基于 Lire 的分析与扩展
需求规格说明书

Version 4.1

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	20170323	宋昱材 刘少凡 吴沂楠 黄飞	刘少凡 宋昱材 吴沂楠 黄飞	初稿
1.1	20170327	刘少凡	黄飞	增添业务需求描述，更新功能需求
1.2	20170331	刘少凡	宋昱材	依据其他组同学评审修改部分内容
1.3	20170331	吴沂楠	刘少凡	根据老师批注修改了部分内容
2.0	20170405	宋昱材 刘少凡 吴沂楠 黄飞	宋昱材 刘少凡 吴沂楠 黄飞	依据上周课堂讨论内容，对需求说明书的每部分都进行了一定的修改与调整
2.1	20170413	刘少凡 黄飞 宋昱材	刘少凡 黄飞 宋昱材	依据F、G组评审修改部分内容
3.0	20170420	宋昱材	刘少凡 黄飞 吴沂楠	细化工作重点具体内容
3.1	20170422	黄飞 宋昱材	刘少凡 黄飞 吴沂楠 宋昱材	修改框架图 修改非功能需求RUCM图
3.2	20170426	刘少凡 黄飞	刘少凡 黄飞 吴沂楠 宋昱材	依据评审意见修改部分内容
4.0	20170510	刘少凡	黄飞 吴沂楠 宋昱材	依据老师评审修改部分内容
4.1	20170516	刘少凡	黄飞 吴沂楠 宋昱材	增加CBIR系统的功能需求用例图与RUCM模型

目录

1 前言

1.1 目的

以 Lire 开源框架及相关资料为输入，分析软件设计需求，结合软件工程综合实验具体要求，输出软件需求规格说明书，作为设计开发的依据并指导后续的开发工作。

1.2 软件需求分析目的

软件需求分析（Software Requirement Analysis）是对软件预期想要实现的业务目标进行分析，在明确定义和描述用户的业务目标及与之对应的具体业务需求的基础上，进一步确定并详细描述对应的软件功能性需求和非功能性需求等内容。

需求分析的具体内容可以归纳为五个方面：软件的总体概述，软件的业务需求，软件的功能需求，软件的非功能性需求，软件的运行要求。

1.3 文档概述

文档用途：本文档主要是介绍 Lire 系统需求及规格说明。

主要内容如下：

- 描述了 Lire 的业务需求；
- 以用例图的形式给出 Lire 系统功能需求，并对用例模型进行详细的描述；
- 使用 RUCM 模型对功能需求进行建模；
- 描述了 Lire 的非功能性需求；
- 描述了与 Lire 实施相关的硬件环境的一些要求；
- 描述了与 Lire 实施相关的软件环境的要求。

1.4 术语和缩略语

表 1.1 术语和缩略语

编号	术语	说明
1	UCM	用例建模
2	RUCM	限制性用例模型
3	Lucene	一个基于 Java 的全文信息检索工具包
4	CBIR	Content Based Image Retrieval，基于内容的图像检索

5	MPEG-7	MPEG-7 标准被称为“多媒体内容描述接口”，为各类多媒体信息提供一种标准化的描述
6	PHOG	Pyramid Histogram of Oriented Gradients，分层梯度方向直方图
7	CEDD	颜色和边缘的方向性描述符
8	FCTH	模糊颜色和纹理直方图
9	索引	一种单独的、物理的数对数据库表，是一种存储结构，其中的数据按照某一列或多列的值进行排序
10	哈希	输入任意长度的 data 数据，经过哈希算法处理后输出一个定长的数据 key。这个过程是不可逆的，无法由 key 逆推 data
11	聚类	将物理或抽象对象的集合分成由类似的对象组成的多个类的过程
12	Document	Lucene 中的一个文档表示为一个 Document
13	Field	Lucene 索引文档里的域，一个文档 Document 可以包含多个 Field 域
14	SIFT	Scale-invariant feature transform，尺度不变特征变换，是用于图像处理领域的一种描述
15	SURF	Speeded-Up Robust Features，加速稳健特征，是一个稳健的图像识别和描述算法
16	Kmeans	算法接受参数 k，然后将事先输入的 n 个数据对象划分为 k 个聚类
17	JDK	Java Development Kit，Java 开发工具
18	CNN	Convolutional Neural Network，卷积神经网络
19	Caffe	一个深度学习框架
20	Pyhton	一种面向对象的解释型计算机程序设计语言
21	Matlab	美国 MathWorks 公司出品的商业数学软件
22	Alexnet	一种卷积神经网络的网络结构
23	vgg	一种卷积神经网络的网络结构
24	resnet	一种卷积神经网络的网络结构
25	JNI	Java Native Interface，提供 API 实现 Java 和其他语言的通信
26	IndexWriter	Lucene 的一个类，用来构造索引文件并写到本地

