

ArcFace Qt Demo 说明文档

ArcFace Qt Demo 说明文档

- 1. 简介
 - 1.1 运行环境
 - 1.2 系统要求
 - 1.3 开发工具
 - 1.4 支持的颜色空间格式
- 2. 快速上手
 - 2.1 环境配置
 - 2.2 运行程序
- 3. 代码解析
 - 3.1 FaceEngine类
 - 3.2 人脸注册模块(IMAGE模式)
 - 3.3 识别照模块(IMAGE模式)
 - 3.4 人脸比对(IMAGE模式)
 - 3.5 摄像头模块
 - 3.6 视频流模块(VIDEO模式)
- 4 Demo 打包
 - 4.1 提取Qt的动态链接库
 - 4.2 检查缺少的DLL文件
 - 4.3 最终打包的文件
 - 4.4 疑难点

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

- VS2015 及以上版本
- Qt 5.0 版本以上

1.4 支持的颜色空间格式

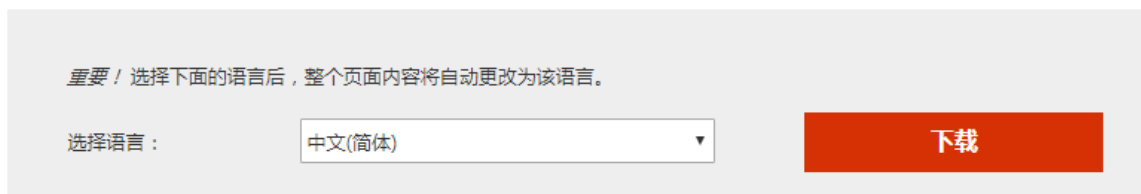
- 支持图像的颜色空间格式：BGR24 / GRAY

2. 快速上手

2.1 环境配置

1. 安装 VS2015 环境安装包 (<https://www.microsoft.com/zh-CN/download/details.aspx?id=48145>)

Visual C++ Redistributable for Visual Studio 2015



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:
 - SDK包中inc 文件夹内头文件拷贝到ArcFaceDetection\ArcFaceDetection\FaceDetection\inc目录下
4. SDK 库配置:
 - x86版本将libarcsoft_face_engine .lib 拷贝到ArcFaceDetection\ArcFaceDetection\FaceDetection\lib\Win32 文件夹下
 - x64版本将libarcsoft_face_engine .lib 拷贝到ArcFaceDetection\ArcFaceDetection\FaceDetection\lib\x64 文件夹下
5. 在运行代码的时候将对应版本ArcFace SDK 和 OpenCV 的 dll 库拷贝到 项目执行目录下, 以免运行时找不到对应的 dll 库
6. 在setting.ini文件中做下图一些初始化配置:

