

# Samaritan：一个利用人脸识别技术实现的课堂点名系统

邵镇炜，刘悦萱

December 30, 2019

## Abstract

本项目旨在实现一个以微信小程序为用户平台的，以人脸识别技术为核心的课堂点名系统，我们称之为Samaritan点名系统。课堂点名是校园生活中的一大需求，对高效性和准确值的要求是该需求的重要特点，而现有的点名系统均不能很好的满足这两大需求点。而随着计算机视觉与深度学习研究的蓬勃发展，人脸识别技术已趋成熟且已有许多的成功的应用案例，这使得利用人脸识别技术进行课堂点名成为可能。本文将围绕该项目背景讨论本项目的重要性与优势，然后从实用性、可实现性等角度提出本项目的实现方案，并对整个系统的架构和各模块的功能与实现方案均作了必要的阐述，最后通过对本项目的软硬件需求、项目进度安排及项目组分工情况的简要的介绍，进一步论证了该项目的可行性。本项目采用结对编程的模式和敏捷开发的原则进行开发活动，并使用Teambition工具进行项目进度的规划与管理。

# Contents

<b>Abstract</b>	<b>1</b>
<b>1 项目背景</b>	<b>3</b>
1.1 现存系统的缺陷	3
1.2 人脸识别技术的发展	3
1.3 基于人脸识别技术的课堂点名系统	3
<b>2 可行性与需求分析</b>	<b>3</b>
2.1 可行性研究	3
2.1.1 硬件设备要求	3
2.1.2 软件资源	3
2.1.3 成本与经济效益	4
2.1.4 项目进度安排与分工	4
2.2 需求分析	4
2.2.1 项目目标与一般约束	4
2.2.2 功能清单	5
<b>3 详细设计</b>	<b>6</b>
3.1 系统总体架构	6
3.2 关键性API	6
3.3 前端设计	7
3.4 业务端设计	8
3.5 算法端设计	9
<b>4 测试</b>	<b>11</b>
4.1 单元测试	11
4.1.1 照片方向问题	11
4.2 深度学习接口多次调用问题	11
4.3 集成测试	12
<b>5 总结</b>	<b>14</b>

# 1 项目背景

## 1.1 现存系统的缺陷

点名签到是高校中最常见的应用需求之一，无论是会议、活动还是课堂、讲座，都有签到需求。通过点名的方式来检查学生出勤情况，以达到对学生的状态进行有效的监管与督促的目的；点名活动通常在课堂上进行，往往会占用课堂实践。因此准确、高效是点名活动最自然的两大诉求。

现有的签到方式以纸质签到和校园卡刷卡签到为主，新兴的签到方式有手机APP点名系统等。这几类签到方式都存在诸多弊端：纸质签到在面对一些大型会议和活动时，面临着签到效率低下、需要进行二次数据整理录入才能完成统计分析的问题；校园卡刷卡签到模式依赖于校园卡、读卡器以及安装了校园卡签到客户端程序的计算机，且每次开会都需要工作人员提前准备并在现场值守，导致签到人力物力消耗过大，签到效率低下；手机APP点名系统多采用老师报验证码，学生输入验证码签到的方式检查出勤情况，一方面限时的输入验证码容易导致漏签，另一方面不在场的学生也可能获取验证码，即使逃课也能完成签到，两方面的原因均会导致签到信息失去其准确性。因此我们有必要实现一种更便捷的、更可靠的点名系统。

## 1.2 人脸识别技术的发展

随着深度学习的发展与成功，越来越多的深度学习技术被应用于生产实践当中。尤其是，计算机视觉相关的深度学习技术的成熟催生了大量便利人们生活与娱乐的应用与产品，人脸检测、人脸识别技术便是其中最热门的研究与应用领域之一。

在20世纪60年代的时候，人们已经开始研究人脸识别技术了。2015年Google的论文 [1] 验证了使用深度学习模型得到的人脸特征的通用性，并提出了围绕特征进行人脸识别、人脸验证等多种任务的技术范式。之后更多研究 [2]、[3]、[4]等在此基础上进行了深入的探索，将人脸识别的准确率推向了高峰。由于人脸信息的特殊性，使得人脸信息可以作为身份验证的依据，因此该技术具有广泛的应用场景和良好的应用前景。

有赖于移动设备上摄像头的像素越来越高和深度学习技术的蓬勃发展，人脸识别技术的应用迎来了新的机遇。2010年底，Facebook推出了人脸识别技术的“Tag Suggestions”功能；与此同时，谷歌也于2011年12月在Google+上推出了类似功能的“Find My Face”；Face.com公司也在App Store中推出了应用人脸识别技术实现的手机拍照应用KLIK。

## 1.3 基于人脸识别技术的课堂点名系统

综上所述，随着人脸识别技术迅猛发展、移动设备智能化程度越来越高，不难发现，依托智能手机平台、基于人脸识别技术的课堂点名系统具有重要的研究意义与广阔的应用前景。与现有的点名系统的比较可以发现，对全班学生进行拍照，对照片进行人脸识别的方式来进行点名是一个更高效也更准确的方案。

# 2 可行性与需求分析

## 2.1 可行性研究

### 2.1.1 硬件设备要求

本项目的硬件设备需求为手机、云服务一台（不带GPU）、学校服务器一台（带GPU）。本项目中的前端将在手机微信小程序中运行。由于微信小程序的request操作仅支持公网域名，我们需要用云服务器和内网穿透技术（<https://github.com/fatedier/frp>）将公网域名映射到学校内网的一台服务器上。我们将在学校服务器上部署业务服务端和人脸识别服务端，其中人脸识别服务端程序需要使用一块GPU。GPU仅负责特征推理，本项目不需要进行模型训练。上述硬件设备需求均已得到满足。

