**基于Lire的分析与扩展**

**扩展与展示计划进度**

Version 1.0

小组成员：

刘少凡

宋昱材

吴沂楠

黄飞

**版本变更记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 变更时间 | 修改人 | 审核人 | 备注 |
| 1.0 | 20170516 | 刘少凡 |  | 从原1.3文档中分离出计划进度相关内容单独写在此文档 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 具体分工 4](#_Toc482690924)

[1.1 第9周开发任务分配 4](#_Toc482690925)

[1.2 第10周开发任务分配 4](#_Toc482690926)

[2实现进度 5](#_Toc482690927)

[2.1 第9周实现进度 5](#_Toc482690928)

[2.2 第10周实现进度 9](#_Toc482690929)

# 1 具体分工

## 1.1 第9周开发任务分配

（1）Caffe环境配置

负责人：宋昱材

工作内容：在Ubuntu 14.04操作系统上完成对Caffe框架的搭建。

（2）Lire代码修改

负责人：刘少凡、黄飞

工作内容：完成继承LireFeature接口的CNN类的具体方法编写。

（3）界面编写

负责人：吴沂楠

工作内容：完成主界面、入库、检索界面的编写

## 1.2 第10周开发任务分配

（1）CNN模型调用代码编写

负责人：宋昱材

工作内容：完成调用CNN模型的C++代码编写，实现JNI机制。与其他组员配合实现系统整合和调试。

（2）Lire代码修改与系统整合

负责人：刘少凡、黄飞

工作内容：完成GetCNN类的相关代码编写，完成入库和检索模块相关代码的修改。与其他组员配合实现系统整合和调试。

（2）界面编写

负责人：吴沂楠

工作内容：完成检索结果展示界面的编写。与其他组员配合实现系统整合和调试。

# 2实现进度

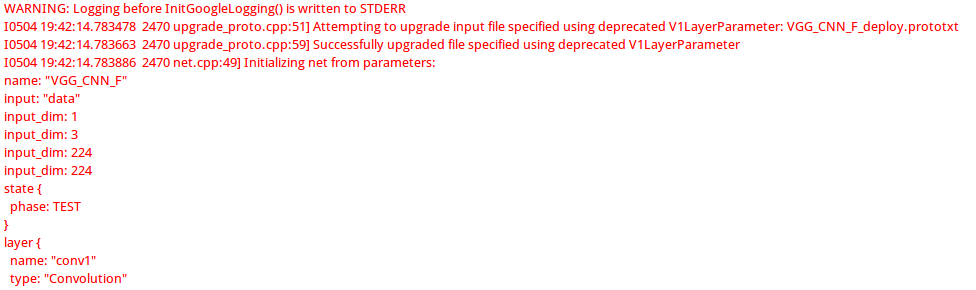
## 2.1 第9周实现进度

（1）Caffe环境配置

负责人：宋昱材

* 实现内容
* 完成Caffe框架的搭建

在Ubuntu 14.04上进行了Caffe框架的搭建，并进行了调用模型的测试，如图5.1所示，为Caffe加载VGG-F模型后的日志记录。



* 确定CNN模型

另一个深度学习框架MatConvNet总结了近几年的多种CNNs模型在相同硬件条件下处理ILSVRC 2012测试数据集的准确率与时间效率，其中部分数据如表5.1所示。

表5.1中的alexnet模型是最早提出的CNNs模型。vgg-f模型沿用了alexnet模型的8层基础架构，并通过调整卷积核的尺寸和数量，改变参数数量，使得模型在保持并一定程度上提升分类准确率的情况下，很大程度提高了处理图片的时间效率。

