Quick Start Demo of SNPE Yolov5 in 6490

Requirement:

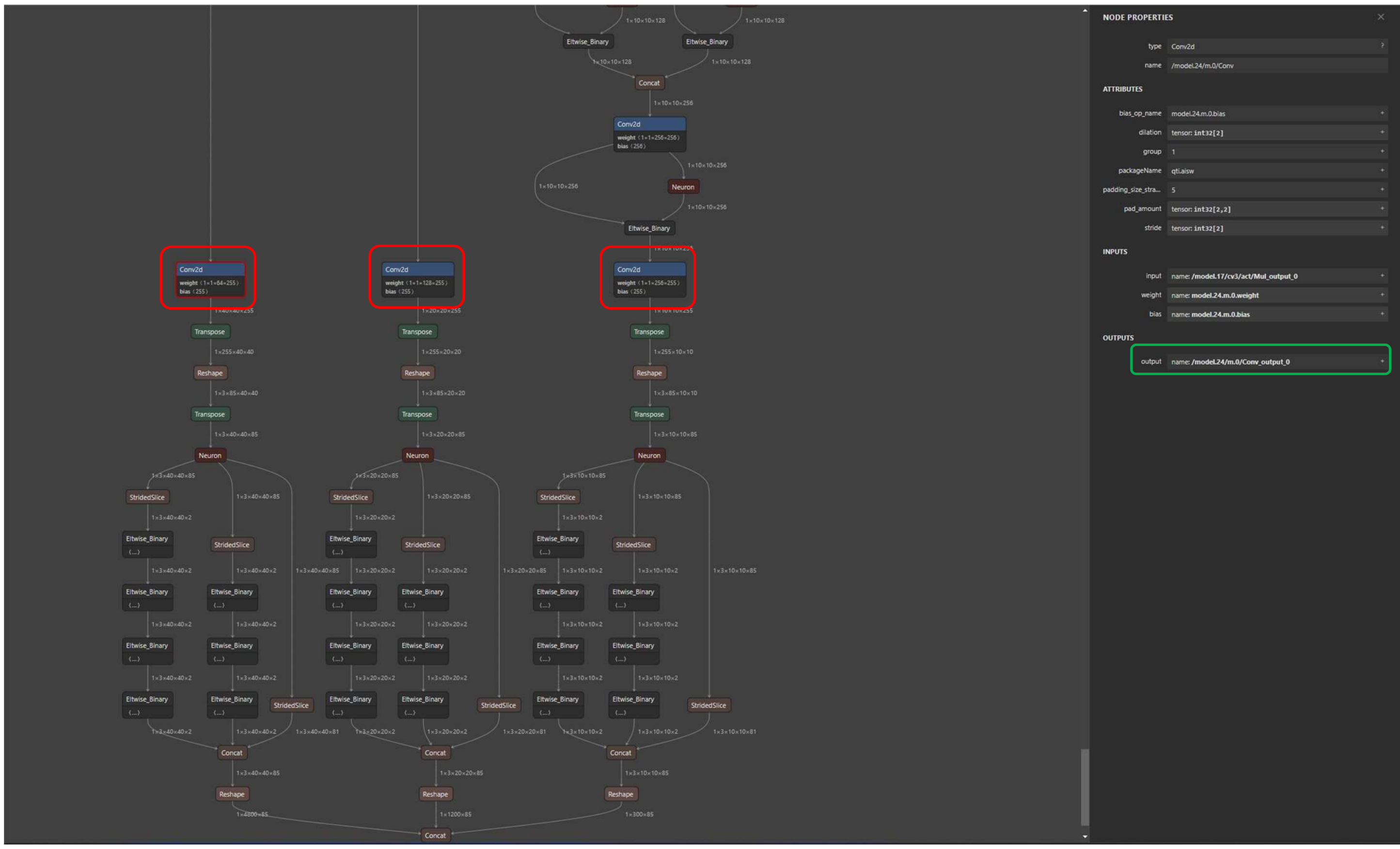
1. Ubuntu 20.04
2. Python 3.8 and Virtual Environment
3. Finished installation of SNPE

Steps:

1. Setup Yolov5 environment and export yolov5n with image size 320 without NMS ONNX model.
2. Activate SNPE virtual environment.

```
source snpe/bin/activate
source /opt/qcom/aistack/snpe/2.19.0.240124/bin/envsetup.sh
```

3. Convert Model to DLC
   To make Yolov5 run more efficiently in edge, we will remove the decode part from the ONNX model which are the nodes before 5D reshape. The below diagram shows how to find the out-node name by Netron (https://netron.app/). You can click the desire operation block in the Netron, the input and output node name of the operation graph will show in the right-hand side. The red blocks show the operation block which you need to click to find the output node name. The green block is the name you need to insert while DLC conversion. The name of the output node should be fill in the flag "--out_node" in the SNPE convert command "snpe-onnx-to-dlc". In the example case, the desired output node names are "/model.24/m.0/Conv_output_0", "/model.24/m.1/Conv_output_0" and "/model.24/m.2/Conv_output_0". User may find the different name due to PyTorch version. Please check the node name before DLC conversion.



NODE PROPERTIES

type: Conv2d
name: /model.24/m.0/Conv

ATTRIBUTES

bias_op_name: model.24.m.0.bias
dilation: tensor: int32[2]
group: 1
packageName: qti.aisw
padding_size_stra...: 5
pad_amount: tensor: int32[2,2]
stride: tensor: int32[2]

INPUTS

input: name: /model.17/cv3/act/Mul_output_0
weight: name: model.24.m.0.weight
bias: name: model.24.m.0.bias

OUTPUTS

output: name: /model.24/m.0/Conv_output_0

The command of ONNX model to DLC is:

```
snpe-onnx-to-dlc -i yolov5n320.onnx \
--out_node /model.24/m.0/Conv_output_0 \
--out_node /model.24/m.1/Conv_output_0 \
--out_node /model.24/m.2/Conv_output_0 \
-o yolov5n320_v213_cut.dlc
```

4. Write a Python script for generating model raw data for quantization (remember dlc input swap the channel order from NCHW to NHWC). The should also create image list text fill called "img_list.txt" for storing the location of input raw file.

   *Note : Image list text file template be found in /opt/qcom/aistack/snpe/2.19.0.240124/docs/SNPE/html/general/tools.html#snpe-dlc-quantize

   gen_raw.py

```
import cv2
import numpy as np
import glob

img_list = glob.glob("./data/images/*.jpg")

f = open("img_list.txt","w")
f.write("%/model.24/m.0/Conv_output_0 /model.24/m.1/Conv_output_0 /model.24/m.2/Conv_output_0\n")
for img_path in img_list:
    img_name = img_path.split("/")[-1].replace(".jpg","")
    img = cv2.imread(img_path)
    img = cv2.dnn.blobFromImage(img, 1/255, (320, 320), [0,0,0], 1, crop=False)
    img = np.transpose(img, (0,2,3,1))

    output_raw_path = img_name + ".raw"
    img.astype(np.float32).tofile(output_raw_path)
    f.write("images:="+output_raw_path+"\n")
f.close()
```

   The img_list.txt should look like this

```
%/model.24/m.0/Conv_output_0 /model.24/m.1/Conv_output_0 /model.24/m.2/Conv_output_0
images:=bus.raw
images:=zidane.raw
```

   Note: change line

```
f.write("%/model.24/m.0/Conv_output_0 /model.24/m.1/Conv_output_0 /model.24/m.2/Conv_output_0\n")
```

   If you have different output node name.

5. Quantize the DLC model

```
snpe-dlc-quantize --input_dlc yolov5n320_v213_cut.dlc \
  --input_list img_list.txt \
  --use_enhanced_quantizer \
  --use_adjusted_weights_quantizer \
  --optimizations cle --axis_quant \
  --enable_htp --htp_socs sm7325 --override_params \
  --act_bitwidth 8 --weights_bitwidth 8 --output_dlc htp_socs_cut.dlc
```

6. Create Graph Prepared DLC

```
snpe-dlc-graph-prepare --input_dlc htp_socs_cut.dlc \
  --output_dlc htp_socs_cut_cached.dlc \
  --htp_socs sm7325 --overwrite_cache_records
```