### 2.1.2 软件资源

本项目使用Django框架和Flask框架搭建服务，使用Mxnet深度学习框架部署深度学习模型。深度学习

模型代码参考[https://github.com/kpzhang93/MTCNN\\_face\\_detection\\_alignment](https://github.com/kpzhang93/MTCNN_face_detection_alignment)和<https://github.com/deepinsight/insightface>。两个代码库均提供了预训练模型可供使用，因此本项目的开发工作主要在各项功能的实现和衔接。

### 2.1.3 成本与经济效益

开发测试阶段，除了租用云服务器的费用外，无其他费用。后期推广应用，在微信小程序平台上或需要额外费用。大部分代码可直接移植于易班等APP，有机会借助易班等平台推广本项目，同时也可以完善易班APP本身的点名功能。因此本项目成本在可接受范围之内，且效益可观。

### 2.1.4 项目进度安排与分工

本项目由邵镇炜负责人脸识别算法系统开发（包括单人特征抽取、多人人脸识别及人脸服务部署）并辅助部分业务系统的开发，刘悦萱负责业务系统开发（微信前端、业务后端及数据库设计），最后由两人配合完成各模块的衔接和调试工作，每一阶段的文档工作也均由两人共同完成。本项目的项目进度使用Teambition（<https://www.teambition.com>）工具进行管理与调整，具体时间安排如图1所示

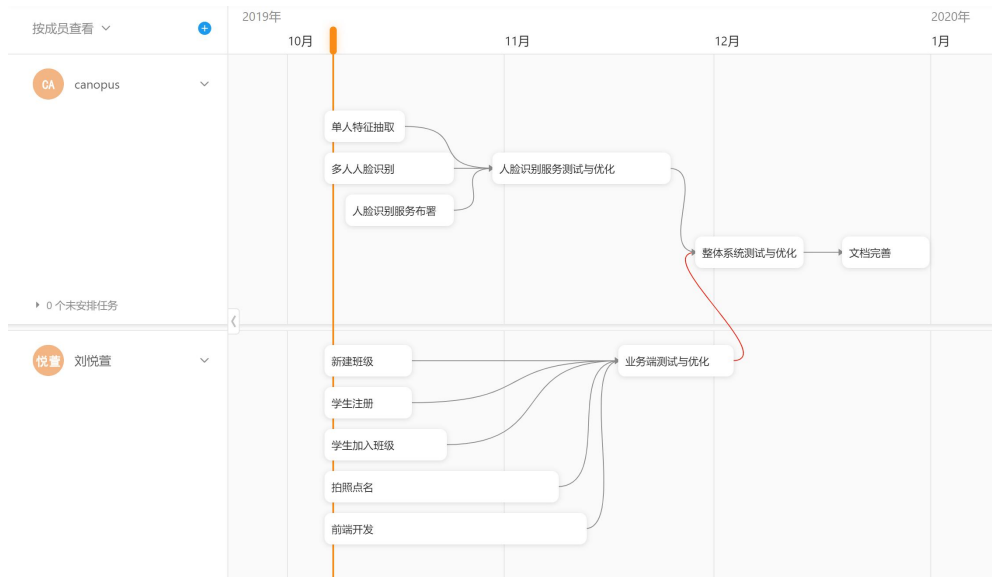


Figure 1: 项目进度与任务分配

## 2.2 需求分析

### 2.2.1 项目目标与一般约束

本软件项目中的主要角色有老师、学生。本项目的主要应用场景为大学校园中的课堂点名，鉴于初高中不允许学生带手机进校园，或难以推广；另一方面，初高中课堂对点名的需求相对较弱。本小节将对软件的需求进行分析。其中该软件的基本约束有：

- 实时性：要求点名时尽快反馈到勤缺勤名单
- 准确性：人脸识别要求较高的准确性以完成点名
- 安全性：对人脸等隐私信息应有一定的保护或加密措施
- 多平台性：使得使用不同终端的老师和学生都可以使用本系统

### 2.2.2 功能清单

对本软件所需功能的分析具体见下表。

角色	功能项	功能项描述
用户 (通用)	1.注册	注册成为系统的新用户
	2.登录	进入系统, 获取系统内的操作权限
	3.登出	退出系统
	4.个人信息维护	设置用户姓名、工号/学号等
	5.找回密码	用户忘记密码时可通过邮箱找回密码
	6.修改密码	用户通过邮箱验证身份, 重设密码

Figure 2: 用户 (通用) 需求

角色	功能项	功能项描述
老师	1.创建教学班	上传教学班基本信息, 获得教学班 id
	2.查看任课班级	查看本学期该教师任课的所有班级, 及各班级基本信息、班内出勤情况
	3.修改任课班级信息	修改所任教班级的基本信息
	4.拍照点名	对教室拍照, 获得缺勤与到勤同学名单
	5.手动修正	对人脸识别得到的缺勤到勤名单进行手动修正
	6.记录请假	将某学生的某次出勤状态调整为请假

Figure 3: 教师需求

角色	功能项	功能项描述
学生	1.加入教学班	输入教学班 ID, 加入教学班
	2.查看选课情况	查看本学期所选课程基本信息
	3.上传照片	上传本人大头照作为人脸识别的标准参照
	4.修改照片	重置本人大头照
	5.查看出勤历史	查看本学期至今的到勤与缺勤次数

