



WHITEPAPER 18

Edge AI at the HMI

Integrating Vision and Machine Learning into Embedded Panel PCs

Compute, camera and deployment engineering for low-latency industrial intelligence



A practical framework for accelerators, camera interfaces, model deployment, thermal limits, latency, privacy and HMI use cases such as inspection and operator assistance.



Created by Crystal Display Systems Limited

Technical whitepaper | August 2026

Executive summary

Edge AI can reduce latency, bandwidth and privacy exposure by processing camera and sensor data at the industrial HMI. It can support inspection, predictive maintenance and operator guidance without relying continuously on cloud services.

The engineering risk shifts to model conversion, camera conditions, thermal throttling, data drift and lifecycle support. A robust design treats the model and toolchain as controlled production software and bounds the authority of uncertain outputs.

Core recommendation

Select the accelerator only after benchmarking the deployed model and complete camera-to-action latency at hot steady state. Preserve a reproducible model conversion, signed update and safe fallback route.

How to use this whitepaper

- Use the requirement table to convert edge-AI integration expectations into measurable pass/fail criteria.
- Use the comparison matrix to shortlist architectures without allowing one headline specification to dominate.
- Apply the weighted scorecard using the actual operating environment, duty cycle and commercial lifecycle.
- Validate the preferred approach in the final mechanical, optical, electrical and software implementation.

Contents

1	Define the operational decision before selecting AI compute
2	Edge-AI compute and deployment architectures
3	Comparative decision matrix
4	Engineering trade-offs that change the answer
5	Application-led recommendations
6	Qualification and sourcing checklist
7	Beyond first-generation edge inference
8	Working with Crystal Display Systems

Key decision principles

- Define the operational decision and acceptable error before selecting TOPS or a model.
- Benchmark the complete camera-to-action pipeline, including pre/post-processing and HMI rendering.
- Size sustained accelerator performance at maximum fanless temperature, not short benchmark bursts.
- Control model, runtime, drivers and calibration data as signed production software.
- Design confidence, fallback, monitoring and human override for inevitable model uncertainty and data drift.

1. Define the operational decision before selecting AI compute

A useful AI requirement states sensor conditions, model accuracy/error costs, end-to-end latency, action authority, data retention, power/thermal limit, update path and drift response. These requirements determine whether CPU, NPU, GPU or split compute is appropriate.

Requirement domain	Questions to resolve
Use case and consequence	Classification, detection, anomaly, OCR, pose, predictive maintenance, operator guidance, confidence threshold and action authority.
Sensor pipeline	Camera type, resolution, frame rate, exposure, lighting, lens, trigger, synchronisation, interface and data retention.
Model workload	Architecture, input tensor, precision, memory bandwidth, pre/post-processing, throughput, latency and update frequency.
Compute platform	CPU, integrated NPU, GPU, accelerator module, RAM, storage, thermal design, power and driver/toolchain support.
Deployment	Model conversion, versioning, signing, rollback, calibration, data drift, monitoring and offline/remote updates.
Safety, privacy and security	Human oversight, fail-safe behaviour, personal data, model/IP protection, network exposure and auditability.

A practical weighting model

Assign each criterion a weight from 1 (low importance) to 5 (critical). Score each candidate from 1 (poor fit) to 5 (strong fit), then multiply score by weight. The arithmetic does not replace engineering judgement, but it exposes where the AI compute architecture depends on assumptions, incomplete test evidence or optimistic supplier data.

Do not average away a fatal constraint

A candidate that fails a non-negotiable requirement for edge-AI integration should be removed before weighted scoring. Typical fatal constraints include unsafe operation, an unachievable environmental limit, incompatible interfaces, unacceptable field reliability or no credible supply route.

Recommended process

- Set pass/fail gates for the mandatory edge-AI integration requirements before comparing desirable features.
- Agree criterion weights with electronics, mechanical, software, product, service and procurement stakeholders.
- Score only from controlled data, supplier evidence or representative testing; mark unknowns explicitly.
- Run sensitivity checks by changing the highest weights and identifying whether the preferred architecture changes.
- Document assumptions, test conditions and revision status so the decision can be revisited after a component or firmware change.

2. Edge-AI compute and deployment architectures

2.1 CPU-only inference

Modern x86 and Arm CPUs can run modest models using optimised libraries and vector instructions. CPU-only deployment simplifies hardware and toolchains but can consume compute needed by the HMI and may not meet multi-camera or low-latency requirements.

- Strengths: broad framework support, simple debugging and no separate accelerator dependency.
- Limitations: lower performance per watt and competition with UI, network and control workloads.
- Best fit: low-rate classification, anomaly scoring and proof-of-concept deployments.

2.2 Integrated SoC NPU

Many Arm and newer x86 SoCs integrate an NPU designed for INT8/FP16 inference at low power. Performance depends on supported operators, compiler conversion, memory sharing and runtime maturity rather than headline TOPS.

