

基于 Lire 的分析与扩展 扩展与展示实现方案

Version 1.5

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	20170426	宋昱材	刘少凡 吴沂楠 黄飞	初稿
1.1	20170502	宋昱材	刘少凡 吴沂楠 黄飞	增加具体分工
1.2	20170505	宋昱材	刘少凡 吴沂楠 黄飞	更新第9周实现进度
1.3	20170510	吴沂楠 刘少凡 黄飞 宋昱材		增添类图、顺序图描述系统静态结构与人机交互过程，更新第10周进度
1.4	20170516	刘少凡		将计划进度方面内容从该文档中分离，只保留设计实现方案
1.5	20170518	黄飞 刘少凡		修改入库和检索的时序图，加入后台的时序部分，修改类图

目录

- 1 改进目标.....4
- 2 工作内容.....4
 - 2.1 Lire 源码修改4
 - 2.2 Caffe 环境配置和模型调用.....4
 - 2.4 JNI 实现.....5
 - 2.3 界面实现.....5
- 3 实现方案.....5
 - 3.1 Lire 源码修改5
 - 3.2 Caffe 环境配置和模型调用.....6
 - 3.3 JNI 实现.....6
 - 3.3.1 Java 端6
 - 3.3.2 C++端6
 - 3.4 界面实现.....7
 - 3.5 系统类图.....8
 - 3.5.1 特征提取模块.....8
 - 3.5.2 图像入库与图像检索实现.....8
 - 3.5.3 界面实现.....10
 - 3.6 系统运行过程.....11
 - 3.6.1 图像入库过程.....11
 - 3.6.2 图像检索过程.....12

1 改进目标

作为一种开源框架，Lire 的代码对开发者完全透明，程序代码具备简明、方便和清晰的构架设计与函数接口来方便用户的使用。而且，Lire 的实现框架使得开发者可以方便地引入新的技术、算法或模块，以满足开发人员的不同需求。因此 Lire 具备良好的可修改性或可扩展性。

项目的改进目标基于上述的可修改性。

项目计划在 Lire 中增加一种新的特征提取方法——CNN 特征。CNN（Convolutional Neural Network），即卷积神经网络。项目计划利用一个已训练好的面向图像分类任务的 CNN 模型，将其作为特征提取工具，从模型中提取某一层输出作为图像特征。

这种尝试的出发点基于实际开发中时常会出现的场景，即 CBIR 系统开发者计划使用 Lire 工具包进行系统开发，但 Lire 工具包中并未实现开发者所希望使用的图像特征，因此需要向工具包中扩展该图像特征。项目站在 CBIR 系统开发者角度，对 Lire 针对特定开发目标进行扩展。

2 工作内容

2.1 Lire 源码修改

Lire 的特征提取方法的具体实现在 imageanalysis 包中，通过 LireFeature 接口定义了特征提取类需要实现的方法，具体的特征提取类如 CEDD 等通过继承该接口进行具体的实现。因此需要通过继承 LireFeature 实现新的 CNN 类。

另外，要在图像入库和图像检索模块增加 CNN 特征类的接口。

2.2 Caffe 环境配置和模型调用

深度学习框架 Caffe 的运行环境需要依赖多种工具和库的支持，因此需要对操作系统的环境进行配置。

编写 C++ 代码，实现对模型的调用。C++ 代码主要实现两个功能，一是读取模型配置文件和参数文件将模型加载到内存中；二是将需要提取特征的图片输入模型，获得模型输出结果。

