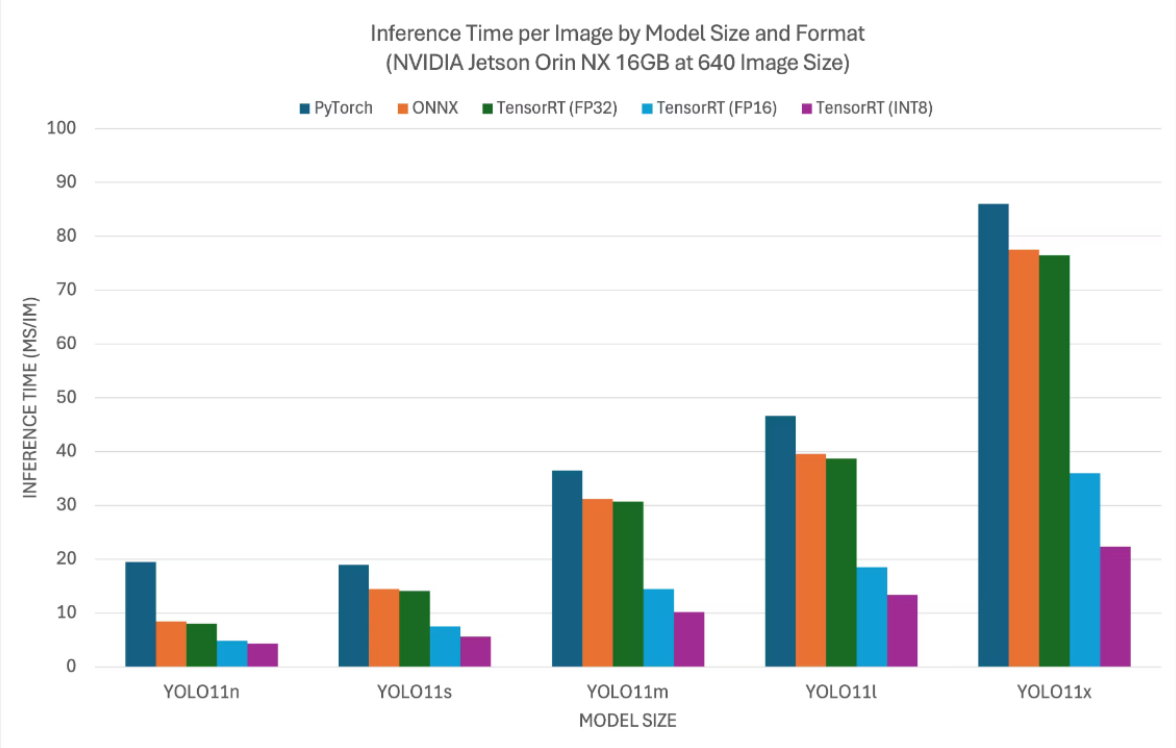
Model Conversion

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 nvpmodel -m 2
#Orin Nx
sudo nvpmodel -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!

