图像识别

SDK 参考

文档版本 10

发布日期 2020-03-25

华为技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： https://www.huawei.com

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

目 录

1 图像识别 SDK 简介 1
2 申请服务 2
3 获取认证信息 3
4 获取图像识别 SDK 4
5 准备环境 5
6 使用 SDK（Java） 6
6.1 Java 开发环境配置 6
6.2 Eclipse 安装与 SDK 工程导入 6
6.3 图像标签示例 8
6.4 名人识别示例 9
6.5 翻拍识别示例 10
6.6 Token 认证方式使用 SDK 11
7 使用 SDK（Java）（废弃） 12
7.1 低光照增强示例 12
7.2 图像去雾示例 13
7.3 超分图像重建示例 14
7.4 视频背景音乐识别示例 15
8 使用 SDK（Python） 17
8.1 Python 开发环境配置 17
8.2 Python 环境 SDK 导入 18
8.3 图像标签示例 18
9 修订记录 21

1 图像识别 SDK 简介

图像识别概述

图像识别（Image Recognition），是指利用计算机对图像进行分析和理解，以识别各种不同模式的目标和对象的技术，包括图像标签，名人识别等。

图像识别以开放API（Application Programming Interface，应用程序编程接口）的方式提供给用户，用户通过实时访问和调用API获取推理结果，帮助用户自动采集关键数据，打造智能化业务系统，提升业务效率。

SDK 概述

图像识别软件开发工具包（Image Recognition Software Development Kit，简称Image SDK）是对图像识别提供的REST API进行的封装，以简化用户的开发工作。用户直接调用Image SDK提供的接口函数即可实现使用图像识别业务能力的目的。

接口与 API 对应关系

图像识别接口与API对应关系请参见**表1-1**。

表 1-1 接口与 API 对应关系表

接口	API
图像标签	POST /v1.0/image/tagging
名人识别	POST /v1.0/image/celebrity-recognition
翻拍识别	POST /v1.0/image/recapture-detect

2 申请服务

申请图像识别服务的具体操作步骤请参见《 图像识别API参考 》的“**如何调用API > 申请服务**”章节。

3 获取认证信息

使用服务API需要进行认证，有Token和AK/SK两种认证方式可选，推荐用户使用AK/SK方式。

步骤1 注册并登录华为云管理控制台。

步骤2 鼠标移动至用户名处，在下拉列表中单击“我的凭证”。

步骤3 选择“访问密钥”页签，点击“新增访问秘钥”按钮。

步骤4 通过邮箱或者手机进行验证，输入对应的验证码。

步骤5 单击“确定”，下载认证账号的AK/SK，AK/SK数据会以本地文件的形式保存，请妥善保管。

----结束

4 获取图像识别 SDK

下载图像识别SDK软件包和文档，请参见：**图像识别SDK**。

获取图像识别Endpoint，请参见：**地区和终端节点**。

5 准备环境

在使用图像识别SDK时，需要准备的环境，以JAVA语言SDK为示例请参见表5-1。

表 5-1 开发环境

准备项	说明
操作系统	Windows系统，推荐Windows 7及以上版本。
安装JDK	Java开发环境的基本配置。版本要求：强烈推荐使用1.8版本。

6 使用 SDK（Java）

6.1 Java 开发环境配置

本SDK包要求的JDK版本必须高于JDK8版本，以下步骤以win7环境配置JDK8 64位为例，若已经下载JDK并配置好环境请忽略本章节。

步骤1 下载**JDK文件**。

步骤2 下载完成后按照提示安装，位置自选，比如安装到本地C:\Program Files\Java\jdk1.8.0_131。

步骤3 配置Java环境变量：右键“计算机>属性>高级系统设置>环境变量”，进行如下操作。

1. 新建系统变量JAVA_HOME，变量值为实际JDK安装位置。
2. 在Path中添加%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin （注意用英文分号分隔）。
3. 新建系统变量CLASSPATH，变量值为%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar 。

步骤4 打开命令行窗口，输入“java -version”，显示如**图6-1**表示配置成功。

图 6-1 Java 版本信息

```
C:\>java -version
java version "1.8.0_131"
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
```

