

ZAI 编译指南

目录

Delphi 编译	2
Delphi Demo 的编译	4
如果电脑不支持 cuda 怎么办?	7
如果你的电脑没有 cuda 支持, 你会遇到的问题	7
怎样构建手机平台的 zAI 人脸应用程序	8
怎样构建 IOT 平台的 zAI 人脸应用程序	8
内核编译 zAI_Cuda_x64 (略过)	8
内核编译 zAI_x64 (略过)	8
zAI 工具链编译.....	9
zDefine.inc 编译开关说明.....	10
编译模式中的 Release 和 Debug 差异.....	11

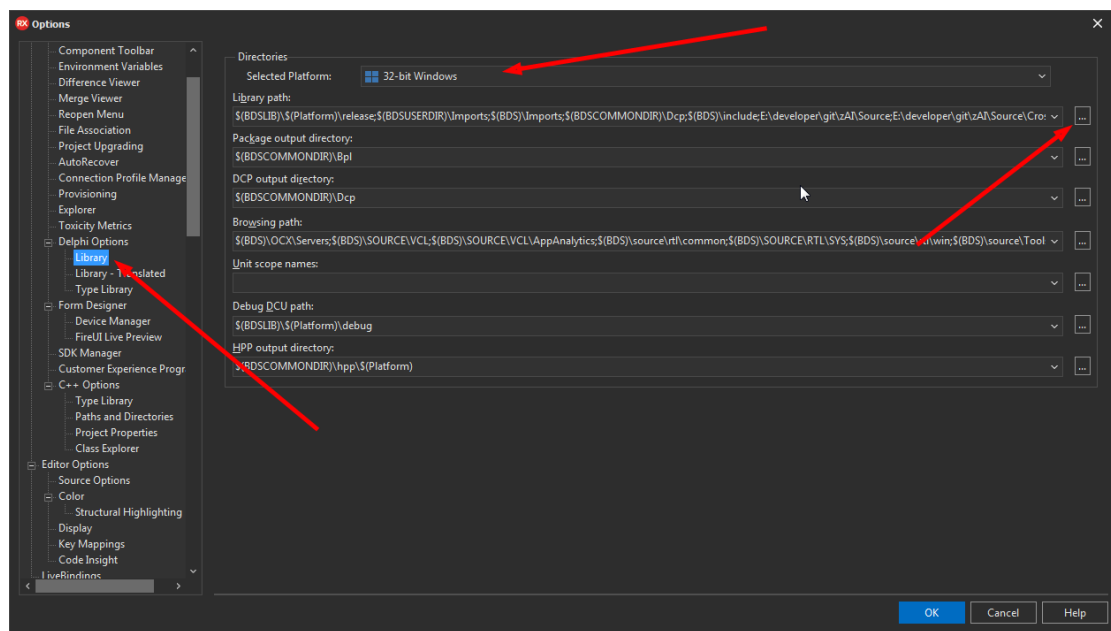
