直男杀手

软件需求规约

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **修订说明** | **作者** |
| 2018-4-14 | <1.0> | 第一次创建 | 徐迅，曹梦奇，卢锌原 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[1引言 1](#_Toc511739148)

[1.1编写目的 1](#_Toc511739149)

[1.2适用范围 1](#_Toc511739150)

[1.3定义 1](#_Toc511739151)

[1.4参考资料 2](#_Toc511739152)

[1.5概述 2](#_Toc511739153)

[2目前系统 3](#_Toc511739154)

[3建议的系统 3](#_Toc511739155)

[3.1概述 3](#_Toc511739156)

[3.2功能需求 3](#_Toc511739157)

[3.2.1 口红色号匹配 3](#_Toc511739158)

[3.2.2 口红试色 3](#_Toc511739159)

[3.3非功能需求 4](#_Toc511739160)

[3.3.1可用性 4](#_Toc511739161)

[3.3.2可靠性 4](#_Toc511739162)

[3.3.3性能 5](#_Toc511739163)

[3.3.4可支持性 6](#_Toc511739164)

[3.3.5设计约束 6](#_Toc511739165)

[3.3.6 接口 7](#_Toc511739166)

[3.3.7 法律、版权及其他声明 7](#_Toc511739167)

[3.3.8 适用的标准 8](#_Toc511739168)

[3.4 系统模型 8](#_Toc511739169)

[3.4.1场景 8](#_Toc511739170)

[3.4.2用例模型 9](#_Toc511739171)

[3.4.3用户界面 11](#_Toc511739172)

软件需求规约

# 1引言

## 1.1编写目的

本文档目的在于为目标读者（软件工程课程指导教师与软件开发组成员）提供关于本软件的需求功能的详述以及系统性能参数的说明。建立系统逻辑模型，准确表达用户的需求。

## 1.2适用范围

本文档主要涉“直男杀手”设计要求与约束，定义了系统的非功能性需求和一系列通过用例说明的功能性需求。

## 1.3定义

本文档中出现的专门术语：

|  |  |
| --- | --- |
| 术语 | 定义描述 |
| 口红Key值 | 在存储口红信息进数据库时需要对这些信息进行处理，其中很重要的就是生成key值。对于每一个品牌、每一个色号的口红，它会对应到一个独一无二的key值。 |
| 网络爬虫 | 网络爬虫是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本。在爬取信息过程中一般基于目标网页特征、目标数据模式进行有选择性的信息筛选和信息处理。 |
| 人脸识别 | 面部识别又称人脸识别、面像识别、面容识别等等，面部识别使用通用的摄像机作为识别信息获取装置。以非接触的方式获取识别对象的面部图像，计算机系统在获取图像后与数据库图像进行比对后完成识别过程。 |
| 驼峰命名法 | 骆驼式命名法（Camel-Case）又称驼峰式命名法，是电脑程式编写时的一套命名规则（惯例）。正如它的名称CamelCase所表示的那样，是指混合使用大小写字母来构成变量和函数的名字。程序员们为了自己的代码能更容易的在同行之间交流，所以多采取统一的可读性比较好的命名方式。 |
| 匈牙利命名法 | 匈牙利命名法是一种编程时的命名规范。基本原则是：变量名=属性+类型+对象描述，其中每一对象的名称都要求有明确含义，可以取对象名字全称或名字的一部分。要基于容易记忆容易理解的原则。保证名字的连贯性是非常重要的。 |
| OpenCV | OpenCV是一个基于BSD许可（开源）发行的跨平台计算机视觉库，可以运行在Linux、Windows、Android和Mac OS操作系统上。它轻量级而且高效——由一系列 C 函数和少量 C++ 类构成，同时提供了Python、Ruby、MATLAB等语言的接口，实现了图像处理和计算机视觉方面的很多通用算法。 |
| Pycharm | PyCharm是一种Python IDE，带有一整套可以帮助用户在使用Python语言开发时提高其效率的工具，比如调试、语法高亮、Project管理、代码跳转、智能提示、自动完成、单元测试、版本控制。此外，该IDE提供了一些高级功能，以用于支持Django框架下的专业Web开发。 |

本文档中出现的缩略词：

|  |  |  |
| --- | --- | --- |
| 缩略词 | 全称 | 定义描述 |
| GBK | 汉字内码扩展规范 | 中华人民共和国全国信息技术标准化技术委员会1995年12月1日制订，国家技术监督局标准化司、电子工业部科技与质量监督司1995年12月15日联合以技监标函1995 229号文件的形式，将它确定为技术规范指导性文件。 |

## 1.4参考资料

1. Perlibakas V. Distance measures for PCA-based face recognition[J]. Pattern Recognition Letters, 2004, 25(6):711-724.
2. Bruegge B, Dutoit A H. Object-Oriented Software Engineering Using UML, Patterns, and Java[M]. 清华大学出版社, 2011.
3. 王云, 周伯生. 标准建模语言UML简介[J]. 计算机应用研究, 1999(12):44-49.
4. Roger S. Pressman. Software Engineering: A Practitioner's Approach[M]. McGraw-Hill Book Co, 1982.
5. 赵本本, 殷旭东, 王伟. 基于Scrapy的GitHub数据爬虫[J]. 电子技术与软件工程, 2016(6):199-202.

