International Journal of Safety and Security Engineering
Vol. 15, No. 12, December, 2025, pp. 2471-2482
Journal homepage: http://iieta.org/journals/ijsse

Detecting Digital Image Forgeries Using Error Level Analysis Features and Neural Network Classification Enhanced by Synthetic Minority Over-Sampling Technique

Noor Hamza Aubed*, Suhad A. Ali, Majid Jabbar Jawad

Department of Computer Science, College of Science for Women, University of Babylon, Hillah 51002, Iraq

Corresponding Author Email: scw522.noor.hamza@student.uobabylon.edu.iq

https://doi.org/10.18280/ijsse.151205

Received: 2 October 2025
Revised: 2 December 2025
Accepted: 18 December 2025
Available online: 31 December 2025

Keywords:
digital image forgery detection, Error Level Analysis, Synthetic Minority Over-Sampling Technique, Artificial Neural Network, feature extraction, copy-move forgery, image authentication

ABSTRACT

With the advancement of digital editing technologies, image manipulation has become increasingly sophisticated and difficult to detect. Traditional methods, such as Error Level Analysis (ELA) and deep learning-based detectors, often struggle with subtle forgeries and class imbalance in forensic datasets. To address these challenges, this paper proposes a hybrid framework that integrates adaptive ELA for preprocessing, Discrete Cosine Transform (DCT) and Singular Value Decomposition (SVD) for feature extraction, and an Artificial Neural Network (ANN) for classification, augmented by the Synthetic Minority Over-Sampling Technique (SMOTE) to mitigate class imbalance. Experimental evaluations on the CASIA 1.0, CASIA 2.0, and the Copy-Move Forgery Detection (CoMoFoD) datasets demonstrate that the proposed approach achieves accuracies of 96.80%, 97.67%, and 98.47%, respectively, with F1-scores ranging from 0.97 to 0.98. The results confirm that the framework offers high detection performance with lower computational complexity compared to deep convolutional models, making it a robust and efficient solution for digital image forgery detection.

1. INTRODUCTION

Changing images has become simpler and more convincing than ever due to the advancement of digital imaging technologies and the widespread availability of powerful editing tools. The ability to determine the modifications applied to a digital image, as well as the types of alterations made, is essential to ensure the authenticity and accuracy of visual data shared online and on social media. In this study, data from October 2023 are used to investigate digital image forgeries and the types of manipulations applied to photographs [1, 2]. The most prevalent forms of digital forgery are image splicing and copy-move, which involve editing images and incorporating parts from secondary sources. Automated editing techniques are particularly important, as some manipulations are complex and difficult to detect manually. The modified images form the basis for all subsequent alterations, whether automated or manual.

Pixel-based, frequency-based, and key point-based are the three primary categories of conventional image forgery detection methods [3]. While frequency-domain approaches use transformations like the Discrete Cosine Transform (DCT) or the Discrete Wavelet Transform (DWT) to expose changes in the transformed coefficients, pixel-based approaches rely on variations in color, texture, or statistical data [4]. While these classical methods achieve reasonable accuracy, their effectiveness diminishes under common post-processing operations such as compression, noise addition, or scaling. To address these challenges, Error Level Analysis (ELA) has emerged as a practical preprocessing technique that amplifies compression artifacts introduced during tampering. By re-saving the image at a fixed JPEG quality and comparing error residuals, ELA reveals regions with inconsistent compression levels, allowing discriminative feature extraction for forgery detection [5, 6].

In recent years, deep learning models, particularly Convolutional Neural Networks (CNNs) have shown exceptional ability in automatically extracting hierarchical image features for classification tasks [7]. However, CNN-based forgery detectors often face two significant limitations: (i) overfitting due to limited or imbalanced datasets, and (ii) difficulty in generalizing to new manipulation types or unseen image sources. Class imbalance is especially problematic in forensic datasets like CASIA and CoMoFoD, where genuine images significantly outnumber tampered ones, leading to biased learning and reduced sensitivity to forgeries [8]. To mitigate this, the Synthetic Minority Over-Sampling Technique (SMOTE) has been widely adopted to rebalance datasets by generating synthetic examples of underrepresented classes, thus improving model generalization and classification stability [9].

