

开发工具

visual studio 2017

开发环境

win 10 x64 系统，已经安装微软运行库（百度搜索微软运行库）

onnxruntime x64 cpu 版本，至少在 1.10 之上（可不考虑，已经集成在工程中，sharedlib）

opencv x64，（可不考虑，已经集成在工程中，sharedlib）

文件说明

rnetonnxdll_win10：动态链接库源码工程文件夹

sharedlib：第三方库，包含 opencv，onnxruntime

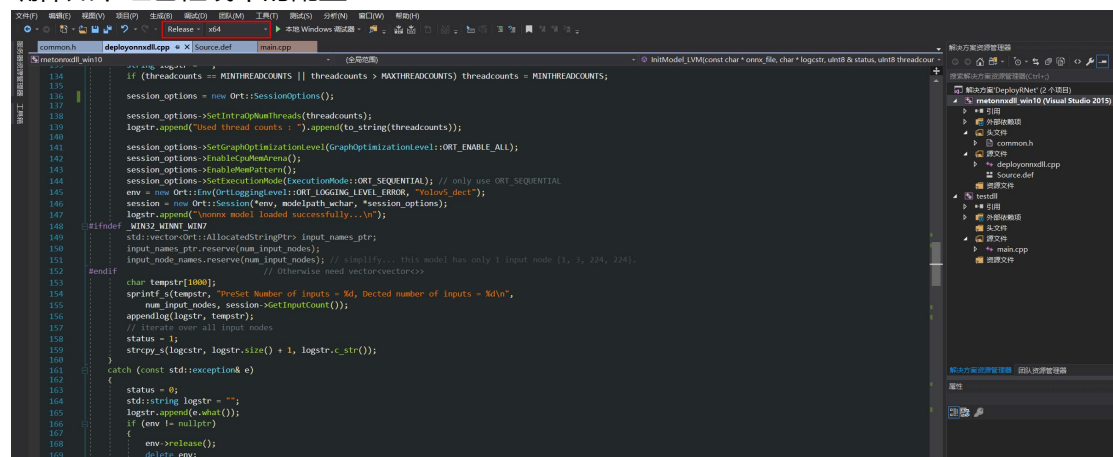
testdll：测试动态链接库，包含动态链接库的调用方法，推荐看看调用方式！

x64：已经编译好的动态库和测试程序；

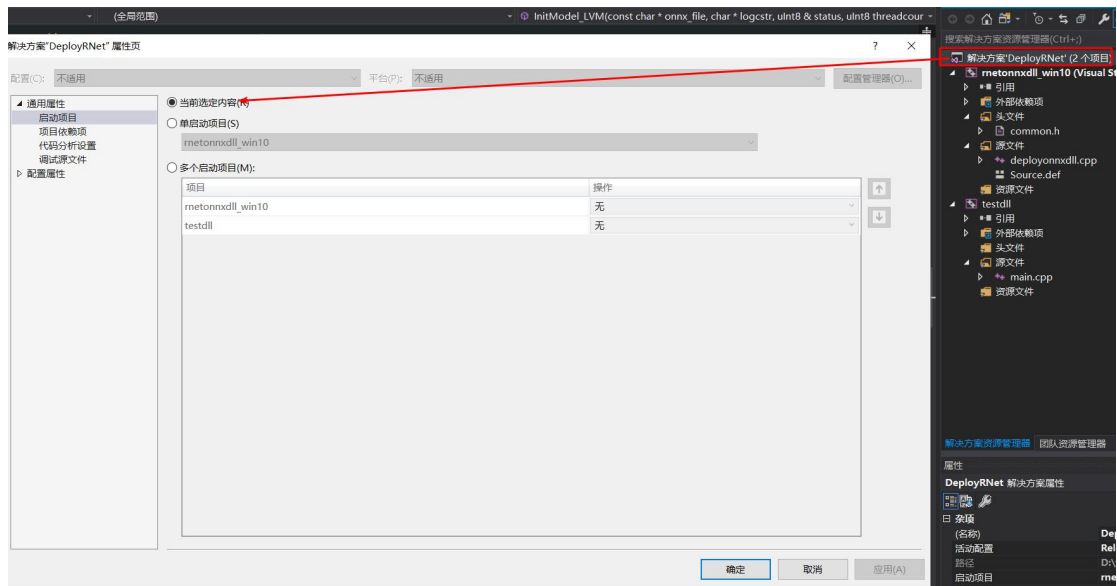
编译说明

vs2017 打开工程文件 DeployRNet.sln；

确保如下红色框线中的配置：



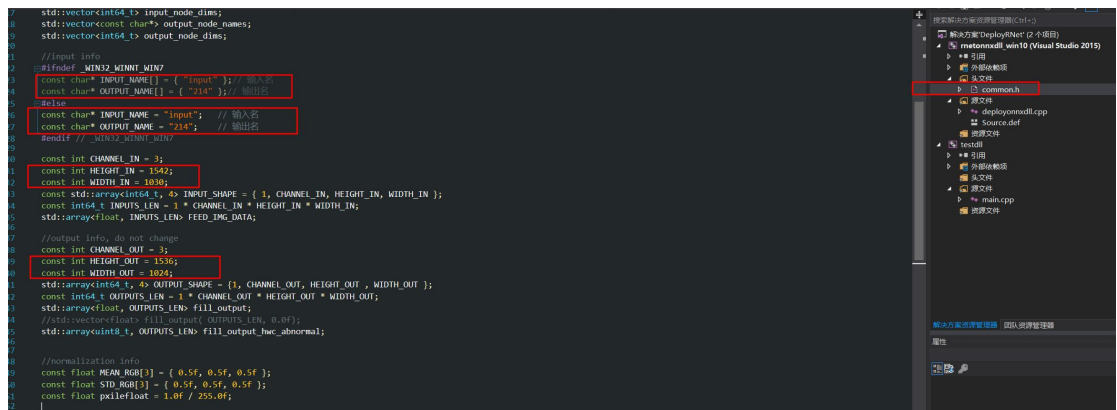
鼠标右击解决方案，进行如下设置：



该解决方案中存在两个项目，一个是动态库编译项目，一个是 test 的项目，用于测试 dll 动态链接库文件，分别进行编译；

文档说明

common.h：保存模型的基本配置：输入输出名，输入输出大小



编译前确保上述红色框线中的值与模型参数信息一致(通过 <https://netron.app/> 可视化模型参数信息)

Source.def：声明哪些函数要导出到 dll 动态链接库中

deployonnx.dll.cpp 动态链接库主要实现逻辑

