目　录

第一章 需求分析

伴随着移动互联网的快速发展，社交媒体成为了人们生活中不可或缺的一部分。但同时又由于互联网的开放性，导致个人的隐私有时无法得到保障。越来越多的用户会在社交媒体上分享自己的生活照片，为了保护自己的隐私，他们会选择在脸部贴上一些可爱的表情。

但是这种手动的贴图方式费时费力，还要自己调整位置，因此，我们想到是否可以利用表情、手势识别技术通过表情识别，自动的将与表情相对应的 emoji 遮盖住人物的面容或手势，不仅保护了隐私，还增添了照片的趣味性。同时添加照片云盘功能，增加平台兼容性，让用户对于人像的处理、保存与分享一站式完成。

面向用户：喜欢图片分享、经常在社交媒体分享人像生活图片并需要保护隐私的人群。

主要功能：根据人像图片对表情、手势一键贴图。提供滤镜、裁剪等传统图片处理工具。将处理好的图片直接分享到社交媒体。Web、安卓双平台的照片云盘功能。

竞品分析	是否根据表情一键贴图	是否有照片云盘	是否能根据手势一键贴图	是否有滤镜	是否移动、网页双平台
Aimoji(本产品)	是	是	是	是	是
美图秀秀	否	否	否	是	是

产品优势：我们的产品在功能上，提供自动化、流水线式的照片处理、分享、保存服务，快捷简便。技术实现上，没有完全依赖于第三方 API，我们团队实现并部署了自建的表情识别 API。可用性上：提供多平台应用，尽可能地扩大用户可用性。产品宣传上：含有部署在互联网上的产品官网，提供下载分发链接，提供功能演示，向用户直观展示产品功能。后期扩展性：基于现有功能的用户，可开发社交功能，提高产品价值。

第二章 概要设计

一、系统整体架构

Web 框架使用 Django 开发，作为 Web 前端和安卓平台的后端，数据库使用 MySql，表情识别和手势识别依赖于百度 API 和我们自己开发的自建 API。

