<智能人形识别>

软件构架文档

版本 <1.1>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 21/10/2018 | 1.1 | 软件构架文档初稿 | 许晓斐、马振文、谭锦豪、封琪 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 参考资料 4

2. 构架目标和约束 4

3. 用例视图 4

3.1 用例实现 4

3.1.1 系统配置 4

3.1.1 获取分析报表 4

4. 逻辑视图 5

4.1 概述 5

4.2 在构架方面具有重要意义的设计包 6

5. 部署视图 7

6. 数据视图（可选） 8

7. 质量 9

7.1 可扩展性设计 9

7.2 可靠性设计 9

7.3 可移植性设计 9

7.4 性能性设计 9

软件构架文档

# 简介

## 目的

本文档将从软件架构层次对智能人形识别系统进行综合概述，以多种不同的架构视图对系统的各个方面进行描述。它用于记录并表述已对系统的构架方面做出的重要决策。

## 范围

本文档用于高级软件开发课程第六小组正在开发中的智能人形识别系统。智能人形识别系统是为工厂提供基于图像处理技术的智能视频监控系统，旨在代替人工监测，对操作人员的安全着装等进行智能识别，以达到维护工厂安全管理的目的。

## 参考资料

RUP的软件架构文档模板。

# 构架目标和约束

架构目标：

(1) 支持本系统所有功能需求和非功能需求的实现；

(2) 能够灵活地增加新的模型和算法。

架构约束：

应采用C/S架构。

# 用例视图

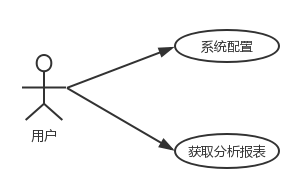


图1 系统用例图

## 用例实现

* + 1. 系统配置

用户通过在web页面上设置安全规则，来规定某一摄像头所对应场景的目标识别策略。

用户可配置的规则如下：

（1）是否为无人区

（2）是否需要佩戴安全帽

