

中国移动云市场 FaceSaaS 产品 操作手册

FaceSaaS 平台项目组

2018/8/29

目 录

1.	修订目录	3
2.	范围	3
3.	应用介绍	3
4.	相关术语与缩略语解释	4
5.	软硬件要求	4
6.	公有云模式	4
7.	操作流程	5
7.1.	材料获取和整理	5
7.2.	生成校签	5
7.3.	配置 HTTP 请求	5
7.3.1.	配置 HTTP Header	6
7.3.2.	配置 HTTP Body	6
7.3.3.	提审与上线	7
8.	产品的主要功能概述	7
9.	功能接口列表	8
9.1.	后端活体检测	8
9.1.1.	应用场景	8
9.1.2.	接口列表	8
9.2.	人脸定位	错误!未定义书签。
9.2.1.	应用场景	错误!未定义书签。
9.2.2.	接口列表	错误!未定义书签。
9.3.	1:1 比对	12
9.3.1.	应用场景	12
9.3.2.	接口列表	12
9.4.	1:N 比对	21
9.4.1.	应用场景	21
9.4.2.	接口列表	21
9.5.	OCR 识别	26
9.5.1.	应用场景	26
9.5.2.	接口列表	26
9.6.	身份证真伪评估	错误!未定义书签。
9.6.1.	应用场景	错误!未定义书签。
9.6.2.	接口列表	错误!未定义书签。

1.修订目录

日期	修订者	版本号	说明
2018.06.06	周杏芳	1.7.14	

2.范围

本文档是 FaceSaaS 人脸识别系统产品在中国移动公众服务云 SAAS 平台操作手册。

3.应用介绍

FaceSaaS 人脸识别系统能够实现基于静态人像图片的 1:1 比对和 1:N 建库检索。同时支持基于 H5 动态视频流的活体检测。针对人像图片，依图算法能够对室外背光等复杂环境，刘海化妆等复杂对象，以及镜面旋转等复杂条件识别处理，基于不同的照片类别定义比对模

型，保证比对效率和性能，达到万分之一误报率下超过 95%的通过率。同时，人像平台提供了千万量级的建库容量，支持跨库检索和多库管理，有效满足黑名单和 VIP 识别等应用场景的需求。

4. 相关术语与缩略语解释

◆ SaaS: Software as a Service (软件即服务)

5. 软硬件要求

全私有或者半私有云部署，服务器的软硬件基本配置要求如表 5.1 所示。

表 5.1 软硬件要求

类别	要求说明
CPU	Intel 2011 年后的产品，CPU flags 包含 sse4_1/sse4_2，建议 8 核及以上
内存	至少 32G
硬盘	Disks 目录在 100G 以上
操作系统	1. 依图操作系统 YituOS 2. Redhat 6.2/6.4/6.5/6.6/6.7/7.2，CentOS 6.5/6.6/7.0/7.1/7.2

6. 公有云模式

公有云模式即指上传照片至依图 SaaS 服务器，从而完成抽特征值及比对过程。通过 API 接口调用依图公有云的服务，可实现上文提到的所有功能。

面向国内客户，依图公有云服务部署在两个集群供客户使用，拥有不同的功能和服务质量。其中，预演 staging 集群模拟生产环境，可用于对接调试，同时也可进行模拟生产调用使用方式的性能和压力测试，会根据该集群上的调用进行上线检查；生产 production 环境是正式发布上线且按调用计费的集群。

为保证公有云的正常运行，集群更换至少需要提前一天准备，客户需要至少一天申请集群变更。

表 6.1 公有云集群分类简介

集群名称	域名	功能用途	环境特点
预演集群	staging.yitutech.com	1. 面向测试用户接口联调 2. 吞吐率、压力及稳定测试	1. 需要用户身份验证 2. 可连接第三方数据源获取网纹照 3. 稳定可用

生产集群	api-xx.yitutech.com	面向上线客户，提供稳定持续可用的高质量服务（按照预估峰值的 1.5 倍）	1. 需用户身份验证 2. 对生产集群的访问会进行计费 3. 稳定可用
------	---------------------	--------------------------------------	---

公有云的部署方式具有最少初期部署、日常维护成本运营成本最低等优点，然而对于某些特定行业需要注意监管问题。

7. 操作流程

7.1. 材料获取和整理

部署需要的前期材料将由依图提供，具体包括：

- （1） 测试集群地址；
- （2） 客户标识（AccessID）和 AccessKey；
- （3） 生成校签（signature）的样例代码压缩包 Yitu_Client_Sample_code.tar，代码为 Java 格式；
- （4） FaceSaaS 产品使用说明书和接口文档。

7.2. 生成校签

根据样例代码 Yitu_Client_Sample_code，完成 staging 测试集群校签，具体步骤如下所示：

- （1） 解压压缩包 Yitu_Client_Sample_Code.tar，得到工程目录，具体目录说明见文件夹中的 README 文件；
- （2） 导入 eclipse 工程，要求 JRE 为 JavaSE-1.6 版本。具体过程为打开 eclipse->File->Import->General->Existing Projects into Workspace->选入工程目录导入；
- （3） 按照依图提供的相关参数修改 config.json 中的对应配置，即修改 AccessID 和 AccessKey；
- （4） 打开 SampleCodeV1PairVerify.java 并运行程序；
- （5） 在 Console 查看执行结果，若比对通过，将会返回 check ok，校签已生成。

7.3. 配置 HTTP 请求

客户端需要基于 HTTP 协议，在网络环境下发送请求给依图服务端，依图服务端解析请求，并调用后台计算资源完成请求所赋予的计算任务，生成需要返回给服务端的响应，客户端接受响应。HTTP 请求分为 Header 和 Body 两部分，需要分别进行配置，具体步骤如下所示：

7.3.1. 配置 HTTP Header

HTTP Header 中包含的参数格式如表 7.1 所示：

表 7.1 HTTP Header 格式

参数名	类型	说明	必填	限制
x-access-id	String	客户标识（AccessID）：表示不同的客户	是	最大 16B
x-signature	String	客户签名（\$signature）： 签名是对请求内容的校验，用于依图服务器确认调用者身份的合法性和请求的合法性，签名之后数据的大小一定为 256 个字节。	是	256 个字节

