程语言之间的数据类型不一致

[Getting Started with JNA](https://github.com/java-native-access/jna/blob/master/www/GettingStarted.md)

**import** com.sun.jna.Library;**import** com.sun.jna.Native;**import** com.sun.jna.Platform;**public** **class** **HelloWorld** {

**public** **interface** **CLibrary** **extends** **Library** {

CLibrary INSTANCE = (CLibrary)Native.load((Platform.isWindows() ? "msvcrt" : "c"), CLibrary.class);

**void** **printf**(String format, Object... args);

}

**public** **static** **void** **main**(String[] args) {

CLibrary.INSTANCE.printf("Hello, World\n");

**for** (**int** i=0;i < args.length;i++) {

CLibrary.INSTANCE.printf("Argument %d: %s\n", i, args[i]);

}

}

}

@FieldOrder({ "wYear", "wMonth", "wDayOfWeek", "wDay", "wHour", "wMinute", "wSecond", "wMilliseconds" })**public** **static** **class** **SYSTEMTIME** **extends** **Structure** {

**public** **short** wYear;

**public** **short** wMonth;

**public** **short** wDayOfWeek;

**public** **short** wDay;

**public** **short** wHour;

**public** **short** wMinute;

**public** **short** wSecond;

**public** **short** wMilliseconds;

}**void** **GetSystemTime**(SYSTEMTIME result);

* 使用案例：java通过jna调用海康sdk

<dependency>

<groupId>com.sun.jna</groupId>

<artifactId>jna</artifactId>

<version>3.0.9</version></dependency>

**public** **interface** **HCNetSDK** **extends** **StdCallLibrary** {

String path = (HCNetSDK.class.getResource("/").getPath()+"HCNetSDK.dll").substring(1).replace("/", "\\");

*//加载dll库文件 // 难点在于要能成功的找到dll*

HCNetSDK INSTANCE = (HCNetSDK) Native.loadLibrary(path, HCNetSDK.class);

*// 结构体*

**public** **static** **class** **NET\_DVR\_USER** **extends** **Structure** {*//DVR用户参数*

**public** **int** dwSize;

**public** NET\_DVR\_USER\_INFO[] struUser = **new** NET\_DVR\_USER\_INFO[MAX\_USERNUM];

}

*// java2cpp 代理接口*

**int** **NET\_DVR\_GetUpgradeProgress**(**int** lUpgradeHandle);

**boolean** **NET\_DVR\_CloseUpgradeHandle**(**int** lUpgradeHandle);

}

## **项目结构**

* B/S 架构，前端html + requireJS，后端java
* 数据库使用 sqlite3.0
* 接口文档使用swagger 2.0

## **参考文献**

* 本人参考了EasyPR C++项目、以及fan-wenjie的EasyPR-Java项目；同时查阅了部分opencv官方4.0.1版本C++的源码，结合个人对java语言的理解，整理出当前项目
* liuruoze/EasyPR：<https://gitee.com/easypr/EasyPR?_from=gitee_search>
* fan-wenjie/EasyPR-Java： <https://github.com/fan-wenjie/EasyPR-Java>
* opencv官方： <https://opencv.org/>