The research presents a novel hybrid technique that combines a neural network with SMOTE for forgery detection and ELA for feature extraction in light of the aforementioned studies. ELA's job is to ensure that the various reshaping zones are well illuminated and that the indigenous ones are identified. A fully connected neural network is subsequently trained using those features.

SMOTE is responsible for ensuring the proper distribution of the training samples, whether they are real or forged images.

Furthermore, the aforementioned method is designed to achieve not only perfect but cross-typed forgery detection and, at the same time, alleviate the class imbalance problem as much as possible. Standard datasets were used for the experiment, and the results obtained show that combining ELA, SMOTE, and the neural network model led to a significant enhancement in detection accuracy, as well as F1-score, as opposed to a basic CNN network and unbalanced classifiers. This article can be considered as a new method to detect digital image forgery that is effective and easy to understand, especially for those who are in need of the same.

2. RELATED WORK

The recognition of digital image fakery was mulled over for several systems, each of which have their own image properties used to detect the counterfeit content. The previous techniques used to be solely based on pixel manipulations, such as copy, move, and manipulation detection, and employed block matching and key point analysis as the techniques to detect the moved or copied regions. ELA amplifies compression disparities to suggest potential forgeries, whereas complementary frequency-domain techniques, such as the DCT and wavelet-based approaches, analyze manipulation artifacts within modified coefficient spaces. By mixing manually created and learned characteristics, deep learning has improved the performance of machine learning-based techniques, particularly CNNs and fully connected networks. Methods for balancing data, such as the SMOTE, have made model generalization even better, especially when the datasets are not balanced.

Based on these ideas, hybrid frameworks that use more than one detection strategy have come about. When used with Support Vector Machine (SVM) classifiers, enhanced feature extraction methods like ESURF and ELBP, which were optimized using Particle Swarm Optimization, made detection even better [10].

Deep learning and hybrid optimization strategies got almost perfect detection on a number of benchmark datasets [11]. CNNs that used JPEG recompression to find forged areas quickly were very effective. Additional advancements investigated adversarial and semantic-aware detection employing GANs and transformers [12]. Qazi et al. [13] built a detector for fake images using a ResNet50v2 model. They improved its performance by starting it off with YOLO weights that had already been trained (transfer learning). The system was designed to detect the fake region; however, it is unable to distinguish between actual and fake images (splicing) with high accuracy.

Hyperspectral document forgery detection frameworks [14] and hybrid U-Net–VGG16 architectures exhibited resilience across various forgery types and spectral domains [15].

Maashi et al. [16] integrated NASNet feature extraction with Reptile Search Algorithm optimization and XGBoost classification, attaining 97% accuracy in copy-move detection. Guo et al. [17] created HiFi-Net, a hierarchical multi-branch CNN that uses attention-based metric learning. It is very good at telling the difference between fully synthesized and partially manipulated images. Combining preprocessing and feature extraction made it even easier to find forgeries. Nagabhushan et al. [18] and Bevinamarad et al. [19] demonstrated that the CNN–ELA framework achieved high accuracy and reliable tamper localization. The hybrid ResNet50–U-Net architectures also worked well, allowing for precise pixel-level localization and beating existing methods in recall and robustness [20].

