

你的颜值有几分(二)





1. 探索新知

1.1

人脸检测准备工作

想要实现今天的效果，将用到【百度智能云】平台人脸检测的接口



1. 注册百度智能云账号



给大家介绍准备工作的详细过程

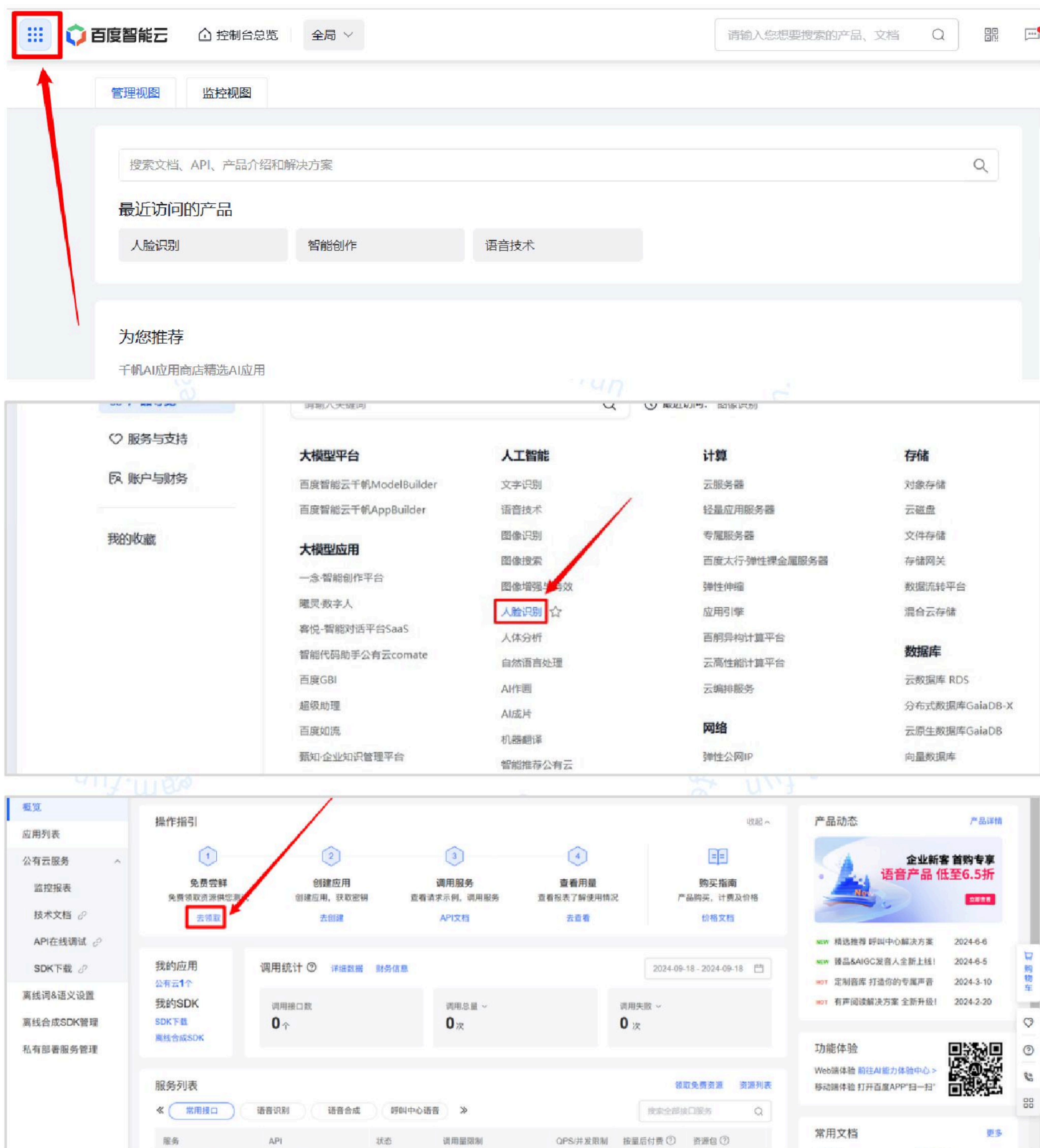


