

# ArcSoftFace

## C++ Demo 说明文档

# 目录

目录 .....	2
1. 简介 .....	3
1.1 运行环境 .....	3
1.2 系统要求 .....	3
1.3 开发工具 .....	3
1.4 支持的颜色空间格式 .....	3
1.5 产品功能简介 .....	3
1.5.1 人脸检测 .....	3
1.5.2 年龄检测 .....	3
1.5.3 性别检测 .....	3
1.5.4 人脸识别 .....	4
2. 快速上手 .....	4
2.1 环境配置 .....	4
2.2 运行程序 .....	4
3. 接入指南 .....	5
3.1 示例代码 .....	5
3.1.1 引擎激活 .....	5
3.1.2 初始化引擎 .....	5
3.1.3 人脸检测 .....	5
3.1.4 特征提取 .....	6
3.1.5 人脸比对 .....	6
3.1.6 年龄、性别信息检测 .....	6
3.1.7 红外活体检测 .....	7
4. 常见问题 .....	8
4.1 常见问题问答 .....	8
4.2 其他帮助 .....	8

# 1. 简介

## 1.1 运行环境

Windows 平台

最低硬件配置

Intel® Core™ i5-2300@2.80GHz 或者同级别芯片

推荐硬件配置

Intel® Core™ i7-4600U@2.1GHz 或者同级别芯片

## 1.2 系统要求

Windows7 及以上

## 1.3 开发工具

VS2013 及以上版本

## 1.4 支持的颜色空间格式

支持图像的颜色空间格式：BGR24/GRAY

## 1.5 产品功能简介

### 1.5.1 人脸检测

从图片中检测人脸信息，人脸在图片中的位置坐标信息。

### 1.5.2 年龄检测

对图片中对应的人脸图片信息数据进行年龄检测。

### 1.5.3 性别检测

对图片中对应的人脸图片信息数据进行性别检测。

### 1.5.4 人脸识别

将从图片中提取的两个人脸特征信息，通过人脸识别 SDK 中人脸比对，即对两个特征值进行比较，通过返回的相似度判断两个人是否是一个人。

## 2. 快速上手

### 2.1 环境配置

1. 安装 VS2013 环境安装包 (vcredist\_x86\_vs2013.exe)
2. 从官网 (<http://www.arcsoft.com.cn/ai/arcface.html>) 申请 sdk，下载对应的 sdk 版本 (x86 或 x64) 并解压
3. 头文件配置: inc 文件夹内文件放入 \ArcFaceDemo\include\inc 文件夹内
4. SDK 库配置:
  - x64 版本:  
将 libarcsoft\_face\_engine.lib 放至 \ArcFaceDemo\lib64\FreeSdk 文件夹下
  - x86 版本:  
将 libarcsoft\_face\_engine.lib 放至 \ArcFaceDemo\lib32\FreeSdk 文件夹下
5. 在运行代码的时候将对应版本 SDK 和 OpenCV 的 dll 库放至项目根目录下，以免运行时找不到对应的 dll 库
6. 将官网申请的 APPID 及 SDKKEY 填写至 ArcFaceEngine.cpp 文件中，注意平台和版本对应
7. 在 Debug 或者 Release 中选择配置管理器，选择对应的平台
8. Demo 使用多字节字符集，如果报 MSB8031 错误，请下载安装 vc\_mbcsmfc.exe (<https://www.microsoft.com/en-us/download/details.aspx?id=40770>)

### 2.2 运行程序

1. 按 F5 启动程序
2. 点击“注册人脸”按钮增加人脸库图片，针对每张照片只提取最大人脸进行检测以及人脸特征检测并保存到缓存中
3. 点击“选择识别照”按钮增加识别照图片，左上角显示检测到的年龄、性别信息
4. 点击“人脸比对”按钮进行人脸比对，左上角显示比对结果
5. 点击“启用摄像头”按钮可以打开摄像头并自动与人脸库进行比对，并动态显示结果
6. 点击“清除人脸库”按钮，即清除当前人脸库

7. 程序执行信息会在界面底部中的编辑框中进行显示
8. 阈值框内在启用摄像头的情况下可以自由更改, 只有达到阈值的比对信息才会显示

## 3. 接入指南

### 3.1 示例代码

#### 3.1.1 引擎激活

```
//激活SDK
HRESULT ArcFaceEngine::ActiveSDK()
{
    HRESULT res = ASFActivation(APPID, SDKKEY);
    if (MOK != res && MERR_ASF_ALREADY_ACTIVATED != res)
        return res;
    return MOK;
}
```

