**基于Lire的分析与扩展**

**需求规格说明书**

Version 4.2

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 |
| 4.2 | 20170530 | 刘少凡 |  | 增加术语词典 |

目录

[1前言 5](#_Toc483643040)

[1.1目的 5](#_Toc483643041)

[1.2软件需求分析目的 5](#_Toc483643042)

[1.3文档概述 5](#_Toc483643043)

[1.4术语和缩略语 5](#_Toc483643044)

[1.5术语词典 6](#_Toc483643045)

[2总体概述 8](#_Toc483643046)

[2.1项目概述 8](#_Toc483643047)

[2.2项目包结构简要分析 8](#_Toc483643048)

[2.3用户定义 9](#_Toc483643049)

[3业务需求 10](#_Toc483643050)

[3.1 业务需求概述 10](#_Toc483643051)

[3.2 图像特征提取 11](#_Toc483643052)

[3.3 图像入库 11](#_Toc483643053)

[3.4 图像检索 12](#_Toc483643054)

[4功能需求 13](#_Toc483643055)

[4.1 Lire用例模型 13](#_Toc483643056)

[4.2 Lire用例说明 13](#_Toc483643057)

[4.2.1图像入库 14](#_Toc483643058)

[4.2.2 全局特征索引构造 15](#_Toc483643059)

[4.2.3 局部特征索引构造 15](#_Toc483643060)

[4.2.4 混合特征索引构造 16](#_Toc483643061)

[4.2.5 图像检索 17](#_Toc483643062)

[4.2.6 图像特征提取 18](#_Toc483643063)

[4.2.7 特征距离计算 18](#_Toc483643064)

[4.3 CBIR系统用例模型 19](#_Toc483643065)

[4.4 CBIR系统用例说明 20](#_Toc483643066)

[4.4.1 CBIR系统图像入库 20](#_Toc483643067)

[4.4.2 CBIR系统图像检索 21](#_Toc483643068)

[5 非功能性需求分析 22](#_Toc483643069)

[5.1 兼容性 22](#_Toc483643070)

[5.1.1操作系统兼容性 22](#_Toc483643071)

[5.1.2数据兼容性 22](#_Toc483643072)

[5.2 可修改性 23](#_Toc483643073)

[5.3 高效性 24](#_Toc483643074)

[6运行要求 25](#_Toc483643075)

[6.1硬件要求 25](#_Toc483643076)

[6.2软件要求 25](#_Toc483643077)

[7工作重点 26](#_Toc483643078)

[7.1 工作内容 26](#_Toc483643079)

[7.2 技术路线 26](#_Toc483643080)

[7.2.1 增加新的特征提取方法 26](#_Toc483643081)

[7.2.2 深度学习框架和模型的选择 27](#_Toc483643082)

[7.2.3 Java对C++的调用 27](#_Toc483643083)

[参考资料 27](#_Toc483643084)

# 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 |

## 1.5术语词典

**表1.2 术语词典**

|  |  |  |
| --- | --- | --- |
| 编号 | 术语 | 说明 |
| 1 | 图像入库 | 提取图像特征并将其以某种方式存储在本地的过程 |
| 2 | 全局特征 | 图像特征的一种分类，包括CEDD、FCTH、PHOG等 |
| 3 | 全局特征索引构造 | 提取图像的某一种或某几种全局特征并构造索引的过程 |
| 4 | 局部特征 | 图像特征的一种分类，包括SIFT、SURF、ShapeContext等 |
| 5 | 局部特征索引构造 | 提取图像的某一种或某几种局部特征并构造索引的过程 |
| 6 | 混合特征索引构造 | 提取图像的某几种全局特征和局部特征并构造索引的过程 |
| 7 | 图像检索 | 根据相同种类的某种图像特征，将待检索图像和图像库图像进行相似度计算，以得到在该种特征下与待检索图像“相似”的图像的过程 |
| 8 | 特征距离计算 | 计算两张图片同种特征的欧氏距离或其他距离的过程 |

# 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



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



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

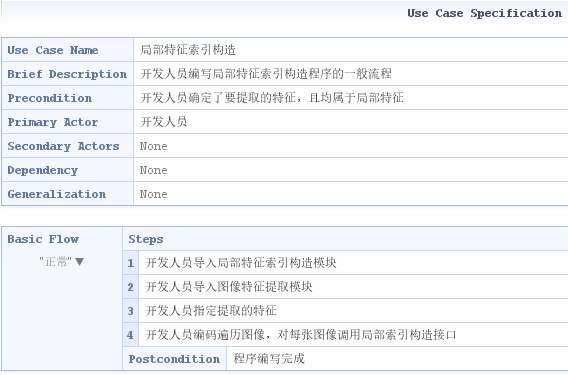


图4. 4局部特征索引构造RUCM模型

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 混合特征索引构造



图4. 5混合特征索引构造RUCM模型

开发人员可以利用Lire编写混合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混合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，对每张图像调用混合索引构造的接口即可。

混合特征索引构造与局部/全局特征索引构造的区别是混合特征索引构造允许在同一次入库过程中即提取想要的全局特征，也提取想要的局部特征。

### 4.2.5 图像检索



图4. 6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。

### 4.2.6 图像特征提取



图4. 7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。

### 4.2.7 特征距离计算



图4. 8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. 9 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



图4. 10 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



图4. 11 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：

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：



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：



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：



图 5.4 高效入库RUCM图



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>