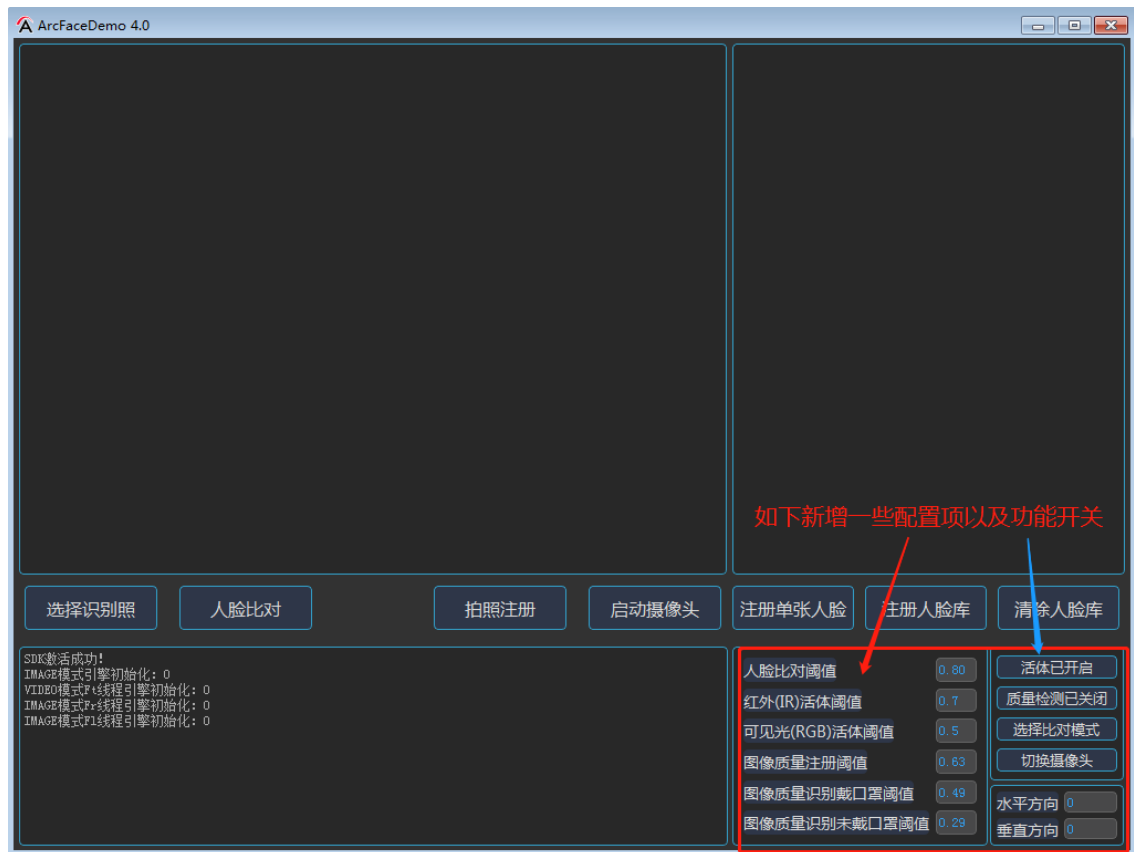
```
[ActiveInfo_x86]
appId=Bbvyu5GeUE8eaBhyLsN
sdkKey=EvvdWPFj7XjL1siCQiSD2qGglSbFKSL
activeKey=8351-1111-D3
[ActiveInfo_x64]
appId=Bbvyu5GeUE8eaBhyLsNcp49HW6tuPx
sdkKey=EvvdWPFj7XjL1siCQiSD2qGjG6l
activeKey=8661-1111-
```

7. 在 Debug 或者 Release 中选择配置管理器, 选择对应的平台

2.2 运行程序

1. 按 F5 启动程序
2. 点击 注册单张人脸 或 注册人脸库 按钮增加人脸库图片, 针对每张照片只提取第一张人脸进行检测, 将人脸特征检测并保存到缓存;
3. 点击 选择识别照 按钮添加识别照图片, 显示检测到的年龄、性别、活体信息;
4. 点击 人脸比对 按钮进行人脸比对, 在编辑框中输出比对结果;
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, 动态显示比对结果;

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. 界面上可以设置阈值、功能开关、摄像头切换以及比对模式；



3. 代码解析

3.1 FaceEngine类

根据自身业务需求对 SDK 接口进行封装，Demo中 `FaceEngine` 类对人脸识别算法接口做了一些封装，便于业务层调用，如下图所示：

```
//激活SDK
HRESULT ActiveSDK(char* appId, char* sdkKey, char* activeKey);
//获取激活信息
HRESULT GetActiveFileInfo(ASF_ActiveFileInfo& activeFileInfo);
//引擎初始化
HRESULT InitEngine(ASF_DetectMode detectMode, ASF_OrientPriority detectFaceOrientPriority);
//释放引擎
HRESULT UnInitEngine();
//人脸检测
HRESULT PreDetectFace(IplImage* image, ASF_MultiFaceInfo& multiFaceInfo, bool isRgb);
//图像质量检测
HRESULT PreImageQualityDetect(IplImage* image, ASF_MultiFaceInfo multiFaceInfo, float rgbFqThreshold);
//人脸特征提取
HRESULT PreExtractFeature(IplImage* image, ASF_SingleFaceInfo& faceRect,
    ASF_FaceFeature& feature, MInt32 featureThreadNum = 1);
//人脸比对
HRESULT FacePairMatching(MFloat &confidenceLevel, ASF_FaceFeature feature1, ASF_FaceFeature feature2,
    ASF_CompareModel compareModel = ASF_LIFE_PHOTO);
//设置活体阈值
HRESULT SetLivenessThreshold(MFloat rgbLiveThreshold, MFloat irLiveThreshold);
//人脸属性检测（年龄、性别、活体、3D角度）
HRESULT FaceProcess(ASF_MultiFaceInfo detectedFaces, IplImage *img, ASF_AgeInfo &ageInfo,
    ASF_GenderInfo &genderInfo, ASF_Face3DAngle &angleInfo, ASF_LivenessInfo& liveNessInfo);
//活体检测
HRESULT livenessProcess(ASF_MultiFaceInfo detectedFaces, IplImage *img, ASF_LivenessInfo& livenessInfo, bool isRgb);
```