2. 打开产品—【人脸识别】

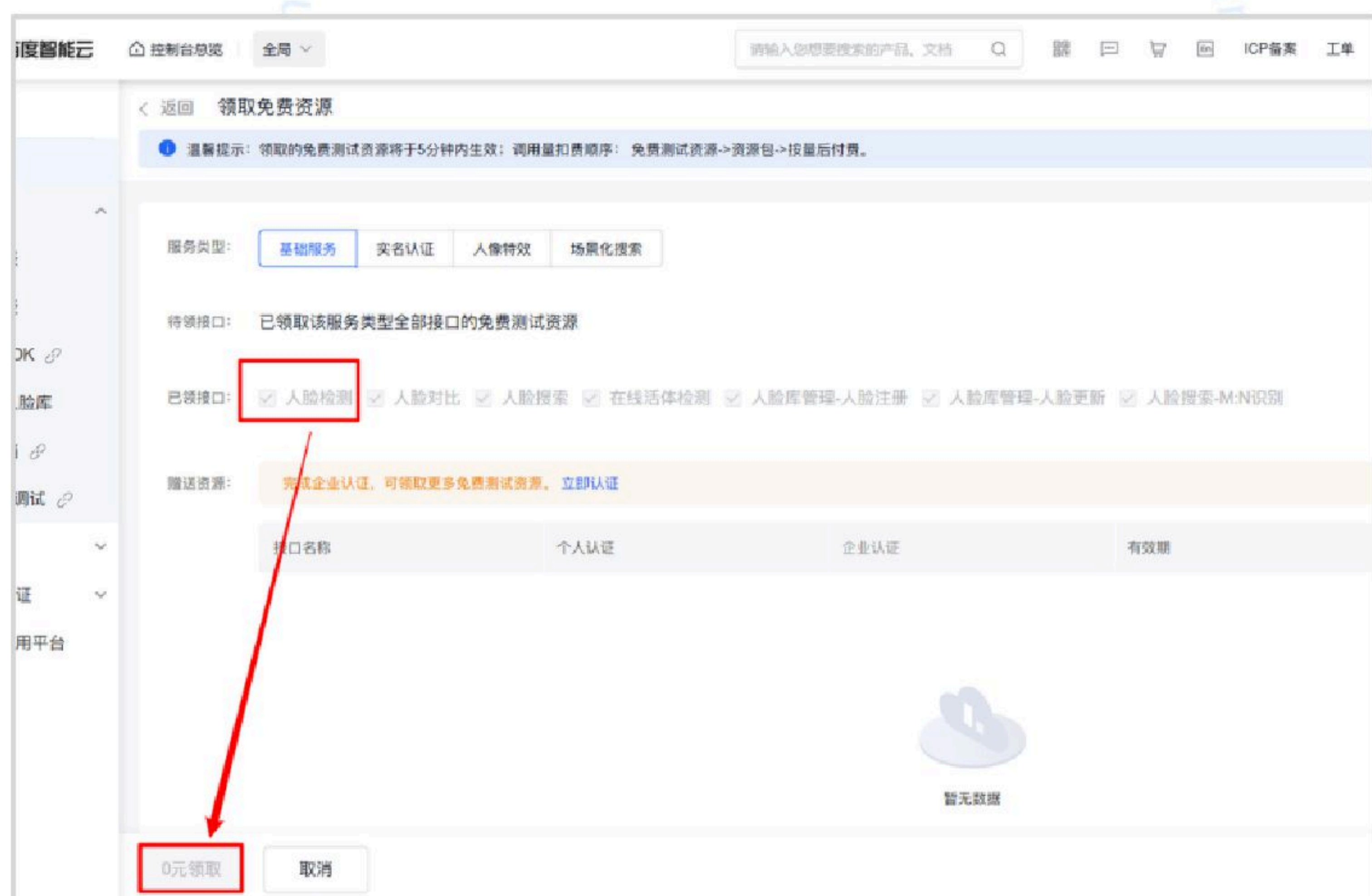


1.1

人脸检测准备工作



3. 领取免费接口

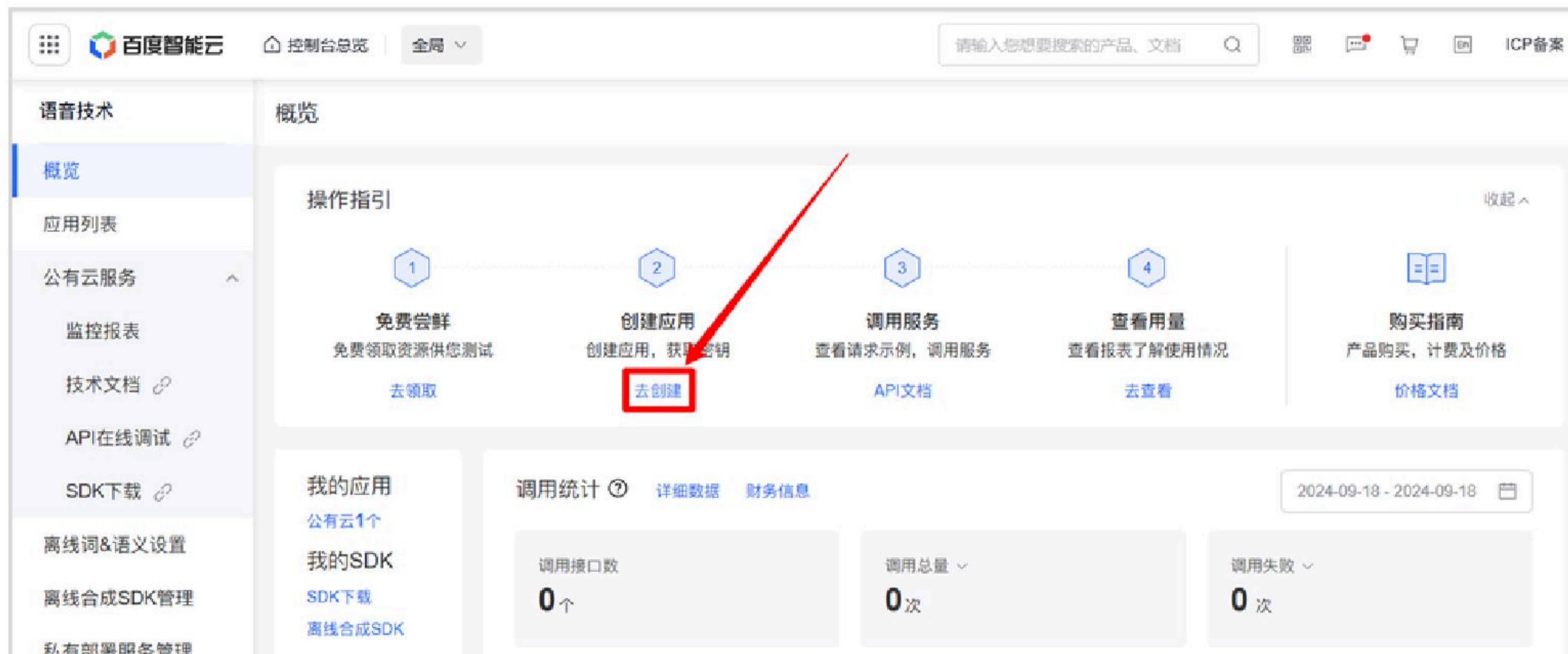


我已经领取过接口，所以两接口会在【已领接口】中。大家在操作时，选择这个接口，也可以直接领取【全部】接口，点击左下方的【0元领取】，即可领取试用（试用时长180天）。

1.1

人脸检测准备工作

4. 创建应用



注意:

在创建应用之前，请确认自己已成功领取免费测试资源



根据业务需求，填写如下应用信息：

【应用名称】 必填项，所创建的应用名称，支持中英文、数字、下划线及中横线

【接口选择】 勾选该应用需要用到的接口

【应用归属】 必选项，选择【个人】即可

【应用描述】 必填项，对此应用的业务场景进行简单描述

5. 获得应用的AK/SK



1.1

人脸检测准备工作

- 安装需要的第三方库



```
代码区 安装第三方库.py
1 # 点击运行【安装第三方库.py】，即可安装本课用到的第三方库
2
3 import pip
4 pip.main(['install','baidu-aip'])
5 pip.main(['install','chardet'])
```

1. 【baidu-aip】百度人工智能开放平台提供的Python SDK（软件开发工具包），用于连接并调用百度AI开放平台的各种API接口
2. 【chardet】用于检测文本文件或字符串的字符编码的Python库

1.2

点击开测

【配置人脸识别服务接口】

1. 引入所需要的库

```
'''
【一、导入库】
'''
...
from aip import AipFace
```

2. 定义常量

```
'''
【二、初始化】
'''
...
APP_ID = '你的 AppID '
API_KEY = '你的 API Key '
SECRET_KEY = '你的 Secret Key '
```

3. 初始化 AipFace 对象

```
APP_ID = '你的 AppID '
API_KEY = '你的 API Key '
SECRET_KEY = '你的 Secret Key '
client = AipFace(APP_ID, API_KEY, SECRET_KEY)
```

【开始颜值检测】



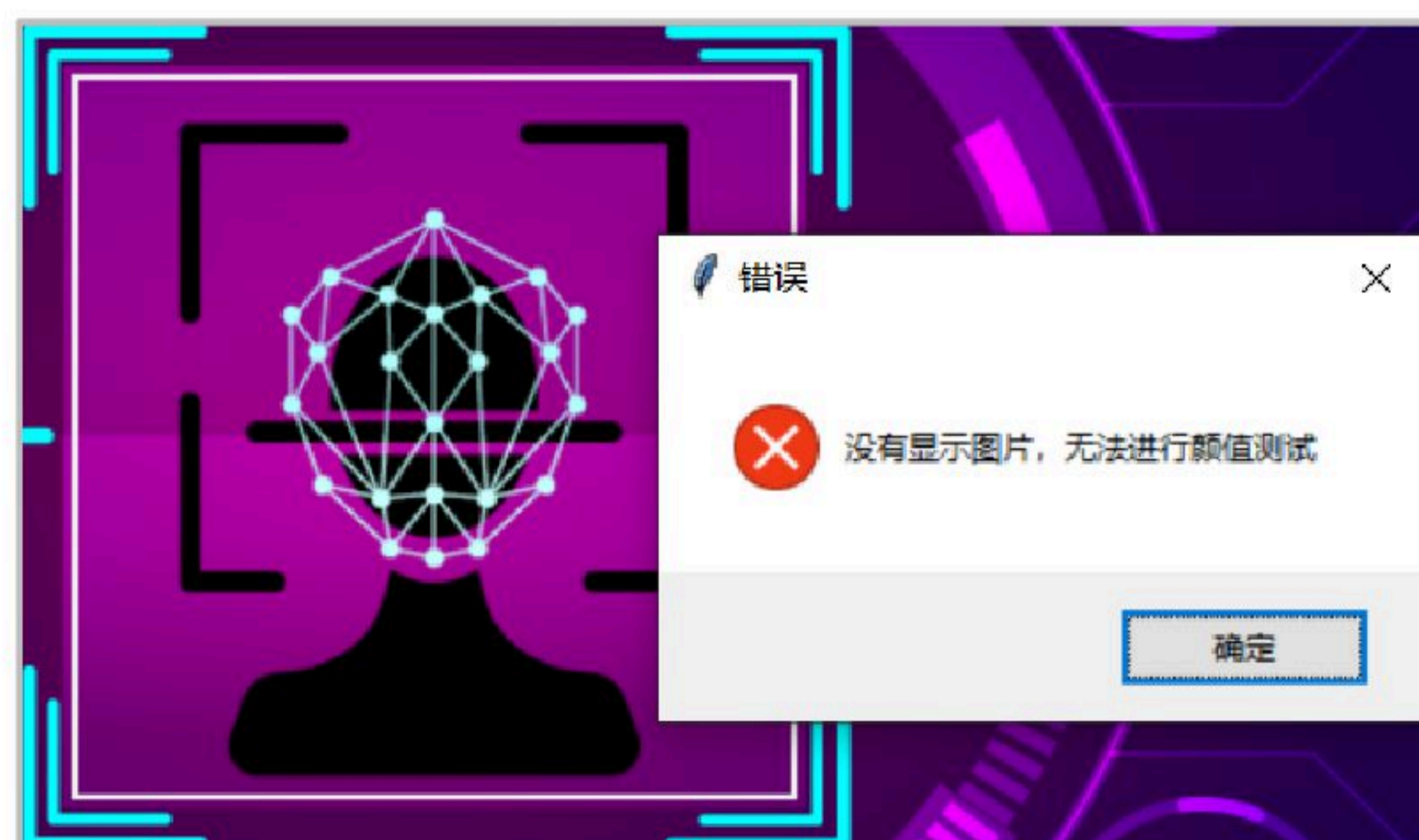
上节课已经定义好了颜值检测的空函数 — ok函数，接下来我们填充这个函数，让它实现颜值检测的功能

```
# 颜值检测开始
def ok():
    # 编写代码
```