## 1.5概述

本文档其他各部分包含了基于人脸识别的此口红试色软件的整体说明、功能需求与非功能性需求分析、以及系统模型规定。

# 2目前系统

目前系统刚刚批准进入设计阶段，没有老系统。市场上已存在诸如“美妆相机”、“千妆魔镜”等口红试色APP，本软件旨在利用网络爬虫技术获得市场口红相关参数，利用openCV与原创算法实现基本人脸识别功能，在实现已有试色APP的功能同时优化算法与软件性能。

# 3建议的系统

## 3.1概述

本项目包含的软件功能需求包括：

* 口红色号匹配：利用人脸识别技术获得图片中同款口红参数；
* 口红试色：挑选一款或多款口红进行试色。

本项目包含的软件非功能性需求包括：

* 可用性需求：用户培训时间、图形标准、子系统的可用性、系统可维护性；
* 可靠性需求：系统开放时间与可持续运行时间、操作权限、MTBF与MTTR、最高代码错误率、错误率；
* 性能需求：对事务的响应时间（平均、最长）、吞吐量、容量、资源利用情况；
* 可支持性需求：编码标准、命名约定、类库、维护访问权和维护实用程序；
* 设计约束需求：编程语言、开发工具、系统开发流程、兼容性约束、测试约束；
* 接口需求：用户接口、软件接口、硬件接口、通信接口；
* 法律、版权及其他声明。

## 3.2功能需求

### 3.2.1 口红色号匹配

用户上传图片；本软件对图片进行离线人脸识别，尤其对唇部进行色彩识别，在后台数据库中查询相似唇色的口红款式，利用我们的比较算法，返回与图片唇色最相似的色号，以及该色号的热卖口红款式。

### 3.2.2 口红试色

用户文字或图片搜索数据库中的口红信息，挑选出想要试色的口红（一款或多款）；本软件对图片进行离线人脸识别，对唇部上色，供用户查看试色效果。

## 3.3非功能需求

### 3.3.1可用性

#### 3.3.1.1 用户培训时间

在打开软件之后会提示是否进入新手教程界面，如是则进入新手教程界面进行软件操作培训，平均培训时间为1min。

#### 3.3.1.2 图形标准

本软件使用PyQt进行GUI开发，图形使用符合Microsoft的GUI设计要求。

#### 3.3.1.3 子系统的可用性

子系统包括，各品牌口红信息的分类，基于关键字的数据库搜索等。经初步分析，在一定容量下，各系统均可正常工作。

#### 3.3.1.3 系统的可维护性

系统未设置降级模式。采用模块化编程，方便子系统功能更改与数据库更新。

### 3.3.2可靠性

#### 3.3.2.1 系统开放时间与可持续运行时间

据作者保守估计，系统可用时间达到90%，持续可运行时间可达1000h。

#### 3.3.2.2 操作权限

用户不能对软件系统数据进行直接访问与更改。

#### 3.3.2.3 平均故障间隔时间(MTBF)与平均修复时间(MTTR)

平均故障间隔时间是指两次相邻故障之间工作时间平均值，本口红试色软件预计平均故障间隔时间为不少于六个月；

平均修复时间指系统在运行期间解除故障所需时间，本口红试色软件预计平均修复时间要小于3小时。

#### 3.3.2.4 最高代码错误率

最高错误或缺陷率通常表示为bugs/KLOC(每千行代码的错误数目)或bugs/function-point(每个功能点的错误数目) 。本口红试色软件要求每千行代码的错误数目少于50个。