----结束

6.2 Eclipse 安装与 SDK 工程导入

以eclipse工具为例给出SDK工程导入步骤，如果使用其他IDE工具，请参照如下处理：

步骤1 **下载对应平台的eclipse版本**，例如：eclipse-jee-mars-R-win32-x86_64.zip。

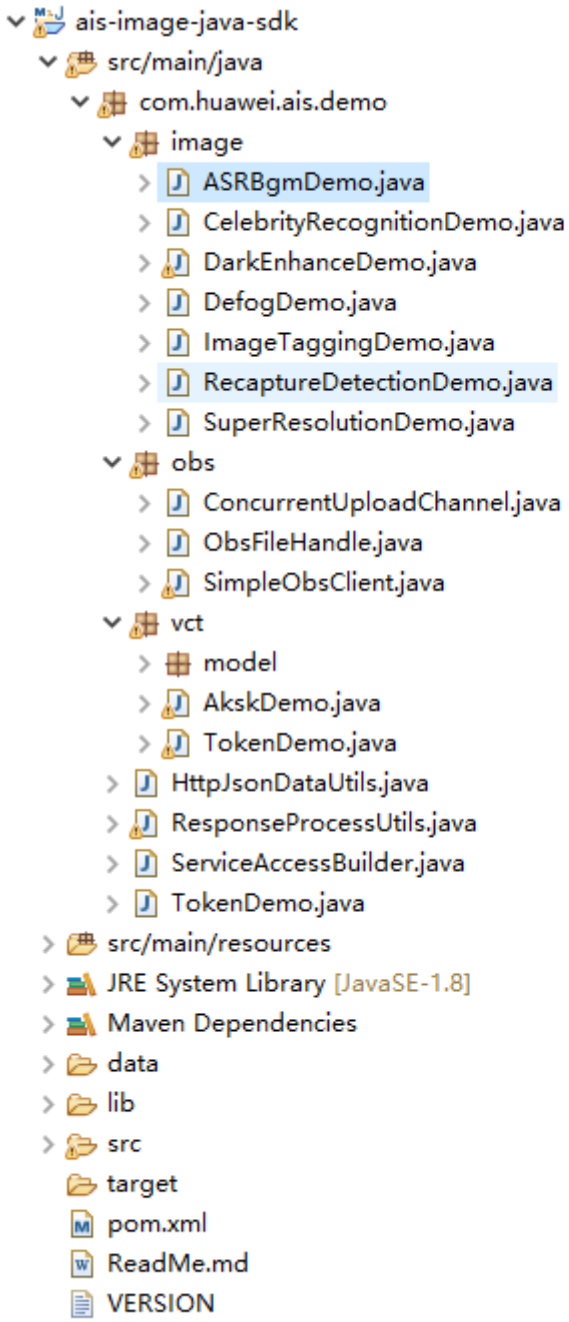
步骤2 解压后直接打开eclipse。

步骤3 确保“Windows>Preferences>Java>Installed JREs”配置正确的JRE路径。

步骤4 在左侧“Package Explorer”页面右键，单击“Import”，选择“Maven>Existing Maven Projects”，单击“Next”，单击“Browse”，选择ais-image-java-sdk所在的本地位置。

步骤5 单击Finish，导入SDK，导入后打开工程，工程目录如图1所示。

图 6-2 工程目录



在使用maven构建时，SDK包中lib目录下的jar包为项目依赖的本地jar包，其他依赖可由pom.xml文件配置，通过中央仓库或配置为华为开源镜像站获取。

----结束

6.3 图像标签示例

图像识别服务认证方式有Token和AK/SK两种方式，本章节对AK/SK方式使用SDK进行示例说明。

该图像标签Demo示例对应URI：POST /v1.0/image/tagging。将AK/SK信息替换为实际AK/SK后，即可运行体验Demo。

步骤1 在ImageTaggingDemo.java文件中配置用户AK/SK。示例代码如下：

```
// 1. 图片标签服务的的基本信息,生成对应的一个客户端连接对象
AisAccess service = ServiceAccessBuilder.builder()
        .ak("######")                       // your ak
        .sk("######")                       // your sk
        .region("cn-north-1")               // 图像识别服务华北-北京一(cn-north-1)的配置
        .connectionTimeout(5000)            // 连接目标url超时限制
        .connectionRequestTimeout(1000)     // 连接池获取可用连接超时限制
        .socketTimeout(20000)               // 获取服务器响应数据超时限制
        .build();
```

步骤2 选择本地图片或者使用Demo默认图片，参考如下示例代码修改ImageTaggingDemo.java文件中图片路径（"*data/image-tagging-demo-1.jpg*"）。

```
//
// 2.构建访问图片标签服务需要的参数
//
String uri = "/v1.0/image/tagging";
byte[] fileData = FileUtils.readFileToByteArray(new File("data/image-tagging-demo-1.jpg"));
String fileBase64Str = Base64.encodeBase64String(fileData);
```

步骤3 执行ImageTaggingDemo.java文件，控制台输出200即表示程序执行成功。**图像标签识别结果**输出到控制台，如**图1 运行结果**所示。

