EdgeGuard: Real-Time Industrial Safety Monitoring using NPU Acceleration on NXP i.MX 8M Plus

Robert-Ionut Rascanu[1], Alexandra-Petronela Bucataru[2]

Abstract

Industrial environments require strict adherence to Personal Protective Equipment (PPE) regulations to prevent injuries.

Traditional monitoring methods rely on human supervision or cloud analytics, which suffer from latency and bandwidth constraints. We propose **EdgeGuard**, an autonomous vision system deployed on the NXP i.MX 8M Plus.
By leveraging the integrated VeriSilicon Neural Processing Unit (NPU), we executed a quantized YOLOv5 model locally. The system achieved a mean Average Precision (mAP@0.5) of **87.4%** with an inference speed of under **12ms**, enabling real-time detection of safety helmets without internet dependency.

Methods and Materials

Hardware Setup:

- **Device:** NXP i.MX 8M Plus EVK.
- **Accelerator:** Integrated VeriSilicon NPU (Neural Processing Unit) using the libvx_delegate driver.
- **Connectivity:** Ethernet-based local network setup for development (SSH/Serial debugging).

Model Architecture:

- **Framework:** YOLOv5 Nano trained via Transfer Learning on a custom dataset of PPE images.
- **Optimization:** Post-Training Quantization (PTQ) was applied to convert weights from 32-bit floating-point to **INT8**. This reduced model size to ~1.9MB for NPU compatibility.

Software Pipeline:

- Data augmentation techniques were applied to the training set (300+ images, tight-fit annotation).
- Inference logic was written in Python using tflite_runtime.
- Custom post-processing logic handles the dequantization of NPU outputs to map bounding boxes to the original video frame.

