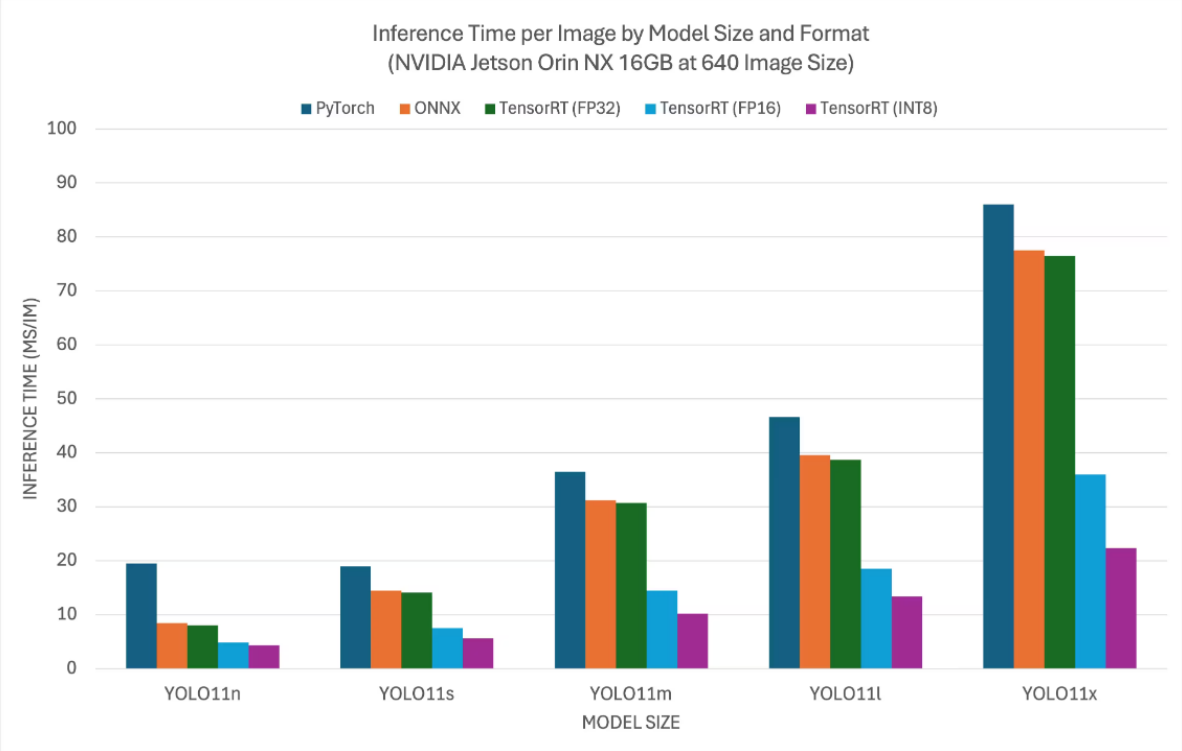
YOLOv11Model conversion

1. Jetson Orin YOLO11 (benchmark)

YOLO11 benchmark data comes from the Ultralytics team, which tests models in multiple formats (data is for reference only)



2. Enable optimal performance of the motherboard

2.1. Enable MAX power mode

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

```
# Jetson orin nano
sudo nvpmodel -m 2
# Jetson orin nx
sudo nvpmodel -m 2
```

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 of different format models provided by the Ultralytics team, we can find that the inference performance is best when using TensorRT!