签名（\$signature）函数可以使用开源的 RSA 算法库，最终的签名数据可由下式得到：

$\$signature = \text{Hex}(\text{RSA_puc_encrypt}(\$content, \$YITUPubKey, \text{RSA_PKCS1_PADDING}))$

通过 RSA 算法签名后数据为固定的 128 字节，然后再通过十六进制编码的方式编码成 256 个字节的内容。其中签名的内容（\$content）不超过 117 个字节，各部分按照顺序如表 7.3.2 所示：

表 7.2 签名（\$signature）内容格式

名称	大小	说明
AccessKey	32 个字节	已提供
MD5（HTTP Body）	32 个字节	MD5 的 http body 应该是 16 个字节，然后再用 ToHex 转换成 16 进制表示。
Unix 时间戳	4 个字节	从 1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的秒数，不考虑闰秒。
随机数	8 个字节	无
客户自定义内容	不超过 41 个字节	可以任意填写客户希望加上的内容，帮助客户确认此请求确实是该客户所发。

综上，HTTP Header 示例输入为：

```
Post /register_user_info HTTP/1.1
Host: http://staging.yitutech.com
Content-Length: $CONTENT_LENGTH
Content-Type: application/json
x-access-id: access_id
x-signature:$signature
{
    HTTP Body
}
```

7.3.2. 配置 HTTP Body

HTTP Body 中请求的内容自动为 JSON 格式的字符串，不支持 gz 压缩。用标准的 base64 编码，编码内容中不包含“/n”。客户可根据功能接口列表在 HTTP Body 中配置所需功能，此处以手机 SDK 验证接口为例展示 HTTP Body 配置功能流程，其他功能与之类似。

（1） 查看接口列表，确定功能输入字段格式与内容如表 7.3 所示：

表 7.3 手机 SDK 验证接口输入字段

类别	参数表	说明
大礼包	Query_image_package	依图活检 SDK 加密编码的大礼包
返回值设定	Query_image_package_return_image_list	是否返回大礼包中解出图片列表
大礼包验证	Query_image_package_check_same_person	检查大礼包图片是否为同一人

(2) 在 HTTP Body 中输入所需字段:

POST /face/basic/check_image_package HTTP/1.1

Host: http://staging.yitutech.com

Content-Length:\$CONTENT_LENGTH

Content-Type: application/json

x-access-id: access_id

x-signature: \$signature

{

“query_image_package”:“活体检测 SDK 加密编码内容”,

“query_image_package_return_image_list”:true,

“query_image_package_check_same_person”:true

}

(3) 查看接口文档, 确定输出字段格式与内容如表 7.4 所示:

表 7.4 手机 SDK 验证接口输出字段

类别	参数名	说明
大礼包解析	Is_valid_package	是否为有效的加密大礼包(验证来自依图 SDK)
	Query_image_contents	大礼包解码图片列表, 如有正脸则为第一张
	Is_same_person	大礼包查询照片是否为同一个人

7.3.3. 提审与上线

上线请提前 24 小时在对接群中申请, 审核通过后, 依图将分配生成集群 URL。

8. 产品的主要功能概述

本产品对外提供 SaaS 服务接口作为解决方案, 数据传输基于 HTTP 协议, 采用 JSON 作为标准数据格式。客户可以根据自身实际情况, 通过互联网订购所需的应用软件服务。产品提供身份证真伪评估、证件 OCR 识别、照片 1:1 比对、1:N 比对、活体检测、人脸定位等多种功能。

9. 功能接口列表

9.1. 后端活体检测

9.1.1. 应用场景

活体检测通常应用于无人值守的远程用户身份认证或有人值守但值守人不可信的场景，完成活体和同人验证。活体检测捕获的用户图片一般配合其他身份信息实现比对服务，比如互联网金融的远程用户登录和用户认证及各种业务操作等。

依图活体检测采用前端和后端相结合的模式，考虑到前端设备性能有限，以及 APP 大小限制，前端只完成动作识别和检测，生成包含动作照和防屏幕翻拍照的加密大礼包，或在前端 H5 的场景下，上传包含固定动作序列的检测视频，由后端依图公有云实现活体和同人认证，基于防攻击算法保证活检的通过率和安全性能。

9.1.2. 接口列表

编号	名称	URL
101.3	手机 SDK 捕获图片数据包验证接口	/face/basic/check_image_package

（一）101.3-手机 SDK 捕获图片数据包验证接口

（1）术语定义

手机 SDK 捕获照片	依图手机 SDK 捕获的活体人脸照片，可能由 1 张或者多张图片组成。
手机 SDK 捕获数据包 （别名大礼包）	根据手机 SDK 捕获的活体人脸照片，加上加密和签名验证生成的字符串，包含了照片信息和对应的验证签名，经过大礼包解密成功的数据能够保证来自 yitu 手机 SDK。

（2）交互机制

服务端启动，开始接收手机 SDK 捕获数据包，并进行签名验证。

（3）接口功能

输入大礼包字符串，接口返回解码的内容，包含图片列表， 用户自定义信息，以及解码是否成功。

（4）传输方式

请求服务端	POST
公有云地址	http://staging.yitutech.com/face/basic/check_image_package

客户端使用 HTTP 请求服务端，请求内容符合（5）（6）中定义。

（5）接口输入

HTTP header:

参数名	参数类型	说明	必填	默认值	限制
x-access-id	string	请求来源	是		最大 16B
x-signature	string		是		恰好 32B

HTTP Body:

参数作用	参数名	参数类型	说明
大礼包	query_image_package	string	查询人脸的大礼包 必填 限制：来自 yitu 活体 SDK 捕获的数据包需要 Base64 编码之后传入
返回 图片相关	query_image_package _return_image_list	bool	是否返回大礼包中解出的图片列表 默认值： true
返回 特征相关	query_image_package _return_feature_string	bool	是否返回正脸特征图片 默认值： false
	query_image_package _feature_number	int	返回 feature string 中 feature 的数量 默认值： 1 应小于等于大礼包中图片的数量
大礼包 验证相关	query_image_package _check_same_person	bool	是否检查活体比对数据包中 查询照片是同一个人 默认值： false
	query_image_package_check _anti_screen	bool	是否开启防屏幕翻拍检测 默认值： false
	query_image_package_check _anti_screen_threshold	string	防屏幕翻拍检测阈值 默认值： "high" 可选字符串参数值： "high" 表示较为严格的检查标准， 通过概率约 90%，误报率约 4% "low" 表示一般程度的检查标准， 通过概率约 95%，误报率约 10%
	query_image_package _check_dark_illumination	bool	是否开启光线昏暗检测 默认值： false 判断光线不足会返回错误
	query_image_package_check _dark_illumination_threshold	string	光线昏暗检测阈值 默认值： "medium" 表示允许的昏暗程度，可选字符串 参数值： "high" 场景非常昏暗，肉眼难以分辨 人脸细节 "medium" 场景比对昏暗，肉眼勉强 可以分辨人脸细节

			"low" 场景相对昏暗, 但肉眼可以轻松分辨人脸细节 示例图请参考[附录 1]
	query_image_package_check_anti_picture	bool	是否开启防照片检测 默认值: false
	query_image_package_check_anti_picture_threshold	string	防照片检测的阈值。 默认值: "medium"
	query_image_package_check_anti_eye_blockage	bool	是否开启眼部遮挡检测, 默认值: false
	query_image_package_check_anti_eye_blockage_threshold	string	眼部遮挡检测的阈值。 默认值: "medium"
	query_image_package_check_anti_hole	bool	是否开启防孔洞检测, 默认值: false
	query_image_package_check_anti_hole_threshold	string	防孔洞检测的阈值。 默认值: "medium"

样例输入:

POST /face/basic/check_image_package HTTP/1.1

Host: http://www.yitu-test.com

Content-Length: \$CONTENT_LENGTH

Content-Type: application/json

x-access-id: access_id

x-signature: e4305df7e61b38adbd59eabef7f8b706

```
{
  "query_image_package": " Base64 编码之后的活体 SDK 捕获的编码的数据包内容",
  "query_image_package_return_image_list": false
}
```

(6) 接口输出

参数名 (多级)		参数类型	说明	样例值
rtn	-	int	请求处理结果的错误代码 0 表示 OK, <0 表示服务器发生错误。	0
message	-	string	请求处理结果的错误信息	OK
query_image_package_result	is_valid_package	bool	是否是有效的大礼包	true
	customer_defined_content	string	用户在大礼包中预先加入的自定义字段	"user defined content"
	query_image_contents	array	大礼包中解码的图片列表, 若有正脸图片. 则正脸图片为第一张.	["IMAGE_1 BASE64", "IMAGE_2 BASE64", "IMAGE_3_BASE64"]
	is_same_person	bool	大礼包照片是否同一	true, 仅在开

			个人	启同人验证下返回
	feature_string	string	特征字符串,可能包含多个特征,默认返回第一张图片的特征	"特征编码的字符串",仅在开启返回特征下返回
	is_anti_screen_check_valid	bool	防屏幕翻拍验证结果是否有效,若无效则以下两个字段结果无效	true
	is_anti_screen_check_passed	bool	防屏幕翻拍的验证结果,是否通过了测试仅在 query_image_package_check_anti_screen 开启时返回	true
	is_anti_screen_check_score	double	防屏幕翻拍的验证分数仅在 query_image_package_check_anti_screen 开启时返回	0
	is_dark_illumination_check_valid	bool	光线状况判断结果是否有效,若无效则以下两个字段结果无效	true
	is_dark_illumination_check_passed	bool	光线状况判断的结果,是否通过测试仅在 query_image_package_check_illumination 开启时返回	true
	is_dark_illumination_check_score	double	光线状况判断的分数仅在 query_image_package_check_illumination 开启时返回	0
	is_anti_picture_check_valid	bool	防照片的判断结果是否有效,若无效则以下两个字段结果无效	true
	is_anti_picture_check_passed	bool	防照片的判断结果,是否通过测试	true
	is_anti_picture_check_score	double	防照片的判断分数	0
	is_anti_eye_blockage_check_valid	bool	防眼部遮挡的结果是否有效,若无效则以下两个字段结果无效	true
	is_anti_eye_blockage	bool	防眼部遮挡的结果,是	true

	_check_passed		否通过测试	
	is_anti_eye_blockage _check_score	double	防眼部遮挡的判断分数	0
	is_anti_hole_check_valid	bool	防孔洞检测结果是否有效,若无效则以下两个字结果无效	true
	is_anti_hole_check_passed	bool	防孔洞检测的结果,是否通过测试	true
	is_anti_hole_check_score	double	防孔洞的判断分数	0

样例输出:

```
{
  "rtn": 0,
  "message": "OK",
  "query_image_package_result": {
    "is_valid_package": true,
    "customer_defined_content": "user defined content",
    "query_image_contents": ["IMAGE_1_BASE64"]
  }
}
```

9.2. 1:1 比对

9.2.1. 应用场景

1:1 比对可应用在需要人脸识别确认用户身份的各种场景,包括有人值守和无人值守环境下的身份认证,如各类柜台业务中辅助柜员确认客户身份和远程登录客户端时的身份验证等。人脸比对的过程,分为从照片中抽取人脸特征值和特征值比对两个部分。前一个部分占用的计算资源和时间均远超前于第二部分(每张人脸的特征值大概 20KB)。

9.2.2. 接口列表

编号	名称	URL
CS201.1	人脸特征抽取服务接口	公有云地址: http://staging.yitutech.com/face/v1/algorithm/recognition/feature_extraction 私有云地址: http://\$IP:9500/face/v1/algorithm/recognition/feature_extraction (其中\$IP 为私有云 IP)
CS202.2	比对服务	公有云地址: http://staging.yitutech.com/face/v1/algorithm/recognition/fac

		e_pair_verification (其中 127.0.0.1 可以替换成具体的私有云 IP)
--	--	---

(一) CS201.1-人脸特征抽取服务接口

(1) 接口功能

对输入图像进行人脸检测，并对人脸区域抽取特征量。

(2) 传输方式

请求服务端	POST
公有云地址	http://staging.yitutech.com/face/v1/algorithm/recognition/feature_extraction
私有云请求地址	http://\$IP:9500/face/v1/algorithm/recognition/feature_extraction (其中\$IP 为私有云 IP)

