Nick Clark · Published July 2, 2026 · PDF

How to Detect AI-Generated Images Without a Watermark

If you moderate uploads, curate a training corpus, or gate a publishing pipeline, you eventually need to tell synthetic images from captured ones when there is no watermark, no C2PA manifest, and no cooperating generator. This guide walks through an architectural approach to that problem: derive a structural identity from the pixels themselves and reason about lineage, recapture, and output-distribution proximity. The approach is disclosed in PCT International Application No. PCT/US26/28630, and the home inventive step is the Content Anchoring inventive step. It is an architecture you build, not a library you install.

What You Are Building

You have an image and a yes-or-no question: was this produced by a generative model, or captured from the world? The easy version of that question assumes the image carries a signal you can read: an invisible watermark from the generator, a signed provenance manifest, or embedded metadata. In practice the images that matter most arrive stripped of exactly those signals. A screenshot has no manifest. A re-encoded, cropped, or re-uploaded file has lost its metadata. A generator that does not cooperate embeds nothing at all.

This guide describes an architecture for making that determination without depending on any embedded or cooperative signal. Everything the classifier reasons about is derived, post hoc, from the artifact itself. You are building three complementary detectors, plus a way to combine their outputs into a single governance signal you can act on. The intended reader is a developer building an upload gate, a corpus-admission pipeline, or a pre-release check inside a generation service. You will come away understanding the design and the tradeoffs; you will not come away with a finished binary.

Why the Obvious Approaches Fall Short

Watermark reading works only when the generator embedded a watermark and the watermark survived the trip. Watermarking approaches embed an identity signal in the content stream or a sidecar record. Both are fragile: as the filed disclosure notes, watermarks are removable through transcoding, cropping, or generative reconstruction, and metadata records are decoupled from content structure and require persistent external storage. If your adversary controls the file, assume the mark is gone.

Provenance manifests such as signed capture credentials are accurate and useful when present, but they are opt-in and cooperative. An image with no manifest is not thereby proven synthetic; absence of a signal is not a signal. You still need a way to reason about the bare pixels.

A trained "AI detector" classifier is a legitimate tool, but it couples you to a specific model of a specific generation of generators. Each time a new architecture ships with a different output signature, the classifier drifts, and you retrain and redeploy inference infrastructure to keep up. That is a real operational cost, and it is the gap the architecture below is designed around: it separates the stable structural feature extraction from the part that changes as generators evolve.

The Architecture

The foundation is a structural identity for the image, computed from its internal variance rather than its filename, hash pointer, or metadata. The disclosure calls this a multi-axis variance vector.

The variance vector. The image is first converted to a normalized grayscale scalar field (the disclosure uses perceptual luma weights of about 0.299 red, 0.587 green, 0.114 blue). A multi-scale variance flow analyzer then subdivides that field into three nested grids (8x8, 16x16, and 32x32 cells) and computes per-cell intensity variance at each scale. From those statistics the pipeline builds a nine-dimensional vector across three axes:

- The **X axis** encodes energy distribution across scale: the slope of mean variance from coarse to fine, its curvature at the medium scale, and the asymptotic fine-scale energy value.
- The **Y axis** encodes frequency-compaction behavior: how the variance standard deviation changes across scales, the spread between per-scale extremes, and how closely multi-scale behavior tracks the global variance.
- The **Z axis** encodes structural phase and orientation, derived from a gradient-magnitude histogram over eight angular bins from 0 to pi radians, canonicalized so the dominant bin sits at index zero. Its components include a horizontal-vertical orientation bias and a diagonal-axial bias, plus a stability coefficient. This axis matters most below.

Two properties make this useful for detection. First, cosine similarity between two vectors is directly computable as their inner product divided by the product of their magnitudes, so "how structurally close are these two images" is a cheap arithmetic operation, no decoding step. Second, the representation is designed to stay stable under format conversion, rescaling within a canonical size, and moderate lossy compression, while shifting predictably when the content itself changes. That stability is what lets you compare a candidate against a large corpus meaningfully.

On top of these signals the architecture runs three detectors, described in the disclosure's adversarial-robustness section.