自建 API 基于 Django 开发。整个项目部署在腾讯云上，以便互联网访问。

架构图如下：

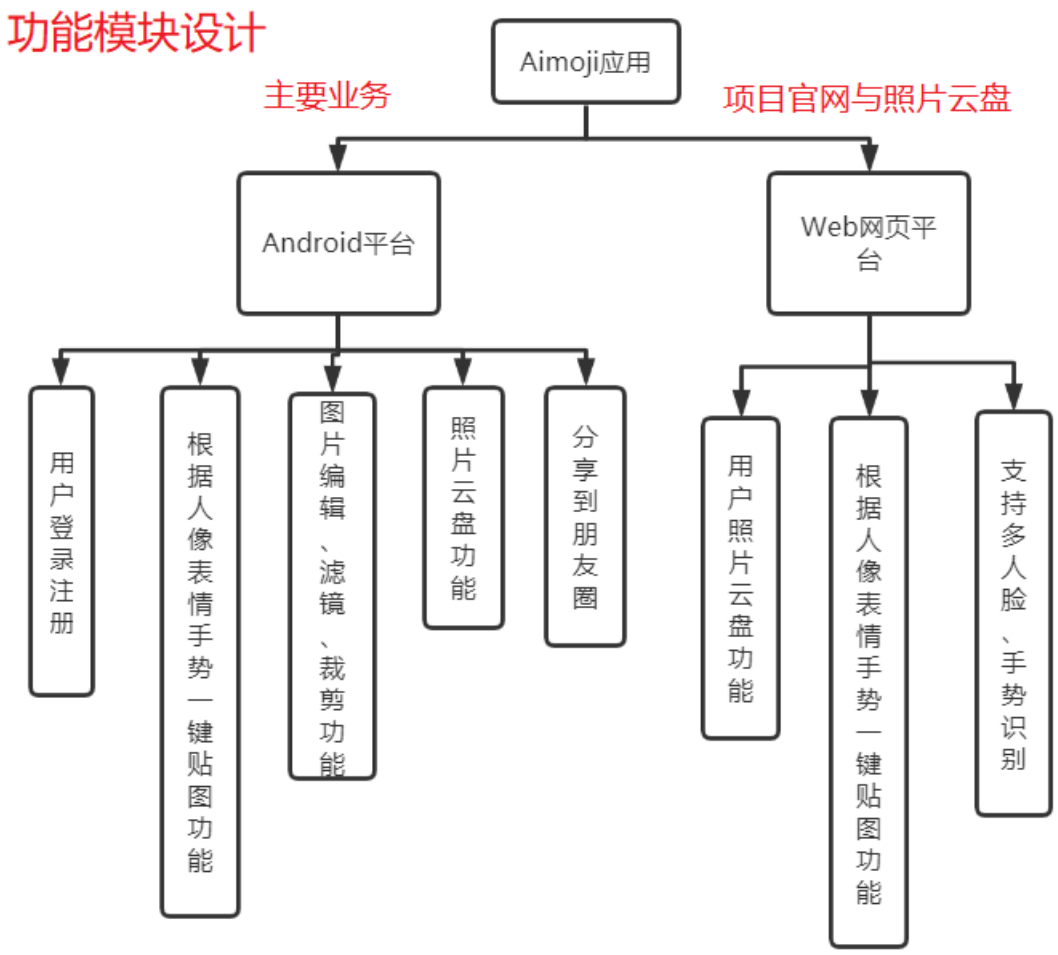
二、功能模块设计

安卓平台：完成此项目的主要业务功能

1. 用户的登录注册
2. emoji 贴图：在人像图片上对表情、手势使用 emoji 继续一键贴图
3. 传统图片编辑：包括截切，滤镜、小贴纸等。
4. 保存功能：保存照片到本地或者照片云盘。
5. 社交媒体分享：将处理过的照片分享到 QQ 空间、朋友圈等社交平台。

Web 平台：作为项目官网，完成功能演示和照片云盘的通用平台

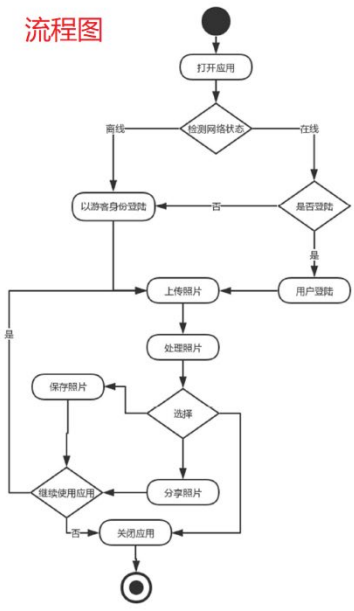
1. 项目展示：展示项目特色，提供下载链接。
2. 功能演示：演示 emoji 贴图效果。
3. 照片云盘：展示同账号下云盘内的所有照片，包含上传功能。