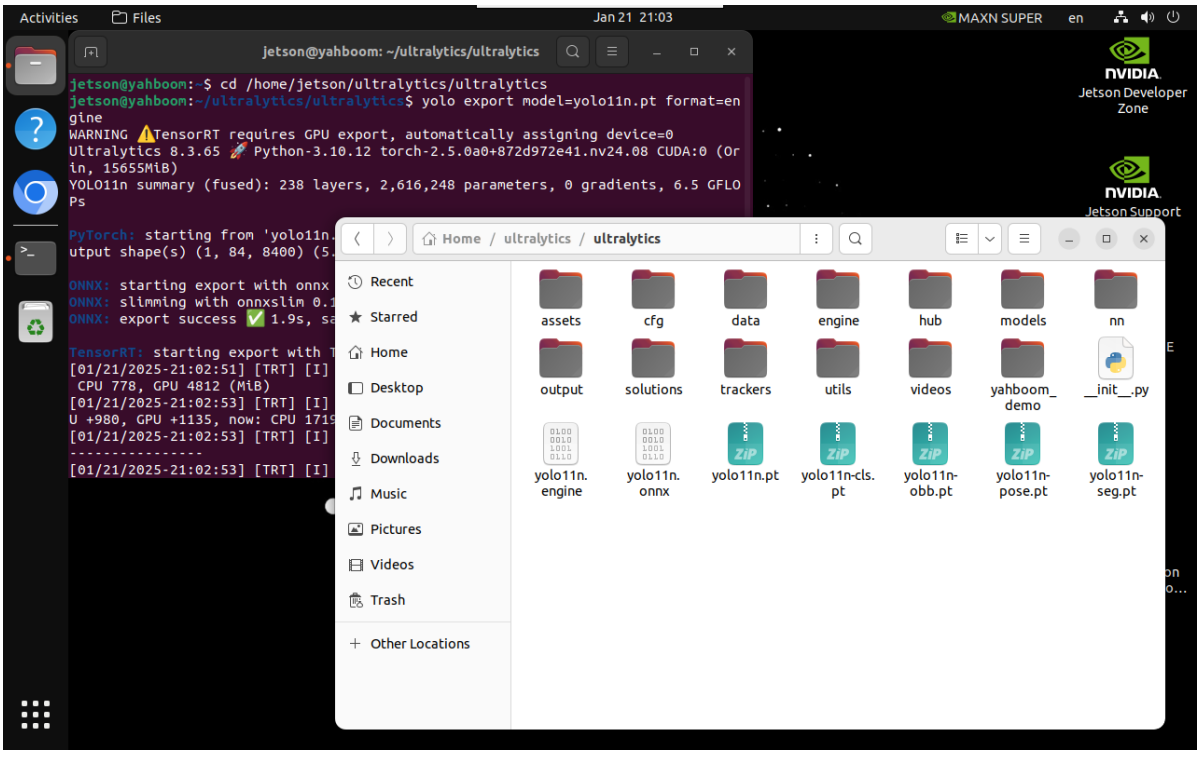
```
When using the export mode of YOLO11 for the first time, some dependencies will
be automatically installed. Just wait for it to be completed automatically!
```

