

天翼云人脸识别产品 用户操作手册

中国电信股份有限公司云计算分公司

目 录

1 产品介绍	1
1.1 产品简介	1
1.2 产品功能	1
1.3 使用场景	2
2 定价	2
人脸识别产品定价	2
3 快速入门	3
3.1 创建人脸识别应用	3
3.1.1 准备工作	3
3.1.2 创建人脸识别应用订单	4
3.1.3 订单确认	4
3.2 人脸识别用户控制台	6
4 API 文档	8
4.1 人脸检测接口说明	8
接口描述	8
接口要求	8

请求说明	8
请求示例	8
请求头说明	8
请求参数	9
请求代码示例	9
返回说明	10
API 调用成功返回值:	10
API 调用失败返回值	11
返回示例:	11
错误码说明	12
4.2 人脸属性识别接口	12
接口描述	12
接口要求	12
请求说明	12
请求示例	12
请求头说明	13
请求参数	13
请求代码示例	13

返回说明	15
API 调用成功返回值:	15
API 调用失败返回值	15
返回示例:	15
错误码说明	16
4.3 人脸比对接口说明	16
接口描述	17
接口要求	17
请求说明	17
请求示例	17
请求头说明	17
请求参数	17
请求代码示例	18
返回说明	19
API 调用成功返回值	19
API 调用失败返回值	19
返回示例:	20
错误码说明	20

4.4 人脸活体检测接口说明	20
接口描述	20
接口要求	21
请求说明	21
请求示例	21
请求头说明	21
请求参数	21
请求代码示例	22
返回说明	23
API 调用成功返回值:	23
API 调用失败返回值	24
返回示例:	24
错误码说明	25
4.5 是否戴口罩识别接口说明	26
接口描述	26
接口要求	26
请求说明	26

请求示例.....	26
请求头说明.....	26
请求参数.....	27
请求代码示例.....	27
返回说明.....	29
API 调用成功返回值:	29
API 调用失败返回值.....	29
返回示例:	29
错误码说明.....	30
5 常见问题.....	32

修订历史

版本号	日期	修订人	修订内容	修订原因
V1.0	2021.3		初版	

1 产品介绍

1.1 产品简介

人脸识别是中国电信云公司自研 AI 平台提供的产品之一，通过自主研发图像识别人工智能模型，提供人脸系列 API 服务-人脸检测、人脸属性识别、人脸比对、人脸活体检测与是否戴口罩识别。通过订购天翼云人脸识别产品，可将此服务快速高效的部署到您的应用中。人脸检测就是识别图片中是否有人脸，有哪些人脸。人脸属性识别主要是通过人脸推算人的性别，年龄等。人脸比对主要是判断两张人脸是否同一个人。人脸活体检测用于判断人脸图片是否为二次翻拍，过滤检测中不符合标准的人脸。是否戴口罩识别用于在人脸检测的基础上分析人员是否佩戴口罩。

天翼云用户通过订购以上人脸识别的应用，无需进行模型开发仅需 API 接口对接到 AI 上云，即可轻松、高效、准确的实现云上产品的人脸识别。人脸识别将高效帮助用户识别用户人脸信息，帮助用户在安防、人脸支付等领域实现智能化应用落地。

1.2 产品功能

- **人脸检测：**

通过用户输入人脸照片，高效快速地检测图像中的人脸，给出人脸的位置和可信度阈值，方便用户后续应用。

- **人脸属性识别：**

在人脸检测基础上，分析检测后的人脸性别、年龄属性，扩展用户使用人脸检测服务的应用场景。

- **人脸比对：**

通过对用户输入的两张人脸图像进行比对，分析是否为同一个人；同时可返回两张图像人脸的相似度以及比对结果等。

- **人脸活体检测：**

基于深度学习方法，分析人脸图像的摩尔纹、成像畸形等信息，实现静默活体判断，有效防止照片等非活体攻击。

- **是否戴口罩识别：**

准确高效识别图像中人脸是否正确佩戴口罩，支持多种类型口罩识别，并可对图片中多张人脸实时检测，有效应对疫情场景。

1.3 使用场景

- **身份核证**

通过人脸比对功能，可对被检测人进行身份核实，适用于楼宇门禁，车站等人证合一验证的场所使用。

- **人脸考勤**

人脸检测功能可助力企业系统对员工进行电子考勤，通过面部图像识别，可精准快捷进行日常考勤。

- **趣味应用**

通过人脸识别后的特征，和其它人脸比对，比如与名人类，娱乐类，动植物类等图像比对，输出趣味化的结果。

2 定价

人脸识别产品定价

人脸识别采用封顶资源包的方式订购：

产品规格	人脸检测定位、人脸属性识别价格 (元/年)	人脸比对价格 (元/年)
100万次/年	450	3100
1000万次/年	4100	30400
5000万次/年	17500	142500
1亿次/年	25000	255000