Delphi 编译

打开 Options

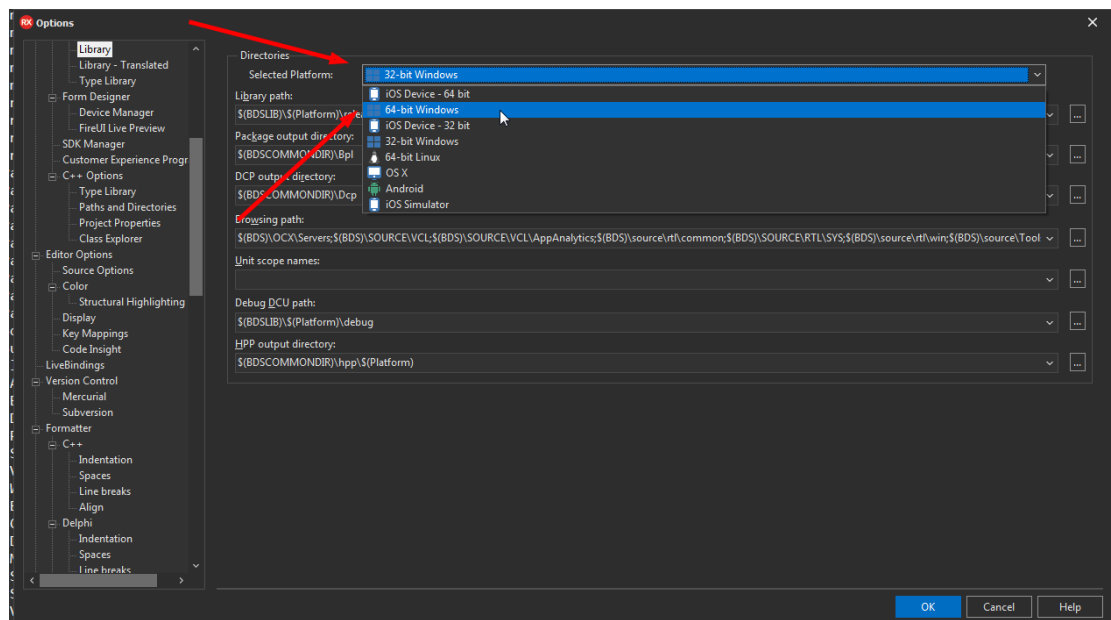
切换到 Library 配置页

先选你需要编译的目标平台

接下来编辑路径

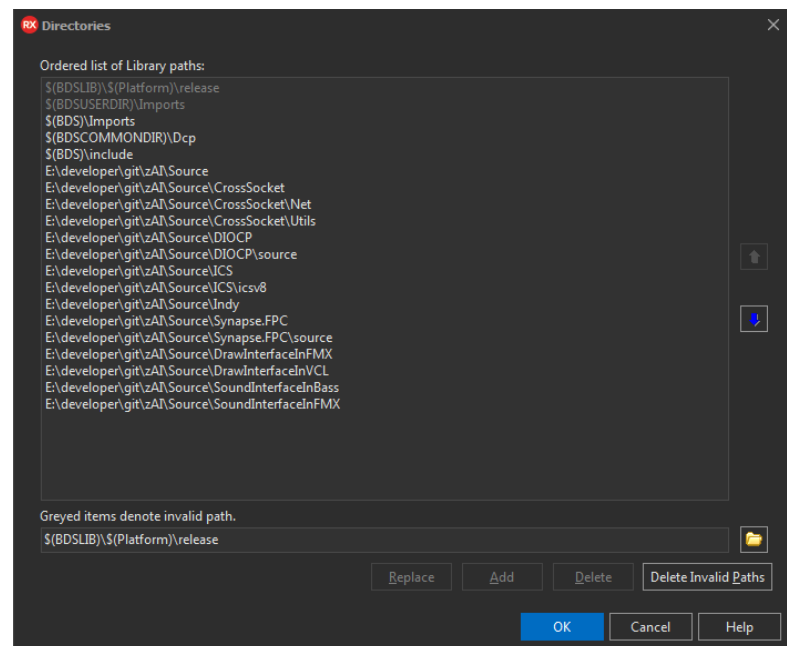


ZAI 的 Core 部分编译（使用 Cuda）必须是 64 位 Windows，首选设置 64 位 Windows 平台

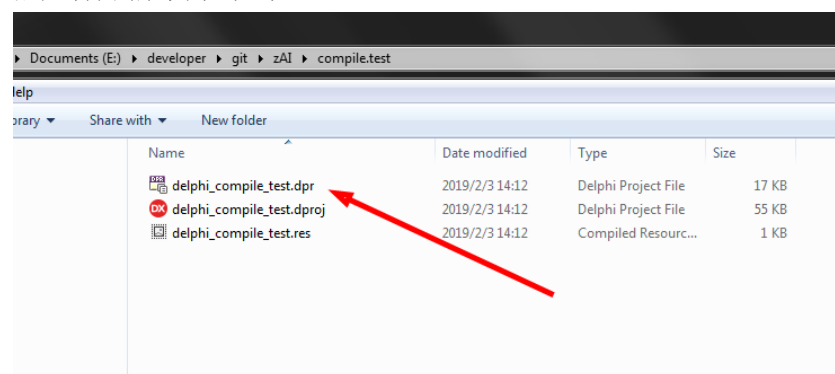