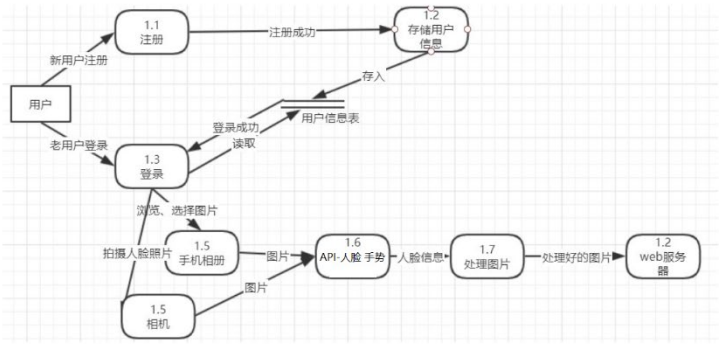
三、程序处理流程与数据流图

安卓端程序使用流程，及前后端处理流程中数据流向如下图。

流程图



数据流图

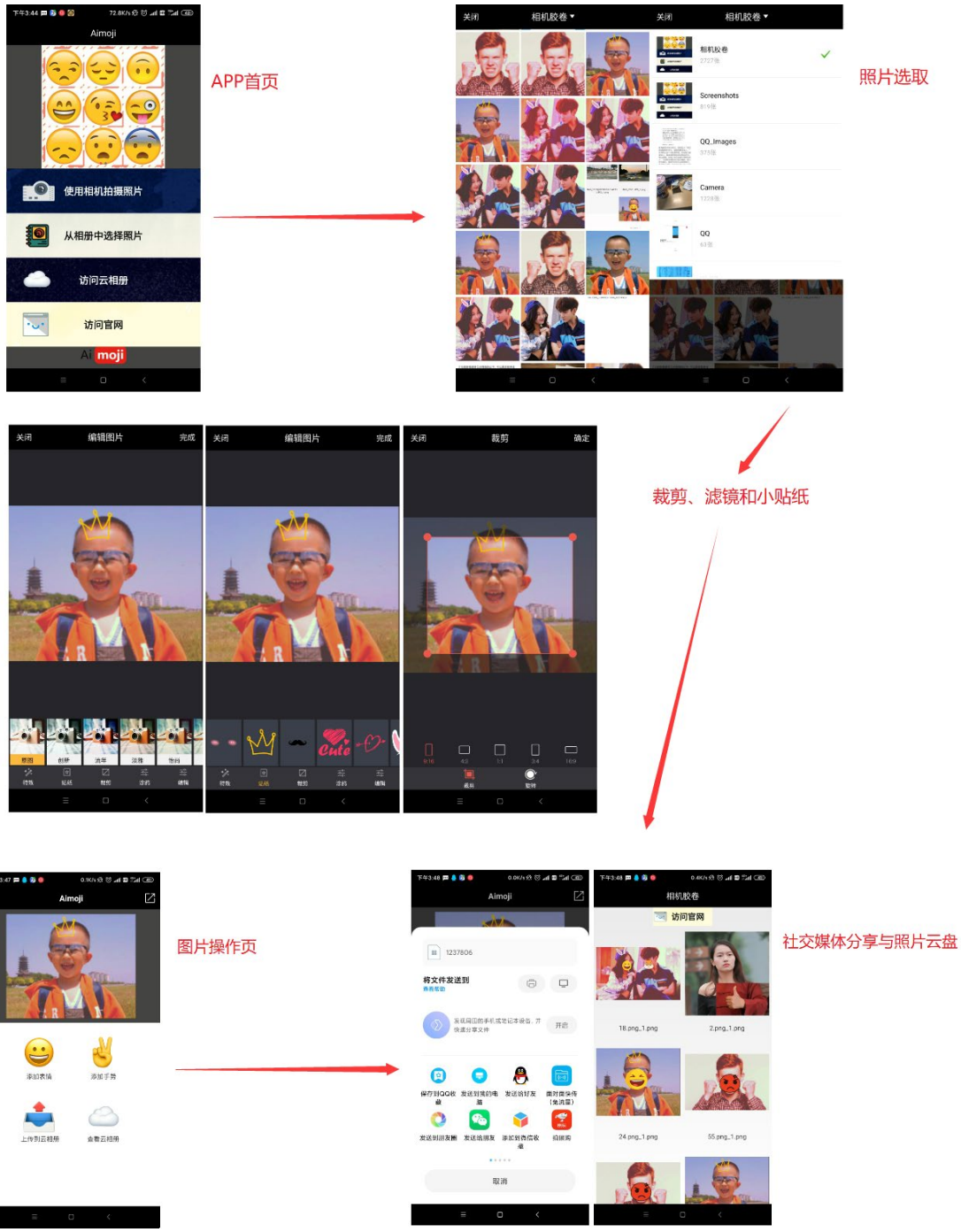


第三章 详细设计

一、界面设计

界面设计要满足需求规约中的一站式、一条龙式的连贯操作，所以在设计时就采用循序渐进，根据输入跳转下一界面的设计方案。即典型流程与界面设计相辅相成。界面会引导用户遵循操作流程。同时字体、图标要大小适当、清晰易识别。

典型流程与 UI 设计如下图：



二、关键算法

由于篇幅所限，本节将主要关注部分关键算法实现。

1) 滤镜实现

在 Android 上使用 ColorMatrix 来处理图片实现自定义的图片滤镜：

在 Android 中图片是以 RGBA 值像素点的形式加载到内存中的，存储形式诶 bitmap，我们可以使用 ColorMatrix 类来修改像素颜色。ColorMatrix 就是 Android 所使用的色彩矩阵。

Android 中的颜色矩阵是一个 4×5 的数字矩阵：

$$A = \begin{bmatrix} a & b & c & d & e \\ f & g & h & i & j \\ k & l & m & n & o \\ p & q & r & s & t \end{bmatrix}$$