1. 判断图片是否上传

```
# 判断图片是否上传
if not hasattr(lab_photo, 'imgtk'):
    messagebox.showerror("错误", "没有显示图片，无法进行颜值测试")
    return
```

这段代码的作用是检测一个名为 `lab_photo` 的对象是否具有属性 `imgtk`，如果该属性不存在，则显示一个错误信息框并返回，阻止进一步的操作。



注意：别忘了导入要用到的模块

```
from tkinter import filedialog, messagebox
```

hasattr(object, name)函数

- 作用

检查一个对象是否具有指定的属性

- 参数

object: 需要检查的对象

name: 一个字符串, 表示需要检查的属性名

- 返回值

如果对象有指定的属性, 返回 True ; 否则返回 False

2. 传入参数, 调用人脸检测API

```
if not hasattr(lab_photo, 'imgtk'):
    ...
img_base64 = img_to_bytes(lab_photo.imgtk) # 将图片转为Base64
imageType = "BASE64" # 设置图片类型
options = {"face_field": "age,gender,face_shape,emotion,beauty"}
result = client.detect(img_base64, imageType, options)
```

这段代码是准备好要传入的参数, 调用人脸检测API。

```
img_base64 = img_to_bytes(lab_photo.imgtk)
```

将 lab_photo 部件中的图片对象转为 BASE64 类型, 进行转换的函数已经预置好 — `img_to_bytes(photoimg)` 函数, 直接使用即可

```
imageType = "BASE64" # 设置图片类型
```

将图片类型设置为“BASE64”

```
options = {"face_field": "age,gender,face_shape,emotion,beauty"}
```

设置 `face_field` 参数, 添加返回的属性:

1.2

点击开测

age (年龄)	
gender (性别)	male:男性、female:女性
face_shape (脸型)	square:正方形、triangle:三角形、oval:椭圆、heart:心形、round:圆形
emotion (情绪)	angry:愤怒、disgust:厌恶、fear:恐惧、happy:高兴、sad:伤心、surprise:惊讶、neutral:无表情、pouty:撅嘴、grimace:鬼脸
beauty (颜值)	美丑打分，范围0-100，越大表示越美。 face_fields包含beauty时返回

```
result = client.detect(img_base64, imageType, options)
```