如果之前设置了 zServer4D, zAnalysis 等等项目的 Library Path, 需要使用 zAI 的库来代替。

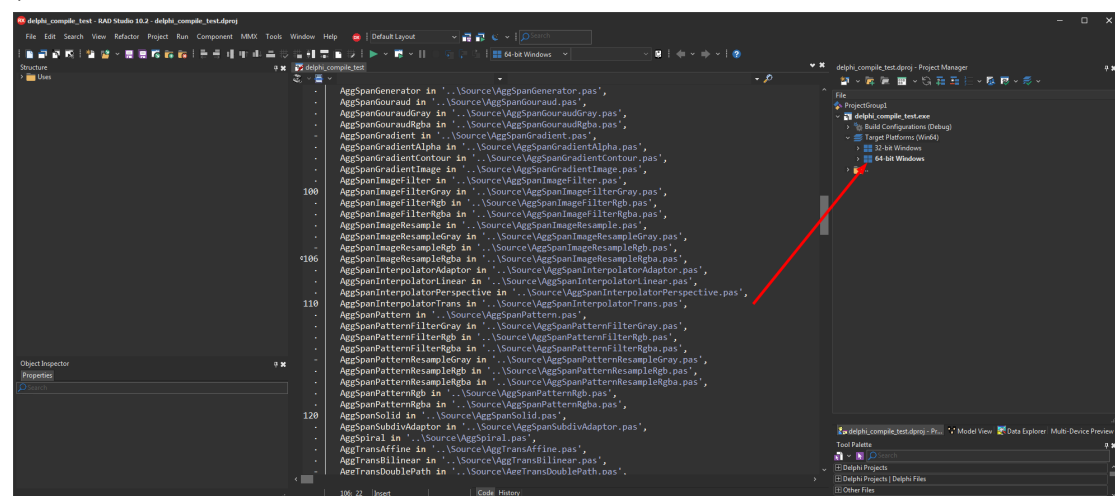
然后在 Library Path 中，要把 zAI 工程 Source 下的目录都添加进来，如果是多平台每个平台都要动手添加



然后打开编译测试工程



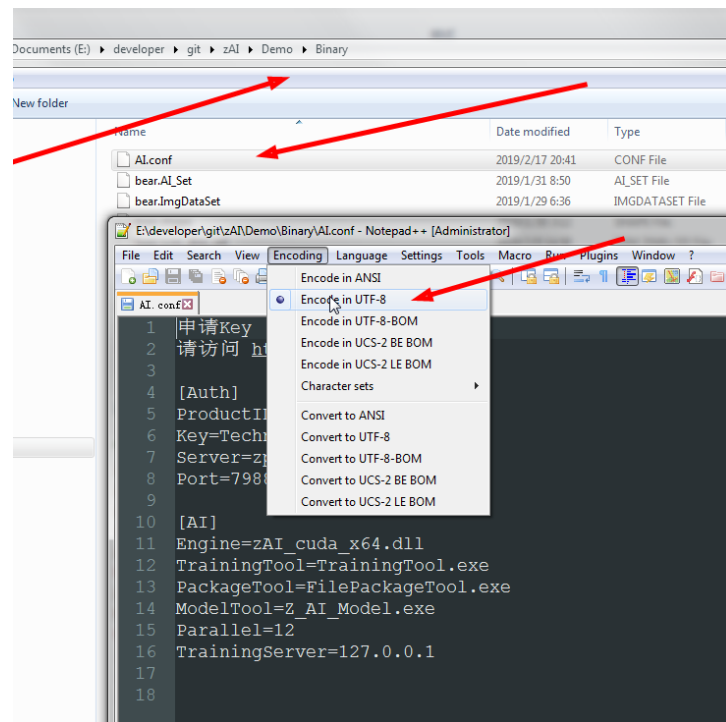
将目标切换到 64 bit windows，编译通过即表示可以正常使用
时至 2019-2 月，zAI 的内核库工程代码不包括 Demo 和工具，大概在 80-90 万行间，请自行检查



