

FacePass Android版产品白皮书

版本号	最后编辑	编辑时间
v2.0.0	叶赛尔	2018-2-27

- 产品概述
- 功能模块
 - 整体模块
 - 抓拍模式
 - 云离线模式
 - 脱机模式
- 核心流程
 - 底库管理
 - 人脸识别
- 算法性能
 - 人脸检测
 - 人脸识别
 - 人脸活体
 - 人脸比对
 - 人脸注册
- 硬件平台
- 授权模式

产品概述

FacePass是旷视科技(Face++)线下全场景的通用型SDK，主要提供1:N检索、1:1比对、人脸检测、活体检测等线下场景常见功能，搭载旷视科技(Face++)全球领先的深度学习算法，提供极致优秀的产品体验。

FacePass是一个“懂业务”的SDK，基于旷视科技(Face++)在安防、通行等领域多年的行业经验，FacePass在算法原子API之上封装了的通用业务逻辑层，并给出了相应的算法最佳实践，从而大幅提高客户的二次开发效率。

FacePass v2

FacePass v2是对FacePass v1的一个升级版本，旨在统一v1时代的抓拍版和离线版，提供一个“一版本、一授权、一定价”的大一统版本。

一版本：提供客户端SDK+服务器SDK的统一版本，不再区分抓拍、离线和脱机的不同工程版本；

一授权：统一通过客户端授权（芯片授权或者软授权），虽然保留了服务器加密狗，但是仅供授权、不再进行实际商务控制（比如控制路数）；

一定价：统一按照路数收费，不再通过服务器并发等形式收费。

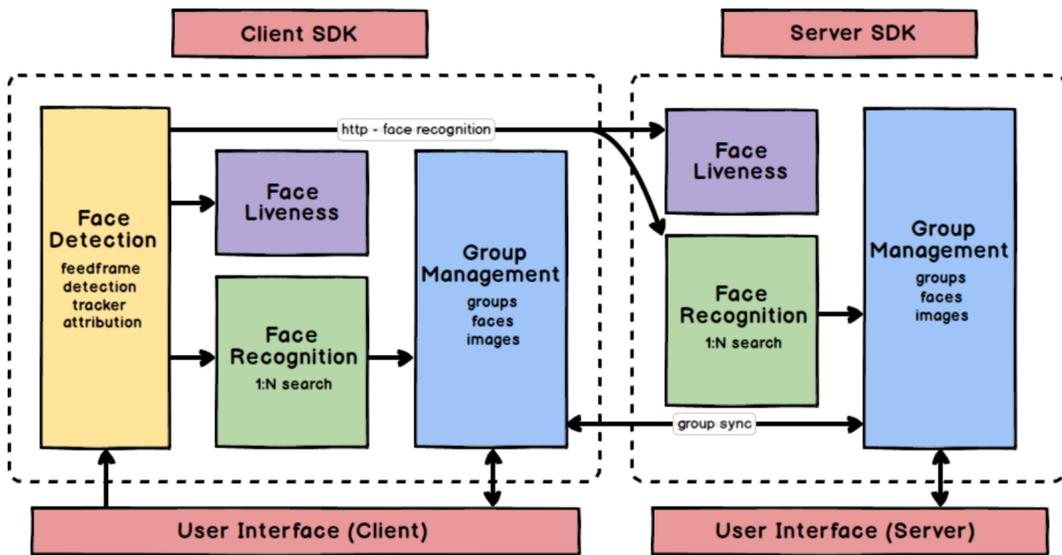
功能模块

我们从三个维度来介绍功能模块组成：

- 工程维度：客户端SDK + 服务器SDK；
- 功能维度：人脸检测、人脸识别、活体检测、底库管理、底库同步、日志管理、设备管理；
- 模式维度：抓拍模式、云离线模式、本地离线模式。