模型在实际运行过程中处理图片的时间复杂度直接影响实际应用系统运行时特征提取的时间效率，因此模型的时间复杂度不能过高。

基于以上原因，确定了基于8层基本CNNs结构的、时间复杂度较低的VGG-F模型作为项目所选择的模型。

**表 5.0‑1 各CNNs模型准确率与时间效率对比**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模型名称 | 提出时间 | 模型层数 | Top-1错误率 | Top-5错误率 | 每秒处理图片量 |
| resnet-50[[26](#_ENREF_26)] | 2015 | 50 | 24.6 | 7.7 | 315.3 |
| resnet-101[[26](#_ENREF_26)] | 2015 | 101 | 23.4 | 7.0 | 212.7 |
| resnet-152[[26](#_ENREF_26)] | 2015 | 152 | 23.0 | 6.7 | 156.6 |
| vgg-verydeep-19[[14](#_ENREF_14)] | 2014 | 19 | 28.7 | 9.9 | 154.5 |
| vgg-verydeep-16[[14](#_ENREF_14)] | 2014 | 16 | 28.5 | 9.9 | 183.1 |
| googlenet[[27](#_ENREF_27)] | 2014 | 22 | 34.2 | 12.9 | 501.8 |
| vgg-f[[28](#_ENREF_28)] | 2013 | 8 | 41.1 | 18.8 | 793.1 |
| alexnet[[13](#_ENREF_13)] | 2012 | 8 | 42.6 | 19.6 | 382.4 |

注：表中数据来源为<http://www.vlfeat.org/matconvnet/pretrained/>。

* 下周工作

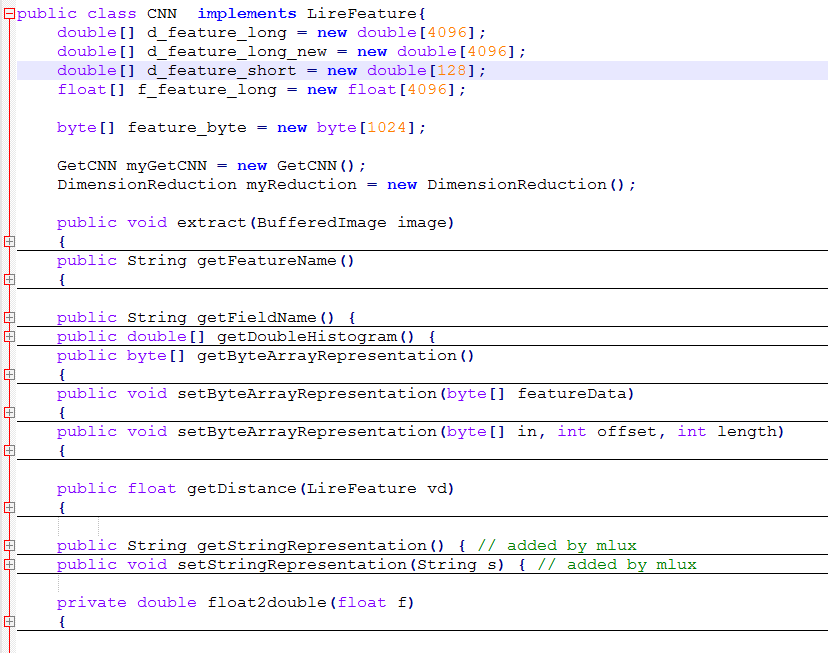
完成调用CNN模型的C++代码编写，实现JNI机制。与其他组员配合实现系统整合和调试。