Figure 4: 学生需求

对上述需求进行进一步的分析, 我们将点名系统抽象为如下的数据流模型。

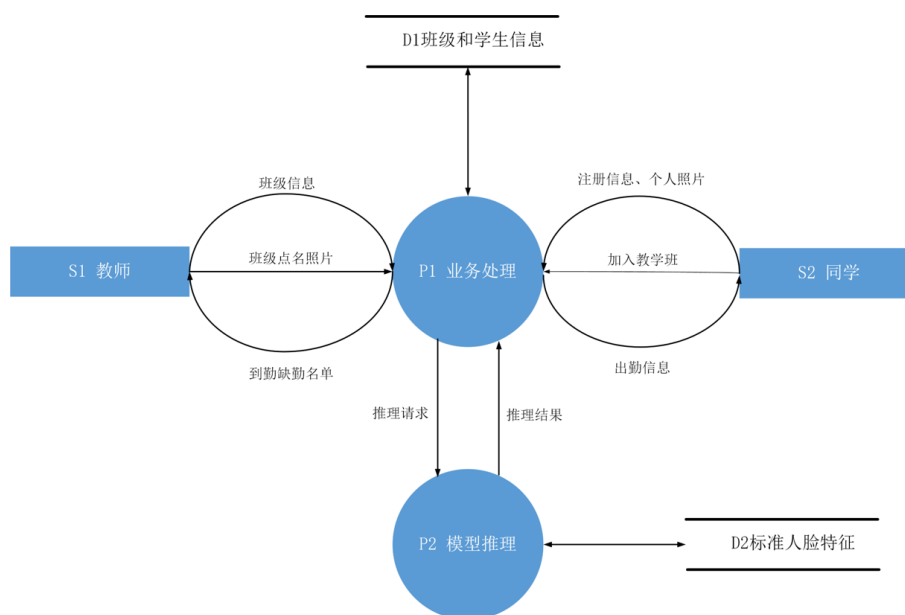


Figure 5: Samaritan点名系统数据流图

### 3 详细设计

本章介绍Samaritan点名系统的设计方案与关键技术。通过对现有系统的逻辑和功能分析和对Samaritan点名系统的需求的分析，我们提出了一个可行的实现方案。下面将从系统总体架构，各模块的功能与设计对该实现方案进行具体阐述。

#### 3.1 系统总体架构

Samaritan点名系统主要分为用户界面，业务后端服务系统及人脸识别服务系统三个模块，采用前端与后端分离、业务与算法模块分离的架构方式；各系统代码相互独立，通过API实现交互，以达到低耦合、高内聚的效果，便于各模块的独立开发与调试。Samaritan点名系统的总体架构如图6

其中用户界面基于微信小程序生态来实现，主要负责界面展示并与用户交互，接收用户填写的注册信息、上传的照片等数据并上传给业务后端。业务后端服务为前端用户界面提供API支持，业务后端服务基于Django框架实现，负责处理用户注册，用户信息维护和点名事件处理等主要业务逻辑，并在需要执行模型推理时向人脸识别服务系统提出相应请求。人脸识别系统利用Flask框架建立服务并实现API供业务后端服务使用，人脸识别系统内使用Mxnet深度学习框架部署深度学习模型。人脸识别系统通过人脸检测、人脸对齐、人脸识别等算法实现单人人脸特征抽取，多人人脸匹配等主要功能。