1. Lineage absence (the orphan detector). A generatively synthesized artifact has no structural lineage to any prior registered artifact. Its variance-vector position reflects the statistical properties of the generator's output distribution, not the profile of any specific real prior image. The lineage query module queries the anchor network for registered parent UIDs within a configured slope-continuity radius of the candidate. If none falls inside that radius, the orphan detector classifies the artifact as structurally unanchored: it has no provable lineage to anything in the governed corpus. Important honesty note from the disclosure itself: unanchored does not mean fraudulent. It means the image cannot be admitted under a policy that requires verifiable provenance, and it warrants heightened scrutiny under a synthetic-content policy.

2. Screenshot and recapture detection (Z-axis gradient). When a display renders an image and a camera or capture device re-captures it, screen rendering introduces a periodic spatial-frequency structure in luminance, from sub-pixel geometry, display-pipeline dithering and compression, and the capturing optics. Per the disclosure, these artifacts show up in the Z-axis gradient histogram as elevated energy in the horizontal and vertical orientation bins relative to the diagonal bins, systematically raising the horizontal-vertical bias score compared to the original. The recapture classifier compares that Z-axis bias against a policy-calibrated threshold and emits a recapture probability. The notable property: this needs no reference image and no corpus lookup. It runs entirely off the candidate's own structure.

3. Synthesis probability (output-distribution proximity). The synthetic content detector compares the candidate's variance vector against a generative-model output distribution, represented as a slope-band-indexed statistical model of the variance profiles of known synthetic content. When the candidate falls inside the high-probability region of the synthetic distribution and outside the authentic distribution for that content category, it emits an elevated synthesis probability. The distribution is

built empirically from samples of generator outputs in a governed reference corpus and can be updated as new architectures appear, which is what lets the system track evolving generators without retraining an inference model or shipping new classifier infrastructure.

The aggregator. A composite risk score aggregator combines lineage absence, recapture probability, and synthesis probability into a single governance signal. That signal routes to a pre-release admissibility decision: admit, reject, regenerate, or escalate, evaluated against a versioned, cryptographically signed policy object that declares the thresholds and exclusion classes.

How to Approach the Build

You implement this yourself. The steps below are the order the disclosure's pipeline implies.

1. **Build the normalizer.** Convert to grayscale with luma weights, rescale to a canonical square (the disclosure uses a 256x256 letterbox with smoothing disabled so anti-aliasing does not inject false variance), and canonicalize orientation from the dominant gradient bin. Get this deterministic first; every downstream signal depends on it.
2. **Implement the nine-dimensional extractor.** Compute per-cell variance at the 8x8, 16x16, and 32x32 grids, then derive the X, Y, and Z components as described above. Verify your cosine similarity by feeding it the same image twice (expect ~1.0) and an unrelated image (expect much lower).
3. **Wire up the recapture detector first.** It is the cheapest win because it needs no corpus. An illustrative interface, faithful to the disclosed method and clearly not a shipping API:

```
# illustrative only
recapture_prob(image):
    z = extract_variance_vector(image).z_axis
    hv_bias = z.horizontal_vertical_bias   # elevated by screen re-capture
    return score_against(hv_bias, policy.recapture_threshold)
```

Calibrate the threshold on your own paired originals-versus-recaptures; the disclosure specifies the signal and that the threshold is policy-calibrated, not a fixed number.

4. **Stand up the corpus and lineage query.** To detect orphans you need registered content to be an orphan *relative to*. Register your known-authentic corpus as UIDs, then implement the lineage query as a slope-continuity-radius search for parent UIDs. No parent within the radius means unanchored.
5. **Fit the synthesis distribution.** Collect generator outputs per content category, index their variance vectors by slope band, and model the synthetic and authentic regions. Plan to refresh this as new generators appear; the architecture is designed to absorb that by updating the distribution rather than retraining a classifier.
6. **Combine and gate on policy.** Aggregate the three scores, then evaluate against a signed policy object that declares thresholds, exclusion classes, and escalation paths. Keep the decision reproducible: the disclosure emphasizes that admissibility decisions should be re-derivable from the versioned policy object and the logged inputs.

