**基于Lire的分析与改进**

**需求规格说明书**

Version 1.1

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 |  | 增添业务需求描述，更新功能需求 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1前言 4](#_Toc478370020)

[1.1目的 4](#_Toc478370021)

[1.2软件需求分析理论 4](#_Toc478370022)

[1.3文档概述 4](#_Toc478370023)

[1.4术语和缩略语 5](#_Toc478370024)

[2总体概述 6](#_Toc478370025)

[2.1项目概述 6](#_Toc478370026)

[2.2项目包结构简要分析 6](#_Toc478370027)

[2.3 需求识别 7](#_Toc478370028)

[2.3.1 图像特征提取 7](#_Toc478370029)

[2.3.2 图像入库 7](#_Toc478370030)

[2.3.3 索引生成 7](#_Toc478370031)

[2.3.4 图像检索 8](#_Toc478370032)

[3功能需求 9](#_Toc478370033)

[3.1 用例模型 9](#_Toc478370034)

[3.2 用例说明 9](#_Toc478370035)

[3.2.1图像入库 10](#_Toc478370036)

[3.2.2 索引生成 11](#_Toc478370037)

[3.2.3 全局特征索引构造 11](#_Toc478370038)

[3.2.4 局部特征索引构造 12](#_Toc478370039)

[3.2.5 图像检索 12](#_Toc478370040)

[3.2.6 图像特征提取 13](#_Toc478370041)

[3.2.7提取全局特征 13](#_Toc478370042)

[3.2.8 提取局部特征 14](#_Toc478370043)

[3.2.9 特征距离计算 14](#_Toc478370044)

[4 非功能性需求分析 15](#_Toc478370045)

[4.1 兼容性 15](#_Toc478370046)

[4.2 可修改性 15](#_Toc478370047)

[4.3 高效性 15](#_Toc478370048)

[4.4 场景需求 16](#_Toc478370049)

[4.4.1 基于Lire的检索引擎的应用场景 16](#_Toc478370050)

[4.4.2 场景的存在的问题 16](#_Toc478370051)

# 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 Reguirement Analysis）是对用户想要实现的功能进行分析，理解用户想要得到的该软件的完整功能，为将来进行软件设计提供一个基本依据。

需求分析的具体内容可以归纳为六个方面：软件的功能需求，软件与硬件或其他外部系统接口，软件的非功能性需求，软件的反向需求，软件设计和实现上的限制，阅读支持信息。

软件需求分析的主要实现目标：

(1) 对实现软件的功能做全面的描述，帮助用户判断实现功能的正确性、一致性和完整性，促使用户在软件设计启动之前周密地、全面地思考软件需求；

(2) 了解和描述软件实现所需的全部信息，为软件设计、确认和验证提供一个基准；

(3) 为软件管理人员进行软件成本计价和编制软件开发计划书提供依据；

## 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：

* 以用例图的形式给出Lire系统功能需求的分解结构，并对用例模型进行详细的描述，主要包括软件系统的用例模型、系统的核心流程等；

⮚使用 RUCM 模型对功能需求进行建模；

⮚描述了与此次系统实施相关的硬件环境的一些要求；

⮚描述了与此系统实施相关的软件环境的要求；

## 1.4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中一列或多列的值进行排序的一种存储结构 |
| 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。同时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 |  |  |
| 15 |  |  |

# 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 需求描述

### 2.3.1 图像特征提取

图像特征提取过程，既发生在图像入库阶段，也发生在图像检索阶段。图像入库阶段，对每一张入库的图像提取相关特征；图像检索阶段，需要先提取示例图像的图像特征，并与数据库中的图像特征进行比对，检索相似图像。

系统提供全局图像特征和局部图像特征的提取功能。全局图像特征包括CEDD、FCTH、PHOG等，局部图像特征包括sift、surf等。

系统提供基本的图像特征操作方法，包括获取图像特征类型名、获取特征不同数据类型的表示、计算两个图像特征的距离（相似度）等。

### 2.3.2 图像入库

系统提供简单的图像入库操作入口，使用户便利地调用实现图像入库操作。

系统提供基于全局图像特征的索引构造器和基于局部图像特征的索引构造器。

索引构造器可容纳多个图像特征提取器。索引构造器首先调用特征提取器对入库图像提取特征，将图像特征转化为Lucene索引文档中的一个Field域，若需要提取多种图像特征，则针对一种图像特征生成一个Field域，最后将Field域添加到Document中，一张入库图片对应一个Document。