整体模块

我们先根据“工程维度”和“功能维度”用一张图表来描述不同功能模块在两个SDK的分布：



说明：

功能模块	SDK分布	输入	输出	其他说明
人脸检测	仅在客户端SDK	客户端相机帧（即所谓feedFrame）	主要包括： <ul style="list-style-type: none">人脸数据（若干人脸框等）属性信息（角度/模糊度等）识别控制信息	实际上过程非常复杂，主要包括： <ul style="list-style-type: none">人脸相机帧处理人脸检测人脸跟踪人脸属性
人脸识别	分布在客户端SDK或者服务器SDK	人脸检测的输出。	主要包括： <ul style="list-style-type: none">识别到的人脸信息（FaceToken）识别分值	人脸识别仅接受FacePass客户端输入，同时识别必须是针对一个底库的。
活体检测	分布在客户端SDK或者服务器SDK	人脸检测的输出	主要包括： <ul style="list-style-type: none">活体分值	当前版本支持单目RGB摄像头活体，暂未支持红外、3D结构光等技术。
底库管理	分布在客户端SDK或者服务器SDK	识别模块的调用	主要包括： <ul style="list-style-type: none">识别到的人脸信息（FaceToken）识别分值	人脸识别必须是针对一个底库的。底库管理支持灵活的API，客户可以根据自己的需要进行配置。
		用户的API操作	人脸的增删改查 底库的增删改查	

对对应到三种不同的使用模式上：

模式	抓拍	识别/活体	底库管理	说明	优势	劣势	建议场景
抓拍模式	客户端	服务器	服务器	客户端只做抓拍，识别和底库管理均在服务器完成。	在服务器上运行算法，可以支持20000人底库识别，和最优的活体检测。	必须云+端配合部署，无法脱机使用，网络要求高。	私有云局域网、带宽稳定。对底库大小、活体品质有较高要求。
云离线模式	客户端	客户端	服务器	客户端保留有一份底库，源于服务器同步下发，客户端不提供底库管理API。	本地识别，支持一段时间脱机使用。可以使用云来统一管理所有端底库。	在客户端上运行算法，支持2000人底库识别，和服务器运算有一定的差距。	私有云局域网、带宽稳定。多台设备共享底库、适合统一管理。
脱机离线模式	客户端	客户端	客户端	客户端保留有一份底库，	本地识别，支持永久脱机使用。		脱机离线模式、一经授权永久使用。