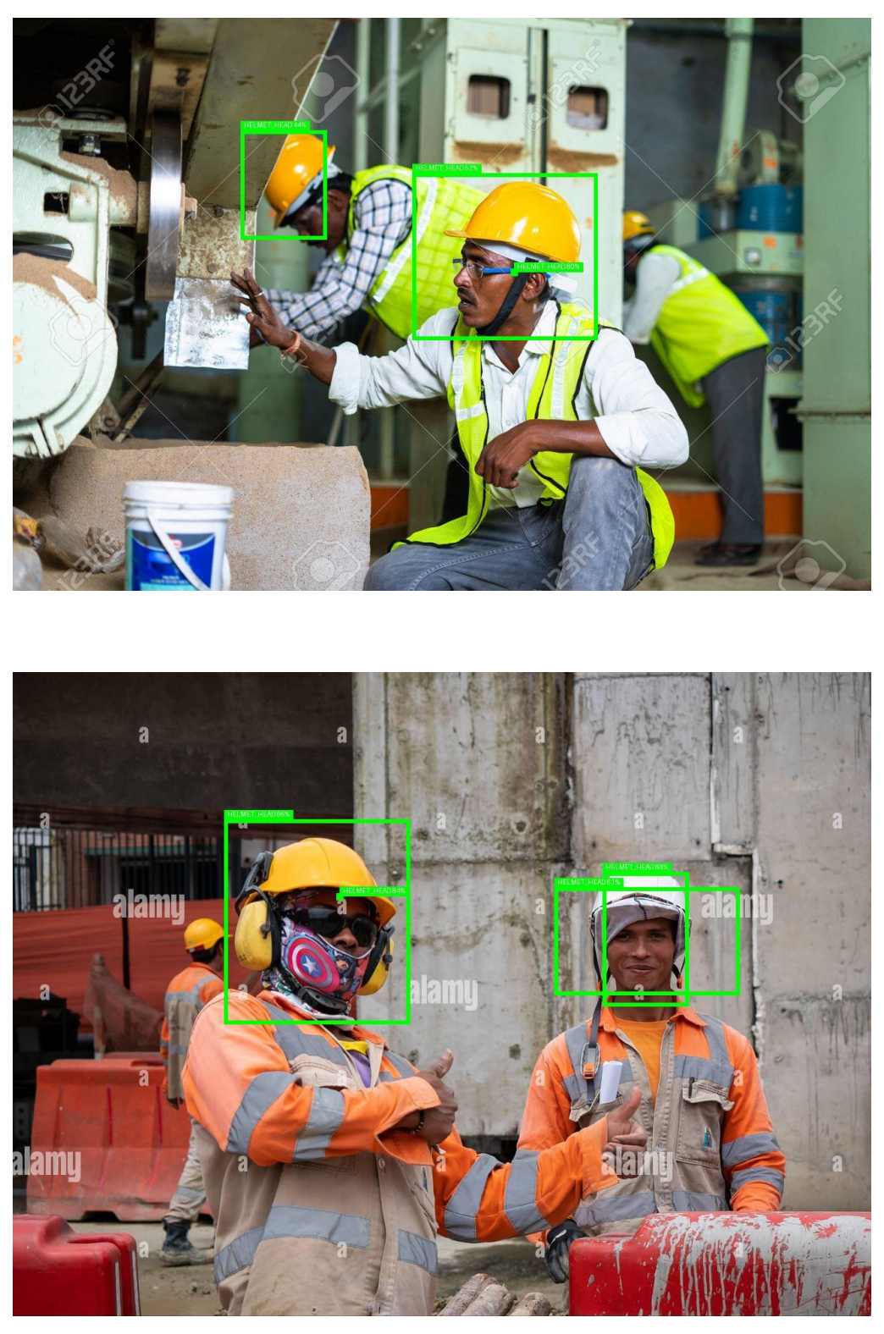

Figure 1. Helmet identification

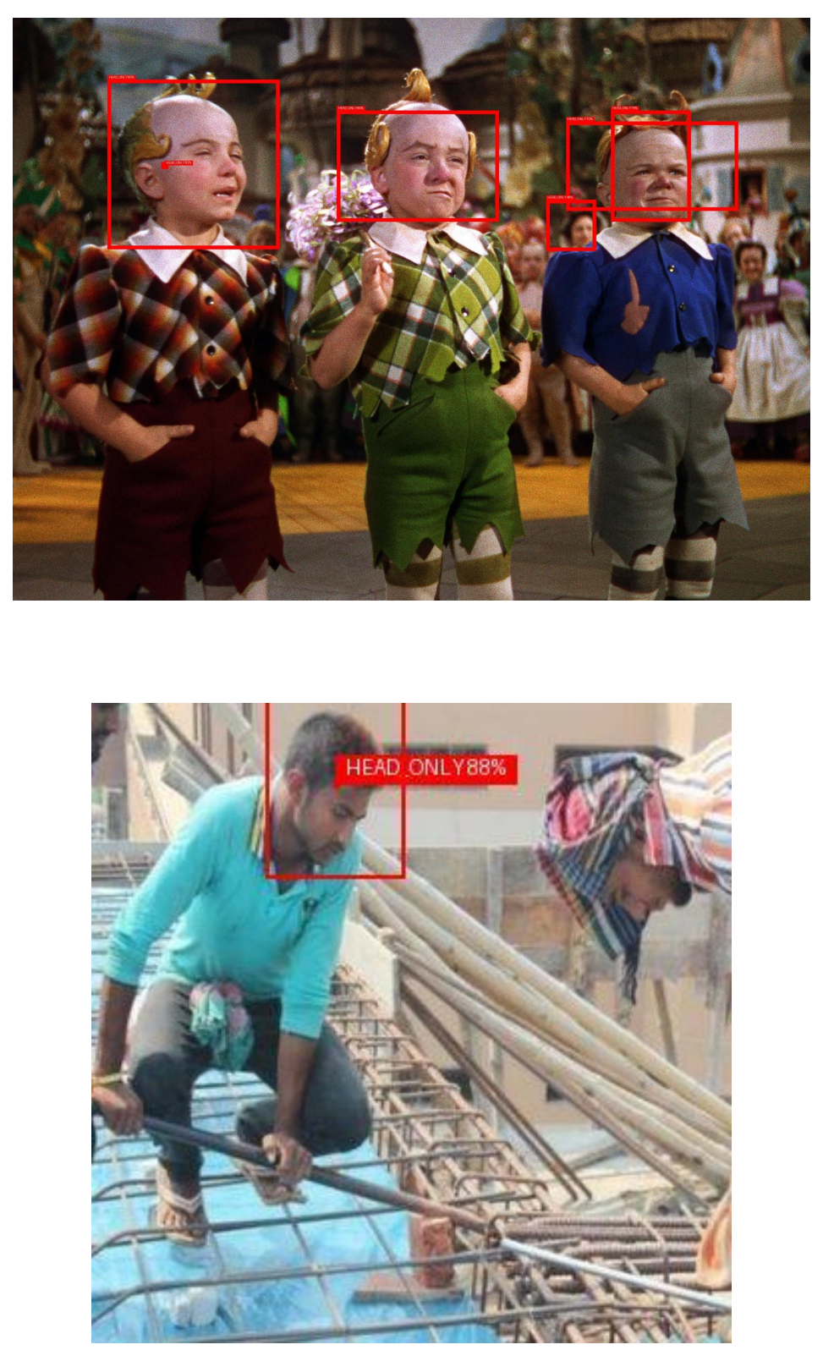

Figure 2. No helmet worn identification

Introduction

Compliance with safety regulations is critical in high-risk sectors like construction and manufacturing. However, sending video feeds to the cloud for analysis introduces unacceptable latency and privacy risks.

- **The Shift to Edge:** Computing must move closer to the data source to reduce reaction time and energy consumption.
- **Hardware Acceleration:** Standard CPUs (like the Cortex-A53) are insufficient for real-time video processing, often exceeding 100ms per frame.
- **Objective:** To implement a specialized object detection pipeline using the i.MX 8M Plus NPU to detect "Hard Hats" vs. "Heads" instantly and locally

Results

1. Model Performance After training for 100 epochs, the quantized YOLOv5 model achieved excellent metrics:

- **mAP@0.5:** 87.44%
- **Precision:** 99.5% (Extremely low false alarm rate)
- **Recall:** 79.5%

2. Inference Speed (CPU vs. NPU)

- **CPU Inference:** ~4000ms - 4700ms (Unusable for real-time).
- **NPU Inference: ~3ms to 11ms**.
- **Throughput:** The system theoretically supports >80 FPS, providing ample headroom for real-time processing (30 FPS target).

3. Visual Detection

- **Green Bounding Box:** Correctly identifies workers wearing helmets.
- **Red Bounding Box:** Identifies "Head only" (Safety Violation).

Table 1. CPU vs. NPU (Hardware Impact)

Metric	CPU (Standard)	NPU (EdgeGuard)	Improve-ment
Inference Time	4761 ms	11.2 ms	425x Faster
Throughput (FPS)	0.2 FPS	89.2 FPS	Real-Time
Model Size	7.5 MB (FP32)	1.9 MB (INT8)	4x Smaller
Power Efficiency	Critical	Instant	976
Safety Check	Delayed	Immediate	301

