**ArcSoftFace**

C# Demo说明文档

目录

[1. 简介 1](#_Toc62224884)

[1.1 运行环境 1](#_Toc62224885)

[1.2 系统要求 1](#_Toc62224886)

[1.3 开发工具 1](#_Toc62224887)

[1.4 环境要求 1](#_Toc62224888)

[1.5 支持的颜色空间格式 1](#_Toc62224889)

[1.6 产品功能简介 1](#_Toc62224890)

[1.6.1 人脸检测/人脸追踪 1](#_Toc62224891)

[1.6.2 年龄检测 1](#_Toc62224892)

[1.6.3 性别检测 1](#_Toc62224893)

[1.6.4 人脸识别 2](#_Toc62224894)

[1.6.5 RGB活体检测 2](#_Toc62224895)

[1.6.6 IR活体检测 2](#_Toc62224896)

[1.6.7 图像质量检测 2](#_Toc62224897)

[1.6.8 口罩检测 2](#_Toc62224898)

[1.6.9 额头区域检测 **错误!未定义书签。**](#_Toc62224899)

[2. 快速上手 2](#_Toc62224900)

[3. 接入指南 3](#_Toc62224901)

[3.1 示例代码 3](#_Toc62224902)

[3.1.1 引擎在线激活 3](#_Toc62224903)

[3.1.2 引擎离线激活 3](#_Toc62224904)

[3.1.3 初始化引擎 3](#_Toc62224905)

[3.1.4 人脸检测 4](#_Toc62224906)

[3.1.5 口罩检测 4](#_Toc62224907)

[3.1.6 图像质量检测 4](#_Toc62224908)

[3.1.7 提取特征 4](#_Toc62224909)

[3.1.8 人脸比对 5](#_Toc62224910)

[3.1.9 RGB活体检测 5](#_Toc62224911)

[3.1.10 IR活体检测 5](#_Toc62224912)

[3.1.11 额头区域检测 **错误!未定义书签。**](#_Toc62224913)

[3.2 通用方法 5](#_Toc62224914)

[3.2.1 从Bitmap中读取BGR数据 5](#_Toc62224915)

[3.2.2 从Bitmap中读取GRAY数据 6](#_Toc62224916)

[4. 常见问题 6](#_Toc62224917)

[4.1 常见问题问答 6](#_Toc62224918)

[4.2 其他帮助 7](#_Toc62224919)

# 简介

## 运行环境

Windows平台

最低硬件配置

Intel® CoreTM i5-2300@2.80GHz 或者同级别芯片

推荐硬件配置

Intel® CoreTMi7-4600U@2.1GHz 或者同级别芯片

## 系统要求

Windows7及以上

## 开发工具

VS2013以上版本、USB摄像头

## 环境要求

.Net Framework 4.5.1以上

Microsoft Visual C++ 2013 Redistributable环境包

## 支持的颜色空间格式

支持图像的颜色空间格式：BGR24、ASVL\_PAF\_GRAY

## 产品功能简介

### 人脸检测/人脸追踪

通过ASFDetectFaces或ASFDetectFacesEx方法从图片中检测人脸信息，获取图片中人脸框的个数、位置坐标信息和角度信息。不同的使用场景，初始化时使用不同模式：

1. 图片检测模式（适用于静态图片识别）：ASF\_DETECT\_MODE\_IMAGE
2. 视频检测模式（适用于摄像头预览，视频文件识别）：ASF\_DETECT\_MODE\_VIDEO

### 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。对应方法：通过ASFProcess方法从图片中检测人脸信息，通过ASFGetAge方法获取年龄检测结果。

### 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。对应方法：通过ASFProcess或ASFProcessEx方法从图片中检测人脸信息，通过ASFGetGender方法获取性别检测结果。

### 人脸识别

通过ASFFaceFeatureExtract或ASFFaceFeatureExtractEx方法从图片中提取人脸特征信息，通过人脸识别SDK中人脸比对的方法：ASFFaceFeatureCompare，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。

