International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Explainable Dual-Model Framework for AI-Generated Image and Image-Tampering Detection

Vindyashree J[1], Pooja A R[1], Amrutha R N[1], Pooja Y N[1]
[1]Department of CSE - Data Science, SJB Institute of Technology, Bangalore, India

***Abstract*—Looking at a photo is no longer enough to tell whether it is real. Image generators can now produce a convincing picture in a few seconds, and almost any phone editor can change part of a real one. We built Trustify AI to check uploaded images for both problems, and also to show the user what each verdict is based on. Two EfficientNet-B0 networks look at every upload. The first was fine-tuned on CIFAKE and looks for signs of AI generation; the second was fine-tuned on CASIA 2.0, after we stripped out duplicate images, and looks for signs of tampering. A small set of rules turns their two scores into a label (Original, AI Generated, or AI Edited/Manipulated). Along with the label, the user sees a confidence value and a trust score, plus the file's properties, any EXIF data that is still present, and Grad-CAM heatmaps. The detectors scored 89.56% and 78.87% on their held-out test sets. When we then tried 13 real photos from outside both datasets, the CIFAKE model called 11 of them fake. For us this was the main lesson of the work, since a good benchmark score clearly did not mean the detector would work on ordinary pictures. So instead of a single score, Trustify AI shows the user several kinds of evidence and states where it is likely to be wrong.**

***Keywords*—Image Authenticity, AI-Generated Images, Image Manipulation, Explainable AI, Grad-CAM, EfficientNet-B0.**

I. INTRODUCTION

Before an image reaches us today, it has usually gone through a phone camera and a chat app or social feed, often an editor, and sometimes a generative model. When a picture finally reaches us, how it looks says very little about its origin. GANs [1] and diffusion models [2] can create believable scenes without any camera. Even a regular editor can change one part of a photo so cleanly that nobody notices. So a useful tool has to decide whether an image is authentic and also explain how it decided.

A lot of forensic work studies one kind of manipulation at a time, or tests on benchmarks whose images were captured and processed quite differently from everyday photos [3]. In practice, a detector often does well on its own test split and then breaks on pictures from somewhere else. For authenticity checks this matters a lot, since telling someone that their real photo is fake can hurt them. Our system, Trustify AI, uses two separately trained EfficientNet-B0 detectors together with a rule-based decision engine, metadata inspection, and Grad-CAM heatmaps, and we report the cases where it fails.

A. Domain Challenges

Checking image authenticity in practice runs into several problems that a single classifier does not handle well:

- **Camera-Free Images:** Generative models can make photorealistic pictures with no camera involved, so there is no capture history to check [1], [2].
- **Invisible Edits:** Ordinary editing tools can change part of a real photo and leave nothing a viewer would notice [5].
- **Opaque Verdicts:** A plain yes or no gives the user no way of judging how far to trust it, and a lone score may be badly calibrated on unfamiliar images.
- **Benchmark-to-Real Gap:** A detector that scores well on its test split can still fail on photos from other cameras, apps, and compression pipelines [3], [4].

B. Problem Statement

Right now, people usually judge whether an image is genuine either by studying it closely or by trusting one classifier score. Checking by eye is slow, and modern generators fool it easily; a single score, on the other hand, gives no reasons at all. The problem addressed in this work is therefore how to build an explainable framework that judges whether a digital image is authentic by combining separate AI-generation and manipulation detectors with metadata and visual explanations, and that is open about its confidence and its limits.

II. LITERATURE SURVEY

Table I summarizes the works this paper builds on, listed from newest to oldest.

Two results from this earlier work influenced our design the most. Wang et al. [4] showed that a classifier trained on one generator can catch images from others, but only with carefully chosen training data and augmentation, and that detectors easily memorize the fingerprints of whatever generator they saw. In her survey, Verdoliva [3] pointed to robustness against

new sources and post-processing as the biggest unsolved issue in media forensics. Earlier work shows that CNN detectors, metadata, and Grad-CAM are each useful. We could not find many systems that use all three together and also report how they perform on images outside the training benchmarks.