* + 1. 获取分析报表

用户可以查看系统对每一场景反馈的安全分析报表，包括了报警次数、报警时间等信息。

# 逻辑视图

## 概述

包图如下：

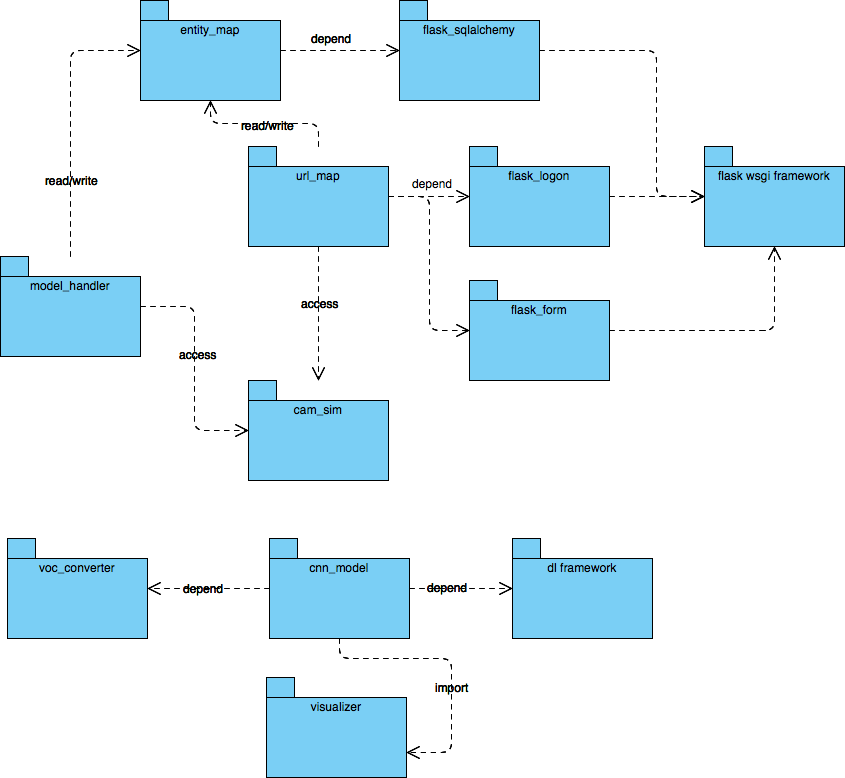


图2 系统包图

整个系统分为Web服务器（上半部分）与模型训练机（下半部分）两部分。

其中web服务器核心为url\_map，entity\_map。前者负责对前端请求的映射，后者负责持久化存储的实体映射。

模型训练机核心为cnn\_model。

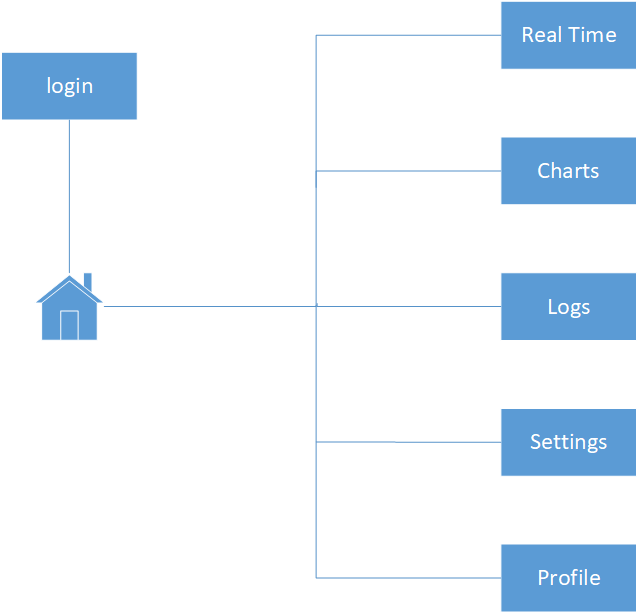


图3 网站导航图

如图为用户页面的网站导航图。

## 在构架方面具有重要意义的设计包

* + 1. url\_map

响应前端请求的映射。

* + 1. entity\_map

负责orm层的实体映射，详见er图

* + 1. cam\_sim

负责模拟摄像头的信号输入。

主要类为Camera。其Getframe()方法将从视频信号中抽取出帧。

* + 1. cnn\_model

模型训练机的主要封装，依赖于外部的深度学习框架。

主要类为HyperParameter，作为深度学习的参数的封装。

# 部署视图

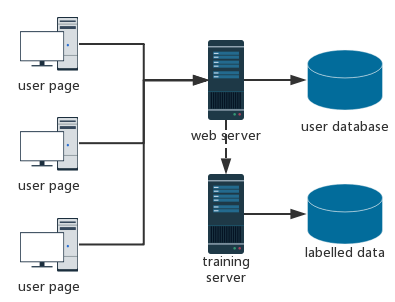


图4 系统部署图

（1）user page

用户页面通过向web服务器发送HTTP请求，获取服务器响应。

（2）web server

web服务器响应用户页面的请求。web服务器的行为分为两部分，收到网页前端的请求后，通过HTTP向用户数据库调用数据，或是向模型训练机调用检测数据，并反馈给前端。

（3）training server

模型训练机基于tensor flow框架，提供训练模型，反馈检测结果的服务。通过向训练数据库调用监控视频文件作为训练样本，对模拟摄像头的视频帧信号进行目标识别，并反馈给web服务器。