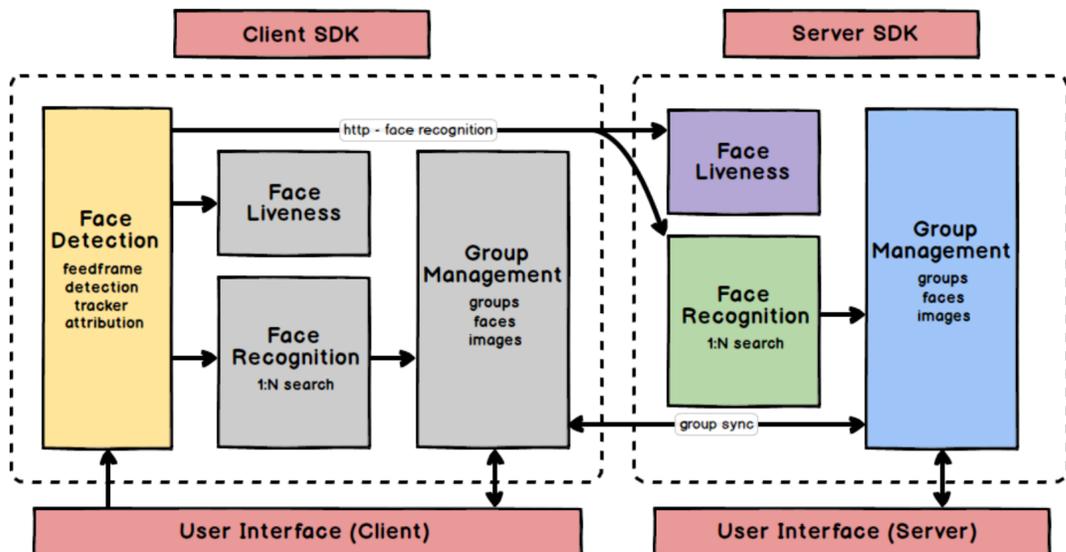
有完整的底库管理API,
可以在无服务器的情况下脱机使用。

可以在本地直接进行底库管理。

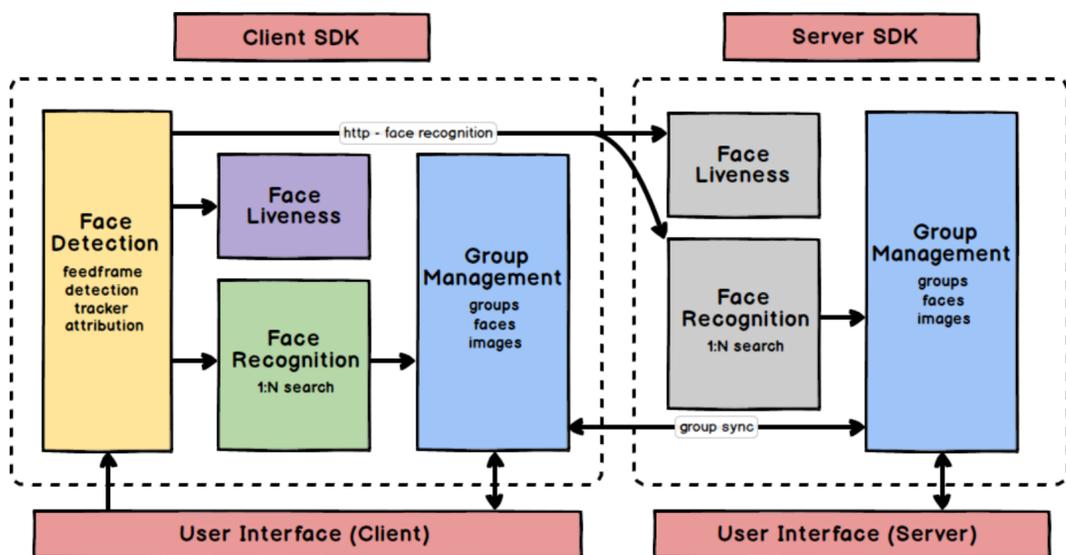
需要本地客户端直接进行底库管理。

图解一下三种模式下对功能模块的使用。

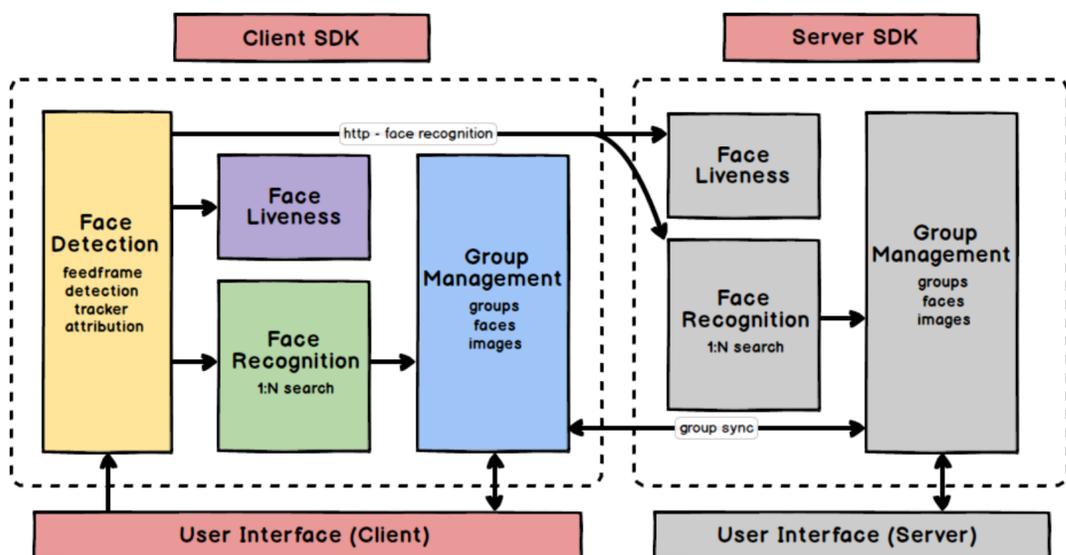
抓拍模式



云离线模式



脱机模式



以上就是对主要核心功能模块以及三种模式下的介绍，客户实际使用中可以根据实际需要来实现。

核心流程

这里我们着重介绍一下两个最核心的流程：

- 1.底库管理：包括人脸注册、底库建立、底库人脸绑定等操作；
