Fibocom 广和通
完 美 无 线 体 验

基于 SC171V2 的 aidlite 加速推理指南

文档版本：V1.0
更新时间：2025 年 8 月 25 日

适用型号

序列	文档版本	适用型号	更新说明
1	V1.0	SC171 开发套件第二代	NA

目录

1 引言

本指南针对广和通 SC171V2 板卡的 aidlite 加速模块使用进行分步展示，可参考 https://docs.aidlux.com/guide/software/sdk/aidlite/aidlite-api-python，可在官网查看完成说明。

注：官网中 AI 应用开发/Aidlite/Python API 中 aidlite.Interpreter.create_instance()函数在实际中的用法与网页中所写用法存在出入，不同板卡 aidlite 版本有出入导致函数使用不同，需要留意，本指南适用于无法直接用 aidlux_gpu 模块加速的版本。

通过本指南，您不仅能够掌握 aidlite 加速模块的基础用法，还能根据实际需求灵活更换模型、调整参数，为后续复杂应用的开发奠定坚实基础。若在实践过程中遇到问题，可随时参考第 4 节的“常见问题解答”或联系技术支持团队。

2 所需软硬件环境

2.1 硬件环境

SC171 开发套件 V2、USB 数据线、电脑

2.2 软件环境

安装 Aidlite SDK（从广和通官方获取，板卡基本自带，无需主动安装）。

3 详细步骤

3.1 准备

1.相关模块信息：

（1）自定义模型属性：对于 AidliteSDK 而言，会处理不同框架的不同模型，每个模型自己也有不同的输入数据类型和不同的输出数据类型。在使用流程中，需要设置模型的输出输出数据类型，就需要用到此数据类型。

成员变量名	类型	值	描述
TYPE_DEFAULT	int	0	无效的 DataType 类型
TYPE_UINT8	int	1	无符号字节数据

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

成员变量名	类型	值	描述
TYPE_INT8	int	2	字节数据
TYPE_UINT32	int	3	无符号 int32 数据
TYPE_FLOAT32	int	4	float 数据
TYPE_INT32	int	5	int32 数据
TYPE_INT64	int	6	int64 数据
TYPE_UINT64	int	7	Uint64 数据
TYPE_INT16	int	8	Int16 数据
TYPE_UINT16	int	9	Uint16 数据
TYPE_FLOAT16	int	10	Float16 数据
TYPE_BOOL	int	11	Bool 数据

（2）自定义配置信息：除了需要设置 Model 具体信息之外，还需要设置一些推理时的配置信息。Config 类用于记录需要预先设置的配置选项，这些配置项在运行时会被用到。创建 Config 实例对象 create_instance()，此函数用于创建 Config 实例对象。

API	create_instance
描述	用于构造 Config 类的实例对象
参数	snpe_out_names：模型输出节点的名称列表（可选）
	number_of_threads：线程数，大于 0 有效（可选）
	is_quantify_model：是否为量化模型，1 表示量化模型 for FAST（可选）
	fast_timeout：接口超时时间（毫秒，大于 0 有效）for FAST（可选）
	accelerate_type：加速硬件的类型（可选）
	framework_type：底层深度学习框架的类型（可选）
返回值	正常：Config 实例对象
	异常：None

下面是对这 3 个主要参数的依次说明和赋值说明:

推理实现类型 class ImplementType

成员变量名	类型	值	描述
TYPE_DEFAULT	int	0	无效 ImplementType 类型
TYPE_REMOTE	int	2	通过 IPC 的后端实现
TYPE_LOCAL	int	3	通过本地调用的后端实现

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

模型框架类型 class FrameworkType

前面提到过，Aidlite SDK 整合了多种深度学习推理框架，所以在前述使用流程中，需要设置当前使用哪个框架的模型，就需要使用此框架类型。本指南选择 tflite 模型格式。