2.4 JNI 实现

由于 Lire 基于 Java 实现，Caffe 模型调用基于 C++实现，因此需要利用 java 的 JNI 机制实现 Lire 对 Caffe 的调用。

2.3 界面实现

前端界面主要包括图像主界面、图像入库界面、图像检索界面和检索结果界面。

3 实现方案

3.1 Lire 源码修改

imageanalysis 包中的 LireFeature 接口定义了图像特征类需要实现的方法，具体内容如下所示：

```
package net.semanticmetadata.lire.imageanalysis;
import java.awt.image.BufferedImage;
public interface LireFeature extends Histogram {
    //获取特征名称
    public String getFeatureName();
    //获取字段名
    public String getFieldName();
    //提取特征
    public void extract(BufferedImage image);
    //获取图像特征的Byte表示
    public byte[] getByteArrayRepresentation();
    //重置图像特征
    public void setByteArrayRepresentation(byte[] featureData);
    //有位移和长度的重置图像特征
    public void setByteArrayRepresentation(byte[] featureData, int
offset, int length);
    //获取图像特征
    public double[] getDoubleHistogram();
    //获取该特征与输入特征的距离
    float getDistance(LireFeature feature);
    //获取特征的字符串表示
    java.lang.String getStringRepresentation();
    //以字符串重置图像特征
```

```
void setStringRepresentation(java.lang.String featureVector);}
```

通过继承 LireFeature 接口，实现 CNN 类。

另外，在 Lire 工具包的 DocumentBuilder.java、DocumentBuilderFactory.java、ImageSearcherFactory.java 和 ImageSearcher.java 中增加相应的构造方法实现和字段添加，实现对索引构造、特征存储和特征搜索代码的调用。

3.2 Caffe 环境配置和模型调用

项目计划在 Ubuntu 14.04 操作系统上配置 Caffe，依据 Caffe 文档依次安装依赖库，实现 Caffe 的安装。

通过 C++代码调用 Caffe 的相关接口实现加载模型和调用模型的相关功能。

3.3 JNI 实现

3.3.1 Java 端

Java 端实现 GetCNN 类，通过调用 native 的 C++方法实现对 Caffe 模型的调用。GetCNN 类应当包含以下成员变量和方法：

表 3.1 GetCNN 类成员变量与方法作用

名称	变量/方法	类型	作用
NetTxt	成员变量	String	记录 CNNs 模型定义文件路径
NetPara	成员变量	String	记录 CNNs 模型参数文件路径
LayerName	成员变量	String	记录 CNNs 模型层名
loadCNN	公共方法	void	JNI 调用本地 Caffe 代码实现 CNNs 模型加载
getCNN	公共方法	float[]	JNI 调用本地 Caffe 代码实现特征提取
getFeature	公共方法	float[]	共外部调用的图像特征提取接口

其中 loadCNN 和 getCNN 为 native 方法，通过调用本地 C++代码实现。通过 javah 命令生成 C++所需的.h 头文，供 C++代码实现时包含。

