

## BANGC 算子与风格迁移实验手册（下）

### 一、实时风格迁移在线推理与离线部署

#### 1. 实验内容：

- a) 模型量化：将单精度模型量化为 INT8 模型。
- b) 在线推理，完成 MLU 推理代码的移植工作，分别比较三种不同模型（原始模型、PowerDifference 算子模型，Numpy 算子模型）在 CPU 和 MLU 上的性能和精度。
- c) 离线推理，完成 MLU CNRT 离线推理代码，并完成部署测试。

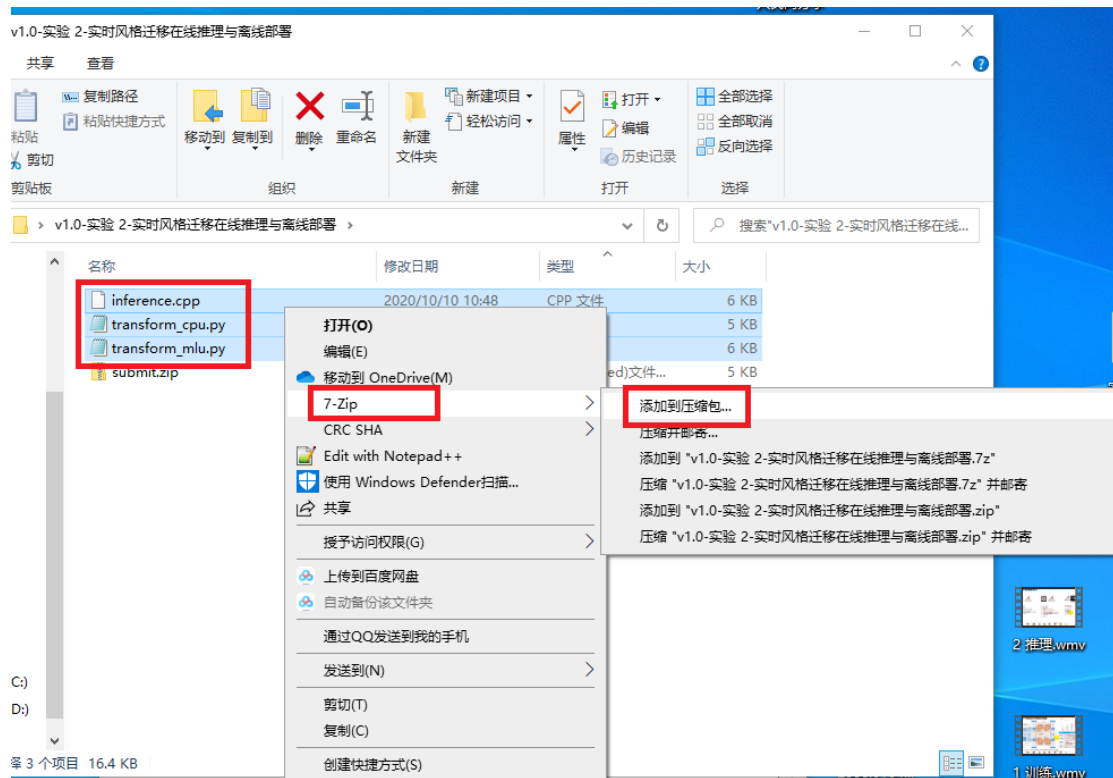
#### 2. 实验步骤：

- a) 模型量化：MLU270 对于 mlp 和 conv 等算子可使用 int 数据类型进行计算以提高效率，所以先要将原始的 float32 数据类型的 pb 模型量化成为 int 类型。在 /opt/AICSE-demo-student/demo/style\_transfer\_bcl/tools/fppb\_to\_intpb 目录中执行：python fppb\_to\_intpb.py udnie\_int8.ini。生成的模型会保存在 /opt/AICSE-demo-student/demo/style\_transfer\_bcl/models/int\_pb\_models 目录下。
- b) 在线推理程序：分别补全 /opt/AICSE-demo-student/demo/style\_transfer\_bcl/src/online\_mlu/transform\_mlu.py 和 /opt/AICSE-demo-student/demo/style\_transfer\_bcl/src/online\_cpu/transform\_cpu.py。比较多种不同推理方式的性能精度差异。
- c) 离线推理程序：在执行 MLU 在线推理时，通过配置文件生成并保存两种离线模型（原始模型、PowerDifference 算子模型）至 /opt/AICSE-demo-student/demo/style\_transfer\_bcl/models/offline\_models 文件夹下。补全 CNRT 离线推理模型代码：  
/opt/AICSE-demo-student/demo/

