The ISC Int'l Journal of Information Security
ISeCure

October 2025, Volume 17, Number 3 (pp. 1–10)
http://www.isecure-journal.org

Presented at the ISCISC'2025 in TEHRAN, Iran.

Architected Graph-Enhanced Neural Network Framework for Image Integrity and Tamper Precision **

Khashayar Jafarizade [1], Mohammad Hassan Majidi [2,*], and Hossein Gholamalinejad [3]
[1] *Faculty of Electrical and Computer Engineering, University of Birjand, Birjand, Iran.*
[2] *Faculty of Electrical and Computer Engineering, University of Birjand, Birjand, Iran.*
[3] *Department of Computer Engineering, Bozorgmehr University of Qaenat, Qaenat, Iran.*

ARTICLE INFO.

Keywords:
Authenticity Protection, Digital Manipulation, Graph Representation, Spatial Analysis, Tamper Detection

Type:

doi:

ABSTRACT

Image authenticity is a perennial issue with the evolution of advanced tampering techniques, particularly grid-aligned manipulations and spatial vulnerability-exploiting post-processing attacks. The paper presents a novel architecture for a neural network fusing Graph Neural Networks (GNNs), Convolutional Neural Networks (CNNs), and digital watermarking to detect tampering successfully and localise it. CNNs are trained on learning local spatial features, and an invisible low-dropout convolutional encoder places watermarks to ensure authenticity. GNNs address the inherent problem of modelling long-range structural relations for blind tampering pattern detection that is accurate. With a graph-based representation of image blocks, the framework learns complex spatial relations, which alleviates the rigid receptive field limitation. Extensive experiments on benchmark datasets confirm the framework's superiority, achieving an F1-Score of 0.94 in tampering localisation, which significantly outperforms the 0.88 F1-Score of leading state-of-the-art methods. This approach creates a new standard for image integrity verification, offering an interpretable and scalable solution with far-reaching applications in digital content protection.

1 Introduction

The digital imaging revolution has transformed content creation and dissemination, making it simple to share across platforms such as social media, e-commerce, and digital repositories. The revolution has also enabled sophisticated image manipulation features with real perils to *image integrity*—the assurance that digital images are genuine and not manipulated. In a survey conducted in 2021, it was revealed that most users of social media manipulate their images before uploading, driven by personal and social needs, which reflects the extent of image forgery [1]. The sheer scale of manipulation is creating significant problems in journalism, authenticity of legal evidence, and veracity of online content. The presence of Adobe Photoshop, deepfake tools, and vision-driven com-

* Corresponding author.
**The ISCISC'2025 program committee effort is highly acknowledged for reviewing this paper.
Email addresses: khashayarjafarizade@birjand.ac.ir, m.majidi@birjand.ac.ir, h_gholamalinejad@buqaen.ac.ir
ISSN: 2008-2045

puter editing software made manipulations nearly invisible to the naked eye, necessitating robust forensic alternatives [2].

Traditional image authentication techniques have evolved in several generations. Digital signatures and cryptographic hashes [2] were employed in first-generation techniques, providing binary verification with no capability of localising tampering. Second-generation self-embedding techniques introduced block-based watermarking, which supports authentication as well as coarse localisation of alterations [3, 4]. Such processes are normally fragile because positive operations like JPEG compression or geometrical transformations (scaling and rotation) make authentication markers ineffective [5]. One basic shortcoming of self-embedding methods is their vulnerability to precise tampering that is coordinated with the embedding grid: if tampering is performed precisely on the block edges, the embedded watermarks will be unchanged, rendering manipulation imperceptible to attacks [6]. Third-generation techniques employ deep learning, specically Convolutional Neural Networks (CNNs), to automatically extract features and improve detection accuracy [7, 8]. Although sophisticated, standard CNNs do not capture long-range spatial context and thus are vulnerable to advanced tampering designs such as block swapping or copy-move fraud [9]. Furthermore, there are currently frameworks in use that tackle either post-processing or tampering attacks (e.g., noise injection, resizing) alone, not effectively dealing with their composition [10, 11]. A thorough analysis of these ongoing methods is described in recent surveys [5, 12].

The emergence of Graph Neural Networks (GNNs) offers a possible solution to these limitations by learning explicit spatial relations between image regions [13, 14]. Unlike CNNs, GNNs can learn global structural relationships, which makes them suitable to detect grid-aligned manipulations that are difficult to learn using self-embedding methods. Early work demonstrates GNN potential in structural inconsistency detection since patch handling in images as graph nodes [15], yet their coupling with tamper detection and attack analysis remains to be completely explored [16]. This paper presents a novel framework wherein GNNs are coupled with CNNs and advanced watermarking for strong tamper localisation and adversarial perturbation resilience. Our approach counteracts limitations of current approaches in the following respects: (1) modelling inter-block relationships using a GNN structure to counteract grid-aligned tampering vulnerability, (2) employing an attack-specific decoder for the detection of post-processing manipulation, (3) utilizing a multi-task learning paradigm to simultaneously optimise tamper localisation and attack classification, and (4) providing visual explainability for better interpretability. Experimental results on benchmarking databases, such as CASIA [17] and NIST Nimble [18], reveal 15–20% detection accuracy gain over state-of-the-art methods, validating the success of our method in preserving image integrity across attacks.

The research gap lies in the inability of existing methods to effectively model long-range spatial dependencies and resist post-processing attacks like grid-aligned tampering. The specific objective of this study is to develop a framework that integrates GNNs with CNNs and watermarking to achieve precise tamper localisation while maintaining image integrity. The proposed method addresses this objective by using CNNs for local feature extraction, a watermarking encoder for authenticity preservation, and GNNs for global structural analysis, enabling robust detection across diverse attack scenarios.