3.3.2 C++端

C++代码包含 java 生成的头文件，实现头文件中定义的具体方法，将代码生

成动态链接库，供 java 代码调用。

3.4 界面实现

前端界面主要包括程序主界面、入库界面、图像检索界面和检索结果界面。
用户操作程序界面的流程如图 3.1 所示。

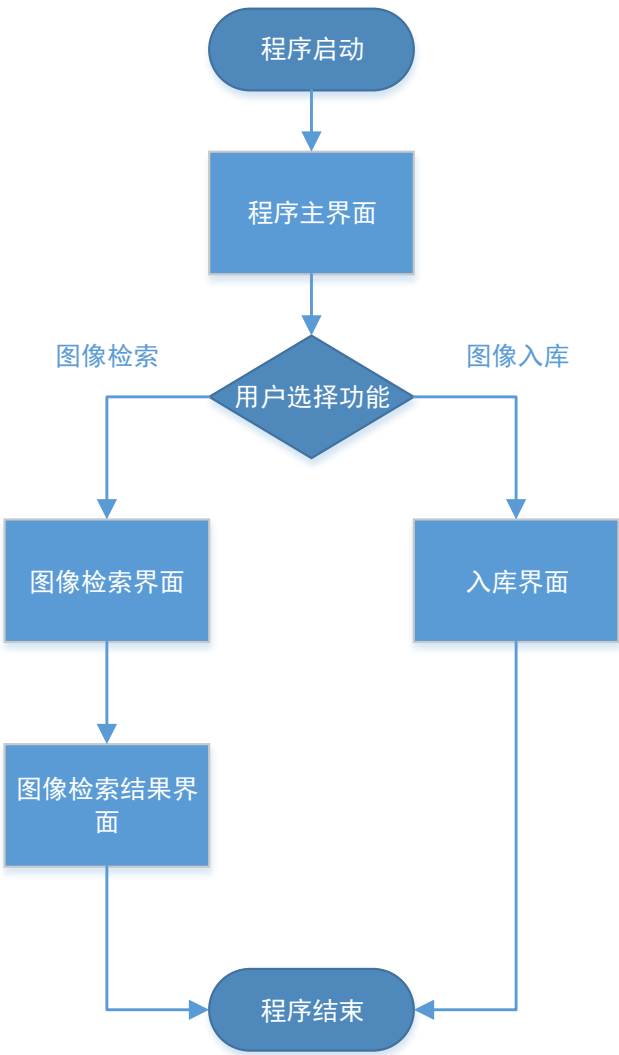


图 3.1 用户操作界面流程图

界面相关类信息如表 3.2 所示。

表 3.2 界面相关类信息

类	界面	功能描述
StartFrame	主界面	提供入库与检索不同功能的选择
StorageFrame	入库界面	选择入库图片路径，进行图片入库
MainFrame	检索界面	选择检索图片，进行图像检索
ImageResultFrame	检索结果界面	检索结果显示