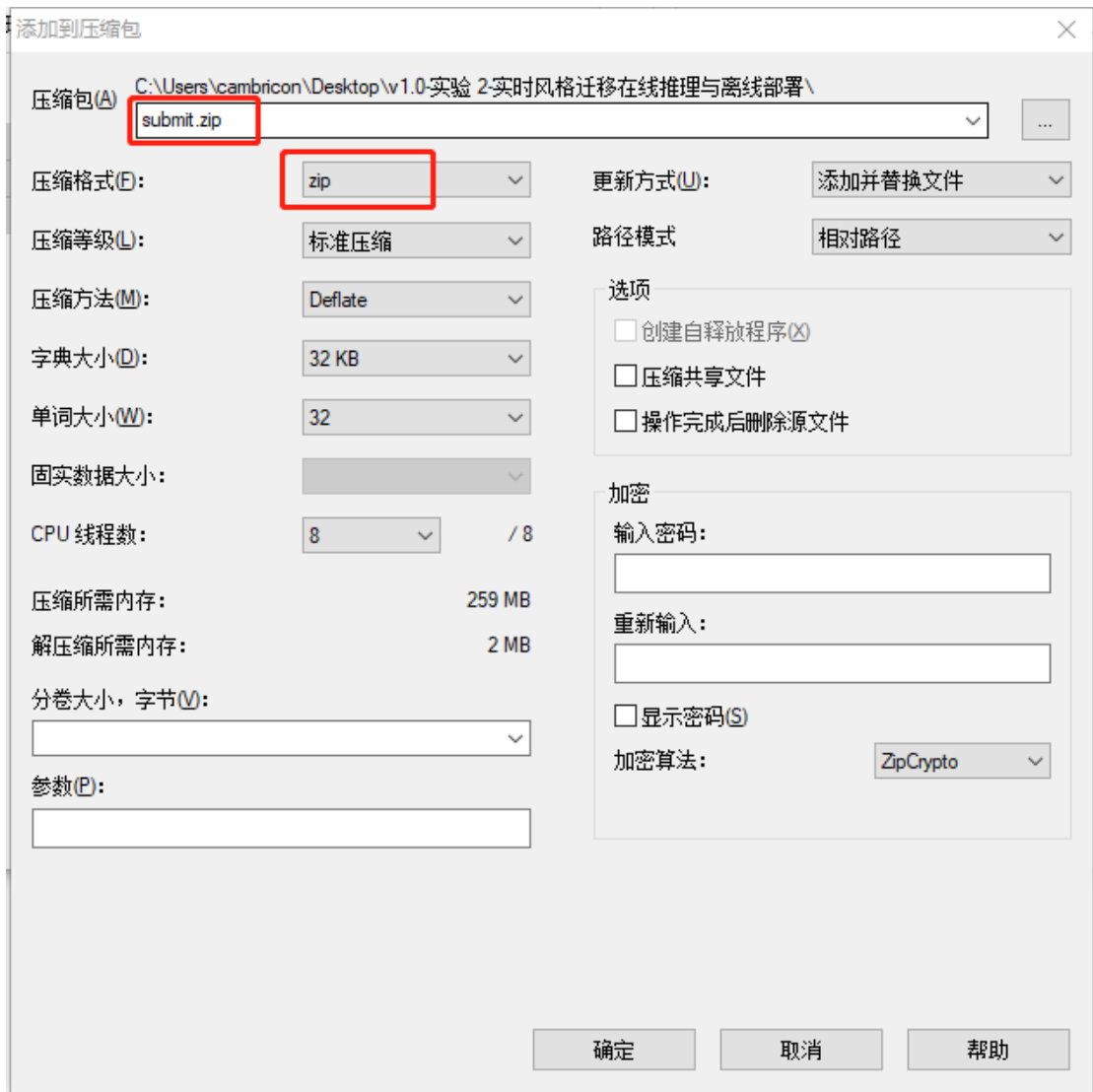
style\_transfer\_bcl/src/offline/src/inference.cpp 然后编译并执行离线推理。

