International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

ForgeGuard: A Hybrid Classical Signal Processing and Lightweight CNN Ensemble for Nine-Category Document Forgery Detection and Localization

Mr. Jenish Ashokbhai Bambhroliya
Computer Science and Engineering
with Cyber Security Independent Researcher
Vadodara, Gujarat, India

Ms. Jinal Ashokbhai Bambhroliya
Cyber Security
Independent Researcher
Oshawa, Ontario, Canada

***Abstract*— *Insurance fraud through document manipulation has become a persistent threat to health financing systems, one that grows more difficult to detect as AI-based forgery tools become widely accessible. The core problem is that most existing detection methods answer only a binary question—real or fake—without telling examiners where tampering occurred or what kind of manipulation was applied. This paper introduces ForgeGuard, a system designed around a different premise: that each category of document tampering leaves a distinct forensic signature, and that those signatures can be detected reliably using methods chosen specifically for each category. ForgeGuard handles nine tampering types—intra-document copy-paste, text overwriting, content insertion, content erasure, document splicing, watermark removal, spacing anomalies, fully AI-generated documents, and partial AI-based field edits—through a hybrid pipeline that pairs classical forensic signal analysis (Error Level Analysis, DCT block fingerprinting, wavelet noise floor, Gabor texture filters) with three lightweight convolutional networks (MobileNetV2, ResNet50, EfficientNet-B0). A two-stage detection funnel with per-document adaptive thresholds separately manages recall and precision. Notably, the system requires no Large Language Model and runs entirely on commodity CPU hardware. On a curated dataset of 630 medical insurance claim documents—augmented to over 4,500 training samples—ForgeGuard reaches a weighted F1 of 0.958 at IoU = 0.5 across all nine categories, improving over the best baseline by more than 17 percentage points.*

Keywords*— *document forgery detection, image forensics, Error Level Analysis, wavelet noise analysis, DCT fingerprinting, hybrid CNN, medical document integrity, multi-category classification, insurance fraud detection

I. INTRODUCTION

Among the different kinds of insurance fraud, document manipulation is particularly troubling because it is hard to detect without examining physical evidence that most reviewers simply do not have access to. A discharge summary with an altered amount, a lab report bearing a stamp that belongs to a different hospital, a name quietly changed by an AI inpainting tool—these alterations look convincing on screen and pass routine checks.

The practical scale of the problem compounds matters. Health insurance systems in countries like India process several million claims each year. No organisation can employ enough trained reviewers to examine every supporting document manually, and even experienced examiners miss subtle manipulations. Automation is not merely convenient; it is necessary.

Existing automated approaches have a common limitation: they frame forgery detection as a binary classification task. The system answers whether a document is genuine or not, but says nothing about where the manipulation occurred, what type it was, or how confident the detection is. That level of output is insufficient for a reviewer who must decide whether to approve, reject, or escalate a claim.

ForgeGuard was built to address this gap. Rather than asking a single binary question about the whole document, it interrogates each candidate region using a detection method suited to the specific tamper type that region might contain. The result is a system that identifies not just whether a document was tampered with, but where, and in what way.

A. Contributions

The paper makes four contributions to the document forensics literature.

First, a nine-category tampering taxonomy is defined that covers the full range of manipulations observed in medical insurance claim documents, from straightforward copy-paste operations to sophisticated AI-generated synthetic records.

Second, a hybrid detection pipeline is proposed in which classical signal processing methods and lightweight convolutional networks are combined through soft-voting ensembles, with each of the nine categories having its own detector tuned to its specific forensic signature.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Third, a two-stage confidence funnel is introduced that uses a loose per-document adaptive threshold in the first stage to maximise recall, then filters candidates in the second stage through multi-method agreement to recover precision.

Fourth, the full pipeline operates without any Large Language Model, external API, or GPU requirement, making it deployable in cost-constrained and network-isolated environments.

II. RELATED WORK

A. Document Image Forensics

Work on document forgery detection has accelerated considerably in the last few years. Okamoto et al. [1] built a dataset by emulating DNN-based forgery attacks on document images and introduced a self-supervised pretraining strategy that combined natural photographs with document images to improve generalisation. Their findings showed that methods designed for natural images—particularly copy-move detectors—perform poorly on documents because forged regions can be blended seamlessly into the structured, uniform document background.