TABLE I
LITERATURE SURVEY OF KEY REFERENCES

Author(s)	Year	Title	Summary and Relevance
J. J. Bird and A. Lotfi	2024	CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images	Releases a balanced set of real CIFAR-10 and diffusion-generated images and inspects a CNN classifier with Grad-CAM; it is the training source for our AI-generation detector.
L. Verdoliva	2020	Media Forensics and Deepfakes: An Overview	Reviews detection methods for manipulated and synthetic media and points to poor robustness on new sources and post-processing as the main open problem.
S.-Y. Wang et al.	2020	CNN-Generated Images Are Surprisingly Easy to Spot… for Now	Shows that a classifier trained on one GAN can generalize to other generators when the training data and augmentation are chosen carefully.
J. Ho et al.	2020	Denoising Diffusion Probabilistic Models	Introduces diffusion models that create high-quality images by learning to reverse a gradual noising process, the family behind many current generators.
M. Tan and Q. V. Le	2019	EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks	Scales depth, width, and resolution together; its smallest model, B0, is the backbone of both our detectors.
R. R. Selvaraju et al.	2017	Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization	Builds class-specific heatmaps from gradients in the last convolutional layers; we use it to show what each detector relied on.
I. Goodfellow et al.	2014	Generative Adversarial Nets	Trains a generator against a discriminator, the approach that first made realistic synthetic images practical.
J. Dong et al.	2013	CASIA Image Tampering Detection Evaluation Database	Provides authentic and tampered photographs (splicing and copy-move) for testing tamper detectors; the source of our manipulation data.
J. Deng et al.	2009	ImageNet: A Large-Scale Hierarchical Image Database	Large labeled image collection whose pretrained weights and channel statistics we use for initialization and normalization.

III. FOUNDATIONAL CONCEPTS AND SYSTEM DESIGN

A. Core Task Distinctions

The work inside Trustify AI splits into three functions, summarized in Table II:

1) **AI-Generation Detection:** Deciding whether an image shows signs of having been produced by a generative model rather than a camera.
2) **Manipulation Detection:** Deciding whether a photograph shows traces of editing such as splicing or copy-move.
3) **Decision Fusion and Explanation:** Turning the two scores into one label and showing the evidence behind it.

TABLE II
TAXONOMY OF AUTHENTICITY ASSESSMENT TASKS

Task	Source of Evidence	Output
AI-generation detection	EfficientNet-B0 on CIFAKE	Signal $s_{ai} \in [0, 1]$
Manipulation detection	EfficientNet-B0 on CASIA 2.0	Signal $s_{man} \in [0, 1]$
Fusion and explanation	Rules, Grad-CAM, EXIF	Label, trust score, heatmap

B. Evidence Sources

The framework relies on three kinds of evidence:

- **Learned Visual Features:** Inside each detector, the convolutional backbone reduces the image to a pooled feature vector, which the classification head turns into a signal. These features are learned rather than hand-made, and they cannot be interpreted directly, so we added an explanation layer.
- **Metadata:** When a file carries EXIF data, we read camera make and model, capture date, editing software, whether GPS data exists, color space, and orientation. If the file has no EXIF data, the system simply reports that, without guessing how the photo was taken.
- **File-Level Properties:** Pillow gives the file type, size, dimensions, compression details, and bit depth, and a SHA-256 digest pins down exactly which file was analyzed. We show these only as background evidence; they are never used as the main classifier.

IV. CORE PRINCIPLES AND ALGORITHMS

Both detectors share the same structure: an EfficientNet-B0 [6] that starts from ImageNet weights [9] and is fine-tuned for a two-class problem. The only differences between them are the training data and which class is treated as positive.

A. Preprocessing and Normalization

Each image is converted to RGB and resized to 224 × 224 pixels, the input size of EfficientNet-B0. Pixel values are scaled to [0, 1], and each channel is

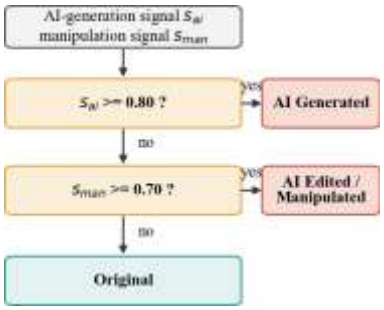


Fig. 1. Decision logic of the rule-based engine.

normalized with the ImageNet statistics the backbone was trained with:

$$\tilde{x}_c = \frac{\tilde{x}_c/255 - \mu_c}{\sigma_c} \quad (1)$$

Here c is the color channel, and μ_c and σ_c are the ImageNet mean and standard deviation for it. We use exactly the same steps during training and inference, so the inputs always have the same statistics.