3.5 系统类图

3.5.1 特征提取模块

主要由 CNN.java 和 GetCNN.java 实现，两个类的类图如图 3.2 所示。

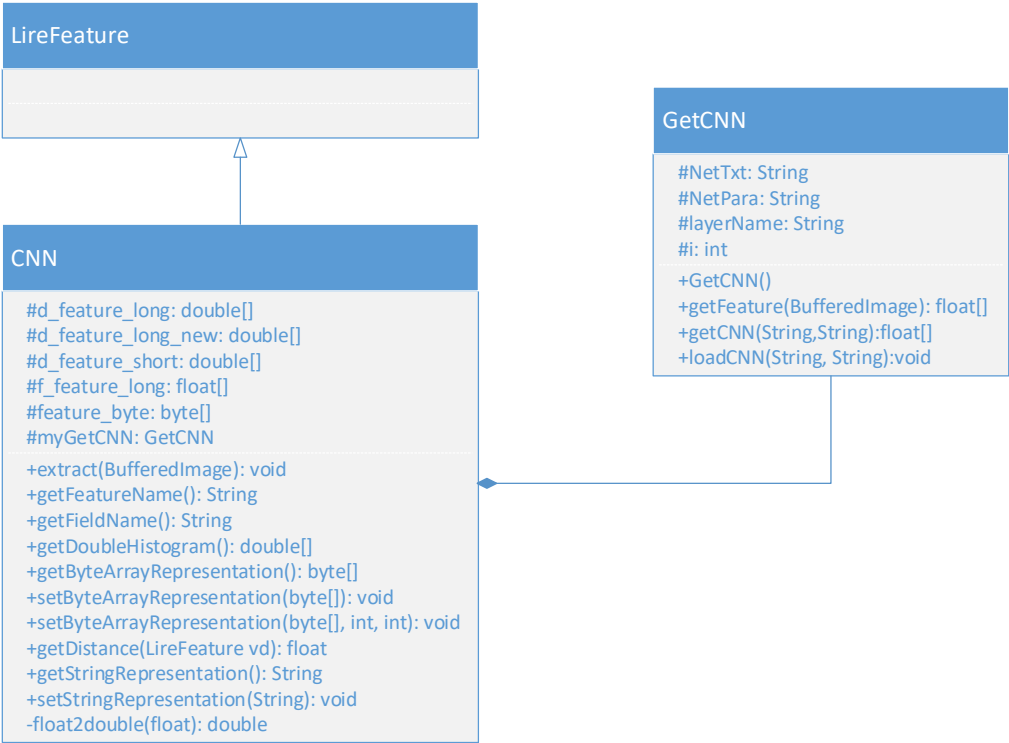


图 3.2 特征提取模块类图

GetCNN 类实现对 VGG-F 模型的调用，提取原始 VGG-F 特征，其成员变量意义与方法作用如表 3.1 所示。CNN 类中有一个 GetCNN 对象。

3.5.2 图像入库与图像检索实现

