

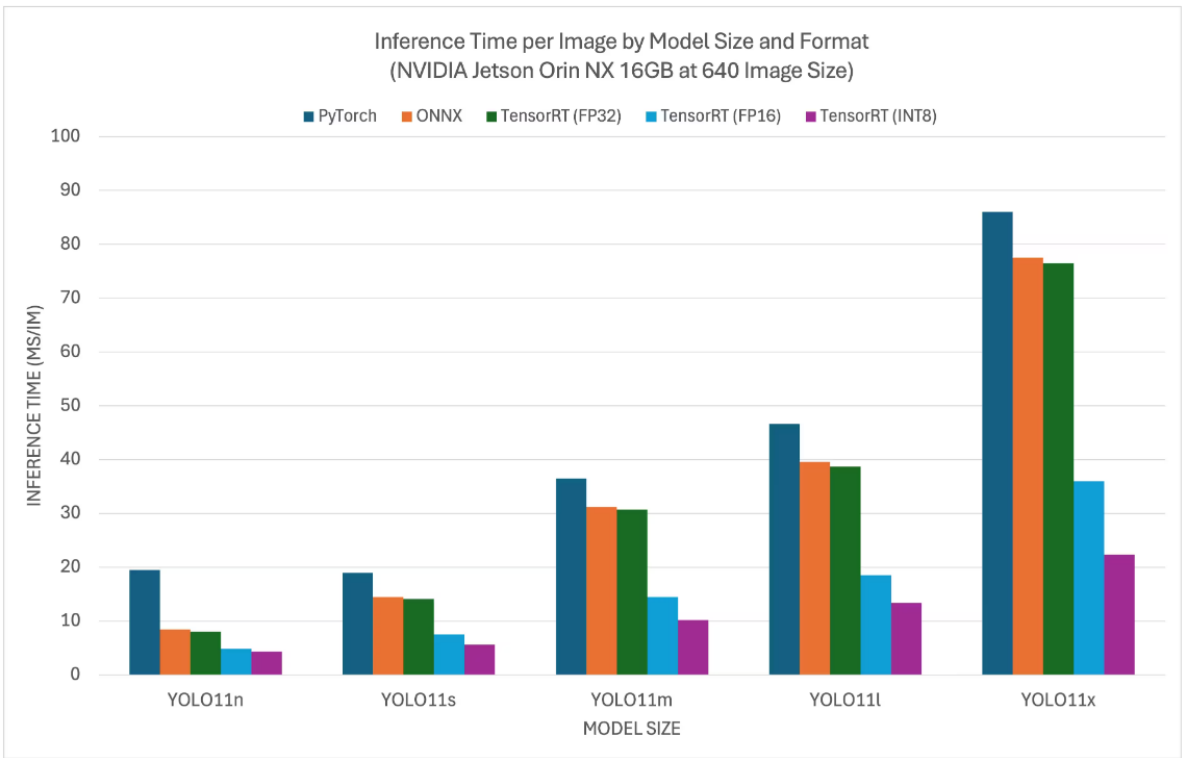
Model Conversion

Model Conversion

- 1. Jetson Orin YOLO11 (Benchmark)
- 2. Enable Maximum Board Performance
 - 2.1. Enable MAX Power Mode
 - 2.2. Enable Jetson Clocks
- 3. Model Conversion
 - 3.1. CLI: pt → onnx → engine
 - 3.2. Python: pt → onnx → engine
- 4. Model Prediction
 - CLI Usage
- Common Issues
 - ERROR: onnxslim
- References

1. Jetson Orin YOLO11 (Benchmark)

YOLO11 benchmark data is sourced from the Ultralytics team, testing various model formats (data for reference only)



2. Enable Maximum Board Performance

2.1. Enable MAX Power Mode

Enabling MAX Power Mode on Jetson will ensure that all CPU and GPU cores are turned on:

```
#Orin Nano
sudo nvpmode1 -m 2
#Orin Nx
sudo nvpmode1 -m 0
```

2.2. Enable Jetson Clocks

Enabling Jetson Clocks will ensure that all CPU and GPU cores run at maximum frequency:

```
sudo jetson_clocks
```

3. Model Conversion

According to the test parameters provided by the Ultralytics team for different format models, we can find that using TensorRT provides the best inference performance!