图 6-3 运行结果


```
"D:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
200
{
    "result":{
        "tags":[
            {
                "confidence":"100.0",
                "tag":"Koala",
                "type":"object",
                "i18n_tag":{
                    "en":"Koala",
                    "zh":"考拉"
                }
            },
            {
                "confidence":"100.0",
                "tag":"Marsupials",
                "type":"object",
                "i18n_tag":{
                    "en":"Marsupials",
                    "zh":"有袋目"
                }
            }
        ]
    }
}
```


----结束

6.4 名人识别示例

图像识别服务认证方式有Token和AK/SK两种方式，本章节对AK/SK方式使用SDK进行示例说明。

该名人识别Demo示例对应URI：POST /v1.0/image/celebrity-recognition。将AK/SK信息替换为实际AK/SK后，即可运行体验Demo。

步骤1 在CelebrityRecognitionDemo.java文件中配置用户AK/SK。示例代码如下：

```
// 1. 名人识别服务的的基本信息,生成对应的一个客户端连接对象
AisAccess service = ServiceAccessBuilder.builder()
        .ak("######")                    // your ak
        .sk("######")                    // your sk
        .region("cn-north-1")             // 图像识别服务华北-北京一(cn-north-1)的配置
        .connectionTimeout(5000)          // 连接目标url超时限制
        .connectionRequestTimeout(1000)     // 连接池获取可用连接超时限制
        .socketTimeout(20000)             // 获取服务器响应数据超时限制
        .build();
```

步骤2 选择一张明星或者网络红人的人像图片，参考如下示例代码修改CelebrityRecognitionDemo.java文件中图片路径（"*data/celebrity-recognition.jpg*"）。

```
//
// 2.构建访问名人识别服务需要的参数
//
String uri = "/v1.0/image/celebrity-recognition";
/**
* TODO 运行此样例时可在data目录放入待识别的人像图片，此处仅用一张人像图片表明设置文件名的位置
*/
```

```
byte[] fileData = FileUtils.readFileToByteArray(new File("data/celebrity-recognition.jpg"));
String fileBase64Str = Base64.encodeBase64String(fileData);
```

步骤3 执行CelebrityRecognitionDemo.java文件，控制台输出200即表示程序执行成功。**名人识别结果**输出到控制台，如图 **运行结果**所示

图 6-4 运行结果

```
"D:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
200
{
    "result":[
        {
            "confidence":0.9989172992292055,
            "face_detail":{
                "w":365,
                "h":514,
                "x":260,
                "y":198
            },
            "label":"迈克尔·杰克逊"
        }
    ]
}
```

----结束

6.5 翻拍识别示例

图像识别服务认证方式有Token和AK/SK两种方式，本章节对AK/SK方式使用SDK进行示例说明。

该翻拍识别Demo示例对应URI：POST /v1.0/image/recapture-detect。将AK/SK信息替换为实际AK/SK后，即可运行体验Demo。

步骤1 在RecaptureDetectionDemo.java文件中配置用户AK/SK。示例代码如下：

```
// 1. 翻拍识别服务的基本信息,生成对应的一个客户端连接对象
AisAccess service = ServiceAccessBuilder.builder()
        .ak("######")                    // your ak
        .sk("######")                    // your sk
        .region("cn-north-1")             // 图像识别服务华北-北京一(cn-north-1)的配置
        .connectionTimeout(5000)          // 连接目标url超时限制
        .connectionRequestTimeout(1000)   // 连接池获取可用连接超时限制
        .socketTimeout(20000)             // 获取服务器响应数据超时限制
        .build();
```

步骤2 选择本地图片或者使用Demo默认图片，参考如下示例代码修改RecaptureDetectionDemo.java文件中图片路径（"*data/recapture-detect-demo-1.jpg*"）。

```
//
// 2.构建访问翻拍识别服务需要的参数
//
String uri = "/v1.0/image/recapture-detect";
byte[] fileData = FileUtils.readFileToByteArray(new File("data/recapture-detect-demo-1.jpg"));
String fileBase64Str = Base64.encodeBase64String(fileData);
```

步骤3 执行RecaptureDetectionDemo.java文件，控制台输出200即表示程序执行成功。**翻拍识别结果**输出到控制台，如**图6-5**所示。

图 6-5 运行结果

```
"C:\Program Files\Java\jdk1.8.0_112\bin\java" ...
200
{"result":{"suggestion":"false","category":"recapture","score":"1.0","detail":[{"label":"recapture","confidence":"1.0"}]}}
```