这里有一个色彩矩阵分量 C，代表着我们要进行色彩变化的原色彩。

$$A = \begin{bmatrix} a & b & c & d & e \\ f & g & h & i & j \\ k & l & m & n & o \\ p & q & r & s & t \end{bmatrix} \quad C = \begin{bmatrix} R \\ G \\ B \\ A \end{bmatrix}$$

矩阵 R 则代表通过矩阵乘法运算 AC 而得到的新的颜色。

$$R = \begin{bmatrix} R1 \\ G1 \\ B1 \\ A1 \end{bmatrix} = AC$$

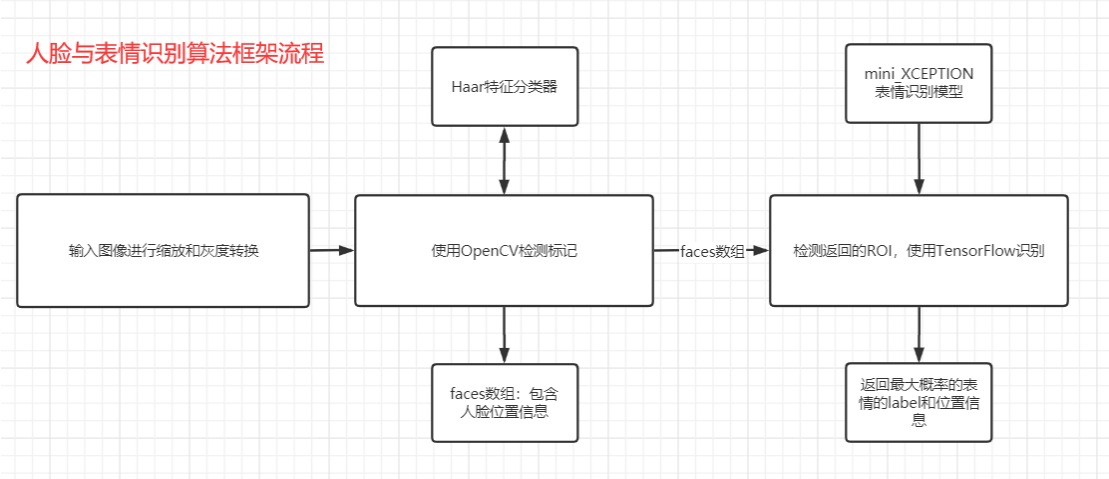
通过矩阵相乘，我们可以得到新的 RGBA 颜色值。

$$R1 = a * R + b * G + c * B + d * A + e$$
$$G1 = f * R + g * G + h * B + i * A + j$$
$$B1 = k * R + l * G + m * B + n * A + o$$
$$A1 = p * R + q * G + r * B + s * A + t$$

颜色矩阵的值可以根据滤镜需要自行设置。第一行的 abcde 值用来决定新的颜色值中的 R 第二行的 fghij 值用来决定新的颜色值中的 G 第三行的 klmno 值用来决定新的颜色值中的 B 第四行的 pqrdt 值用来决定新的颜色值中的 A 其中第五列 ejot 值分别用来决定每个分量重的 offset，即偏移量。