- 2.人脸识别：包括track的概念、feedframe的输入输出、人脸识别、活体检测等。

底库管理

底库是一个人脸特征值的集合，包含了客户希望用于“被识别”的人脸。

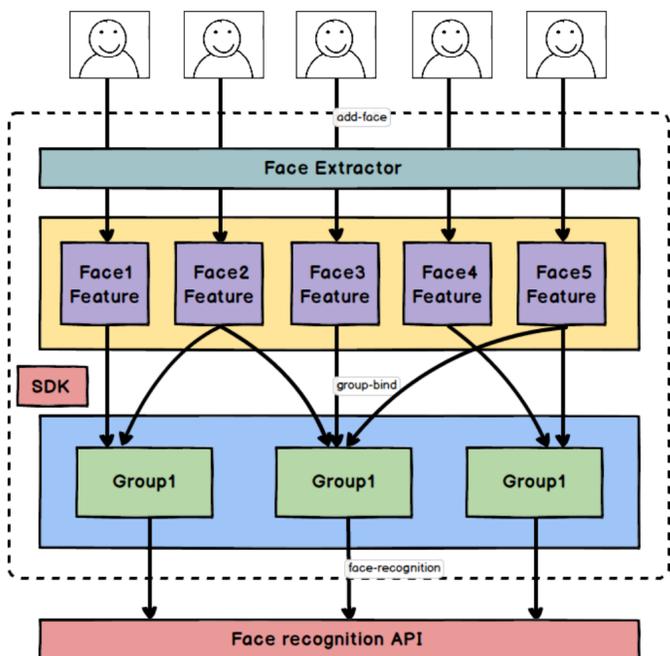
FacePass的人脸识别都是基于底库的，一次人脸识别的输入参数是一张目标人脸和一个底库名称。

人脸是通过人脸注册的API进行注册的，SDK内部是通过一个extractor把一张人脸降维成一个可用于识别算法的特征值。

人脸特征值库是独立于底库的，人脸库是以池子的形式存在，底库和人脸是绑定关系。

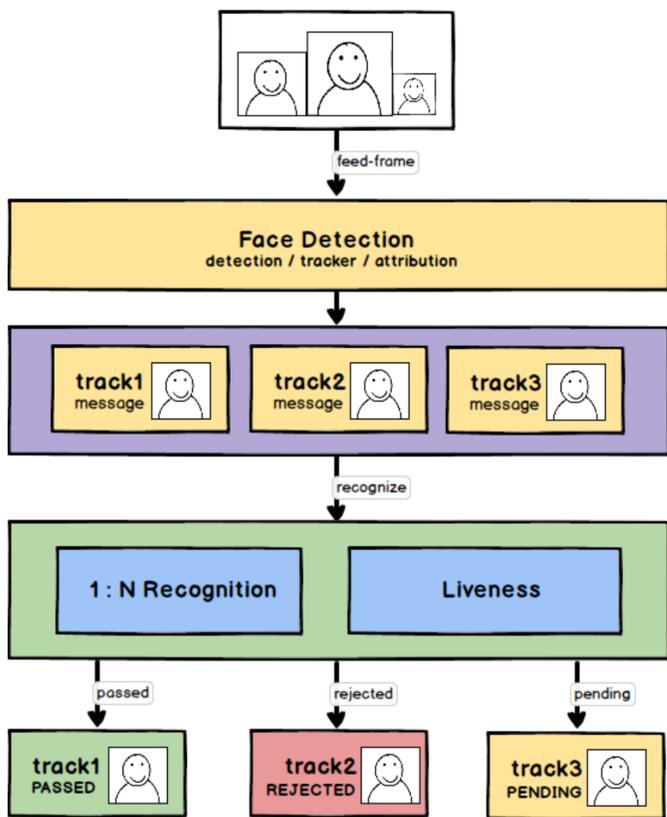
也就是说，人脸注册完之后，都是进了人脸大池子，然后每个底库都可以从这个池子里选择需要的目标人员进行绑定。