#### 3.2 关键性API

本小节介绍业务端和算法端进行交互的关键性API。

**单人照片处理：**

- 接收参数：用户id、用户照片，提取此用户照片的特征值，以备人脸识别主算法使用
- 返回值定义：0:成功，1:照片不存在，2:特征文件丢失，3:照片质量太差

**课堂多人照片处理：**

- 接收参数：班内所有学生的学号列表
- 返回值定义：未到学生的学号列表、识别结果图片

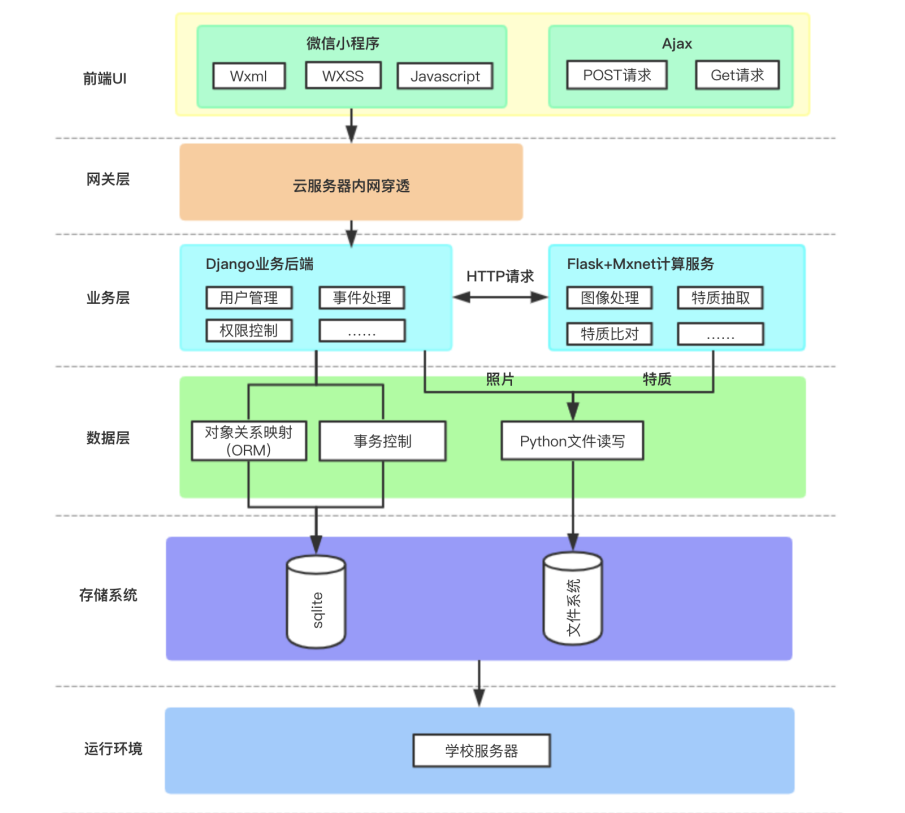


Figure 6: Samaritan点名系统总体架构

### 3.3 前端设计

前端用户界面基于微信小程序搭建，具有跨平台特性。微信小程序框架采用视图与数据逻辑分离的设计模式，分为视图层(WXML、WXSS)和逻辑层（JavaScript）两部分，逻辑层通过响应式数据绑定技术来与视图实现数据同步，从而使页面内数据可以利用Ajax技术局部、独立地更新，让开发更专注于数据和逻辑。

前端总共由四类页面构成：

- 通用页面（包括登陆页面、照片上传页面、用户注册弹窗）
- 学生用户专属页面（学生基本信息页面、学生课程考勤详情）
- 教师专属页面（教师基本信息页面、所属课程考勤详情；
- 签到主功能页面（签到照片上传页面、签到详情展示及信息修正页面）

前端界面的效果展示如下：

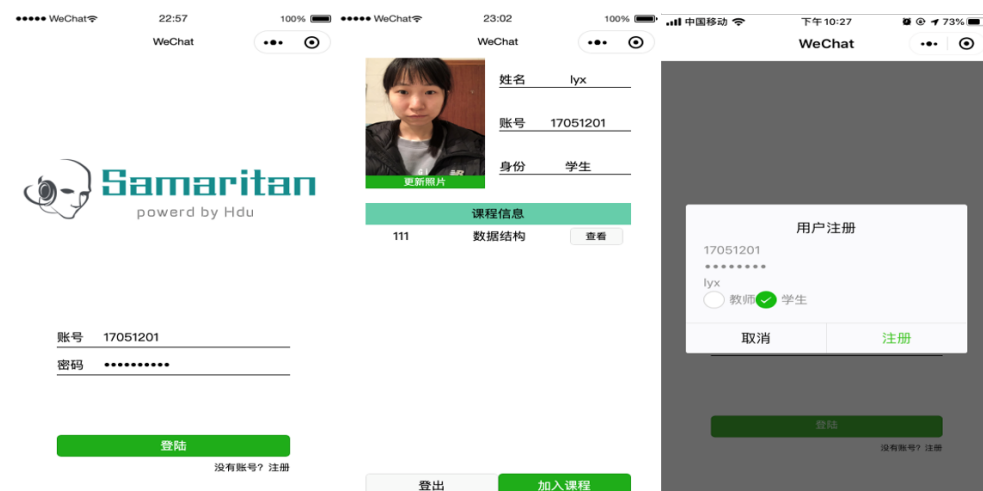


Figure 7: 前端界面

### 3.4 业务端设计

业务后端使用Django框架搭建，作为用户界面与人脸识别算法后端的中间系统，负责接收用户界面回传的参数并进行事件处理，提供用户管理、签到信息创建等服务，实现权限控制，并将处理后的数据传递给人脸识别算法后端，配合算法后端完成签到功能。采用MVVM架构模式，数据模型与业务逻辑代码分离，数据系统则实现两部分工作：数据存储和数据操纵。

存储层面上，共需要存储两部分信息：业务数据信息（用户/课程/签到）以及用户原始图片信息。其中业务数据利用轻量级数据库sqlite存储，图片则通过文件系统组织，并与算法后端约定图片文件命名方式，以实现业务与算法后端之间的图片文件共享。

数据操纵上，采用对象关系映射（ORM）技术，将面向对象程序语言中的对象，自动持久化到sqlite数据库中，并使用事务处理，进一步维护数据存储的完整性、安全性。在数据库设计中，我们将系统抽象为如下E-R图。