2) 表情识别算法简述

整体算法的执行流程如下图：



人脸检测部分详述（采用 haar 特征）

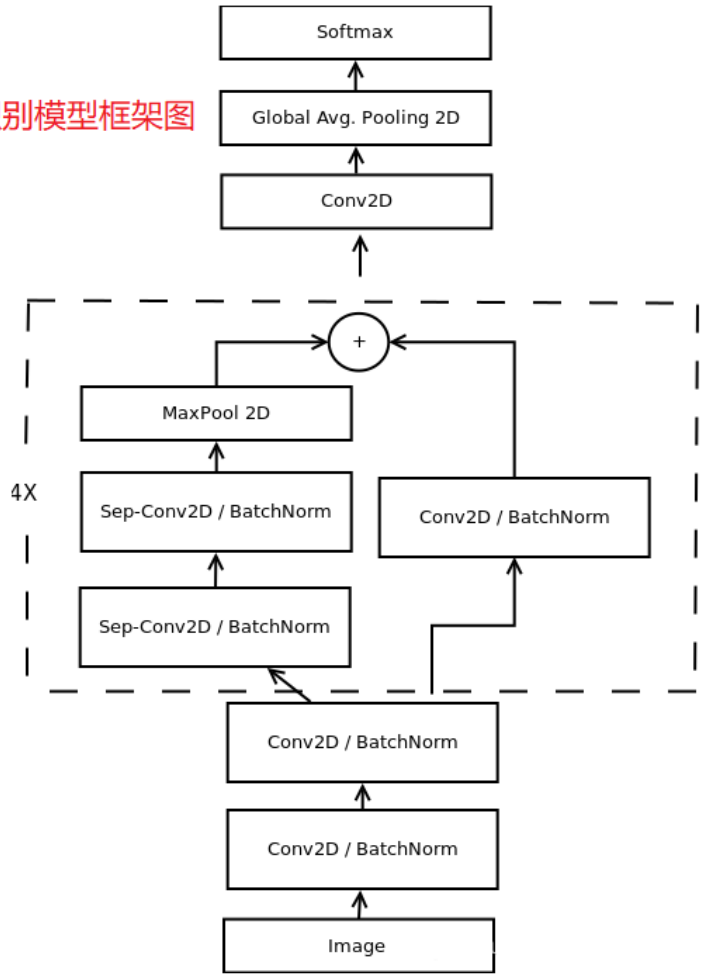
Haar 特征是一种反映图像的灰度变化的，像素分模块求差值的一种特征。主要分为以下四类：边缘特征，线性特征，对角线特征，中心特征；用黑白两种矩形框组合成特征模板，在特征标题模板内用黑色矩形像素和减去白色矩形像素和来表示这个模版的特征值。Haar 特征值反映了图像的灰度变化情况。例如：脸部的一些特征能由矩形特征简单的描述，如：眼睛要比脸颊颜色要深，鼻梁两侧比鼻梁颜色要深，嘴巴比周围颜色要深等。但矩形特征只对一些简单的图形结构，如边缘、线段较敏感，所以只能描述特定走向（水平、垂直、对角）的结构。

这类矩形特征模板由两个或多个全等的黑白矩形相邻组合而成，而矩形特征值是白色矩形的灰度值的和减去黑色矩形的灰度值的和，矩形特征对一些简单的图形结构，如线段、边缘比较敏感。如果把这样的矩形放在一个非人脸区域，那么计算出的特征值应该和人脸特征值不一样，所以这些矩形就是为了把人脸特征量化，以区分人脸和非人脸。这里可以使用 OpenCV 官方的 xml 模型。

表情识别部分详述

这里获取到 cv 检测出的多个 ROI 传递到基于 KERAS 利用 CNN 主流架构之 MINI_XCEPTION 训练性别分类模型 HDF5

表情识别模型框架图



3) 透明图像贴图要点

注意要操作透明图像，首先要保证图像都包括透明通道，即 RGBA 中的 A 通道，所以图片要为 png 格式。

此外还要注意基底图片。在调用 paste 方法的时候设置 mask 参数为 alpha 透明（就是类似 PS 里的蒙版透明），也可以将图片本身作为蒙版。

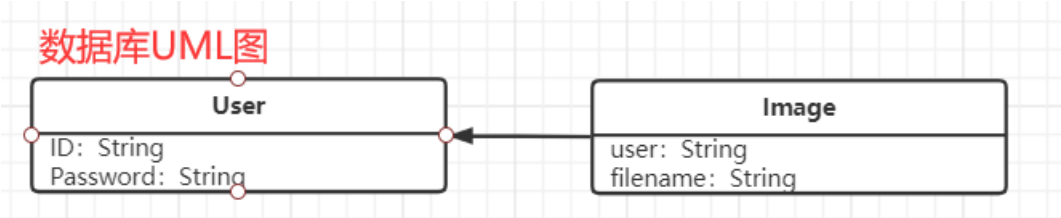
具体代码如下图：

```
face = Image.open(image.name)
face.convert("RGBA")
box = (int(left), int(top), int(left + width), int(top + height))
emoji = Image.open('./images/emoji_happy.png')
emoji.convert('RGBA')
emoji = emoji.resize((box[2] - box[0], box[3] - box[1]))
face.paste(emoji, box, emoji)
face.save('./result.png')
```

三、数据库设计

数据库由于数据不是很复杂，只建立了两个 Table。ID 表存储用户信息，Image 表存储用户的照片云盘中的图片信息。

其中，ID 为 User 表的主键，Image 中的 User 关联 User 表的 ID 作为外键。



第四章 主要测试

由于篇幅所限，本章节将提炼精简测试报告，主要展示项目的关键测试项目，记录测试过程，改进、修正过程和测试结果。

A.