调用人脸检测 API，并传入 Base64 编码的图像数据、图像类型和选项。
client.detect 是对某个面部检测 API 的调用，result 保存了 API 返回的结果。

3. 未检测到人脸



若请求错误，返回值将包含以下参数：
error_code: 错误码
error_msg: 错误描述信息，帮助理解和解决发生的错误

```
...
result = client.detect(img_base64, imageType, options)
if 'error_msg' in result and result['error_msg'] == 'pic not has face':
    messagebox.showerror("错误", "图像中未检测到人脸")
return
```

1.2

点击开测

4. 返回结果的处理, 输出

正常运行 `print(result)` 结果如下:

```
{'error_code': 0, 'error_msg': 'SUCCESS', 'log_id': 3038836016, 'timestamp': 1728974383, 'cached': 0, 'result': {'face_num': 1, 'face_list': [{'face_token': 'aa5ee4664b75c117c04b053d02adel58', 'location': {'left': 111.59, 'top': 65.35, 'width': 39, 'height': 48, 'rotation': -24}, 'face_probability': 1, 'angle': {'yaw': -5.69, 'pitch': 14.61, 'roll': -19.36}, 'age': 26, 'beauty': 61.58, 'gender': {'type': 'female', 'probability': 0.99}, 'face_shape': {'type': 'heart', 'probability': 0.47}, 'emotion': {'type': 'fear', 'probability': 0.5178618431}}]}}
```

`'result'` in result `'face_list'` in result['result']

```
if 'error_msg' in result and result['error_msg'] == 'pic not has face':  
    ...  
if 'result' in result and 'face_list' in result['result']:  
    # 解析结果  
    # 构造结果字符串  
    # 使用对话框显示结果  
else:  
    messagebox.showerror("错误", "检测失败, 请重试")
```

【解析结果】

解析结果提取关键属性:

```
{'error_code': 0, 'error_msg': 'SUCCESS', 'log_id': 3038836016, 'timestamp': 1728974383, 'cached': 0, 'result': {'face_num': 1, 'face_list': [{'face_token': 'aa5ee4664b75c117c04b053d02adel58', 'location': {'left': 111.59, 'top': 65.35, 'width': 39, 'height': 48, 'rotation': -24}, 'face_probability': 1, 'angle': {'yaw': -5.69, 'pitch': 14.61, 'roll': -19.36}, 'age': 26, 'beauty': 61.58, 'gender': {'type': 'female', 'probability': 0.99}, 'face_shape': {'type': 'heart', 'probability': 0.47}, 'emotion': {'type': 'fear', 'probability': 0.5178618431}}]}}
```

