

Zero-Shot Visual Anomaly Detection on the Artificial Cognition Substrate

The first real run of VAAS-X's vision encoder -- previously an unexecuted wrapper around a third-party model -- including an honest benchmark result and a real round trip through the live Artificial Cognition ingest/retrieval endpoints on production.

3

REAL INDUSTRIAL
DOMAINS TESTED

0.83

MEAN AUROC, ZERO
TRAINING EXAMPLES

2

DATASETS CHECKED
AND REJECTED ON
LICENCE TERMS FIRST

0

GPU REQUIRED

The problem

Industrial visual inspection -- spotting a defective part on a line, a damaged component, a print error -- is traditionally either a fixed rule that breaks the moment the product changes, or a supervised model trained per product line on thousands of labelled defect images that often doesn't exist yet for a new line. VAAS-X's Artificial Cognition substrate asks a different question: can an embedding, with no training at all, place normal images close together and a real defect far enough away to be caught by distance alone -- the same operational-memory pattern already proven for SCADA telemetry and wearable IMU data, applied to images.

What's actually running

Encoder: wraps torchvision's ViT-B/16, pretrained on ImageNet, classification head removed. **No training happened for this result** -- a deliberate first step, not a shortcut. **Composite layer:** the real P9 composite-embedding code projects the 768-dim ViT output into a 256-dim per-modality space, then fuses it into a 1024-dim vector -- the same mechanism that lets any modality land in one shared geometry for cross-modal retrieval. **Scoring:** nearest-neighbor cosine distance to a reference set of normal-only images -- the standard zero-shot anomaly detection pattern, and the direct image analogue of the frozen-baseline approach already running in production for the SCADA detector.

The benchmark: a real, commercially-licensed dataset

Dataset: VisA (Amazon, "SPot-the-Difference"), 12 real object classes, 10,821 images (9,621 normal, 1,200 real labelled anomalies). **Licence:** **CC BY 4.0**, verified directly from Amazon's own GitHub repository's licence file -- not inferred from a mirror. Two other candidate datasets were checked at their real source and rejected before any benchmark ran: MVTec AD (CC BY-NC-SA, non-commercial) and a popular Kaggle casting-defect set (CC BY-NC-ND at its actual source, despite a third-party mirror wrongly labelling it CC BY 4.0).

Protocol: VisA's official one-class split -- train the reference set on normal images only, test on a held-out mix of real normal and real anomalous images, score by distance, compute AUROC against the dataset's own labels. No threshold tuning on the test set.

Category	Domain	Test set	AUROC
pcb1	Electronics	100 normal + 100 anomaly	0.909
macaroni1	Food	100 normal + 100 anomaly	0.823
capsules	Pharmaceutical	60 normal + 100 anomaly	0.752
Mean			0.828

Real production round trip

The benchmark above ran entirely locally. Separately, the full intended architecture was exercised against live production, over the network, for the first time: encode a real VisA image locally, project and fuse it, then `POST /ac/ingest` against the live resolver using a disposable Professional-tier test key (the `ac` capability is gated at Professional tier and above -- confirmed, not assumed). One normal and one anomalous image were ingested, then queried with a second, held-out normal image.

Result: the query correctly ranked the other normal image more similar (cosine score 0.957) than the anomalous image (0.838) -- the right answer, from a real server round trip. No model or dependency was installed on the production server -- the encoder runs entirely client-side; production only stores and indexes the embedding it's sent.

Live camera support (RTSP), verified against real hardware

A separate local MCP server adds RTSP ingestion on top of the same encoder -- a background watcher samples a live camera, decodes and encodes every frame locally, and sends only the resulting embedding onward. Verified against a real residential camera, not a simulated feed: the watcher connected over RTSP/TCP, sampled 5 frames, encoded and ingested 3 of them against live production (three real `/ac/ingest` calls, each 200 OK), then auto-stopped cleanly at its configured limit -- zero reconnects, zero errors, camera credentials never appeared in any output. Not yet verified against real hardware: live anomaly alerting when a real defect appears on a real feed, multi-camera concurrency, and reconnect behaviour under an actual camera dropout.

Scope, honestly

What's proven: a frozen, off-the-shelf image embedding -- zero custom training -- already produces a real, usable anomaly signal across three different industrial domains on a real, commercially-licensed benchmark, and the full client-encode/server-ingest/server-query pipeline works end to end against live production, correctly gated behind the Professional tier.

What this does not yet show: only 3 of VisA's 12 categories were tested; whether custom training would do meaningfully better than zero-shot is untested; this is a public benchmark, not a specific customer's real images; audio, depth, and thermal encoders remain untouched; and no live camera or physical deployment has been tested.

Deployment model

Identical to every other vertical: ingestion and retrieval run against the hosted resolver behind a single API key, with no local GPU or vector database to operate. The vision model runs wherever the customer's device or server is -- never on VAAS-X's own infrastructure.

Get started: The `ac` capability requires the Professional tier -- `pip install vaas-x[ac]` and see the Artificial Cognition integration guide for the full `ArtificialCognition.encode()/.ingest()/.retrieve()` path.
Full guide: vaasx.com/docs/getting-started.html | Enterprise: hello@vaasx.com