2 总体概述

2.1 项目概述

Lire（Lucene Image Retrieval）是开源项目 Caliph and Emir（项目主页：http://www.semanticmetadata.com）的子项目。Lire 提供一种构造基于内容的图像检索系统（Content Based Image Retrieval System）的简单方式。

Lire 可以为 CBIR 系统创建图像特征的 Lucene 索引库。Lire 支持多种不同的底层图像特征，如 MPEG-7 标准的视觉描述符，以及 PHOG，CEDD，FCTH 等。

Lire 提供简单且易扩展的索引搜索的方法。

2.2 项目包结构简要分析

- 图像分析包：net.semanticmetadata.lire.imageanalysis

  imageanalysis 包实现各种图像特征的提取功能，包括全局特征和局部特征。LireFeature 类为各种图像特征类的父类，该类定义图像特征类必须实现的各类方法。

- 图像索引构造包：net.semanticmetadata.lire.builders

  builders 包提供图像入库的索引构造器，包括全局特征构造器和局部特征构造器。

- 图像检索包：net.semanticmetadata.lire.seachers

  searchers 包提供各种不同类型的图像检索器。

- 索引算法包：net.semanticmetadata.lire.indexers

  indexers 包提供索引生成相关的各类算法实现，包括各种哈希算法、并行化算法等。

- 聚类算法包：net.semanticmetadata.lire.classifiers

  classifiers 包实现多种聚类算法，包括 Kmeans 算法等。

- 检索结果筛选包：net.semanticmetadata.lire.filters

  filters 包实现了两种对检索结果进行筛选的方法。

- 工具包：net.semanticmetadata.lire.utils

utils 包提供系统各功能块所需的各类常用操作的实现，如命令行操作、数组数据类型转换、文件操作、图像操作、Lucene 操作、距离计算等。

2.3 用户定义

Lire 的用户定义为开发人员，主要为 CBIR 系统的开发人员。

3 业务需求

3.1 业务需求概述

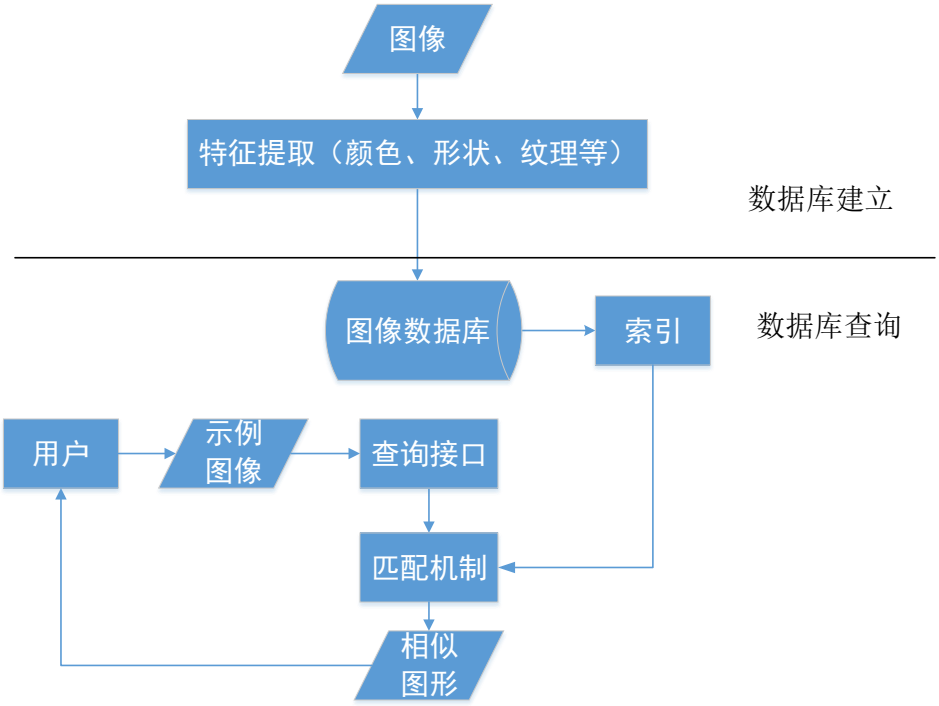

图 3.1 CBIR 系统的一般框架

一个基于内容图像检索系统（CBIR 系统）的一般框架如图 3.1 所示。CBIR 系统开发人员的开发目标是实现 CBIR 系统框架中的各个模块，并实现模块的整合，组成一个 CBIR 系统。Lire 作为一个工具包为构建 CBIR 系统提供功能支持。