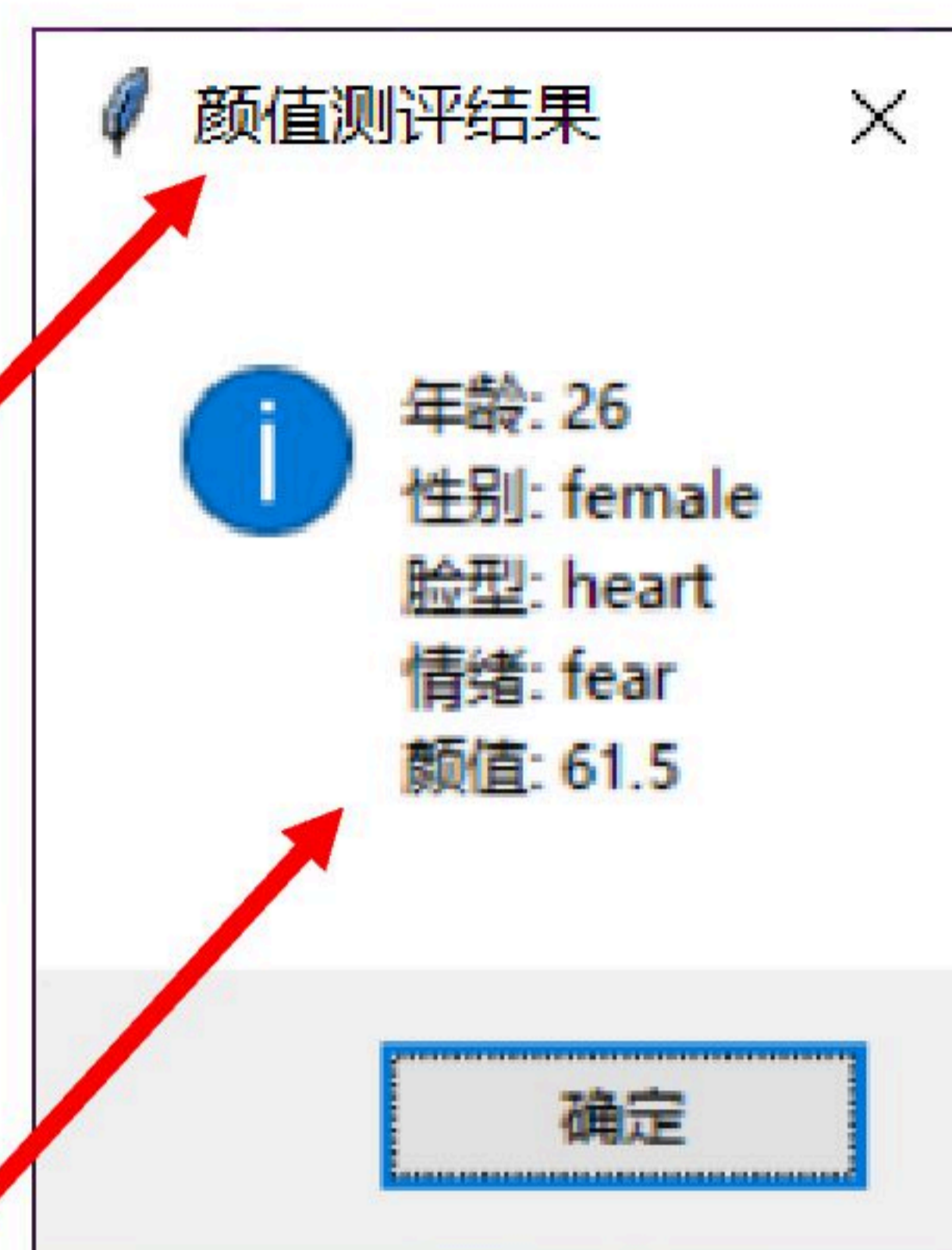
解析结果

```
age = result['result']['face_list'][0]['age']  
gender = result['result']['face_list'][0]['gender']['type']  
face_shape = result['result']['face_list'][0]['face_shape']['type']  
emotion = result['result']['face_list'][0]['emotion']['type']  
beauty = result['result']['face_list'][0]['beauty']
```

1.2

点击开测

【构造结果字符串、对话框显示结果】



构造结果字符串

```
output_str=f"年龄:{age}\n性别:{gender}\n脸型:{face_shape}\n情绪:{emotion}\n颜值:{beauty}"
```

使用对话框显示结果

```
messagebox.showinfo("颜值测评结果", output_str)
```

完整代码

```
...
【导入库】
...
# 【预置内容, 请勿改动】
import io
import base64
import cv2
import tkinter as tk
from PIL import Image, ImageTk
from tkinter import filedialog, messagebox
from aip import AipFace
...
【初始化】 (变量初始化、窗口初始化、API配置信息)
...
# 【预置内容, 请勿改动】
cap = None
imgtk = None
update_id = None
# 初始化Tkinter
root = tk.Tk()
root.geometry('700x700')
root.title('颜值测评')
root.resizable(0, 0)
# 百度AI的API配置信息
APP_ID = '你的 AppID '
API_KEY = '你的 API Key '
SECRET_KEY = '你的 Secret Key '
client = AipFace(APP_ID, API_KEY, SECRET_KEY)
...
【函数】
...
# 【预置内容, 请勿改动】 图像 --> 字节数据 --> Base64 编码格式
def img_to_bytes(photoimg):
    ...
# 【预置内容, 请勿改动】 打开摄像头
def open_camera():
    ...
# 【预置内容, 请勿改动】 关闭摄像头
def close_camera():
    ...
# 加载本地图片
def load():
    ...
```