图像入库与图像检索相关类图如图 3.3 所示：

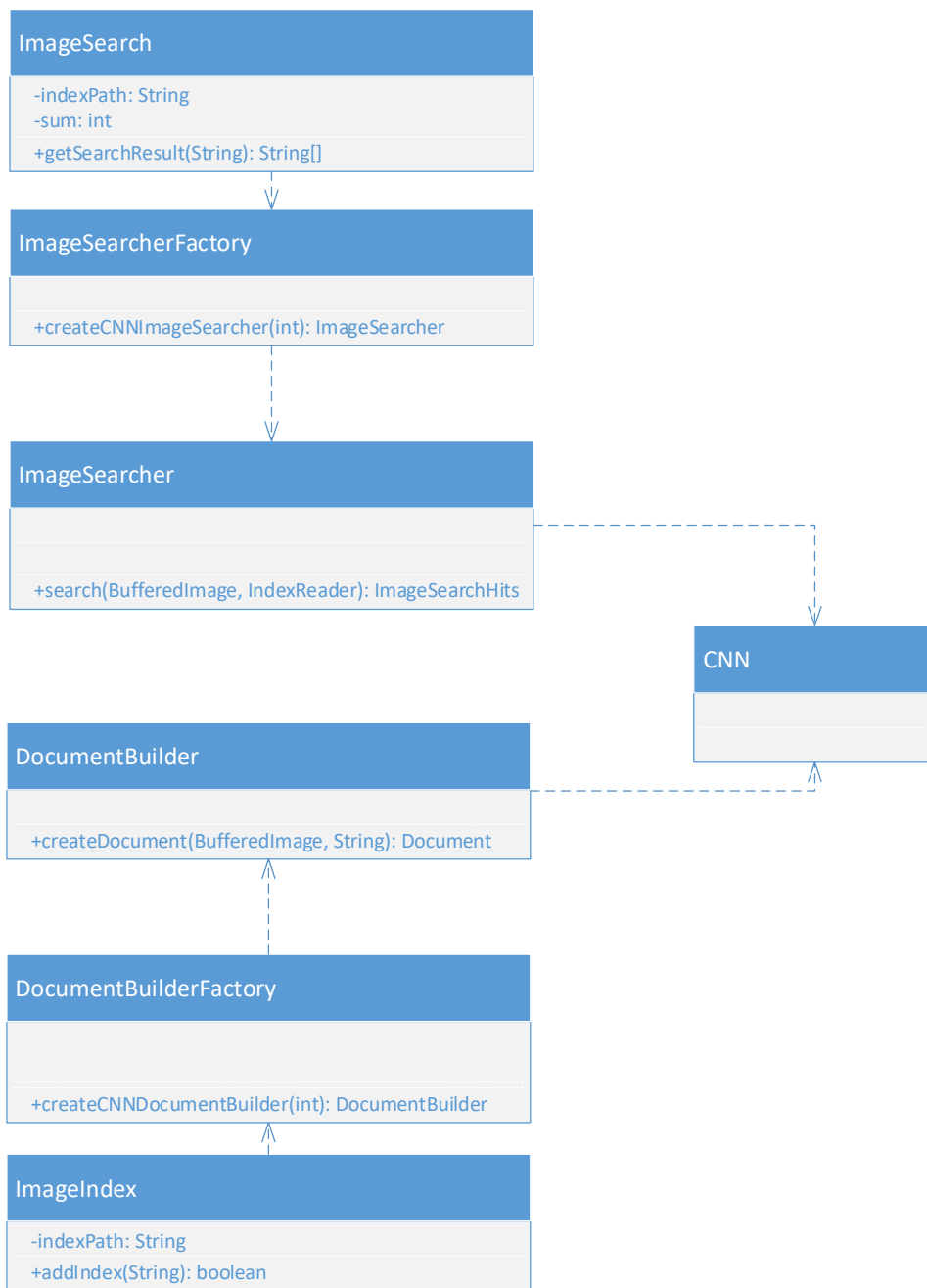


图 3.3 图像入库与图像检索相关类图

图像入库时 **ImageIndex** 创建 **DocumentBuilderFactory** 对象，调用相应接口得到 **DocumentBuilder** 对象，通过 **DocumentBuilder** 对象调用 **CNN** 类相关方法进行特征提取，完成特征存储和索引生成。

