



Preliminary Comparative Analysis of GPU and NPU Inference Stability for Battery Defect Inspection under Continuous Operation



Aryang Lee · Jihoon Choi · Jiu Jeong · Minsu Choi · Hyungjin Son · Dukyun Nam*

School of Computer Science and Engineering, Kyungpook National University * Corresponding author

ABSTRACT

Battery inspection systems may run 24/7 and share resources with other programs, so both speed and latency consistency can affect real-time inspection. We deploy a COCO-pretrained YOLO11x on an NPU (NXP Ara240) and a GPU (RTX 5080) and run concurrent inference on a shared host for 150 minutes. The GPU was about 5.9× faster (11.37 vs. 66.74 ms) with a lower absolute p99, while the NPU showed a smaller CV (0.81% vs. 6.46%) and p99 excess (1.91 vs. 2.46 ms). Latency did not drift, and detections were identical across cycles. Both absolute latency and consistency should be considered.

01 Introduction

AI-based visual inspection is increasingly used for battery defect detection [1, 2]. Lines run 24/7 while other programs share the system, so latency variation may affect inspection timeliness, and tail latency can matter for real-time requirements [3, 4]. However, prior studies mainly reported accuracy and average speed on a single GPU.

Battery inspection. Kang et al. reported 9.43 ms on an A100 [1]; Xu et al.'s AOI pipeline reached only about 19 FPS [2].

Latency variation. Heat-induced clock changes, shared-resource contention, background processes, and power limits are major causes of tail latency [4].

Dedicated accelerators. Deterministic execution, as in the TPU, has been reported to suit 99th-percentile latency requirements [5].

02 Experiments

Both accelerators shared one host and ran concurrently on the 5,000 COCO val2017 images, fed in a fixed order for 2.5 hours. This setup approximates a resource-sharing environment in which multiple inspection programs may use the same industrial PC. Each device started after 20 warm-up inferences, with synchronized GPU timing.

Metrics. We report the mean, p50–p99.9, maximum, std, and CV (std/mean), since tail latency may affect real-time requirements, and assess drift by comparing the first and last 5 minutes [3, 4].

TABLE 1

Experimental setup

Item	NPU	GPU
Device	NXP Ara240 (M.2, 40 eTOPS)	RTX 5080 (16 GB, 360 W)
Model	YOLO11x INT8 (NXP DVM) [6]	YOLO11x (Ultralytics weights)
Runtime	Ara runtime r3.0	PyTorch
Host	Ryzen 9 9950X, 60 GB RAM, Linux (shared)	Same

Measurement. Both devices used 640×640 inputs at batch 1 with conf 0.25 and NMS IoU 0.7. End-to-end latency was recorded for every frame, and COCO bbox mAP was compared across repetition cycles.

03 Results

3.1 Latency distribution and stability

The GPU averaged 11.37 ms, 5.87× faster than the NPU (66.74 ms), with a lower absolute p99 (13.83 vs. 68.65 ms). The NPU also had a smaller absolute std (0.539 vs. 0.734 ms) and p99 excess (1.91 vs. 2.46 ms), so its low variation is not merely an effect of its larger mean; its advantage is a shorter tail relative to the mean.

No thermal drift. The mean changed only +0.32% (GPU) and +0.04% (NPU) from the first to the last 5 minutes, with no failures. The GPU shoulder near 13 ms (Fig. 1) may thus stem from per-image workload, post-processing, and device execution rather than heat.

TABLE 2

150-minute concurrent run

Metric	RTX 5080	Ara240
Throughput (FPS)	87.86	14.98
Mean latency (ms)	11.37	66.74
p50 / p95 (ms)	11.28 / 12.92	66.68 / 67.67
p99 / p99.9 (ms)	13.83 / 14.50	68.65 / 69.62
Max (ms)	18.99	74.99
Std. dev. (ms)	0.734	0.539
CV (%)	6.46	0.81
p99 – mean (ms)	2.46	1.91

FIG. 1

150-minute latency distribution

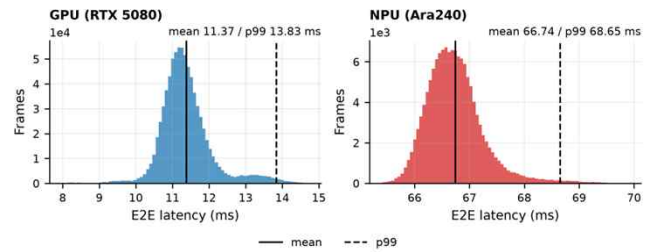


Fig. 1. Solid lines show the mean and dashed lines the p99. The NPU is tightly concentrated, while the GPU has a shoulder near 13 ms.

3.2 Reproducibility run and accuracy

Reproducibility. A separate 40-minute run matched the mean (11.38 / 66.70 ms), p99, and CV. One rare NPU delay of 109.23 ms raised neither the mean nor p99, but worst-case predictability cannot yet be established.

TABLE 3

Detection accuracy (40-minute run)

Metric	RTX 5080	Ara240
mAP50–95 (%)	48.16	46.58
mAP50 (%)	61.60	60.22
mAP75 (%)	52.55	51.19
Per-cycle detections (cycles)	Identical (41)	Identical (7)

Accuracy. The NPU's mAP50–95 was 1.58 points lower on its INT8 path, yet both devices produced identical detections in every cycle: inspection quality did not change over operating time.

04 Concluding Remarks

Consistency matters only once the required time is met: the GPU led in speed and p99, the NPU in a narrower distribution.

FINDING 01 · Speed

The GPU was 5.87× faster, with a lower absolute p99 (13.83 vs. 68.65 ms).

FINDING 02 · Consistency

The NPU had a smaller CV (0.81% vs. 6.46%) and p99 excess (1.91 vs. 2.46 ms).

FINDING 03 · Deadline

At 40 ms [1], every GPU frame met the deadline, but no NPU frame did.

FUTURE WORK

We plan standalone, 24-hour, and same-precision tests, and a defect model on the NPU.

REF References

- [1] D.-J. Kang et al., "Battery pouch cell defect detection system using YOLOv11 lightweight model," J. Electrochem. Sci. Technol., 2026.
- [2] Y. Xu et al., "PBSD-Net: Prismatic battery surface defect detection ...," IEEE Trans. Autom. Sci. Eng., 2025.
- [3] V. J. Reddi et al., "MLPerf inference benchmark," ISCA, 2020.
- [4] J. Dean and L. A. Barroso, "The tail at scale," Commun. ACM, 2013.
- [5] N. P. Jouppi et al., "In-datacenter performance analysis of a tensor processing unit," ISCA, 2017.
- [6] NXP Semiconductors, "Ultralytics YOLO11 detection models for Ara240," Hugging Face, 2026.