



Edge AI and TinyML Training Course: Embedded Machine Learning for Industrial IoT Devices



Edge AI and TinyML Training Course: Embedded Machine Learning for Industrial IoT Devices

Introduction:

Edge AI and TinyML training on embedded machine learning for industrial IoT devices is the subject of this 5-day course for embedded, automation, instrumentation and IoT engineering functions, who produce an Edge AI Deployment Plan for a case plant or device fleet. Many plants want machine learning on cameras, vibration sensors and gateways, yet models built for servers overflow microcontroller memory, drain batteries or wait on a network link that drops. Nominees already build or integrate field devices and work through hands-on lab sessions on microcontroller boards and edge modules. CoreConcept Training Center delivers this edge AI and TinyML course.

Course Objectives:

- Qualify industrial use cases for device, gateway or cloud inference using latency, bandwidth and connectivity criteria
- Select microcontrollers, NPUs and GPU modules against memory, throughput and energy budgets for a given workload
- Compress trained models through INT8 quantisation, pruning and knowledge distillation while tracking accuracy loss
- Deploy vision, keyword and vibration models with TensorFlow Lite Micro and ONNX Runtime on target boards
- Build on-device anomaly detection for accelerometer and acoustic signals with duty-cycled power profiles
- Plan over-the-air model delivery and fleet monitoring that flag degraded devices before output quality drops

Target Audience:

- Embedded engineering functions that write firmware for sensors, controllers and smart cameras
 - Automation and instrumentation functions responsible for condition monitoring on plant equipment
 - IoT solution functions that integrate gateways, field devices and plant data platforms
 - Machine vision functions responsible for inspection cameras on production and packing lines
 - Reliability engineering functions that add sensing and analytics to rotating assets
-

Course Outline:

Day 1: Edge AI Landscape and Use-Case Qualification

- Cloud, Gateway and Device Inference Tier Comparison
- Latency, Bandwidth and Connectivity Criteria for On-Device Placement
- TinyML Device Classes From Microcontrollers to Edge Modules
- Industrial Use-Case Screening Matrix for Cameras and Sensors
- Current-State Inventory of Plant Assets and Data Streams

Day 2: Edge Hardware Architecture and Inference Runtimes

- Microcontroller Memory Map With Flash, SRAM and Tensor Arena
- NPU, VPU and GPU Module Selection Using TOPS Ratings
- TensorFlow Lite Micro Interpreter and Operator Resolver Setup
- ONNX Runtime and Vendor SDK Execution Provider Comparison
- MLPerf Tiny Benchmark Results for Hardware Shortlisting

Day 3: Model Compression and On-Device Inference Builds

- INT8 Post-Training Quantisation With Representative Calibration Samples
- Quantisation-Aware Retraining to Recover Lost Classifier Accuracy
- Structured Pruning for Smaller Flash and RAM Footprint
- Knowledge Distillation From Teacher Network to Compact Student
- Keyword Spotting and Vibration Classifier Flashing on Microcontroller Boards

Day 4: On-Device Vision, Anomaly Detection and Fleet Operations

- Low-Resolution Camera Defect Detection Within Frame Rate Budgets
- Autoencoder Anomaly Scoring on Accelerometer and Acoustic Signals
- Power Budget Profiling With Duty Cycling and Sleep Modes
- Over-the-Air Model Delivery Using Staged Fleet Rollout Rings
- On-Device Confidence Logging and Drift Indicators for Fleet Dashboards

Day 5: Lab on a Case Plant and Edge AI Deployment Plan

- Case Plant Briefing for Pump, Conveyor and Camera Sites
 - Hardware Shortlist and Memory Budget for Case Devices
 - Compressed Model Accuracy, Latency and Energy Trade-Off Table
 - Fleet Update and Monitoring Rules for the Case Rollout
 - Edge AI Deployment Plan Completion and Peer Design Review
-

Skills You Will Gain:

- Inference Placement Analysis
- Edge Hardware Selection
- Model Quantisation
- Neural Network Pruning
- Microcontroller Model Deployment
- Vibration Anomaly Detection
- Low-Power Inference Design
- Fleet Model Rollout

Why Attend This Course:

- Deliver an Edge AI Deployment Plan for a case plant or device fleet to the engineering manager and the operations technology lead
- Decide whether a model belongs on a microcontroller, an edge module, a gateway or the cloud, using measured memory, latency and energy figures
- Avoid stalled pilots, flat batteries and devices that stop working when the plant network drops by sizing models to the hardware before purchase
- Share the use-case screening matrix, hardware shortlist template and compression checklist with embedded and automation colleagues

Conclusion:

Back at work, the participant hands the Edge AI Deployment Plan to the engineering manager and the operations technology lead, who use its hardware shortlist, memory and energy budgets and compression results to decide which devices to buy and which models to run on them first. The fleet update and monitoring rules guide how new model versions reach sensors and cameras. After the first rollout under the plan, the unit should review on-device accuracy against the lab figures, battery life, failed updates and devices showing drift.

Frequently Asked Questions FAQ:

What should participants know before an edge AI and TinyML course for industrial IoT devices?

Participants should read basic Python, understand how a trained classifier is evaluated and have worked with microcontroller or embedded Linux boards. Neural network theory beyond these basics is not required. Bringing a description of one sensor or camera use case from their site makes the lab work more relevant.

How does edge AI and TinyML for industrial IoT devices differ from a deep learning or MLOps course?

This course starts from an already trained model and fits it onto constrained hardware: memory, power, compression and fleet updates on devices. Deep learning courses focus on building and training networks, and MLOps courses run models as cloud or data-centre services.

Why does quantisation matter for embedded machine learning on IoT devices?

Quantisation stores weights and activations as 8-bit integers instead of 32-bit floats, shrinking the model and allowing integer-only NPUs and microcontrollers to run the model quickly and with less energy, usually with only a small loss of accuracy.

What does a participant take back from the edge AI and TinyML course?

Each participant takes back an Edge AI Deployment Plan for a case plant or fleet, with a use-case screening matrix, hardware shortlist, memory and energy budgets, compression results and fleet update and monitoring rules ready to adapt to devices in their own organisation.