CBIR 系统的模块结构如图 3.2 所示，CBIR 系统可以大体分为查询与显示模块、图像库管理模块、图像特征提取模块和相似性匹配模块。CBIR 系统应提供图像入库和图像检索两个基本功能。图像入库阶段，CBIR 系统用户输入待入库图像，特征提取模块提取图像特征，图像库管理模块将图像特征与图像标识符（如图像存储路径）一起存入数据库并生成索引。图像检索阶段，CBIR 系统用户输入检索实例图像，特征提取模块提取示例图像特征，相似度匹配模块将实例图像特征与数据库中保存的特征进行相似度比较，得到检索结果。

CBIR 系统实现工具包的功能是辅助开发人员完成系统开发，因此工具包应当能够提供各个模块的多种具体实现形式，供开发人员方便地选择、调用和组合，以产生适应不同需求的 CBIR 系统。同时，工具包应当具有较强的可扩展性，便

于 CBIR 系统开发人员根据具体的实现目标修改或扩展工具包的代码。

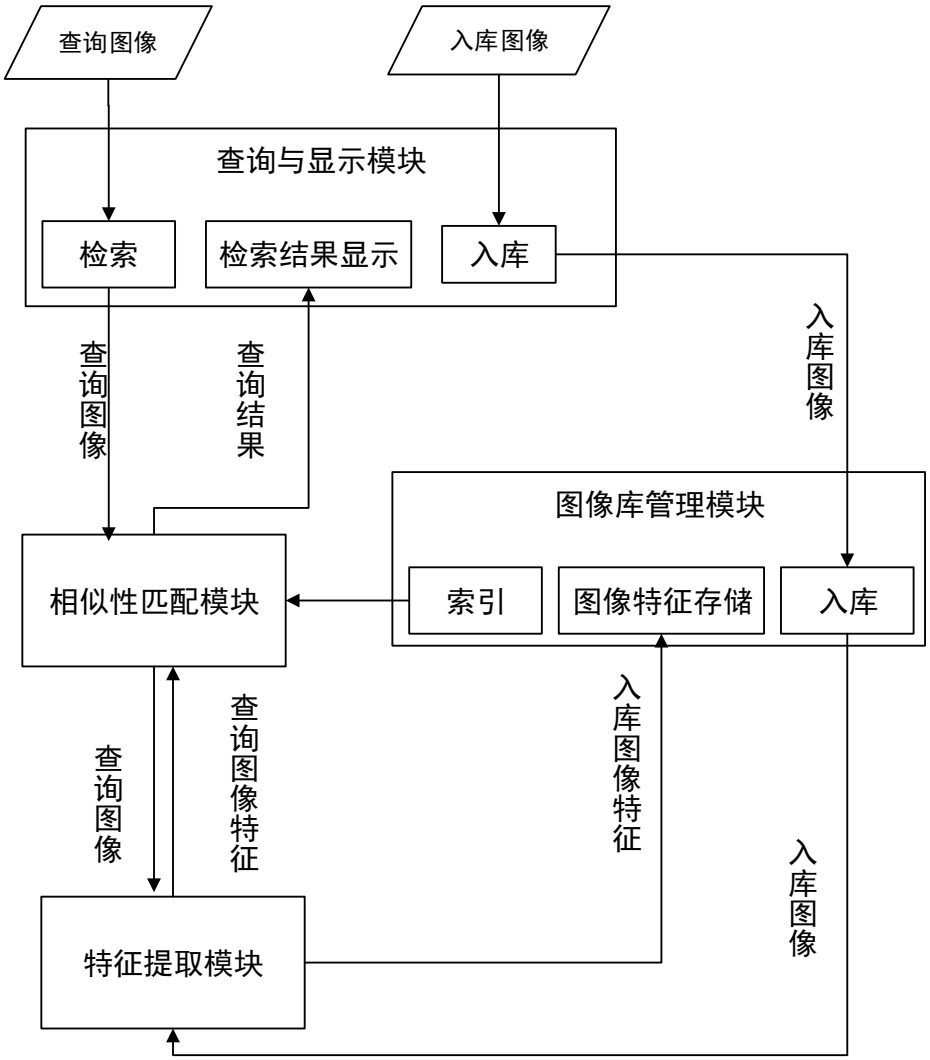

图 3.2 CBIR 系统模块结构图

3.2 图像特征提取

Lire 工具包应当提供尽可能多的当前主流的图像特征提取算法，并提供简洁的调用接口，供 CBIR 系统开发者进行选择。同时，工具包应当提供可扩展的机制，方便 CBIR 系统开发人员引入新的图像特征提取算法。