----结束

6.6 Token 认证方式使用 SDK

图像识别服务认证方式有Token和AK/SK两种方式，在本节Demo示例中使用Token认证方式。

（用户参照如下步骤操作后，即可体验图像标签服务，用户修改用户名和密码后，无需另外编写代码）

步骤1 打开com.huawei.ais.demo包下面的TokenDemo.java文件，修改main函数中的username和password为系统中实际注册的用户名和密码，示例代码如下：

```
/**
 * 调用主入口函数
 */
public static void main(String[] args) throws URISyntaxException, UnsupportedOperationException,
IOException {
    String username = "zhangshan";   // 此处，请输入用户名
    String password = " *******";     // 此处，请输入对应用户名的密码
    String domainName = "*******";   // 账户为主账号，此处输入与用户名一致，若为子账号，此处输入为主账户用户名
    String token = getToken(username, password, domainName, projectName);
    System.out.println(token);
    requestImageTaggingBase64(token, "data/image-tagging-demo-1.jpg");
    }
```

步骤2 修改后可直接运行TokenDemo.java，在控制台可看到使用Token方式图像标签服务的识别结果。

----结束

7 使用 SDK（Java）（废弃）

7.1 低光照增强示例

图像识别服务认证方式有Token和AK/SK两种方式，本章节对AK/SK方式使用SDK进行示例说明。

该低光照增强Demo示例对应URI：POST /v1.0/vision/dark-enhance。将AK/SK信息替换为实际AK/SK后，即可运行体验Demo。

步骤1 在DarkEnhanceDemo.java文件中配置用户AK/SK。示例代码如下：

```
// 1. 配置好低光照增强服务的的基本信息,生成对应的一个客户端连接对象
AisAccess service = ServiceAccessBuilder.builder()
        .ak("######")                    // your ak
        .sk("######")                    // your sk
        .region("cn-north-1")             // 图像识别服务华北-北京一(cn-north-1)的配置
        .connectionTimeout(5000)            // 连接目标url超时限制
        .connectionRequestTimeout(1000)     // 连接池获取可用连接超时限制
        .socketTimeout(20000)               // 获取服务器响应数据超时限制
        .build();
```

步骤2 选择本地图片或者使用Demo默认图片，参考如下示例代码修改DarkEnhanceDemo.java文件中原图片路径（"*data/dark-enhance-demo-1.bmp*"）和处理后生成图像保存路径（"*data/dark-enhance-demo-1.cooked.bmp*"）。

```
//
// 2.构建访问低光照增强服务需要的参数
//
String uri = "/v1.0/vision/dark-enhance";
byte[] fileData = FileUtils.readFileToByteArray(new File("data/dark-enhance-demo-1.bmp"));
String fileBase64Str = Base64.encodeBase64String(fileData);

JSONObject json = new JSONObject();
json.put("image", fileBase64Str);
json.put("brightness", 0.9);

// 3.传入低光照增强服务对应的uri参数, 传入低光照增强服务需要的参数，
// 该参数主要通过JSON对象的方式传入, 使用POST方法调用服务
HttpResponse response = service.post(uri, json.toJSONString());

// 4.验证服务调用返回的状态是否成功，如果为200, 为成功, 否则失败。
ResponseProcessUtils.processResponseStatus(response);

// 5.处理服务返回的字符流，生成对应的低光照增强处理后对应的图片文件。
ResponseProcessUtils.processResponseWithImage(response, "data/dark-enhance-demo-1.cooked.bmp");
```


步骤3 执行DarkEnhanceDemo.java文件，**低光照增强识别结果**输出到控制台。控制台输出200即表示程序执行成功。

低光照增强的原图与处理效果图对比如**图7-1**所示。

图 7-1 原图与处理效果图对比

----结束

7.2 图像去雾示例

图像识别服务认证方式有Token和AK/SK两种方式，本章节对AK/SK方式使用SDK进行示例说明。

该图像去雾Demo示例对应URI：POST /v1.0/vision/defog。将AK/SK信息替换为实际AK/SK后，即可运行体验Demo。

步骤1 在DefogDemo.java文件中配置用户AK/SK。示例代码如下：

```
// 1. 配置好去雾服务的的基本信息,生成对应的一个客户端连接对象
AisAccess service = ServiceAccessBuilder.builder()
        .ak("######")                    // your ak
        .sk("######")                    // your sk
        .region("cn-north-1")             // 图像识别服务华北-北京一(cn-north-1)的配置
        .connectionTimeout(5000)            // 连接目标url超时限制
        .connectionRequestTimeout(1000)     // 连接池获取可用连接超时限制
        .socketTimeout(20000)               // 获取服务器响应数据超时限制
        .build();
```

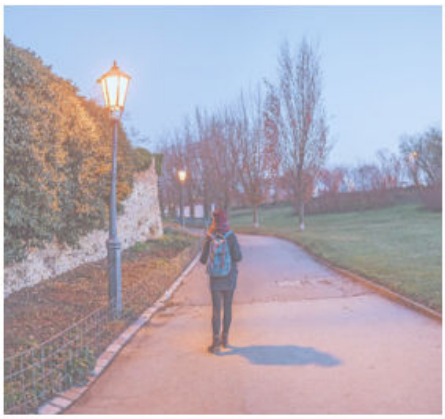
步骤2 选择本地图片或者使用Demo默认图片，参考如下示例代码修改DefogDemo.java文件中原图片路径（"*data/defog-demo-1.png*"）和处理后生成图像保存路径（"*data/defog-demo-1.cooked.png*"）。