CTP-Net [2] took a different approach by exploiting character texture, noting that printed text has predictable stroke characteristics that are disrupted when regions are tampered with. The network was evaluated on datasets of Chinese and English documents and showed meaningful improvements over prior pixel-level forgery detectors, though it was designed for a single tamper category rather than multiple.

ADCD-Net [3] addressed the compression robustness problem directly by introducing adaptive DCT feature extraction combined with hierarchical content disentanglement. The system showed substantially better performance under JPEG compression and cropping attacks than fixed-feature-extraction methods. ForgeGuard draws on DCT analysis in both the copy-paste detector (T1) and the AI-generated document detector (T8).

DocForgeNet [4] proposed a dual-stream fusion network processing RGB and DCT features through parallel CNN and transformer branches linked by cross-linear attention. Their evaluation on a dataset of 170,000 document images demonstrated strong precision and recall. Like most prior work, however, DocForgeNet is a binary detector and does not support classification of tamper types.

The Fine-Grained Document Tampering Detection (FGDTD) benchmark [5] is the closest precedent for multi-category document forensics, identifying both tampered regions and tampering methods across 16,479 images from twelve document datasets. It targets pixel-level segmentation masks rather than bounding boxes, however, and does not include AI-generated document detection as a category.

The AIForge-Doc benchmark [6] is worth noting separately because it established that even well-regarded detectors achieve only AUC = 0.751 on AI-forged financial and form documents, confirming that detecting machine-generated fakes remains an open and hard problem.

B. Insurance Fraud and Structured Data

The insurance fraud literature has focused almost exclusively on structured claim records rather than document images. Hong et al. [7] built a graph-based learning system that captures relationships between claims to identify anomalous billing patterns. Work reviewed in Scientific Reports [8] applied deep learning models to tabular insurance data. These systems operate on billing codes and procedure records—they have no visibility into the supporting document images and cannot detect visual manipulation.

C. AI-Generated Content and Mixed Forgeries

The OSTF dataset [9], presented at AAAI 2025, examined forgeries produced by replacing text in natural scene photographs with diffusion-generated content. One of the more important findings was that detectors trained on purely synthetic forgeries generalise poorly to forgeries crafted by humans. This underscores the need for domain-specific training data, which is why the ForgeGuard dataset focuses on medical insurance documents specifically. A hybrid ViT-SVM approach [10] and multi-label Swin-T work [11] both noted that detecting multiple tamper types simultaneously is harder than binary detection and remains underexplored.

D. Summary of Gaps

Reviewing the literature identifies four gaps ForgeGuard is designed to address: no existing single system handles all nine document tampering categories; no prior work provides type-specific bounding-box localization for medical insurance documents; multilingual document forensics (English plus Hindi) has received almost no attention; and no published system handles both classical forgeries and AI-generated documents in the same pipeline without relying on LLMs or GPU infrastructure.

III. TAMPERING TAXONOMY

Nine tampering categories were defined through systematic review of real-world manipulation patterns observed in medical insurance claim processing. The categories are listed in Table I together with the primary forensic signals exploited by each detector. Two aspects of this taxonomy are worth noting briefly.

Categories T1 through T7 represent manipulations that could in principle be performed without AI tools—though digital tools make some of them far easier than they once were. Categories T8 and T9 specifically address the growing threat from AI-based generation: T8 targets documents fabricated entirely by a generative model, while T9 targets the subtler case where one or two fields have been quietly altered using AI inpainting while the surrounding document remains authentic.

T9 arguably deserves particular attention. A fraudster who understands that binary detectors look for gross statistical inconsistencies can produce a document that passes those checks simply by inpainting a single name or amount rather than replacing large blocks of content. ForgeGuard's Named Entity Recognition localisation step was designed specifically to find these targeted edits.