3.3 图像入库

Lire 工具包应当提供一种高效的数据库存储方式，能够快速高效地进行存储、读取和检索。工具包应当提供适应多种图像特征的图像入库算法，并且提供尽可能多的当前主流的索引生成算法。工具包应提供简洁的上层调用接口，方便开发

人员快速实现不同的图像入库功能模块。另外，工具包应当提供可扩展的机制，方便 CBIR 系统开发人员扩展新的算法。

3.4 图像检索

Lire 工具包应当提供尽可能多的主流图像特征距离计算方法，提供高效、准确的多种排序算法，并将图像检索功能封装为简洁的调用接口，供开发人员快速实现不同的图像检索功能模块。

4 功能需求

因为 Lire 主要面向的是 CBIR 系统的开发，下面将从 Lire 的功能需求和典型 CBIR 系统的功能需求两方面来描述。

4.1 Lire 用例模型

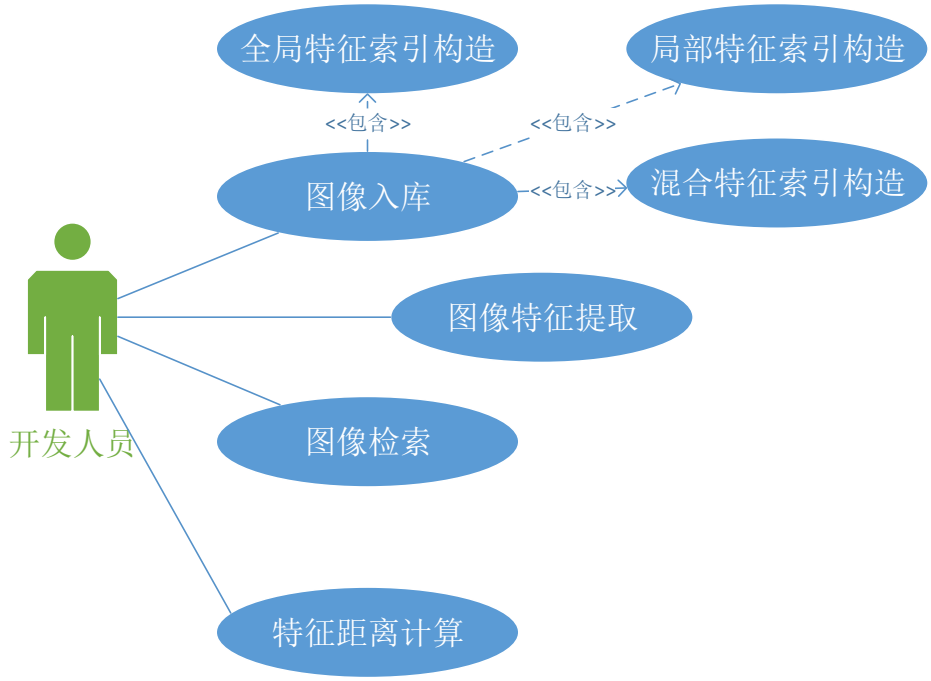

图 4. 1 Lire 用例图

4.2 Lire 用例说明

下面使用 RUCM 模型描述用例，通过 RUCM 模型能够对用例进行规范的描述。RUCM 即限制性用例建模，它的目标是：

1. 使 UCMs 更加可理解并且更精确。
2. 从 UCMs 自动生成分析模型。

RUCM 由以下两部分组成：

1. 一个用于系统组织 UCSs 的用例模板。
2. 限制用户写 UCSs 的一系列规则。

4.2.1 图像入库

Use Case Specification	
Use Case Name	图像入库
Brief Description	开发人员编写图像入库程序的一般流程
Precondition	Lire包已配置好
Primary Actor	开发人员
Secondary Actors	None
Dependency	INCLUDE USE CASE 全局特征索引构造, INCLUDE USE CASE 局部特征索引构造, INCLUDE USE CASE 混合特征索引构造
Generalization	None