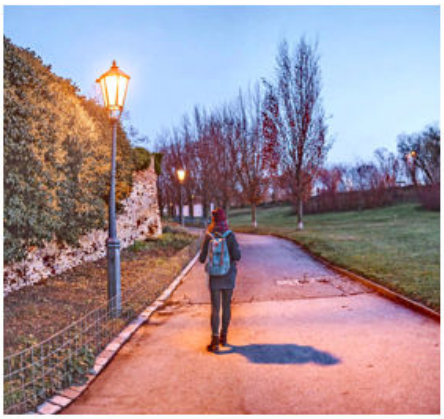
```
//
// 2.构建访问去雾服务需要的参数
//
String uri = "/v1.0/vision/defog";
byte[] fileData = FileUtils.readFileToByteArray(new File("data/defog-demo-1.png"));
String fileBase64Str = Base64.encodeBase64String(fileData);

JSONObject json = new JSONObject();
json.put("image", fileBase64Str);
json.put("gamma", 1.5);
json.put("natural_look", true);

// 3.传入去雾服务对应的uri参数, 传入去雾服务需要的参数，
```

```
// 该参数主要通过JSON对象的方式传入, 使用POST方法调用服务
HttpResponse response = service.post(uri, json.toJSONString());

// 4.验证服务调用返回的状态是否成功，如果为200, 为成功, 否则失败。
ResponseProcessUtils.processResponseStatus(response);

// 5.处理服务返回的字符流，生成对应的去雾处理后对应的图片文件。
ResponseProcessUtils.processResponseWithImage(response, "data/defog-demo-1.cooked.png");
```

步骤3 执行DefogDemo.java文件，**图像去雾识别结果**输出到控制台，控制台输出200即表示程序执行成功。图像去雾的原图与处理效果图对比如**图7-2**。

图 7-2 原图与处理效果图对比





----结束

7.3 超分图像重建示例

图像识别服务认证方式有Token和AK/SK两种方式，本章节对AK/SK方式使用SDK进行示例说明。

该超分图像重建Demo示例对应URI：POST /v1.0/vision/super-resolution。将AK/SK信息替换为实际AK/SK后，即可运行体验Demo。

步骤1 在SuperResolutionDemo文件中配置用户AK/SK。示例代码如下：

```
// 1. 配置好超分图像重建服务的的基本信息,生成对应的一个客户端连接对象
AisAccess service = ServiceAccessBuilder.builder()
        .ak("######")                    // your ak
        .sk("######")                    // your sk
        .region("cn-north-1")             // 图像识别服务华北-北京一(cn-north-1)的配置
        .connectionTimeout(5000)            // 连接目标url超时限制
        .connectionRequestTimeout(1000)     // 连接池获取可用连接超时限制
        .socketTimeout(20000)               // 获取服务器响应数据超时限制
        .build();
```

步骤2 选择本地图片或者使用Demo默认图片，参考如下示例代码修改SuperResolutionDemo.java文件中原图片路径（"*data/super-resolution-demo-1.png*"）和处理后生成图像保存路径（"*data/super-resolution-demo-1.cooked.png*"）。

```
//
// 2.构建访问超分图像重建服务需要的参数
//
String uri = "/v1.0/vision/super-resolution";
byte[] fileData = FileUtils.readFileToByteArray(new File("data/super-resolution-demo-1.png"));
String fileBase64Str = Base64.encodeBase64String(fileData);
```

```
JSONObject json = new JSONObject();
json.put("image", fileBase64Str);
json.put("scale", 3);
json.put("model", "ESPCN");

// 3.传入超分图像重建服务对应的uri参数, 传入超分图像重建服务需要的参数，
// 该参数主要通过JSON对象的方式传入, 使用POST方法调用服务
HttpResponse response = service.post(uri, json.toJSONString());

// 4.验证服务调用返回的状态是否成功，如果为200, 为成功, 否则失败。
ResponseProcessUtils.processResponseStatus(response);

// 5.处理服务返回的字符流，生成对应的超分图像重建处理后对应的图片文件。
ResponseProcessUtils.processResponseWithImage(response, "data/super-resolution-demo-1.cooked.png");
```