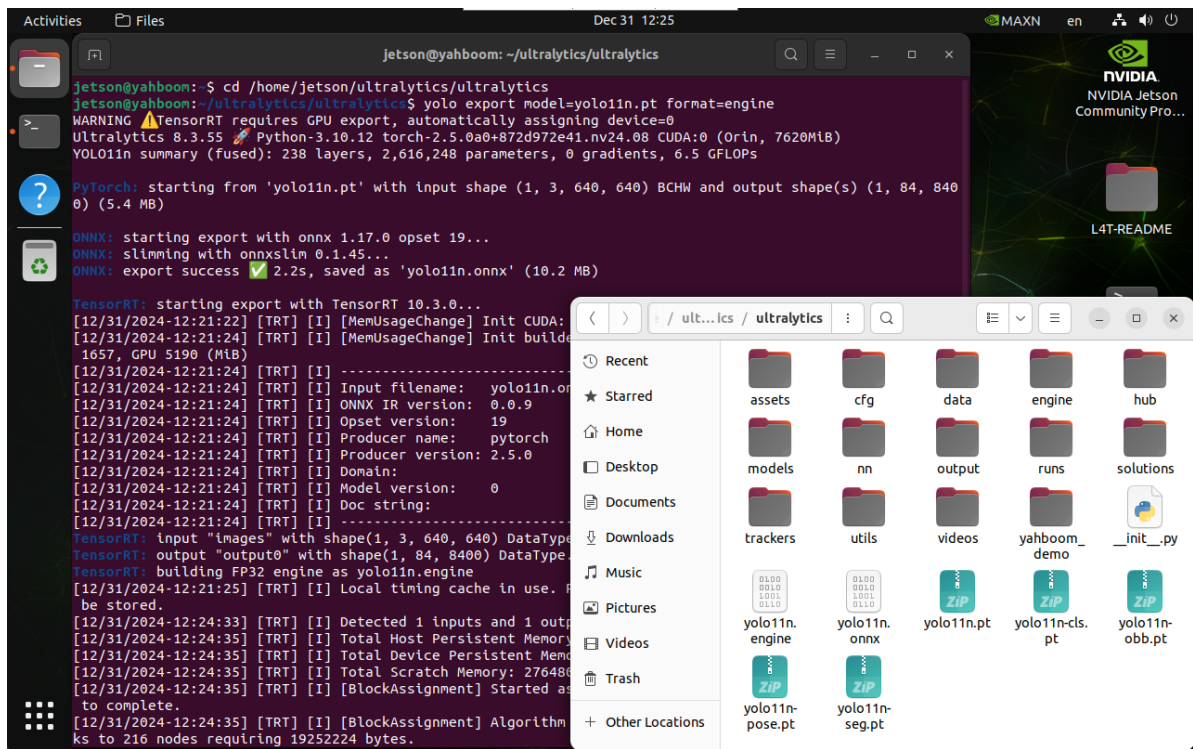
First time using YOLO11 export mode will automatically install some dependencies, please wait for the automatic completion!

3.1. CLI: pt → onnx → engine

Convert PyTorch format model to TensorRT: The conversion process will automatically generate ONNX model

```
cd /home/jetson/ultralytics/ultralytics
```

```
yolo export model=yolo11n.pt format=engine
# yolo export model=yolo11n-seg.pt format=engine
# yolo export model=yolo11n-pose.pt format=engine
# yolo export model=yolo11n-cls.pt format=engine
# yolo export model=yolo11n-obb.pt format=engine
```



3.2. Python: pt → onnx → engine

Convert PyTorch format model to TensorRT: The conversion process will automatically generate ONNX model

```
cd /home/jetson/ultralitics/ultralitics/yahboom_demo
```

