Fibocom 广和通
完 美 无 线 体 验

图像超分辨率（real esrgan x4plus）案例

----基于 SC171 开发套件 V3

文档版本：V1.0
更新时间：2025 年 3 月 27 日

适用型号

序列	文档版本	适用型号	更新说明
1	V1.0	SC171 开发套件第三代	NA

目录

1 引言

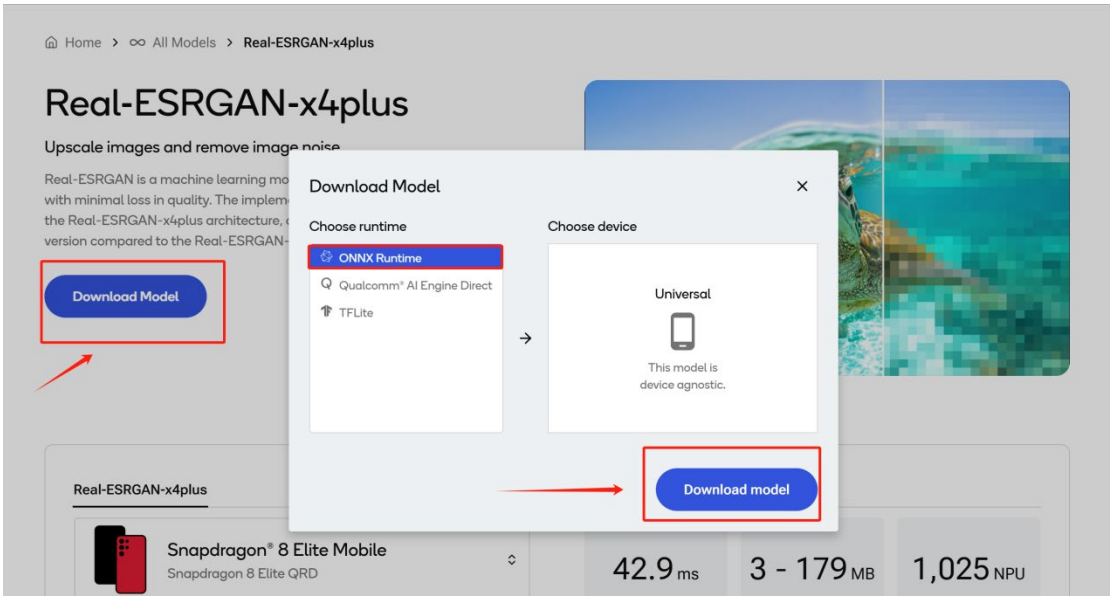
本指南以图像超分辨率（使用 Real ESRGAN X4Plus 模型）为例，详细演示如何在 SC171 开发套件 V3 上通过 Fibocom AI Stack 实现高效的深度学习模型推理。

若实践过程中遇到问题，请参考第 4 节“Q&A”或联系 Fibocom 技术支持团队获取帮助。

2 模型下载

点击进入模型链接 Real-ESRGAN-x4plus - Qualcomm AI Hub

选择 Download Model，可以选择三个模型类型下载，这里推荐选择 ONNX 格式



3 模型转化

详情请见《Fibo AI Stack 模型转化指南》，最后将转化好的 dlc 格式模型导入开发板中。

4 详细步骤

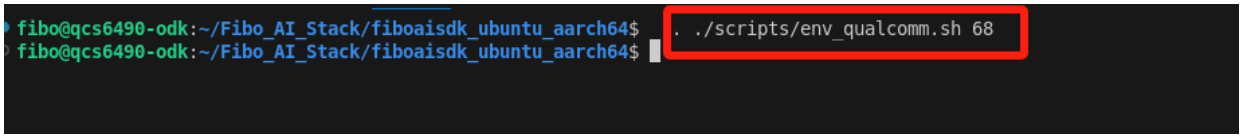
本案例所用到的工程源码见链接:

https://pan.baidu.com/s/1dIzDKX8qH_zVVlOd-W7mPw?pwd=msr6

4.1 准备

1. 进入 Ubuntu 的 UI 界面，打开开发板终端，调用脚本配置环境参数：

```
cd /home/fibo/Fibo_AI_Stack/fiboaisdk_ubuntu_aarch64
. ./scripts/env_qualcomm.sh 68
```



2. 导入所需的库，在终端输入以下命令：

```
pip3 install numpy pillow
```

3. 需要的参数：

- dlc_path：模型文件路径（.dlc 文件）
- img_path：输入图片路径
- 模型输入、输出张量名称与类型

4. 将本案例中的 test1.py、utils1.py 与 api_infer.py 放置在开发板的同一路径下

4.2 Python sample 代码

1. 主程序内容：

```python
# 导入依赖库
from api_infer import *          # Fibo AI推理SDK接口
from utils1 import (preprocess_for_realesrgan, postprocess_realesrgan)  #从本地模块 utils1.py 中，仅导入预处理和后处理函数
import numpy as np               # 数值计算库
from PIL import Image            # 图像处理库

#配置区（用户需修改部分）
dlc_path = "/home/fibo/Fibo_AI_Stack/SR/real_esrgan_x4plus.dlc"  # 模型文件路径(.dlc格式)
img_path = "/home/fibo/Fibo_AI_Stack/SR/0014.jpg"          # 测试图片输入路径
save_path = "/home/fibo/Fibo_AI_Stack/SR/result.jpg"       # 超分结果保存路径

#初始化SNPE推理引擎
snpe_ort = SnpeContext(dlc_path, [], Runtime.GPU, PerfProfile.BALANCED, LogLevel.INFO)
#初始化推理引擎，返回0表示成功
assert snpe_ort.Initialize() == 0  # 如果初始化失败会触发AssertionError

#图像预处理
img = Image.open(img_path)
img.show()
input = preprocess_for_realesrgan(img_path)

#执行模型推理
#准备输入数据字典（键名"image"需与模型输入节点名称匹配）
input_feed = {"image": input}
#输出层名称列表（空列表表示获取所有输出）
output_names = []
#执行推理，返回包含输出结果的字典
outputs = snpe_ort.Execute(output_names, input_feed)

#结果后处理
result_image = postprocess_realesrgan(outputs=outputs, save_path=save_path)
result_image.show()

#释放资源
assert snpe_ort.Release() == 0
```