Basic Flow "正常"▼	Steps	
	1	开发人员导入文件读取模块
	2	开发人员调用文件读取接口读取图像文件夹路径内的所有图像
	3	IF 开发人员希望提取的特征均为全局特征
	4	THEN INCLUDE USE CASE 全局特征索引构造
	5	IF 开发人员希望提取的特征均为局部特征
	6	THEN INCLUDE USE CASE 局部特征索引构造
	7	IF 开发人员希望提取的特征既有全局特征也有局部特征
	8	THEN INCLUDE USE CASE 混合特征索引构造
	9	开发人员导入Lucene的IndexWriter模块
	10	开发人员创建一个IndexWriter对象将上面生成的索引写到本地
	11	开发人员关闭IndexWriter
	Postcondition	程序编写完成

图 4. 2 图像入库 RUCM 模型

开发人员可以利用 Lire 编写图像入库的程序，编写该程序的一般流程如下：首先开发人员需要配置好 Lire 包；之后开发人员需要导入文件读取的模块用来读取准备入库的图像；之后开发人员根据自己希望提取的特征的种类来进行全局特征索引的构造或者局部特征索引的构造或者混合特征的索引构造；之后开发人员导入 Lucene 的 IndexWriter 模块，创建一个 IndexWriter 对象将上面生成的索引写到本地；最后开发人员关闭 IndexWriter 模块即可。

4.2.2 全局特征索引构造

Use Case Specification	
Use Case Name	全局特征索引构造
Brief Description	开发人员编写全局特征索引构造程序的一般流程
Precondition	开发人员确定了要提取的特征，且均属于全局特征
Primary Actor	开发人员
Secondary Actors	None
Dependency	None
Generalization	None

Basic Flow	Steps	
"正常"▼	1	开发人员导入全局特征索引构造模块
	2	开发人员导入图像特征提取模块
	3	开发人员指定提取的特征
	4	开发人员编码遍历图像，对每张图像调用全局索引构造接口
	Postcondition	程序编写完成

图 4. 3 全局特征索引构造 RUCM 模型

开发人员可以利用 Lire 编写全局特征索引构造的程序，编写该程序的一般流程如下：开发人员首先需要导入全局特征索引构造的模块和图像特征提取的模块；之后开发人员指定想提取的特征；最后开发人员编码遍历图像，对每张图像调用全局索引构造的接口即可。

4.2.3 局部特征索引构造

Use Case Specification	
Use Case Name	局部特征索引构造
Brief Description	开发人员编写局部特征索引构造程序的一般流程
Precondition	开发人员确定了要提取的特征，且均属于局部特征
Primary Actor	开发人员
Secondary Actors	None
Dependency	None
Generalization	None

Basic Flow	Steps	
"正常"▼	1	开发人员导入局部特征索引构造模块
	2	开发人员导入图像特征提取模块
	3	开发人员指定提取的特征
	4	开发人员编码遍历图像，对每张图像调用局部索引构造接口
	Postcondition	程序编写完成

图 4. 4 局部特征索引构造 RUCM 模型

开发人员可以利用 Lire 编写局部特征索引构造的程序，编写该程序的一般流程如下：开发人员首先需要导入局部特征索引构造的模块和图像特征提取的模块；之后开发人员指定想提取的特征；最后开发人员编码遍历图像，对每张图像调用局部索引构造的接口即可。

4.2.4 图像检索

Use Case Specification	
Use Case Name	图像检索
Brief Description	开发人员编写图像检索程序的一般流程
Precondition	Lire包已配置好
Primary Actor	开发人员
Secondary Actors	None
Dependency	None
Generalization	None

Basic Flow	Steps	
"正常"▼	1	开发人员调用Java接口读取待检索图像
	2	开发人员导入Lucene的IndexReader模块
	3	开发人员调用Lucene的IndexReader模块的接口，读取数据库图像索引
	4	开发人员导入图像检索模块
	5	开发人员调用图像检索模块的接口，检索相似图片
	Postcondition	程序编写完成

图 4. 5 图像检索 RUCM 模型

开发人员可以利用 Lire 编写图像检索的程序，编写该程序的一般流程如下：首先开发人员需要配置好 Lire 包；之后开发人员调用 Java 接口读取待检索图像；之后开发人员导入 Lucene 的 IndexReader 模块并调用它的接口读取数据库图像索引；最后开发人员导入图像检索模块并调用它的接口检索相似图片。

4.2.5 图像特征提取

Use Case Specification	
Use Case Name	图像特征提取
Brief Description	开发人员编写图像特征提取程序的一般流程
Precondition	Lire包已配置好
Primary Actor	开发人员
Secondary Actors	None
Dependency	None
Generalization	None