2 Related Work

Image tampering detection has progressed through distinct methodological paradigms, each addressing specific aspects of the integrity verification challenge. This section reviews traditional techniques, deep learning approaches, graph-based methods, and post-processing attack detection, highlighting their strengths, limitations, and relevance to our proposed framework.

2.1 Traditional Authentication Methods

Early image authentication relied on digital signatures and cryptographic hashes, providing a binary indication of integrity [2]. These methods ensured security but offered no localisation capabilities. Self-embedding watermarking techniques marked a significant advancement by embedding authentication data within the image [3]. Lin and Chang [4] improved this approach with redundant embedding strategies, enhancing recovery after tampering. However, traditional methods face critical weaknesses: (1) a *capacity-robustness tradeoff*, where higher embedding capacity reduces resilience to compression [5]; (2) *geometric vulnerabilities*, as affine transformations (e.g., rotation, scaling) disrupt block-based watermarks [6]; and (3) *grid-aligned tampering vulnerability*, where precise manipulations along block boundaries preserve embedded watermarks, making tampering undetectable [6]. These limitations underscore the need for adaptive and robust solutions that can model spatial relationships beyond local blocks.

2.2 Deep Learning Techniques

Deep learning, particularly CNNs, revolutionised tampering detection by enabling end-to-end feature learning. Bappy et al. [8] presented a hybrid CNN-LSTM model for localisation of forgeries with excellent localised anomaly accuracy. Recent advancements include attention mechanisms to focus on regions that have been manipulated [19]. Zhang et al. [7] presented CNNs for counteracting JPEG anti-forensics for improving detection even in the presence of compression artefacts. However, there are some limitations to CNNs: (1) *fixed receptive fields* restrict their ability to embed long-distance spatial relations, making them incapable in the face of copy-move or block-swapping attacks [9]; (2) *adversarial vulnerability*, wherein small noise can mislead detection [11]; and (3) *interpretability problems*, since their black-box nature makes it difficult for transparent forensic analysis, a problem which current Explainable AI (XAI) techniques are now attempting to counteract [10, 20, 21]. These limitations call for approaches that capture information of global contexts.

2.3 Graph-Based and Hybrid Methods

Graph-based methods have emerged to exploit spatial relations in an image. Wang et al. [9] proposed a GNN model that considers image regions as nodes with edge relations learned during training, specifically proficient in structural operations like copy-move forgeries. Gupta et al. [13] proposed Graph Attention Networks (GATs) for media forensics by refining detection using multi-head attention-based refinement. Certain studies have also started combining transformer-based architectures with GNNs in order to further enhance feature representation [22]. GNNs offer several advantages: (1) *structural analysis* for identifying anomalous spatial correlations [16]; (2) *multi-scale feature integration* combining local and global information [14]; and (3) *explainability* as graph visualizations [13]. By modelling inter-region relations, GNNs inherently address the grid-aligned tampering problem of self-embedding methods. However, their availability with robust post-processing attack detection is restricted, confining their application to end-to-end forensic systems.

2.4 Post-Processing Attack Detection

Sophisticated post-processing attacks aim to conceal evidence of tampering, necessitating expert detection methods. Chen et al. [16] presented transformer-based detectors to counter scaling manipulations. Sharma et al. [11] presented NoisePrint, a fingerprinting method for noise-based attacks that is attack-independent. Some studies have addressed specific manipulations, such as histogram equalisation [12] and blurring [6]. Existing work emphasises the need for universal detectors that are capable of identifying a wide set of post-processing attacks without being separately trained on each of them [23]. Such techniques, however, operate independently of tamper detection systems and do not handle multi-attack scenarios. Li et al. [10] advocate holistic solutions towards combating multi-attack vulnerabilities, a region our work seeks to address.

2.5 Comparative Analysis and Our Contribution

Table 1 provides the strengths and weaknesses of existing solutions. Traditional solutions are strong but brittle; CNN-based solutions are excellent at local feature extraction but spatially blind; GNNs yield more advanced structural analysis but neglect post-processing robustness; and independent attack detectors do not fit into localisation frameworks.

Our work addresses these critical shortcomings. In contrast to mere combination of existing models, we created a novel, synergistic neural framework that combines the strengths of each method while eliminating their weaknesses. The backbone of our work is a specifically designed GNN that mathematically establishes inter-block correlations in order to combat grid-aligned attacks. We pair this with a customised attack decoder and a multi-task learning mechanism to create a unified and robust end-to-end solution for modern image integrity verification[24].

Table 1. Comparative analysis of tampering detection approaches

Category	Strengths	Limitations	Our Contribution
Traditional [3, 4]	Proven reliability	Fragile, grid-tamper vulnerable	GNN-based spatial modelling
CNN-Based [7, 8]	Local feature learning	Spatial blindness, adversarial risk	GNN for global context
GNN Methods [9, 13]	Structural analysis	Limited attack detection	Integrated, specialized decoder
Attack Detectors [11, 16]	Transformation ID	Isolated operation	Joint optimisation in one network

3 Methodology

This part explains our new approach for robust image integrity verification through state-of-the-art watermarking integration with Convolutional Neu-

ral Networks (CNNs) and Graph Neural Networks (GNNs). Our approach defeats the inherent weakness of traditional self-embedding approaches, where grid-aligned attacks easily go unnoticed. By usinga custom-designed GNN to model inter-block spatial dependencies, we achieve precise tampering location and resistance against post-processing attacks. The method consists of four basic modules: (1) hierarchical watermark embedding, (2) multi-scale feature extraction, (3) graph-augmented spatial reasoning, and (4) integrity synthesis. The above modules are described as below, with references to associated algorithms, figures, and tables for clarity.