#### 3.1.2 初始化引擎

```
//初始化引擎
HRESULT ArcFaceEngine::InitEngine(MLong detectMode)
{
    m_hEngine = NULL;
    MInt32 mask = 0;

    if (ASF_DETECT_MODE_IMAGE == detectMode)
    {
        mask = ASF_FACE_DETECT | ASF_FACERECOGNITION | ASF_AGE | ASF_GENDER | ASF_FACE3DANGLE;
    }
    else
    {
        mask = ASF_FACE_DETECT | ASF_FACERECOGNITION;
    }

    HRESULT res = ASFInitEngine(detectMode, ASF_OP_0_HIGHER_EXT, NSCALE, FACENUM, mask, &m_hEngine);
    return res;
}
```

初始化时要根据需要设置需要初始化的属性; 应用程序关闭时, 必须销毁引擎, 否则会造成内存泄漏。

```
HRESULT ArcFaceEngine::UnInitEngine()
{
    //销毁引擎
    return ASFUninitEngine(m_hEngine);
}
```

#### 3.1.3 人脸检测

使用人脸检测功能需要在初始化引擎时将人脸检测对应的宏 ASF\_FACE\_DETECT 添加到 mask 参数中, 根据要求传入参数, 调用方式如下:

```
MRESULT ArcFaceEngine::PreDetectFace(IplImage* image, ASF_SingleFaceInfo& faceRect)
{
    if (!image)
    {
        return -1;
    }

    IplImage* cutImg = cvCreateImage(cvSize(image->width - (image->width % 4), image->height),
        IPL_DEPTH_8U, image->nChannels);

    PicCutOut(image, cutImg, 0, 0);

    ASF_MultiFaceInfo detectedFaces = { 0 };//人脸检测
    //FD
    MRESULT res = ASFDetectFaces(m_hEngine, cutImg->width, cutImg->height,
        ASVL_PAF_RGB24_B8G8R8, (MUInt8*)cutImg->imageData, &detectedFaces);
    if (res != MOK || detectedFaces.faceNum < 1)
    {
        cvReleaseImage(&cutImg);
        return -1;
    }
}
```

### 3.1.4 特征提取

使用人脸特征提取功能需要在初始化引擎时将人脸检测对应的宏 ASF\_FACERECOGNITION 添加到 mask 参数中, 根据要求传入参数, 调用方式如下:

```
// 预先提取人脸特征
MRESULT ArcFaceEngine::PreExtractFeature(IplImage* image, ASF_FaceFeature& feature, ASF_SingleFaceInfo& faceRect)
{
    if (!image || image->imageData == NULL)
    {
        return -1;
    }

    IplImage* cutImg = cvCreateImage(cvSize(image->width - (image->width % 4), image->height),
        IPL_DEPTH_8U, image->nChannels);

    PicCutOut(image, cutImg, 0, 0);

    if (!cutImg)
    {
        cvReleaseImage(&cutImg);
        return -1;
    }

    MRESULT res = MOK;

    ASF_FaceFeature detectFaceFeature = { 0 };//特征值

    res = ASFFaceFeatureExtract(m_hEngine, cutImg->width, cutImg->height, ASVL_PAF_RGB24_B8G8R8,
        (MUInt8*)cutImg->imageData, &faceRect, &detectFaceFeature);

    if (MOK != res)
    {
    }
}
```

### 3.1.5 人脸比对

使用人脸特征比对功能的调用方式如下:

```
MFloat confidenceLevel = 0;
MRESULT pairRes = dialog->m_videoFaceEngine.FacePairMatching(confidenceLevel, faceFeature, regisFeature);

if (MOK == pairRes && confidenceLevel > maxThreshold)
{
    maxThreshold = confidenceLevel;
    maxIndex = curIndex;
}
```