图像检索时 **ImageSearch** 创建 **ImageSearcherFactory** 对象，调用相应接口得到 **ImageSearcher** 对象，**ImageSearcher** 对象通过调用 **CNN** 类相关方法进行特征提取、特征距离计算，完成检索。

最终，完成了对 **Lire** 的扩展工作，将 **VGG-F** 特征加入到 **Lire** 中，并实现了

图像入库模块和图像检索模块。

注：因为 DocumentBuilder、DocumentBuilderFactory、ImageSearcher 和 ImageSearcherFactory 类均为 Lire 自带，所以该处类图只画出了我们在其中修改或增添的函数。CNN 类在图 3.2 中有详细描述，所以图 3.3 中不再细画。

3.5.3 界面实现

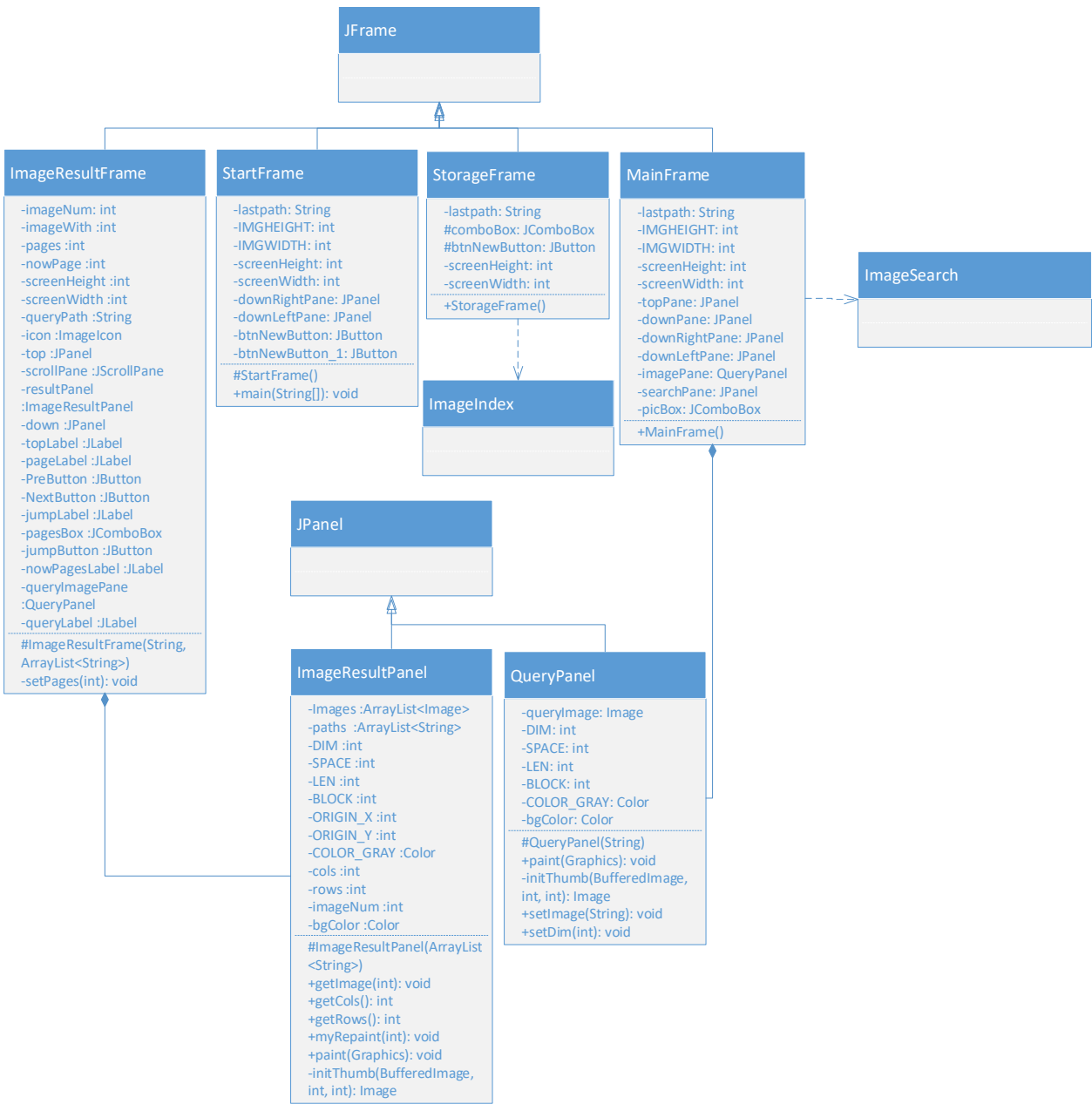


图 3.4 界面相关类图

系统主要有四个界面，分别为主界面、入库界面、检索界面和检索结果界面，

它们分别对应四个类 StartFrame、StorageFrame、MainFrame 和 ImageResultFrame，这四个类有共同的父类 JFrame(Java 自带)。在 MainFrame 类中有一个 QueryPanel 类的实例，用来显示待检索图像。在 ImageResultFrame 类中有一个 ImageResultPanel 类的实例，用来显示检索结果。QueryPanel 和 ImageResultPanel 都继承自 Java 自带的 JPanel 类。MainFrame 中有按钮在响应时会创建 ImageSearch 对象来实现搜索功能，StorageFrame 中有按钮在响应时会创建 ImageIndex 对象来实现创建索引功能。