完整代码

```
def ok(): # 颜值检测开始
    # 判断 lab_photo 对象是否具有属性 imgtk
    if not hasattr(lab_photo, 'imgtk'):
        messagebox.showerror("错误", "没有显示图片, 无法进行颜值测试")
        return
    img_base64 = img_to_bytes(lab_photo.imgtk) # 将图片转为Base64
    imageType = "BASE64" # 设置图片类型
    options = {"face_field": "age,gender,face_shape,emotion,beauty"}
    result = client.detect(img_base64, imageType, options)
    if 'error_msg' in result and result['error_msg'] == 'pic not has
face':
        messagebox.showerror("错误", "图像中未检测到人脸")
        return
    if 'result' in result and 'face_list' in result['result']:
        # 解析结果
        age = result['result']['face_list'][0]['age']
        gender = result['result']['face_list'][0]['gender']['type']
        face_shape =
result['result']['face_list'][0]['face_shape']['type']
        emotion = result['result']['face_list'][0]['emotion']['type']
        beauty = result['result']['face_list'][0]['beauty']
        # 构造结果字符串
        output_str=f"年龄:{age}\n性别:{gender}\n脸型:{face_shape}\n情
绪:{emotion}\n颜值:{beauty}"
        # 使用对话框显示结果
        messagebox.showinfo("颜值测评结果", output_str)
    else:
        messagebox.showerror("错误", "检测失败, 请重试")
'''
【组件配置】
'''
# 设置背景图
# 设置缩略图
# 本地上传按钮
# 拍照上传按钮
# 开始测评按钮

'''
【启动事件循环】
'''
root.mainloop()
```



2. 强化练习

1. 如果API返回错误信息'pic not has face', 程序会怎么做? ()

- A. 显示年龄结果
- B. 显示错误消息
- C. 重新上传图片
- D. 提供美颜功能

2. 如果成功检测到人脸, gender可能的值有哪些? ()

- A. 蓝色或绿色
- B. 男性或女性
- C. 成年或未成年
- D. 快乐或悲伤

3. 如果检测成功, 会通过哪个函数呈现最终结果? ()

- A. print()
- B. messagebox.showerror()
- C. messagebox.showinfo()
- D. open()

4. 已知二维列表 gz 中存储了某单位部门部分人员的工资数据, 如下所示:

`gz=[['a1', '李明', 6200], ['a2', '吴海', 5600], ['a3', '殷小飞', 6536]]`, 那么想要查询‘殷小飞’的工资情况可使用? ()

- A. `gz(3)`
- B. `gz[3]`
- C. `gz(2,2)`
- D. `gz[2][2]`



2. 强化练习

5. 执行下面Python代码后，有关说法错误的是？（ ）

```
a = int(input())
b = int(input())
if a and b:
    print("1")
elif not (a or b):
    print("2")
elif a or b:
    print("3")
else:
    print("4")
```

- A. 如果先后输入1和1，则将输出1
- B. 如果先后输入0和1或者1和0，则将输出3
- C. 如果先后输入0和0，则将输出2
- D. 存在一组a和b的输入，输出是4

3. 术语箱

age	年龄	emotion	情绪
gender	性别	beauty	美丽, 魅力
face_shape	脸型		

4. 课后挑战

增加检测项目

任务描述:

在课堂作品的基础上, 增加以下的检测项目, 并在对话框显示出来

glasses (是否戴眼镜)	none: 无眼镜, common: 普通眼镜, sun: 墨镜
mask (是否戴口罩)	没戴口罩/戴口罩, 取值0或1 0: 没戴口罩、1: 代表戴口罩
face_type (人脸类型)	human: 真实人脸、cartoon: 卡通人脸