An attractive deployment property: the disclosure describes a client-side variant where normalization, variance extraction, and UID hashing run in a standard browser via the Canvas 2D API, so the raw image never leaves the device and only the computed UID and decision are transmitted. If data minimization matters to you, evaluate at upload time on the client.

What This Does Not Give You

This is an architecture, not a drop-in library. There is no package to install and no endpoint to call; you build the extractor, the corpus, the distribution model, and the policy layer yourself. Nothing here is benchmarked or productized in this document, and no accuracy or false-positive rate is claimed, because the filed disclosure states the mechanisms, not performance numbers. Do not represent it as a shipping detector.

Be honest about the signals' limits. Lineage absence is not proof of synthesis; it is absence of provenance, and a genuinely novel real photograph that was never registered will also read as unanchored. Recapture detection identifies a display-recapture signature, which overlaps with but is not identical to "AI-generated": a photographed AI image and a photographed real print can both trip it. The synthesis detector is only as current as the distribution you fit, and covers the generator families you sampled. The strength of the design is in combining all three under an explicit policy, not in trusting any one of them alone. Two detectors depend on a governed corpus you must build and keep current; only the recapture detector works standalone.

Disclosure Scope

The approach described here is disclosed in PCT International Application No. PCT/US26/28630, whose home inventive step is the Content Anchoring inventive step: deriving a structural, variance-based content identity from an artifact itself, with no embedded watermark, no enrollment, and no central registry, and reasoning about lineage, recapture, and output-distribution proximity from that identity. This guide is educational. It explains the disclosed architecture so a skilled developer can build an implementation; it is not a warranty, a benchmark, a product, or an offer of software, and it grants no license to the disclosed subject matter.

Content Anchoring (/content-anchoring)

All 40 steps → (/inventive-steps)

Computable identity for media. Provenance from structural variance.

PCT/US26/28630 (/patents/pct-us26-28630)

PRIMARY TECHNICAL DISCLOSURE

- Content Anchoring: Computable Identity for Media That Changes (/articles/content-anchoring-computable-identity-for-media-that-changes)

SECONDARY TECHNICAL

- Multi-Axis Variance Vector Extraction: Nine Dimensions of Structural Content Identity (/articles/content-anchoring/variance-vector)
- Quadrant Decomposition: Spatial Sub-Region Fingerprinting for Partial Similarity Detection (/articles/content-anchoring/quadrant-decomposition)
- 320-Bit UID Construction: Multi-Segment Hashing for Negligible Collision Probability (/articles/content-anchoring/uid-construction)
- Structure Signature: Background-Invariant Matching Through Gradient-Only Descriptors (/articles/content-anchoring/structure-signature)
- Constellation Signature: Geometry-Invariant Matching Across Crop, Scale, and Occlusion (/articles/content-anchoring/constellation-signature)
- Five-Band Variance Classification: Content Routing by Structural Complexity (/articles/content-anchoring/variance-classification)
- Variance Saturation-Governed Cache Eviction: UID Density Replacing Static TTL (/articles/content-anchoring/cache-eviction)
- Multi-Root Composite Lineage Graphs: Provenance Through Variance Vector Similarity (/articles/content-anchoring/composite-lineage)
- Multi-Modal Content Identity: Unified Pipeline Across Image, Audio, Text, and Video (/articles/content-anchoring/multi-modal-identity)
- Rights-Grade Pre-Release Admissibility: Policy Evaluation Before Content Commitment (/articles/content-anchoring/pre-release-admissibility)
- Training Corpus Governance: Verifiable Lineage From Training Data to Model (/articles/content-anchoring/training-corpus-governance)
- Consultation Event Logging: Deterministic Records of Every Generation Reference (/articles/content-anchoring/consultation-logging)