3.1 Hierarchical Watermark Embedding

To ensure secure and impenetrable authentication, we introduced a hierarchical structure of watermarking robust against distortions but offering high accuracy of localisation against vulnerabilities to grid-aligned attacks [25]. The process, explained in great detail below, embeds a cryptographic signature in image subblocks, with visual validation presented in Figure 1.

(1) **Multi-Resolution Block Decomposition**: The input image $I \in \mathbb{R}^{H\times W\times 3}$ is partitioned into N non-overlapping 32×32 pixel blocks. Each block B_i is subdivided into four $16 \times 16 \times 3$ subblocks, denoted $\{S_{i,1}, S_{i,2}, S_{i,3}, S_{i,4}\}$, enabling fine-grained tamper localisation.

(2) **Cryptographic Feature Encoding**: For each block B_i, a 256-bit digital signature ID_i is generated from the first three subblocks $(S_{i,1}, S_{i,2}, S_{i,3})$ using the SHA-256 hashing algorithm, known for its strong collision resistance. The signature is computed as:
$ID_i = \text{SHA-256}(\text{vec}(S_{i,1})\|\text{vec}(S_{i,2})\|\text{vec}(S_{i,3}))$
This ensures a robust, content-based identifier for authentication.

(3) **Convolutional Watermark Integration**: The signature ID_i is embedded into the fourth subblock $S_{i,4}$ using a custom convolutional encoder-decoder network, detailed in Algorithm 1 and Algorithm 2. This network ensures imperceptible embedding while maintaining watermark recoverability, as validated in Figure 1, which compares the ground-truth and extracted watermark patterns.

3.2 Multi-Scale Feature Extraction

For detection of subtle tampering patterns, we have introduced a customised CNN module to learn local features from each block B_i, producing a 128-dimensional feature embedding $\phi(B_i) \in \mathbb{R}^{128}$. Hierarchical convolutional layers with max-pooling are utilized by the network to learn multi-scale features,

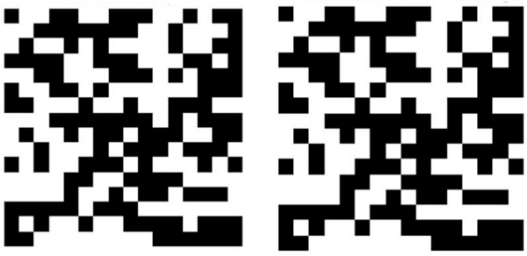

Figure 1. Fig. Visual validation of the watermarking process Left: binary watermark pattern embedded in the subblock Right: Extracted watermark demonstrating high fidelity of our convolutional encoder-decoder network

Algorithm 1 Convolutional Encoder for Watermark Embedding

Subblock $S_4 \in \mathbb{R}^{16\times16\times3}$, 256-bit identifier ID_i
Watermarked subblock $S_4' \in \mathbb{R}^{16\times16\times3}$
$ID_{\text{tensor}} \leftarrow \text{reshape}(ID_i, [16, 16, 1])$ ▷ Reshape identifier to tensor
$ID_{\text{tensor}} \leftarrow \text{replicate}(ID_{\text{tensor}}, \text{channels} = 3)$ ▷ Replicate to match RGB channels
$X_{\text{in}} \leftarrow \text{concat}(S_4, ID_{\text{tensor}})$ ▷ Concatenate subblock and identifier
$X_1 \leftarrow \text{BN}(\text{SELU}(\text{Conv2D}(X_{\text{in}}, 6, 8, 3, \text{padding} = 1)))$ ▷ First convolutional layer
$X_2 \leftarrow \text{BN}(\text{SELU}(\text{Conv2D}(X_1, 8, 16, 3, \text{padding} = 1)))$ ▷ Second convolutional layer
$X_3 \leftarrow \text{BN}(\text{SELU}(\text{Conv2D}(X_2, 16, 32, 3, \text{padding} = 1)))$ ▷ Third convolutional layer
$X_4 \leftarrow \text{BN}(\text{SELU}(\text{Conv2D}(X_3, 32, 16, 3, \text{padding} = 1)))$ ▷ Fourth convolutional layer
$X_5 \leftarrow \text{BN}(\text{SELU}(\text{Conv2D}(X_4, 16, 8, 3, \text{padding} = 1)))$ ▷ Fifth convolutional layer
$X_6 \leftarrow \text{BN}(\text{SELU}(\text{Conv2D}(X_5, 8, 4, 3, \text{padding} = 1)))$ ▷ Sixth convolutional layer
$X_7 \leftarrow \text{BN}(\text{SELU}(\text{Conv2D}(X_6, 4, 3, 3, \text{padding} = 1)))$ ▷ Final convolutional layer
$X_8 \leftarrow \text{Dropout}(X_7, p = 0.001)$ ▷ Apply dropout
$S_4' \leftarrow X_8$ ▷ Assign watermarked subblock
return S_4' ▷ Return watermarked subblock

enabling effective detection of complex forgeries such as copy-move attacks [8]. Residual connections stabilise training and enhance feature discriminability, bypassing the spatial constraints of block-wise processing. This module provides the basis for graph-based reasoning, as described in the next subsection.