(3) 接口输入

HTTP Body

参数名	参数类型	说明	必填	默认	限制
image_content	string	输入图像 (JPG) 的 BASE64 编码	是		JPG 编码，图片最大 1MB
auto_rotate	bool	是否开启自动旋转 (如果图片存在 90, 180, 270 度旋转，则开启此选项)	否	false	目前只对非水印照有效
flip_image	bool	是否开启镜像	否	false	
max_faces_allowed	int	最多抽取人脸特征数	否	1	1 或者 2
remove_watermark	bool	是否返回去水印照片	否	false	当且仅当照片类型为水印照时有效
image_type	int	输入图像的类型	是		登记照片的类型 证件照 = 1 证件照翻拍 = 2 类证件照 = 3 芯片照 = 4 金融行业水印的证件照 (不带横纹的老版网纹照) = 5 公安行业水印证件照 (带横纹的老版网纹照) = 7 铁丝网水印的证件照 (新版网纹照) = 9 自动类型水印照 (能够自动区分金融，

					公安， 铁丝网， 正确率 99.5%) = 101
idcardocr	bool	开启身份证识别，将返回身份证信息	否	false	仅在上传翻拍身份证照时有意义 (image_type=2)
idcardocr_mode	int	1: 身份证正面识别 2: 身份证背面识别 3: auto 自动区分正面背面	否	1	仅在上传翻拍身份证照时有意义 (image_type=2)

输入样例：

Sample Input

POST face/v1/algorithm/recognition/feature_extraction HTTP/1.1

Host: http://www.yitu-test.com

Content-Length: \$CONTENT_LENGTH

Content-Type: application/json

x-access-id: access_id

x-signature: e4305df7e61b38adbd59eabef7f8b706

```
{
  "image_content": "照片的 Base64 编码",
  "image_type": 3
}
```

(4) 接口输出

参数名	参数类型	说明	样例值
rtn	int	请求处理结果的错误代码 0 表示 OK， <0 表示服务器发生错误	0
message	string	请求处理结果的错误信息	OK
feature	string	特征向量，一般大小在 26K 左右	
flip_feature	string	特征向量，当输入参数 flip_image 为 true 时,则输出该字符串	
face_num	int	图像识别得到的人脸个数	1
processed_image_content	string	去水印后网纹照图片内容	"BASE64..." 开启去水印会返回 (remove_watermark=true)
image_type	int	图像的类型: 证件照 = 1 证件照翻拍 = 2 类证件照 = 3 芯片照 = 4 小尺寸金融行业水印的证件照（不带横纹的老版网纹照） = 5 中尺寸金融行业水印的证件照（不带横纹的老版网纹照） = 6	1

		小尺寸公安行业水印证件照(带横纹的老版网纹照) = 7 中尺寸公安行业水印证件照(带横纹的老版网纹照) = 8 小尺寸铁丝网水印的证件照(新版网纹照) = 9 中尺寸铁丝网水印的证件照(新版网纹照) = 10	
idcard_type	int	识别出的身份证图片类型(只在开启了 idcard_ocr 功能之后返回) 1: 身份证正面识别 2: 身份证背面识别 -1: 无效的身份证照片, 既不是正面也不是背面	1

输出样例:

Sample Input

```
{
  "rtm": 0,
  "message": "OK",
  "feature": "abcdefg",
  "face_num": 1
}
```

(二) CS202.2 - 比对服务

(1) 接口功能

1:1 比对: 提供登记照片和查询照片, 判断查询照片中是否有登记照片中的人。

(2) 传输方式

请求服务端	POST
公有云地址	http://staging.yitutech.com/face/v1/algorithm/recognition/face_pair_verification
私有云地址	http://127.0.0.1:9500/face/v1/algorithm/recognition/face_pair_verification (其中 127.0.0.1 可以替换成具体的私有云 IP)

(3) 接口输入

参数名	参数类型	说明	必填 默认值	限制
query_image_content	string	查询图片(JPG)的 Base64 编码	否	<1M 查询图片和特征量两个必须有一个
query_image_feature	string	查询人脸的特征量	否	-
query_image_package	string	查询人脸的大礼包	否	来自 yitu 活体 SDK 捕获的照片
query_image_package_check_same_	bool	是否检查大礼包中查询照片是同一个人	否 false	默认会不开启检查

person				
query_image_package_return_image_list	bool	是否返回大礼包中解出的图片列表	否 false	是否返回大礼包中解码出的图片列表， Base64 编码
query_image_type	int	查询图片的类型	是 301	查询照片的类型： 类证件照 = 301 类证件照（手持身份证场景） = 302 类证件照（VTM 场景） = 303 芯片照 = 4 身份证翻拍照 = 2
database_image_content	string	登记照片(JPG)的 Base64 编码	否	<1M 查询图片和特征量两个必须有一个
database_image_feature	string	登记人脸的特征量	否	-
database_image_type	int	登记照片的图片的类型	是	登记照片的类型 证件照 = 1 身份证翻拍照 = 2 类证件照 = 3 身份证芯片照 = 4 金融行业水印的证件照（不带横纹的老版网纹照） = 5 公安行业水印证件照（带横纹的老版网纹照） = 7 铁丝网水印的证件照（新版网纹照） = 9 自动类型水印照（能够自动区分 金融， 公安，铁丝网， 正确率 99.5%） = 101 其中， 101 仅支持图片比对
auto_flip	bool	是否开启自动镜像优化	否 false	只支持查询照为非水印照
auto_rotate_for_query	bool	是否开启查询照的自动旋转识别	否 false	只支持非水印照
auto_rotate_for_database	bool	是否开启登记照的自动旋转识别	否 false	只支持非水印照
true_negative_rate	string	期望的误报率	否 99.99	只能填以下三种字符串 99.9 表示千分之一的误报率 99.99 表示 万分之一的误报率 99.999 表示 十万分之一的误报率
max_faces_all	int	查询照片中至多有几个	否	>=1

