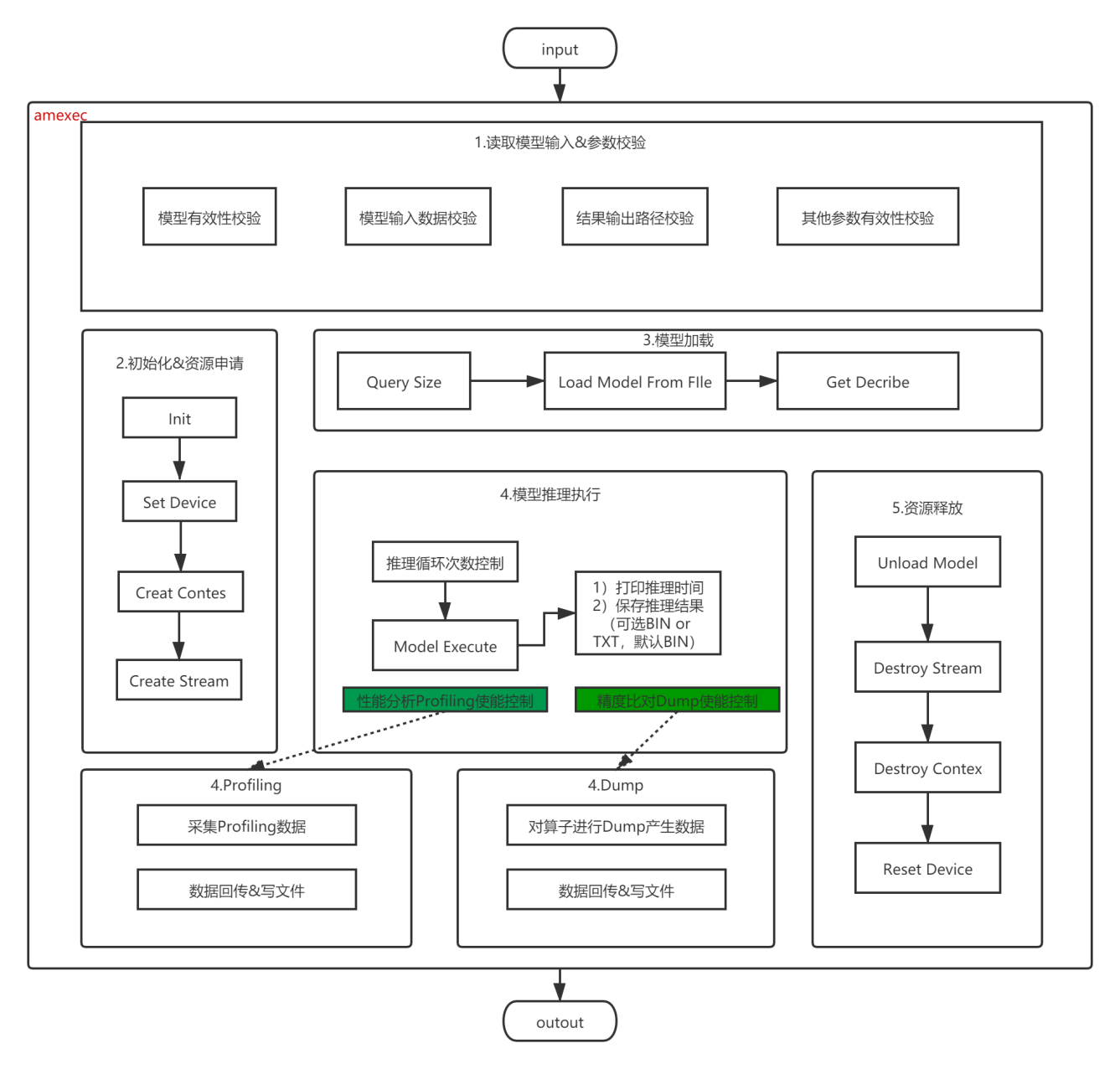
# 模型推理工具流程

## 方案设计



## 代码流程

### 2.1 读取模型输入&参数校验

模型只接受后缀名为.om的Ascend 离线模型作为输入

模型输入数据只接受后缀名为.bin的二进制格式数据作为输入，不接受直接将图片作为输入，视频流也需要经过处理，变成模型输入可接受的输入数据。用户需要提前通过transferPic\_batch.py生成bin文件。

输出路径如果不存在，尝试创建路径，创建失败则推出程序，用户需要保证此工具所属用户对输入路径有可读写权限

其他参数校验包括输出格式、循环次数等等。

2-1 参数校验

getopt\_long读参

Check模型输入路径、输出路径、其他参数有效性验证

IF校验不通过

报错并退出

ELSE

进入循环，第n次推理 n = 0 to loop\_num //一共循环loop\_num次，由用户指定

### 2.2 初始化&资源申请

通过调用acl接口进行初始化、申请创建Context、Stream

2-2 初始化&资源申请

aclInit初始化

aclSetDevice配置Device

aclrtCreateContext 创建Context

aclrtCreateStream 创建 Stream

aclrtGetRunMode 获取RunModel

### 2.3 模型加载、推理、输出

模型加载调用acl接口从给定路径加载模型，从指定模型输入数据读取数据送入模型推理。模型推理调用acl 推理接口aclmdExecute，在模型推理调用前后需要记录系统时间，同时在推理调用结束后，打印推理时间。推理完成后拿到output，通过调用aclmdlGetDatasetNumBuffers、aclmdlGetDatasetBuffer、aclGetDataBufferAddr、aclGetDataBufferSize接口将output数据拿出来写文件到用户指定的输出目录。每次推理创建一个基于时间戳的子目录，存放本次推理结果。如果模型有多个输出，需要将多个输出单独保存为文件，根据用户指定的格式，可选文本输出格式.txt，若未指定则默认保存为二进制.bin文件。

2-3 模型加载、推理、输出

LoadModelFromFileWithMem //加载模型

CreateDesc

CreateOutput

vector<void\*> picDevBuffer //读入模型输入数据，遍历所有

For i 0 to file.size

Do 加载输入数据

CreateInput

End

Execute 模型推理

OutputModelResult 结果输出

outputNum = aclmdlGetDatasetNumBuffers

For i 0 to outputNum

每个输出分别保存为文件

End

2.4 资源释放

当执行完输出结果保存后，进行资源释放并重置device方便后续使用

2-4 资源释放

DestroyResource

aclrtDestroyStream 释放stream

aclrtDestroyContext 释放context

aclrtResetDevice 重置device

End