成员变量名	类型	值	描述
TYPE_DEFAULT	int	0	无效 FrameworkType 类型
TYPE_SNPE	int	1	SNPE 1.x (DLC) 模型
TYPE_TFLite	int	2	TFLite 模型
TYPE_RKNN	int	3	RKNN 模型
TYPE_QNN	int	4	QNN 模型
TYPE_SNPE2	int	5	SNPE 2.x (DLC) 模型
TYPE_NCNN	int	6	NCNN 模型
TYPE_MNN	int	7	MNN 模型
TYPE_TNN	int	8	TNN 模型
TYPE_PADDLE	int	9	Paddle 模型
TYPE_MS	int	10	MindSpore 模型
TYPE_ONNX	uint8_t	11	onnx 模型
TYPE_QNN216	uint8_t	101	QNN2.16 模型
TYPE_SNPE216	uint8_t	102	SNPE2.16 模型
TYPE_QNN223	uint8_t	103	QNN2.23 模型
TYPE_SNPE223	uint8_t	104	SNPE2.23 模型
TYPE_QNN229	uint8_t	105	QNN2.29 模型
TYPE_SNPE229	uint8_t	106	SNPE2.29 模型
TYPE_QNN231	uint8_t	107	QNN2.31 模型
TYPE_SNPE231	uint8_t	1068	SNPE2.31 模型

推理加速硬件类型 class AccelerateType

对于每个深度学习推理框架而言，可能会支持运行在不同的加速硬件单元上，所以在前述使用流程中，需要设置当前模型期望运行在哪种计算单元，就需要使用此加速硬件单元类型。

成员变量名	类型	值	描述
TYPE_CPU	int	1	CPU 通用加速单元
TYPE_GPU	int	2	GPU 通用加速加速
TYPE_DSP	int	3	高通 DSP 加速单元

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

2. 导入所需的库，在终端输入以下命令:

```
pip3 install <所需安装包名>
```

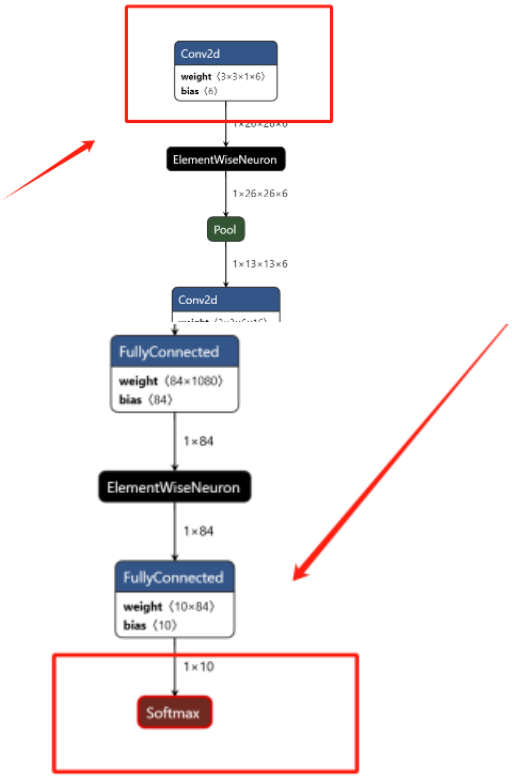
3. 需要准备的参数：

--dlc_path：模型文件路径（.dlc 文件）

--input_path：输入文件路径

--模型相关数据:输入张量信息、输出张量信息

对于模型相关数据不了解的用户，可以通过 Netron 网站，导入模型文件查看模型相关信息，如图：



3.2 Python sample 代码

1. 调用库，定义文件路径
2. 加载模型，并对输入输出数据进行定义。
3. 创建 Config 实例对象。
4. 创建解释器对象，初始化解释器，解释器加载模型。
5. 根据使用的模型来更改对输入内容的前处理。
6. 执行模型推理操作。
7. 获取推理结果数据，根据使用的模型来更改模型输出后处理的代码部分。

自定义模型属性与**模型信息区域**，需要参考上文中提到的模块配置信息进行修改，**前后处理函数**与**模型输入输出张量信息**需要自行修改。

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

下面代码以手写数字识别案例为例作为展示

导入必要的库

```python
import aidlite  # 导入AIDLite库，用于模型推理
import numpy as np  # 导入NumPy库，用于数值计算
import cv2  # 导入OpenCV库，用于图像处理
import os  # 导入os库，用于文件路径操作

#读取标签
def read_label_list():
    with open('label.txt', 'r', encoding="utf8") as f:
        data = f.read().splitlines()  # 读取所有行并去除换行符
    return data
```