3.2 人脸注册模块(IMAGE模式)

```
575
576 //注册单张人脸
577 +void ArcFaceDetection::registerSingleFace() { ... }
604
605 // 人脸库操作
606 +void ArcFaceDetection::registerFaceDatebase() { ... }
624
625 +void ArcFaceDetection::showRegisterThumbnail() { ... }
687
688 +void ArcFaceDetection::clearFaceDatebase() { ... }
708
```

3.3 识别照模块(IMAGE模式)

```
709
710 // 识别照操作
711 +void ArcFaceDetection::selectRecognitionImage() { ... }
774
775 +void ArcFaceDetection::detectionRecognitionImage(IplImage* recognitionImage, ASF_SingleFaceInfo& faceInfo, QString& strInfo) { ... }
829
830 +void ArcFaceDetection::scaleFaceRect(MRECT srcFaceRect, MRECT* dstFaceRect, double nScale) { ... }
837
```

3.4 人脸比对(IMAGE模式)

```
837
838 //人脸比对模块
839 +void ArcFaceDetection::faceCompare() { ... }
872
```

3.5 摄像头模块

```
121
122 // 摄像头模块
123 +void ArcFaceDetection::openCamera() { ... }
202
203 +void ArcFaceDetection::closeCamera() { ... }
209
210 +void ArcFaceDetection::operationCamera() { ... }
250
```

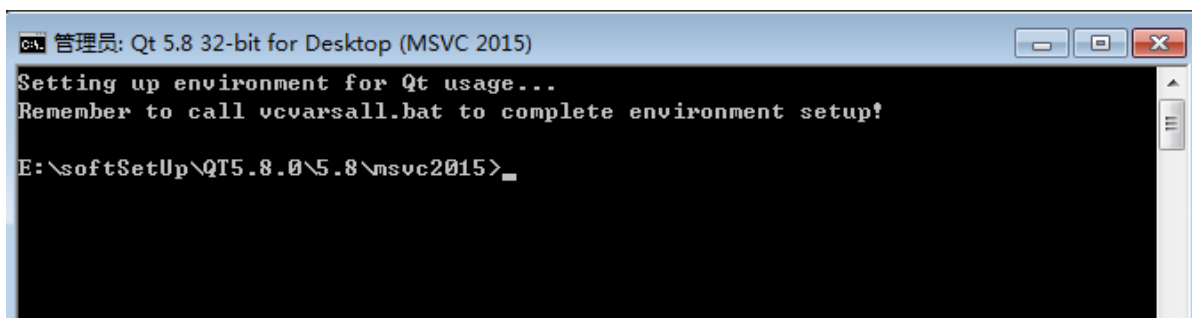
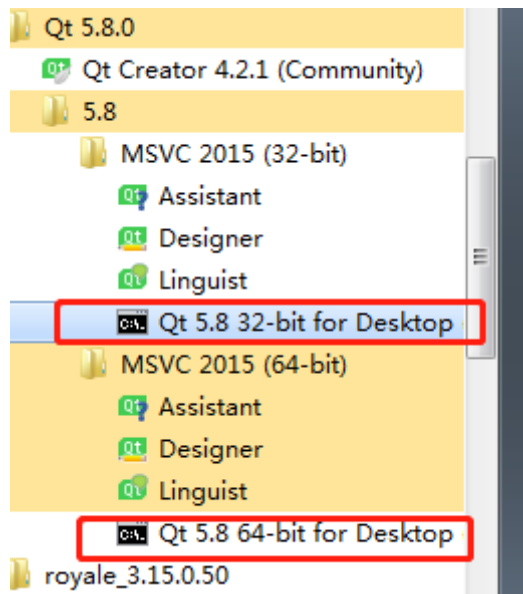
3.6 视频流模块(VIDEO模式)

```
250
251 //ft预览
252 +void ArcFaceDetection::ftPreviewData() { ... }
357
358 //特征提取、比对
359 +void ArcFaceDetection::frPreviewData() { ... }
440
441 //活体检测
442 +void ArcFaceDetection::livenessPreviewData() { ... }
556
```