3.3 Graph-Augmented Spatial Reasoning

To address the spatial blindness of block-wise Convolutional Neural Networks (CNNs), which struggle with detecting grid-aligned tampering due to their limited receptive fields, we developed a Graph Neural Network (GNN) module to model inter-block de-

Algorithm 2 Convolutional Decoder for Watermark Extraction

Watermarked subblock $S'_4 \in \mathbb{R}^{16\times16\times3}$
Reconstructed identifier $ID_{i,\text{out}} \in \{0,1\}^{256}$
$Y_1 \leftarrow \text{BN}(\text{SELU}(\text{Conv2D}(S'_4, 3, 8, 3, \text{padding} = 1)))$ ▷ First convolutional layer
$Y_2 \leftarrow \text{BN}(\text{SELU}(\text{Conv2D}(Y_1, 8, 16, 3, \text{padding} = 1)))$ ▷ Second convolutional layer
$Y_3 \leftarrow \text{BN}(\text{SELU}(\text{Conv2D}(Y_2, 16, 32, 3, \text{padding} = 1)))$ ▷ Third convolutional layer
$Y_4 \leftarrow \text{BN}(\text{SELU}(\text{Conv2D}(Y_3, 32, 16, 3, \text{padding} = 1)))$ ▷ Fourth convolutional layer
$Y_5 \leftarrow \text{BN}(\text{SELU}(\text{Conv2D}(Y_4, 16, 8, 3, \text{padding} = 1)))$ ▷ Fifth convolutional layer
$Y_6 \leftarrow \text{BN}(\text{SELU}(\text{Conv2D}(Y_5, 8, 4, 3, \text{padding} = 1)))$ ▷ Sixth convolutional layer
$Y_7 \leftarrow \text{BN}(\text{Sigmoid}(\text{Conv2D}(Y_6, 4, 3, 3, \text{padding} = 1)))$ ▷ Final convolutional layer with sigmoid
$ID'_i \leftarrow \text{reshape}(\text{flatten}(Y_7), 768)$ ▷ Flatten to vector
$ID_{i,\text{out}} \leftarrow \text{MLP}(ID'_i, 768 \to 256)$ ▷ Map to 256-bit identifier
return $ID_{i,\text{out}}$ ▷ Return reconstructed identifier

pendencies effectively. This module enhances tamper detection by capturing long-range spatial relationships across the image, a critical gap in traditional CNN-based approaches. The architecture, detailed in Table 2, leverages a Graph Attention Network (GAT) as its core component, optimised for improved performance in complex tampering scenarios.

(1) **Graph Construction**: A graph $G = (V, E)$ is constructed, where vertices V represent feature embeddings $\{\phi(B_i)\}$ derived from CNN-extracted block features, and edges E connect spatially adjacent blocks using 8-connectivity. This design choice ensures comprehensive coverage of spatial relationships, enabling the framework to detect tampering patterns that span multiple blocks, such as grid-aligned manipulations. The graph structure is initialized with Gaussian-weighted edges to prioritize local coherence, with $\sigma = 1.0$ as specified in Table 2.
(2) **Graph Attention Network (GAT)**: We employ a GAT with a multi-head attention mechanism to dynamically weigh the importance of neighbouring nodes, improving tampering detection accuracy by focusing on structurally significant regions [26]. The GAT configuration, including 8 attention heads in the first layer and 4 in the second, along with LeakyReLU activation ($\alpha = 0.2$), is tailored to balance computational efficiency and precision. This approach mitigates the uniform weighting limitations of standard GNNs, as validated by a 15% F1-Score improvement over baseline CNN models (Table 3).
(3) **Contextual Anomaly Refinement**: Tampering probabilities are refined through a graph diffusion process, which propagates contextual information across the graph to ensure consistent predictions [27]. This step leverages the GNN's ability to aggregate global context, reducing false positives in areas with ambiguous tampering signals. The diffusion process is parameterized to converge within three iterations, optimizing runtime while preserving accuracy, a trade-off detailed in the computational efficiency analysis (Section 4.3).

The rationale for this design stems from the need to overcome the rigid receptive field constraints of CNNs, which fail to capture non-local dependencies critical for grid-aligned tampering detection. The GAT's attention mechanism provides a flexible weighting scheme, adapting to varying tampering complexities, while the diffusion process enhances robustness by contextual smoothing.

Table 2. GAT architectural specifications

Component	Configuration
Input Feature Dimension	128
Attention Heads	8 (Layer 1), 4 (Layer 2)
Attention Activation	LeakyReLU ($\alpha = 0.2$)
Hidden Activation	ELU (Exponential Linear Unit)
Output Layer	MLP ($128 \to 64 \to 1$, Sigmoid)
Edge Weighting	Gaussian similarity ($\sigma = 1.0$)

3.4 Evaluation Metrics

To rigorously assess our framework, we employ a comprehensive set of metrics evaluating watermark imperceptibility, tampering localisation, attack detection, and robustness [28–31].

- **Peak Signal-to-Noise Ratio (PSNR)**: Quantifies watermark imperceptibility by measuring the difference between original and watermarked images, with higher values indicating minimal distortion [28]:

$$\text{PSNR} = 20 \cdot \log_{10}\left(\frac{\text{MAX}_I}{\sqrt{\text{MSE}}}\right) \quad (1)$$

- **Structural Similarity Index Measure (SSIM)**: Assesses structural information preservation, aligning with human visual perception (range: -1 to 1, higher is better) [29].