（4）user database

用户数据库储存操作人员的用户名、密码信息，提供web服务器调用，响应用户的登入登出操作。

（5）labelled data

训练数据库储存监控视频样本，提供模型训练机调用。

# 数据视图（可选）

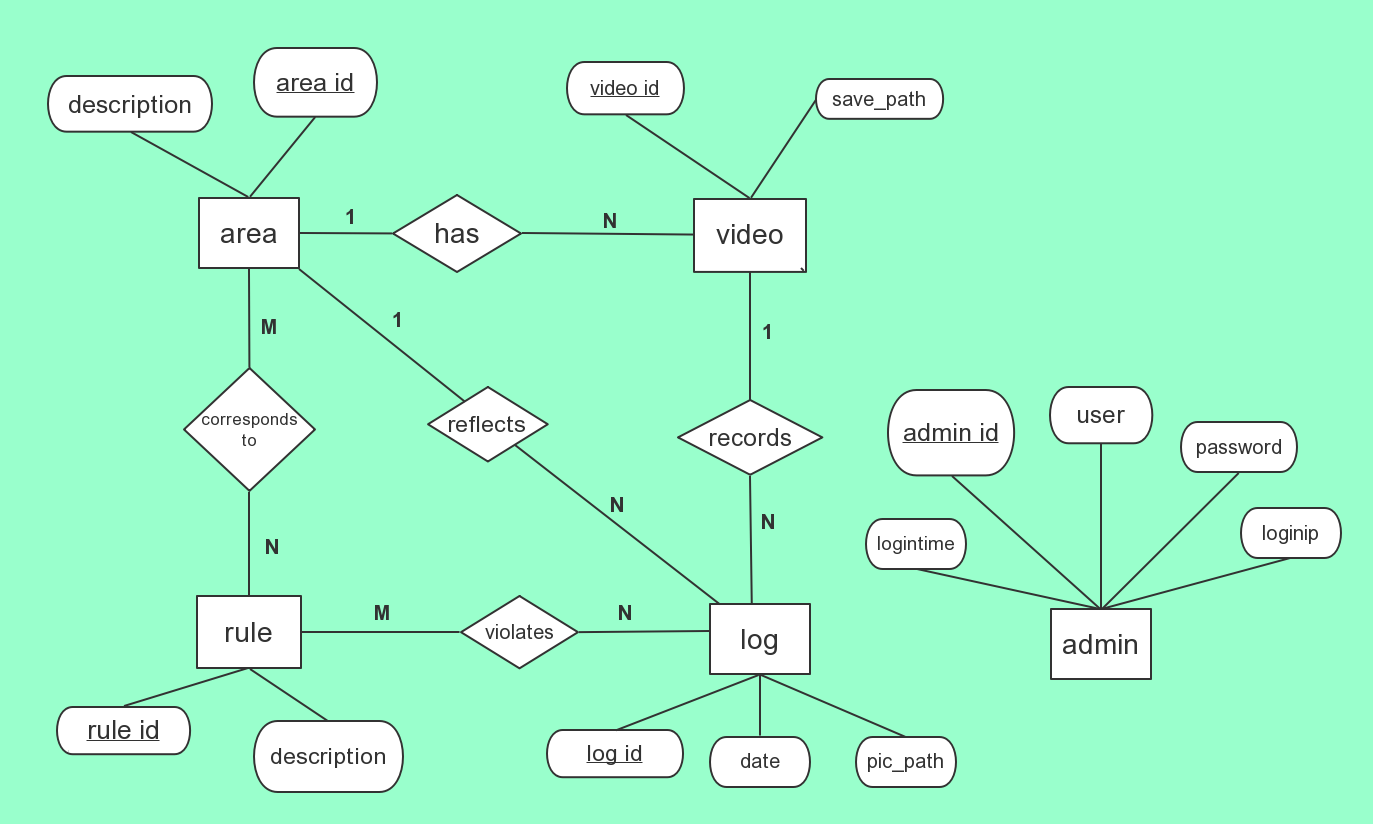


图5 数据视图

由此，可以建立起以下这些关系模式（下划线表示主键，斜体表示外键）：

|  |  |
| --- | --- |
| area | id, description |
| rule | id, description |
| video | id, save\_path, *area\_id* |
| log | id, date, pic\_path, *area\_id*, *video\_id* |
| rule\_log | id, *rule\_id*, *log\_id* |
| area\_rule | id, *area\_id*, *rule\_id* |
| admin | id, user, password, logintime, loginip |

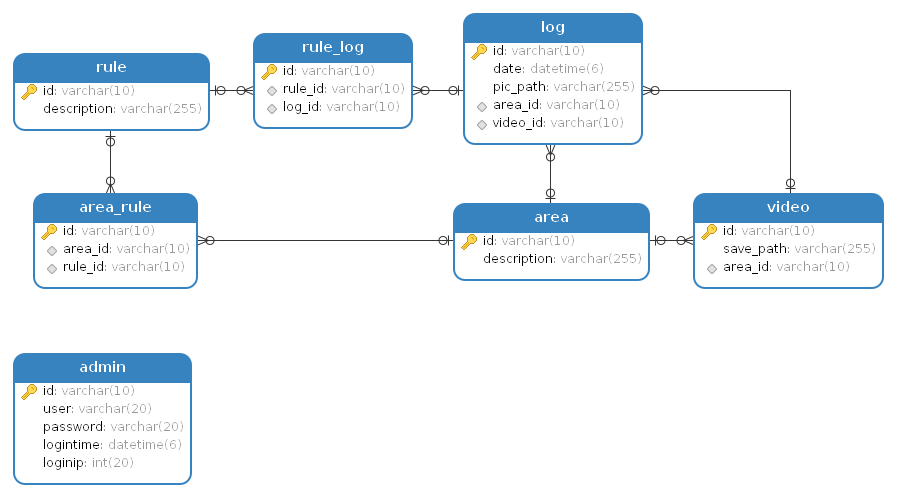


图6 E-R图

# 质量属性的设计

## 可扩展性设计

需求：

可以添加新的监控地区，对不同的地区可以划分危险等级，并设置它们的监视类别，监控等级

设计策略：

对模型进行分解，分解成几个单一的模型，这些模型之间可以自由组合，这样就可以生成不同的监视类别，不同的组合可以对应各个危险等级

## 可靠性设计

需求：

软件的故障次数应小于每月2次，最大延时不能超过1分钟

设计策略：

对软件进行测试，减少bug的数量，同时对模型进行优化分解，提高性能以减少故障次数。

## 可移植性设计

需求：

软件可以运行在多个平台，不同的电脑操作系统上

设计策略：

软件架构使用B/S架构，对不同的游览器进行兼容，以适应不同的操作系统

## 性能设计

需求：

1可检索的报警日志时间范围为3个月，检索的响应时间小于0.5秒。

实时报警的计算延迟不超过10秒。

2监控视频在识别人物的同时不能有明显的卡顿现象

设计策略：

1可以先获取部分检索数据，显示响应前端，再对剩下的数据不断获取处理

2模型剪裁，对计算出的模型进行模型剪裁，优化模型以提高性能。