3.6 系统运行过程

系统的功能主要为图像入库与图像检索，下面从这两个方面来描述人机交互过程。

3.6.1 图像入库过程

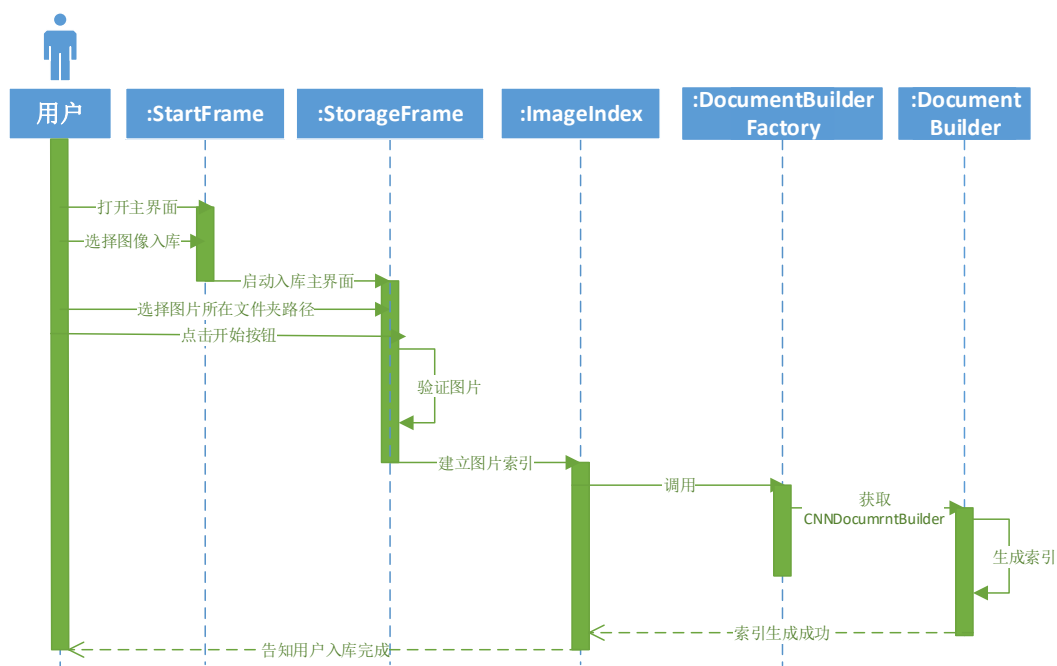


图 3.5 图像入库过程顺序图

图像入库过程的用户输入是一个文件夹路径，该文件夹中包含此次入库的所有图像，最终系统将图像入库之后通知用户入库完成。

过程如图 3.4，首先用户打开系统主界面，然后选择图像入库，主界面会再

启动一个入库主界面，然后用户在入库主界面选择图片所在文件夹的路径，并点击开始按钮，系统验证图片后会创建一个 ImageIndex，ImageIndex 会调用 DocumentBuilderFactory 的接口来获取 DocumentBuilder 对象，最终通过该对象来生成索引，索引构建完成之后，系统告知用户入库完成。

注：系统会遍历入库文件夹的子文件夹。

3.6.2 图像检索过程

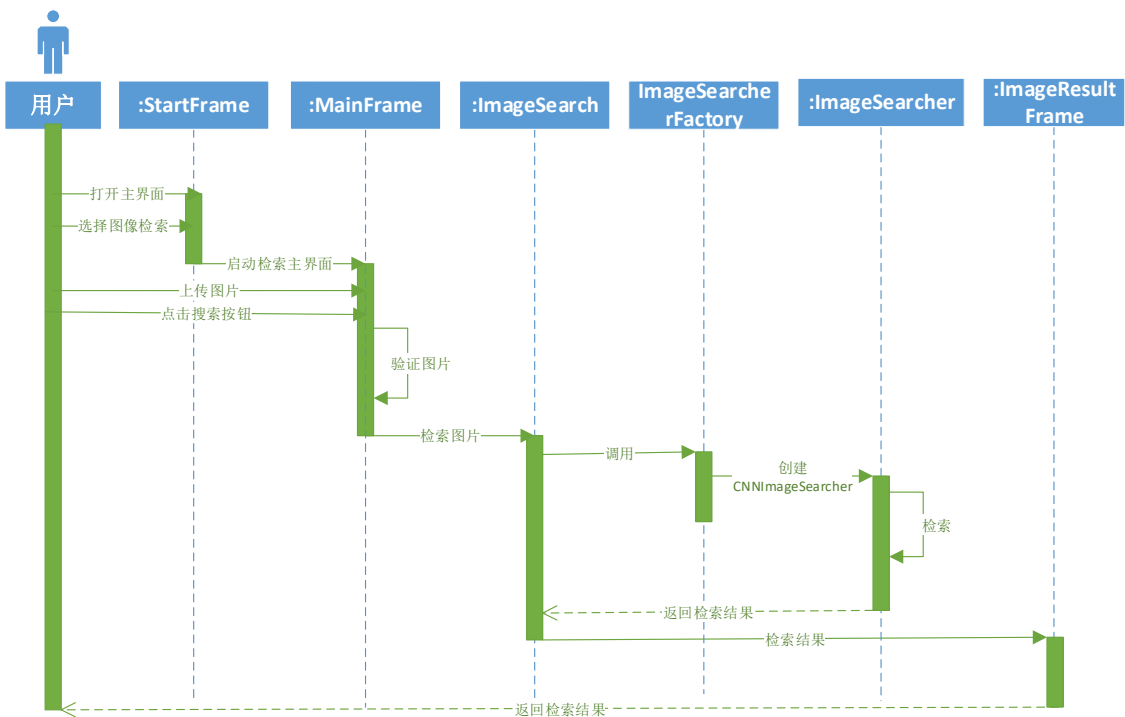


图 3.6 图像检索过程顺序图

图像检索过程的用户输入是一张图片，最终系统将检索库中相似图片，按照相似度返回检索结果。

过程如图 3.5，首先用户打开系统主界面，然后选择图像检索，主界面会再启动一个检索主界面，然后用户在检索主界面上上传图片，并点击搜索按钮，系统验证图片后会创建一个 ImageSearch，ImageSearch 调用 ImageSearcherFactory 的接口来获取 ImageSercher 对象，通过该对象来检索相似图片，最终将创建一个 ImageResultFrame 来展示检索结果。

注：该过程的输入要求为 jpg 格式图片，检索结果最多显示 6 页，每页 40 张图片。