3.1、CLI：pt → onnx → engine

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

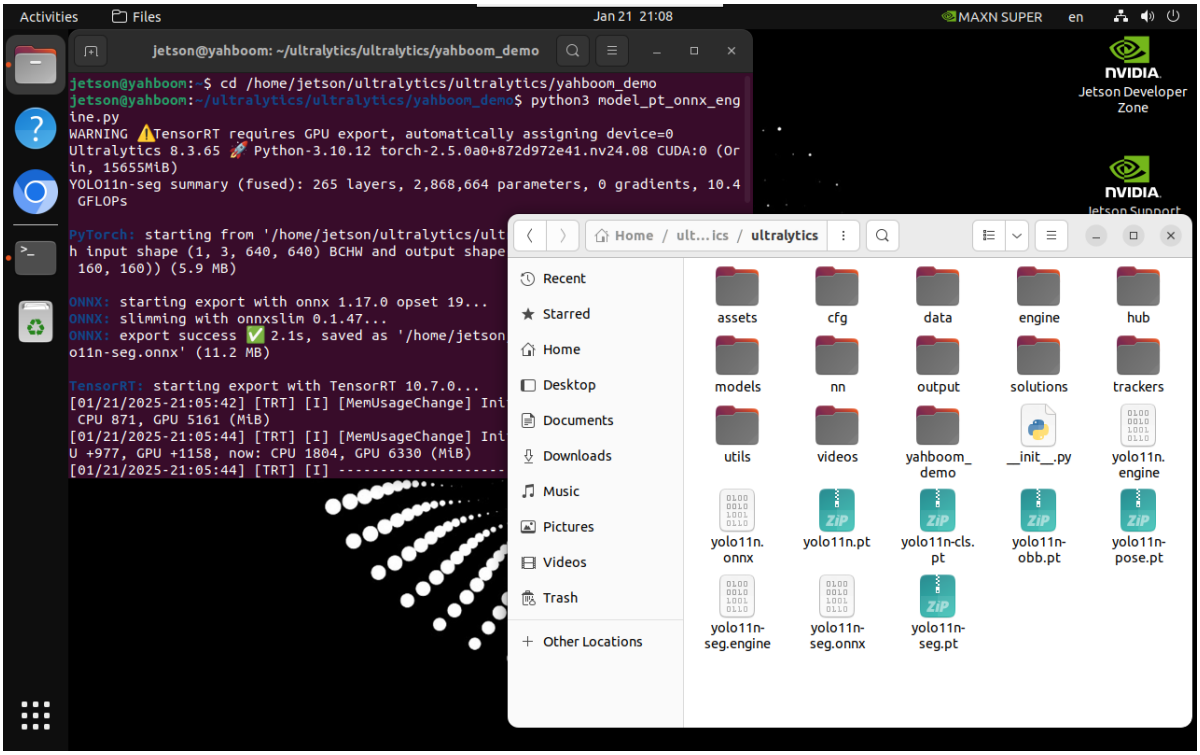
```
cd /home/jetson/ultralytics/ultralytics/yahboom_demo
```

```
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 file is located in the converted model file location



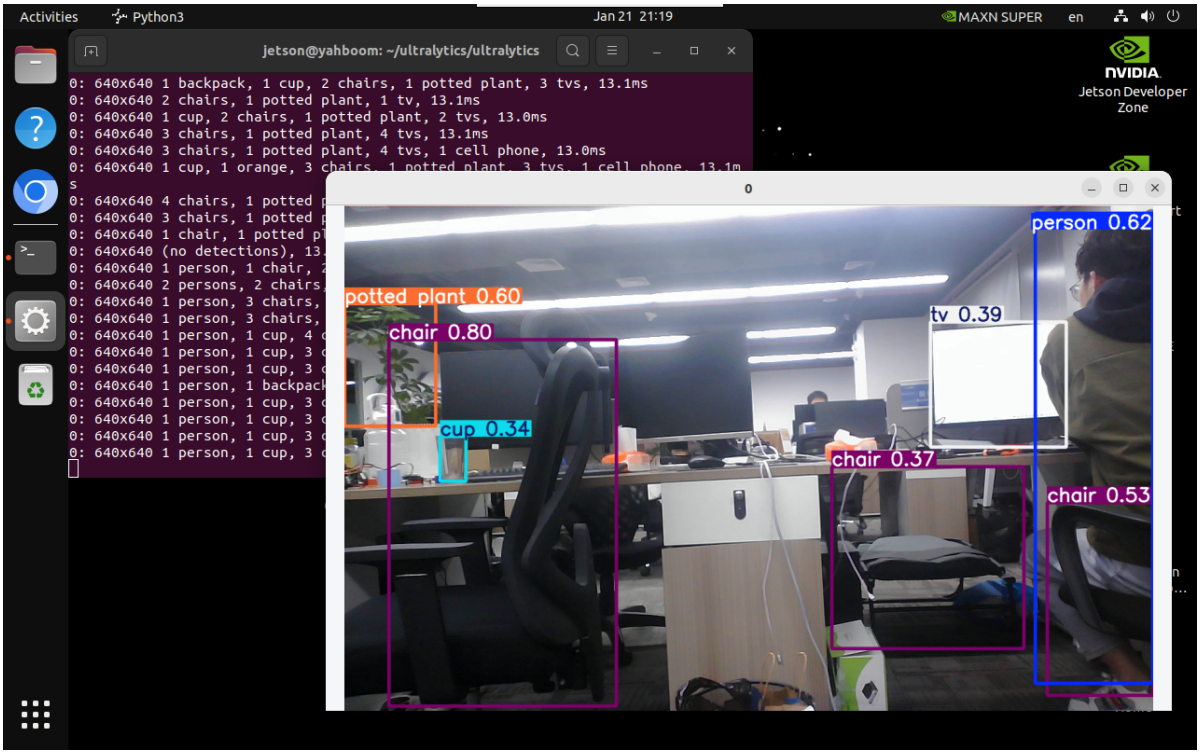
4. Model prediction