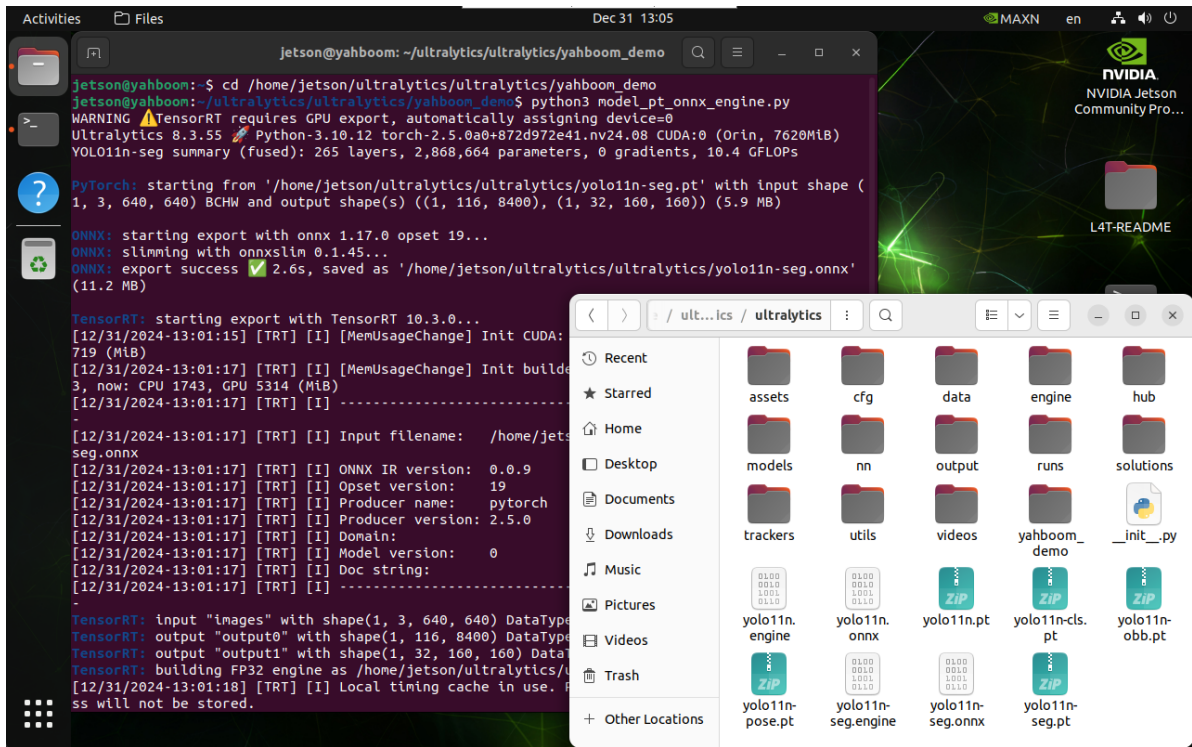
```
python3 model_pt_onnx_engine.py
```

```
from ultralytics import YOLO

# Load a YOLO11n PyTorch model
# model = YOLO("/home/jetson/ultralitics/ultralitics/yolo11n.pt")
model = YOLO("/home/jetson/ultralitics/ultralitics/yolo11n-seg.pt")
# model = YOLO("/home/jetson/ultralitics/ultralitics/yolo11n-pose.pt")
# model = YOLO("/home/jetson/ultralitics/ultralitics/yolo11n-cls.pt")
# model = YOLO("/home/jetson/ultralitics/ultralitics/yolo11n-obb.pt")

# Export the model to TensorRT
model.export(format="engine")
```

Note: The converted model files are located at the same location as the original model files



