Nick Clark · Published July 2, 2026 · PDF

How to Detect a Screenshotted or Re-Photographed Document

If you accept ID cards, invoices, or signed forms, some of what arrives is not an original file but a photo of a screen or a screenshot of another screenshot, and you need to catch that before you trust it. This guide describes an architectural approach to detecting recaptured images from the artifact itself, with no watermark, no enrollment, and no reference lookup. The approach is disclosed in PCT International Application No. PCT/US26/28630 and belongs to the Content Anchoring inventive step; it is an architecture you build, not a library you install.

What You Are Building

You are building a check that decides, from an image alone, whether the image is likely a fresh digital artifact or a recapture: a photo of a document shown on a screen, or a screenshot of a screenshot. This is the problem behind a very common search: someone submits a KYC document, an insurance claim photo, a signed PDF page, or a product image, and you want to know whether they photographed a monitor instead of sending the real file.

The audience is anyone running an upload or intake flow where recapture is a fraud or integrity signal: identity verification, claims processing, marketplace listing review, content provenance, and generative pipelines that must not admit laundered inputs.

The goal is a per-image recapture probability score you can threshold and route on, computed without asking the sender for anything extra and without comparing against a master copy you may not have.

This guide teaches the architecture from a filed patent disclosure. It is not a downloadable SDK, and no drop-in package is implied. You implement it.

Why the Obvious Approaches Fall Short

The first instinct is metadata. EXIF fields, capture timestamps, and device tags can hint that an image came from a camera. They are informative when present, but they are trivially stripped, rewritten, or absent after a screenshot, and most upload pipelines re-encode images and discard them anyway. Metadata tells you about the container, not about the pixels.

The second instinct is a watermark or a signed original. If you control the source, you can embed a signal and check for it later. That works inside a closed loop, but it does nothing for content that arrives from outside your system, and embedded marks can be removed by transcoding, cropping, or regeneration. This is a real and useful family of techniques; it simply does not cover the case where you never touched the original.

The third instinct is a trained classifier: collect recaptured and clean samples, fit a model, and score new images. This can work, but it couples you to a labeled dataset and a model you have to host, monitor, and retrain as screens and phone cameras change. It also gives you an opaque number that is hard to audit when a decision is challenged.

The structural gap is this: recapture leaves a physical trace in the image itself, in how edges are oriented, and that trace can be read directly as a geometric property rather than inferred by a black box or looked up against a reference you do not have.

The Architecture

The disclosed approach treats every artifact as a normalized scalar field and extracts a multi-axis variance vector from its internal structure. Three axes are defined: an X axis encoding cross-scale energy distribution, a Y axis encoding cross-scale frequency compaction, and a Z axis encoding structural phase persistence based on gradient orientation. Recapture detection lives on the Z axis.

The spec describes why recapture is detectable. When a display renders an image and a camera or screen-capture device recaptures the rendered output, the recapture introduces a characteristic variance signature in the luminance channel. Per the disclosure, this comes from the periodic spatial frequency structure of the display's sub-pixel geometry, from compression and dithering artifacts in the display pipeline, and from the optical point-spread function of the capturing lens or sensor. Screens and their pixel grids are strongly aligned to horizontal and vertical axes, so these artifacts show up as extra energy along horizontal and vertical edge orientations.

The Z axis is where that shows up. The disclosed extraction computes a gradient histogram over eight angular bins spanning zero to pi radians across the interior pixels of the normalized scalar field, then canonicalizes the histogram by rotating it so the dominant bin sits at index zero. From this histogram it derives a horizontal-vertical orientation bias, defined as the mean weight of horizontal bins minus the mean weight of vertical bins, and a diagonal-axial bias, defined as the mean weight of diagonal bins minus the mean weight of axial bins. According to the spec, recaptured artifacts show elevated energy in the horizontal and vertical orientation bins relative to the diagonal bins, producing a horizontal-vertical bias score that is systematically elevated compared to the original digital artifact.

The screenshot recapture classifier reads exactly this signal. It evaluates the Z-axis horizontal-vertical bias against a policy-calibrated threshold and produces a recapture probability score. The disclosure states plainly that this method requires no reference

to the original artifact and operates entirely from the structural features of the candidate artifact itself, enabling recapture detection without corpus lookup. That is the property that makes it usable on inbound content you have never seen before.