CLI usage

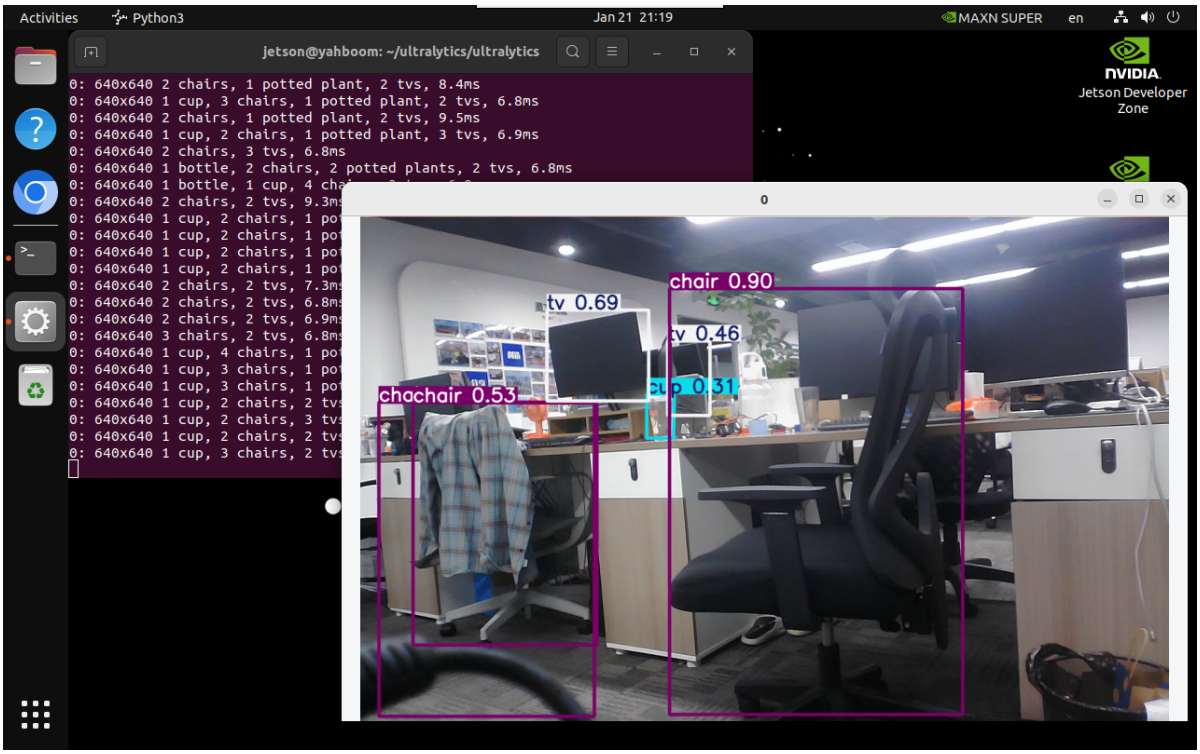
CLI currently only supports calling USB cameras. CSI camera users can directly modify the previous python code to call onnx and engine models!

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

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

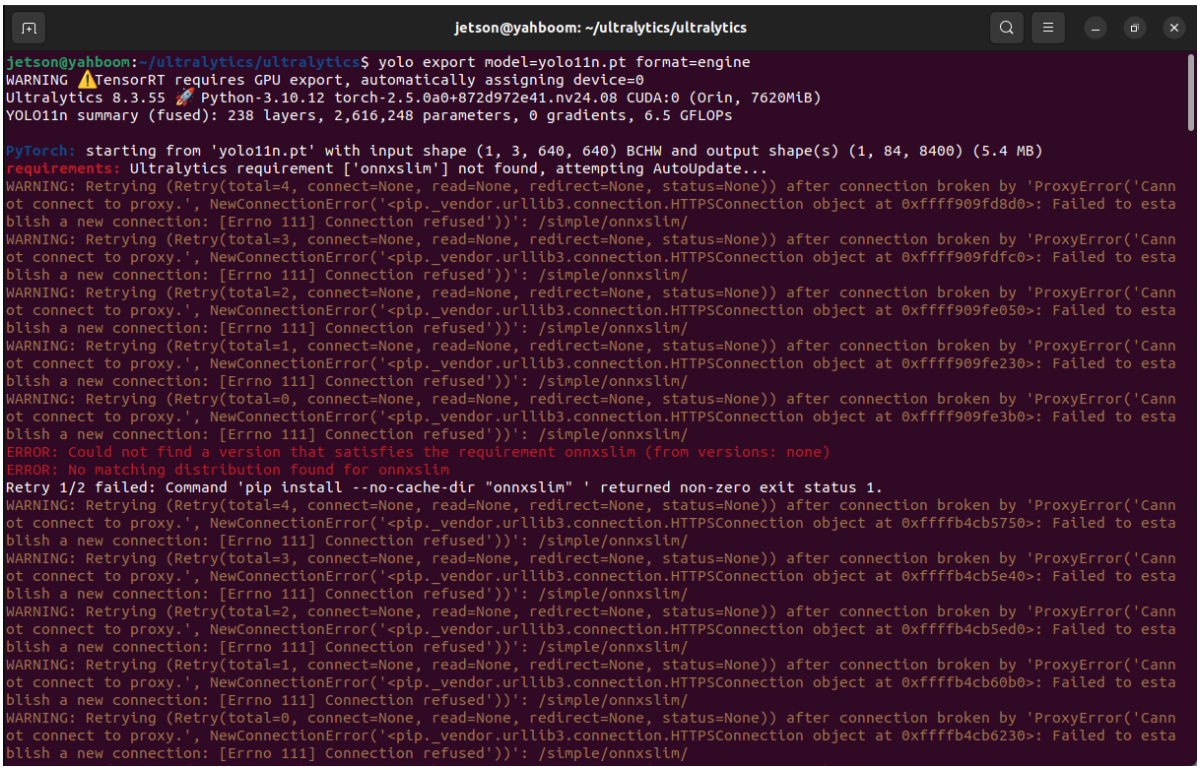