- Strengths: high efficiency, compact fanless integration and low data-transfer overhead.
- Limitations: vendor-specific toolchains, unsupported model operators and uncertain long-term SDK/BSP support.
- Best fit: dedicated panel PCs with well-defined models and sustained low-power inference.

Toolchain ownership

Archive model source, conversion scripts, calibration data, compiler/runtime versions and validation outputs. A deployed binary without a reproducible conversion path is not maintainable.

2.3 Discrete GPU

A discrete GPU provides strong throughput, mature vision libraries and flexibility for larger models or multiple streams. It increases power, thermal density, cost and driver complexity and may be difficult to integrate fanlessly at the panel.

- Strengths: high performance, broad framework ecosystem and strong parallel pre/post-processing.
- Limitations: thermal/power burden, larger enclosure and platform-driver lifecycle dependencies.
- Best fit: high-end inspection, multi-camera analytics and development-intensive applications.

2.4 Dedicated accelerator module

M.2, PCIe or USB accelerators can add inference to an existing panel PC. They preserve the host architecture but add interface bandwidth, module thermals, runtime dependencies and another lifecycle item.

2.5 Split edge architecture

A display HMI can connect to a separate AI box or camera controller. This moves heat and upgrade risk away from the panel but adds network latency, synchronisation, cybersecurity and two failure domains.

3. Comparative decision matrix

Criterion	CPU only	Integrated NPU	Discrete GPU	Accelerator / split
Low-power efficiency	Limited	Excellent	Limited	Strong
Peak throughput	Limited	Strong	Excellent	Strong
Model flexibility	Excellent	Moderate	Excellent	Strong
Fanless integration	Excellent	Excellent	Limited	Strong
Toolchain portability	Excellent	Limited	Strong	Moderate
Multi-camera scaling	Limited	Strong	Excellent	Excellent
Upgrade path	Moderate	Moderate	Strong	Excellent
Development simplicity	Excellent	Moderate	Moderate	Moderate
Lifecycle dependency	Excellent	Moderate	Moderate	Moderate
Relative system cost	Excellent	Strong	Limited	Moderate

Indicative only. Actual performance depends on the complete system architecture, supplier implementation, environmental conditions and validation method.

Example weighted scorecard

Criterion	Weight	CPU only	Integrated NPU	Discrete GPU
Two 1080p vision streams	5	2	4	5
30 ms end-to-end latency	5	2	5	5
25 W fanless budget	5	5	5	2
Model update flexibility	4	5	3	5
Toolchain maturity	4	5	3	5
Ten-year support	4	4	3	4
System cost	3	5	4	2
Weighted total		116	118	122

For this two-camera fanless HMI, an integrated NPU leads because latency and power dominate. A discrete GPU provides maximum model flexibility but exceeds the thermal budget; CPU-only is simple but cannot sustain the throughput.

Reading the result

- CPU-only remains appropriate for low-rate or low-consequence analytics and as a functional fallback.
- Integrated NPU selection must be proven by converted model accuracy and pipeline latency, not TOPS alone.
- Discrete GPU should be used where model flexibility and multi-stream throughput justify the enclosure and lifecycle cost.

4. Engineering trade-offs that change the answer

4.1 TOPS is not application throughput

Operator support, memory bandwidth, tensor shape, quantisation and pre/post-processing determine frames per second and latency. Benchmark the exact model and runtime.

4.2 Quantisation can change model decisions

INT8 or mixed precision improves performance but may alter accuracy, confidence and edge cases. Validate against a controlled dataset after conversion.

4.3 Camera timing and lighting often dominate accuracy

Exposure, motion blur, lens contamination, trigger jitter and illumination change model input more than accelerator choice. Engineer the optical acquisition system.

4.4 Thermal throttling creates variable latency

An accelerator may meet targets when cold and miss them after hot soak. Log clocks, power and end-to-end latency at sustained worst load.

4.5 Data drift is a lifecycle mechanism

Products, lighting, operators and processes change. Monitor input/score distributions, retain representative review data and define retraining/revalidation triggers.

4.6 AI output must have bounded authority

Use confidence thresholds, plausibility checks, human confirmation or safe fallback where false positives/negatives have operational consequence.

5. Application-led recommendations

Visual quality inspection

Use controlled lighting/triggering, integrated NPU or GPU and a reject mechanism with confidence review and sample traceability.

Operator assistance

Overlay instructions or warnings locally for low latency. Keep the operator in control and avoid obscuring primary machine status.

Predictive maintenance HMI

Combine sensor trends with local inference and explainable diagnostics. Preserve raw/summary data needed to investigate false alarms.

Privacy-sensitive people analytics

Process locally, minimise retained images and define access/retention. Evaluate bias and consent requirements for the use case.

Remote AI appliance with local panel

Split compute when heat or upgrade cadence is incompatible with the display head. Secure and monitor the network dependency.