### RGB活体检测

对图片中对应的人脸图片信息数据进行活体检测。对应方法：通过ASFProcess或ASFProcessEx方法从图片中检测人脸信息，通过ASFGetLivenessScore方法获取活体检测结果。

### IR活体检测

对图片中对应的人脸图片信息数据进行活体检测。对应方法：通过ASFProcess\_IR或ASFProcessEx\_IR方法从图片中检测人脸信息，通过ASFGetLivenessScore\_IR方法获取活体检测结果。

### 图像质量检测

对图片中相应的人脸框区域进行图像质量检测。对应方法：通过ASFImageQualityDetectEx方法获取图像质量检测结果。

### 口罩检测

对图片中对应的人脸图片信息数据进行口罩检测。对应方法：通过ASFProcess或ASFProcessEx方法从图片中检测人脸信息，通过ASFGetMask方法获取口罩检测结果。

# 快速上手

1. 安装VS2013环境安装包（vcredist\_x86\_vs2013.exe），确认本地.NET Framework版本
2. 从虹软开发者中心官网（<https://ai.arcsoft.com.cn/ucenter/user/userlogin>）下载ArcFace 4.0 SDK包(x86或x64，平台请根据项目编译环境选择)并解压
3. 将libs中的文件拷贝到工程bin目录的对应平台的Debug或Release目录下
4. 将对应APP\_ID、SDK\_KEY、ACTIVE\_KEY替换App.config文件中对应内容
5. 在Debug或者Release中选择配置管理器，选择对应的平台
6. 按F5启动程序
7. 点击“注册人脸”按钮增加人脸库图片
8. 点击“选择识别图”按钮增加人脸图片
9. 点击“开始匹配”按钮进行比较
10. 根据下面文本框查看相关信息

# 接入指南

## 示例代码

### 引擎在线激活

retCode = FaceEngine. ASFOnlineActivation(appId, is64CPU ? sdkKey64 : sdkKey32, is64CPU ? activeCode64 : activeCode32);

### 引擎离线激活

string deviceInfo;

//获取设备信息

retCode = imageEngine.ASFGetActiveDeviceInfo(out deviceInfo);

if (retCode != 0)

{

    MessageBox.Show("获取设备信息失败，错误码:" + retCode);

}

else

{

    File.WriteAllText("ActiveDeviceInfo.txt", deviceInfo);

}

//官网执行离线激活操作，将生成的离线授权文件,再执行以下操作

string offlineActiveFilePath = "离线激活文件路径";

//离线激活

retCode = imageEngine.ASFOfflineActivation(offlineActiveFilePath);

### 初始化引擎

图片模式引擎初始化

//初始化引擎

DetectionMode detectMode = DetectionMode.ASF\_DETECT\_MODE\_IMAGE;

//Video模式下检测脸部的角度优先值

ASF\_OrientPriority videoDetectFaceOrientPriority = ASF\_OrientPriority.ASF\_OP\_ALL\_OUT;

//Image模式下检测脸部的角度优先值

ASF\_OrientPriority imageDetectFaceOrientPriority = ASF\_OrientPriority.ASF\_OP\_ALL\_OUT;

//人脸在图片中所占比例，如果需要调整检测人脸尺寸请修改此值，有效数值为2-32

int detectFaceScaleVal = 16;

//最大需要检测的人脸个数

int detectFaceMaxNum = 5;

//引擎初始化时需要初始化的检测功能组合

int combinedMask = FaceEngineMask.ASF\_FACE\_DETECT | FaceEngineMask.ASF\_FACERECOGNITION | FaceEngineMask.ASF\_AGE | FaceEngineMask.ASF\_GENDER | FaceEngineMask.ASF\_FACE3DANGLE;

//初始化引擎，正常返回值为0

retCode = imageEngine.ASFInitEngine(detectMode, imageDetectFaceOrientPriority,

 detectFaceScaleVal, detectFaceMaxNum, combinedMask);