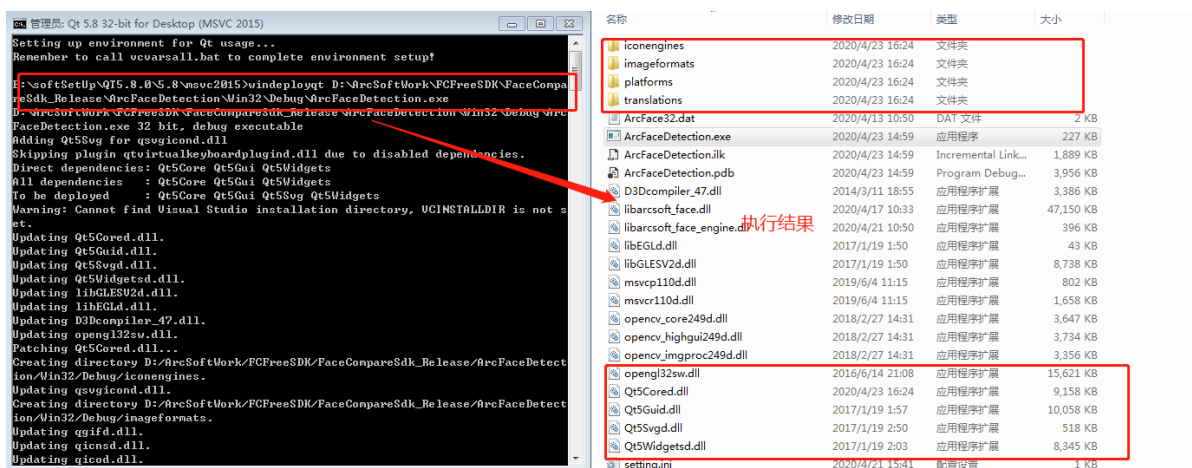
4 Demo 打包

4.1 提取Qt的动态链接库

1. 打开Qt自带的 windeployqt 工具，如下（因为我VS用的编译器是MSVC2015，所以选择它）



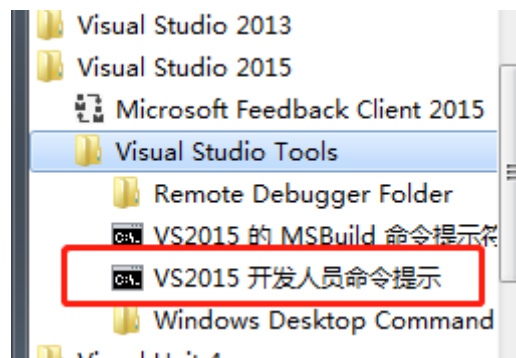
2. 使用 windeployqt 工具命令：“windeployqt 文件名.exe”，回车键:



Qt动态链接库提取成功。但只是这样做不一定在其他电脑上运行，所以有接下来的步骤。

4.2 检查缺少的DLL文件

1. 打开vs2015目录下的 vs2015 开发人员命令提示 工具；



2. 进入源工程所在目录（这次进入的目录为vs创建工程生成exe的目录）；

下面两种方法，第一种直接在控制台中列出所依赖的库，很方便。第二种则输出到指定的文件中，信息输出更详细。

- 使用命令：dumpbin /dependents 文件名.exe

```
管理员: VS2015 开发人员命令提示

C:\Program Files (x86)\Microsoft Visual Studio 14.0>
C:\Program Files (x86)\Microsoft Visual Studio 14.0>dumpbin /dependents D:\ArcSoftWork\FCFreeSDK\FaceCompareSdk_Release\ArcFaceDetection\Win32\Debug\ArcFaceDetection.exe
Microsoft (R) COFF/PE Dumper Version 14.00.24215.1 命令
Copyright (C) Microsoft Corporation. All rights reserved.

Dump of file D:\ArcSoftWork\FCFreeSDK\FaceCompareSdk_Release\ArcFaceDetection\Win32\Debug\ArcFaceDetection.exe

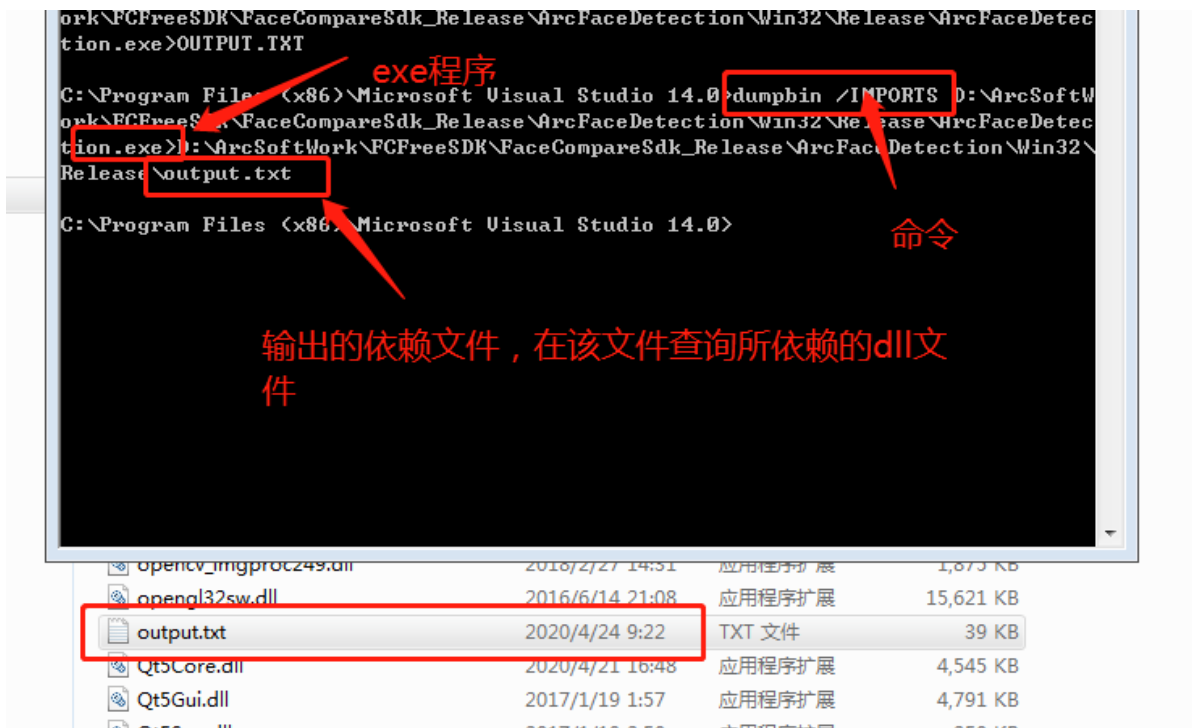
File Type: EXECUTABLE IMAGE

Image has the following dependencies:

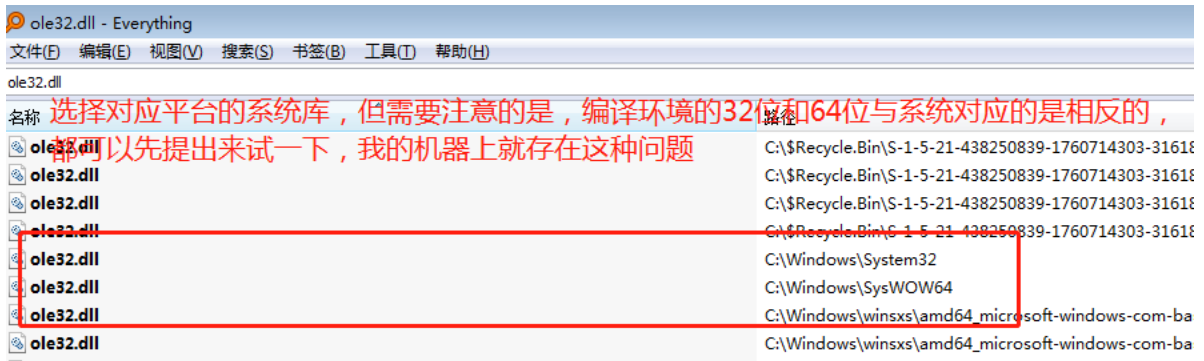
opencv_core249d.dll
opencv_highgui249d.dll
opencv_imgproc249d.dll
Qt5Cored.dll
Qt5Guid.dll
Qt5Widgets.dll
KERNEL32.dll
ole32.dll
OLEAUT32.dll
MSUCP140D.dll
libarcsoft_face_engine.dll
UCRUNTIME140D.dll
ucrtbased.dll
SHELL32.dll

Summary
```