步骤3 执行SuperResolutionDemo.java文件，**超分图像重建结果**输出控制台，控制台输出200即表示程序执行成功。

超分图像重建的原图与处理效果图对比如 **图1**。

图 7-3 原图与处理效果图对比


----结束

7.4 视频背景音乐识别示例

图像识别服务认证方式有Token和AK/SK两种方式，本章节对AK/SK方式使用SDK进行示例说明。

视频背景音乐识别Demo示例对应URI：POST /v1.0/bgm/recognition。将AK/SK信息替换为实际AK/SK后，即可运行体验Demo。

步骤1 在ASRBgmDemo.java文件中配置用户AK/SK。示例代码如下：

```
// 1. 视频背景音乐识别服务的的基本信息,生成对应的一个客户端连接对象
AisAccess service = ServiceAccessBuilder.builder()
```



```
    .ak("######")                    // your ak
    .sk("######")                    // your sk
    .region("cn-north-1")             // 图像识别服务华北-北京一(cn-north-1)的配置
    .connectionTimeout(5000)            // 连接目标url超时限制
    .connectionRequestTimeout(1000)     // 连接池获取可用连接超时限制
    .socketTimeout(20000)               // 获取服务器响应数据超时限制
    .build();
```

步骤2 将ASRBgmDemo.java文件中视频的URL修改为OBS中视频的URL（"*https://obs-test-llg.obs.cn-north-1.myhuaweicloud.com/bgm_recognition*";），示例代码如下：

```
JSONObject json = new JSONObject();

// 视频的OBS URL（注：obs链接需要和region区域一致，不同的region的obs资源不共享）
String url = "https://obs-test-llg.obs.cn-north-1.myhuaweicloud.com/bgm_recognition";
json.put("url", url);

// 3.传入视频背景音乐识别服务对应的uri参数, 传入视频背景音乐识别服务需要的参数，
// 该参数主要通过JSON对象的方式传入, 使用POST方法调用服务
HttpResponse response = service.post(uri, json.toJSONString());

// 4.验证服务调用返回的状态是否成功，如果为200, 为成功, 否则失败。
ResponseProcessUtils.processResponseStatus(response);
```

说明

需要传入obs链接的服务，链接需要和所在region区域一致，不同的region的obs资源不共享。

步骤3 执行ASRBgmDemo.java文件，控制台输出200即表示程序执行成功。视频背景音乐识别结果输出到控制台，如图7-4所示。

图 7-4 运行结果

```
"D:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
200
{
  "result": {
    "audio_name": "It Gets Better"
  }
}

Process finished with exit code 0
```

----结束

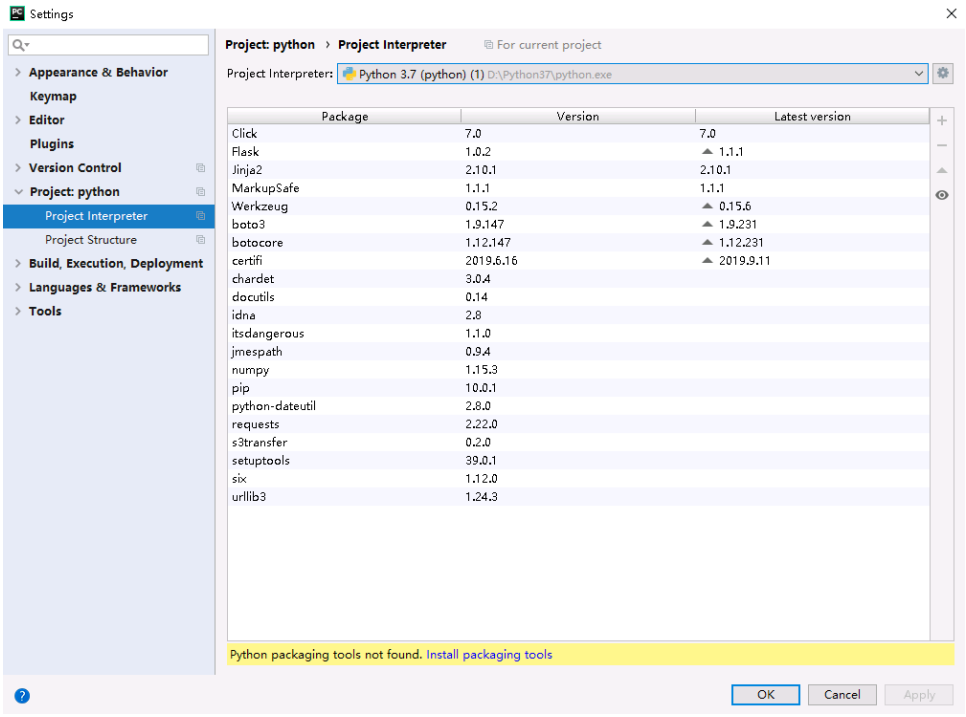
8 使用 SDK（Python）

8.1 Python 开发环境配置

使用图像识别Python版本SDK包，需要您配置Python开发环境。

1. 从**Python官网**下载并安装合适的Python版本。兼容Python2.6+以及Python3.x。推荐使用Python3.x版本，如下以Python3.7 版本为例进行说明。
2. 从**PyCharm官网**下载并安装最新版本。
3. 在PyCharm开发工具中配置Python环境，在菜单依次选择“File > Settings > Project Interpreter”。
4. 在页面上方选择您的Python安装路径，如图 **PyCharm配置python环境所示**。选择好目标Python之后单击页面下方“Apply”完成配置。