订购说明：

1、为满足客户不同业务使用量需求，每类 API 设置四档套餐，如：用户评估认为应用每年人脸属性识别的 API 调用大概为 500 万次，则可以购买五个 100 万次/年的调用次数包，若大概为 1000 万次，则可以购买一个 100 万次/年的调用次数包

2、人脸检测和人脸属性识别按照所检测的图像张数进行计数，人脸比对按照 API 调用次数进行收费（用户在比对时会指定两张照片，其中一张可能是系统中已存的，一张是用户新上传的，也有可能两张需比对的图像都是新上传的）

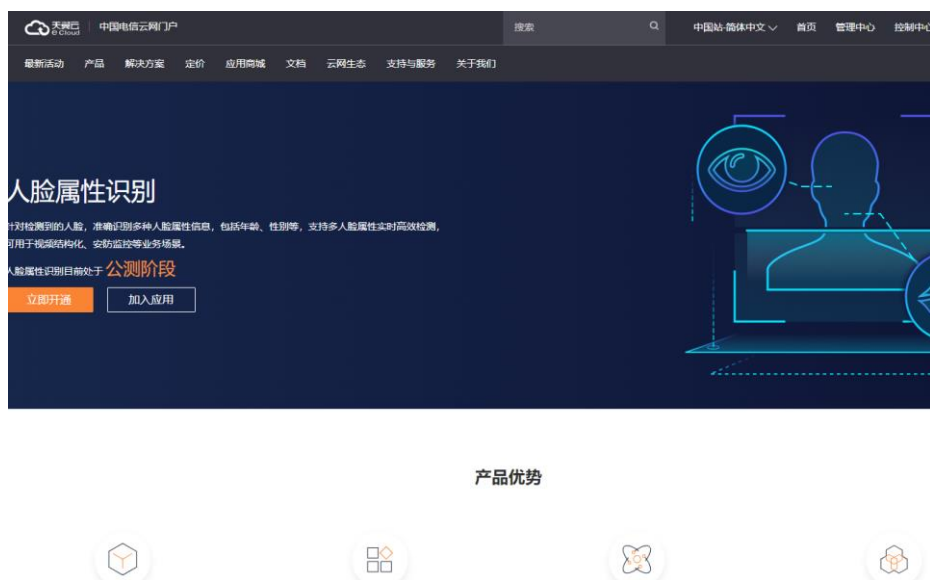
3. 用户预先购买 N 个固定额度的资源包，后续使用过程中产生的人脸检测或属性识别的张数，以及人脸识别的次数从资源包中抵扣，有效期内未使用的资源不会流转至下一年。有效期内资源包中的次数用完之后，用户无法再调用 AI 产品 API。对于有效期内未使用的资源包剩余调用次数，本产品不支持退订。

3 快速入门

3.1 创建人脸识别应用

3.1.1 准备工作

首先，用户使用天翼云网门户账号登录天翼云网门户后，点击菜单栏“产品”项，进入【大数据与AI】，即可看到人脸识别产品相应服务。以“人脸属性识别”为例，点击“人脸属性识别”名称即可进入“人脸属性识别”产品详情页。



点击【立即使用】按钮，进入人脸识别创建页面，点击“新的应用”



3.1.2 创建人脸识别应用订单

用户根据自身需要输入“应用名称”与“应用概述”。

应用名称：长度为 4-15 个字符。



新建应用

* 应用名称: 人脸属性识别应用场景 ①

应用概述: 人脸属性识别应用场景 ①

取消 确定

用户创建完应用后，点击“确定”和“下一步”，进入应用订购页面

我的能力

人脸识别	安全生产	图像识别	印刷文字识别	自然语言处理	内容安全
人脸属性识别					
API	状态	调用量限制	购买次数包	领取免费额度	
人脸属性识别	① 待付费	—	购买	立即领取	
人脸检测					
API	状态	调用量限制	购买次数包	领取免费额度	
人脸检测	① 待付费	—	购买	立即领取	
是否戴口罩识别					
API	状态	调用量限制	购买次数包	领取免费额度	
是否戴口罩识别	① 待付费	—	购买	立即领取	