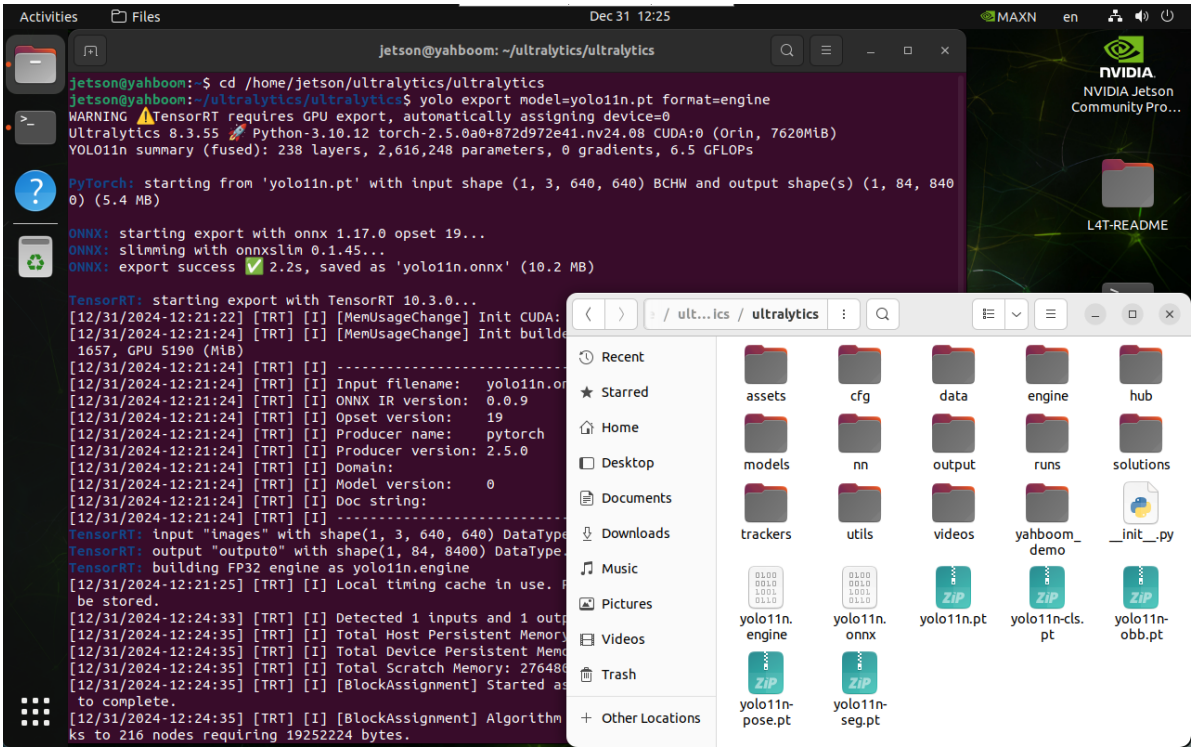
```
First time using YOLO11 export mode will automatically install some dependencies,
please wait for the automatic completion!
```

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/ultralytics/ultralytics/yahboom_demo
```

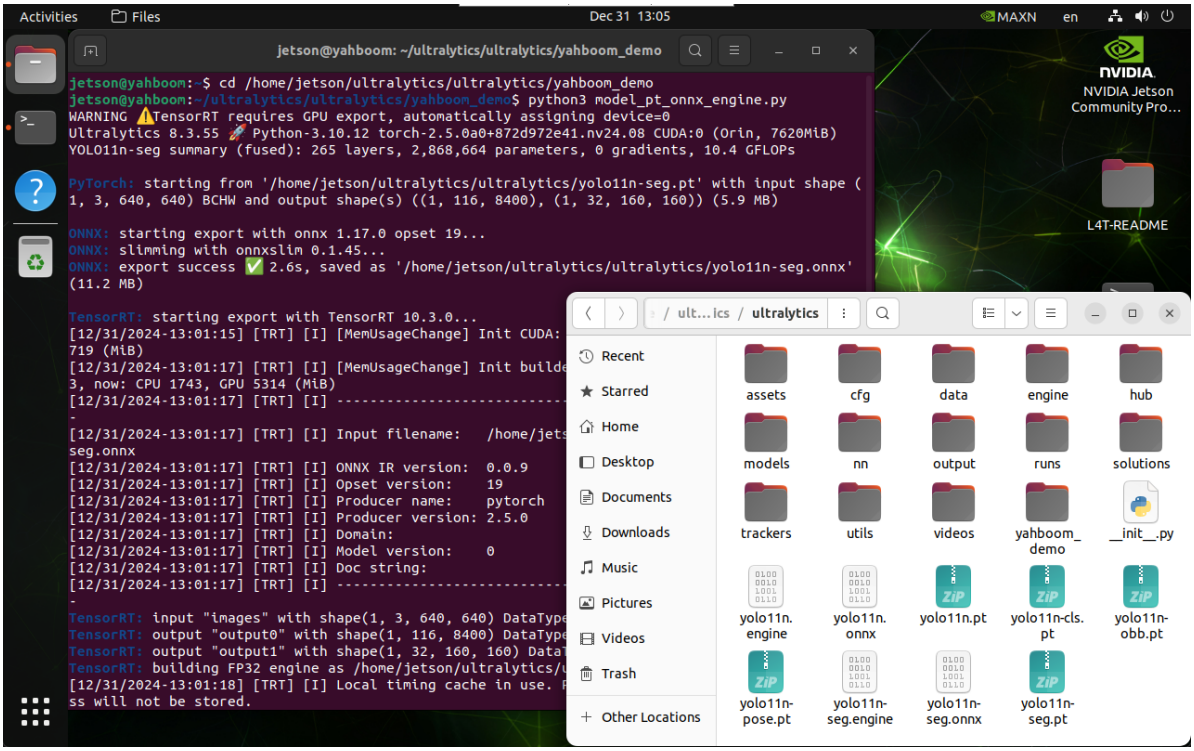
```
python3 model_pt_onnx_engine.py
```