视频模式初始化

DetectionMode detectModeVideo = DetectionMode.ASF\_DETECT\_MODE\_VIDEO;

int combinedMaskVideo = FaceEngineMask.ASF\_FACE\_DETECT | FaceEngineMask.ASF\_FACERECOGNITION;

retCode = videoEngine.ASFInitEngine(detectModeVideo, videoDetectFaceOrientPriority,

detectFaceScaleVal, detectFaceMaxNum, combinedMaskVideo);

初始化时要设置需要用的算法功能。

### 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方法类型(FaceEngineMask.ASF\_FACE\_

DETECT做初始化，将图片文件（图片文件一般为RGB格式图像数据）作为参数传入FaceEngine.ASFDetectFaces()或 FaceEngine.ASFDetectFacesEx()的人脸检测方法即可，示例代码如下：

//人脸检测

MultiFaceInfo multiFaceInfo = new MultiFaceInfo();

faceEngine.ASFDetectFacesEx(image, out multiFaceInfo);

### 口罩检测

口罩检测需要在初始化引擎时设置口罩检测算法功能。使用FaceEngine.ASFProcessEx()或者FaceEngine.ASFProcess()方法检测人脸信息，然后通过FaceEngine.ASFGetMask()方法获取口罩检测结果，示例代码如下：

MaskInfo maskInfo = new MaskInfo();

//人脸信息处理

retCode = faceEngine.ASFProcessEx(image, multiFaceInfo, FaceEngineMask.ASF\_MASKDETECT);

if (retCode == 0)

{

    //获取IR活体检测结果

    retCode = faceEngine.ASFGetMask(out maskInfo);

}

### 图像质量检测

图像质量检测需要在初始化引擎时设置图像质量检测算法功能，当人脸检测完成后，将图像和人脸检测结果、口罩检测结果传入FaceEngine. ASFImageQualityDetectEx()方法来提取人脸的图像质量，示例代码如下：

//图像质量检测

float confidenceLevel;

retCode = faceEngine.ASFImageQualityDetectEx(image, multiFaceInfo, out confidenceLevel, faceIndex, isMask);

### 提取特征

提取特征功能需要在初始化引擎时将人脸识别功能类型初始化(FaceEngineMask.ASF\_FACERECOG

NITION)，将图像、人脸检测结果、口罩检测结果作为参数传入FaceEngine.ASFFaceFeatureExt

ract()或FaceEngine.ASFFaceFeatureExtractEx()方法来提取人脸特征信息，示例代码如下：

//特征提取

FaceFeature faceFeature = new FaceFeature();

retCode = faceEngine.ASFFaceFeatureExtractEx(image, multiFaceInfo, registerOrNot, out faceFeature, faceIndex, isMask);

### 人脸比对

人脸比对功能是通过对比两个人脸特征信息，返回两者的相似程度。通过人脸检测，提取特征后，通过FaceEngine.ASFFaceFeatureCompare()的人脸比对方法对比两个人脸特征信息，获取它们的相似度，示例代码如下：

float similarity = 0f;

int retCode = videoRGBImageEngine.ASFFaceFeatureCompare(feature, imagesFeatureList[i],

out similarity);

### RGB活体检测

RGB活体检测功能是通过RGB图像检测图像中的人是否为活体。使用FaceEngine.ASF

ProcessEx()或者FaceEngine.ASFProcess()方法检测RGB图像的人脸信息，然后通过FaceEngine

.ASFGetLivenessScore()方法获取RGB活体检测结果，示例代码如下：

LivenessInfo livenessInfo = new LivenessInfo();

//人脸信息检测

retCode = faceEngine.ASFProcessEx(image, multiFaceInfo, FaceEngineMask.ASF\_LIVENESS);

if (retCode == 0)

{

    //获取活体检测结果

    retCode = faceEngine.ASFGetLivenessScore(out livenessInfo);

}

return livenessInfo;

### IR活体检测

IR活体检测功能是通过红外摄像头获取人脸检测数据，根据检测数据判断图片中是否有活体。使用FaceEngine.ASFProcess\_IR()或FaceEngine.ASFProcessEx\_IR()方法检测IR图像的人脸信息，然后通过FaceEngine.ASFGetLivenessScore\_IR()方法来获取IR活体检测结果，示例代码如下：