### 3. 上传自动评测平台

- a) 将程序按以下文件列表整理，打包成压缩文件。请注意文件名和图中保持一致。



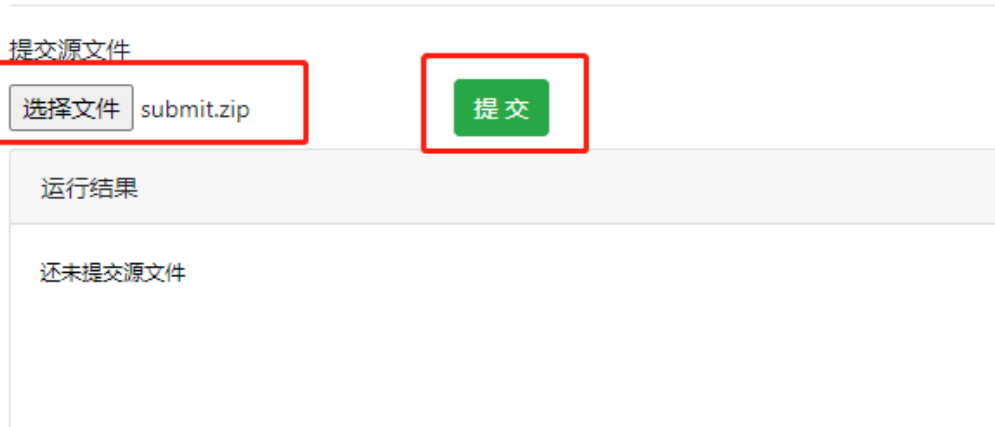
- b) 压缩文件格式为 zip 格式。文件名命名为 submit.zip



c) 将 submit.zip 上传到评测平台。选择文件并点击提交。

### 3. 评分标准

- 60分 标准：完成模型量化与TensorFlow在线推理程序，输出CPU端、MLU端 精度正确的风格迁移图片，输出
- 80分 标准：在60分的基础上，完成离线推理程序的编写，执行离线推理时 风格迁移图片精度正常；
- 100分标准：在80分基础上，执行离线推理时在图片精度正常的同时使用 BangC 实现 PowerDifference 等相比使用纯 CNML 算子库生成的离线模型，推理时延时更低，性能更好。



提交源文件

选择文件 submit.zip

提交

运行结果

尚未提交源文件

d) 上传之后请等待系统评测。

- 60分 标准：完成模型量化与TensorFlow在线推理程序，输出CPU端、MLU端 精度正确的风格迁移图片，输出
- 80分 标准：在60分的基础上，完成离线推理程序的编写，执行离线推理时 风格迁移图片精度正常
- 100分标准：在80分基础上，执行离线推理时在图片精度正常的同时使用 BangC 实现 PowerDif
- 相比使用纯 CNML 算子库生成的离线模型，推理时延时更低，性能更好。



提交源文件

选择文件 submit.zip

正在处理...

运行结果

下载源文件

已经成功提交，正在排队等待评判....., 前面还有4个评测任务。

e) 评测结束之后会自动显示评测结果。

提交源文件

选择文件 submit.zip

提交

运行结果

下载源文件

得分80.00 最后一次提交时间:2020-10-19 20:10:28

Accept

### CPU推理输出图

chicago\_udnie\_cpu



chicago\_udnie\_power\_diff\_cpu



chicago\_udnie\_power\_diff\_numpy\_cpu



### MLU推理输出图

chicago\_udnie\_int8\_mlu



chicago\_udnie\_int8\_power\_diff\_mlu



chicago\_udnie\_int8\_power\_diff\_numpy\_mlu