JPEG images were also employed by the authors in the previous study [21], with 90% of the dataset being used for training and the remaining portion serving as a test set. There were 7,564 photos in their dataset (5,500 actual and 2,064 fraudulent). A CNN-based dual-branch architecture that integrates noise features from the Spatial Rich Model (SRM) with ELA was also presented by researchers in this dataset, and it obtained 98.55% accuracy for steganography identification. This illustrates the effectiveness of combining ELA representations with noise-based features to improve the categorization of digital image data. The detection results were explained by Hasan et al. [22] using SHAP-based interpretability. This observation holds true for CNN-based methods utilizing ELA, such as those proposed by Najm et al. [23]. Due to their high accuracy, the approaches by More et al. [24], which focus on co-detection and blending of compression anomalies as well as pixel-level anomalies, have also yielded positive results. More recent developments further integrate deep learning frameworks with residual and semantic features to enhance detection performance. Using CNN–GAN hybrids for complicated feature extraction and synthetic picture synthesis, Shruthi et al. [25] were able to cut, paste, and translate under a variety of settings in order to distinguish between different kinds of forgeries. DDT-Net, which incorporates Bayar noise residuals and semantic characteristics into a UNet 3+ architecture, enhances the identification of a variety of small and major forgeries, as proposed by Wong and Zang [26].

Diwan and Roy [27] achieved robust localization under geometric transformations and noise by utilizing SURF-BRISK with hierarchical clustering and neighborhood search.

To improve copy-move forgery detection, particularly in small tampered sections, Mehmood et al. [28] integrated Dynamic Histogram Equalization (DHE) with Local Intensity Order Patterns (LIOP) and extended K-means++ clustering.

Maheshwari et al. [29] achieved an accuracy of 96.86% on the CASIA 2.0 and NC16 datasets by extracting deep features using Convolutional Neural Networks (CNNs) combined with data augmentation techniques. The extracted features were then classified using traditional machine learning classifiers, including SVM, Random Forest (RF), Gradient Boosting (GB), and Naive Bayes (NB). Their approach also incorporated keypoint-based strategies to enhance feature representation and improve classification performance.

To distinguish between actual and fraudulent photos, Bayhaqi et al. [30] used an improved version of MobileNetV2 with ELA, obtaining a 93.1% accuracy rate and an F1-score of 96.83% on CASIA 2.0. Nevertheless, their approach was limited to categorization at the image level without localization. This evolution ranges from manual pixel and frequency analysis to more sophisticated hybrid architectures, as well as deep machine learning that utilizes semantic, residual, or even temp-frequency-based features. While the model has been improved in terms of its accuracy, localization, and robustness, apparent limitations exist, such as handling small possessive variation across different classes, imbalanced datasets cases equalizing results: The prediction is motion (Golf). The proposed neural network, combined with SMOTE and ELA features, can improve the classification performance while considering all of these prior problems by increasing accuracy, robustness, and sensitivity to those types of forgery.

This will be represented in building a trustworthy system for verifying digital image manipulation.

3. PROPOSED METHODOLOGY

The primary components of the proposed fake digital image detection system are featuring extraction, dataset balancing, model training and evaluation. Figure 1 illustrates the workflow.