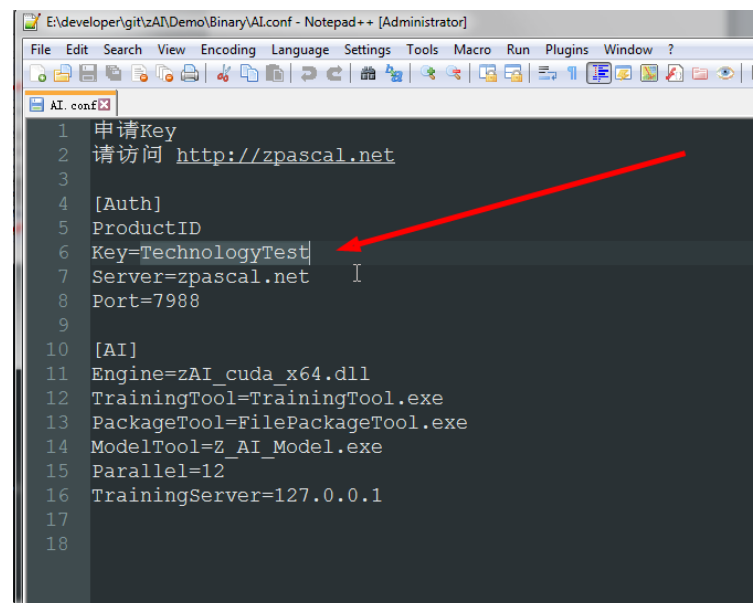
Delphi Demo 的编译

首先请用编译测试工程编译一次，都通过再来编译 Demo

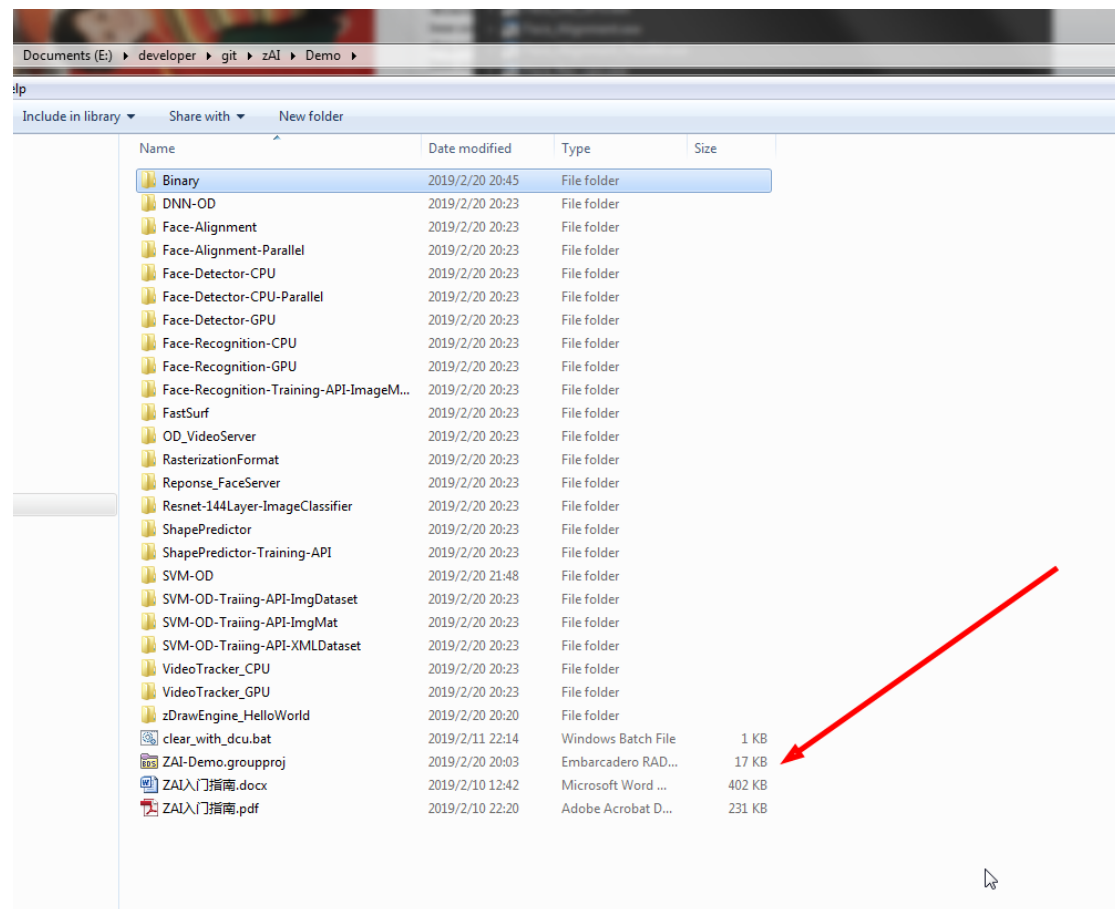
首先，到 zAI\Demo\Binary 目录打开 AI.conf
这个文本文件需要使用 utf8 编码