Basic Flow	Steps	
"正常"▼	1	开发人员调用Java方法读取图像
	2	开发人员导入图像特征提取模块
	3	开发人员创建想要的特征类的实例
	4	开发人员调用计算特征的接口
	Postcondition	程序编写完成

图 4. 6 图像特征提取 RUCM 模型

开发人员可以利用 Lire 编写图像特征提取的程序，编写该程序的一般流程如下：首先开发人员需要配置好 Lire 包；之后开发人员调用 Java 方法读取图像，然后导入图像特征提取模块；最后开发人员创建想要的特征类实例并调用计算特征的接口即可。

4.2.6 特征距离计算

Use Case Specification	
Use Case Name	特征距离计算
Brief Description	开发人员编写特征距离计算程序的一般流程
Precondition	Lire包已配置好
Primary Actor	开发人员
Secondary Actors	None
Dependency	None
Generalization	None

Basic Flow	Steps	
"正常"▼	1	开发人员调用Java方法读取图像
	2	开发人员导入图像特征提取模块
	3	开发人员创建想要的特征类的实例
	4	开发人员调用特征的距离计算接口
	Postcondition	程序编写完成

图 4. 7 特征距离计算 RUCM 模型

开发人员可以利用 Lire 编写特征距离计算的程序，编写该程序的一般流程如下：首先开发人员需要配置好 Lire 包；之后开发人员调用 Java 方法读取图像，然后导入图像特征提取模块；最后开发人员创建想要的特征类实例并调用特征的距离计算接口即可。

4.3 CBIR 系统用例模型

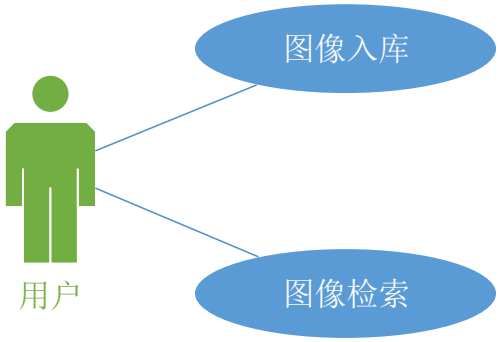

图 4. 8 CBIR 系统用例图

4.4 CBIR 系统用例说明

4.4.1 CBIR 系统图像入库

Use Case Specification	
Use Case Name	图像入库
Brief Description	将一个文件夹中的图像入库
Precondition	系统初始化完成
Primary Actor	用户
Secondary Actors	None
Dependency	None
Generalization	None

Basic Flow "正常"	Steps
1	用户点击图像入库按钮
2	用户选择图像文件夹
3	用户点击开始按钮
4	系统 VALIDATES THAT 图像可读
5	系统申请索引文件夹的锁
6	系统提取图像特征
7	系统生成索引
8	系统写索引到本地
9	系统释放锁
Postcondition	图像入库成功

Specific Alternative Flow "异常1"	RFS 4
1	系统报错
2	ABORT
Postcondition	图像入库失败

Specific Alternative Flow "异常2"	RFS 5
1	系统申请锁失败
2	系统报错
3	ABORT
Postcondition	图像入库失败

图 4. 9 CBIR 系统图像入库 RUCM 模型

用户利用 CBIR 系统进行图像入库时，流程一般如下：首先用户点击图像入库按钮，接下来选择图像文件夹并点击开始按钮，系统会验证该文件夹内图像是否可读，如果不可读，系统会报错并中止该次图像入库，如果可读，那么系统会申请索引文件夹的锁，如果申请失败，那么系统会报错并中止该次图像入库，如果成功，那么接下来系统会提取图像特征，然后生成索引，最后将索引写到本地之后再释放锁，图像入库成功。

4.4.2 CBIR 系统图像检索

Use Case Specification	
Use Case Name	图像检索
Brief Description	在系统中检索某图片的相似图片
Precondition	系统初始化完成
Primary Actor	用户
Secondary Actors	None
Dependency	None
Generalization	None

Basic Flow "正常"▼	Steps	
	1	用户点击图像检索按钮
	2	用户选择想要检索的图片
	3	用户点击搜索按钮
	4	系统 VALIDATES THAT 图像可读
	5	系统提取被检索图像的特征
	6	系统将被检索图像的特征与库中特征进行相似度计算并排序
	7	系统将检索结果按相似度从高到低的顺序展示给用户
	Postcondition	图像检索完成