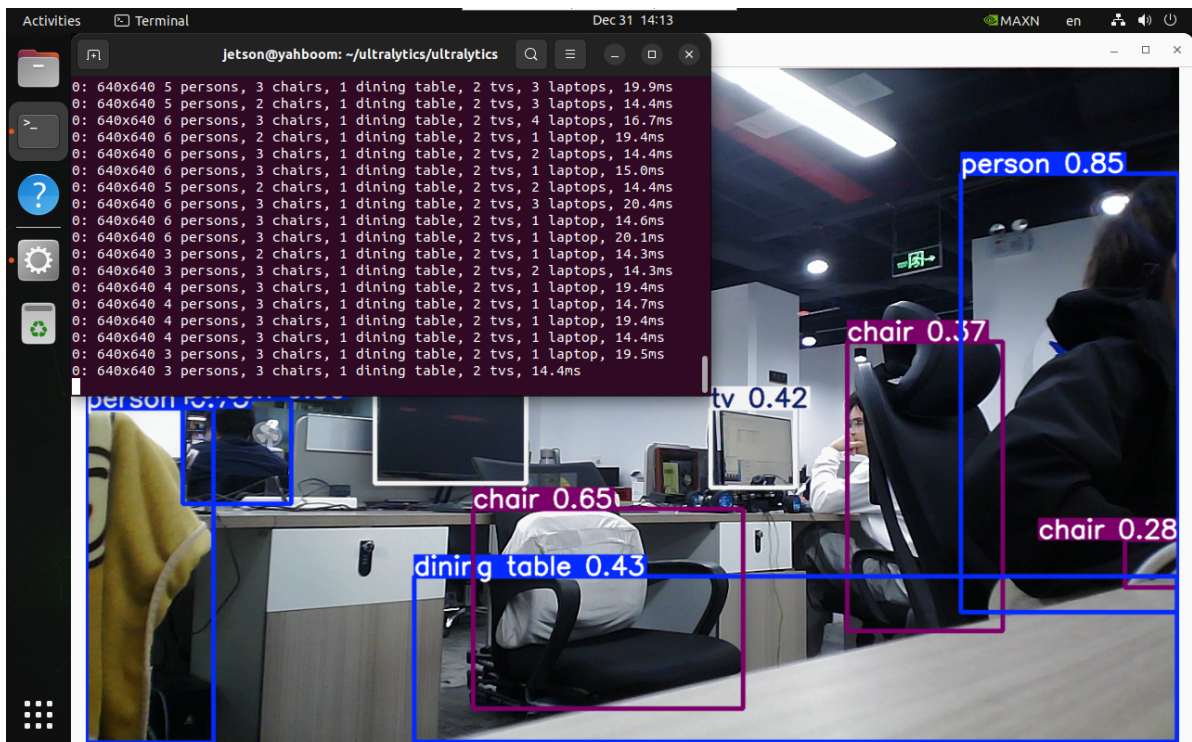
4. Model Prediction

CLI Usage

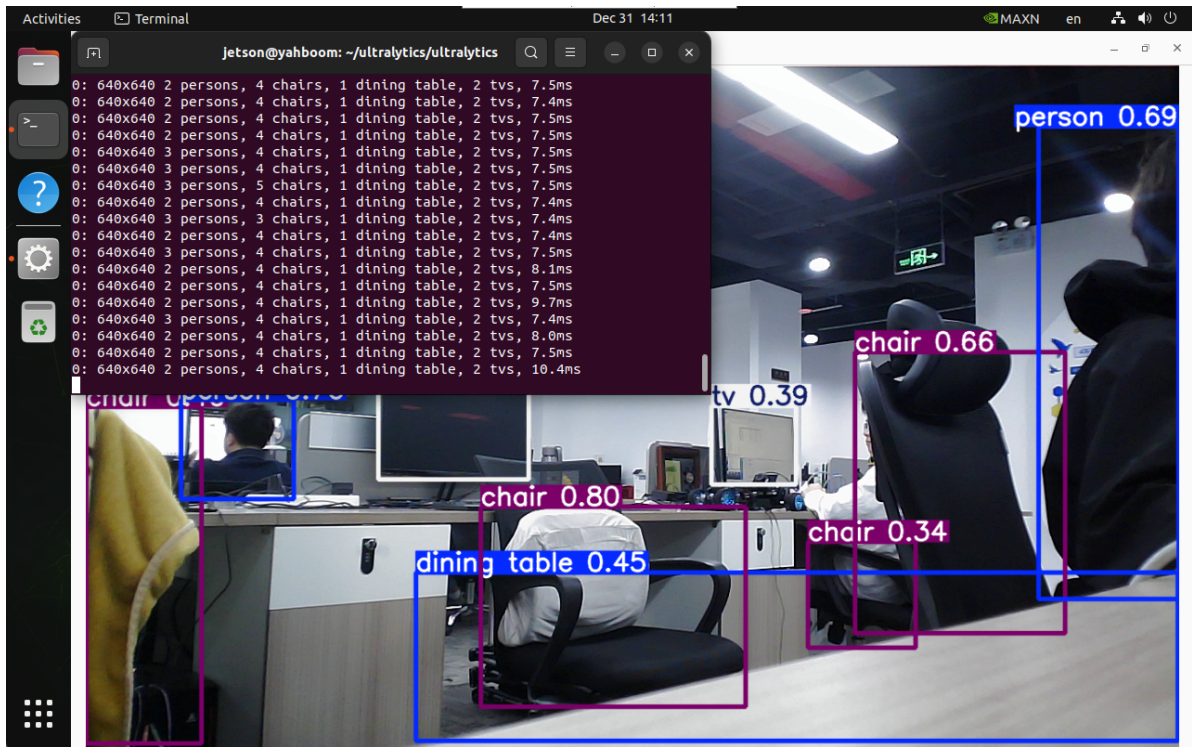
CLI currently only supports USB cameras. For CSI cameras, users can directly modify the previous Python code to call ONNX and ENGINE models!