图 8-1 PyCharm 配置 python 环境



8.2 Python 环境 SDK 导入

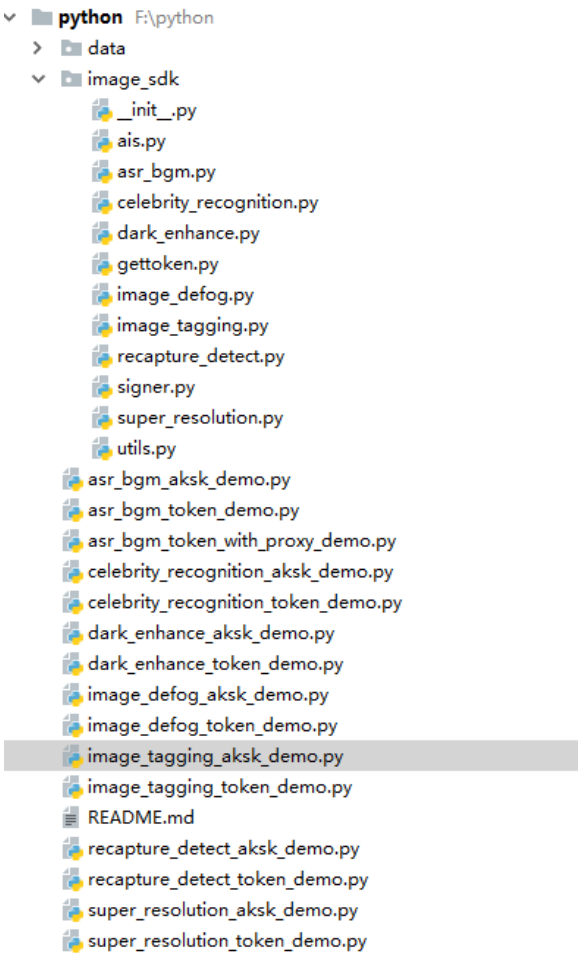
1. 在**图像识别SDK**，选择图像识别 Python SDK工具包下载并解压。

   解压包结构和内容说明如下：

```
├──── python
│    ├──── data                          #示例图片
│    ├──── image_sdk                     #图像识别sdk代码
│    │    ├──── ais.py                   #图像识别使用常量配置
│    │    ├──── image_tagging.py         #图像识别-图像标签sdk代码
│    │    ├──── signer.py
│    ├──── image_tagging_aksk_demo.py    #图像标签ak/sk方式demo
│    ├──── image_tagging_token_demo.py   #图像标签token方式demo
```

   2. 在PyCharm界面，单击“File > Open File or Project ”，选择解压后SDK工具包的存放路径，导入SDK。如**图8-2**所示。

图 8-2 导入后工程目录



8.3 图像标签示例

本服务认证方式有两种方式。本章节以图像标签为例，分别使用**AK/SK**和**Token**认证方式进行调用。图像标签对应URI：POST /v1.0/image/tagging。

AK/SK 认证方式使用 SDK

1. 在“image_tagging_aksk_demo.py”文件中配置“app_key”，“app_secret”。示例代码如下：

```
if __name__ == '__main__':
    # Services currently support North China-Beijing(cn-north-4)
    init_global_env('cn-north-4')
    #
    # access image tagging,post data by ak,sk
    #
    app_key = '*************'
    app_secret = '*************'
```

2. 图像标签支持**调用文件**和**调用URL**两种调用方式。需要您在“image_tagging_aksk_demo.py”文件中修改图片文件的本地路径或URL路径。
   - 调用文件，需修改图片的本地路径。将“encode_to_base64”的“data/image-tagging-demo.jpg”替换为需要识别的图片路径。示例代码如下：

     ```
     # call interface use the file
     result = image_tagging_aksk(app_key, app_secret, encode_to_base64('data/image-tagging-demo.jpg'), '', 'zh', 5, 60)
     print(result)
     ```