Specific Alternative Flow "异常1"▼	RFS 4	
	1	系统报错
	2	ABORT
	Postcondition	图像检索失败

图 4. 10 CBIR 系统图像检索 RUCM 模型

用户利用 CBIR 系统进行图像检索时，流程一般如下：首先用户点击图像检索按钮，接下来选择想要检索的图片并点击搜索按钮，系统会验证该图像是否可读，如果不可读，系统会报错并中止该次图像检索，如果可读，那么系统会提取被检索图像的特征，然后将该特征与库中特征进行相似度计算并排序，最后系统将检索结果按照相似度从高到低的顺序展示给用户，图像检索完成。

5 非功能性需求分析

5.1 兼容性

作为一个跨平台的 CBIR 系统开发工具包，Lire 应该具有比较强大的兼容性。

5.1.1 操作系统兼容性

理想的软件应具有跨平台性，因此基于 Lire 开发的应用程序应能运行于不同的操作系统（如 Windows，Mac OS X 和 Linux）上。应该选择具有跨平台特性的编程语言。

主要的应用场景是开发者可以在不同的操作系统上方便地使用 Lire 开发或是部署用 Lire 开发的系统。RUCM 图如下：

Use Case Specification	
Use Case Name	系统迁移
Brief Description	LIre和用Lire开发的程序具有跨平台性
Precondition	使用Lire在已安装JDK6.0的Windows操作系统上开发完成了一个CBIR系统，并在目标机上已安装好Linux系统（Ubuntu 16）和JDK6.0
Primary Actor	开发人员
Secondary Actors	None
Dependency	None
Generalization	None

Basic Flow	Steps	
"正常"▼	1	开发人员将完成的所有java文件拷贝到目标机上
	2	开发人员在目标机上使用JDK编译java文件
	3	开发人员在目标机上启动程序
	Postcondition	完成CBIR系统的迁移

图 5.1 系统迁移 RUCM 图

5.1.2 数据兼容性

Lire 提取图像各类特征之后生成的索引数据应该以跟操作系统和运行环境无关的方式储存，使不同平台上能够共享生成的图像数据库。

主要的应用场景是当开发者在一个操作系统平台上使用 Lire 进行了一个图像数据库的特征提取和索引生成后，又恰好需要在另一个平台上使用时，可以直接将生成的索引文件复制到相应的路径下，不需要进行额外的数据格式转换。RUCM 图如下：

Use Case Specification	
Use Case Name	数据迁移
Brief Description	产生的索引文件具有跨平台性，将生成的索引文件从Windows系统迁移到Ubuntu系统中
Precondition	使用Lire在Windows系统下进行了图像数据库的特征提取和索引生成
Primary Actor	开发人员
Secondary Actors	None
Dependency	None
Generalization	None

Basic Flow	Steps	
"正常"▼	1	开发人员在目标机（Ubuntu 16）上配置Lire
	2	开发人员将在Windows系统下生成的所有索引文件复制到目标机相应的路径下
	3	开发人员在目标机上正常使用图像检索功能
	Postcondition	完成了索引文件的迁移

图 5.2 数据迁移 RUCM 图

5.2 可修改性

作为一种完全开源的框架，Lire 的代码应当对开发者完全透明。为了进行高效的开发。要求所有的程序代码具备一种简明、方便和清晰的构架设计与函数接口来方便用户的使用。此外，作为一个框架，Lire 应该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 必须具备良好的可修改性或可扩展性。

主要的应用场景是当开发者使用 Lire 时，可以很方便地修改 Lire 任何一个模块的实现，同时也可以在其中添加一个新模块或者在某个模块中添加新的算法以满足自己的需求。RUCM 图如下：

Use Case Specification	
Use Case Name	扩展算法
Brief Description	Lire的代码对开发人员完全可见，针对图像特征提取模块进行扩展
Precondition	已初步使用Lire构造出一个CBIR系统，且Lire图像特征提取模块没有提供想使用的算法
Primary Actor	开发人员
Secondary Actors	None
Dependency	None
Generalization	None

Basic Flow	Steps	
"正常"▼	1	开发人员参照Lire的代码组织方式编写相应的Java文件
	2	开发人员在代码相应位置调用新添加的算法
	3	开发人员重新编译所有文件
	Postcondition	在Lire图像特征提取模块中添加算法完成