主要导出如下函数，函数出现的先后顺序为调用顺序：

```
void InitModel_LVM( const char* onnx_file, char *logcstr, UInt8 &status, UInt8 threadcounts, UInt8 isShowModelInfo)
```

初始化模型函数

onnx_file: onnx 模型文件的路径

logcstr: 日志字符串

status: 模型初始化状态, 1 正常, 0 异常

后两个参数可忽略, 默认分别使用 2, 1

```
void CheckSession_LVM(char *logcstr, uInt8 &status)
```

检查模型环境函数

logcstr: 日志字符串

status: 模型环境检查状态, 1 正常, 0 异常

```
int Infer_LVM(char *src_img_path, int isRGBsrc)
```

输入图像推理函数

src_img_path: 图像的路径

isRGBsrc: 图像读取后的格式, opencv 读取后的格式为 bgr, 因此该值为 0, 若为 rgb 则是 1

返回值为 0 表示正常

```
void GetOutput_LVM(char *save_filepath_abs)
```

保存推理的结果

save_filepath_abs: 保存的图片路径

调用流程

参考 testdll 项目实现流程, 先定义加载动态库函数的声明, 再加载动态库匹配函数, 再直接使用函数, 最后依次调用上述四个函数

注意

确保路径的正确性、避免路径中包含中文、确保模型的输入输出与 common.h 中的值保持一致。