TABLE I. NINE-CATEGORY TAMPERING TAXONOMY

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

ID	Category	Description	Forensic Signal
T1	Copy-Paste	Content duplicated within document	DCT block matching, ORB-RANSAC
T2	Overwriting	Existing text replaced by overwriting	Font metric deviation, multi-quality ELA
T3	Insertion	New text, stamps, or signatures added	JPEG ghost analysis, noise mismatch
T4	Erasure	Original content removed or concealed	Gabor texture, DCT flatness
T5	Splicing	Sections merged from different documents	FFT phase shift, font clustering
T6	Watermark Removal	Official watermarks removed	Spectral holes, entropy loss
T7	Spacing Anomaly	Word or character spacing manipulated	Per-line gap statistics
T8	Fully Synthetic	Entire document generated by AI tools	Wavelet HH variance, power spectrum
T9	Partial AI Edit	Specific fields edited using AI inpainting	NER localisation, noise inconsistency

IV. THE FORGEGUARD SYSTEM

A. Overall Architecture

The pipeline has four stages: preprocessing, parallel detection, postprocessing, and output generation. Each stage is described below. All nine category detectors run independently on the same preprocessed page, and their outputs are merged in the postprocessing stage.

B. Preprocessing

PDF pages are rendered to JPEG at 300 DPI using PyMuPDF. Two distinct copies of each page are maintained throughout all subsequent processing. The forensic copy retains the raw, unmodified pixel data from rendering. It is used only for Error Level Analysis, noise profiling, frequency analysis, and wavelet decomposition—operations whose validity depends on the pixel values not being altered. The clean copy undergoes deskewing via Hough Transform angle estimation (correction applied only where skew exceeds 0.5 degrees) and then Non-Local Means denoising (h = 3). The clean copy serves OCR extraction, font metric analysis, and spacing statistics.

Pages with an estimated resolution below 150 DPI are upsampled to 300 DPI using Lanczos interpolation before either copy is produced. CLAHE with a clip limit of 2.0 is applied to the clean copy's L channel in LAB space to recover local contrast in documents captured under uneven lighting.

C. Two-Stage Detection Funnel

A single global threshold applied across all documents performs poorly because different document types have very different baseline noise levels. A discharge summary printed on a laser printer and scanned at 300 DPI will look quite different from a hand-completed form photographed on a mobile phone. To handle this, ForgeGuard applies an adaptive threshold derived from per-page statistics.

In Stage 1, each sub-method within a detector fires at a loose threshold of page_mean + 1.0σ. The intention is to flag everything plausibly suspicious; false positives at this stage are acceptable because Stage 2 will filter them. In Stage 2, a candidate region is passed to the output only if at least two independent sub-methods agree on it, with each sub-method applying a tighter threshold of page_mean + k×page_std. The k values were chosen by grid search on the validation split: T1 = 2.5, T2 = 1.8, T3 = 1.5, T4 = 2.0, T5 = 2.2, T6 = 1.5, T7 = 2.5, T8 uses an absolute threshold on the HH subband variance, and T9 = 1.5.

D. Category Detectors

T1 – Copy-Paste: Three methods are combined through soft voting. Block DCT Fingerprinting divides the forensic copy into overlapping 16×16-pixel blocks at stride 8, computes 2D DCT coefficients, L2-normalises each 256-dimensional vector, and sorts all block vectors lexicographically. Pairs of adjacent sorted vectors with cosine similarity exceeding 0.97, where the spatial distance between block centres is greater than 30 pixels, are flagged as copy-paste candidates (weight 0.40). ORB Keypoint Matching extracts 5,000 keypoints using OpenCV ORB, matches descriptors with Hamming distance, and applies RANSAC homography at 3-pixel inlier threshold; any region pair with eight or more inliers is confirmed (weight 0.35). PatchMatch finds near-duplicate 8×8 pixel patches at different spatial locations (L2 distance < 12.0) and clusters matches using DBSCAN (weight 0.25). The output YAML for T1 records only bounding-box coordinates (h, w, x, y) for both source and destination regions.

T2 – Overwriting: A key diagnostic hint is that overwritten text tends to have measurably different stroke characteristics from the surrounding authentic text. Font Metric Analysis via Tesseract HOCR extracts stroke width per word; words whose stroke-width z-score exceeds 2.5 relative to the page mean are treated as suspicious (weight 0.35). Multi-quality ELA re-saves the forensic copy at quality levels 50, 70, and 90; the score ela_90 − ela_70 captures the double-compression artifact that overwriting introduces (weight 0.35). Stroke Width Transform variance above 6.0 per text region indicates layered text (weight 0.20). Laplacian of Gaussian double-edge detection identifies blocks where edge density is more than twice the page median (weight 0.10). T2 YAML contains only bounding-box coordinates.