### 2.3.3 索引生成

系统提供多种索引生成过程中需要的算法。

系统提供三种哈希算法，包括位采样哈希算法、局部敏感哈希算法和距离空间哈希算法。

系统提供并行化的索引生成算法。实现图像文件读取和图像特征提取的并行化操作，实现多种特征的并行化提取操作等。

### 2.3.4 图像检索

系统提供多种图像检索器，满足不同需求的图像检索操作。

图像检索器首先针对输入的示例图片，提取图像的特征。之后，读取图像特征库中已入库图像的特征索引，将入库图像特征与示例图像特征比对，计算特征距离，得到距离最近的前N张图片，并将图片生成结果序列，返回结果。

系统提供多种不同类型的图像检索器。比如，针对不同哈希模式的图像检索器，提针对不同Lucene索引存储形式（文本、DocValues等）的图像检索器。

系统提供不同的结果筛选策略。如，针对一种图像特征检索得到的结果序列，使用另外一种特征对之前的结果进行重排序。

# 3业务需求

## 3.1图像入库

用户希望将图像进行入库以供检索时，可以选定想入库的图片，然后选择想保存的特征，之后系统会提取相应图片的相应特征，并且构造出相应的索引，完成图像入库。

如果系统无法读取用户图片，系统会提示错误，入库失败。

如果系统无法写文件到本地，系统会提示错误，入库失败。

## 3.2 图像检索

用户希望检索某张图片时，可以选定想检索的图片，然后系统会提取图片的相应特征，并且与库中图片特征进行对比，最终用户得到检索结果。

如果系统无法读取用户图片，系统会提示错误，检索失败。

## 3.3 图像删除

用户希望删除库中某图片时，可以选定想删除的图片，然后系统会将指定图片的本地索引删除。

如果系统对本地索引没有读写权限，系统会提示错误，删除失败。

# 4功能需求

## 4.1 用例模型



图4. 1 用例图

## 4.2 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

### 4.2.2 索引生成



图4. 3索引生成RUCM模型

### 4.2.3 全局特征索引构造



图4. 4全局特征索引构造RUCM模型

### 4.2.4 局部特征索引构造



图4. 5局部特征索引构造RUCM模型

### 4.2.5 图像检索



图4. 6图像检索RUCM模型

### 4.2.6 图像特征提取



图4. 7图像特征提取RUCM模型

### 4.2.7提取全局特征



图4. 8提取全局特征RUCM模型

### 4.2.8 提取局部特征



图4. 9提取局部特征RUCM模型

### 4.2.9 特征距离计算



图4. 10特征距离计算RUCM模型

### 4.2.10 删除库中图像



图4. 11删除库中图像RUCM模型

# 5 非功能性需求分析

## 5.1 兼容性

作为一个跨平台的创建基于图像特性的Lucene索引的检索系统，系统应该具有比较强大的兼容性：

* 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。

* 数据兼容性

Lire提取图像的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的无关的方式储存。使不同平台上能够共享生成的图像数据库。

## 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，为了使系统运行更加高效，系统可以方便地引入新的技术、算法或模块，因此系统必须具备很好的可修改性或可扩展性。

## 5.3 高效性

本系统作提供了对图片的检索功能，需要支持对大规模数据的索引及搜索，因此需要具有很优秀的高效性。这主要体现在两个方面：

* 高效实现大规模数据实时索引入库

作为一个图像检索系统，如果提取图像特征或将索引入库占据了过多时间，那么必然影响了检索系统的实时性。因此程序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，因此程序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。

## 5.4 场景需求

### 5.4.1 基于Lire的检索引擎的应用场景

Lire是一个高效的，基于Java实现的图像检索系统。

图像检索是将图像的图像特征提取出来并生成索引库，可以根据给定的图片在图像数据库中进行检索，并返回相近或类似的图片。

由于图像具有很多个层次的信息，这就使得使用综合多个维度的图像信息才能够实现比较好的检索效果。

基于Lire的图像检索可以应用在人脸匹配，人脸识别，视频检索等等领域。

### 5.4.2 场景的存在的问题

1. 保证图像检索的查全率；（查全率=被检出相关信息量/相关信息总量（%））

2. 保证图像检索的查准率；（查准率=被检出相关信息量/被检出信息总量（%））

3. 大数据规模检索的效率；由于可能有大量检索的要求，需要保证检索的效率；如果Lire的设计或算法影响了效率，则需要尝试对Lire进行性能上的优化

4．图像相似度的度量标准的选择

# 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
* 编译环境：JRE6.0版本及以上。