#### 3.3.2.5 错误率

小错误：指页面显示异常，某些口红图片无法显示、带人脸的图片无法识别，

错误率：小于10% ；

大错误：对于用户的输入或命令数据库无响应，

错误率：小于3%；

严重错误：系统崩溃，数据库运行失常造成数据存储及读取命令无效，数据泄露，

缺陷率： 避免发生。

### 3.3.3性能

#### 3.3.3.1 事务的响应时间（平均、最长）

本软件可离线运行，在不受硬件与处理器性能影响的情况下，平均响应时间为3s，最长响应时间为10s。

#### 3.3.3.2 吞吐量

本软件每秒处理的图片数不超过100张，数据库访问配额控制在5000（次/分钟）。

#### 3.3.3.3 容量

为保证用户体验，根据现有数据结构进行分析，本软件数据库中最多可存储10k条口红信息。

#### 3.3.3.4 资源利用情况

为确保本软件开发正常进度，开发过程需要至少三台PC机（多核、内存1GB、硬盘空间20G，可连接Internet，网络连接速度10MB/s）。

### 3.3.4可支持性

#### 3.3.4.1 编码标准

软件统一使用GBK编码标准。

#### 3.3.4.2 命名约定

变量命名使用匈牙利命名法规则；

类名、函数名使用驼峰命名法；

命名全部采用英语，不允许出现汉语拼音。

#### 3.3.4.3类库

现阶段已使用python3类库（待扩充）：

Import scrapy

Import lxml

Import json

from PyQt5 import QtCore

from PyQt5 import QtGui

from PyQt5 import QtWidgets

#### 3.3.4.4 维护访问权和维护实用程序

用户具有上传图片、一键匹配口红型号、搜索并访问口红信息等权限。

### 3.3.5设计约束

#### 3.3.5.1 编程语言

网络爬虫采用python3编写，数据库使用python3语言进行搭建与使用。

#### 3.3.5.2 开发工具

在PyCharm 环境下进行python3代码编写。

#### 3.3.5.3 系统开发流程

首先使用python的scrapy库完成大范围爬虫，之后利用PyQt完成基本代码编写，最后进行第二轮爬虫扩充数据库容量并不断测试。

#### 3.3.5.4 兼容性约束

测试服务器系统要求为Windows，最低版本为Windows 7。

#### 3.3.5.5 测试约束

测试在PyCharm 环境下进行结束后，上传软件安装包到服务器进行第二轮调试。

### 3.3.6 接口

#### 3.3.6.1用户界面

本软件通过PC机进行运行、操作。输入输出的相对时间将由PC机本身的处理速度来决定。

#### 3.3.6.2 硬件接口

本软件不需要特定的硬件或硬件接口进行支撑。

#### 3.3.6.3 软件接口

本软件不需要特定的软件或软件接口进行支撑。

#### 3.3.6.4通信接口

本软件没有特殊的通讯接口，通讯接口由所使用的PC机决定。

### 3.3.7 法律、版权及其他声明

用户使用口红匹配试色服务时应遵守的重要条款：

(1) 本软件提供的所有信息仅供个人参考使用；

(2) 本软件提供的所有信息均由网络爬虫技术提供，如涉及口红品牌知识产权问题，我方将根据中国大陆相关法律法规进行处理；

(3) 不得使用本软件从事违法活动；

(4) 利用本软件从中国境内向境外传输技术性资料时不得违反中国有关法律、法规。

### 3.3.8 适用的标准

暂无具体标准。

## 3.4 系统模型

### 3.4.1场景

<场景一>

|  |  |
| --- | --- |
| 场景名称 | 文字查询口红 |
| 参与者实例 | UserA：用户 |
| 事件流 | 1. UserA进入文字搜索界面 2. UserA将想要查询的关键词（颜色、品牌、色号）输入在文字搜索框输入给软件 3. UserA点击文字搜索按钮 4. 软件分析文字内容并得到相关口红信息，在数据库中进行比对，并进行排序，返回最优秀的结果。 5. 软件得到结果并显示在用户界面上 |

<场景二>

|  |  |
| --- | --- |
| 场景名称 | 图片查询口红 |
| 参与者实例 | UserA：用户 |
| 事件流 | 1. UserA进入图片搜索界面 2. UserA将想要查询的图片（带有清晰人脸、并在唇上有想要查询的颜色）上传给软件 3. UserA点击图像搜索按钮 4. 软件解析图片内容并得到相关口红信息，在数据库中比对，并进行排序，返回最为优秀的结果。 5. 软件得到结果并显示在用户界面上 |

<场景三>

|  |  |
| --- | --- |
| 场景名称 | 口红试妆 |
| 参与者实例 | UserA：用户 |
| 事件流 | 1. UserA进入试妆界面 2. UserA输入想要试用口红的Key值（可以通过文字、图片搜索来获得，或直接选择；每个品牌——色号 对应一个key值），并上传希望试妆的脸部图片 3. UserA点击试妆按钮 4. 软件调用试妆功能 5. 软件获得结果并返回在用户界面上 |