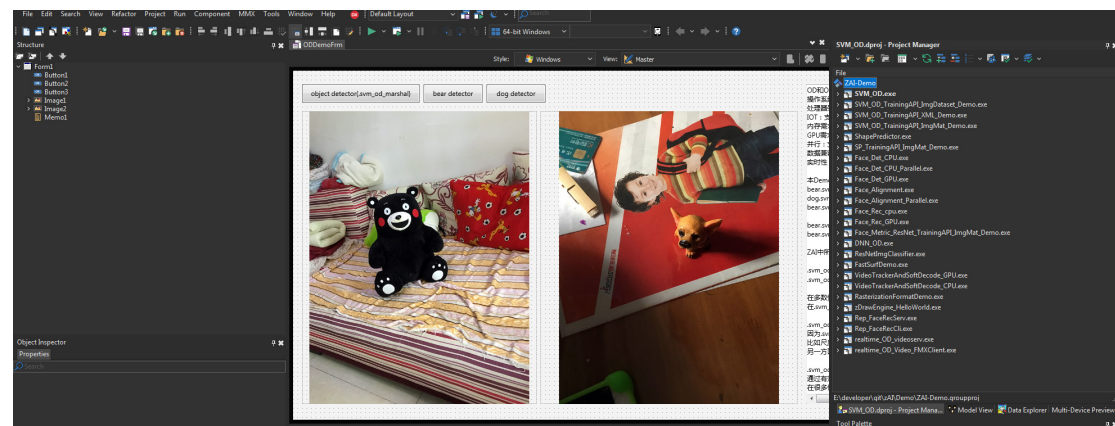
然后将领取到的 zAI 评估 Key 填入，保存



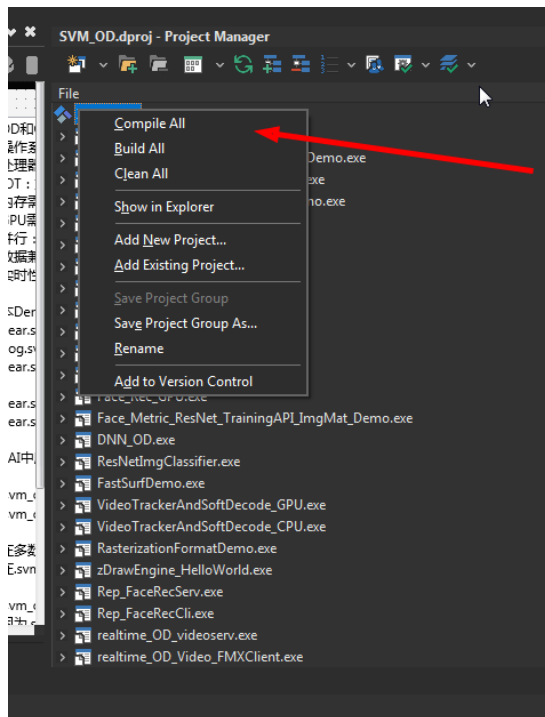
使用 Delphi XE10.2 以后的版本打开 ZAI-Demo.groupproj



将目标平台全部切换至 64 bit windows
开始全部构建一次



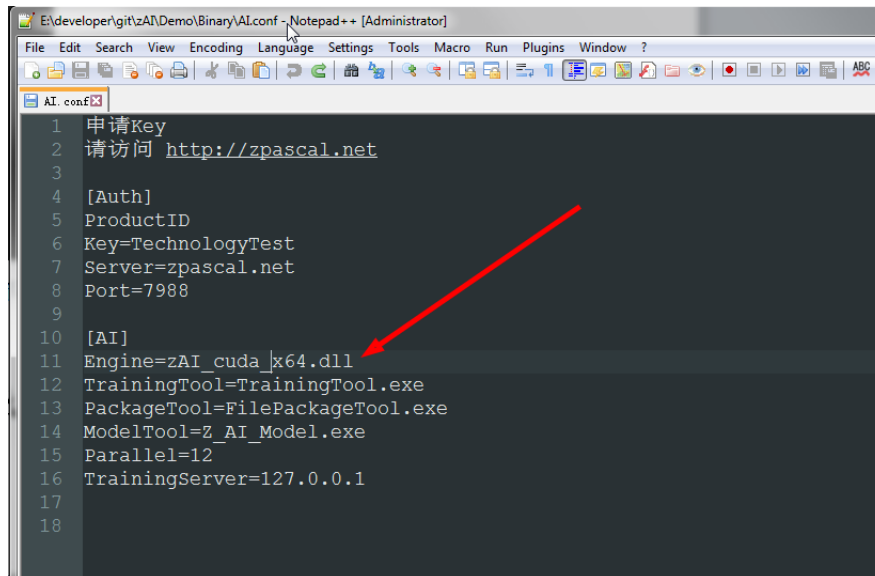
全部构建一次



这些 demo 全部编译完大概 2000 万行，不出问题，5 分钟以后完成

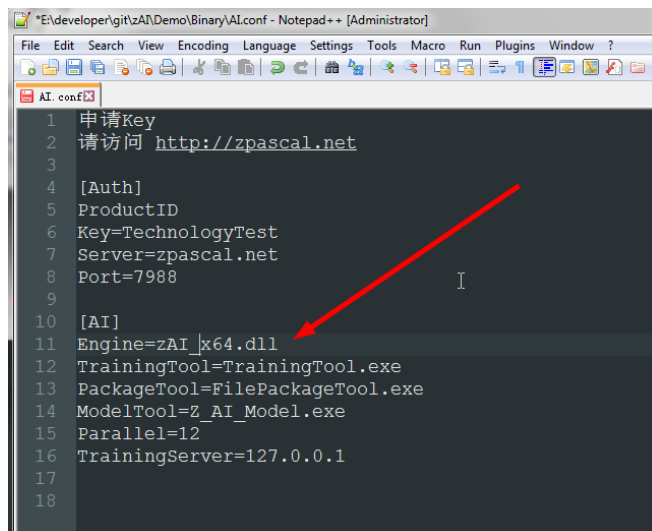
如果电脑不支持 cuda 怎么办？

打开 AI.conf



```
1 申请Key
2 请访问 http://zpascal.net
3
4 [Auth]
5 ProductID
6 Key=TechnologyTest
7 Server=zpascal.net
8 Port=7988
9
10 [AI]
11 Engine=zAI_cuda_x64.dll
12 TrainingTool=TrainingTool.exe
13 PackageTool=FilePackageTool.exe
14 ModelTool=Z_AI_Model.exe
15 Parallel=12
16 TrainingServer=127.0.0.1
17
18
```