LivenessInfo livenessInfo = new LivenessInfo();

 //人脸信息处理

retCode = faceEngine.ASFProcessEx\_IR(image, multiFaceInfo, FaceEngineMask.ASF\_IR\_LIVENESS);

if (retCode == 0)

{

    //获取IR活体检测结果

    retCode = faceEngine.ASFGetLivenessScore\_IR(out livenessInfo);

}

return livenessInfo;

## 通用方法

### 从Bitmap中读取BGR数据

从Bitmap中读取BGR数据的方法比较复杂，可以参考ImageUtil.ReadBMP(Image image)方法。

### 从Bitmap中读取GRAY数据

从Bitmap中读取GRAY数据的方法比较复杂，可以参考ImageUtil.ReadBMP\_IR(Image image)方法。

# 常见问题

## 常见问题问答

|  |  |
| --- | --- |
| 问题 | 参考回复 |
| 启动后引擎初始化失败 | 1. 请选择对应的平台，如x64,x86 2. 删除bin下面对应的asf\_install.dat； 3. 请确保App.config下的APPID、SDKKEY与当前SDK的平台、版本保持一致。 |
| 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byte[]数组格式。 |
| SDK人脸比对的阈值设为多少合适？ | 推荐值为**0.8**，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。存入数据库时，请以Blob的格式进行存储，不能以string或其他格式存储。 |
| VS中调试激活时，返回90113 SDK激活失败,请打开读写权限 | 1、调试环境下：当前VS没有权限，请使用管理员身份运行  2、IIS环境部署下的Web服务：请将SDK文件夹添加IUSR和IIS\_USRS、NETWORK\_SERVICE用户的写入、修改权限,并在IIS应用程序池中将高级设置中的标识设为LocalSystem |
| 在.Net项目中出现堆栈溢出问题 | .Net平台设置的默认堆栈大小为256KB，SDK中需要的大小为512KB以上，推荐调整堆栈的方法为：  new Thread(new ThreadStart(delegate {  ASF\_MultiFaceInfo multiFaceInfo = FaceUtil.DetectFace(pEngine, imageInfo);  }), 1024 \* 512).Start(); |
| 在.Net项目中出现x64不能加载SDK的问题 | 首先使用 bool is64 = Environment.Is64BitProcess; 查看当前是否是x64位编译器，如果不是则按在Visual Studio中选择菜单“工具>选项>项目和解决方案>Web项目”，在对话框中勾选“对网站和项目使用IIS Express的64位版本”。 |
| X86模式下批量注册人脸有内存溢出或图片空指针 | 请增加虚拟内存或每次批量注册人脸控制在20张图片范围内 |
| 图片中有人脸，但是检测时未检测到人脸 | 1. 请调整detectFaceScaleVal的值； 2. 请确认图片的宽度是否为4的倍数； 3. 请确认图片是否通过ImageUtil.ReadBMP方法进行数据调整。 |
| 尝试读取或写入受保护的内存 | 尝试读取或写入受保护的内存是内存操作不当导致，请查看您程序中是否有内存使用后未释放，或者使用了已经释放的内存！ |
| 销毁引擎时程序报错 | 请先确认销毁引擎的时候，引擎是否处于被占用的状态。 |
| 多人脸活体检测，调用ASFProcess()接口，返回值为9 | 错误码9表示缓冲上溢。建议在活体检测引擎初始化时，将detectFaceMaxNum值设置为不小于实际要检测人脸的值。 |
| 多线程调用注意事项 | 1.同一个引擎可以使用多线程调用不同算法;  2.多线程调用同一个算法接口需要启用不同的引擎 |

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

## 其他帮助

如您想要了解更多虹软的产品，请访问虹软官网<http://www.arcsoft.com.cn/>，或者您在开发的过程中遇到了问题，或者对我们的人脸识别SDK有什么意见或建议，欢迎在虹软官方论坛[https://ai.arcsoft.com.cn//bbs/portal.php](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。