T3 – Content Insertion: When content is inserted from an external source, it almost always carries a different JPEG compression history than the host document. JPEG Ghost Analysis captures this by re-saving the forensic copy at six quality levels (50, 60, 70, 80, 90, 95) and taking the minimum per-pixel absolute difference; regions where that minimum exceeds 8.0 indicate foreign content (weight 0.45). Noise Profile Mahalanobis Distance flags regions whose local noise standard deviation lies more than 2.5σ from the page noise model (weight 0.35). Sobel Gradient Direction comparison identifies regions whose illumination gradient direction is inconsistent with the surrounding page, measured by cosine distance exceeding 0.4 (weight 0.20). A MobileNetV2 stamp classifier, trained on augmented samples and applied at three window scales, provides supplementary CNN confidence. The T3 YAML output includes a sub-type field specifying whether the detected insertion is text, a stamp, or a signature.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

T4 – Content Erasure: Erasure leaves behind a region that is conspicuously smoother or texturally inconsistent with its surroundings. A Gabor filter bank with 8 orientations and 3 scales (λ = 4, 8, 16 pixels) followed by GLCM feature extraction identifies blocks with Mahalanobis distance exceeding 3.0 from a Gaussian mixture model fitted to authentic background blocks (weight 0.35). Bilateral Smoothness Analysis flags regions where the bilateral-filtered residual falls below 2.0 AND local standard deviation below 1.5—characteristic of digital painting (weight 0.30). DCT Coefficient Variance Detection flags 8×8 blocks where the sum of absolute non-DC DCT coefficients is below 60% of the page mean (weight 0.25). Autocorrelation Analysis detects the periodic repetition patterns left by content-aware fill algorithms at offsets exceeding 16 pixels (weight 0.10). The T4 YAML type field is always 'erased'.

T5 – Document Splicing: JPEG-encoded documents tile DCT coefficients on an 8×8-pixel grid; when two sections of a composite document came from sources that were not aligned to the same grid, the grid phase shifts at the boundary. FFT Block Grid Phase Shift detects this by computing the 2D FFT in horizontal strips of 64 pixels and measuring phase change of the 1/8-frequency peak between adjacent strips; a shift exceeding $\pi/4$ signals a splice (weight 0.40). Font DBSCAN clusters text lines by their [stroke width, character height, inter-character spacing] vectors; two or more distinct clusters indicate text from different source documents (weight 0.30). YCbCr Chrominance Clustering applies KMeans (k = 2) to 2D Cb-Cr histograms per 64×64-pixel block; distinct spatial clusters indicate different sources (weight 0.20). Seam Line Detection via Hough Transform flags near-horizontal lines spanning more than 70% of the page width (weight 0.10). Vertical position determines the sub-type: top 20% of page height is header, bottom 20% is footer, middle 60% is body. The T5 YAML records body_source and header_source fields.

T6 – Watermark Removal: Removing a watermark leaves behind a frequency-domain absence where the watermark's periodic signal used to be. FFT Spectral Hole Detection measures this absence as 1 − (observed peak energy / expected peak energy) (weight 0.50). Local Shannon Entropy Mapping flags regions with entropy below 1.5 bits in zones where the watermark would have contributed texture variation (weight 0.30). Gabor Background Texture Regularity checks for over-smoothness at the expected watermark angle, configurable per document type and defaulting to 45 degrees (weight 0.20). T6 YAML records only bounding-box coordinates.

T7 – Spacing Anomaly: A design decision that proved critical for this detector was computing all statistics per line rather than per page. Documents routinely mix body text, captions, headings, and tables with legitimately different spacing norms; using page-level statistics to flag spacing anomalies produces an unacceptable false positive rate. Per-line normalisation was found to suppress roughly 80% of false positives without affecting recall. Inter-Word Gap Analysis fits a Gaussian to gaps in each line of at least four words and flags gaps with absolute z-score exceeding 2.5 (weight 0.40). Inter-Character Gap Analysis applies the same threshold within words where OCR confidence exceeds 70% (weight 0.35). Baseline Alignment fits a regression line through word baselines per line; deviations exceeding 3.0 pixels suggest character replacement (weight 0.15). Kerning Consistency measures common character-pair kerning ratios (weight 0.10). A veto rule suppresses false positives from legitimate justified text: if all inter-word gaps in a line are uniformly expanded (mean more than 1.5 times normal AND standard deviation less than 0.3 times the mean), the line is classified as justified rather than manipulated. The T7 YAML records a stretch_factor field equal to anomalous_gap divided by the line mean gap.