3.1.3 订单确认

在“人脸属性识别”列点击“购买”，进入产品规格及数量配置页面。

人脸属性识别

选择配置

规格	公测截止时间	单价	购买数量
100万次	2020-11-30 23:59:59	¥0.00	- 1 +
1000万次	2020-11-30 23:59:59	¥0.00	- 0 +
5000万次	2020-11-30 23:59:59	¥0.00	- 1 +
10000万次	2020-11-30 23:59:59	¥0.00	- 0 +

配置清单

清空

规格	购买数量	价格
100万次	1	0
5000万次	1	0

费用合计

¥0元

配置费用 ¥ 0

参考价格，具体扣费请以账单为准。了解计费详情

下一步：确认配置

用户根据需选择要开通的资源包规格及数量，点击“下一步：确认配置”，进入配置确认界面。

配置

产品名称	计费方式	规格	公测截止时间	购买数量	单价
人脸属性识别	次数包	100万次	2020-11-30 23:59:59	1	¥0.00
人脸属性识别	次数包	5000万次	2020-11-30 23:59:59	1	¥0.00

协议

☐ 我已阅读并同意相关协议，[中国电信天翼云公测服务协议（自研人脸识别产品）](#)

配置费用 ¥ 0

参考价格，具体扣费请以账单为准。了解计费详情

上一步

立即购买

用户勾选服务协议，点击“立即购买”，进入订单支付页面

订单号：20201106185755918191 产品类型：包周期新购 创建时间：2020-11-06 18:57:55 更新时间：2020-11-06 18:57:55

产品1	配置	订购数量	规格来源池	周期	金额（元）
人脸识别	-	x 1	-	1次	0.00
人脸识别	-				0.00

费用合计：0.00元

重要提示：对云主机的使用请遵循国家相关法律法规之规定，对于违反相关法律法规的行为，服务商将关闭服务器，并视情况决定是否关闭用户帐号，停止所有服务，不退余款。弹性IP不支持降级操作，购买时请慎重选择。

- 支付订单前请务必确认所有资源已进行实名认证，如元，请点击[立即实名认证](#)
- 云市场订单不能使用优惠券和代金券支付
- 订单不支持同时使用代金券和优惠券

支付方式

预付款

账户余额支付（元）

无可用账户余额

代金券支付（元）

无可用代金券

优惠券

无可用优惠券

订单费用：+ 0.00元

账户余额支付：- 0.00元

代金券支付：- 0.00元

优惠券优惠：- 0.00元

需支付 0.00元

立即购买

用户确认订单费用是否准确，费用无误点击“立即支付”。完成支付后进入“支付结果”界面。

订单管理 - 云订单 > 订单详情 > 支付结果

支付成功 [返回订单列表](#)

订单号: 20201106185755918191 产品类型: 包周期新购 创建时间: 2020-11-06 18:57:55 更新时间: 2020-11-06 19:08:03

产品	配置	订购数量	所属资源池	周期	金额 (元)
人脸识别	-	x 1	-	1次	0.00
人脸识别	-				0.00

费用合计: 0.00元

订单费用: + 0.00元
 账户余额支付: - 0.00元
 代金券支付: - 0.00元
实际支付: 0.00元

3.2 人脸识别用户控制台

点击页面右上角“控制台”，进入控制中心。

天翼云 Cloud 中国电信云网门户 搜索 中国站-简体中文 首页 管理中心 控制台

天翼云诸葛AI平台 开放能力 AI开发中心 文档中心 支持与反馈 搜索 控制台 提供者中心 chenz@cf

总览

我的能力 我的应用 应用监控 应用告警

已开通能力

人脸识别 通用型OCR 工作版识别 人脸比对 图片鉴黄

能力调用排行榜 2020-11-05 ~ 2020-11-05 能力异常分布情况 2020-11-05 ~ 2020-11-05

排名 API组合包 调用次数(次)

