Following the previous error of cuda version mismatch. I extracted engine file inside deepstream container 6.4 with cuda 12.2. I extracted for both yolo and alexnet at 16FPS and 32FPS. Both precision of Alexnet classifier model is working well but it fails for yolo due to multiple errors.

During the process of extracting .pth (PyTorch model) to .onnx, we encountered several dependency mismatches within the DeepStream container. To avoid conflicts and maintain **a clean and isolated environment,** I created a **separate Conda environment with pytorch** 1.13.0+cu117 (downgraded from PyTorch 2.6 to 1.9 then to 1.13) **.**

**Summary of What I Tried & Errors Encountered**

**1️ initial Conversion (PTH → ONNX → TensorRT Engine)**

* **Process:**
  + Used mmdeploy/tools/deploy.py to convert .pth to .onnx and then to .engine.
  + Command:
  + DEPLOY\_CONFIG=/working/michael-mmyolo-playground/deploy\_trt.py
  + MODEL\_CONFIG=/working/michael-mmyolo-playground/custom\_train.py
  + MODEL=/working/training-output/best\_coco\_bbox\_mAP\_epoch\_788.pth
  + DEPLOY\_SCRIPT=/working/mmdeploy/tools/deploy.py
  + TEST\_IMAGE=/working/test.jpg

python3 $DEPLOY\_SCRIPT \

$DEPLOY\_CONFIG \

$MODEL\_CONFIG \

$MODEL \

$TEST\_IMAGE \

--work-dir /working/training-output/exports \

--device cuda:0 \

--log-level INFO \

--show \

--dump-info

**command used to verify engine file.(** trtexec --loadEngine=/working/ training-output/exports/end2end.engine --verbose)

**Error:**

* + The generated engine file contained **TRTBatchedNMS**, which **failed to load** with:

[E] getPluginCreator could not find plugin: TRTBatchedNMS version: 1

[E] Engine deserialization failed

**2️ Checking the ONNX File**

* **Process:**
  + Verified if TRTBatchedNMS exists in ONNX before conversion:

strings /working/training-output/exports/end2end.onnx | grep NMS

* + **Output:** TRTBatchedNMS was **still present** in the ONNX model.

**3️ Attempted to Enable EfficientNMS Instead**

* **Process:**
  + Modified deploy\_trt.py by setting:

use\_efficientnms = True

* **Error:**
  + Even after enabling EfficientNMS, TRTBatchedNMS **was still present** in the ONNX model.

**4️ Removing NMS During PTH → ONNX Conversion**

* **Process:**
  + Checked if remove\_nms=True could be passed in deploy\_trt.py.
* **Error:**
  + The ONNX model **still contained TRTBatchedNMS**, meaning the removal flag was ignored or ineffective.

**5️ Manually Stripping NMS from ONNX Using ONNX Surgeon**

* **Process:**
  + Identified dependent layers using:

onnx.utils.polish\_model(onnx\_model)

* + Removed TRTBatchedNMS and redirected dependent layers.
* **Error:**
  + When testing the modified ONNX, it **failed during conversion to TensorRT** with:

[E] ModelImporter.cpp: Assertion failed: (ctx->tensors().count(inputName)) && "Node input was not registered."

**6️ Checking TensorRT Plugin Availability**

* **Process:**
  + Verified if TensorRT had the required plugins:

trtexec --dumpPlugins

* **Error:**
  + TRTBatchedNMS **was missing** in TensorRT plugins, meaning it wasn't being registered.

Would you like to continue debugging, or move to an alternative approach?