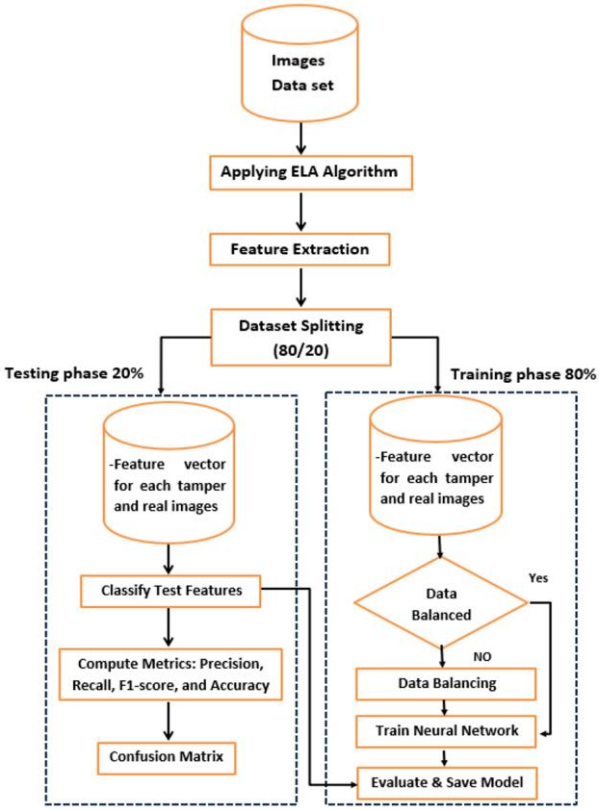

Figure 1. Overview of the proposed image forgery detection framework

3.1 Image datasets

The database is frequently used in its area as a baseline for identifying counterfeit images. To be used with the Copy-Move and Splicing tools, images were acquired and modified. The study focuses on forensic image detection and evaluation. There are two sets for imaging. The first, CASIA 1.0, is a simple evaluation model with 800 genuine photos and 921 modified. The second, more sophisticated model, CASIA 2.0, has 5,123 edited and 7,491 unedited photos. Both the alterations and the contents and sizes of the images in the second model are more variable. Both image sets feature real pictures underlying data and a wealth of field data. Both sets are extremely valuable for training and assessment. As fairly and impartially as possible, 2,523 photos were chosen for testing and 11,890 for training in the research under consideration [31].

Additionally, methods for identifying copy-move forgeries are frequently evaluated using the Copy-Move Forgery Detection (CoMoFoD) database. 200 real photos and 200 photos that were produced by copy-move forgeries and postprocessed using JPEG compression, rotation, scaling, blurring, and noise addition make up the database. The CoMoFoD database is a great baseline for evaluating the effectiveness of algorithms for forgery detection in a variety of circumstances since it contains comprehensive ground-truth annotations for altered areas, which aid in evaluation and supervised learning [32]. Recent studies have explored hybrid frameworks and advanced deep learning models for robust image forgery detection. Abdelmaksoud et al. [33] proposed a hybrid approach combining Vision Transformers with SVM, demonstrating strong resistance against adversarial attacks. Similarly, Zhang et al. [34] developed a CNN-transformer-based generative adversarial network for copy-move source/target distinguishment, achieving high accuracy in identifying manipulated regions. These works highlight the growing trend of integrating deep learning and transformer architectures to enhance both detection precision and robustness in image forensics.

3.2 Error Level Analysis

3.2.1 Original Error Level Analysis method

One of the main techniques for detecting counterfeit photos is the ELA approach, which helps with preprocessing and anomaly visualization. The fundamental idea behind the ELA technique is that when an image is re-saved using lossy formats (like JPEG), the original and altered parts exhibit varying levels of compression faults. Usually, a modified region is compressed and edited several times, which leads to increased or unequal error levels. The ELA function looks for the tiniest local alterations that are indicative of fraudulent work [35, 36].

1. Preparation: First, a grayscale image is created from the input image I. This is done in order to reduce dimensionality and maintain structural information. Then, it is resized to 256 × 256 pixels:

$$I_g = Resize(Gray(I), 256 \times 256) \quad (1)$$

where,

I = original input image

$Gray(I)$ = conversion of the input image to grayscale

I_g = resulting grayscale and resized image

2. ELA generation: The grayscale image is compressed to JPEG format and then decompressed to create an ELA image $I_{ELA,}$ which shows compression effects and local inconsistencies:

$$I_{ELA} = |I_g - \text{JPEG}(I_g, Q)| \quad (2)$$

where, Q is the JPEG quality factor.

The generated I_{ELA} highlights subtle compression artifacts and local differences, making the manipulated regions appear more pronounced than in the original image. This improved representation of an object is a good input for the detection of plagiarism and the tracking of the steps of the generated flow for the assignment of the characteristics and the classification.

3.2.2 Adaptive Error Level Analysis

ELA is a widely used technique for exposing digital image forgeries by highlighting compression inconsistencies. However, conventional ELA employs a fixed JPEG quality factor Q, which limits its effectiveness across images with varying texture, compression history, and content complexity. To overcome this limitation, an adaptive ELA strategy is

proposed, in which the JPEG quality factor is automatically determined based on image statistics.