（2）Lire代码修改

负责人：刘少凡、黄飞

* 实现内容

通过继承LireFeature接口，实现了CNN类，主要工作为实现如图所示的相关方法。



* 下周工作

完成GetCNN类的相关代码编写，完成入库和检索模块相关代码的修改。与其他组员配合实现系统整合和调试。

（3）界面编写

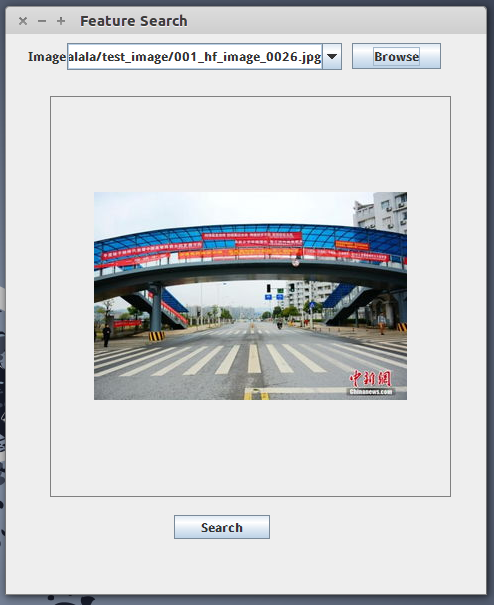
负责人：吴沂楠

* 实现内容
* 主界面



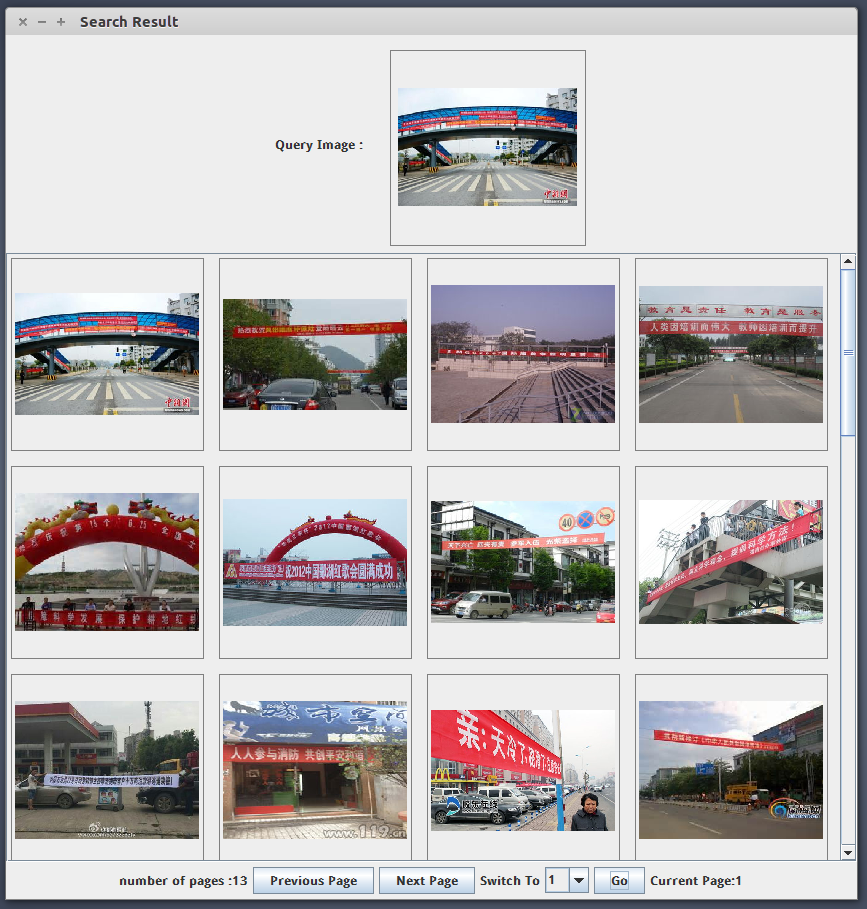
通过点击检索和入库按钮，分别进入检索界面和入库界面

* 检索界面



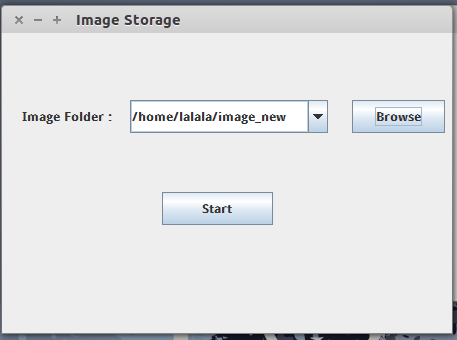
点击Browse按钮，弹出文件浏览窗口，用户点击文件夹进入存放检索图片的位置，文件浏览窗口自动过滤只保留jpg格式图片并显示在文件浏览窗口中，用户从中选择检索示例图片，点击检索按钮，检索结果由结果展示结果显示。