具体如下图示：



人脸识别

先参见下图：



FacePass中有track的概念，可以理解为“一个人”，同一个人摄像头前连续多帧人脸，都会被视为同一个track。

对应的，识别结果也是针对track而言的：同一个track如过有一帧已经识别通过，这个track就没有再进行识别的必要了。

客户的操作接口其实非常简单，就是一个feedFrame函数，将一帧相机数据输入给SDK，SDK就会输出当前帧中每一个track的信息了。

信息主要包括：

- track的Id：用于不同帧的人脸之间关联到同一个人；
- 抓拍人脸图：从相机帧中截取的小脸的数据流，是用于发送给识别的源数据；
- 人脸信息：包括人脸rect、角度pose、模糊blur等，可供客户业务层做判断，以实现必要的业务逻辑或者GUI显示；
- 控制信息：用于人脸识别时候必要的控制字段等。

用户可以根据这些信息，封装成合适的格式，发送识别接口进行人脸识别。

目前v2.0支持两种不同的识别模式：

- 抓拍模式：客户端只是抓拍，然后通过http封包，发送给/service/recognize函数进行识别并得到返回结果；
- 离线模式：在客户端同时进行抓拍和识别，所以直接调用客户端SDK的recognize函数即可；

注：云离线模式和脱机模式都属于离线模式，差别在于底库的管理，在识别的问题上两者是一致的，所以这里放到一起描述了。

不管哪一种模式，识别本质上都做了：

- 1:N人脸识别：根据上传的人脸图片流和一个底库名称，去找出底库中和目标图片最相似的人，并给出对应的得分；
- 活体检测：如果开启了活体开关，则会对人脸进行活体检测，未通过检测的图片会被视为活体攻击；

识别过程完成之后SDK会对应的更新每个track的状态，比如：

- 如果识别和活体都通过了，则此track视为通过，不再进行识别；
- 如果识别或活体之一没通过，会继续重试，等待下一帧识别；
- 如果连续重试次数太多，超过了上限，也会视为识别不通过；

具体的状态和参数定义参见SDK文档，这里仅做概念上的说明。

算法性能

算法性能我们主要分为三块：

- 效果：识别的准确度等算法表现；
- 性能：算法在指定硬件上运算的速度；
- 并发：主要协议在指定硬件的并发度。

人脸检测

算法模型	定义	定义说明
Detect	人脸检测	算法针对全图进行人脸的检测，相对耗时。
Track	人脸跟踪	算法针对已检出人脸进行就近的跟踪，相对快速，但是无法检测到画面中新增的人脸。
Pose	角度属性	人脸角度，[-90, 90]，分为yaw/pitch/roll三个方向，具体参见SDK文档。角度越大，越不合适做人脸识别。
Blur	模糊属性	人脸模糊度，[0,1]。人脸越模糊，越不合适做人脸识别。

效果类型	效果参数	参数定义	效果指标	备注
Detect	检出率	正样本集 = 存在合格人脸的照片	99.7%	高检出率确保真人不被漏检，

		检出率 = 成功检出人脸的正样本 / 正样本集总数		
	误检率	负样本集 = 不存在合格人脸的照片 误检率 = 错误检出人脸的负样本 / 负样本集总数	0.1%	误检率依赖于负样本集的选择， 目前FacePass使用的是某国际标准负样本集。
Track	中断率	track中断率 = 断track导致的新增track数 / 总track数	0.9%	track中断会导致一个人视为多个人，增加负载。
Pose	角度差	角度差 = (人脸实际角度 - 算法检测角度) 的绝对值	2.8°	角度差越小，说明算法识别角度越准。
Blur	模糊差	模糊差 = (人脸实际模糊度 - 算法检测模糊度) 的绝对值	0.01	模糊差越小，说明算法识别模糊度越准。

