+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 10/14/2026

Sign Up

TensorRT Training with Ultralytics YOLO

2 days (14 hours)

Overview

Combine TensorRT and Ultralytics YOLO to develop detection applications tailored to production constraints. Explore FP32, FP16, and INT8 precisions as well as compilation settings to find the optimal balance between speed, prediction quality, and GPU memory consumption.

Our TensorRT with Ultralytics YOLO training will help you transition from a trained model to an optimized inference engine by understanding the technical choices that determine its performance on your target hardware.

You'll learn how to set up a benchmark, master the ONNX and TensorRT workflow, and configure input dimensions, batches, and build memory. You'll also experiment with INT8 quantization using representative data to evaluate its impact on detection.

By the end of the training, you'll be able to compare configurations based on latency, throughput, and mAP, and then integrate the selected engine into an image or video processing pipeline.

This training also covers profiling and error diagnosis, as well as the reproducibility of exports and compatibility constraints, to prepare for reliable and measurable deployments.

Like all our training courses, this one will introduce you to **the latest stable version** of the technology and its new features.

Objectives

- Prepare a quality and performance benchmark for a YOLO detection model
- Export an Ultralytics YOLO model to a TensorRT engine optimized for the target hardware
- Configure input dimensions, batches, and computation precisions
- Evaluate trade-offs between latency, throughput, memory, and detection quality
- Integrate and validate an optimized engine into a reproducible inference pipeline

Target Audience

- AI Engineer
- Computer Vision Engineer
- Data Scientist
- Python Developer
- MLOps Engineer

Prerequisites

- Proficiency in Python for writing and running data processing scripts
- Knowledge of tensors and inference with PyTorch or equivalent experience
- First-time use of Ultralytics YOLO for model prediction or validation
- Understanding of object detection principles and precision and recall metrics

Technical Prerequisites

- Computer running Ubuntu 22.04 LTS or Ubuntu 24.04 LTS with installation privileges, or remote access to an equivalent environment
- Recommended configuration for workshops: 4-core processor, 16 GB of RAM, 30 GB available and an NVIDIA Turing GPU or newer with 8 GB of video memory
- Python 3.10 environment with PyTorch and Ultralytics in versions that have been finalized and validated prior to the session
- Compatible NVIDIA driver, CUDA stack, and TensorRT, with trtexec installed via a system package, a validated archive or container
- Visual Studio Code or equivalent editor and a stable internet connection allowing for the download of dependencies, models, and datasets

Agenda for our TensorRT with Ultralytics YOLO training course

[Day 1 - Morning]

Prepare the model and benchmark data

- Integrating TensorRT into the Ultralytics YOLO model deployment workflow
- Environment check: GPU, NVIDIA driver, CUDA, and dependencies
- Anatomy of the detection pipeline: preprocessing, inference, and postprocessing

- Preparing images and annotations; distinguishing between calibration data and an independent evaluation dataset
- Building a PyTorch baseline: detection quality, latency, throughput, and GPU memory
- Hands-on Workshop: Load a pre-trained YOLO model, run a validation, and log benchmark metrics on the target GPU.

[Day 1 - Afternoon]

Exporting and Building a TensorRT Engine

- PyTorch, ONNX, and TensorRT engine conversion pipeline: graph, operators, and artifacts
- Exporting with Ultralytics: engine format, resolution, batch size, and build memory budget
- Static and dynamic dimensions: minimum, optimal, and maximum dimension profiles
- Choosing FP32 and FP16 precision based on hardware and validation results
- Output handling: coordinates, scores, classes, and NMS based on the model architecture
- Hands-on Workshop: Building FP32 and FP16 engines, loading them with Ultralytics, and comparing their predictions to the ground truth.

[Day 2 - Morning]

Quantifying and measuring trade-offs

- Principles of INT8 quantization and potential effects on scores and object localization
- Preparing a representative calibration: diversity of scenes, objects, and acquisition conditions
- Configuring the quantized export and reading logs based on the TensorRT version
- Benchmark protocol with trtexec: warm-up, repetitions, synchronization, and CPU-GPU transfers
- Comparison of FP32, FP16, and INT8: mAP, latency, throughput, and memory consumption under

- Hands-on workshop: Export an INT8 engine with calibration, measure its quality on an indie game, and select a configuration based on acceptance criteria.

[Day 2 - Afternoon]

Integrating and ensuring reliable deployment

- Integrating the engine into a Python pipeline for image and video detection
- Measuring end-to-end latency and identifying the costs of decoding, preprocessing, and postprocessing
- Diagnosing errors related to dimensions, memory, operators, and deserialization
- Engine compatibility constraints across GPUs, platforms, and versions; reconstruction strategy
- Deployment reproducibility: release builds, export settings, non-regression tests, and performance monitoring

- Hands-on workshop: Deploy a video pipeline using the selected engine, verify quality and latency criteria, and then document its reconstruction.

Target Audience

This training is intended for both individuals and companies—large or small—seeking to train their teams in a new advanced IT technology or to acquire specific domain knowledge or modern methodologies.

Assessment upon enrollment

The pre-training assessment complies with Qualiopi quality standards. Upon final registration, the learner receives a self-assessment questionnaire that allows us to evaluate their estimated proficiency with various types of technologies, as well as their expectations and personal goals for the upcoming training, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could pose challenges for monitoring and ensuring the smooth running of the training session.

Teaching Methods

Hands-On Training: 60% Practical, 40% Theoretical. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the instructor—supported by examples and reflection sessions—and group work.

Assessment

At the end of the session, a multiple-choice quiz is administered to verify that participants have successfully acquired the skills.

Certification

A certificate will be issued to each trainee who has completed the entire training program.

Training organization registered under number 11 75 54743 75. This registration does not constitute state accreditation.
© Ambient IT 2015-2026. All rights reserved. Paris, France - Switzerland - Belgium - Luxembourg