### 3.1.6 年龄、性别信息检测

使用年龄、性别信息检测功能需要在初始化引擎时将人脸检测对应的宏

ASF\_AGE、ASF\_GENDER 添加到 mask 参数中，根据要求传入参数，调用方式如下：

```
MRESULT ArcFaceEngine::FaceASFProcess(ASF_MultiFaceInfo detectedFaces, IplImage *img, ASF_AgeInfo &ageInfo,
    ASF_GenderInfo &genderInfo, ASF_Face3DAngle &angleInfo)
{
    if (!img)
    {
        return -1;
    }

    MInt32 lastMask = ASF_AGE | ASF_GENDER | ASF_FACE3DANGLE;

    IplImage* cutImg = cvCreateImage(cvSize(img->width - (img->width % 4), img->height), IPL_DEPTH_8U, img->nChannels);
    PicCutOut(img, cutImg, 0, 0);

    if (!cutImg)
    {
        cvReleaseImage(&cutImg);
        return -1;
    }

    int res = ASFProcess(m_hEngine, cutImg->width, cutImg->height, ASVL_PAF_RGB24_B8G8R8,
        (MUInt8*)cutImg->imageData, &detectedFaces, lastMask);

    res = ASFGetAge(m_hEngine, &ageInfo);

    res = ASFGetGender(m_hEngine, &genderInfo);

    res = ASFGetFace3DAngle(m_hEngine, &angleInfo);

    cvReleaseImage(&cutImg);

    return res;
}
```

### 3.1.7 红外活体检测

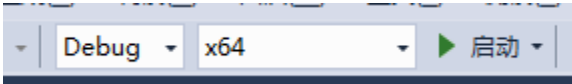
使用红外检测功能需要在初始化引擎时将人脸检测对应的宏 ASF\_IR\_LIVENESS 添加到 mask 参数中，根据要求传入参数，调用方式如下：

```
bool isIRAlive = false;
if (g_cameraNum == 2)
{
    IplImage* tempIRImage = NULL;
    lock_guard<mutex> lock(g_mutex);
    {
        if (dialog->m_curIrVideoImage)
        {
            tempIRImage = cvCloneImage(dialog->m_curIrVideoImage);
        }
    }

    if (dialog->m_irDataValid)
    {
        ASF_LivenessInfo irLiveNessInfo = { 0 };
        MRESULT irRes = dialog->m_imageFaceEngine.FaceASFProcess_IR(multiFaceInfo, tempIRImage, irLiveNessInfo);
        if (irRes == 0 && irLiveNessInfo.isLive[0] == 1)
        {
            dialog->m_curIRVideoShowString = "IR活体";
            isIRAlive = true;
        }
        else
        {
            dialog->m_curIRVideoShowString = "IR假体";
        }
    }
    else
    {
        dialog->m_curIRVideoShowString = "IR假体";
    }
}
```

## 4. 常见问题

### 4.1 常见问题问答

问题	参考回复
启动后引擎初始化失败	<ol style="list-style-type: none"><li>1. 请选择对应的平台，如 x64, x86;</li></ol>  <ol style="list-style-type: none"><li>2. 删除工程目录下对应的 asf_install.dat, freesdk_132512.dat;</li><li>3. 请确保 ArcFaceEngine.cpp 中 appid 和 appkey 与当前 sdk 版本一致。</li></ol>
SDK 支持那些格式的图片人脸检测?	目前 SDK 支持的图片格式有 jpg, jpeg, png, bmp 等。
使用人脸检测功能对图片大小有要求吗?	推荐的图片大小最大不要超过 2M, 因为图片过大会使人脸检测的效率不理想, 当然图片也不宜过小, 否则会导致无法检测到人脸。
使用人脸识别引擎提取到的人脸特征信息是什么?	人脸特征信息是从图片中的人脸上提取的人脸特征点, 是 unsigned char* 格式。
SDK 人脸比对的阈值设为多少合适?	推荐值为 <b>0.8</b> , 用户可根据不同场景适当调整阈值。
可不可以将人脸特征信息保存起来, 等需要进行人脸比对的时候直接拿保存好的人脸特征进行比对?	可以, 当人脸个数比较多时推荐先存储起来, 在使用时直接进行比对, 这样可以大大提高比对效率。存入数据库 (当启动程序时读入缓存) 或存入缓存中;
图片中有人脸, 但是检测时未检测到人脸	<ol style="list-style-type: none"><li>1. 请调整 nscale 的值, video 为 2-16, image 为 2-32;</li><li>2. 请确认图片的宽度是否为 4 的倍数, 高度除 RGB24 格式外, YUV 格式高度需要满足 2 的倍数;</li></ol>
编译错误 MSB8031	请下载安装 vc_mbcsmfc.exe ( <a href="https://www.microsoft.com/en-us/download/details.aspx?id=40770">https://www.microsoft.com/en-us/download/details.aspx?id=40770</a> )

更多常见问题请访问 <https://ai.arcsoft.com.cn/manual/faqs.html>。

### 4.2 其他帮助

您在开发过程中遇到了问题, 或者对我们的人脸识别 SDK 有意见或建议, 欢迎在虹软官方论坛 <https://ai.arcsoft.com.cn/bbs/portal.php> 上发帖提问, 我们的工作人员会竭力为您解答。