T8 – Fully AI-Generated: Real scanned documents carry scanner-specific noise that AI generation tools do not replicate accurately. Wavelet Noise Floor Analysis decomposes the forensic greyscale image with the Daubechies-4 wavelet and measures variance in the HH diagonal detail subband. Values below 0.30 indicate a synthetically smooth image; values above 8.00 suggest a GAN artifact; the authentic scanner noise range is 0.80 to 4.50 (weight 0.30). Radially Averaged Power Spectrum analysis identifies the periodic spectral peaks at 1/N frequencies (N = 64, 128, 256) that GAN and diffusion models characteristically produce (weight 0.25). Halftone Frequency Band Analysis checks for the presence or conspicuous absence of halftone dot patterns at approximately 45 degrees in the FFT spectrum, which real print-and-scan documents reliably exhibit (weight 0.10). PDF Metadata Forensics flags missing scanner manufacturer strings, identical creation and modification dates, and generic placeholder metadata (weight 0.15). Text Regularity Analysis flags documents where the coefficient of variation of inter-word gaps is below 0.05 on a purportedly scanned document; genuine scans always show natural variation above 0.15 from physical printing and paper handling (weight 0.10). A ResNet50 classifier trained on augmented samples contributes 50% of the final ensemble score when model weights are available. T8 is a document-level decision; no YAML bounding-box file is produced.

T9 – Partial AI Edits: Detection is a two-stage process. Stage 1 localises named entity regions using spaCy (PERSON, DATE, MONEY entity types) supplemented by regex patterns for domain-specific identifiers such as insurance policy numbers and INR currency amounts. Stage 2 verifies each localised candidate. Local Noise Inconsistency computes the ratio of entity-region noise standard deviation to surrounding-region noise; a ratio below 0.5 or above 2.0 indicates manipulation (weight 0.35). Font Metric Deviation compares stroke width and character spacing z-scores of the entity text against the body text distribution; a combined z-score above 3.0 is flagged (weight 0.30). An EfficientNet-B0 inpainting artefact classifier trained on synthetically inpainted patches contributes CNN confidence (weight 0.20). Format Validation applies rule-based checks including date calendar validity, dates predating the scheme's 2018 launch, unrealistic currency amounts, and malformed identifier patterns (weight 0.10). A copy-paste residue sub-scan using the T1 detector covers the case where a fraudster relocated text rather than generating it (weight

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

0.05). The T9 YAML records the detected forged value in a 'new' field and the estimated original value in an 'old' field.

E. Postprocessing

After all nine detectors have been applied, four postprocessing steps are run sequentially. Non-maximum suppression at an IoU threshold of 0.40 removes duplicate bounding boxes within each category. Overlapping same-category boxes with IoU above 0.60 are merged. A global confidence veto discards boxes smaller than 15×15 pixels and those with confidence below 0.35. A margin veto removes detections in blank page margins covering more than 30% of the page area, which are typically false positives from the texture detectors.

V. DATASET AND AUGMENTATION

A. Data Collection

The dataset consists of 630 medical insurance claim documents assembled from real-world document processing workflows. The document types include discharge summaries, laboratory reports, diagnostic imaging reports, and itemised billing statements. All nine tampering categories are represented, with 50 to 74 raw samples per category. File formats are a mix of scanned PDFs and direct JPEG and PNG images. Resolution ranges from 72 DPI (mobile phone photographs) to 300 DPI (flatbed scanner output). Content is in English and Hindi.

B. Augmentation Strategy

Given that 50 to 70 raw samples per category is insufficient to train convolutional classifiers without overfitting, a nine-type augmentation pipeline was used to expand each category to a minimum of 483 samples. The augmentation types were: brightness adjustment (factor 0.7–1.3), contrast adjustment (0.8–1.2), Gaussian blur (σ = 0.5–1.5), additive Gaussian noise (σ = 2–8), JPEG recompression (quality 60–85), slight rotation (±3 degrees), scale variation (0.90–1.10×), shadow simulation (fold effect), and region cropping (10–20% margin removal). Augmentations are applied in random combinations of one to three per sample.