前处理函数

```python
#前处理函数
def preprocess_image(image_path):
    img = cv2.imread(image_path)  # 读取图像文件
    if img is None:
        raise ValueError(f"Failed to load image from {image_path}!")  # 图像加载失败抛出异常
    # 转换为灰度图
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  # 将BGR图像转换为灰度图像
    # 调整图像大小
    img = cv2.resize(img, (28, 28))  # 调整图像大小为28x28像素
    # 添加批次和通道维度
    img = np.expand_dims(img, axis=0)  # 添加批次维度，形状变为(1, 28, 28)
    img = np.expand_dims(img, axis=-1)  # 添加通道维度，形状变为(1, 28, 28, 1)
    # 归一化到[0,1]范围
    img = img.astype(np.float32) / 255.0  # 将像素值从0-255缩放到0-1之间
    print("Preprocessed image shape:", img.shape)  # 打印预处理后的图像形状
    return img
```

后处理函数

```python
#后处理函数
def postprocess_output(output_data, label_list):
    print("Raw output data:", output_data)  # 打印原始输出数据
    # 确保output_data是二维数组
    if output_data.ndim == 1:
        # 如果是一维数组，将其转换为二维数组
        output_data = output_data.reshape(1, -1)  # 重塑为(1, 10)形状
        print("Reshaped output data:", output_data)  # 打印重塑后的输出数据
    # 获取预测结果
    w = np.argmax(output_data)  # 找到输出数组中最大值的索引
    # 根据output_data的形状选择正确的索引方式获取置信度
    if output_data.ndim == 2:
        confidence = output_data[0, w]  # 对于二维数组，使用[0, w]索引
    else:
        confidence = output_data[w]  # 对于一维数组，使用[w]索引
    return label_list[w], confidence
```


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

```python
def main():
    # 1. 定义模型和文件路径
    model_path = os.path.abspath('MyModel.tflite')  # 获取模型的绝对路径
    image_path = '<输入文件路径>'  # 待识别的图像文件路径

    # 2. 创建AIDLite模型对象
    model = aidlite.Model.create_instance(model_path)  # 创建模型实例
    if model is None:
        print("Create model failed!")  # 模型创建失败处理
        exit(1)  # 退出程序

    # 3. 创建配置实例对象
    config = aidlite.Config.create_instance()  # 创建配置实例
    if config is None:
        print("Create config failed!")  # 配置创建失败处理
        exit(1)  # 退出程序

    # 设置配置参数
    config.framework_type = aidlite.FrameworkType.TYPE_TFLITE  # 设置框架类型为TFLite
    config.accelerate_type = aidlite.AccelerateType.TYPE_GPU  # 设置加速类型为GPU加速
    config.implement_type = aidlite.ImplementType.TYPE_LOCAL  # 设置实现类型为本地执行
    config.number_of_threads = 4  # 设置线程数为4
    config.is_quantify_model = 0  # 设置是否为量化模型（0表示非量化模型）
    config.fast_timeout = -1  # 设置超时时间（-1表示无超时限制）
    print("Create Config success!")  # 配置创建成功提示
```

自定义配置信息

```python
# 4. 创建解释器对象
interpreter = aidlite.InterpreterBuilder.build_interpretper_from_model_and_config(model, config)
if interpreter is None:
    print("build_interpretper_from_model_and_config failed!")  # 解释器创建失败处理
    exit(1)  # 退出程序
print("Create Interpreter success!")  # 解释器创建成功提示

# 初始化解释器
result = interpreter.init()  # 初始化解释器
if result != 0:
    print("interpreter->init() failed!")  # 解释器初始化失败处理
    exit(1)  # 退出程序
print("Interpreter init success!")  # 解释器初始化成功提示

# 加载模型到解释器
result = interpreter.load_model()  # 加载模型
if result != 0:
    print("interpreter->load_model() failed!")  # 模型加载失败处理
    exit(1)  # 退出程序
print("Interpreter load model success!")  # 模型加载成功提示
```