```
yolo predict model=yolo11n.engine source=0 save=False show
```



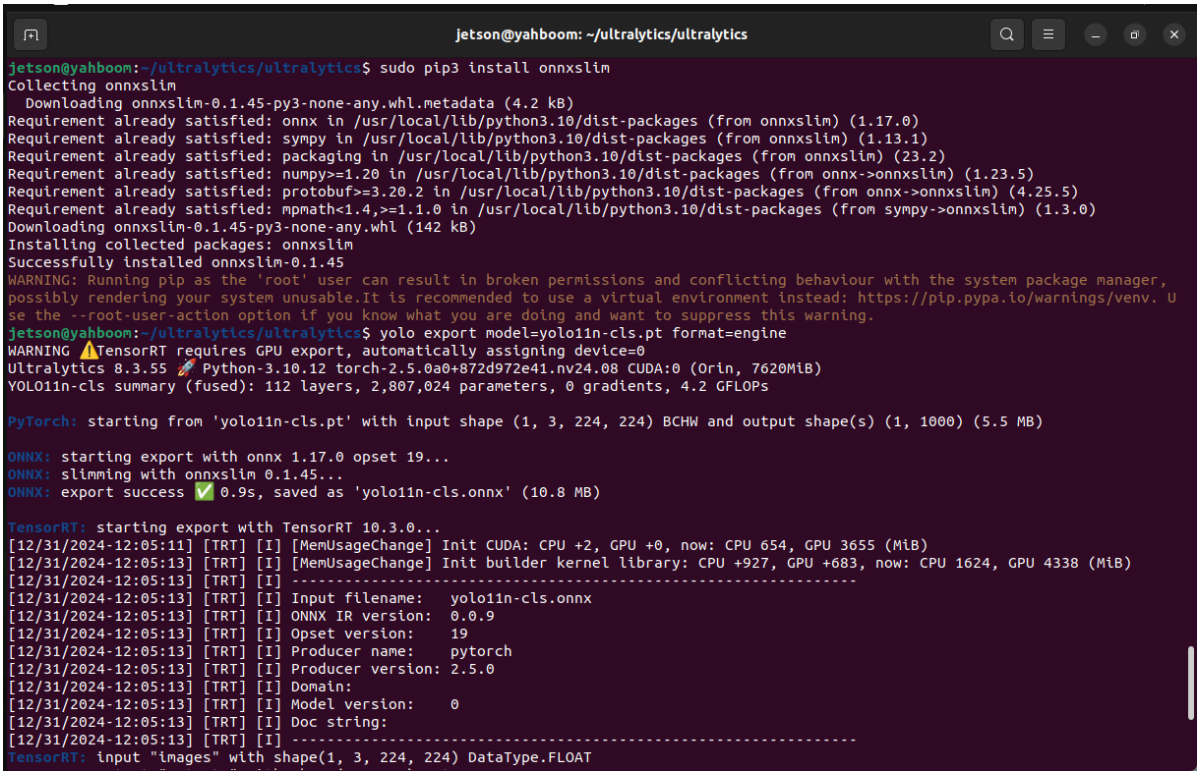
Frequently Asked Questions

ERROR: onnxslim



Solution: Enter the onnxslim installation command in the terminal

```
sudo pip3 install onnxslim
```



References

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

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