Fibocom 广和通
完 美 无 线 体 验

人体目标检测（Yolov5）案例

----基于 SC171 开发套件 V3

文档版本：V1.0
更新时间：2025 年 3 月 19 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 YOLOv5 人体目标检测模型为例，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，完成一个人体目标检测案例。

若实践中遇到问题，请参考第 4 节的 Q&A 或联系 Fibocom 技术支持团队。

2 详细步骤

2.1 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 opencv-python Pillow scikit-image
```

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
- 模型相关数据:输入张量名称、输出张量名称

4. 将本案例中的 sample_infer_yolov5.py、utils.py 与 api_infer.py 放置在同一路径下

2.2 Python sample 代码

1. 宏定义：调用库，定义文件路径，修改为自己的模型与图片路径

```python
# 导入必要的库
from api_infer import *  # 导入SNPE推理相关库
import numpy as np
import cv2
from utils import detect_postprocess, draw_detect_res, preprocess_img  # 导入自定义工具函数
from PIL import Image

#配置区
dlc_path = "/home/fibo/Fibo_AI_Stack/yolov5/yolov5.dlc"  # 模型文件路径（.dlc格式）
img_path = "/home/fibo/Fibo_AI_Stack/yolov5/bus.jpg"      # 测试图片输入路径
save_path = "/home/fibo/Fibo_AI_Stack/yolov5/result_bus.jpg"  # 结果保存路径
```

2. 前后处理函数内容见 utils.py

3. 程序主内容

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
# 初始化SNPE推理上下文
snpe_ort = SnpeContext(dlc_path, [], Runtime.GPU, PerfProfile.BALANCED, LogLevel.INFO)
assert snpe_ort.Initialize() == 0  # 初始化 SNPE 环境

# 图像预处理
image = cv2.imread(img_path)
# target_shape=(640,640): 模型输入尺寸(YOLOv5标准输入)
# div_num=255: 像素归一化(0-255 -> 0-1)
pic = preprocess_img(image,target_shape=(640,640),div_num=255)
```

前处理

```python
# 执行模型推理
input_feed = {"serving_default_input_1:0": pic}
output_names = []
outputs = snpe_ort.Execute(output_names, input_feed)
```

模型推理

```python
# 输出后处理,提取检测结果
date = outputs['StatefulPartitionedCall:0']
date = np.array(date)  # 转换为numpy数组
# 重塑输出形状（YOLOv5输出格式：1x25200x6）
# 维度说明：1(batch) x 25200(锚框数) x 6(x,y,w,h,conf,class)
pred = date.reshape(1, 25200, 6)
# 可视化处理,重新加载原始图像（RGB格式）
img = cv2.imread(img_path)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
# 后处理参数说明：
# [1080,810,3]：原始图像尺寸（需根据实际修改）
# [640,640,3]：模型输入尺寸
# conf_thres=0.5：置信度阈值（低于此值过滤）
# iou_thres=0.45：NMS的IoU阈值
pred = detect_postprocess(pred,[1080,810,3],[640,640,3],conf_thres=0.5,iou_thres=0.45)
# 绘制检测框并保存结果
res_img = draw_detect_res(img, pred)
res_img = Image.fromarray(res_img)  # 转换为PIL图像对象
res_img.save("output.jpg")          # 保存结果
blended2 = Image.open("output.jpg")
blended2.show()  # 弹出窗口显示结果
```

后处理

2.3 模型的运行推理

调用 python 执行文件进行推理，在开发板终端输入以下命令：

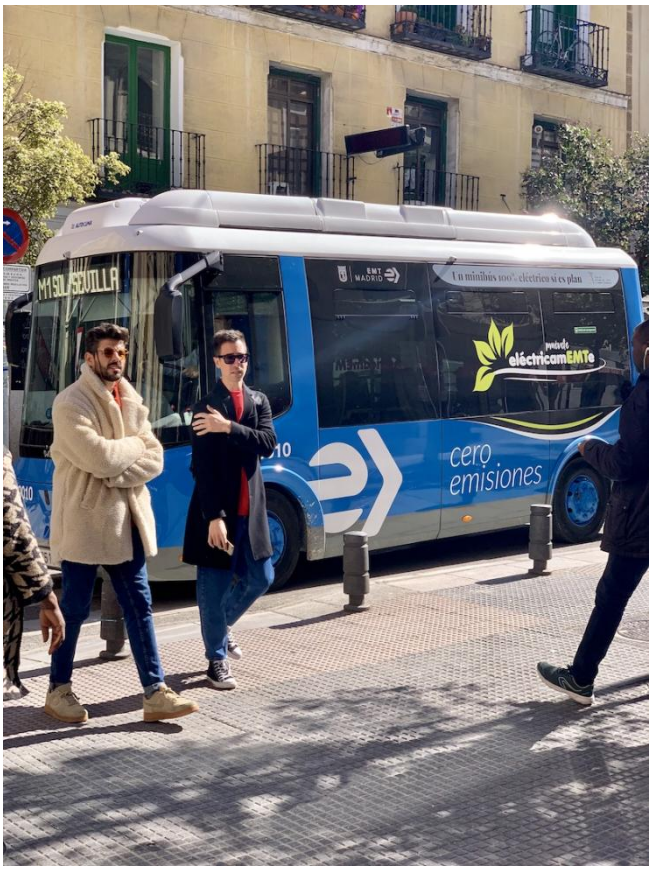
python3 <Python 文件地址>

如：

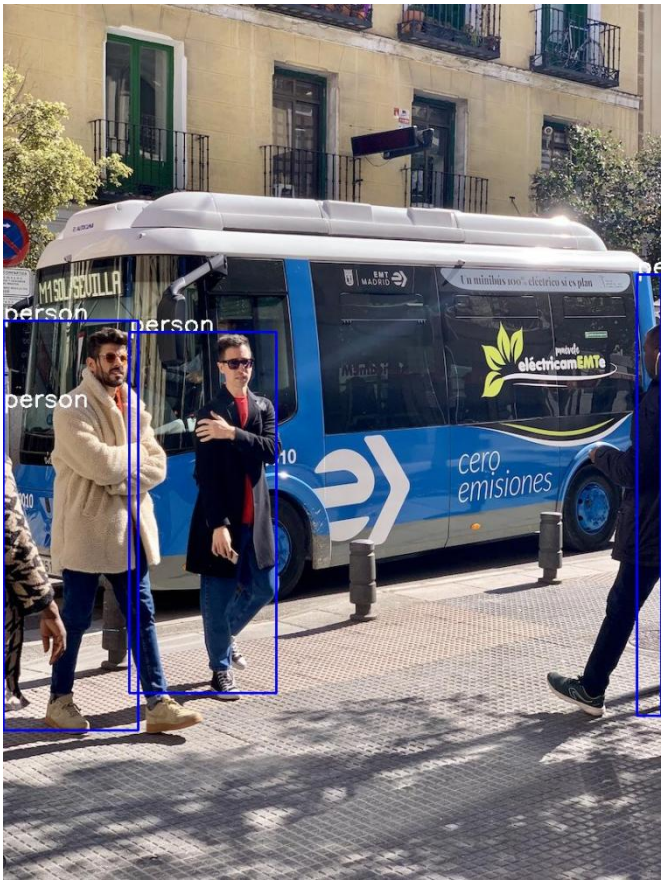
python3 /home/fibo/Fibo_AI_Stack/yolov5/sample_infer_yolov5.py

输入内容，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



输出结果，执行成功，结果如图显示：



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

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。

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. 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。

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 传真: +86 755-26520841
邮箱: market@fibocom.com 官网: www.fibocom.com.cn