自定义输入输出张量形状

```python
# 定义输入输出张量的形状
input_details = [[1, 28, 28, 1]]  # 输入张量形状：批次大小=1, 高度=28, 宽度=28, 通道数=1
output_details = [[1, 10]]  # 输出张量形状：批次大小=1, 输出类别数=10
```

```python
# 设置模型属性
model.set_model_properties(
    input_shapes=input_details,
    input_data_type=aidlite.DataType.TYPE_FLOAT32,
    output_shapes=output_details,
    output_data_type=aidlite.DataType.TYPE_FLOAT32
)
```

自定义模型属性

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

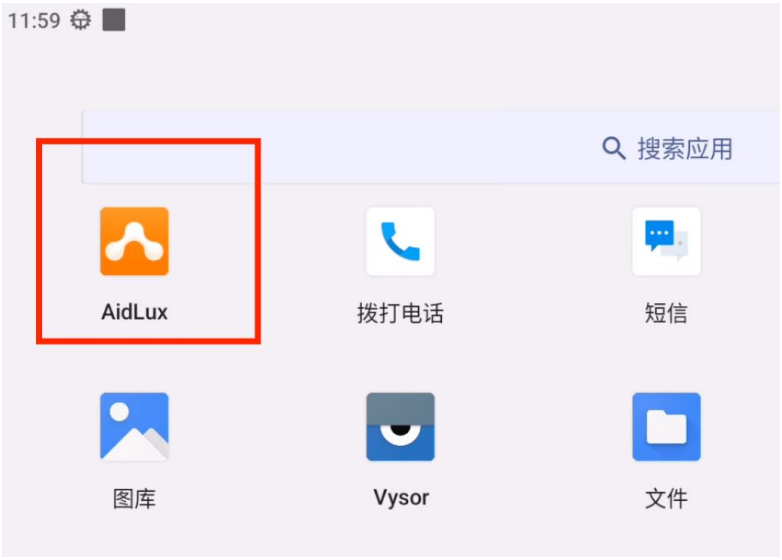
```python
# 5. 图像预处理
try:
    processed_img = preprocess_image(image_path)  # 调用预处理函数处理图像
except ValueError as e:
    print(e)  # 打印错误信息
    exit(1)  # 退出程序

# 设置输入张量
result = interpreter.set_input_tensor(0, processed_img)  # 将预处理后的图像数据设置为输入张量
if result != 0:
    print("interpreter->set_input_tensor() failed!")  # 设置输入张量失败处理
    exit(1)  # 退出程序

# 6. 执行模型推理
result = interpreter.invoke()  # 执行模型推理
if result != 0:
    print("interpreter->invoke() failed!")  # 模型推理失败处理
    exit(1)  # 退出程序

# 获取输出张量
output_data = interpreter.get_output_tensor(0)  # 获取输出张量数据
if output_data is None:
    print("interpreter->get_output_tensor() failed!")  # 获取输出张量失败处理
    exit(1)  # 退出程序

# 7. 结果后处理
label_list = read_label_list()  # 读取标签列表
predicted_label, confidence = postprocess_output(output_data, label_list)  # 调用后处理函数处理输出数据
# 打印预测结果
print("Predicted label:", predicted_label)  # 打印预测的标签
print("Confidence:", confidence)  # 打印置信度

if __name__ == "__main__":
    main()  # 执行主函数
```

