

SKEye-Linux-SDK 说明文档

1. 概述

[SKEye](#) 是[盛开互动](#)依托十余年研发积淀推出的图像识别技术平台，致力于为智能产品、应用提供精准快速的“云+端”图像识别 SDK。SKEyeSDK 是一套基于 SKEye 人工智能开放平台的图像识别服务，该文档将详细说明如何在 Linux(Ubuntu 64 位)平台上快速集成该 SDK，并通过调用 SDK 接口使用 SKEye 图像识别服务。

2. 系统需求

Linux(Ubuntu 64 位) 16.04 及以上版本系统

3. 使用步骤

支持使用 G++ 进行集成使用，在新建工程后，将 .so(.a) 和 .h 文件拷贝至对应目录，具体使用可参考 Demo 工程。

4. Demo 说明

案例展示如何调用 SKEyeSDK 接口，图像识别结果会打印到控制台上，用户只需要将结果数据拷贝到自己申请好的内存中(不需要释放结果数据空间)。

5. 关键参数说明

ApiKey: char 数组类型，指当前 APP 所申请的 SKEye 的 API_KEY

ApiSecret: char 数组类型，指当前 APP 所申请的 SKEye 的 API_SECRET

ServiceName: char 数组类型，指 SKEye 识别服务名称

ImageId: int 类型，图片 id(方便确认请求顺序和返回顺序是否一致)

ImageUrl: char 数组类型，指图片的网络 url/本地路径

ImageData: unsigned char * 类型，指 BGR、倒立、无行对齐，图片数据

ImageWidth: unsigned int 类型，指图片的宽度

ImageHeight: unsigned int 类型，指图片的高度

CallBackImageResultFunction: void CallBackImageResult * 类型，指图片结果回调。

6. 接口说明

6.1 SDK 初始化

```
void SKEyeSDK_Init(char *ApiKey, char *ApiSecret);
```

参数:

- ApiKey: 当前 APP 所申请的 SKEye 的 API_KEY
- ApiSecret: 当前 APP 所申请的 SKEye 的 API_SECRET

示例代码:

```
char ApiKey[] = "466da4221010f8341912da500b4a23b";
char ApiSecret[] = "ddc8bd498e7c29c929862ab7391a7b89";
SKEyeSDK_Init(ApiKey, ApiSecret);
```

6.2 调用图像识别接口

(1) 识别一个 BGR、倒立、无行对齐位数据格式图像

①直接调用

```
char *SKEyeSDK_Image(unsigned char *ImageData, unsigned
int ImageWidth, unsigned int ImageHeight, char *
ServiceName);
```

- 参数:
 - ImageData: BGR、倒立、无行对齐, 图片数据
 - ImageWidth: 图片的宽度
 - ImageHeight: 图片的高度
 - ServiceName: 识别服务名称
- 返回值:

[Json 格式的字符串](#)

②接口回调

```
Void SKEyeSDK_Image(unsigned char *ImageData, unsigned
int ImageWide, unsigned int ImageHight, char
*ServiceName, CallBackImageResult ImageResult);
Void SKEyeSDK_Image(int ImageId, unsigned char *ImageData,
unsigned int ImageWide, unsigned int ImageHeight, char
*ServiceName, CallBackImageResult ImageResult);
```

- 参数:
 - ImageId: 图片 Id (方便确认请求顺序和返回顺序是否一致)
 - ImageData: BGR、倒立、无行对齐, 图片数据
 - ImageWidth: 图片的宽度
 - ImageHeight: 图片的高度
 - ServiceName: 识别服务名称

- ImageResult: 回调接口
- 接口回调返回值:
[Json 格式的字符串](#)

直接调用示例代码:

```
// 直接调用
unsigned char *ImageData; unsigned
int ImageWith, ImageHeight; char
ImageUrl []="4.bmp";
//GetImageData 获取 BGR、倒立、无行对齐 图像数据
if (!GetImageData (ImageUrl, ImageData, ImageWidth,
ImageHeight))
{ printf("Open is error\n");
return 0;
}
char ApiKey[] = "466da4221010f8341912da500b4a23b"; char
ApiSecret[] = "ddc8bd498e7c29c929862ab7391a7b89";
SKEyeSDK_Init(ApiKey, ApiSecret);
char*ImageJsonData= SKEyeSDK_Image(ImageData, ImageWith,
ImageHeight, ServiceName);
```