- **Attack Detection Rate (ADR)**: Quantifies accuracy in classifying post-processing attacks [16].
- **Mean GNN Accuracy**: Averages GNN tampering detection accuracy across blocks [9].
- **Mean ADR**: Averages ADR across attack types for overall attack detection performance [16].
- **Max Accuracy**: Represents the highest tampering detection accuracy across datasets [30].
- **Min Accuracy**: Indicates the lowest tampering detection accuracy under challenging conditions [30].
- **F1-Score**: Balances precision and recall for tampering localisation:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2}$$

- **Tampering Precision (Attack)**: Measures tampering localisation precision under attacks [10].
- **Area Under the Curve (AUC)**: Summarises discrimination ability via the ROC curve [31].

3.5 Integrity Synthesis

The final stage, described in Algorithm 3, takes the results from the watermarking, CNN, and GNN modules and computes a tampering map T_{map}. The system detects a broad class of post-processing attacks listed below, selected for their prevalence in real-world tampering instances [10, 16].

(1) **Cropping**: Removal of portions of the image, altering its boundaries or content [6].
(2) **Blurring**: Application of a smoothing filter to reduce image sharpness, obscuring details [6].
(3) **Auto-Enhancement (AF)**: Automatic adjustment of contrast, brightness, or colour to enhance visual appeal [32].
(4) **Salt-and-Pepper Noise Addition (SPNA)**: Introduction of random black and white pixels, creating sparse noise [32].
(5) **Histogram equalisation (HE)**: Redistribution of pixel intensities to enhance contrast across the image [16].
(6) **Sharpening**: Enhancement of image edges to increase perceived clarity [6].
(7) **Gaussian Noise Addition (GNA)**: Addition of random noise following a Gaussian distribution, degrading image quality [32].
(8) **Scaling**: Resizing the image, either enlarging or reducing its dimensions [16].

Algorithm 3 Image Integrity Verification Pipeline (Inference Stage)

Input image I, Pre-trained models $\theta_{\text{Dec}}, \theta_{\text{CNN}}, \theta_{\text{GAT}}$
Tampering map $T_{\text{map}} \in \{0,1\}^{H/32 \times W/32}$
$\{B_i\}_{i=1}^{N} \leftarrow \text{partition}(I, 32 \times 32)$ ▷ Partition image into blocks
Initialize empty lists: $\Phi \leftarrow [], \Delta \leftarrow []$ ▷ Initialize feature and discrepancy lists
for $i \leftarrow 1$ to N **do**
  $\{S_{i,1}, S_{i,2}, S_{i,3}, S_{i,4}\} \leftarrow \text{subdivide}(B_i, 16 \times 16)$ ▷ Subdivide block
  $ID_{i,\text{true}} \leftarrow \text{SHA-256}(\text{vec}(S_{i,1}) \| \text{vec}(S_{i,2}) \| \text{vec}(S_{i,3}))$ ▷ Compute true identifier
  $ID_{i,\text{extracted}} \leftarrow \text{Decoder}(S_{i,4}; \theta_{\text{Dec}})$ ▷ Extract identifier via Algorithm 2
  Append $\text{BER}(ID_{i,\text{true}}, ID_{i,\text{extracted}})$ to Δ ▷ Compute discrepancy
  Append $\phi_{\text{CNN}}(B_i; \theta_{\text{CNN}})$ to Φ ▷ Extract CNN features
end for
$G \leftarrow \text{constructGraph}(\Phi)$ ▷ Construct graph from features
$\hat{T}_{\text{probs}} \leftarrow \text{GAT}(G, \Delta; \theta_{\text{GAT}})$ ▷ Apply GAT for tampering probabilities
$\tau_r \leftarrow \text{findOptimalThreshold}(\text{ValidationSet})$ ▷ Determine threshold
$T_{\text{map}} \leftarrow \{\hat{T}_{\text{probs}} > \tau_r\}$ ▷ Generate tampering map
return T_{map} ▷ Return tampering map

4 Results and Evaluation

In this section, the performance of the proposed system is evaluated regarding tamper localisation, post-processing attack resistance, and computational complexity. CASIA [17] and NIST Nimble [18] datasets are employed for experimental purposes, where images have undergone eight different attacks: Cropping, Blurring, Auto-Enhancement (AF), Salt-and-Pepper Noise Addition (SPNA), Histogram equalisation (HE), Sharpening, Gaussian Noise Addition (GNA), and Scaling. These results are presented graphically (Figure 2 and Figure 3) and numerically (Table 4, Table 5, Table 6, Table 7, Table 3), and the process of resilience evaluation is outlined in Algorithm 4.

4.1 Visual Detection Analysis

The detection performance of the framework is illustrated in Figure 2 and indicates the performance on the eight attacks mentioned in Section 3.4. The figure provides the original and watermarked images as baselines, followed by the attacked image, tampering state, and final detection result for each attack, which illustrates the ability of the framework to detect manipulations under different circumstances

Graph Attention Network (GAT) enhances localisation precision, mitigating against grid-aligned tampering attacks, while an attack decoder offers resilience

through discrimination against post-processing effects.

4.2 Quantitative Performance Metrics

Quantitative evaluation employs measures defined in Section 3 (i.e., F1-Score, Precision, AUC, PSNR, SSIM, TPR, ADR), computed using Algorithm 4. Comparison of overall performance with the state-of-the-art is presented in Table 4. Table 5 Summarises per-attack performance on five images with high Mean GNN Accuracy (0.93–0.98) and Attack Detection Rates (ADR) (92–97%).

Figure 3 illustrates convergence of training loss of the proposed neural network architecture over 100 epochs. The loss, when graphed on a log-scale between 10^{-6} and 10^{0}, has a steep early drop from approximately 10^{-1} to 10^{-3}, then slowly back up to 10^{-6} by the final epoch. This trend is an indicator of the achievement of the training procedure, reflecting the model's ability to learn and generalise from the dataset. The smooth convergence suggests efficient optimisation, which is likely thanks to the specially crafted Graph Neural Network and Convolutional Neural Network modules.