性能类型	参数定义	硬件选型	性能指标
Detect	对标准1080P图片进行一次detect的速度	RK3288	75ms
Track	对标准1080P图片的人脸进行一次track的速度		10ms
Pose	对标准1080P图片进行一次pose检测的速度		10ms
Blur	对标准1080P图片进行一次blur检测的速度		10ms

注：客户端上的检测/跟踪并发意义并不大，同时受制于不同客户端对资源的分配策略，故这块我们未做专门测试。

人脸识别

算法模型	定义	定义说明
1:N Recognition	人脸识别	从底库的N个人中，找到最相似的人，并给出对应分值

参数名称	参数定义	备注
通过率	正样本集 = 目标人员在底库内的样本 通过率 = 目标人员正识通过的正样本数 / 正样本集总数	正识 = 首位命中底库本人 并且 分值高于阈值， 所以通过率受阈值影响。
误识率	负样本集 = 目标人员不在底库内的样本 误识率 = 目标人员误识通过的负样本数 / 负样本集总数	误识 = 首位命中底库他人 并且 分值高于阈值， 所以误识率也受阈值影响。

识别图片的质量、现场环境、底库大小、阈值设置等，都会影响识别的通过率和误识率。

下表给出客户端和服务端两个平台下、一般场景下、不同底库大小下、建议阈值下的通过率和误识率。

效果平台	底库大小	通过率	误识率	阈值
客户端	500人	99.2%	2.0%	73
	1000人	99.0%	2.0%	73
	2000人	98.5%	2.0%	73
服务器	5000人	99.8%	0.2%	77
	10000人	99.7%	0.2%	78
	20000人	99.5%	0.2%	79

性能平台	硬件选型	平均响应时间	并发上限
客户端	RK3288	350ms	-
服务器	i7	280ms	6.5
	E5	300ms	25
	GTX1070	180ms	100

人脸活体

算法模型	定义	定义说明
Liveness	活体检测	检测一张人脸是否为真人、或是手机照片等攻击手段。

参数名称	参数定义	备注
通过率	正样本集 = 真人的人脸 通过率 = 检测为真人的正样本数 / 正样本集总数	通过率受阈值影响。
误识率	负样本集 = 非真人的人脸（照片、手机等） 误识率 = 检测为真人的负样本数 / 负样本集总数	误识率也受阈值影响。

活体的效果依赖因素也非常多，包括图片质量、现场环境、活体阈值等。

尤其需要指出的是，活体识别对于不同材质的攻击手段的防范能力是有差别的，下表给出了常见材质的活体表现。

攻击材质	通过率	误识率
手机/Pad/Mac	99.5%	0.8%
胶卷照片	99.5%	1.0%
A4纸打印	99.5%	1.6%
铜版纸打印	99.5%	0.6%
折边纸质面具	99.5%	4.2%

性能平台	硬件选型	平均响应	并发上限
客户端	RK3288	350ms	-
服务器	i7	200ms	6.5
	E5	240ms	25
	GTX1070	210ms	100

人脸比对

算法模型	定义	定义说明
1:1 Compare	人脸比对	比对两张人脸，给出相似度，并根据阈值判定是否视为同一个人

参数名称	参数定义	备注
------	------	----

参数名称	参数定义	备注
通过率	正样本集 = 本人比对本人 通过率 = 目标人员正识通过的正样本数 / 正样本集总数	正识 = 相似度分值高于阈值，所以通过率受阈值影响。
误识率	负样本集 = 本人比对他入 误识率 = 目标人员误识通过的负样本数 / 负样本集总数	误识 = 相似度分值高于阈值，所以误识率也受阈值影响。