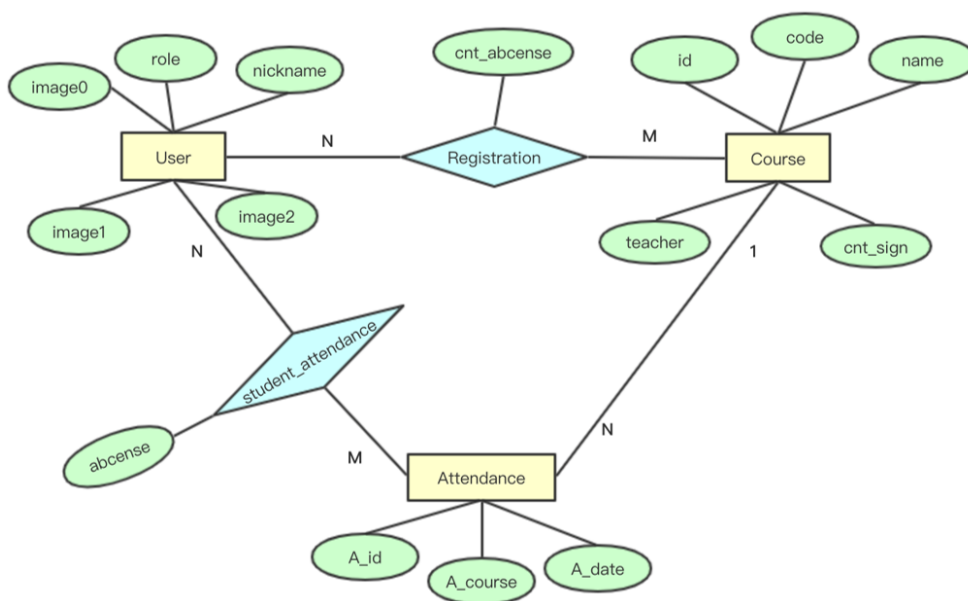


Figure 8: 数据库E-R图

业务端的模块层次图如下。



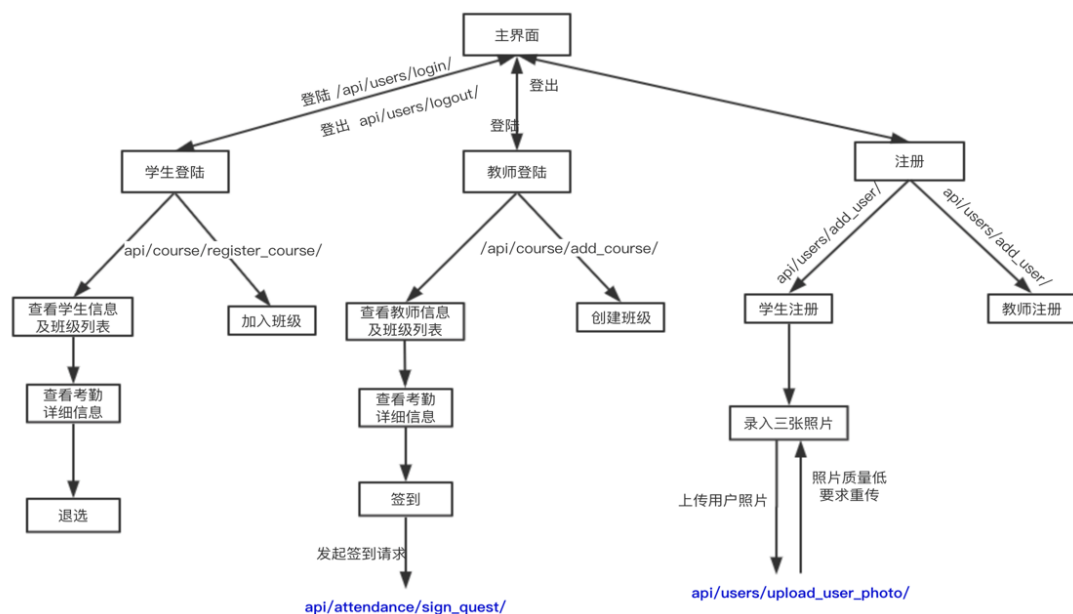


Figure 9: 业务端设计

### 3.5 算法端设计

人脸识别服务系统通过Flask框架实现API服务，该子系统的模块框架见下图。Flask模块负责处理来自业务端的API请求，并根据具体请求内容调用深度学习模型执行相应的推理过程。其中，API实现部分较为简单，此处不再详述，本小节主要介绍本项目使用的人脸识别核心算法。