owed		人脸	1	
enable_verify_detail	bool	是否返回比对的详细信息， 包含查询照每张人脸和 登记照 人脸分别比对的分数	否 false	
return_face_rect	bool	是否返回查询照每张人脸的位置	否 false	仅当开启 enable_verify_detail = true 时有效
query_image_package_check_anti_screen	bool	是否开启防屏幕翻拍检测	否 false	默认不开启
query_image_package_check_anti_screen_threshold	string	防屏幕翻拍检测阈值	否 "low"	可选字符串参数值： "high" 表示较为严格的检查标准，通过概率约 90%，误报率约 4%。 "low" 表示一般程度的检查标准，通过概率约 95%，误报率约 10%。
query_image_package_check_dark_illumination	bool	是否开启光线昏暗检测	否 false	默认不开启 判断光线不足会返回错误
query_image_package_check_dark_illumination_threshold	string	光线昏暗检测阈值	否 "medium"	表示允许的昏暗程度，可选字符串参数值： "high" 场景非常昏暗，肉眼难以分辨人脸细节； "medium" 场景比对昏暗，肉眼勉强可以分辨人脸细节； "low" 场景相对昏暗，但肉眼可以轻松分辨人脸细节 示例图请参考[附录 1]
query_image_package_check_anti_picture	bool	是否开启防照片检测	否 false	
query_image_package_check_anti_picture_threshold	string	防照片检测的阈值。	否 "medium"	
query_image_package_check_anti_eye_blockage	bool	是否开启眼部遮挡检测，	否 false	
query_image_package_check_anti_eye_blockage_threshold	string	眼部遮挡检测的阈值。	否	

package_check _anti_eye_block age_threshold			"medium"	
query_image_ package_check _anti_hole	bool	是否开启防孔洞检测， 默认值：	否 false	
query_image_ package_check _anti_hole_thr eshold	string	防孔洞检测的阈值。	否 "medium"	
query_image_ additional_info	object	红外照额外信息 { "face_key_point_info": "BALABALA", "infrared_face_image_co ntent": "BASE64" }	否 false	face_key_point_info 和 infrared_face_image_content 默认为空 infrared_face_image_content 为 红外照片

输入样例：

Sample Input

POST /face/v1/algorithm/recognition/face_pair_verification HTTP/1.1

Host: http://staging.yitutech.com

Content-Length: \$CONTENT_LENGTH

Content-Type: application/json

x-access-id: access_id

x-signature: e4305df7e61b38adbd59eabef7f8b706

网纹照比类证件照：

```
{
  "database_image_content": "BASE64 编码的 JPG 登记照片图片内容",
  "database_image_type": 101,
  "query_image_content": "BASE64 编码的 JPG 比对照片图片内容",
  "query_image_type": 301,
  "true_negative_rate": "99.9"
}
```

网纹照比活体检测加密包：

```
{
  "database_image_content": "/9j/...(BASE64 编码的 JPG 网纹照)",
  "database_image_type": 101,
  "query_image_package": "Ymv...(BASE64 编码加密包)",
  "query_image_package_return_image_list": true,
  "query_image_package_check_same_person": true,
  "auto_rotate_for_database": true,
  "true_negative_rate": "99.99"
}
```

(4) 接口输出

参数名 (多级)		参数类型	说明	样例值
rtn	-	int	请求处理结果的错误代码, 0 表示 OK, <0 表示服务器发生错误。	0
global_request_id		string	请求的流水号	eSkrdLifumv0Q QUAB W8AAAAAADu hAQa
message	-	string	请求处理结果的错误信息	OK
pair_verify_result	-	int	0 表示认为是同一个人 1 表示认为不是同一个人	0
pair_verify_similarity	-	double	相似值, 值越大越相似 分数取值范围 0 -100	75
query_image_package_result	-	json::object	如果输入参数中有 query_image_package, 则会返回解大礼包结果	
	customer_defined_content	string	用户在大礼包中预先加入的自定义字段	
	query_image_contents	array	大礼包中解码的图片列表, 若有正脸图片. 则正脸图片为第一张.	
	is_same_person	bool	大礼包同人验证结果	
	is_anti_screen_check_passed	bool	防屏幕翻拍的验证结果, 是否通过了测试, 仅在 query_image_package_check_anti_screen 开启时返回	true
	is_anti_screen_check_score	double	防屏幕翻拍的验证分数仅在 query_image_package_check_anti_screen 开启时返回	0
	is_dark_illumination_check_passed	bool	光线状况判断的结果, 是否通过测试	true 仅在 query_image_package_check_illumination 开启时返回

	is_dark_illumination_check_score	double	光线状况判断的分数	0 仅在 query_image_package_check_illumination 开启时返回
	is_anti_picture_check_valid	bool	防照片的判断结果是否有效 若无效则以下两个字段结果无效	true
	is_anti_picture_check_passed	bool	防照片的判断结果, 是否通过测试	true
	is_anti_picture_check_score	double	防照片的判断分数	0
	is_anti_eye_block_check_valid	bool	防眼部遮挡的结果是否有效 若无效则以下两个字段结果无效	true
	is_anti_eye_block_check_passed	bool	防眼部遮挡的结果, 是否通过测试	true
	is_anti_eye_block_check_score	double	防眼部遮挡的判断分数	0
	is_anti_hole_check_valid	bool	防孔洞检测结果是否有效, 若无效则以下两个字段结果无效	true
	is_anti_hole_check_passed	bool	防孔洞检测的结果, 是否通过测试	true
	is_anti_hole_check_score	double	防孔洞的判断分数	0
verify_detail	-	json::object	比对的详细信息, 仅当开启 enable_verify_detail = true 时返回	
	score_list	array	查询照每张人脸分别和登记照人脸比对的分数, 按从高到低排序	[80.25, 67.89, 23.45]
	query_face_rect	array	查询照每张人脸的矩形位置列表, 与 score_list 中的每个比对应, 仅当开启 return_face_rect = true 时返回	[{}, {}, {}]

	query_face_rect[i]	json::object	查询照每张人脸的矩形位置，包含左上角坐标、矩形的宽度和高度	{ "h": 366, "w": 366, "x": 84, "y": 75 }
--	--------------------	--------------	-------------------------------	---