A deliberate choice was made not to apply augmentations that would destroy the forensic signals being targeted. ELA-based detectors require accurate JPEG compression history; any augmentation involving JPEG recompression of the full image at fixed quality was therefore excluded from samples used for T2 and T3 training. Similarly, noise augmentation was excluded from T8 training samples to preserve the wavelet noise floor characteristics being learned.

TABLE II. DATASET STATISTICS (80/20 TRAIN-VALIDATION SPLIT)

Category	Raw	Augmented	Train	Validation
T1 Copy-Paste	63	504	403	101
T2 Overwriting	58	522	418	104
T3 Insertion	71	497	398	99
T4 Erasure	54	513	411	102
T5 Splicing	67	536	429	107
T6 Watermark	52	520	416	104
T7 Spacing	69	483	387	96
T8 AI-Generated	74	518	415	103
T9 Partial AI	61	488	391	97
Total	569	4,581	3,668	913

VI. EXPERIMENTS AND RESULTS

A. Implementation

ForgeGuard is implemented in Python 3.10. Signal processing relies on OpenCV 4.9, scikit-image 0.23, scipy 1.13, and pywavelets 1.6. OCR uses pytesseract 0.3.10 backed by Tesseract 5.x configured for both English and Hindi (eng+hin). Named entity recognition uses spaCy en_core_web_sm. The three CNN classifiers are built with PyTorch 2.3 (CPU build) using weights initialised from ImageNet via torchvision and timm. Training uses the Adam optimiser with an initial learning rate of 1×10^{-3} and cosine annealing. MobileNetV2 is trained for 15 epochs with batch size 32; ResNet50 for 10 epochs; EfficientNet-B0 for 12 epochs. No GPU was used at any stage of training or evaluation.

B. Evaluation Protocol

All results are reported as F1 score at IoU = 0.5. A predicted bounding box counts as a true positive if it overlaps a ground-truth box of the same category with IoU of at least 0.5. The weighted overall score is the sum of (category_weight × category_F1) across the nine categories. Weights reflect the practical importance of each tamper type: T3 (insertion), T8 (AI-generated), and T9 (partial AI edits) each carry 15%; T1, T2, T4, T5, and T7 each carry 10%; T6 (watermark removal) carries 5%. Category T8 is a document-level classification with no bounding-box requirement.

C. Ablation Study

Three configurations are compared: (A) classical signal processing only, (B) trained CNNs only, and (C) the full ForgeGuard hybrid with two-stage filtering.

TABLE III. ABLATION STUDY (WEIGHTED F1, IoU = 0.5)

Category	Wt.	Classical	CNN Only	ForgeGuard
T1 Copy-Paste	10%	0.921	0.873	0.960
T2 Overwriting	10%	0.906	0.841	0.951
T3 Insertion	15%	0.918	0.934	0.972
T4 Erasure	10%	0.895	0.812	0.941
T5 Splicing	10%	0.911	0.863	0.951
T6 Watermark	5%	0.887	0.794	0.934
T7 Spacing	10%	0.908	0.771	0.942

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Category	Wt.	Classical	CNN Only	ForgeGuard
T8 AI-Generated	15%	0.901	0.931	0.971
T9 Partial AI	15%	0.864	0.891	0.962
Weighted F1	—	0.901	0.873	0.958

The results reveal a pattern worth noting. For categories T1 through T7—those associated with physically identifiable manipulations—classical methods outperform CNN-only classifiers. The signal processing is deterministic and has no dependency on having seen similar training examples. Categories T8 and T9 tell the opposite story: the CNN classifiers learn subtle texture statistics from the augmented synthetic documents that are difficult to encode in explicit signal processing rules. The hybrid achieves the best of both, with the gains most pronounced in T9 (+9.8 percentage points over classical) and T8 (+7.0 points).

D. Comparison with Prior Methods

Table IV shows ForgeGuard alongside three baselines evaluated on the same dataset. CTP-Net and DocForgeNet scores were obtained by running the original implementations; the binary ELA classifier is a simple convolution-based approach applied to Q90 ELA maps.

TABLE IV. COMPARISON WITH BASELINE METHODS

Method	T1	T3	T7	T8	T9	Wt. F1
Binary ELA	0.641	0.698	N/A	0.534	0.487	0.601
CTP-Net [2]	0.782	0.819	0.612	0.641	0.598	0.718
DocForgeNet [4]	0.831	0.861	0.683	0.712	0.661	0.779
ForgeGuard	0.960	0.972	0.942	0.971	0.962	0.958

Binary ELA cannot produce any output for T7 because ELA has no sensitivity to spacing manipulation; the gap is marked N/A rather than zero. The improvement of ForgeGuard over DocForgeNet is largest in T7 (+25.9 percentage points) and T9 (+30.1 points), both of which are categories where category-specific detector design provides an advantage that generic dual-stream fusion cannot replicate.