- Model Output Provenance Fingerprint: Structural Proximity Without Model Access (/articles/content-anchoring/output-provenance)
- Creator Attribution and Compensation Routing: Payment From Consultation Lineage (/articles/content-anchoring/creator-attribution)
- Adversarial Robustness and Deepfake Detection: Content Identity as Detection Substrate (/articles/content-anchoring/adversarial-robustness)
- Client-Side Execution Architecture: Privacy-Preserving Variance Computation on Device (/articles/content-anchoring/client-side-execution)
- UID Resolution Query Protocol: Distributed Lookup Across Anchor Node Networks (/articles/content-anchoring/uid-resolution)
- Orientation Canonicalization: Rotation-Invariant Processing Through Gradient Normalization (/articles/content-anchoring/orientation-canonicalization)
- Cross-Band Resolution Pathfinding: Traversal Between Variance Bands Under Mutation (/articles/content-anchoring/cross-band-resolution)
- Identity by Position: Media as a Third Navigable Space (/articles/content-anchoring/identity-by-position)

APPLICATIONS · GENERAL

- Forbidden-Content Blocking at Upload and Generation Time: Pre-Release Exclusion Against Signed Policy (/articles/content-anchoring/forbidden-content-blocking)
- Structural Provenance for Software Supply Chains: Binary and Firmware Identity Independent of SBOM Metadata (/articles/content-anchoring/software-supply-chain-provenance)
- Rights-Grade Generative AI: How to Pay Creators, Exclude Forbidden Content, and Prevent Infringement Before Release (/articles/content-anchoring/rights-grade-generative-ai)
- Deepfake Detection by Structural Provenance: Verifying Synthetic Media Without Watermarks (/articles/content-anchoring/deepfake-provenance)
- Creator Economy Attribution Without Platform Intermediaries (/articles/content-anchoring/creator-attribution-economy)
- Verifying Source Photos and Video in the Newsroom: Content Anchoring for Journalism (/articles/content-anchoring/journalism-verification)
- Detecting Image Manipulation and Proving Figure Provenance in Research Publications (/articles/content-anchoring/academic-research-integrity)
- Content Anchoring for Legal Evidence Chains (/articles/content-anchoring/legal-evidence-chain)
- Content Anchoring for Insurance Claims Evidence (/articles/content-anchoring/insurance-claims-evidence)
- Content Anchoring for Real Estate Documentation (/articles/content-anchoring/real-estate-documentation)

- Art Authentication and Provenance Verification with Content Anchoring (/articles/content-anchoring/art-authentication)
- Detecting Screenshot and Recapture Fraud in Identity-Document KYC With Structural Content Identity (/articles/content-anchoring/identity-document-kyc-recapture)

APPLICATIONS · SPECIFIC

- C2PA vs Content Anchoring: Attached Provenance or Content-Intrinsic Identity? (/articles/content-anchoring/c2pa)
- Google SynthID Alternative: Content-Intrinsic Identity Beyond Watermarking (/articles/content-anchoring/google-synthid)
- Beyond Shutterstock: Content-Intrinsic Identity That Survives Re-Encoding and Cropping (/articles/content-anchoring/shutterstock)
- Spotify Alternative for Music Provenance: Structural Content Identity Beyond the ISRC Database (/articles/content-anchoring/spotify)
- Getty Images Alternative for Provenance: Structural Content Identity Beyond Metadata (/articles/content-anchoring/getty-images)
- Adobe Stock vs Structural Content Identity: Licensing Records Are Not Content Identity (/articles/content-anchoring/adobe-stock)
- YouTube Content ID vs Content Anchoring: Matching Against a Database, or Identity in the Content Itself (/articles/content-anchoring/youtube-content-id)
- Audible Magic Alternative: Structural Content Identity Beyond Database-Matched Fingerprinting (/articles/content-anchoring/audible-magic)
- Digimarc vs Structural Content Identity: Watermarks Are Added, Not Intrinsic (/articles/content-anchoring/digimarc)
- Irdeto vs Structural Content Identity: DRM Protects the Channel, Not the Payload (/articles/content-anchoring/irdeto)
- Truepic alternative: capture-time provenance versus structural identity derived from the artifact itself (/articles/content-anchoring/truepic)
- Microsoft PhotoDNA vs structural content identity: hash-matching known images versus screening artifacts before release (/articles/content-anchoring/microsoft-photodna)
- Pex alternative: structural content identity vs enrolled fingerprint matching (/articles/content-anchoring/pex)

Content Anchoring overview → (/content-anchoring)