输出样例：

Sample Input

```
{
  "rtn": 0,
  "message": "OK",
  "pair_verify_result": 0,
  "pair_verify_similarity": 60.0
}
```

9.3. 1:N 比对

9.3.1. 应用场景

1:N 比对可用于黑名单检索和 VIP 识别，通过建立人像库，快速判定当前用户的身份。如当新用户注册或申请贷款时，可以在库中查找是否为黑名单中的人；或可用于注册查重，将新用户照片在已有用户库中查找，找出使用不同身份信息反复注册的用户；在营业厅或零售门店，可以快速识别 VIP 客户，并提供定制化服务，提升用户体验。

*注：当前 1:N 检索支持跨库检索和多库操作，库容量支持千万量级。但是私有云版本只支持单机版，无动态扩容，100 万人像库需 2G 内存。

9.3.2. 接口列表

编号	名称	URL
102.1	人像库管理接口	/face/basic/register_in_repo
103.1	人像库检索接口	/face/basic/retrieval_in_repo

（一）102.1-人像库管理接口-人像库上传

（1）接口功能

上传用户人员信息至依图人像库。

（2）传输方式

请求服务端	POST
公有云地址	http://staging.yitutech.com/face/basic/register_in_repo

（3）接口输入

参数名（多级）	参数类型	说明	必填	默认	限制
mode		int 上传模式，包括如下几种： 1: 上传图片模式(默认) 2: 上传特征模式	否	1	
user_info	user_id	string 人像标识 ID	是	""	最长 18 个字节，不能重复，不能包含下划线
	repo_name	string 上传到库的名称（用户自定义），不同的库之间彼此独立，不能为空	是	-	最长 18 个字节，不能包含下划线
	label	int 库类型 1: 黑名单库 2: 用户库（用于 1:N） 3: 人像库（用于自动聚类）	是	-	
	timestamp	int 人像产生的时间戳 毫秒数	否	-	
	description	json::object 描述此照片的信息，自定义描述	否		
	image_content	string 经 Base64 编码的用户登记照的内容。 照片要求为 JPEG 格式。	当 mode 为 1 时 必填	""	最大图片大小： 3MB 需满足相应图片类型定义[附录 1]
	feature_content	string 经 Base64 编码的用户登记照特征	当 mode 为 2 时 必填		
options	image_type	int 图片类型，包括如下几种： 1: 证件照 301: 类证件照	否	1	具体各图片类型的定义 请 参 考 [附录 1]

输入样例：

```
{
  "mode": 1,
  "user_info": {
    "user_id": "b001",
    "label": 1,
    "image_content": "BASE64 编码的 JPG 比对照片图片内容"
  },
  "options": {
    {
```

```

    "image_type": 301
  }
}

```

(4) 接口输出

参数名（多级）	参数类型	说明	示例值	限制
rtn	int	请求处理结果的错误代码 0 表示 OK，否则表示异常。	0	返回值的具體語義請參考[附录 2]
message	string	请求处理结果的 debug 信息	"OK"	

输出样例：

```

{
  "rtn": 0,
  "message": "OK"
}

```

(二) 102.1-人像库管理接口-人像库删除

(1) 接口功能

上传用户人员信息至依图人像库。

(2) 传输方式

请求服务端	POST
公有云地址	http://staging.yitutech.com/face/basic/register_in_repo

(3) 接口输入

参数名（多级）	参数类型	说明	必填	默认	限制
user_info	user_id	string	人像标识 ID 会删除库中所有人像 ID = 给定 ID 的人像	是	最长 18 个字节 不能包含下划线
	label	int	库类型 1: 黑名单库 2: 注册库	是	
	repo_name	string	上传到库的名称（用户自定义） 不同的库之间彼此独立	是	最长 18 个字节 不能包含下划线

输入样例：

```

{
  "user_info":
  {
    "user_id": "b001",
    "label": 1
  }
}

```

(4) 接口输出

参数名（多级）	参数类型	说明	示例值	限制
rtn	int	请求处理结果的错误代码 0 表示 OK，否则表示异常。	0	返回值的具体语义请参考[附录 2]
message	string	请求处理结果的 debug 信息	"OK"	

输出样例：

```
{
  "rtn": 0,
  "message": "OK"
}
```

（三）103.1-人像库检索接口

（1）传输方式

请求服务端	POST
公有云地址	http://staging.yitutech.com/face/basic/retrieval_in_repo

（2）接口输入

参数名（多级）	参数类型	说明	必填	默认	限制
mode	int	查询信息上传模式， 包括如下一种： 1: 上传图片进行查询（默认） 2: 上传特征进行查询	否	1	
user_info	label	用户类型标识 1: 在黑名单库中进行检索 2: 注册用户	是		
	repo_name	在指定库中进行搜索 如果为空，表示在所有库中进行检索 默认为空	否	""	最大 18 个字节 库名不能包含下划线
	image_content	经 Base64 编码的用户 登记照的内容。 照片要求为 JPEG 格式。	mode 为 1 时 必填	""	最大图片大小： 3MB
	feature_content	经 Base64 编码的用户 登记照特征内容	mode 为 2 时 必填	""	
options	image_type	图片类型，包括如下一种： 301: 类证件照	是		
	auto_rotate	开启自动旋转矫正	否	false	
	auto_flip	开启自动镜面翻转	否	false	暂不支持此参数
	true_	期望误报率，用来确定分数	否	"99."	目前可选：

	negative_rate		阈值 一般来说, 10,000 的库做 1:N 如果期待的布控比对误报率为 1/1000 则应该选择 $10,000 * 1,000$ = 千万分位误报率		99999	"99.9" 千分位误报率 "99.99" 万分位误报率 "99.999" 十万分位误报率 "99.9999" 百万分位误报率 "99.99999" 千万分位误报率 "99.999999" 亿分位误报率 "99.9999999" 十亿分位误报率
	return_face_image	bool	是否返回比对命中的人脸图片	否	false	
	return_topk_face	int	返回前多少个最相似的人像	否	1	return_topk_face <= 10 按照相似度从高到低排序

