**This is extra research! This information is not finalized please refer to other documentation first.**

**Running a YOLOv8 model on a Neural Compute Stick 2 (NCS2) with a Raspberry Pi**:i involves several steps, including setting up the environment, installing necessary libraries, converting the YOLOv8 model to a format compatible with the NCS2, and running inference. Here's a detailed guide:

Prerequisites

1. **Hardware**:
    - Raspberry Pi (preferably Raspberry Pi 4 for better performance).
    - Intel Neural Compute Stick 2.
    - USB power supply for the Raspberry Pi.
2. **Software**:
    - Raspbian OS installed on the Raspberry Pi.
    - Python 3 and pip installed.

Step 1: Install Required Libraries

Start by updating your Raspberry Pi and installing the necessary libraries.

bash
Copy code
sudo apt update
sudo apt upgrade
sudo apt install python3-pip
sudo apt install git

Step 2: Install OpenVINO Toolkit

1. **Download the OpenVINO Toolkit**: You can download the latest OpenVINO toolkit from the Intel website.
2. **Install OpenVINO**: Follow the instructions in the OpenVINO installation guide to install the toolkit on your Raspberry Pi. You may need to modify environment variables as suggested in the installation instructions.

bash
Copy code
source /opt/intel/openvino/bin/setupvars.sh

Step 3: Convert YOLOv8 Model to OpenVINO Format

**Export YOLOv8 Model**: You can use the Ultralytics repository to export the model to OpenVINO format. First, clone the YOLOv8 repository:
bash
Copy code
git clone https://github.com/ultralytics/yolov8.git
cd yolov8

1.

**Install Ultralytics YOLO**:
bash
Copy code
pip install -r requirements.txt

2.

**Export the model**: Use the following command to export your YOLOv8 model to the OpenVINO format:
bash
Copy code
python export.py --weights yolov8.pt --img-size 640 --batch-size 1 --include openvino

3. This will generate an OpenVINO model in the yolov8/ directory.

Step 4: Run Inference
**Install the OpenVINO Inference Engine**:
After setting up OpenVINO, you need to install the inference engine:
bash
Copy code
sudo apt install openvino-inference-engine

1.

**Run the inference**: Create a Python script to run the inference using the NCS2. Here is a sample script:
python
Copy code
import cv2
import numpy as np
from openvino.inference_engine import IECore

# Load the model
model_xml = "path/to/your/model.xml"
model_bin = "path/to/your/model.bin"

```
ie = IECore()
net = ie.read_network(model=model_xml, weights=model_bin)
exec_net = ie.load_network(network=net, device_name="MYRIAD")

# Load an image
image = cv2.imread('image.jpg')
input_blob = next(iter(net.input_info))
out_blob = next(iter(net.outputs))

# Pre-process the image
n, c, h, w = net.input_info[input_blob].input_data.shape
image = cv2.resize(image, (w, h))
image = image.transpose((2, 0, 1))  # HWC to CHW
image = image.reshape((n, c, h, w))

# Run inference
result = exec_net.infer(inputs={input_blob: image})

# Process the results
detections = result[out_blob]
```

2.

**Run your script**: Make sure to change the paths to your model files and the image file, then run the script:
bash
Copy code
python3 your_script.py

3.

Step 5: Additional Considerations

- **Power Supply**: Ensure that your Raspberry Pi has a sufficient power supply, especially when using the NCS2.
- **Performance**: Running models on the Raspberry Pi might be limited by its computational power; performance may vary based on the model complexity and input size.
- **Camera Input**: If you're using a camera, you can replace the image loading part with code that captures frames from the camera.

Resources

- OpenVINO Documentation
- Ultralytics YOLOv8 Repository