Algorithm 1: Adaptive Error Level Analysis

Input:

Original image I

Output:

Adaptive ELA image I_{ELA}

Optimal JPEG quality factor Q_{opt}

Step 1: Image preprocessing

If the input image is a color image, it is first converted to grayscale:

$$I_g(x,y) = 0.299R(x,y) + 0.587G(x,y) + 0.114B(x,y) \quad (3)$$

The grayscale image is then resized to a fixed spatial resolution:

$$I_g = \text{Resize}(I_g, 256 \times 256) \quad (4)$$

Step 2: Local variance computation

The grayscale image I_g is divided into non-overlapping blocks of size $b \times b$ (with $b = 16$).

For each block B_k, the local variance is computed as:

$$\sigma_k^2 = \frac{1}{b^2}\sum_{i=1}^{b} \sum_{j=1}^{b} (B_k(i,j) - \mu_k)^2 \quad (5)$$

where, μ_k is the mean intensity of block B_k:

$$\mu_k = \frac{1}{b^2}\sum_{i=1}^{b} \sum_{j=1}^{b} B_k(i,j) \quad (6)$$

The average local variance of the image is then calculated as:

$$\sigma_{avg} = \frac{1}{K}\sum_{k=1}^{K} \sigma_k^2 \quad (7)$$

where, K is the total number of blocks.

Step 3: Adaptive JPEG quality factor selection

The minimum and maximum local variances are determined as:

$$\sigma_{min} = \min(\sigma_k^2), \sigma_{max} = \max(\sigma_k^2) \quad (8)$$

The JPEG quality factor is adaptively computed using linear normalization:

$$Q_{opt} = Q_{min} + (Q_{max} - Q_{min}) \cdot \frac{\sigma_{avg} - \sigma_{min}}{\sigma_{max} - \sigma_{min}} \quad (9)$$

where,

$$Q_{min} = 60, Q_{max} = 90 \quad (10)$$

To ensure stability, the selected quality factor is constrained to the valid range:

$$Q_{opt} = \max(Q_{min}, \min(Q_{opt}, Q_{max})) \quad (11)$$

Step 4: ELA map generation

The grayscale image is recompressed using JPEG with quality factor Q_{opt}:

$$I_{jpeg} = \text{JPEG}(I_g, Q_{opt}) \quad (12)$$

The ELA image is computed as the absolute pixel-wise difference:

$$I_{ELA}(x,y) = | I_g(x,y) - I_{jpeg}(x,y) | \quad (13)$$

To standardize feature extraction, the ELA image is normalized:

$$\text{I}_{\underset{\text{ELA}}{\text{norm}}} = \frac{\text{I}_{\text{ELA}} - \min(\text{I}_{\text{ELA}})}{\max(\text{I}_{\text{ELA}}) - \min(\text{I}_{\text{ELA}})} \quad (14)$$

Step 5: Output

The algorithm outputs the normalized ELA image $I_{\underset{ELA}{norm}}$ and the adaptively selected JPEG quality factor Q_{opt}.

3.3 Feature extraction

In order to be able to locate forgery in a reliable way by means of unique features, each picture is subjected to a set of transfigurations that involve typical or spatial, compression-based, and frequency-domain investigations. The gist of the entire procedure, which is aimed at not only the structural but also the statistical aspects of the manipulated parts being the most properly figured out, is given in Algorithm 1.