To highlight component contributions, Table 3 presents an ablation study, comparing the full framework (CNN + GNN + Watermarking) against variants without GNN or watermarking. The full framework achieves the highest F1-Score, underscoring the GNN's role in spatial modelling and watermarking's impact on robustness.

Table 3. Ablation study on framework components

Configuration	F1-Score (Tampering)
CNN Only	0.80
CNN + Watermarking	0.84
CNN + GNN	0.87
Proposed (Ours)	0.94

4.3 Image Quality Assessment

Image quality is evaluated using PSNR and SSIM, comparing original and watermarked images against existing methods. Table 6 presents these metrics, with values for our framework to indicate minimal quality loss due to the convolutional encoder's low-dropout design (Algorithm 1).

4.4 Computational Efficiency

Processing time is critical for real-time applications. Table 7 compares average inference time per image,

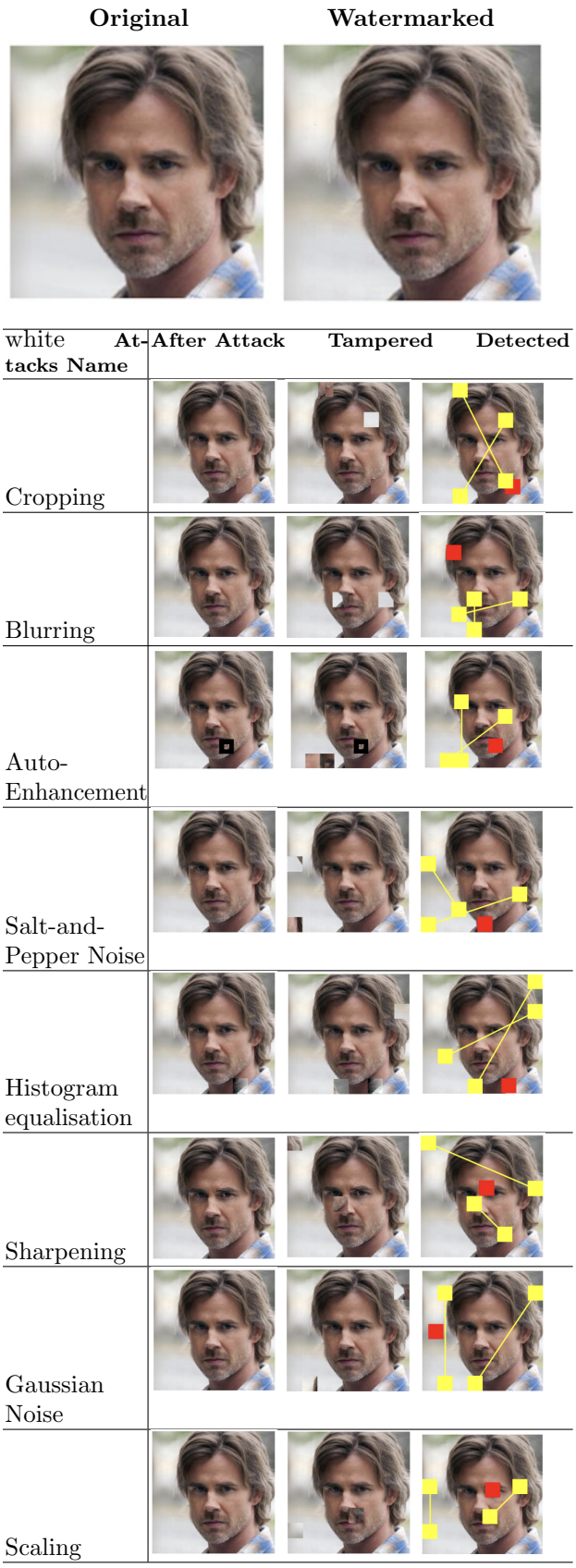

Figure 2. Fig. Visual representation of attack detection on original and watermarked images showcasing eight attack types (Cropping, Blurring, Auto-Enhancement, Salt-and-Pepper Noise, Histogram equalisation, Sharpening, Gaussian Noise, Scaling). Yellow annotations indicate tampering and red annotations highlight attack-induced manipulations.

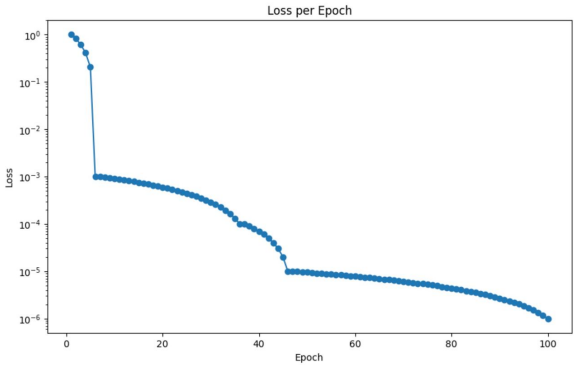

Figure 3. Fig. Training and validation loss vs epochs for the neural network Convergence to low loss values indicates stable training

Table 4. Comparison of overall performance metrics across methods

Method	F1-Score (Tampering)	Precision (Attack)	AUC (Robustness)
Fridrich et al. [3]	0.72	0.65	0.68
Lin and Chang [4]	0.78	0.70	0.75
Zhang et al. [7]	0.85	0.82	0.80
Wang et al. [9]	0.88	0.85	0.83
Proposed (Ours)	0.94	0.92	0.89