自定义前处理函数

自定义后处理函数

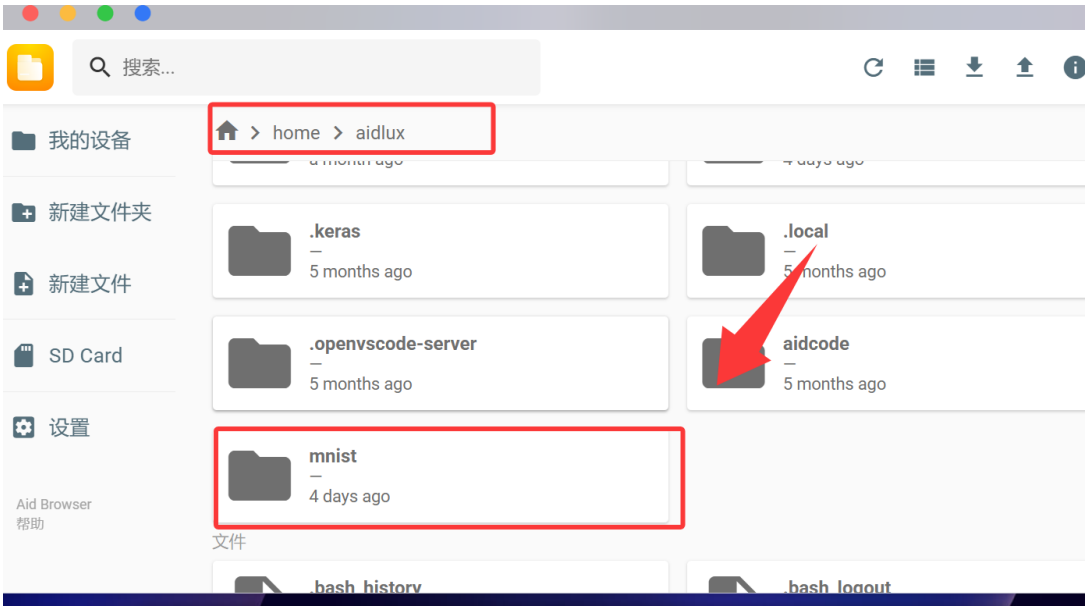
3.3 模型的运行推理

运行板卡，上滑到应用界面，打开 AidLux

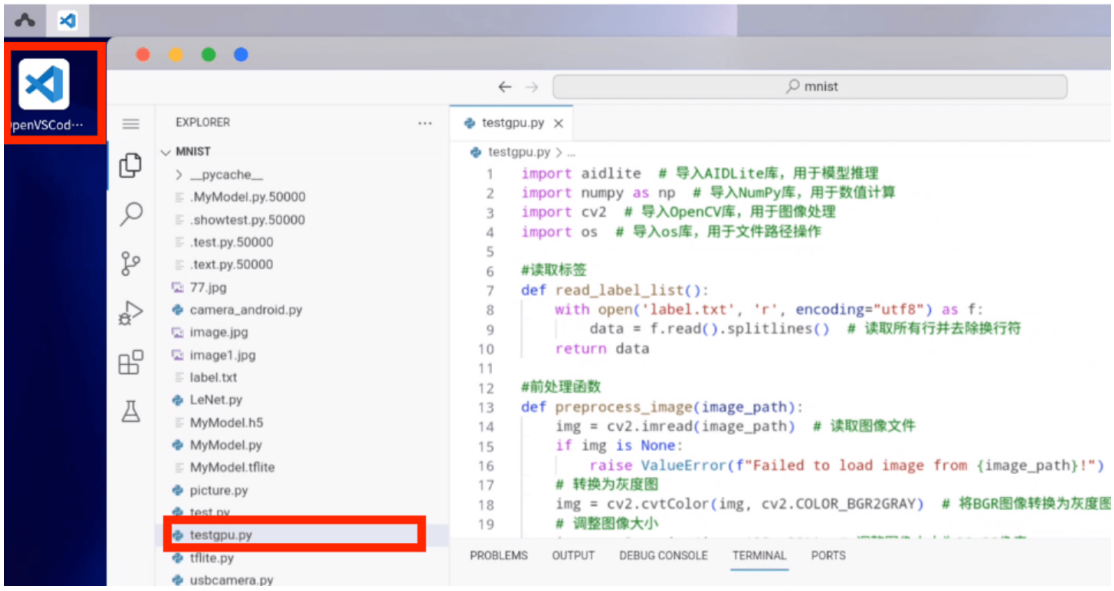


上传工程源码到板卡中，如图所示

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

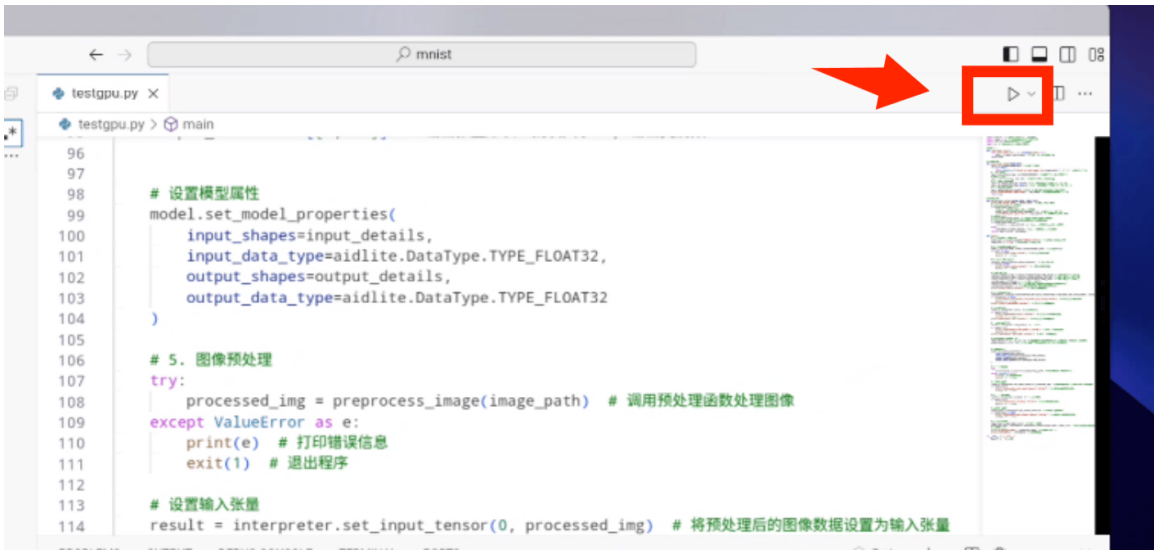


打开 VS code，通过 VS code 打开 minst 文件夹，修改 testgpu.py 文件，更改路径为个人板卡中的地址

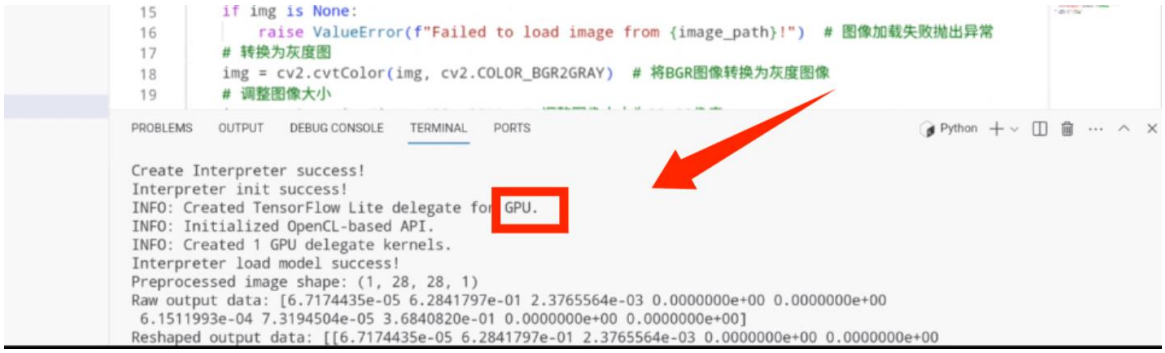


调用 testgpu 执行文件进行推理，右上角选择运行文件

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



如图所示，成功调用 GPU 进行模型推理



4 Q&A

4.1 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：

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 文件路径正确。
4. 检查文件权限，确保当前用户有权限访问模型文件。

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

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

4.3 推理结果不准确

可能原因：

1. 输入图像分辨率不符合模型要求。
2. 模型未正确加载或配置。

解决办法：

1. 确保输入图像格式正确使用，可以输出输入数据来检查格式是否符合模型要求。
2. 检查模型，输入输出 Tensor 名称等信息正确。
3. 重新加载模型并运行推理，确保模型配置无误。
4. 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 邮箱: market@fibocom.com 传真: +86 755-26520841 官网: www.fibocom.com.cn