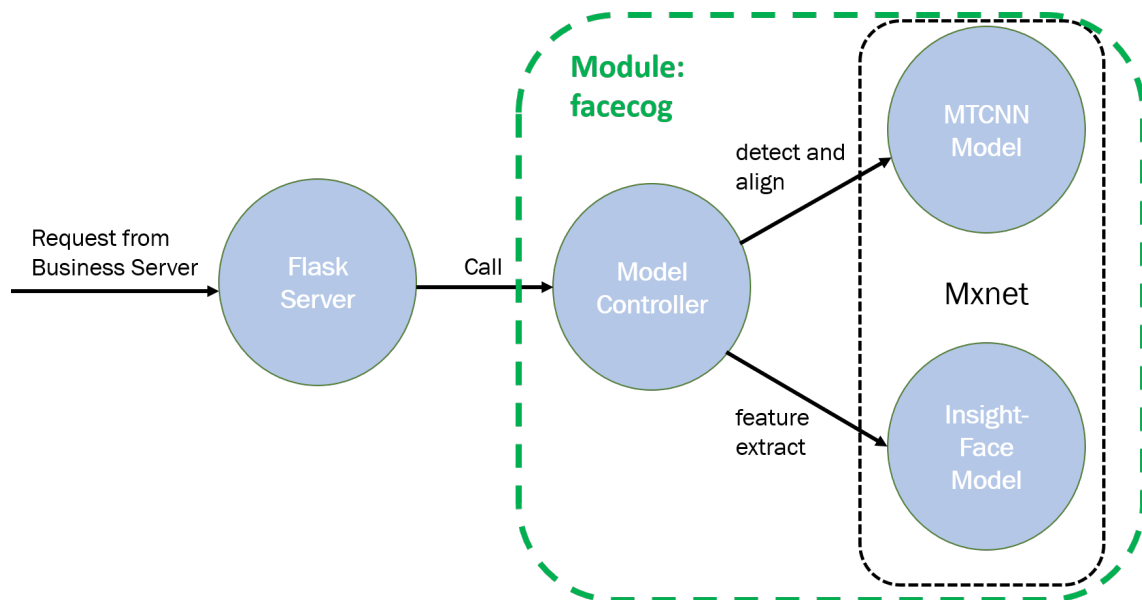


Figure 10: 人脸识别实现逻辑：相似度计算并找出最相近的人脸实际上可以理解为一个最近邻分类问题

当前流行的人脸识别相关技术所采用的通用算法流程是，首先通过人脸检测模型在图片中检测出人脸，然后根据识别框剪裁出人脸图像并做一定程度的对齐操作，然后将对齐的人脸图像输入人脸特征抽取模型得到相应输出特征，最后对得到的特征进行一系列距离计算等操作完成所要求的识别、比对等功能。在实际的实现中，我们往往会先存储不同人的标准人脸照片的特征，当有一张照片需要识别，就执行上述检测和特征抽取的

计算，然后将得到的特征与预先存好的标准特征库中的特征进行比对并判断（逐一计算相似度，或使用最近邻算法）此照片中的人是谁，该实现框架的具体流程可见下图。

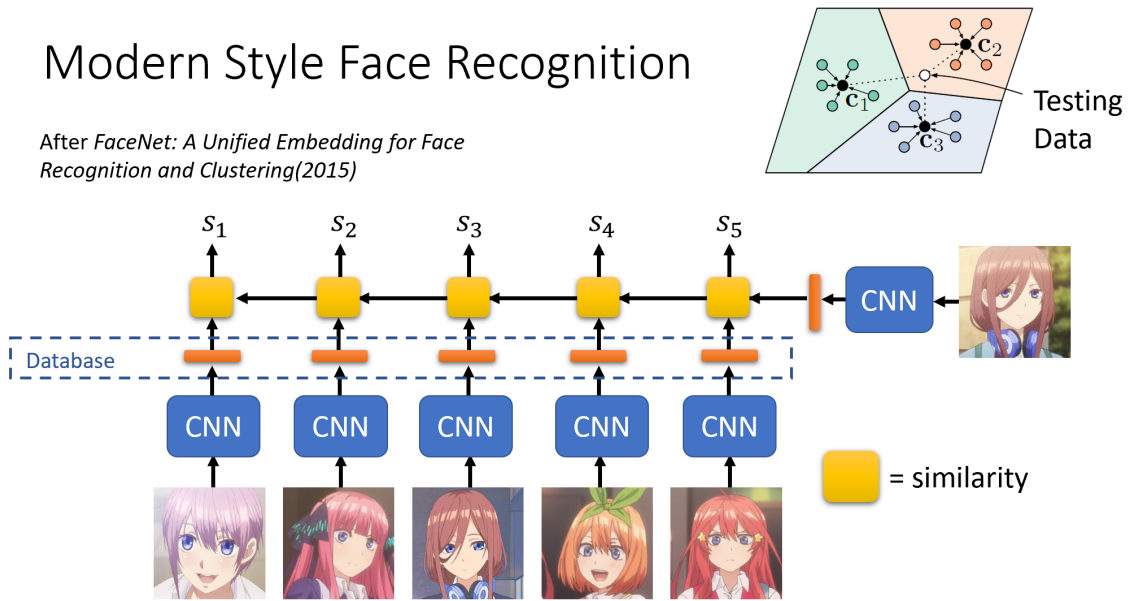


Figure 11: 人脸识别实现逻辑：相似度计算并找出最相近的人脸实际上可以理解为一个最近邻分类问题

本项目中进行的人脸特征抽取操作均采用该流程，其中人脸检测采用Kaipeng Zhang等人所提出的模型 [5]，人脸特征抽取则采用Jiankang Deng等人提出的模型 [2]。本项目所使用的人脸识别模型的架构如图12所示。

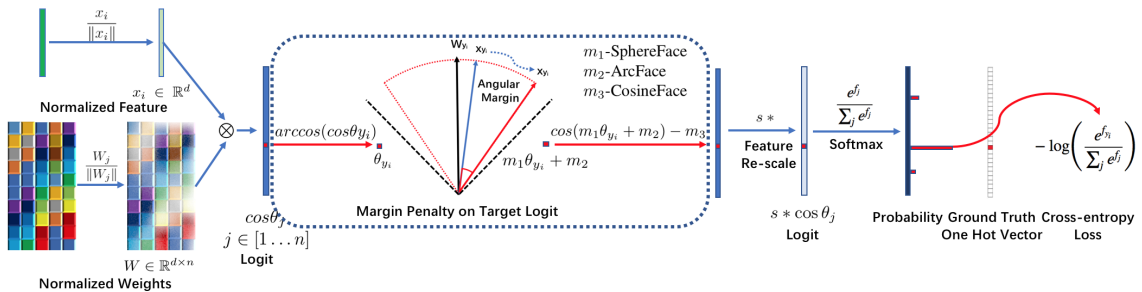


Figure 12: Insightface人脸识别模型

人脸识别系统主要负责提供两种功能，即单人特征抽取和多人人脸识别。

单人特征抽取主要用于得到每位学生的人脸的特征并存入文件系统，作为后续人脸对比的标准特征。为了保证人脸识别系统的健壮性，对于每一位同学我们会需要三张大头照，并得到三个特征。在得到三个特征的过程中，我们需要确认三张图片中是否均存在人脸，且均为同一人，对不符合要求的，人脸识别系统会向业务端反馈错误信息；对于符合要求的特征，存入文件系统作为标准特征。