用户点击“我的应用”，可以查看已经开通的应用。

我的应用

新增应用

默认应用 2020-07-28 已创建 开通 删除

人脸属性识别应用场景 2020-11-06 已创建 开通 删除

情感分析应用场景 2020-11-06 已创建 开通 删除

工作版识别应用场景1 2020-11-05 已创建 开通 删除

工作版识别应用场景 2020-11-05 已创建 开通 删除

人脸识别订购体验 2020-11-05 已创建 开通 删除

OCR文字识别体验 2020-11-05 已创建 开通 删除

通用OCR场景应用 2020-11-04 已创建 开通 删除

人脸识别检测测试 2020-11-04 已创建 开通 删除

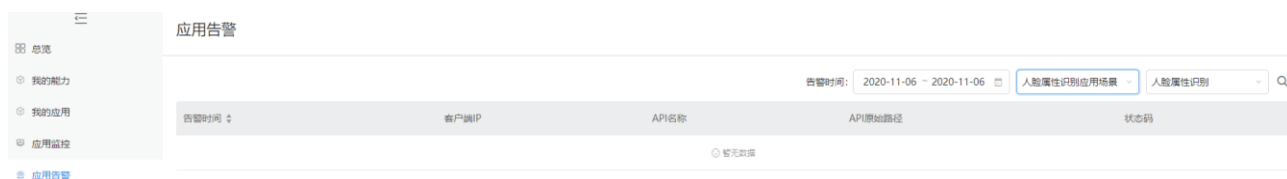
人脸识别检测应用0114 2020-11-04 已创建 开通 删除

人脸检测测试1104 2020-11-04 已创建 开通 删除

用户点击“应用监控”，可以查看所创建应用的请求次数、响应时间与请求流量。



用户点击“应用告警”，可以了解应用使用过程中系统所反馈的异常情况。



4 API 文档

4.1 人脸检测接口说明

接口描述

用于检测输入图像中的人脸，输出人脸位置坐标

接口要求

图片格式：

- 1. 图片单张大小不超过 2MB
- 2. 图片格式支持 jpg/png/bmp 等

请求说明

请求示例

HTTP 方法：POST

请求 URL：[https://ai.ctyun.cn/v1/aiop/api/\[系统生成的服务码\]/face/PERSON/person/detectFaceFromBase64](https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/face/PERSON/person/detectFaceFromBase64)

请求头说明

参数	值(说明)
Content-Type	application/json
Sign	MD5(AppKey + AppSecret + Timestamp + Path + Http Body), 32 位小写的 MD5 加密串

Timestamp	13 位时间戳
AppKey	平台分配给应用独立的 AppKey

请求参数

参数	是否 必选	类型	可选值 范围	说明
imageContent	true	String	-	传入图片的 base64 编码

请求代码示例

Bash:

```
curl -X POST "https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/face/PERSON/person/detectFaceFromBase64" -H "Content-Type: application/json" -H 'Sign: XXXXX' -H 'Timestamp: XXXXX' -H 'AppKey: XXXXX' --data '{"imageContent": "AAAAAAAAA...."}'
```

Python:

```
# -*- coding: utf-8 -*-
import requests
import base64
import hashlib
import time
import json

'''
人脸区域检测
'''

def md5(key):
    input_name = hashlib.md5()
    input_name.update(key.encode("utf-8"))
    return input_name.hexdigest()
```

```
request_url = "人脸区域检测的 API URL"

# 打开图片文件
f = open(r'[本地文件]', 'rb')
img_base64 = base64.encodebytes(f.read()).decode()
params = {"imageContent": img_base64}

app_key = '平台分配的 APP Key'
app_secret = '平台分配的 APP Secret'
time_stamp = str(int(round(time.time() * 1000)))

headers = {
    'content-type': 'application/json',
    'Sign': md5(app_key + app_secret + time_stamp + '/v1/aiop/api/[系统生成的服务码]/face/PERSON/person/detectFaceFromBase64' + json.dumps(params)),
    'Timestamp': time_stamp,
    'AppKey': app_key
}
response = requests.post(request_url, data=json.dumps(params), headers=headers)
if response:
    print(response.json())
```

返回说明

API 调用成功返回值：

返回字段	类型	字段说明
code	int	返回状态，返回 0 表示成功，返回错误代码参考 1.5 错误代码列表
message	string	返回对应的可读消息

result['face_num']	int	图片中人脸的数量
result['face_list']	list	每个人脸的详细信息
result['face_list'][0]['face_location']	dict	人脸所处的位置

API 调用失败返回值

返回字段	是否必选	类型	字段说明
code	true	int	错误码, 参见本文档错误码说明部分
message	true	string	返回对应的可读消息, 值为 error
details	true	string	错误消息

返回示例:

HTTP/1.1 200 OK

Content-Type: application/json

```
{
  "message" : "success",
  "code" : "0",
  "result":{
    "face_num" : 1,
    "face_list" : [{
      "face_location" : {
        "top" : 36,
        "left" : 48,
        "width" : 58,
        "height" : 72
      }
    }]
  }
}
```

错误码说明

错误代码	错误消息	错误可能原因
4101	图片为空	传入的参数为空
4102	请求参数格式错误	参数格式不满足要求，如请求体不是 json 格式等
4103	图片大小超过 2M	图片大小超过 2M
4104	图片解码失败	图片 base64 编码有误或者图片格式不支持
4105	未检测到人脸	上传图片中不包含人脸

4.2 人脸属性识别接口

接口描述

用于检测输入图像中的人脸年龄、性别等属性

接口要求

图片格式：

1. 图片单张大小不超过 2MB
2. 图片格式支持 jpg/png/bmp 等

请求说明

请求示例

HTTP 方法：POST

请求 URL：[https://ai.ctyun.cn/v1/aiop/api/\[系统生成的服务码\]/face/PERSON/person/detectAgeGende](https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/face/PERSON/person/detectAgeGende)

[rFromBase64](#)

请求头说明

参数	值(说明)
Content-Type	application/json
Sign	MD5(AppKey + AppSecret + Timestamp + Path + Http Body), 32 位小写的 MD5 加密串
Timestamp	13 位时间戳
AppKey	平台分配给应用独立的 AppKey

请求参数

参数	是否必选	类型	可选值范围	说明
imageContent	true	string	-	传入图片的 base64 编码

请求代码示例

Bash:

```
curl -X POST "https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/face/PERSON/person/detectAgeGenderFromBase64" -H "Content-Type: application/json" -H 'Sign: XXXXX' -H 'Timestamp: XXXXX' -H 'AppKey: XXXXX' --data '{"imageContent": "AAAAAAAAA...."}'
```

Python:

```
# -*- coding: utf-8 -*-
```

```
import requests
import base64
import hashlib
import time
import json
'''
人脸属性检测
'''

def md5(key):
    input_name = hashlib.md5()
    input_name.update(key.encode("utf-8"))
    return input_name.hexdigest()

request_url = "人脸属性检测的 API URL"
# 打开图片文件
f = open(r'[本地文件]', 'rb')
img_base64 = base64.encodebytes(f.read()).decode()
params = {"imageContent": img_base64}

app_key = '平台分配的 APP Key'
app_secret = '平台分配的 APP Secret'
time_stamp = str(int(round(time.time() * 1000)))

headers = {
    'content-type': 'application/json',
    'Sign': md5(app_key + app_secret + time_stamp + '/v1/aiop/api/[系统生成的服务码]/face/PERSON/person/detectAgeGenderFromBase64' + json.dumps(params)),
    'Timestamp': time_stamp,
    'AppKey': app_key
}

response = requests.post(request_url, data=json.dumps(params), headers=headers)
if response:
    print(response.json())
```

返回说明

API 调用成功返回值：

返回字段	类型	字段说明
code	int	返回状态，返回 0 表示成功
message	string	返回对应的可读消息
result['face_num']	int	图片中人脸的数量
result['face_list']	list	每个人脸的详细信息
result['face_list'][0]['face_location']	dict	人脸所处的位置
result['face_list'][0]['gender']	str	人脸的性别属性
result['face_list'][0]['age']	str	人脸的年龄属性

API 调用失败返回值

返回字段	是否必选	类型	字段说明
code	true	int	错误码，参见本文档错误码说明部分
message	true	string	返回对应的可读消息，值为 error
details	true	string	错误消息