测试功能	Emoji 贴图		
前提条件	获取到包含人脸的 png 图像		
测试过程			
序号	步骤	期望结果	实际结果
1	上传包含各种表情的人像	上传正常	上传正常
2	收取贴图后的图像	显示对应图片	部分 emoji 部分背景为黑色
错误分析			
序号	错误描述	错误分析	
1	部分表情的 emoji 贴图后，背景为黑色，而非透明	检查发现出问题的 emoji 资源文件的格式为 webp 格式，webp 格式不像 png 格式包含透明通道	
错误解决			
序号	解决方案	再次测试结果	
1	使用 PS 导入 webg 文件，添加透明图层，最终另存为 png 格式	贴图正常	

B.

测试功能	照片云盘上传

前提条件	已经登录用户的账号，服务器运行正常		
测试过程			
序号	步骤	期望结果	实际结果
1	在安卓平台选择图片	选取正常	选取正常
2	点击上传到照片云盘	在 Web 端的照片云盘看到上传的图片	上传失败，服务器端报 HTTP500 错误
错误分析			
序号	错误描述	错误分析	
1	安卓端在发送 POST 请求使用 form 上传文件时，服务器无法正常接收。	检查安卓代码发现，未对上传的 form 进行编码	
错误解决			
序号	解决方案	再次测试结果	
1	对 form 进行 multipart/form-data 的形式进行编码	上传正常，web 端可以查看	

C.

测试功能	服务器部署性能测试		
前提条件	服务器运行正常，项目部署完成		
测试过程			
序号	步骤	期望结果	实际结果
1	在浏览器输入 http://212.64.48.72/	浏览器正常显示页面	浏览器正常显示页面
2	点击功能演示，并上传样例人像图片	正常显示处理后照片和分析结果	正常显示处理后照片和分析结果
3	在安卓平台重复第二部操作	正常显示处理后照片和分析结果	正常显示处理后照片和分析结果
可用性分析			
序号	预期结果	实际结果	
1	保证主流平台，和安卓 APP 在各种网络环境下均可访问 web 服务	在 Windows、Mac、Android、iOS 等平台的主流浏览器在接通互联网下都可访问 web 服务	
性能分析			
序号	预期结果	实际结果	

1	静态界面可在 0.5 秒内完成加载	使用 Chrome 的 Network 选项卡监控重复加载测试，在 1M 带宽下能够满足 0.5s 左右完成加载显示
2	功能演示应在 1s 内完成上传下载操作	上传 500kb 左右的样例演示图片，上传基本能够在 0.5s 左右，从上传到演示结果全部加载完成需要 1.5s 左右。限制原因：服务器上传带宽有限，为 1M。

第五章 安装及使用

一、安装环境

若想自行部署 Web 项目，需要如下环境：（Linux 和 Windows 均可）

推荐：Windows 10、Ubuntu 18.04 LTS

所需组件	推荐版本
MySQL	8 及以上版本
Python	3.8
Django	3.0.7
PyMySQL	0.9.3
Keras	2.3.1
Pillow	7.1.2
imutils	0.5.3
matplotlib	3.2.1
numpy	1.18.5
opencv-python	4.2.0.34
pandas	1.0.4
scipy	1.4.1
scikit-learn	0.23.1
tensorflow	2.2.0

在控制台输入如下命令即可运行 Web 服务

python3 manage.py makemigrations python3 manege.py migrate	将数据库要执行的内容记录到文件并操作数据库
python3 manage.py runserver 0.0.0.0:80	运行 Django