6. Qualification and sourcing checklist

	Area	Verification action
☐	Data / accuracy	Define representative train/validation/field sets, classes, confidence, false-positive/negative and edge cases.
☐	Pipeline	Measure sensor trigger, capture, preprocessing, inference, postprocessing, HMI update and actuator latency.
☐	Performance	Benchmark sustained throughput, memory, CPU/GPU/NPU utilisation and thermal throttling.
☐	Conversion	Compare source and deployed model outputs after quantisation/compiler optimisation.
☐	Robustness	Test lighting, blur, occlusion, contamination, camera loss, corrupt frames and out-of-distribution inputs.
☐	Deployment	Verify signed model/runtime updates, version identity, rollback and power-failure recovery.
☐	Security / privacy	Protect model, credentials and data; validate network segmentation and retention/access controls.
☐	Lifecycle	Define drift monitoring, revalidation, toolchain/BSP support and accelerator replacement strategy.

Questions to ask a display partner

- What end-to-end operational metric replaces the headline TOPS figure?
- Which model operators, precision and compiler/runtime versions are supported?
- How much accuracy changes after quantisation and deployment conversion?

- What sustained latency and temperature were measured in the final fanless enclosure?
- How are model version, calibration data, update signing, rollback and audit records controlled?
- What happens when confidence is low, the camera fails or the input is outside the trained domain?

Minimum evidence pack for design approval

- Use-case requirements with accuracy, latency, authority and safe-fallback definition.
- Controlled camera/lighting and sensor-interface specification.
- Model source/version, dataset lineage, conversion scripts and deployed runtime manifest.
- Accuracy, robustness and sustained thermal-performance test reports.
- Signed deployment/update/rollback package and device/model identity logs.
- Drift monitoring, revalidation, privacy/security and accelerator lifecycle plan.

7. Beyond first-generation edge inference

Edge AI systems are moving toward heterogeneous accelerators, on-device adaptation and stronger model observability. Adoption should improve a measured operational outcome while preserving reproducibility and safety.

Transformer and multimodal models

Larger models can combine image, text and sensor context, but memory, latency and validation scope grow rapidly.

On-device learning or adaptation

Local adaptation may respond to process drift but complicates validation, traceability, cybersecurity and rollback.

Federated learning

Fleet devices can contribute model updates without centralising raw data. Communication, poisoning resistance and version control are significant.

Event-based and smart cameras

Processing in or near the sensor can reduce bandwidth and latency. Vendor-specific pipelines and calibration/lifecycle need control.

Explainability and confidence monitoring

Visual explanations or uncertainty estimates can aid operators and service, but must be validated for reliability rather than treated as guaranteed causality.

Adoption gate

A new AI technique should enter production only when it improves a defined operational metric, remains reproducible from controlled source and data, and has bounded failure, update and monitoring behaviour.

8. Working with Crystal Display Systems

Crystal Display Systems Limited supplies industrial panel PCs and embedded AI platforms with Arm/x86 processors, integrated NPUs, GPU or accelerator options, camera and I/O integration, custom thermal design and rugged displays.

Relevant CDS resources

Resource	Purpose
<u>AI-Driven Embedded Innovation</u>	Embedded AI accelerators and edge-processing solutions.
<u>Industrial Panel PC Solutions</u>	Integrated industrial HMI and panel PC solutions with customisation options.
<u>Industrial Panel PC Range</u>	Current panel PC platforms, processors, I/O and rugged configurations.
<u>Industrial Boxed PCs and Panel PCs</u>	Industrial computing platforms, customisation and long-life support.
<u>Industrial Automation Displays</u>	Display, monitor and computing solutions for automation and machine HMI.
<u>CDS Knowledge Centre</u>	Display guides and technical training documents.
<u>Request a Quote</u>	Submit project requirements for technical and commercial review.
<u>Contact CDS</u>	Contact details and enquiry routes.

Contact Crystal Display Systems Limited

Crystal Display Systems Limited

Address: Unit 6 M2M Park, Fort Bridgewood, Maidstone Road, Rochester, Kent, ME1 3DQ, United Kingdom

Telephone: +44 (0)1634 791 600

Email: info@crystal-display.com

www.crystal-display.com

Conclusion

Edge AI at the HMI succeeds when the complete sensing-to-action pipeline is engineered and maintainable. Accelerator choice follows model accuracy, latency, power, toolchain and lifecycle, not TOPS alone. The production system controls data, conversion, model identity, thermal performance, confidence, fallback and updates as rigorously as conventional embedded software.

Final engineering rules

- Define operational accuracy, latency and authority before choosing compute.
- Benchmark the exact camera, model, conversion and HMI pipeline at hot steady state.
- Control model source, dataset, toolchain, runtime and calibration as released software.
- Design low-confidence, sensor-fault and out-of-domain fallback behaviour.
- Monitor drift and maintain a secure revalidation/update path over the product life.

Disclaimer

This whitepaper provides general engineering guidance and does not replace application-specific design review, supplier specifications, regulatory assessment or formal product qualification. Performance depends on the exact components, implementation, environment and usage profile.