返回示例：

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "message": "success",
```

```
"code" : "0",
"result":{
  "face_num" : 1,
  "face_list" : [{
    "face_location" : {
      "top" : 36,
      "left" : 48,
      "width" : 58,
      "height" : 72
    },
    "gender" : "Male",
    "age" : "23"
  }]
}
```

错误码说明

错误代码	错误消息	错误可能原因
4101	图片为空	传入的参数为空
4102	请求参数格式错误	参数格式不满足要求，如请求体不是 json 格式等
4103	图片大小超过 2M	图片大小超过 2M
4104	图片解码失败	图片 base64 编码有误或者图片格式不支持
4105	未检测到人脸	上传图片中不包含人脸

4.3 人脸比对接口说明

接口描述

用于检测输入的两张图像中的人脸相似度

接口要求

图片格式：

1. 图片单张大小不超过 2MB
2. 图片格式支持 jpg/png/bmp 等

请求说明

请求示例

HTTP 方法：POST

请求 URL：[https://ai.ctyun.cn/v1/aiop/api/\[系统生成的服务码\]/face/compare/PERSON/person/compareFromBase64](https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/face/compare/PERSON/person/compareFromBase64)

请求头说明

参数	值(说明)
Content-Type	application/json
Sign	MD5(AppKey + AppSecret + Timestamp + Path + Http Body), 32 位小写的 MD5 加密串
Timestamp	13 位时间戳
AppKey	平台分配给应用独立的 AppKey

请求参数

参数	是否必选	类型	可选值范围	说明
img1Base64	true	String	-	人脸比对的第一张图片

img2Base64	true	String	-	人脸比对的第二张图片
------------	------	--------	---	------------

请求代码示例

Bash:

```
curl -X POST "https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/face/compare/PERSON/person/compareFromBase64" -H "Content-Type: application/json" -H 'Sign: XXXXX' -H 'Timestamp: XXXXX' -H 'AppKey: XXXXX' --data '{
  "img1Base64": "AAAAAAAAAAAA....",
  "img2Base64": "AAAAAAAAAAAA...."
}'
```

Python:

```
# -*- coding: utf-8 -*-
import requests
import base64
import hashlib
import time
import json

'''
人脸比对
'''

def md5(key):
    input_name = hashlib.md5()
    input_name.update(key.encode("utf-8"))
    return input_name.hexdigest()

request_url = "人脸比对的 API URL"

# 打开图片文件
f1 = open(r'[本地文件]', 'rb')
img1_base64 = base64.encodebytes(f1.read()).decode()
f2 = open(r'[本地文件]', 'rb')
```

```
img2_base64 = base64.encodebytes(f2.read()).decode()

params = {"img1Base64": img1_base64, "img2Base64": img2_base64}

app_key = '平台分配的 APP Key'
app_secret = '平台分配的 APP Secret'
time_stamp = str(int(round(time.time() * 1000)))

headers = {
    'content-type': 'application/json',
    'Sign': md5(app_key + app_secret + time_stamp + '/v1/aiop/api/[系统生成的服务码]/face/compare/PERSON/person/compareFromBase64' + json.dumps(params)),
    'Timestamp': time_stamp,
    'AppKey': app_key
}

response = requests.post(request_url, data=json.dumps(params), headers=headers)
if response:
    print(response.json())
```

返回说明

API 调用成功返回值

返回字段	类型	字段说明
code	int	返回状态，返回 0 表示成功
message	string	返回对应的可读消息
result	float	两张人脸的相似度

API 调用失败返回值

返回字段	是否 必选	类型	字段说明
code	true	int	错误码，参见本文档错误码说明部分

message	true	string	返回对应的可读消息，值为 error
details	true	string	错误消息

返回示例：

HTTP/1.1 200 OK

Content-Type: application/json

```
{  
  "message" : "success",  
  "code" : "0",  
  "result": 0.99999  
}
```

错误码说明

错误代码	错误消息	错误可能原因
4101	图片为空	传入的参数为空
4102	请求参数格式错误	参数格式不满足要求，如请求体不是 json 格式等
4103	图片大小超过 2M	图片大小超过 2M
4104	图片解码失败	图片 base64 编码有误或者图片格式不支持
4105	未检测到人脸	上传图片中不包含人脸

4.4 人脸活体检测接口说明

接口描述

用于检测输入图像中的人脸是否为活体

接口要求

图片格式:

1. 图片单张大小不超过 2MB
2. 图片格式支持 jpg/png/bmp 等

请求说明

请求示例

HTTP 方法: POST

请求 URL: [https://ai.ctyun.cn/v1/aiop/api/\[系统生成的服务码\]/face-fas-action/person/detectFasFromBase64](https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/face-fas-action/person/detectFasFromBase64)

请求头说明

参数	值(说明)
Content-Type	application/json
Sign	MD5(AppKey + AppSecret + Timestamp + Path + Http Body), 32 位小写的 MD5 加密串
Timestamp	13 位时间戳
AppKey	平台分配给应用独立的 AppKey

请求参数

参数	是否必选	类型	可选值范围	说明
imageContent	true	string	-	传入图片的 base64 编码,

				图片使用常规的 base64 编码方式，编码后，不包含前缀，剔除前缀例如" data:image/jpeg;base64,"
--	--	--	--	--

请求代码示例

Bash:

```
curl -X POST "https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]face-fas-action/person/detectFasFromBase64" -H "Content-Type: application/json" -H 'Sign: XXXXX' -H 'Timestamp: XXXXX' -H 'AppKey: XXXXX' --data '{"imageContent":"AAAAAAAAA...."}'
```

Python:

```
# -*- coding: utf-8 -*-  
  
import requests  
  
import base64  
  
import hashlib  
  
import time  
  
import json  
  
'''  
人脸活体检测  
'''  
  
def md5(key):  
    input_name = hashlib.md5()  
    input_name.update(key.encode("utf-8"))  
    return input_name.hexdigest()
```

```
request_url = "人脸活体检测的 API URL"

# 打开图片文件
f = open(r'[本地文件]', 'rb')
img_base64 = base64.encodebytes(f.read()).decode()
params = {"imageContent": img_base64}

app_key = '平台分配的 APP Key'
app_secret = '平台分配的 APP Secret'
time_stamp = str(int(round(time.time() * 1000)))

headers = {
    'content-type': 'application/json',
    'Sign': md5(app_key + app_secret + time_stamp + '/v1/aiop/api/[系统生成的服务码]/face-fas-action/person/detectFasFromBase64' + json.dumps(params)),
    'Timestamp': time_stamp,
    'AppKey': app_key
}

response = requests.post(request_url, data=json.dumps(params), headers=headers)
if response:
    print(response.json())
```

返回说明

API 调用成功返回值:

返回字段	类型	字段说明
code	int	返回状态，返回 0 表示成功
message	string	返回对应的可读消息

<code>result['face_num']</code>	int	图片中人脸的数量
<code>result['face_list']</code>	list	每个人脸的详细信息
<code>result['face_list'][0]['face_location']</code>	dict	人脸所处的位置
<code>result['face_list'][0][FaceAntiSpoofing]</code>	string	活体人脸/非活体人脸/人脸清晰度差，无法判断

API 调用失败返回值

返回字段	是否必选	类型	字段说明
<code>code</code>	true	int	错误码，参见本文档错误码说明部分
<code>message</code>	true	string	返回对应的可读消息，值为 error
<code>details</code>	true	string	错误消息

返回示例：

HTTP/1.1 200 OK

Content-Type: application/json

```
{
  "message" : "success",
  "code" : "0",
  "result":{
    "face_num" : 1,
    "face_list" : [{
```



```

"face_location" : {
  "top" : 36,
  "left" : 48,
  "width" : 58,
  "height" : 72
},
"FaceAntiSpoofing" : "活体人脸/非活体人脸/人脸清晰度较差, 无法判断",
}
}
}

```

错误码说明

错误代码	错误消息	错误可能原因
4101	请求内容错误	传入内容为空, 或者传入的参数名错误
4102	请求参数格式错误	参数格式不满足要求, 如请求体不是 json 格式等
4103	图片大小超过 2M	图片大小超过 2M
4104	图片解码失败	图片为空, base64 编码有误, 或图片格式不支持
4105	未检测到人脸	上传图片中不包含人脸
5000	服务内部错误, 请联系管理员	接口服务出现未知错误, 请联系管理员解决

4.5 是否戴口罩识别接口说明

接口描述

用于检测输入图像中的人脸是否为活体

接口要求

图片格式：

1. 图片单张大小不超过 2MB
2. 图片格式支持 jpg/png/bmp 等

请求说明

请求示例

HTTP 方法：POST

请求 URL：[https://ai.ctyun.cn/v1/aiop/api/\[系统生成的服务码\]/face-fas-action/person/detectFasFromBase64](https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/face-fas-action/person/detectFasFromBase64)

请求头说明

参数	值(说明)
Content-Type	application/json
Sign	MD5(AppKey + AppSecret + Timestamp + Path + Http Body), 32 位小写的 MD5 加密串
Timestamp	13 位时间戳
AppKey	平台分配给应用独立的 AppKey

请求参数

参数	是否 必选	类 型	可选值 范围	说明
imageContent	true	string	-	传入图片的 base64 编码， 图片使用常规的 base64 编 码方式，编码后，不包含前 缀，剔除前缀例如" data:i mage/jpeg;base64,"

请求代码示例

Bash:

```
curl -X POST "https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]face-fas-action/person/detectFasFromBase64" -H "Content-Type: application/json" -H 'Sign: XXXXX' -H 'Timestamp: XXXXX' -H 'AppKey: XXXXX' --data '{"imageContent":"AAAAAAAAA...."}'
```

Python:

```
# -*- coding: utf-8 -*-
```

```
import requests
```

```
import base64
```

```
import hashlib
```

```
import time
```

```
import json
```

```
'''
```

```
人脸活体检测
```

```
'''  
  
def md5(key):  
    input_name = hashlib.md5()  
    input_name.update(key.encode("utf-8"))  
    return input_name.hexdigest()  
  
request_url = "人脸活体检测的 API URL"  
  
# 打开图片文件  
f = open(r'[本地文件]', 'rb')  
img_base64 = base64.encodebytes(f.read()).decode()  
params = {"imageContent": img_base64}  
  
app_key = '平台分配的 APP Key'  
app_secret = '平台分配的 APP Secret'  
time_stamp = str(int(round(time.time() * 1000)))  
  
headers = {  
    'content-type': 'application/json',  
    'Sign': md5(app_key + app_secret + time_stamp + '/v1/aiop/api/[系统生成的服务码]/face-fas-action/person  
/detectFasFromBase64' + json.dumps(params)),  
    'Timestamp': time_stamp,  
    'AppKey': app_key  
}  
  
response = requests.post(request_url, data=json.dumps(params), headers=headers)  
  
if response:  
    print(response.json())
```

返回说明

API 调用成功返回值：

返回字段	类型	字段说明
code	int	返回状态，返回 0 表示成功
message	string	返回对应的可读消息
result['face_num']	int	图片中人脸的数量
result['face_list']	list	每个人脸的详细信息
result['face_list'][0]['face_location']	dict	人脸所处的位置
result['face_list'][0]['FaceAntiSpoofing']	string	活体人脸/非活体人脸/人脸清晰度差，无法判断

API 调用失败返回值

返回字段	是否必选	类型	字段说明
code	true	int	错误码，参见本文档错误码说明部分
message	true	string	返回对应的可读消息，值为 error
details	true	string	错误消息

返回示例：

HTTP/1.1 200 OK

Content-Type: application/json

```
{
  "message" : "success",
  "code" : "0",
  "result":{
    "face_num" : 1,
    "face_list" : [{
      "face_location" : {
        "top" : 36,
        "left" : 48,
        "width" : 58,
        "height" : 72
      },
      "FaceAntiSpoofing" : "活体人脸/非活体人脸/人脸清晰度较差, 无法判断",
    }]
  }
}
```

错误码说明

错误代码	错误消息	错误可能原因
4101	请求内容错误	传入内容为空, 或者传入的参数名错误
4102	请求参数格式错误	参数格式不满足要求, 如请求体不是 json 格式等
4103	图片大小超过 2M	图片大小超过 2M
4104	图片解码失败	图片为空, base64 编码有误, 或图片

		格式不支持
4105	未检测到人脸	上传图片中不包含人脸
5000	服务内部错误，请联系管理员	接口服务出现未知错误，请联系管理员解决

5 常见问题

1. Q: AI 人脸识别有哪些 API 服务?

A: 有 5 个 API 服务: 人脸检测、人脸属性识别、人脸比对、人脸活体检测与是否戴口罩识别。

2. Q: AI 人脸识别对于上传的图片是否有要求?

A: (1) 图片单张大小不超过 2MB; (2) 图片格式支持 jpg、png、bmp 等

3. Q: 人脸识别是 HTTP GET 请求还是 HTTP POST 请求?

A: HTTP POST 请求。

4. Q: 人脸比对的 API 怎么用? 上传几张图片?

A: 人脸对比的 API 主要是对比两张图片中的人脸是否为同一人, 因此上传至少上传 2 张人脸图片 (即两张图片为一组, 可以为多组, 进行批量人脸的对比)

5. Q: 在哪里可以查询到请求 URL 中的“系统生成的服务码”? 比如人脸检测接口请求 URL:

`https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/face/PERSON/person/detectFaceFromBase64`

A: “系统生成的服务码”可在相应接口的产品详情页中获得, 以人脸检测接口为例。登录“人脸检测”产品详情页, 查看“能力详情”版块信息。



点击“能力详情”版块下的“API 接口”。调用地址中红框标出的内容即为“系统生成的服务码”。

