

4-16 运行人脸识别程序

140.143.132.225:8000/project-1/doc-64

一、创建Docker容器

上节课我们在Docker中安装了人脸识别镜像，因为人脸识别程序是用Python写的，而且需要很多依赖库，安装起来非常麻烦，所以我就把依赖环境和人脸识别程序封装成Docker镜像，只要你在本地Docker上面导入镜像，创建出容器，就能运行Python人脸程序了。

把 `demo.tar` 文件上传到Linux根目录，然后解压缩

```
1. tar -xvf demo.tar
```

解压缩之后，`demo`文件夹中就包含了人脸识别Python程序，我们只需要把`demo`文件夹挂载到Docker容器，那么在容器中就能访问Linux主机的`demo`文件夹了。下面开始创建容器，映射端口号，挂载目录。

1. #创建容器，把容器3000端口映射到宿主机3000端口，把/demo映射到宿主机的/demo
2. `docker run -d -it -p 3000:3000 -v /demo:/demo --name node face`
3. #查看容器运行状态
4. `docker ps -a`
5. #进入到node容器
6. `docker exec -it node bash`

二、运行人脸识别程序

进入到node容器之后，然后进入 `/demo` 目录，运行人脸识别程序

1. `cd /demo`
2. #把Python程序挂起到后台运行
3. `nohup python3 -c "from app import app;" > log.out 2>&1 &`

三、接口调用

人脸识别程序结合了Flask框架，提供Web接口，具体如下

1. 创建人脸模型数据

当Emos系统的MySQL数据库中不存在签到员工的人脸模型数据，这时候应该调用人脸识别程序的Web接口，上传照片文件，然后由Python程序识别照片中的人脸，返回人脸模型数据。Java系统接收到人脸模型数据之后，把数据保存在MySQL数据表里面。

接口名称: `/create_face_model`

请求类型: POST

传入参数: `icode`

返回结果：人脸模型数据

2. 执行人脸签到识别

接口名称：/checkin

请求类型：POST

传入参数：icode

返回结果：人脸识别结果