多人人脸识别主要用于课堂点名场景，人脸识别系统会接收一份名单和一张图片，在图片中寻找名单中的同学，反馈未找到的同学名单；同时人脸识别系统还会在原图上框出检测到的人脸，以不同颜色的框标识识别到的人和未识别出来的人，该图片将存入文件系统供业务端使用。在多人人脸识别的过程中，我们会把从图片中抽取的多个人脸特征与名单中学生的三个标准特征均进行比对，降低识别错误率。

## 4 测试

在实现了上述系统设计之后，我们在本项目的后期进行了单元测试和集成测试，我们在单元测试中尽可能多的发现了潜在Bug并进行了修复，最后成功完成了集成测试。

### 4.1 单元测试

在单元测试中，我们使用了如下技术

- **Mock**即一个模拟架构缺失部分的程序，按照测试者的意愿做出响应，返回被测对象需要得到的信息。由于业务端与算法分离开发，前期无法得到算法端真实的返回数据，所以使用mock的方法模拟算法端返回值进行初步调试。
- **curl命令** curl是一个利用url语法在命令行下进行网络请求/文件传输的命令本项目中，我们用它单独测试各接口的正确性。按照api约定格式发起curl请求，看返回值是否为期望值命令示例：`curl -d '{"sid": "17051201"}' -X GET -H "Content-Type: application/json" localhost:5002/key/`
- **Jupyter-Notebook**基于Jupyter-Notebook良好的交互性和可视化能力，使用该工具对facecog模块进行测试更为高效和直观。

在单元测试阶段我们主要发现了如下问题，并实施了具体的解决方案。

#### 4.1.1 照片方向问题

由于手机相机拍摄的照片带有“转向”信息（Exif），而numpy、Tensorflow、电脑自带图片查看器在处理Exif上存在差异，从而导致识别效果不佳。

**解决方案：**业务端接受图片时统一查看图片的Exif信息，必要时统一把它转为正向；这样一来算法端接受并处理的图片，全部为正向。



Figure 13: 照片方向问题

### 4.2 深度学习接口多次调用问题

Flask自带多进程，当多个请求同时到达系统，Flask会同时多次调用mxnet网络模型，此时mxnet出现了崩溃现象。为了避免崩溃，我们需要将调用序列化，因此实现了一个进程通信队列，Flask端将调用信息封装成包插入队列，使得网络模型同一时刻只会处理一个运算请求，处理完一个才接着从队列中取下一个运算请求的信息，具体理解可见下图。

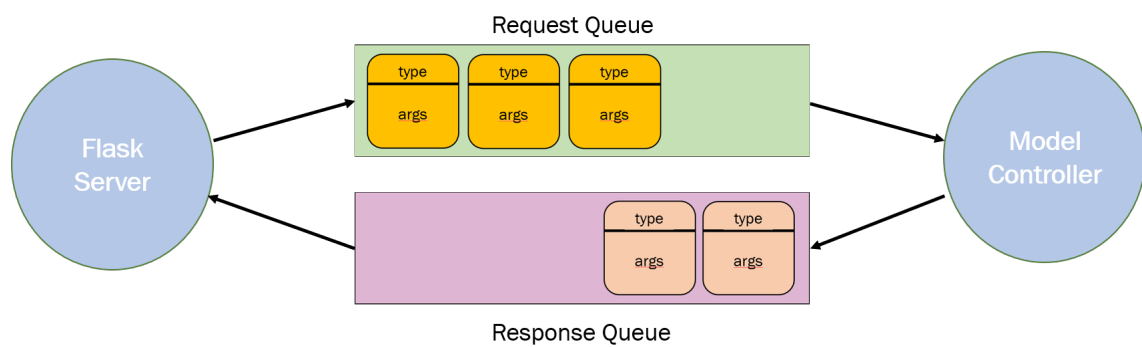


Figure 14: 算法端进程通信队列

### 4.3 集成测试

本小节展示我们的集成测试情况。

**Step1:** 收集班内学生注册信息

<input type="checkbox"/> 16	17201215	dt	2	<a href="#">static/input/17201215_0.jpg</a>	<a href="#">static/input/17201215_1.jpg</a>	<a href="#">static/input/17201215_2.jpg</a>
<input type="checkbox"/> 15	17051201	lyx	2	<a href="#">static/input/17051201_0.jpg</a>	<a href="#">static/input/17051201_1.jpg</a>	<a href="#">static/input/17051201_2.jpg</a>
<input type="checkbox"/> 13	17051136		2	<a href="#">static/input/17051136_0.jpg</a>	<a href="#">static/input/17051136_1.jpg</a>	<a href="#">static/input/17051136_2.jpg</a>
<input type="checkbox"/> 12	17051914		2	<a href="#">static/input/17051914_0.jpg</a>	<a href="#">static/input/17051914_1.jpg</a>	<a href="#">static/input/17051914_2.jpg</a>
<input type="checkbox"/> 11	17051406		2	<a href="#">static/input/17051406_0.jpg</a>	<a href="#">static/input/17051406_1.jpg</a>	<a href="#">static/input/17051406_2.jpg</a>
<input type="checkbox"/> 10	17081815	17081815	2	<a href="#">static/input/17081815_0.jpg</a>	<a href="#">static/input/17081815_1.jpg</a>	<a href="#">static/input/17081815_2.jpg</a>
<input type="checkbox"/> 9	17051329		2	<a href="#">static/input/17051329_0.jpg</a>	<a href="#">static/input/17051329_1.jpg</a>	<a href="#">static/input/17051329_2.jpg</a>
<input type="checkbox"/> 8	17051403	kq	2	<a href="#">static/input/17051403_0.jpg</a>	<a href="#">static/input/17051403_1.jpg</a>	<a href="#">static/input/17051403_2.jpg</a>