E. Main Results

TABLE V. FORGEGUARD FULL RESULTS (IoU = 0.5)

Category	Wt.	Precision	Recall	F1	Target	Pass
T1 Copy-Paste	10%	0.971	0.950	0.960	0.95	✓
T2 Overwriting	10%	0.963	0.940	0.951	0.94	✓
T3 Insertion	15%	0.974	0.970	0.972	0.96	✓
T4 Erasure	10%	0.951	0.931	0.941	0.93	✓
T5 Splicing	10%	0.961	0.941	0.951	0.94	✓
T6 Watermark	5%	0.941	0.927	0.934	0.92	✓
T7 Spacing	10%	0.952	0.932	0.942	0.93	✓
T8 AI-Generated	15%	0.978	0.963	0.971	0.96	✓
T9 Partial AI	15%	0.971	0.953	0.962	0.95	✓
Weighted Overall	100%	0.964	0.951	0.958	—	✓

All nine categories exceed their individual F1 targets. The highest individual F1 scores are in T3 (0.972) and T8 (0.971). T6 (watermark removal, 0.934) is the most challenging category, which is consistent with the relatively subtle evidence that watermark removal leaves. Weighted overall F1 is 0.958, against a target of 0.945.

F. Runtime

Processing one document page takes approximately 25.1 seconds on an Intel Core i7 CPU with no GPU acceleration. The dominant costs are T9 (4.1 s, NER plus CNN inference) and T1 (3.1 s, block DCT sorting plus ORB matching). Running all nine detectors in parallel would reduce total time to an estimated 6 seconds. Even at 25 seconds sequential, batch processing overnight is practical for organisations handling thousands of claims.

VII. DISCUSSION

A. Strengths of the Hybrid Design

The ablation results tell a clear story about why the hybrid outperforms either component alone. Classical signal processing methods are reliable where the forensic evidence is deterministic—the double compression artifact in overwritten text, the DCT grid phase discontinuity at a splice boundary, the absence of scanner noise in a synthetic document. These signals do not require training examples; they arise from physical and mathematical properties of imaging and JPEG compression.

CNN classifiers add value precisely where those deterministic signals are weak or absent: the subtle inpainting texture of a partially AI-edited field, the specific spatial pattern of GAN spectral artifacts. Neither approach alone covers all nine categories well. The combination does.

B. Limitations

Three limitations are worth being candid about. The dataset is small by the standards of general computer vision benchmarks. Augmentation mitigates this but does not eliminate the risk that the validation F1 scores overestimate performance on the full hidden evaluation dataset, which may contain document types or manipulation techniques not well represented in the 630-document training corpus.

The T8 wavelet classifier was not tested against output from the most recent diffusion models, such as FLUX or Stable Diffusion 3.5, which produce images with substantially more realistic sensor noise than earlier models. It is likely that the noise floor threshold would need recalibration for such inputs.

Finally, JPEG compression at quality below 50 significantly degrades ELA-based detection for T2 and T3.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Mobile-photographed documents are sometimes saved at very high compression ratios by messaging applications before submission, and the system's performance on such inputs would need separate evaluation.

C. Deployment Considerations

ForgeGuard was designed for use as an assistive tool by human document reviewers, not as an autonomous decision-making system. In practice, the output should be treated as a ranked list of regions requiring closer examination, not as a definitive verdict on document authenticity. False positive rates in PII-redacted zones are expected and are tolerated in the evaluation protocol. Any deployment in a context that affects individuals' claims should include a clearly documented human review step for any positive detection.

VIII. CONCLUSION

This paper presented ForgeGuard, a nine-category document forgery detection and localisation system designed for medical insurance claim documents. The central design principle—that each tampering category has a distinct forensic signature and deserves a detector tuned to that signature—leads naturally to a hybrid architecture combining classical signal processing with category-specific lightweight CNN classifiers.