图 5.3 扩展算法 RUCM 图

5.3 高效性

Lire 作为 CBIR 系统的开发框架，需要支持对大规模数据的索引及搜索，因此需要具有高效性。这主要体现在两个方面：

- 高效实现大规模数据索引入库

开发一个图像检索系统，如果提取图像特征或将索引入库占据了过多时间，那么必然影响了开发效率。因此 Lire 需要高效完成大规模数据的索引入库过程。

- 迅速响应查询条件并返回结果

衡量检索性能的一个重要指标就是响应时间，因此 Lire 需要能够在较短的时间内响应请求并返回特征相似的所有结果。

主要的应用场景是当开发人员需要使用大量的图像数据构建图像数据库时，索引入库的性能对开发的效率有着重要的影响；同样当已建立起一个大规模的图像数据库后，高效的检索对开发完成的 CBIR 系统的使用体验也有着直接的影响。总之，高效性是对于一个 CBIR 系统开发与使用层面上的直接保障。RUCM 图如下：

Use Case Specification	
Use Case Name	高效入库
Brief Description	高效实现大规模数据入库
Precondition	开发人员需要使用大规模图像数据（1万张图片）构建图像数据库
Primary Actor	开发人员
Secondary Actors	None
Dependency	None
Generalization	None

Basic Flow	Steps	
"正常"▼	1	开发人员输入大规模图像数据
	2	开发人员提取所有图像的特征
	3	开发人员将索引入库
	Postcondition	图像数据库构建完毕

图 5.4 高效入库 RUCM 图

Use Case Specification	
Use Case Name	高效检索
Brief Description	迅速响应查询条件并返回结果
Precondition	已使用大规模图像数据构建图像数据库
Primary Actor	开发人员
Secondary Actors	None
Dependency	None
Generalization	None

Basic Flow	Steps	
"正常"▼	1	开发人员给定检索图片
	2	Lire响应请求并返回特征相似的所有结果，耗时1s以内
	Postcondition	图像检索完成

图 5.5 高效检索 RUCM 图

6 运行要求

6.1 硬件要求

- CPU:　Intel i5 及以上
- 内存：4G 内存及以上
- 硬盘：20G 硬盘及以上

6.2 软件要求

- 操作系统：Windows7 版本及以上、Mac OS X 10.7.3 和更高版本、Oracle Linux 5.5 以

上、Red Hat Enterprise Linux 5.5 和更高版本、Suse Linux Enterprise Server 10 SP2，11.x 及以上、Ubuntu Linux 10.04 和更高版本

- 编译环境：JDK6.0 版本及以上

7 工作重点

7.1 工作内容

项目后续计划完成的改进工作与 Lire 的可修改性有关。

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。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。

7.2 技术路线

为了实现 7.1 节中工作内容，本节拟定了如下的技术路线，首先是拟定了新的特征提取方法如何加入到 Lire 工具包中，其次拟定了该特征如何获得，最后拟定了不同语言之间的协调工作的方式。

7.2.1 增加新的特征提取方法

特征提取方法的具体实现在 imageanalysis 包中。LireFeature 是一个 Java 接口，它定义了特征提取类必须实现的方法，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。图像入库和图像检索的相关类在操作图像特征时，均基于 LireFeature 接口进行操作，不与具体的特征相关，因此不需做过多改动。

项目需要在 imageanalysis 包中增加新的特征实现类 CNN 类，继承 LireFeature 接口并实现具体方法。调用 CNN 模型需要深度学习框架的支持，同时需要确定

CNN 模型的具体类型。

7.2.2 深度学习框架和模型的选择

基于项目组成员已有的开发经验，项目计划使用 Caffe 深度学习框架。Caffe 是由 Berkeley Vision and Learning Center 开发的深度学习框架，该框架基于 C++编程语言实现，同时提供 python 和 matlab 的调用接口。Caffe 提供了多种已训练的 CNNs 模型，同时也存在大量开发者不断贡献新的已训练模型。目前在卷积神经网络和图像识别、图像分类领域有很多研究都基于 Caffe 框架展开，Caffe 框架也被应用于实际产品开发的很多场景。Caffe 可以通过 C++、python 和 matlab 进行调用，并不支持 java 接口，因此需要解决 java 调用 C++的问题。

目前 CNN 模型的类型有很多，如 alexnet、vgg、resnet 等，因此需要对模型进行对比和选择，选择适合于图像检索中特征提取的模型。

7.2.3 Java 对 C++的调用

项目计划利用 Java 的 JNI 机制实现对 Caffe 代码的调用。JNI 是 Java Native Interface 的缩写，它提供了若干的 API 实现了 Java 和其他语言的通信（主要是 C&C++）。从 Java1.1 开始，JNI 标准成为 java 平台的一部分，它允许 Java 代码和其他语言写的代码进行交互。

参考资料

[1] http://www.semanticmetadata.net/lire/

[2]http://blog.csdn.net/camu7s/article/details/49611823