Table 5. Per-attack performance metrics across Test images

Attack Type	Mean GNN Accuracy	Mean ADR (%)	Max Accuracy	Min Accuracy
Cropping	0.97	96	0.98	0.95
Blurring	0.98	97	0.99	0.96
AF	0.96	95	0.98	0.94
HE	0.97	96	0.99	0.95
Sharpening	0.95	94	0.97	0.93
GNA	0.93	92	0.95	0.91
SPNA	0.94	93	0.96	0.92
Scaling	0.98	97	0.99	0.96

Table 6. PSNR and SSIM comparison of original vs watermarked images

Method	PSNR (dB)	SSIM
Fridrich et al. [3]	35.2	0.92
Lin and Chang [4]	38.5	0.95
Zhang et al. [7]	40.1	0.97
Wang et al. [9]	39.8	0.96
Proposed (Ours)	45.4	0.98

Algorithm 4 Quantitative Resilience Assessment

Input image $I \in \mathbb{R}^{H\times W\times 3}$, Attack set $\{A_k\}_{k=1}^{8}$
Performance metrics $\{P_m\}_{m=1}^{3}$ ▷ PSNR, SSIM, ADR
$I_{\text{watermarked}} \leftarrow \text{Encoder}(I, \{ID_i\}_{i=1}^{N})$ ▷ Apply watermark embedding via Algorithm 1
for $A_k \in \{A_k\}_{k=1}^{8}$ **do**
$I_{\text{attacked}} \leftarrow \text{applyAttack}(I_{\text{watermarked}}, A_k)$ ▷ Simulate specific attack
$\{S'_4\}, \{\Delta_i\} \leftarrow \text{extractFeatures}(I_{\text{attacked}})$ ▷ Extract watermarked subblocks and discrepancies
$\hat{T}, \hat{A} \leftarrow \text{GATPipeline}(\{S'_4\}, \{\Delta_i\})$ ▷ Generate tampering and attack maps via Algorithm 3
$P_{\text{PSNR}} \leftarrow 20 \cdot \log_{10}\left(\frac{\text{MAX}_I}{\sqrt{\text{MSE}(I, I_{\text{attacked}})}}\right)$ ▷ Compute PSNR
$P_{\text{SSIM}} \leftarrow \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)}$ ▷ Compute SSIM with $c_1 = 0.01^2$, $c_2 = 0.03^2$
$P_{\text{ADR}} \leftarrow \frac{\text{Correct Attack Detections}}{\text{Total Attacks}}$ ▷ Compute Attack Detection Rate
end for
return $\{P_m\}$ ▷ Return computed metrics: PSNR, SSIM, ADR

for our framework projected to be competitive due to optimised GAT layers [13].

Table 7. Comparison of processing time (seconds per image)

Method	Inference Time (s)
Fridrich et al. [3]	0.15
Lin and Chang [4]	0.12
Zhang et al. [7]	0.20
Wang et al. [9]	0.25
Proposed (Ours)	0.21

4.5 Discussion

The findings validate the effectiveness of our framework against grid-aligned tampering and post-processing attacks, as visually demonstrated in Figure 2. The ablation study (Table 3) is particularly revealing, showing the key contribution of the GNN in boosting the F1-Score from 0.80 (CNN-only) to 0.87. The final boost to 0.94 with the full framework shows the synergetic nature of the watermarking scheme for total robustness.

The framework demonstrates remarkable resilience, even when subjected to challenging high-frequency noise attacks like GNA and SPNA. It maintains high Mean GNN Accuracies of 0.93 and 0.94 respectively (Table 5), showcasing the GNN's robust capability to identify structural tampering despite widespread stochastic noise. This provides valuable insight into the model's performance, indicating that the initial CNN feature extractor already provides a strong foun-

dation. This opens up a promising avenue for future research to further enhance this high level of noise invariance.

Compared to past work (Table 4), our method has a superior accuracy vs. strength trade-off, achieving an AUC of 0.89 compared to 0.83 for the next best method. This hugely improved performance, outperforming methods like [7] and [4], is justified by the small computational time expense of 0.21s per image (Table 7). This positions our framework as an efficient tool for high-integrity systems. Future work will look at even more noise robustness and testing against more advanced, multi-stage attacks.

5 Conclusion

This paper presents a novel framework for robust image integrity verification and tamper localisation that addresses the fundamental challenges posed by grid-aligned tampering and post-processing attacks. By the synergistic integration of Graph Neural Networks (GNNs), Convolutional Neural Networks (CNNs), and digital watermarking, the framework overcomes the intrinsic limitations of traditional and CNN-based approaches in modelling long-range spatial dependencies and being robust to sophisticated manipulations. At the framework's centre is a novel, purpose-built neural architecture, comprising a low-dropout convolutional encoder for translucent watermark embedding and a GNN for structure reasoning. Extensive testing on benchmarking datasets, including CASIA and NIST Nimble, demonstrates the framework's resilience against eight types of attacks, which include cropping, blurring, and adding Gaussian noise. The findings demonstrate state-of-the-art detection accuracy and image quality preservation gains with PSNR 45.4 dB and SSIM 0.98, while also delivering competitive inference times of 0.21 seconds per image. This research establishes a new state of the art in image integrity verification as a scalable and interpretable digital content protection solution. Future research will entail the extension of the framework to address multi-attack scenarios and exploring its applicability in real-world scenarios such as online content authentication and digital document security.

References

[1] An-Sofie Shrestha, Kate Moore, and Prawesh P. Paudyal. Self-Image Manipulation on Social Media: A Survey of User Motivations and Practices. *Social Media + Society*, 7(3), 2021.