B. Rule-Based Decision Engine

The CIFAKE detector treats FAKE as the positive class and returns the AI-generation signal s_{ai}, a number from 0 to 1 that rises as the model grows more certain that the image is synthetic. The CASIA detector treats manipulated images as positive and returns the manipulation signal s_{man}. Since CASIA contains older types of tampering such as splicing, and no edits made with generative AI, we only call this a manipulation detector. The label AI Edited/Manipulated is simply the name shown in the application. The engine does not average the two signals. Instead, each signal has its own threshold, and the rules are checked in a fixed order (Fig. 1):

$$\begin{aligned} &\text{y=AI Generated, if } s_{ai} \geq \tau_{ai} \\ &\text{y=AI Edited/Manipulated, if } s_{ai} \in \tau_{ai} \\ &\text{and } s_{man} \geq \tau_{man} \\ &\text{y=Original, otherwise} \end{aligned} \quad (2)$$

with $\tau_{ai} = 0.80$ and $\tau_{man} = 0.70$. We chose these values for practical reasons, and we have not tested whether they work as general forensic thresholds. The same caution applies to the trust score, which summarizes part of the model evidence for the user; it is not a calibrated probability that the image is real.

C. Grad-CAM Explainability

We use Grad-CAM [8] to show where the detector that made the call was looking. For a target class c, feature map k gets a weight equal to the average gradient of the class score y^c with respect to that map:

$$\alpha_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \quad (3)$$

and the heatmap is the weighted sum of the feature maps passed through a ReLU:

$$L^c = ReLU\left(\sum_k \alpha_k^c A^k\right) \quad (4)$$

An AI Generated result comes with the CIFAKE heatmap, and a manipulation result with the CASIA heatmap. In both cases the heatmap only marks regions that influenced the model output, and it should not be taken as proof that those regions were edited.

V. DATASETS AND MODEL TRAINING

A. CIFAKE

CIFAKE [7] puts real CIFAR-10 images next to synthetic ones made by a latent diffusion model. It has 100,000 training and 20,000 test images, half real and half fake, each only 32 × 32 pixels. Training on the whole set would have taken too long on our hardware. We therefore used a balanced subset of 20,000 images (10,000 per class) for training and kept the full 20,000-image test set for testing.

B. CASIA 2.0 and Leakage Control

CASIA 2.0 [5] has about 12,000 color images, some authentic and some tampered. While checking the processed files for exact duplicates, we found duplicate groups sitting across the original split boundaries, which means a model could have trained on images that later appeared in its test set. So we discarded the old splits and rebuilt them using only unique images. That left 9,916 images (5,818 authentic, 4,098 manipulated), divided into 7,932 for training, 990 for validation, and 994 for testing, with no exact-duplicate group crossing the new boundaries. About 59% of the cleaned set is authentic. Because of this imbalance, accuracy alone can make a model look better than it is, so we always report precision and recall together.

C. Training Configuration

TABLE III
TRAINING AND INFERENCE CONFIGURATION

Item	CIFAKE detector	CASIA detector
Backbone	EfficientNet-B0, ImageNet-pretrained	
Input	224 × 224 pixels, ImageNet mean and std	
Training data	20,000 images (10,000 per class)	7,932 / 990 / 994 (train / val / test)
Optimizer	AdamW	
Learning rate	1×10^{-3} (Stage 1), 1×10^{-5} (Stage 2)	1×10^{-4}
Epochs	3 + 5 (two stages)	5
Batch size / workers	8 / 0	8 / 0

Both detectors (Table III) were fine-tuned as binary classifiers with AdamW on a laptop GPU, which

limited us to a batch size of 8 and a data loader with no extra worker processes. The CIFAKE model was trained in two stages: three epochs at a learning rate of 1×10^{-3}, then five more at 1×10^{-5}. The CASIA model ran for five epochs at 1×10^{-4}, and we kept the checkpoint that did best on the validation set.

TABLE IV
HELD-OUT AND EXTERNAL PERFORMANCE OF THE TWO DETECTORS

Evaluation	Images	Accuracy	Precision	Recall	F1	Specificity	FPR
CIFAKE, held-out	20,000	89.56%	94.14%	84.36%	88.98%	94.75%	5.25%
CASIA, held-out	994	78.87%	70.64%	83.70%	76.61%	75.47%	24.53%
CIFAKE, external genuine	13	15.4%	n/a	n/a	n/a	15.4%	84.6%
CASIA, external genuine	13	69.2%	n/a	n/a	n/a	69.2%	30.8%

n/a: the external set contains only genuine images, so precision, recall, and F1 are undefined.