Figure 15: 后台已注册名单

**Step2:** 教师注册并创建课程、学生加入相应课程

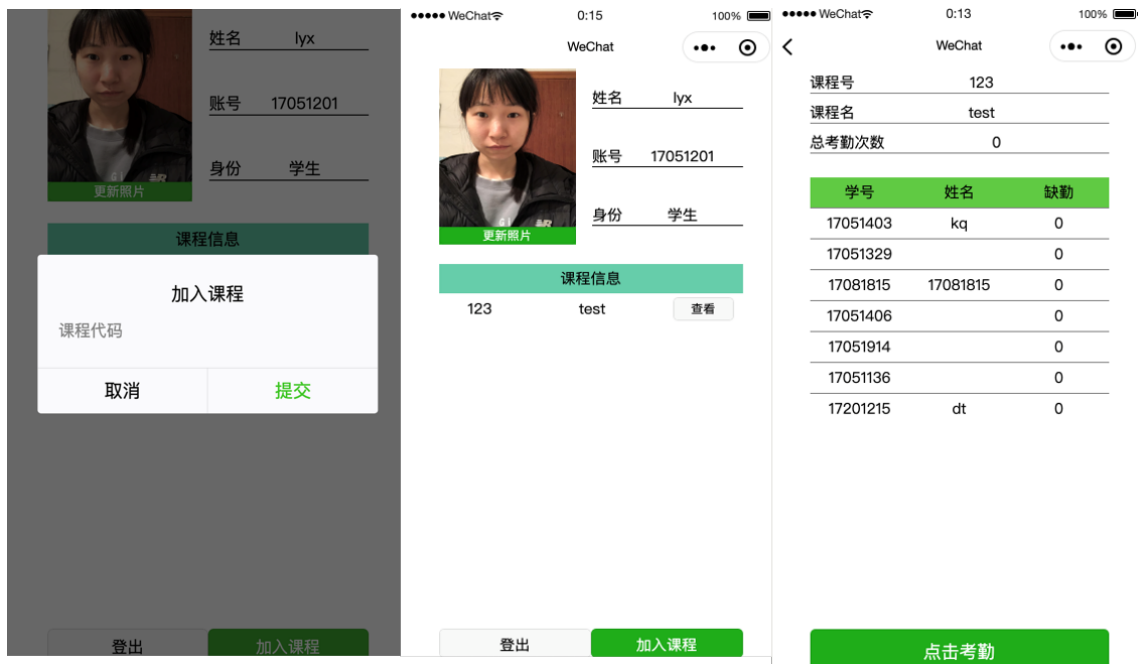


Figure 16: 加入班级

**Step3:** 教师发起签到，并收到初步的签到识别结果（由人脸识别算法给出）

**Step4:** 教师根据实际到课情况，手动修改签到结果，并最终确定签到结果



Figure 17: 点名结果

**Step5:** 相应课程及学生的到课数据写入数据库，数据更新

**Step6:** 验证相应课程与学生数据更新均为正确，表明测试通过



Figure 18: 信息查询

## 5 总结

本项目综合考察了现有点名系统的缺陷，并抓住人脸识别技术的发展成熟的时间点，将人脸识别技术与移动智能平台深度结合，提出了依托智能手机平台、基于人脸识别技术的课堂点名系统。基于理论基础，本项目深度考察了日常课堂场景，对人脸识别点名系统进行了完备的需求分析；架构设计方面，本项目考察了各大前后端平台，综合考虑运行性能与开发的便捷性，最终确定满足签到系统需求的总体架构；人脸识别算法核心设计方面，本项目研究人脸识别历史研究成果，参考了当前流行的人脸识别算法。

本项目最终基本按照预期的进度安排和最初的软件分析设计，进行了成功的实现和测试，完成了预期的任务。在本项目的实施过程中，项目成员受益良多，我们将我们在本项目中的心得体会总结如下：

- 一个完整的软件是十分复杂的，因此软件过程十分重要，做好需求分析、架构设计、接口定义等分析和设计过程，可以为软件的正确、按期实现提供保证，可以有效提高编码效率；应尽量降低软件中各模块的耦合度，可以提高可测试性。
- 计算机网络、操作系统和数据库原理相关知识在软件工程中十分重要
- 要充分利用Github资源；DL相关的bug参考Google比参考百度更好
- 要尽量在单元测试阶段发现bug，在该阶段进行调试与代码修正的时间成本相对较低

## References

- [1] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In CVPR, 2015
- [2] Deng J, Jia G, Zafeiriou S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition[J]. 2018.
- [3] Wu C Y, Manmatha R, Smola A J, et al. Sampling Matters in Deep Embedding Learning[J]. 2017.
- [4] Liu Y , Li H , Wang X . Learning Deep Features via Congenerous Cosine Loss for Person Recognition[J]. 2017.
- [5] Zhang K , Zhang Z , Li Z , et al. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks[J]. IEEE Signal Processing Letters, 2016, 23(10):1499-1503.