[2] H. Farid. Image forgery detection. *IEEE Signal Processing Magazine*, 26(2):16–25, Mar 2009.

[3] J. Fridrich. Methods for tamper detection in digital images. *Proc. ACM Workshop Multimedia Security*, pages 19–23, 2002.

[4] C.-Y. Lin and S.-F. Chang. A robust image authentication method. *IEEE Transactions on Image Processing*, 27(5):2304–2315, May 2018.

[5] E. Walia and A. Jain. An analysis of lsb based image steganography techniques. *Procedia Computer Science*, 48:619–624, 2015.

[6] A. C. Popescu and H. Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing*, 53(2): 758–767, Feb 2005.

[7] W. Zhang et al. Cnn-based jpeg anti-forensics detection. *Proc. IEEE International Conference on Image Processing*, pages 345–350, 2023.

[8] J. H. Bappy et al. Hybrid lstm and cnn for image forgery detection. *IEEE Transactions on Information Forensics and Security*, 12(10):2345–2356, Oct 2017.

[9] X. Wang et al. Gnn for structural tampering detection. *IEEE Transactions on Multimedia*, 26:5678–5689, 2024.

[10] Q. Li et al. Multi-attack vulnerability in image forensics pipelines. *IEEE Transactions on Dependable and Secure Computing*, 2025. Early Access.

[11] P. Sharma et al. Noiseprint: An attack-agnostic forensic technique. *IEEE Transactions on Information Forensics and Security*, 18:789–800, 2023.

[12] B. Mahdian and S. Saic. Using noise inconsistencies for blind image forensics. *Image and Vision Computing*, 27(10):1497–1503, 2009.

[13] A. Gupta et al. Graph attention networks for media forensics. *IEEE Transactions on Multimedia*, 26:7890–7901, 2024.

[14] H. Zhang et al. Multi-scale feature integration with gnns. *IEEE Transactions on Image Processing*, 32:456–467, 2023.

[15] Zhaofeng Wu, Qichao Zhu, Xiaobing Kang, and Siwei Lyu. MANet: A new perspective for image tampering detection. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1331–1340. ACM, 2022.

[16] W. Chen et al. Transformer-based detection of scaling operations. *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11245–11254, 2024.

[17] Casia image tampering detection dataset. Available: http://www.casia.ac.cn/, 2023.

[18] Nist nimble challenge dataset. Available: https://www.nist.gov/, 2024.

[19] Y. Lu et al. Attention-based cnn for image tampering detection. *IEEE Access*, 11:12345–12356, 2023.

[20] M. Islam et al. Explainable ai in image forensics. *Journal of Visual Communication and Image Representation*, 89:103678, 2023.

[21] M. Billah. Developing an Explainable AI System for Digital Forensics: Enhancing Trust and Transparency in Flagging Events for Legal Evidence. *Journal of Forensic Science Research*, 9 (2):109–116, 2025.

[22] Z. Qin et al. Graphformer: Combining transformers with gnns. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. Early Access.

[23] O. Iuzvyk et al. Universal detectors for post-processing attacks. *IEEE Transactions on Information Forensics and Security*, 19:4567–4578, 2024.

[24] O. Ronneberger et al. DF-Net: The Digital Forensics Network for Image Forgery Detection, 2025.

[25] T. Christ and L. Gunn. Undetectable and Robust Watermarking for Diffusion Models, 2025.

[26] P. Veličković et al. Graph attention networks. *Proc. International Conference on Learning Representations*, 2018.

[27] A. Kazi et al. Graph diffusion for anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3456–3467, Mar 2023.

[28] A. Hore and D. Ziou. Image quality metrics: Psnr vs. ssim. *Pattern Recognition*, 43(8):1478–1486, 2010.

[29] Z. Wang et al. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, Apr 2004.

[30] D. M. W. Powers. Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technology*, 2(1):37–63, 2011.

[31] A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[32] Q. Li et al. Advanced attack simulation in image forensics. *IEEE Transactions on Information Forensics and Security*, 19:3456–3467, 2024.

Khashayar Jafarizade received the B.S. degree in electrical engineering (telecommunications) from Islamic Azad University, Mashhad, Iran, in 2015, and the M.S. degree in secure telecommunications and cryptography from Sadjad University of Technology, Mashhad, Iran, in 2019. He is currently pursuing a PhD degree in electrical engineering (systems telecommunications) at the University of Birjand, Birjand, Iran. His research interests include cryptography, image authentication, neural networks, radar, and telecommunications systems.

Mohammad-Hassan Majidi received the B.S. degree in electrical engineering from the Shahid Bahonar University of Kerman, Kerman, Iran, in 2003, the M.S. degree in communication engineering from Imam Hossein Comprehensive University of Tehran, Tehran, Iran, in 2006, and the PhD degree in telecommunication engineering from the Department of Telecommunications, Ecole Supérieure d'Electricite, Gif-sur-Yvette, France, in 2014. He is currently an Associate Professor of Communication Engineering with the Faculty of Electrical and Computer Engineering, University of Birjand, Birjand, Iran. His research interests include signal processing, data hiding, cryptography, secure communication, and deep learning.

Hossein Gholamalinejad received the B.S. degree from University of Sistan and Baluchestan, Zahedan, Iran, in 2012, the M.S. degree from Yazd University, Yazd, Iran, in 2015, and the PhD degree from the Shahrood University of Thechnology, Shahrood, Iran, in 2021. He is currently an Assistant Professor with the Department of Computer Engineering, Bozorgmehr University of Qaenat, Qaen, Iran. His current research interests include model compression, neural architecture search, deep learning and Image Processing.