- 使用命令：dumpbin /IMIMPORTS 文件名.exe>output.txt



3. 比如上面显示的是ole32.dll，然后我使用 everything 软件寻找ole32.dll文件



4.3 最终打包的文件

该结果是Release版本下打包的，Debug下的结果可能会有差异，但是只要按照上述步骤执行即可。

| | | | |
|-----------------------------------|-----------------|--------|-----------|
| iconengines | 2020/4/22 20:02 | 文件夹 | |
| imageformats | 2020/4/22 20:02 | 文件夹 | |
| platforms | 2020/4/22 20:02 | 文件夹 | |
| translations | 2020/4/22 20:02 | 文件夹 | |
| api-ms-win-crt-heap-l1-1-0.dll | 2017/1/18 23:35 | 应用程序扩展 | 13 KB |
| api-ms-win-crt-locale-l1-1-0.dll | 2017/1/18 23:35 | 应用程序扩展 | 12 KB |
| api-ms-win-crt-math-l1-1-0.dll | 2017/1/18 23:35 | 应用程序扩展 | 22 KB |
| api-ms-win-crt-runtime-l1-1-0.dll | 2017/1/18 23:35 | 应用程序扩展 | 16 KB |
| api-ms-win-crt-stdio-l1-1-0.dll | 2017/1/18 23:35 | 应用程序扩展 | 18 KB |
| api-ms-win-crt-string-l1-1-0.dll | 2017/1/18 23:35 | 应用程序扩展 | 18 KB |
| ArcFaceDetection.exe | 2020/4/22 11:43 | 应用程序 | 99 KB |
| ArcFacePro32.dat | 2020/4/21 19:34 | DAT 文件 | 2 KB |
| D3Dcompiler_47.dll | 2014/3/11 18:55 | 应用程序扩展 | 3,386 KB |
| kernel32.dll | 2018/3/26 11:51 | 应用程序扩展 | 1,088 KB |
| libarcsoft_face.dll | 2020/4/17 10:33 | 应用程序扩展 | 47,150 KB |
| libarcsoft_face_engine.dll | 2020/4/21 10:50 | 应用程序扩展 | 396 KB |
| libEGL.dll | 2017/1/19 1:50 | 应用程序扩展 | 14 KB |
| libGLESV2.dll | 2017/1/19 1:50 | 应用程序扩展 | 1,937 KB |
| msvcp110d.dll | 2019/6/4 11:15 | 应用程序扩展 | 802 KB |
| msvcp140.dll | 2018/4/4 22:41 | 应用程序扩展 | 452 KB |
| msvcr110d.dll | 2019/6/4 11:15 | 应用程序扩展 | 1,658 KB |
| ole32.dll | 2017/4/17 23:12 | 应用程序扩展 | 1,385 KB |
| oleaut32.dll | 2017/4/17 23:12 | 应用程序扩展 | 568 KB |
| opencv_core249.dll | 2018/2/27 14:31 | 应用程序扩展 | 2,086 KB |
| opencv_highgui249.dll | 2018/2/27 14:31 | 应用程序扩展 | 2,097 KB |
| opencv_imgproc249.dll | 2018/2/27 14:31 | 应用程序扩展 | 1,875 KB |
| opengl32sw.dll | 2016/6/14 21:08 | 应用程序扩展 | 15,621 KB |
| output.txt | 2020/4/24 9:22 | TXT 文件 | 39 KB |
| Qt5Core.dll | 2020/4/21 16:48 | 应用程序扩展 | 4,545 KB |
| Qt5Gui.dll | 2017/1/19 1:57 | 应用程序扩展 | 4,791 KB |
| Qt5Svg.dll | 2017/1/19 2:50 | 应用程序扩展 | 250 KB |
| Qt5Widgets.dll | 2017/1/19 2:03 | 应用程序扩展 | 4,340 KB |
| setting.ini | 2020/4/21 15:41 | 配置设置 | 1 KB |
| shell32.dll | 2017/5/10 23:12 | 应用程序扩展 | 12,579 KB |
| vcruntime140.dll | 2018/4/4 22:41 | 应用程序扩展 | 86 KB |

4.4 疑难点

项目若在 Release 模式和 Debug 模式下能够正常运行，但是单独点击项目生成目录下的 Release 文件夹或者 Debug 文件夹下的 exe 文件，出现以下错误，说明是缺少 DLL 文件，可以利用 Depends.exe 查询所缺 DLL。