识别图片的质量、现场环境、阈值设置等，都会影响识别的通过率和误识率。

下表给出客户端和服务端两个平台下、一般场景下、不同建议阈值下的通过率和误识率。

效果平台		通过率	误识率	建议阈值
客户端		99.7%	1%	63
		98.7%	0.1%	70
		94.0%	0.01%	86
服务器		99.8%	1%	63
		99.7%	0.1%	70
		99.5%	0.01%	86

性能平台	硬件选型		平均响应	并发度
客户端	RK3288		350ms	-
服务器	i7		420ms	1*
	E5		450ms	1*
	GTX1070		220ms	1*

注：当前版本1:1限制了并发上限为1TPS。

人脸注册

算法模型	定义	定义说明
Add-Face	人脸注册	将一张包含人脸的照片进行注册的操作

参数名称	参数定义	备注
注册率	正样本集 = 事实上符合注册标准的照片 注册率 = 正确注册的正样本数 / 正样本集总数	实际使用过程中非常依赖照片集的质量，以下参数是基于某国际标准人脸集合。

效果类型		注册率
Add-Face		99.0%

性能平台	硬件选型		平均响应	并发上限
客户端	RK3288		350ms	1
服务器	i7		260ms	10
	E5		280ms	10
	GTX1070		130ms	30

人脸检测

硬件平台

FacePass致力于打造旷视科技(Face++)线下全场景通用SDK产品，所以客户端会逐步增加对不同平台的兼容适配，包括但不限于：

- Arm Android
- Arm Linux
- x86 Linux
- x86 Windows
- iOS
- Tx1
- FPGA

当前版本支持客户端Arm Android和服务端x86 Linux。

另外，实际硬件的配置、操作系统等都会影响到算法的实际效果，以下给出的是客户端和服务器各自的建议配置：

类型	型号	性能
CPU	A17	1.6GHz 4核
内存	-	2G

类型	数量	型号	性能
CPU	1	i7-4790k	4GHz 4核
内存	1	-	16G

客户端的性能很影响整个识别体验，所以我们强烈建议客户使用RK3288以上的配置。

一般来说，算法性能和主频成正比，客户可以自行计算对应的算法性能（但是不影响算法效果，比如识别率等）。

操作系统	建议版本
Android	5.0及以上版本均可

类型	数量	型号	性能
CPU	1	i7-4790k	4GHz 4核
GPU	1	gtx 1070	1.6GHz 1920核 8G gddr5

服务器

目前主流服务器的CPU配置可以满足算法运行需要，下表给出的是不同门禁机路数下的建议配置。

注：以下配置仅对使用“抓拍模式”的客户有意义，云离线模式和脱机模式无需云端运行识别和活体算法，所以除非有特殊的需要，否则性能消耗不大。

类型	数量	型号	性能
CPU	1	i7-4790k	4GHz 4核
内存	1	-	16G

类型	数量	型号	性能
CPU	1	i7-4790k	4GHz 4核
GPU	1	gtx 1070	1.6GHz 1920核 8G gddr5

类型	数量	型号	性能
内存	1	-	32G

100路门禁机：

类型	数量	型号	性能
CPU	2	E5-2630	2.4GHz 4核
GPU	2	gtx 1070	1.6GHz 1920核 8G gddr5
内存	1	-	64G

操作系统	建议版本
Unbutu	16.04及以上版本均可

授权模式

FacePass采用客户端授权的模式，分为两种模式：

- 软件云授权：客户注册Face++账号、充值授权路数，并在出厂前通过客户端SDK调用授权接口进行激活。
- 硬件芯片授权：客户采购加密授权芯片并集成到客户端。

同时，如果采用抓拍模式或者云离线模式的，服务器端会需要插加密狗以进行设备授权。

实际使用过程中，请参考SDK文档，如有必要可以咨询商务或者售后支持人员。

感谢对FacePass的支持！