```
from ultralytics import YOLO

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



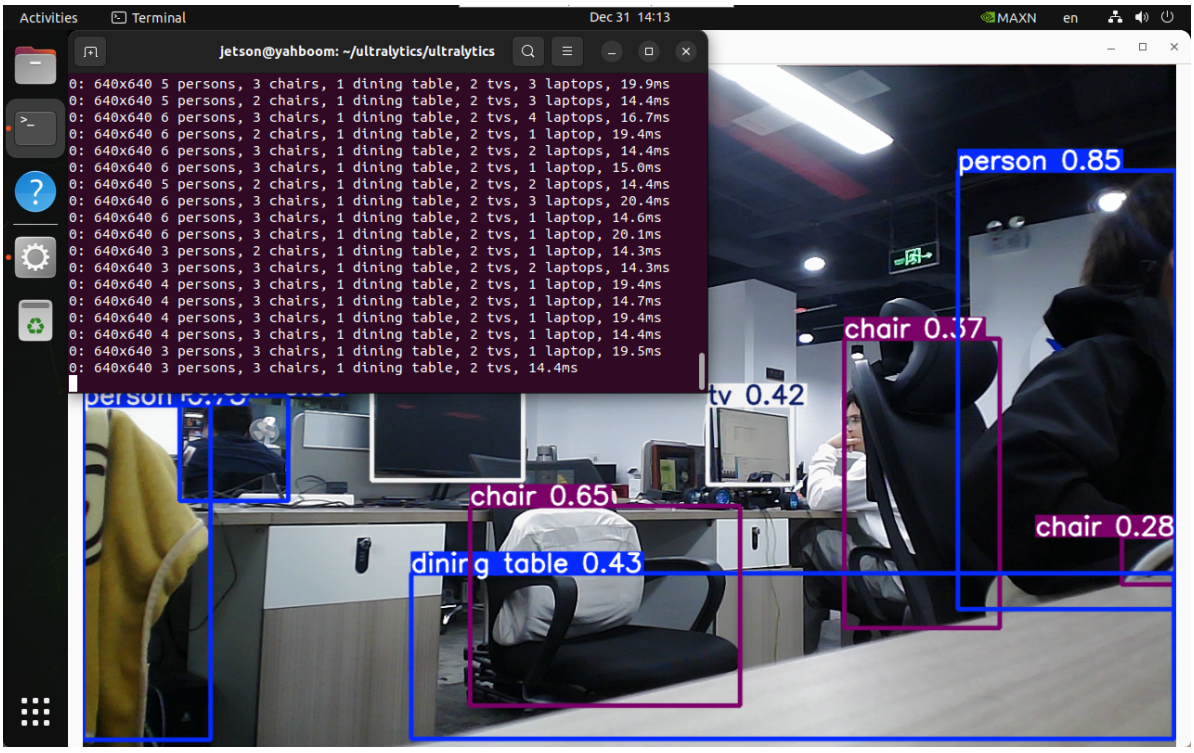
4. Model Prediction

CLI Usage

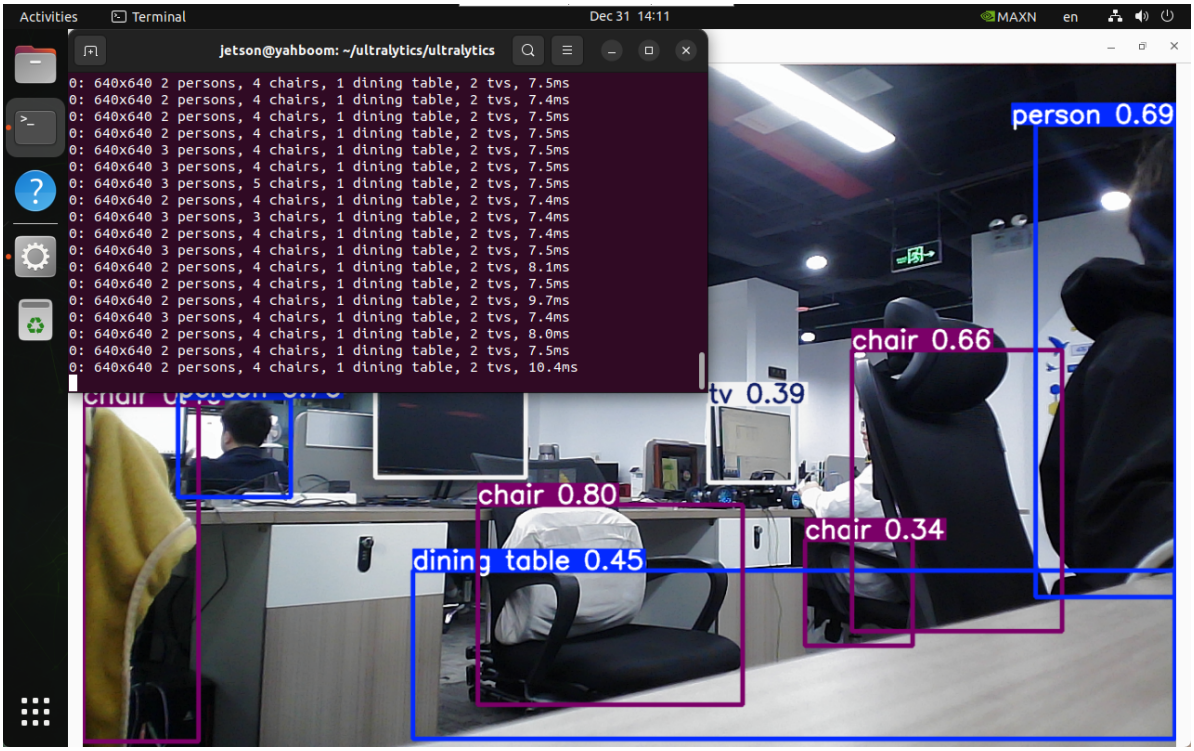
CLI currently only supports USB cameras. For CSI cameras, users can directly