将 zAI_cuda_x64.dll 改为 zAI_x64.dll



```
1 申请Key
2 请访问 http://zpascal.net
3
4 [Auth]
5 ProductID
6 Key=TechnologyTest
7 Server=zpascal.net
8 Port=7988
9
10 [AI]
11 Engine=zAI_x64.dll
12 TrainingTool=TrainingTool.exe
13 PackageTool=FilePackageTool.exe
14 ModelTool=Z_AI_Model.exe
15 Parallel=12
16 TrainingServer=127.0.0.1
17
18
```

如果你的电脑没有 cuda 支持，你会遇到的问题

- 1, Gpu 的训练将会慢到无法忍受
- 2, zAI 中的 MMOD,RNIC,MericDNN 三大核心模块将会无法使用

怎样构建手机平台的 zAI 人脸应用程序

- 1, 按 delphi 编译指南部分, 设置好 Android 下各个库的引用路径, 路径注意不要用中文, clang 的编译体系不支持中文路径
- 2, 将 zAI\Demo\Reponse_FaceServer\zAI_Reponse_FaceClient.pas 引用至你的工程目录中
- 3, 按 zAI\Demo\Reponse_FaceServer\Rep_FaceRecCli.dproj 使用范式使用 TFaceClient 即可

怎样构建 IOT 平台的 zAI 人脸应用程序

- 1, 入手 IOT 的[完全攻略.pdf](#) 文档
- 2, 按文档去做, 自行在 iot 搭建好调试+编译平台 (Lazarus+FPC)
- 3, 在 Lazarus 正确编译 zAI 库你还需要以下两份文档
 - a) Linux 桌面开发指南.pdf
 - b) 在 Lazarus 或则 CodeTyphon 编译时出现缺失 mtpocs 库的解决办法.pdf
- 4, 将 zAI\Demo\Reponse_FaceServer\zAI_Reponse_FaceClient.pas 引用至你的工程目录中
- 5, 按 zAI\Demo\Reponse_FaceServer\Rep_FaceRecCli.dproj 使用范式使用 TFaceClient 即可

内核编译 zAI_Cuda_x64 (略过)

cuda 内核构建需要使用 vc2015sp3.1 + cuda10.2sdk + openblas, 这一部分过于复杂, 就此略过

内核编译 zAI_x64 (略过)