The system achieves a weighted F1 of 0.958 across all nine categories on a curated dataset of 630 medical insurance documents augmented to over 4,500 training samples, improving over the best prior baseline by more than 17 percentage points. It operates without any Large Language Model, external API, or GPU requirement.

Future directions include building a larger publicly available annotated dataset of medical document forgeries to enable more robust CNN training, extending detection to camera-phone-captured and video-frame-extracted documents, and developing per-detection natural language explanations suitable for non-specialist reviewers.

ACKNOWLEDGMENT

This research paper is hereby dedicated to my revered teachers and dear family members whose guidance in their own ways has immensely helped me during the process of learning from nursery to the present time. I wish to wholeheartedly thank all my teachers for their invaluable guidance, knowledge, inspiration, and encouragement that has been of great help to me in completing this research successfully. I owe an equal debt of gratitude to my dear family members who have provided me constant encouragement and motivation by their continuous support, patience, and confidence in me so that I could successfully complete this milestone of mine. This research paper, thus, reflects my effort as well as the qualities and inspirations that I received during my entire academic career.

REFERENCES

[1] Y. Okamoto, G. Osada, I. Yahiro, R. Hasegawa, P. Zhu, and H. Kataoka, "Image Generation and Learning Strategy for Deep Document Forgery Detection," arXiv:2311.03650, 2023.

[2] J. Chen et al., "CTP-Net: Character Texture Perception Network for Document Image Forgery Localization," Proc. AAAI Conf. Artif. Intell., 2024.

[3] K. Wong, J. Zhou, H. Wu, Y. Si, and J. Zhou, "ADCD-Net: Robust Document Image Forgery Localization via Adaptive DCT Feature and Hierarchical Content Disentanglement," arXiv:2507.16397, 2025.

[4] DocForgeNet: Dual Cross-Stream Fusion for Robust Forgery Detection in Scanned Documents, Springer ICPR Proceedings, 2025.

[5] S. Sun and X. Qin, "Towards Fine-Grained Document Tampering Detection: New Dataset and Benchmark," PRCV 2025, Shanghai, China, Oct. 2025.

[6] AIForge-Doc: A Benchmark for Detecting AI-Forged Tampering in Financial and Form Documents, arXiv:2602.20569, 2026.

[7] Hong et al., "Multi-channel Heterogeneous Graph Structure Learning for Health Insurance Fraud Detection," 2024.

[8] Dept. CSE, Amrita Vishwa Vidyapeetham, "Insurance Claims Estimation and Fraud Detection with Optimized Deep Learning," Sci. Rep., vol. 15, art. 27296, Jul. 2025.

[9] C. Qu, Y. Zhong, F. Guo, and L. Jin, "Revisiting Tampered Scene Text Detection in the Era of Generative AI," AAAI 2025, pp. 694–702.

[10] F. Siddiqui et al., "Hybrid Framework for Image Forgery Detection using Vision Transformer and SVM," Sci. Rep., vol. 15, art. 40371, Nov. 2025.

[11] L. Li et al., "Multi-label Classification for Image Tamper Detection Based on Swin-T Segmentation Network," PeerJ Comput. Sci., 2025.

[12] A. Verma, P. Pandey, and M. Khari, "ELA-Conv: Forgery Detection in Digital Images Based on ELA and CNN," Springer RTIP2R, 2024, pp. 210–225.

[13] A. Akram et al., "Advanced Digital Image Forensics: A Hybrid Framework for Copy-Move Forgery Detection," J. Forensic Sci., vol. 70, no. 5, pp. 1801–1823, 2025.

[14] Unmask Tampering: Efficient Document Tampering Localization under Recapturing Attacks, ACM CCS 2025.

[15] Explainable Image-Centric Forgery Detection: A Survey, IEEE Trans. Intell. Syst. Technol., 2025.

[16] H. R. Suresh et al., "Deep Learning-based Image Forgery Detection System," Int. J. Electron. Secur. Digit. Forensics, vol. 16, no. 2, pp. 160–172, 2024.

[17] P. Badar, G. Geetha, and T. R. Mahesh, "Image Integrity and Tampering Detection: Hybrid Approach using ORB-SSD and CNN," Edelweiss ASET, vol. 9, no. 10, pp. 704–720, 2025.

[18] J. V. Beusekom, F. Shafait, and T. M. Breuel, "Text-Line Examination for Document Forgery Detection," IJDAR, 2013.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)