modify the previous Python code to call ONNX and ENGINE models!

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

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

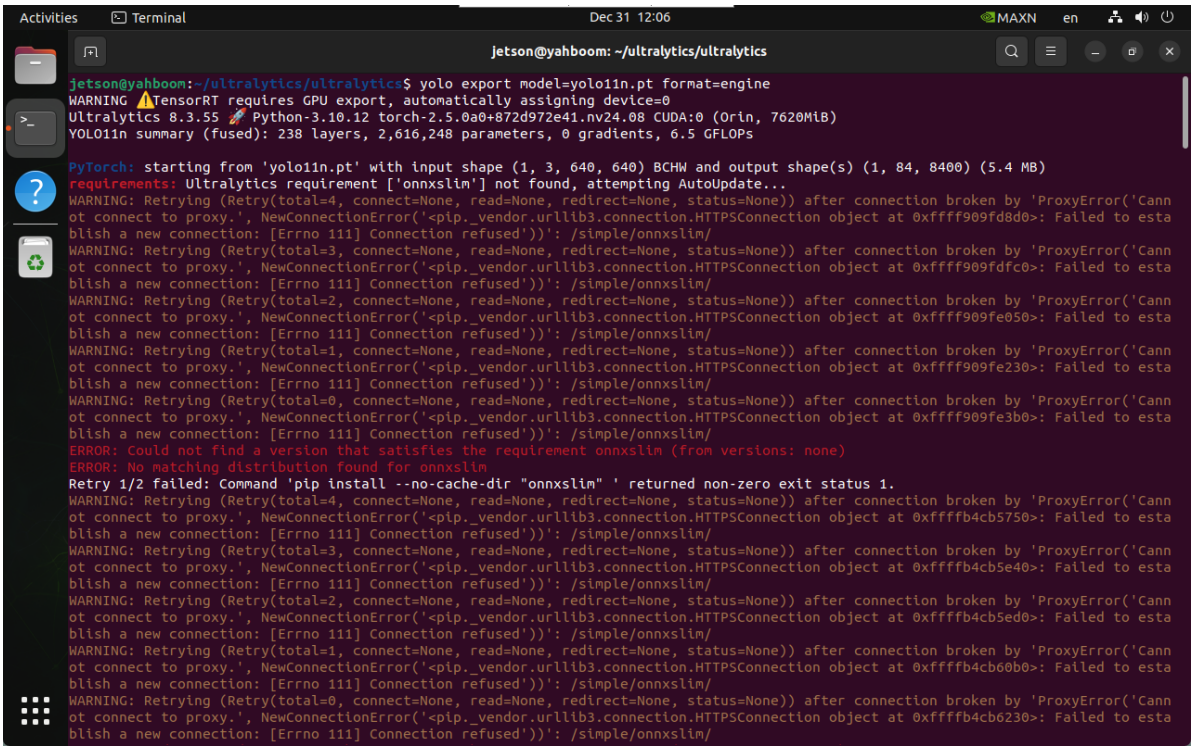



```
yolo predict model=yolo11n.engine source=0 save=False show
```