输入样例:

```

{
  "session_id": "bb73024589d9d4d66a11ad28bd4e18db",
  "mode": 1,
  "user_info": {
    "label": 1,
    "image_content": "BASE64 编码的 JPG 图片内容"
  },
  "options": {
    "image_type": 301,
    "auto_rotate": false,
    "auto_flip": false,
    "true_negative_rate": "99.99",
    "return_face_image": true
  }
}
  
```

(4) 接口输出

参数名 (多级)	参数类型	说明	示例值	限制
rtn	int	请求处理结果的错误代码 0 表示 OK, 否则表示异常。	0	返回值的 具体语义 请参考[附

					录 2]
message		string	请求处理结果的 debug 信息	"OK"	
repo_result json::array	is_same_person	bool	系统判断是否和检索人像为同一人	true	
	user_id	string	库中人像用户名	"b001"	
	confidence	double	库中人像和检索人像的相似度	75.0	
	description	json::object	库中人像的 description		
	face_image_content	string	库中人像的人脸照片 base64 编码		

输出样例：

```
{
  "rtn": 0,
  "message": "OK",
  "repo_result": [
    {
      "user_id": "b001",
      "confidence": 79.2,
      "face_image_content": "BASE64 编码的 JPG 图片内容"
    }
  ]
}
```

9.4. OCR 识别

9.4.1. 应用场景

身份证识别的应用场景十分广泛，在有人值守的柜台或无人值守的远程终端都能够发挥作用，具体适用于银行业务办理的远程开户、放贷和各种类型的柜台业务办理。它能够有效避免手动输入身份信息的繁琐过程和可能出现的造假情况，同时还能够提高业务办理的效率，提升用户体验。另外，身份证识别获取的信息由于有着规范的格式，能够作为接下来的业务流程依据，根据身份信息建立合理的风控模型。

依图 OCR 采用前端+后端模式，前端基于 SDK 实现身份证自动捕获，可以保证最佳光照和角度条件，通过将该捕获图片的 Base64 编码传至对应接口，保证后端信息识别的成功率和准确率。

*注：当前 OCR 在对少数民族，4 字名字的识别存在一定误差，且不支持临时身份证的识别。

9.4.2. 接口列表

编号	名称	URL
105.1	OCR 通用接口	/face/basic/ocr

105.2	证件鉴别通用接口	/face/basic/distinguish
105.3	证件分类通用接口	/face/basic/license_classification

(一) 105.1- OCR 通用接口

(1) 接口功能

识别用户上传的证件照片并返回识别结果。

(2) 传输方式

请求服务端	POST
公有云地址	http://staging.yitutech.com/face/basic/ocr

(3) 接口输入

参数名 (多级)	参数类型	说明	必填	默认	限制
mode	int	OCR 内容模式, 包括如下几种: 1: 上传图片进行查询 2: 上传特征进行查询	否	1	只能为 1,2
user_info	image_content	mode = 1: 经 Base64 编码的翻拍证件照的内容。 照片要求为 JPEG 格式。	mode = 1 时 (默认) 时必填 mode = 2 时不填		最大图片大小: 3MB 需满足相应图片类型定义[附录 1]
	feature_content	mode = 2: 经 Base64 编码的用户特征内容	默认下不填 mode = 2 时必填		限制大小: 3MB
options	ocr_type	识别的类型, 包括如下几种: 1: 翻拍身份证照 2: 驾驶证 3: 行驶证	是		
	auto_rotate	开启自动旋转矫正 (目前只支持身份证 OCR)	否	false	
	ocr_mode	当 ocr_type = 1 时 1: 身份证正面识别 2: 身份证背面识别 3: auto 自动区分身份证正面背面双面 4: 身份证双面识别	当 ocr_type 为 1 时必填,	1	

(4) 接口输出

身份证相关接口输出说明:

参数名 (多级)	参数类型	说明	示例值	限制
ret	int	请求处理结果的错误代码	0	返回值的语义

			0 表示 OK， 则表示异常。		请参考[附录 2]
message		string	请求处理结果的 debug 信息	"OK"	
idcard_	name	string	身份证翻拍照上的姓名信息	"张三"	
ocr_res	addre	string	身份证翻拍照上的地址信息	上海市...	
ult	ss				
	citizen_id	string	身份证翻拍照上的身份证号信息	310...	
	valid_date_begin	string	身份证有效期 开始时间	20141213	当识别结果不合理时会返回"?"
	valid_date_end	string	身份证有效期 结束时间	20341213	当识别结果不合理时会返回"?" 当识别结果为"长期"时会返回"长期"
	gender	string	性别	男	
	nation	string	民族	汉	
	birthday	string	出生年月日	19880409	
	agency	string	签发机关	上海市公安局浦东分局	
	idcard_type	int	识别出的身份证图片类型 1: 身份证正面识别 2: 身份证背面识别 3: 身份证双面 -1: 无法识别	1	

驾驶证相关接口输出说明

参数名		参数类型	说明	示例值
rtn		int	请求处理结果的错误代码 0 表示 OK， 则表示异常。	0
message		string		"OK"
driving_license_ocr_result	birthday	string	生日	20141213
	issue_date	string	签发日期	20141213
	valid_date_begin	string	驾驶证有效起始时间	20141213
	valid_date_end	string	驾驶证有效结束时间（新版驾驶证）	20141213
	valid_period	int	驾驶证有效期年限（旧版驾驶证）	6
	vehicle_type	string	准驾车型	
	name	string	姓名	张三

	address	string	住址	上海市..
	citizen_id	string	身份证号	319...
	gender	string	性别	女
	license_type	int	识别出的驾驶证类型 1: 新版 2: 旧版 -1: 无法识别 -2: 不是驾驶证	1