```
cd /home/jetson/ultralalytics/ultralalytics
```

```
yolo predict model=yolo11n.onnx source=0 save=False show
```

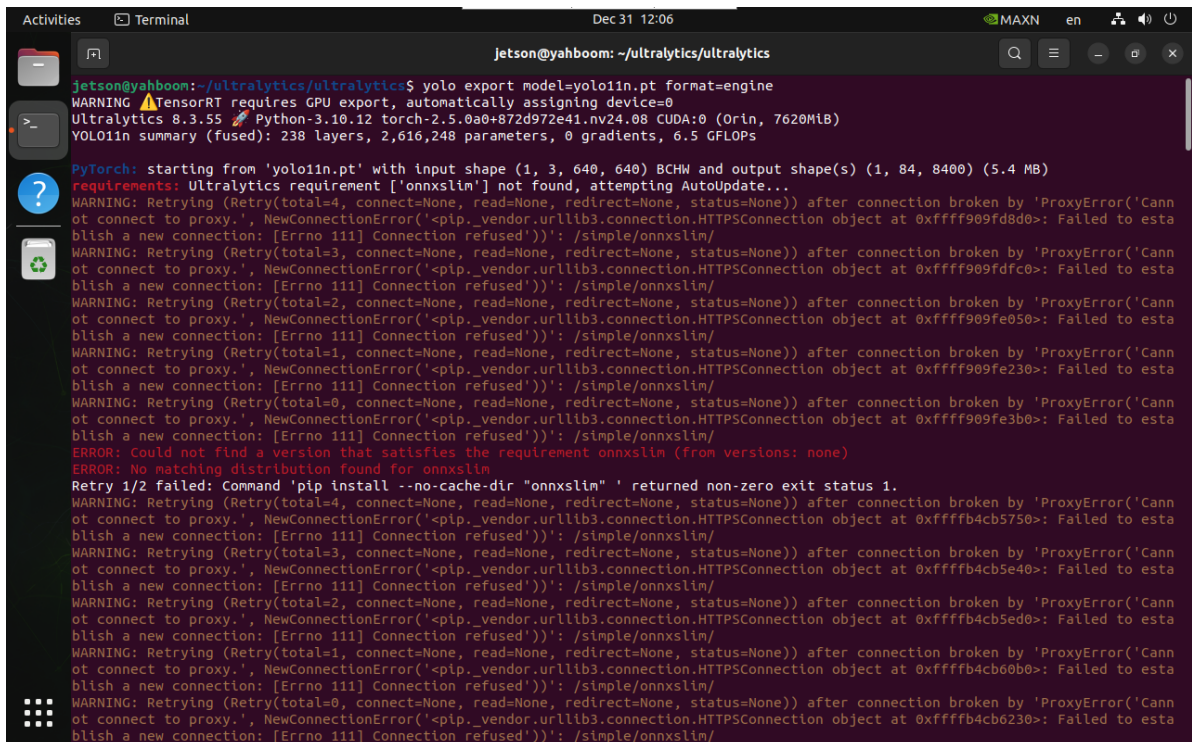


```
yolo predict model=yolo11n.engine source=0 save=False show
```