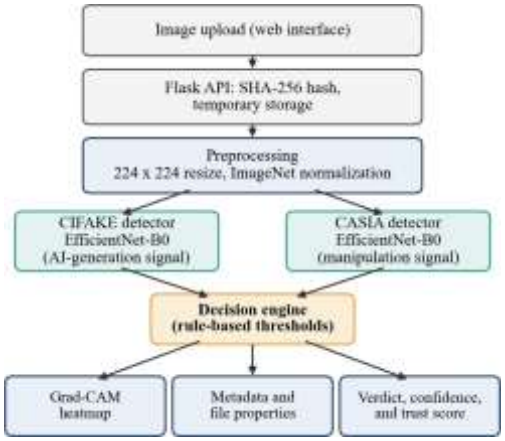


Fig. 2. Proposed workflow of the Trustify AI system.

VI. SYSTEM ARCHITECTURE AND IMPLEMENTATION

Fig. 2 shows how an image moves through the system: upload, preprocessing, AI-generation detection, manipulation detection, decision fusion, metadata extraction, Grad-CAM generation, and finally display.

The web front end hands every upload to a Flask API. On the server side, we hash the file with SHA-256, keep a temporary copy, run the two classifiers, and read the metadata. The API then returns a structured response containing the label, confidence, trust score, the reasons behind the label, both detector signals, the metadata, and the Grad-CAM images. Because the two detectors never share a score, the final screen can still show whether a suspicion came from generation or from editing. The backend is Python with Flask and Flask-CORS, and inference runs on PyTorch, Torchvision, and Pillow. The main modules are:

- **Upload and Preprocessing:** Accepts JPEG, PNG, WEBP, and TIFF files under a size limit, computes the digest, and builds the normalized tensor.
- **Detection and Decision:** Loads both detectors once at startup, runs them on each request, and applies the rules of Eq. (2).
- **Metadata and Explainability:** Reads file properties and EXIF fields with Pillow and computes the Grad-CAM heatmap for the detector behind the decision.
- **Application Interface:** Shows the result, metadata, histograms, heatmaps, and a summary report, with a short form of the hash as the file ID. The temporary file is deleted once the analysis is done.

VII. PERFORMANCE EVALUATION AND RESULTS

We measure both classifiers with the usual confusion-matrix metrics, where TP, TN, FP, and FN are the counts of true positives, true negatives, false positives, and false negatives.

A. Evaluation Metrics

Accuracy is the fraction of samples the model gets right:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Precision asks how many positive predictions were correct, and recall how many real positives the model found:

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN} \tag{6}$$

F1 is their harmonic mean, and the false-positive rate (FPR) is how often a genuine image gets flagged by mistake:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}, \quad FPR = \frac{FP}{FP + TN} \tag{7}$$

In our application the FPR matters most, because it counts how often a real photo is wrongly accused.

B. Benchmark Results

Fig. 3 shows the confusion matrices on the held-out test sets, and Table IV collects the resulting scores. On the CIFAKE test set, the detector picked out 8,436 of

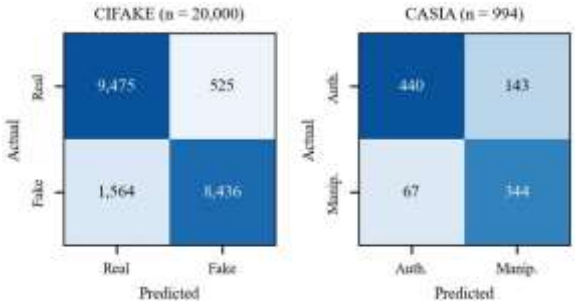


Fig. 3. Confusion matrices of the CIFAKE and CASIA detectors on their held-out test sets.

the 10,000 fakes and 9,475 of the 10,000 real images, for 89.56% accuracy. The CASIA detector found 344 of the 411 manipulated images and cleared 440 of the 583 authentic ones, for 78.87%. For CASIA, recall is clearly higher than precision. In other words, the model catches most edited images but also gives a fair number of false alarms.

C. Ẽxternal Image Evaluation

We also wanted to see how the models behave outside the benchmarks, so we ran both of them on 13 real photographs collected from outside sources. The CIFAKE detector got just 2 right; it called the other 11 generated. The CASIA detector passed 9 and wrongly flagged 4. The last two rows of Table IV give the rates worked out from these counts.

VIII. SPECIAL FOCUS: BENCHMARK GAP AND RESPONSIBLE USE

A. Benchmark Behavior Versus Real Photographs

On its own test set, the CIFAKE detector is fairly cautious: it misses more fakes (1,564) than it wrongly accuses real images (525). With outside photos the pattern reverses, and the FPR rises from 5.25% to 84.6%. The CASIA detector holds up better, at 24.53% on its test split and 30.8% outside, although 4 mistakes out of 13 are far too few to claim that the two rates match. We think this is mainly a domain gap rather than a basic flaw in the architecture. CIFAKE images have very low resolution, so the model may have learned resolution, compression, or texture patterns specific to that dataset instead of real signs of generation. Phone and camera photos differ from CIFAKE in exactly these ways. We have not yet tested this explanation.