   - 调用URL，需修改图片URL路径。将“demo_data_url”的图片URL路径替换为需要识别的图片URL路径。示例代码如下：

     ```
     demo_data_url = 'https://sdk-obs-source-save.obs.cn-north-4.myhuaweicloud.com/tagging-normal.jpg'
     # call interface use the url
     result = image_tagging_aksk(app_key, app_secret, '', demo_data_url, 'zh', 5, 30)
     print(result)
     ```

3. 运行“image_tagging_aksk_demo.py”，识别结果输出到控制台表示执行成功。

```
{"result":{"tags":[{"confidence":"98.38","i18n_tag":{"en":"Person","zh":"人"},"tag":"人","type":"object"},
{"confidence":"97.12","i18n_tag":{"en":"Children","zh":"儿童"},"tag":"儿童","type":"object"},
{"confidence":"96.39","i18n_tag":{"en":"Sandbox","zh":"(供儿童玩的)沙坑"},"tag":"(供儿童玩的)沙坑
","type":"scene"},{"confidence":"89.28","i18n_tag":{"en":"Play","zh":"玩耍"},"tag":"玩耍
","type":"object"},{"confidence":"87.99","i18n_tag":{"en":"Toy","zh":"玩具"},"tag":"玩具
","type":"object"}]}}

Process finished with exit code 0
```

Token 认证方式使用 SDK

1. 在“image_tagging_aksk_demo.py”文件中配置实际注册的用户名“user_name”和密码“password”，非IAM登录用户user_name和account_name一致。示例代码如下：

```
if __name__ == '__main__':

    # Services currently support North China-Beijing(cn-north-4)
    init_global_env('cn-north-4')

    #
    # access image tagging,post data by token
    #
    user_name = '******'
    password = '******'
    account_name = '******'  # the same as user_name in commonly use

    token = get_token(user_name, password, account_name)
```

2. 图像标签支持**调用文件**和**调用URL**两种调用方式。需要在“image_tagging_aksk_demo.py”文件中修改图片文件的本地路径或URL路径。
   - 调用文件，需修改图片的本地路径。将“encode_to_base64”的“data/image-tagging-demo.jpg”替换为需要识别的图片路径。示例代码如下：

     ```
     # call interface use the file
     result = image_tagging(token, encode_to_base64('data/image-tagging-demo.jpg'), '', 'zh', 5,
     ```

```
60)
    print(result)
```

- 调用URL，需修改图片URL路径。将“demo_data_url”的图片URL路径替换为需要识别的图片URL路径。示例代码如下：

```
    demo_data_url = 'https://sdk-obs-source-save.obs.cn-north-4.myhuaweicloud.com/tagging-normal.jpg'
    # call interface use the url
    result = image_tagging(token, '', demo_data_url, 'zh', 5, 30)
    print(result)
```

3. 运行“image_tagging_aksk_demo.py”，识别结果输出到控制台表示执行成功。

```
{"result":{"tags":[{"confidence":"98.38","i18n_tag":{"en":"Person","zh":"人"},"tag":"人","type":"object"},
{"confidence":"97.12","i18n_tag":{"en":"Children","zh":"儿童"},"tag":"儿童","type":"object"},
{"confidence":"96.39","i18n_tag":{"en":"Sandbox","zh":"(供儿童玩的)沙坑"},"tag":"(供儿童玩的)沙坑
","type":"scene"},{"confidence":"89.28","i18n_tag":{"en":"Play","zh":"玩耍"},"tag":"玩耍
","type":"object"},{"confidence":"87.99","i18n_tag":{"en":"Toy","zh":"玩具"},"tag":"玩具
","type":"object"}]}}

Process finished with exit code 0
```


9 修订记录

发布日期	修订说明
2020-05-29	第十一次正式发布。 • 使用SDK章节拆分为**6-使用SDK（Java）**章节和**7-使用SDK（Java）(废弃）**章节。
2019-05-15	第十次正式发布。 修改 • **获取认证信息**章节。 • **使用SDK（Java）**章节。
2019-01-31	第九次正式发布。 修改 • 将图像识别对应Endpoint修改为“image.cn-north-1.myhuaweicloud.com”。
2018-12-14	第八次正式发布。
2018-11-15	第七次正式发布。 新增 • **图像识别SDK简介**。 • **准备环境**。
2018-10-30	第六次正式发布。 新增 • **名人识别示例**。
2018-09-19	第五次正式发布。 修改 • 申请服务操作步骤参见文档为《图像识别API参考》。
2018-07-16	第四次正式发布。 修改 • **Eclipse安装与SDK工程导入**，修改工程导入路径。

发布日期	修订说明
2018-07-02	第三次正式发布。 新增 • **低光照增强示例**。 • **图像去雾示例**。 • **超分图像重建示例**。
2018-04-20	第二次正式发布。 修改 • **申请服务**的获取信息。
2018-01-03	第一次正式发布。