X64 内核构建需要使用 vc2015sp3.1 + openblas, 这一部分过于复杂, 就此略过

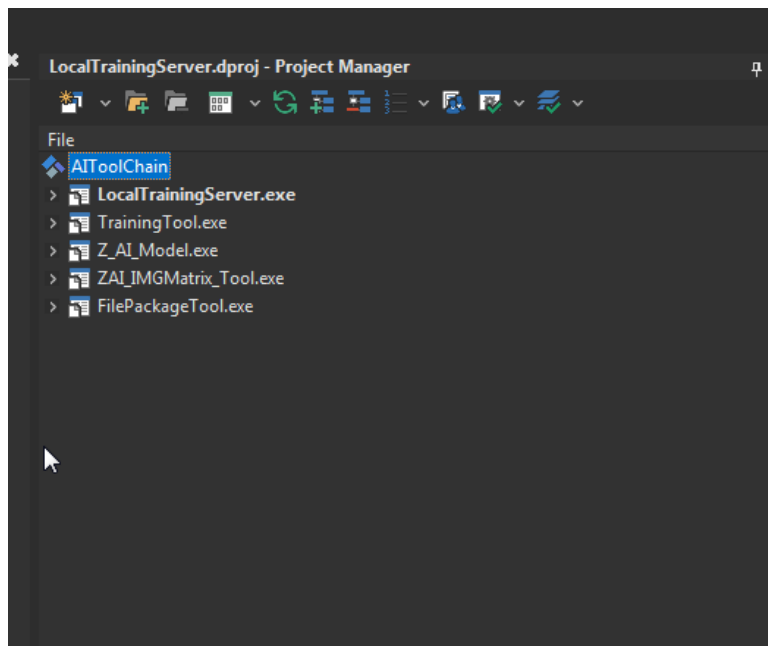
zAI 工具链编译

zAI 的工具链都使用 Delphi 编写，这一部分不对非授权用户开放

构建 zAI 1.2 的工具链必须使用 XE10.3.2 以上的版本，早期版本会出现 Keybind 的 bug

编译前请按 [Delphi 编译指南](#) 部分的内容正确设置环境

打开 AIToolChain.groupproj



LocalTrainingServer: 训练服务器，可以多个任务同时一起训练，并且能保存和还原训练进度

TrainingTool: 训练服务器使用的 Shell 工具，因为训练都长时间运行，服务器避免了在自身 VM 环境(windows 一个 app 启动，都使用虚拟环境，相当于一个 app 对应一个 vm 在运行)，每次训练，都使用 TrainingTool 开辟一个独立 vm 运行，待训练完成，取结果用。

Z_AI_Model: 小规模建模编辑器工具，用于编辑和组织生成数据集，它可以处理以下数据集
ImgDataset，训练数据集

AISet，编辑器数据集

XML，通用数据集

ZAI_IMGMatrix_Tool, 大规模数据集编辑工具，图片样本高于 1000 张的分类和批处理用

FilePackageTool, zAI 的训练都是打包格式，比如我训练时输入 10 个文件，再输出 10 个文件，这些文件都可以使用 FilePackageTool 在进行打包，编辑，提取

zDefine.inc 编译开关说明

zDefine.inc 在 zAI\source 目录可以找到它，这是一个全局的编译代码库

parallel: 如果打开，会启动并行处理，假如在 IOT 和手机这类平台运行程序，这个开关可以关闭，因为多数时候手机并不会让并行程序有明显提升

Z_AI_Dataset_Build_In: 如果打开，数据集会自动编译到 exe 中，不会启动时跑去读取，zAI_BuildIn.OXC(压缩后 70M)的那个文件了

LARGE_RASTER_FONT: MemoryRaster 是内核光栅系统，它不使用操作系统的字库，像素，全部自己内置来处理，这个开关表示，光栅字体采用亚洲字符集支持（日韩体），默认情况下 MemoryRaster 的字库只支持简繁体中文+英文字库，日韩编码的字符不会在 MemoryRaster 中显示。

ZDB_BACKUP: ZDB 是内核数据库，这个开关表示 ZDB 每次打开和保存数据库，都会创建一个文件副本，以保证数据存储安全

CriticalSimulateAtomic: 如果打开会屏蔽掉系统内置的原子锁，改用互斥锁来模拟原子锁，这个开关主要用于多线程访问数据控制。

SoftCritical: 使用软件来模拟原子锁，假如 CriticalSimulateAtomic+ SoftCritical 都被打开了，性能也会因此掉一大截，这两个开关主要用于手机，IOT 这类设备上多线程对数据的访问

编译模式中的 Release 和 Debug 差异

Release 拥有不受约束的内存访问权，关闭指针越界检查

Release 拥有 Inline 修饰符

Debug 受指针越界限制，假如我们要访问高于 2G 内存的数据，用 Debug 将会收到越界异常，

Debug 是用于调试程序问题而生的变异模式

Debug 都没有 Inlien 修饰符

By.qq600585

2019-2