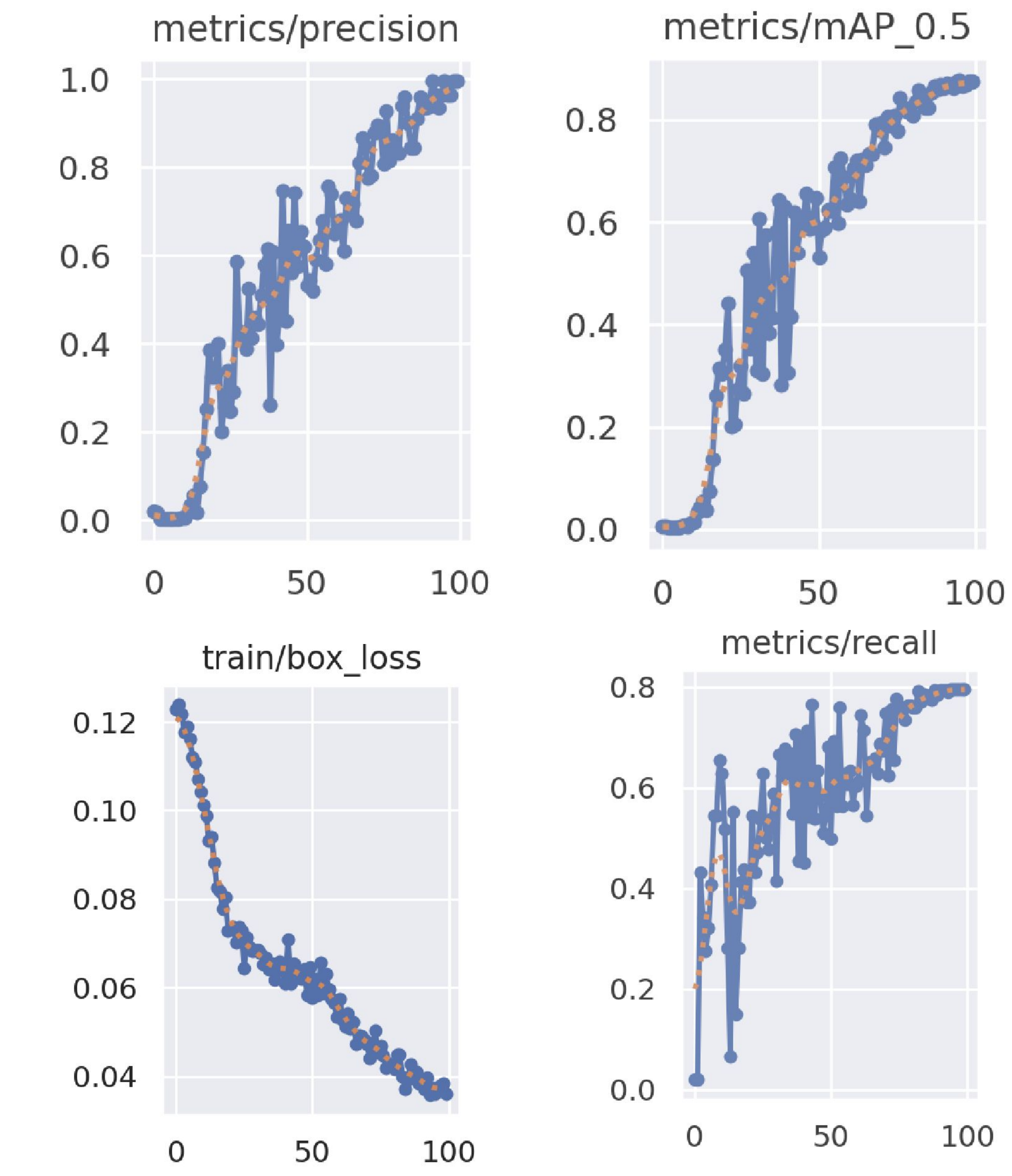

Chart 1 to 4 Evolution of Precision, Accuracy, Recall and Loss during training

Discussion

Challenges & Solutions:

- **Quantization Artifacts:** Initial deployments resulted in "300% confidence" scores and split bounding boxes. This was caused by interpreting INT8 (0-255) outputs directly as floats. We resolved this by implementing a rigorous dequantization function using the model's scale and zero_point parameters.
- **NPU Fallback:** Early tests showed 4-second inference times because the NPU delegated unsupported operations back to the CPU. Converting the model to a fully supported INT8 TFLite format solved this, unlocking the millisecond-level performance.

Implications: The results confirm that edge devices like the i.MX 8M Plus can replace expensive cloud servers for safety monitoring. The 99.5% precision ensures that alerts are trustworthy, while the local processing guarantees GDPR compliance.

Conclusions

EdgeGuard successfully demonstrates that complex Deep Learning models can be deployed on resource-constrained edge devices.

By utilizing NPU acceleration and INT8 quantization, we achieved a **300x speed improvement** over CPU inference without significant loss in accuracy.

The system provides a scalable, private, and real-time solution for industrial safety compliance.

Contact

robert-ionut.rascanu@student.tuiasi.ro
alexandra-petronela.bucataru@student.tuiasi.ro

Gheorghe Asachi Technical University of Iaşi

Faculty of Automatic Control and Computer Engineering

References

1. X. Wang et al., "Convergence of Edge Computing and Deep Learning," IEEE Comm. Surveys & Tutorials, 2020.
2. Z. Zhou et al., "Edge Intelligence: Paving the Last Mile of AI," IEEE, 2019.
3. "Training Machine Learning models at the Edge: A Survey," arXiv:2403.02619v3.
4. MIT News, "Hardware Acceleration in Embedded Systems," 2023.