(5) 典型调用样例

身份证正面 OCR 识别

输入示例	输出示例
Http Header POST /face/basic/ocr HTTP/1.1 Host: 121.42.167.95 Content-Length: \$CONTENT_LENGTH Content-Type: application/json x-access-id: \$ACCESS_ID x-signature: \$SIGNATURE Http Body <pre>{ "user_info": { "image_content": "...", }, "options": { "auto_rotate": true, "ocr_mode": 1, "ocr_type": 1 } }</pre>	<pre>{ "idcard_ocr_result": { "address": "贵州省六盘水市北海路 244-1-402 号", "citizen_id": "111111111111111111X", "gender": "男", "nation": "汉", "idcard_type": 1, "birthday": "1990.09.06", "name": "老王" }, "message": "OK", "rtn": 0 }</pre>

身份证背面 OCR 识别

--	--

--	--

--	--

--	--

--	--

（二）105.2-证件鉴别通用接口

（1）接口功能

鉴别客户上传的证件的真伪。

（2）传输方式

请求服务端	POST
公有云地址	http://staging.yitutech.com/face/basic/distinguish

(3) 接口输入

参数名（多级）		参数类型	说明	必填	默认	说明
mode		int	1:上传图片模式 2:上传特征模式	否	1	只能为 1,2
user_info	image_content	string	mode = 1 经 Base64 编码的 翻拍证件照的内容。 照片要求为 JPEG 格式。	mode =1 （默认） 时必须填 否则不填		最大图片大小：3MB 需满足相应图片 类型定义[附录 1]
	feature_content	string	mode = 2 Base64 编码的用户特征 内容	mode = 2 时必须填		最大特征大小：3MB
options	distinguish_type	int	需要鉴别的类型，包括 如下几种： 1: 身份证假证鉴别 2: 驾驶证假证鉴别	否	1	只能为 1,2
	auto_rotate	bool	开启自动旋转矫正 (for 身份证鉴别)	否	false	

(4) 接口输出

身份证相关接口输出说明

参数名 (多级)	参数类型	说明	示例值	限制
rtn	int	请求处理结果的错误代码 0 表示 OK, 则表示异常。	0	返回值的语义请参考[附录 2]
message	string	请求处理结果的 debug 信息	"OK"	
idcard_distinguish_result	is_valid	bool	是否是真实有效的身份证照片 (deprecated, 请以 result 为准)	是: 表示确实是真实的身份证正面照 否: 表示不是真实有效的身份证正面照
	confidence	double	是有效证件照的置信度 0-1	目前暂时没有置信度
	debug_image	string	若为真证, 返回经过矫正后的照片; 若为假, 返回标出假证位置的图片	base64...
	result	int	鉴别结果	-2 不是有效证件 -1 无法判断

					0 伪造证件 1 真实证件
	reason	string	判断理由		只有不为真实的身份证正面照时才有, 原因目前有以下几种: 假身份证 无法判别 图片过于模糊 不是身份证

驾驶证相关接口输出说明

参数名 (多级)		参数类型	说明	示例值	限制
rtn		int	请求处理结果的错误代码, 0 表示 OK, 否则表示异常。	0	返回值的语义请参考 [附录 2]
message		string	请求处理结果的 debug 信息	"OK"	
driving_license_distinguish_result	is_valid	bool	是否是真实有效的驾驶证		是: 表示确实是真实的驾驶证 否: 表示不是真实有效的驾驶证
	confidence	double	是有效证件照的置信度 0-1		目前暂时没有置信度
	result	int	鉴别结果		-2 不是有效证件 -1 无法判断 0 伪造证件 1 真实证件
	reason	string	判断理由		只有不为真实的驾驶证时才有 原因有以下几种: 无法判断 假证 太模糊, 无法判断 质量太差, 无法判断 不是驾驶证, 无法判断 不是单个证

(5) 典型调用样例

驾驶证真假鉴别

输入示例	输出示例
Http Header POST /face/basic/distinguish HTTP/1.1 Host: 121.42.167.95	<pre>{ "driver_license_distinguish_result": { "is_valid": true, "confidence": 9 } }</pre>

Content-Length: \$CONTENT_LENGTH Content-Type: application/json x-access-id: \$ACCESS_ID x-signature: \$SIGNATURE Http Body <pre>{ "user_info": { "image_content": "...", }, "options": { "auto_rotate": true, "distinguish_type": 2 } }</pre>	<pre>}, "message": "OK", "rtn": 0 }</pre>
---	---

(三) 105.3-证件分类通用接口

(1) 接口功能

识别用户上传的证件照片并返回证件分类结果。

(2) 传输方式

请求服务端	POST
公有云地址	http://staging.yitutech.com/face/basic/license_classification

(3) 接口输入

参数名(多级)	参数类型	说明	必填	默认值	说明
mode		int 内容 mode 1 为 image_content 2 为 feature_content		1	只能是 1 2
user_info	image_content	string mode = 1 : 默认经 Base64 编码的翻拍证件照的内容。照片要求为 JPEG 格式。	mode = 1 (默认) 时候必填 mode = 2 不填		最大图片大小: 3MB 需满足相应图片类型定义[附录 1]
	feature_content	string mode = 2 : 经过 base64 编码的特征内容(如果提供了这个参数, 不提供 image_content 参数)	mode = 2 时候必填 mode = 1 时候不填		最大图片大小: 3MB
options	license_type	int 需要分类的类型, 包括如下一种: 1: 身份证证件分类	是		只能为 1

输入样例:

Http Header

POST /face/basic/license_classification HTTP/1.1

Host: 121.42.167.95
Content-Length: \$CONTENT_LENGTH
Content-Type: application/json
x-access-id: \$ACCESS_ID
x-signature: \$SIGNATURE

Http Body

```
{
  "user_info": {
    "image_content": "...",
  },
  "options": {
    "license_type": 1
  }
}
```

(4) 接口输出

参数名（多级）		参数类型	说明	示例值	限制
rtn		int	请求处理结果的错误代码 0 表示 OK，否则表示异常。	0	返回值的语义请参考[附录 2]
message		string	请求处理结果的 debug 信息	"OK"	
classification_result	result	int	分类结果 1 身份证正面 2 身份证反面 3 身份证双面 -1 无效证件	1	

输出样例：

```
{
  "classification_result": {
    "result": 1,
  },
  "message": "OK",
  "rtn": 0
}
```