2. utils1.py 文件内容：

Real-ESRGAN 模型的图像预处理函数：

深圳市广和通无线股份有限公司 | 总机: +86 755-26733555 传真: +86 755-26520841
邮箱: market@fibocom.com 官网: www.fibocom.com.cn

参数:img_path(str): 输入图像文件路径
返回:np.ndarray: 预处理后的图像数组，带批次维度 (1, 128, 128, 3)
处理流程:
1. 打开图像并确保 RGB 三通道格式
2. 调整尺寸至模型要求的输入大小
3. 转换为 numpy 数组并归一化像素值
4. 添加批次维度

Real-ESRGAN 模型的输出后处理函数:
参数:
outputs(dict): 模型输出的字典对象
output_name(str): 输出张量在字典中的键名（默认'upscaled_image'）
save_path(str): 结果保存路径（默认'output.jpg'）
返回: Image.Image: 处理后的 PIL 图像对象
处理流程:
1. 从输出字典提取目标张量
2. 调整张量形状（根据实际模型输出调整）
3. 反归一化像素值到 0-255 范围
4. 转换为 8 位无符号整型
5. 保存图像

```python
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt 调用库

def preprocess_for_realesrgan(img_path: str) -> np.ndarray:
    # 打开图像文件，转换为RGB格式（避免单通道或RGBA问题）
    img = Image.open(img_path).convert("RGB")

    # 使用LANCZOS插值法将图像缩放到128x128
    # 这是Real-ESRGAN输入尺寸，可根据实际模型调整
    img = img.resize((128, 128), Image.LANCZOS)

    # 转换为float32类型的numpy数组
    img_array = np.array(img).astype(np.float32)

    # 将像素值从0-255范围归一化到0-1范围
    img_array = img_array / 255.0

    # 添加批次维度N，从HWC(高,宽,通道)变为NHWC(批次,高,宽,通道)
    return np.expand_dims(img_array, axis=0)
```

前处理函数

```python
def postprocess_realesrgan(outputs: dict,output_name: str = "upscaled_image",save_path: str = "output.jpg") -> Image.Image:
    # 从模型输出字典中提取目标张量
    output_tensor = np.array(outputs[output_name])

    # 重要：根据实际模型输出形状调整
    # 这里假设输出需要重塑为(1,512,512,3)格式
    # 如果模型输出尺寸不同，需要修改此处
    output_tensor = output_tensor.reshape(1, 512, 512, 3)

    # 去除单维度条目（批次维度等）
    img_array = output_tensor.squeeze()

    # 将模型输出的0-1范围值还原到0-255范围
    img_array = img_array * 255.0

    # 确保像素值在有效范围内并转换数据类型
    img_array = np.clip(img_array, 0, 255).astype(np.uint8)

    # 将numpy数组转换为PIL图像对象
    img = Image.fromarray(img_array)

    # 保存最终结果（JPEG格式）
    img.save(save_path)
    return img
```

后处理函数

深圳市广和通无线股份有限公司 | 总机: +86 755-26733555 传真: +86 755-26520841
邮箱: market@fibocom.com 官网: www.fibocom.com.cn

4.3 模型的运行推理

调用 python 主程序执行文件进行推理，在开发板终端输入以下命令：
python3 <Python 文件地址>
如：
(python3 /home/fibo/Fibo_AI_Stack/SR/test1.py)
输入内容，如图所示

输出结果，执行成功，结果如图所示：

5 Q&A

5.1 无法找到文件

可能原因：
1. 模型、标签等文件路径配置错误。
2. 模型文件未正确下载或解压。
3. 无权限访问文件。

解决办法：

深圳市广和通无线股份有限公司 | 总机: +86 755-26733555 邮箱: market@fibocom.com 传真: +86 755-26520841 官网: www.fibocom.com.cn

1. 修改模型的存放路径。
2. 确保模型文件已正确下载并解压到指定路径。
3. 确保运行的 Python 文件处于正确路径下。
4. 检查文件权限，确保当前用户有权限访问模型文件。

5.2 环境配置失败

可能原因：

1. 环境变量未正确设置。
2. 缺少必要的 Python 包。

解决办法：

1. 确保已正确执行环境配置脚本：

```
cd /home/fibo/Fibo_AI_Stack/fiboaisdk_ubuntu_aarch64
. ./scripts/env_qualcomm.sh 68
```

2.检查环境变量是否已正确设置，确保路径中没有错误。
3.安装必要的 Python 包。

5.3 推理结果不准确

可能原因：

1. 输入图像分不符合模型要求。
2. 模型未正确加载或配置。

解决办法：

1. 检查模型，输入输出 Tensor 名称等信息正确。
2. 重新加载模型并运行推理，确保模型配置无误。
3. 检查输入图像的预处理步骤，确保图像数据格式符合模型要求。

深圳市广和通无线股份有限公司 | 总机: +86 755-26733555 传真: +86 755-26520841
邮箱: market@fibocom.com 官网: www.fibocom.com.cn