Because the label depends on both detectors, a real image is only called Original if neither detector fires, so its chance of a false flag is at least the FPR of the less specific one. The rates in Table IV use each model's default decision rather than the engine's 0.80 and 0.70 thresholds, so the combined system still needs an end-to-end measurement.

B. What Each Label Means

Table V explains how each label is produced, and what it does and does not tell the user. The AI-generation rule is checked first, so an image that looks both synthetic and edited is labeled AI Generated, while its manipulation score stays visible among the detector signals.

TABLE V
DECISION OUTCOMES AND THEIR LIMITS

Outcome	Condition	Supports	Does not establish
AI Generated	$s_{ai} \geq 0.80$	Cues the CIFAKE model links to synthetic images	That the image was generated; the detector over-flagged genuine images
AI Edited / Manipulated	$s_{ai} < 0.80$ and $s_{man} \geq 0.70$	Cues the CASIA model links to tampering	That an edit exists, or that generative AI made it
Original	Neither threshold met	No model evidence above threshold	That the image is authentic or unedited

C. Human Review and Privacy

The system is meant to support a human reviewer and should not replace one. The reviewer should look at the detector signals, the heatmap, and the metadata together, count missing EXIF fields as neither good nor bad, and use the SHA-256 digest to make sure the analyzed file is the right one. A wrong authenticity label can damage a person's reputation. In our view, wrongly accusing a real photo is worse than missing a fake one. For that reason, the system should not be the only evidence behind decisions in journalism, law, education, or disciplinary cases. Uploaded files may also contain personal details, for example GPS coordinates. Our implementation deletes each temporary file after analysis. Still, anyone who runs the system beyond a local machine would need clear rules on data retention, access, and consent.

IX. LIMITATIONS AND OPEN RESEARCH CHALLENGES

A. Generalization to Real Photographs

On our small outside set, the CIFAKE detector raised many false alarms, which suggests that what it learned does not yet carry over to real photos from new sources. A bigger and more varied collection of real images, taken with many cameras, run through different editing workflows and compression levels, and set against many generative models, would help the detectors generalize and give the outside evaluation statistical weight. Controlled ablation experiments on resolution and recompression would also help to identify what causes the gap.

B. Calibration and End-to-End Evaluation

The thresholds were set by hand and need proper calibration on a separate validation set, and 13 real photos are not nearly enough to say how accurate the system will be once deployed. The combined system also needs an end-to-end evaluation.

C. Generative Editing and Missing Evidence

CASIA is closer to traditional manipulation than to modern generative editing, so the CASIA detector should not be sold as a detector of AI edits; detecting AI-assisted edits and localizing edited regions more precisely remain open problems. The system can also only work with what is left in the file: a re-encoded image, a screenshot, or a photo sent through a service that strips metadata may have little or no EXIF data. Finally, Grad-CAM shows what the model attended to, not pixel-level proof of tampering, and how good these explanations really are has yet to be studied.

X. CONCLUSION

We have described Trustify AI, a framework that checks whether a digital image is authentic and explains how it reached its answer. The framework combines two EfficientNet-B0 detectors with metadata inspection, a rule-based decision engine, and Grad-CAM explanations. The CIFAKE detector scored 89.56% on its benchmark and the CASIA detector 78.87% on a split free of leakage. The outside test, however, gave a very different picture. The AI-generation detector flagged 11 of 13 real photographs, which shows a large generalization gap. We take this as a strong reason to show several kinds of evidence and to report results openly, instead of relying on benchmark accuracy alone. The framework is a workable first step toward transparent authenticity checking, but it is not yet ready for high-stakes forensic use. Before that, it will need larger and more varied training data, calibrated thresholds, and independent validation.

REFERENCES

[1] I. Goodfellow et al., “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[2] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6840–6851.

[3] L. Verdoliva, “Media forensics and deepfakes: An overview,” *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 5, pp. 910–932, 2020.

[4] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, “CNN-generated images are surprisingly easy to spot… for now,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8695–8704.

[5] J. Dong, W. Wang, and T. Tan, “CASIA image tampering detection evaluation database,” in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process. (ChinaSIP)*, 2013, pp. 422–426.

[6] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[7] J. J. Bird and A. Lotfi, “CIFAKE: Image classification and explainable identification of AI-generated synthetic images,” *IEEE Access*, vol. 12, pp. 15642–15650, 2024.

[8] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.