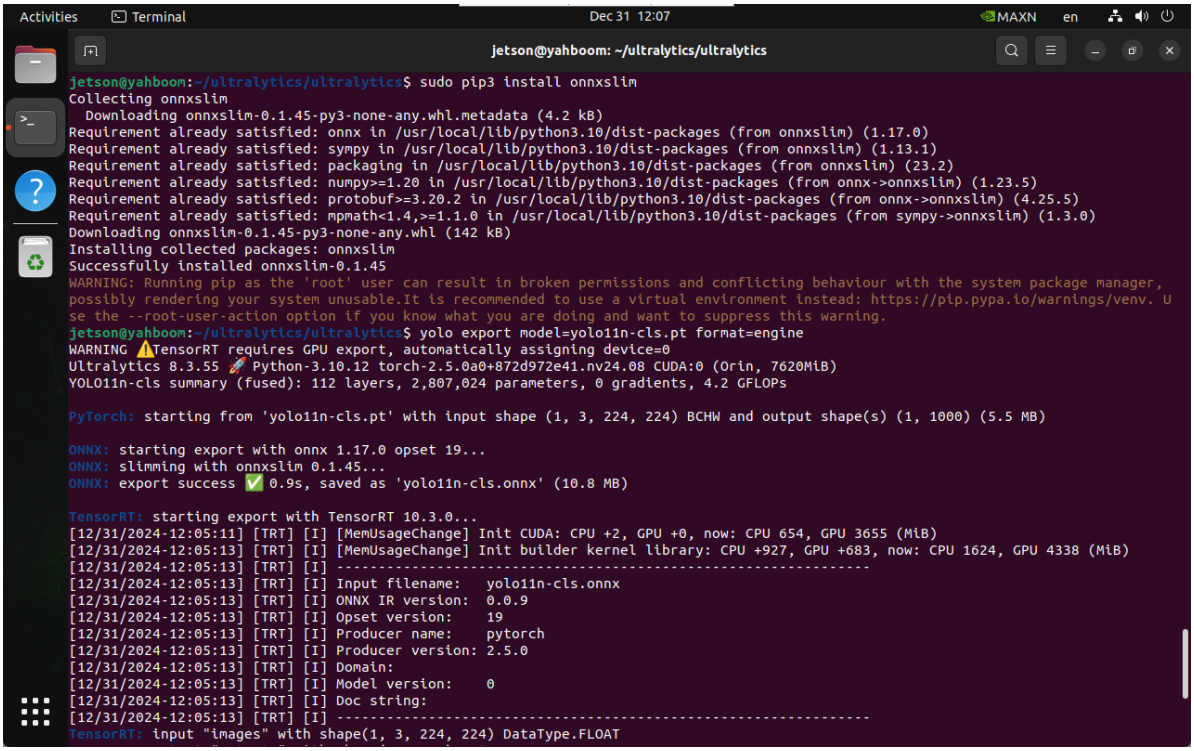
Common Issues

ERROR: onnxslim



Solution: Enter onnxslim installation command in terminal

```
sudo pip3 install onnxslim
```



References

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

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