The disclosure situates recapture detection alongside two adjacent structural signals that share the same variance vector. A lineage query module asks the anchor network whether any registered parent artifact falls within a slope continuity radius of the candidate; if none does, an orphan detector flags the artifact as structurally unanchored, meaning it has no provable connection to registered content. A synthetic content detector compares the candidate variance vector against a slope-band-indexed statistical model of known generative outputs and produces a synthesis probability. A composite risk score aggregator then combines lineage absence, recapture probability, and synthesis probability into a single governance signal that routes into a pre-release admissibility engine, which decides admit, reject, regenerate, or escalate against a signed policy object. Recapture is one input to that decision, not the whole of it.

How to Approach the Build

The following is an ordered plan faithful to the disclosed pipeline. The sketches below are illustrative and describe the disclosed steps; they are not a working library.

1. Normalize the artifact to a scalar field. Convert the image to grayscale using a perceptual luminance weighting (the spec uses roughly 0.299 red, 0.587 green, 0.114 blue) with values in the range zero to one. Rescale to a canonical square canvas (the spec uses 256 by 256) by scaling the longest edge and letterboxing on a black fill, with image smoothing disabled so rescaling does not inject artificial edge variance. The disclosure notes this can run client-side using only the standard Canvas 2D API, so the raw file need never leave the device.
2. Build the gradient orientation histogram. Over the interior pixels, compute gradient magnitude and orientation, and accumulate magnitude into eight angular bins spanning zero to pi. Rotate the histogram so the dominant bin is at index zero to

make the representation rotation-invariant.

```
# illustrative, per the disclosure
hist       = gradient_histogram(field, bins=8, range=[0, pi])
hist       = rotate_to_dominant(hist)          # dominant bin -> index 0
hv_bias    = mean(horizontal_bins) - mean(vertical_bins)
diag_bias  = mean(diagonal_bins)  - mean(axial_bins)
```

3. Derive the Z-axis biases. Compute the horizontal-vertical bias and the diagonal-axial bias as defined above. The horizontal-vertical bias is the load-bearing recapture feature: the spec expects it to be systematically elevated on recaptured images because screen sub-pixel geometry and capture optics concentrate energy on horizontal and vertical orientations.
4. Calibrate the threshold. The spec calls the threshold policy-calibrated; it does not hand you a number. Gather a set of known-clean originals and a set of known recaptures representative of your channel (your document types, your typical phones and screens), measure the horizontal-vertical bias distribution for each, and pick an operating threshold at the false-positive rate your workflow tolerates. Recalibrate as your device mix changes.
5. Emit a recapture probability score, not a verdict. The classifier outputs a score against the threshold. Keep it as a probability so downstream policy can weigh it against other signals rather than hard-blocking on one feature.
6. Combine with lineage and synthesis signals if you have them. If you also run the anchor network, feed recapture probability, lineage-absence from the orphan detector, and synthesis probability into the composite risk aggregator, and let a signed policy object decide the action. Recapture alone is a flag; the disclosure treats the final admissibility call as a policy decision over multiple structural inputs.

7. Log the decision. The disclosure records generation and consultation events for auditability. At minimum, persist the score, the threshold version, and the policy object version so a challenged decision is reproducible.

What This Does Not Give You

This is an architecture, not a shipping library. There is no package to install and nothing here is benchmarked or production-proven. You implement the normalization, the gradient histogram, the bias computation, the calibration, and the scoring yourself, and you own the accuracy you achieve.

The recapture signal is a probability, not proof. The disclosure describes a systematic elevation in horizontal-vertical bias, not a guaranteed separation, and it does not state detection rates or error bounds. A clean original that happens to be dominated by horizontal and vertical structure (a table, a form, a screenshot of text that was never recaptured) can push the bias up, and a high-quality recapture can push it down. Calibration is where you manage that tradeoff, and it is channel-specific: a threshold tuned on one set of screens and cameras will not transfer unchanged.

Recapture is also not the same as fraud. A user may legitimately photograph a screen. Structural unanchoredness, per the disclosure, is likewise not inherently impermissible; it only matters under a policy that requires provenance. Treat these as inputs to a policy decision, not as accusations.

Finally, this addresses recapture geometry, not every attack. It is not a general deepfake detector, not a document tamper-localizer, and not a substitute for content-specific validation of the fields inside the document.

Disclosure Scope

The approach described here is disclosed in PCT International Application No. PCT/US26/28630, within the Content Anchoring inventive step covering structural content identity and rights-grade admissibility for digital artifacts. This guide is educational: it explains an architecture a skilled developer can build from the disclosure, and every mechanism described traces to that filing. It is not a warranty, a benchmark, a shipping product, or an offer of software, and nothing here guarantees any particular detection accuracy in your deployment.

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)