二、安卓平台安装流程

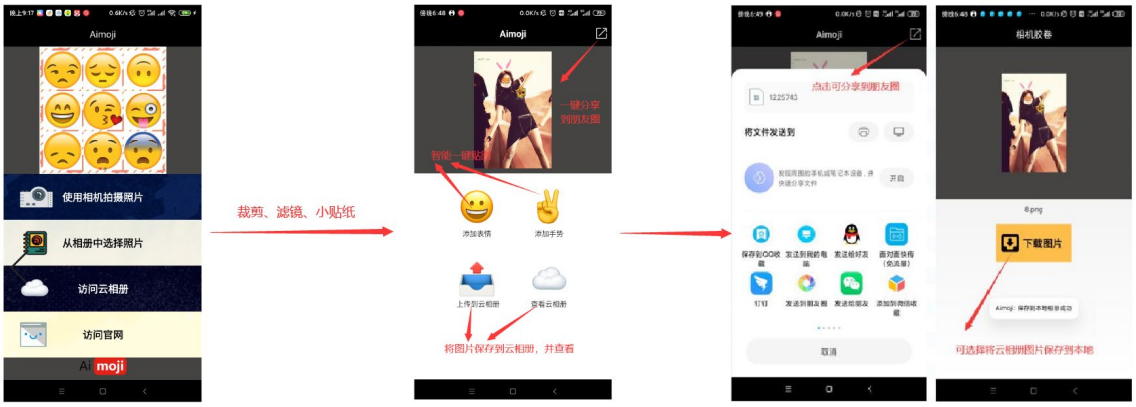
在浏览器输入 212.64.48.72 访问官网，单机“点击下载 apk”按钮即可获取安装包。

在安卓系统上，打开下载的 apk 文件，之后根据手机提示即可安装。

（限 Android5.0 以上版本）

三、安卓平台典型使用流程

前提条件：已登录到账号或使用游客模式。



可以拍摄新照片或者从相册选取照片　　根据需要选择对应操作　　对于处理后的图片进行分享或者保存

第六章 项目总结

本次项目开发推进过程中有许多收获与感悟，具体如下：

一、技术提升

本次项目设计的技术点有：安卓开发、web 开发、机器学习算法应用，更详细一点说，还涉及到前后端分离、对接、数据库等层面。因此，通过此次项目开发，我们对于

这些技术点的认知以及熟悉程度都得到了提升。与此同时由于该项目是一个工作量比较大的项目，我们采用 GitHub 作为代码托管平台进行协同开发和版本管理，在此过程中，我们的工程能力、协同与团队能力也得到了提升。

二、项目开发与管理

由于本项目从一开始就是按照软件开发流程进行推进的，即软件需求分析、软件设计、开发、测试流程。在各个环节都有进行适当的工作量与工作计划的分配与调整。所以我们在整个过程中，并没有出现因为需求不合理、设计不合理等因素导致的项目延迟、返工等问题。同时合理得当的分工让我们的开发流程平稳推进。使用 git 让项目整合十分轻松，版本管理十分清晰。这一点对于我们来说是非常宝贵的开发经验，可以沿用到日后的项目迭代以及其他项目开发过程中。

具体分工如下：尹航负责 Web 前端与后端，表情识别 API 的开发。苏昌盛与王艺瑾负责安卓开发和资源收集工作。

三、解决问题的能力

在项目开发过程中，我们遇到了许多的技术问题，有的问题有先例可以参考，有的问题则没有，我们通过查看技术文档，借助搜索引擎，反复研读相关文献，进行解决方案的分析、试错最终基本解决了出现的技术问题，因此对于独立解决问题的能力，阅读文献的能力，语言描述能力都有很大的提升。

四、未来规划

本项目已经实现了最核心的功能，即登录注册、图片编辑、图片处理等功能，与此同时仍然有许多想要升级的地方。首先是 UI 的进一步设计。对于一个应用来说界面易用度是一大重要因素。美化会从 UI 的美观程度、跳转流畅度着手。此外，虽然已经有了安卓端，但是对于 ios 用户并不友好，而且安卓 APP 有些许臃肿。所以在未来我们打算开发微信小程序作为另一个平台，让 APP 能够在更多设备上更好地运行。算法的提升会从目标检测的速度、精确度以及多次运行后的内存、显存等资源回收入手，提高持续使用性能和效率。

此外想基于照片云盘的算法增加一个面向所有用户开放的照片社区，增加社交 APP 的社交属性，作为提高用户使用时间的一个黏性功能。