Common Issues

ERROR: onnxslim



Solution: Enter onnxslim installation command in terminal

```
sudo pip3 install onnxslim
```

```
Activities Terminal Dec 31 12:07 jetson@yahboom: ~/ultralytics/ultralytics

jetson@yahboom:~/ultralytics/ultralytics$ sudo pip3 install onnxslim
Collecting onnxslim
  Downloading onnxslim-0.1.45-py3-none-any.whl.metadata (4.2 kB)
Requirement already satisfied: onnx in /usr/local/lib/python3.10/dist-packages (from onnxslim) (1.17.0)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from onnxslim) (1.13.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from onnxslim) (23.2)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from onnx->onnxslim) (1.23.5)
Requirement already satisfied: protobuf>=3.20.2 in /usr/local/lib/python3.10/dist-packages (from onnx->onnxslim) (4.25.5)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy->onnxslim) (1.3.0)
Downloading onnxslim-0.1.45-py3-none-any.whl (142 kB)
Installing collected packages: onnxslim
Successfully installed onnxslim-0.1.45
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager,
possibly rendering your system unusable. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv. U
se the --root-user-action option if you know what you are doing and want to suppress this warning.
jetson@yahboom:~/ultralytics/ultralytics$ yolo export model=yolo11n-cls.pt format=engine
WARNING ⚠️TensorRT requires GPU export, automatically assigning device=0
Ultralytics 8.3.55 Python-3.10.12 torch-2.5.0a0+872d972e41.nv24.08 CUDA:0 (Orin, 7620MiB)
YOLO11n-cls summary (fused): 112 layers, 2,807,024 parameters, 0 gradients, 4.2 GFLOPs

PyTorch: starting from 'yolo11n-cls.pt' with input shape (1, 3, 224, 224) BCHW and output shape(s) (1, 1000) (5.5 MB)

ONNX: starting export with onnx 1.17.0 opset 19...
ONNX: slimming with onnxslim 0.1.45...
ONNX: export success ✅ 0.9s, saved as 'yolo11n-cls.onnx' (10.8 MB)

TensorRT: starting export with TensorRT 10.3.0...
[12/31/2024-12:05:11] [TRT] [I] [MemUsageChange] Init CUDA: CPU +2, GPU +0, now: CPU 654, GPU 3655 (MiB)
[12/31/2024-12:05:13] [TRT] [I] [MemUsageChange] Init builder kernel library: CPU +927, GPU +683, now: CPU 1624, GPU 4338 (MiB)
[12/31/2024-12:05:13] [TRT] [I] -----
[12/31/2024-12:05:13] [TRT] [I] Input filename:      yolo11n-cls.onnx
[12/31/2024-12:05:13] [TRT] [I] ONNX IR version:  0.0.9
[12/31/2024-12:05:13] [TRT] [I] Opset version:    19
[12/31/2024-12:05:13] [TRT] [I] Producer name:    pytorch
[12/31/2024-12:05:13] [TRT] [I] Producer version: 2.5.0
[12/31/2024-12:05:13] [TRT] [I] Domain:
[12/31/2024-12:05:13] [TRT] [I] Model version:    0
[12/31/2024-12:05:13] [TRT] [I] Doc string:
[12/31/2024-12:05:13] [TRT] [I] -----
TensorRT: input "images" with shape(1, 3, 224, 224) DataType.FLOAT
```

References

<https://docs.ultralytics.com/guides/nvidia-jetson/>

<https://docs.ultralytics.com/integrations/tensorrt/>