接口回调示例代码:

```
//接口回调
void CallBackImageResultFunction (char *ImageResult)
{
    //这里添加代码
    printf("%s\n", ImageResult);
} unsigned char *ImageData;
Int ImageId=1;
unsigned int ImageWith, ImageHeight;
char ImageUrl []="4.bmp";
//GetImageData 获取 BGR、倒立、无行对齐 图像数据
if (!GetImageData (ImageUrl, ImageData, ImageWidth,
ImageHeight))
{ printf("Open is error\n");
return;
} char ApiKey[] = "466da4221010f8341912da500b4a23b";
char ApiSecret[] = "ddc8bd498e7c29c929862ab7391a7b89";
```

```
SKEyeSDK_Init(ApiKey, ApiSecret);  
//回调  
SKEyeSDK_Image(ImageData, ImageWith, ImageHeight,  
ServiceName, CallBackImageResultFunction);  
//回掉图片 Id (方便确认请求顺序和返回顺序是否一致)  
SKEyeSDK_Image(ImageId, ImageData, ImageWith, ImageHeight,  
ServiceName, CallBackImageResultFunction);
```

(2) 识别一个 url (网络 url、本地路径) 图像

①直接调用

```
char *SKEyeSDK_ImagePath(char * UrlOrLocalPath, char  
*ServiceName);
```

- 参数:
 - UrlOrLocalPath: 图像的 url (网络 url 或者本地路径)
 - ServiceName: 识别服务名称
- 返回值:
[Json 格式的字符串](#)

②接口回调

```
typedef void(*CallBackImageResult)(char *ImageResult);  
void SKEyeSDK_ImagePath(char *UrlOrLocalPath, char  
*ServiceName, CallBackImageResult ImageResult);  
void SKEyeSDK_ImagePath(int ImageId, char *UrlOrLocalPath,  
char *ServiceName, CallBackImageResult ImageResult);
```

- 参数:
 - ImageId: 图片 Id (方便确认请求顺序和返回顺序是否一致)
 - UrlOrLocalPath: 图像的 url (网络 url 或者本地路径)
 - ServiceName: 识别服务名称
 - ImageResult: 回调接口
- 接口回调返回值:
[Json 格式的字符串](#)

直接调用示例代码:

```
// imageUrl (网络 url 或者本地路径)  
char imageUrl [] = "object3.jpg"; char serviceName[] =  
"objects"; char *imageResult=NULL; char apiKey[] =  
"466da4221010f8341912da500b4a23b"; char apiSecret[] =  
"ddc8bd498e7c29c929862ab7391a7b89";
```

```
SKEyeSDK_Init(ApiKey, ApiSecret);  
ImageResult =SKEyeSDK_ImagePath(ImageUrl,ServiceNames);
```

接口回调示例代码:

```
// ImageUrl (网络 url 或者本地路径)  
Int ImageId=0;  
char ImageUrl [] = "object3.jpg";  
char ServiceName[] = "objects";  
void CallBackImageResultFunction (char * ImageResult)  
{  
    printf("%s\n", ImageResult);  
}  
  
char ApiKey[] = "466da4221010f8341912da500b4a23b";  
char ApiSecret[] = "ddc8bd498e7c29c929862ab7391a7b89";  
SKEyeSDK_Init(ApiKey, ApiSecret);  
//回掉  
SKEyeSDK_ImagePath(ImageUrl, ServiceName,  
CallBackImageResultFunction);  
//回掉图片 Id (方便确认请求顺序和返回顺序是否一致)  
SKEyeSDK_ImagePath(ImageId ,ImageUrl, ServiceName,  
CallBackImageResultFunction);
```

JSON 格式返回值:

```
{  
  "error code":"0",  
  "error msg":"OK",  
  "tags":[  
    {  
      "tag name":"运动鞋",  
      "tag_confidence":81  
    },  
    {  
      "tag name":"跑鞋",  
      "tag_confidence":18  
    }  
  ]  
}  
  
{  
  "error code":"0",  
  "error msg":"OK",  
  "image id ":"image_id",  
  "tags":[  
    {  
      "tag name":"运动鞋",  
      "tag_confidence":81  
    },  
    {  
      "tag_name":"跑鞋",  

```

```
    "tag_confidence":18  
  }  
]  
}
```

7. 补充

如果遇到问题，请按以下步骤尝试解决：

- 查看 [SKEye 人工智能开放平台 API 文档](#)
- 仔细阅读 Demo 接口调用源码
- 联系我们的工作人员
- 客服电话：010-62538800
- 技术支持邮箱：support@interjoy.com.cn
- 开发平台用户 QQ 群:617518775