* 检索结果展示界面



界面上方显示用户输入的实例图片，下方显示检索结果。

* 入库界面



用户输入入库图片文件夹的路径，点击开始，进行入库。

* 下周工作

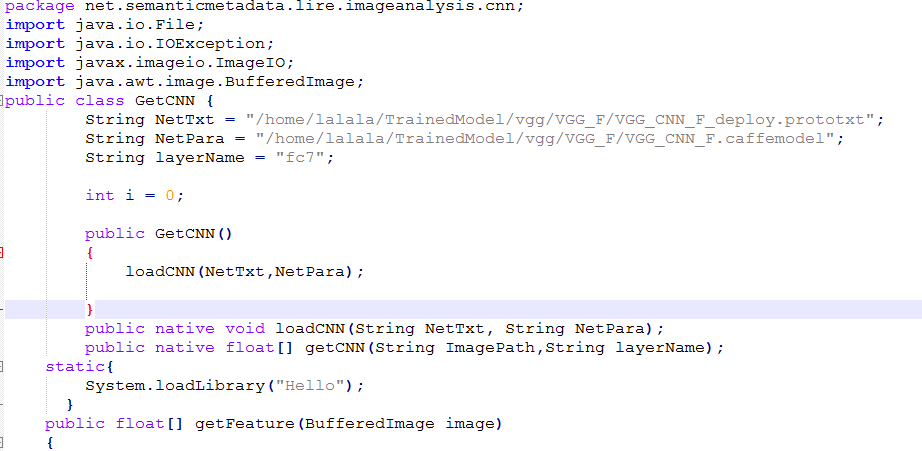
协助宋昱材完成调用CNN模型的C++代码编写，实现JNI机制。与其他组员配合实现系统整合和调试。

## 2.2 第10周实现进度

（1）GetCNN类实现

负责人：刘少凡、黄飞

实现GetCNN类，GetCNN类主要实现两个native方法，loadCNN和getCNN方法。相关代码如图所示。



loadCNN方法输入模型定义文件路径和模型参数文件路径，通过C++代码，将CNN模型加载到内存中。

getCNN方法输入图片路径和层数，通过C++代码，调用内存中的CNN模型提取图像特征。

getFeature方法为提供给CNN类的外部接口，CNN类通过getFeature方法调用getCNN方法提取特征。

调用javah命令，生成GetCNN类对应的C++头文件。

修改图像入库和图像检索的相关代码，增加CNN特征相关的接口。

（2）C++代码实现

负责人：宋昱材、吴沂楠

通过包含GetCNN类生成的头文件，实现C++的函数，主要为两个函数，即GetCNN类中loadCNN方法和getCNN方法对应的C++函数。

Java\_net\_semanticmetadata\_lire\_imageanalysis\_cnn\_GetCNN\_loadCNN方法通过调用Caffe的模型操作相关接口实现对模型配置文件和模型参数文件的读取和加载。

Java\_net\_semanticmetadata\_lire\_imageanalysis\_cnn\_GetCNN\_getCNN方法通过调用Caffe的模型测试相关接口，输入图片进行模型运算，从内存中读取运算结果，得到图像特征。



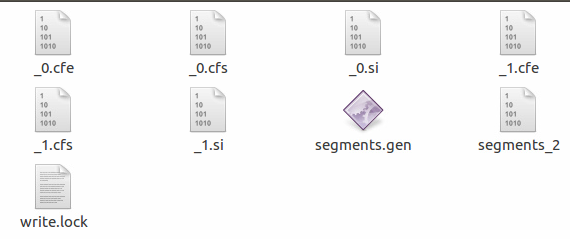
C++代码完成后，将C++代码编译生成动态链接库，供java调用。

（3）系统整合

负责人：刘少凡、宋昱材、吴沂楠、黄飞

将Lire代码与C++代码生成的动态链接库进行整合，实现系统的JNI机制。并对系统进行简单的测试，确保运行成功。

入库成功后，设置的保存索引的文件夹目录下会出现新的索引文件，如图，其中\_0开头的是第一次入库生成的索引，\_1开头的是第二次入库生成的索引。



检索完成后，会依序显示检索结果，如图。