### 3.4.2用例模型

a)

|  |  |
| --- | --- |
| 参与者名称 | 参与者解释 |
| 用户 | 软件用户、软件的使用者 |

b)

|  |  |  |
| --- | --- | --- |
| 用例 | 用例级别 | 用例描述 |
| 图像搜索 | 用户目标级别 | 用户输入图片，软件解析图片获得相关信息。再以这信息为输入调用搜索这一用例。并在用户界面上进行显示。 |
| 文字搜索 | 用户目标级别 | 用户输入文字，软件解析图片获得相关信息。再以这信息为输入调用搜索这一用例。并在用户界面上进行显示。 |
| 搜索 | 子功能 | 图片搜索、文字搜索调用，根据相关信息，在数据库中比对，并进行排序，最后返回最优结果。 |
| 试妆 | 用户目标级别 | 用户输入代表一种品牌——色号的key值，并输入自己想要试妆的带有脸部的照片，软件进行模拟试妆，并返回结果 |

C）

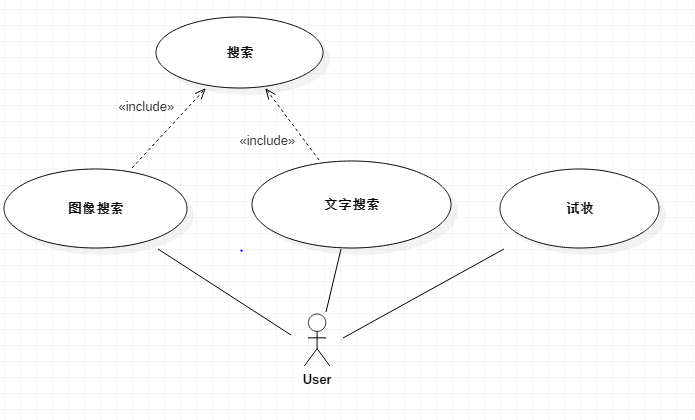
|  |  |
| --- | --- |
| 用例名称 | 图像搜索 |
| 参与者 | 用户 |
| 入口条件 | 软件正常工作 |
| 出口条件 | 用户获得搜索结果后离开软件 |
| 事件流 | 1. 用户选择并进入图像搜索界面 2. 用户上传想要搜索的图片 3. 软件解析图片并获取相关信息，调用搜索用例。并获得结果。 4. 将结果呈现在用户界面上 |
| 质量需求 | 整个搜索过程现在30秒内实现 |

|  |  |
| --- | --- |
| 用例名称 | 文字搜索 |
| 参与者 | 用户 |
| 入口条件 | 软件正常工作 |
| 出口条件 | 用户获得搜索结果后离开软件 |
| 事件流 | 1. 用户选择并进入文字搜索界面 2. 用户输入想要搜索的文字描述 3. 软件分析文字并获取相关信息，调用搜索用例。并获得结果。 4. 将结果呈现在用户界面上 |
| 质量需求 | 整个搜索过程现在30秒内实现 |

|  |  |
| --- | --- |
| 用例名称 | 搜索 |
| 参与者 | 无 |
| 入口条件 | 软件进入文字搜索或图像搜索功能，并已经解析出相关口红信息 |
| 出口条件 | 返回相关结果给文字搜索或图片搜索功能 |
| 事件流 | 1. 软件获得相关信息 2. 在数据库中比对，并获得相关项 3. 对相关项进行排序 4. 返回排序后的最优结果 |
| 质量需求 | 整个搜索过程现在15秒内实现 |

|  |  |
| --- | --- |
| 用例名称 | 试妆 |
| 参与者 | 用户 |
| 入口条件 | 软件正常工作 |
| 出口条件 | 用户获得试妆结果后离开软件 |
| 事件流 | 1. 用户选择并进入试妆界面 2. 用户输入想要试妆口红的key值（每一个品牌——色号在本系统中都一一对应一个key值），并上传想要试妆并带有脸部的图片。   3、软件根据key值获得相关口红信息，并分析图片进行模拟试妆  4、将结果呈现在用户界面上 |
| 质量需求 | 整个试妆现在一分钟内实现 |

d)



### 3.4.3用户界面

主界面共有三个：

1. 文字搜索：这一界面有一文字输入框来输入相关文字描述，并由一个搜索按钮来启动文字搜索功能。点击后会跳转到搜索结果页面，显示排序后的最优的口红匹配项（包括品牌、颜色、价格、key值）
2. 图像搜索：这一界面包含一上传按钮来上传想要搜索的图像，并由一个搜索按钮来启动图像搜索功能。点击后会跳转到搜索结果页面，显示排序后的最优的口红匹配项（包括品牌、颜色、价格、key值）
3. 试妆：这一界面包含一个key值输入框来输入key值，有一上传按钮来上传想要试妆的图像，并由一个试妆按钮来启动试妆功能。点击后会跳转到结果页面，显示模拟试妆后的图片。
4. 三个页面之间都包含点击链接来相互转换。

此外，新手指南界面正在设计中，但不作为此软件的主要开发内容。