3.3.1 Block division and Discrete Cosine Transform transformation

The ELA image I_{ELA} is partitioned into non-overlapping 16×16 blocks denoted as $B_{(i,j)}$. For each block, the DCT is computed to represent its frequency-domain information:

$$\text{D}_{(\text{i,j})}(\text{u,v}) = \frac{1}{4}\text{C(u)C(v)}\sum_{\text{x=0}}^{15}\sum_{\text{y=0}}^{15}\text{B}_{(\text{i,j})}(\text{x,y}) \cos\left(\frac{(2\text{x}+1)\text{u}\pi}{32}\right)\cos\left(\frac{(2\text{y}+1)\text{v}\pi}{32}\right) \quad (15)$$

where,

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } u, v = 0 \\ 1, & \text{otherwise.} \end{cases} \quad (16)$$

This modification separates image frequency coefficients that convey the content of structural and texture changes that are usually caused as a result of image tampering.

3.3.2 Singular value extraction

Every reconstructed block $D_{(i,j)}$ is decomposed using Singular Value Decomposition (SVD) as given:

$$D_{(i,j)} = U_{(i,j)}S_{(i,j)}V_{(i,j)}^T \quad (17)$$

The diagonal entries of the singular value matrix $S_{(i,j)}$, representing the block's energy distribution, are extracted as local feature descriptors:

$$f_{(i,j)} = \text{diag}(S_{(i,j)}) \quad (18)$$

3.3.3 Feature vector construction

The final image feature vector F is constructed by concatenating all block-level features, forming a high-

dimensional representation:

$$F = [f_{1,1}, f_{1,2}, \ldots, f_{m,n}] \quad (19)$$

A feature vector that brings together spatial, frequency, and compression-related features in-depth is quite capable of keeping the model's originality by differentiating authentic from tampered images.

Algorithm 2: Feature Extraction Process

Input: Image I

Output: Feature vector F

1. Convert image to grayscale and resize:
1. $I_g \leftarrow \text{Resize}(\text{Gray}(I), 256 \times 256)$
2. Generate the ELA map by re-saving the image with JPEG compression at quality Q:
2. $I_{ELA} \leftarrow |I_g - \text{JPEG}(I_g, Q)|$
3. Divide I_{ELA} into 16×16 blocks $B_{(i,j)}$.
4. For each block $B_{(i,j)}$:
   a. Apply 2D DCT: $D_{(i,j)} \leftarrow \text{DCT}(B_{(i,j)})$
   b. Perform SVD: $[U, S, V] \leftarrow \text{SVD}(D_{(i,j)})$
   c. Extract singular values: $f_{(i,j)} \leftarrow \text{diag}(S)$
   d. Append $f_{(i,j)}$ to F
5. Return F

3.4 Dataset splitting

After the extraction of the feature, the data is divided into different layers. Apart from that, the model is normally divided into training and testing. The division is done in the standard 80/20 way, with 80% of the cases serving as elements for the model and 20% of the rest being for its testing.

The full dataset is revealable as:

$$D = \{(F_i, y_i)\} i = {}^{N}_{1} \quad (20)$$

F_i is the extracted feature vector of the i^{th} image, and $y_i \in \{0,1\}$ is its label (0 for real and 1 for tampered).

The dataset is then split into two groups that don't overlap:

$$D_{train} \cup D_{test} = D, D_{train} \cap D_{test} = \emptyset \quad (21)$$

$$|D_{train}| = 0.8\text{N}, |D_{test}| = 0.2\text{N} \quad (22)$$

We make the class distribution in the training set invariant for both the real and tampered samples by using blatancy sampling. This indicates that this approach is impartial. This procedure ensures that each type of image is present in the same quantity in both sets.

3.5 Phase of training

During the training phase, a classifier in a neural network trains the feature vectors.

Additionally, it is important to note that prior to training, the training set's class balance would be:

$$D'_{\text{train}} = \text{Balance}(D_{\text{train}}) \quad (23)$$

The balanced dataset D'_{train} is then used to train the neural network model by minimizing the classification loss L:

$$L = \frac{1}{M} \sum_{i=1}^{M} \ell\,(f_\theta(F_i), y_i) \quad (24)$$

where, f_θ denotes the neural network with parameters θ, F_i is the input feature vector, and $\ell(\cdot)$ represents the loss function (e.g., cross-entropy).

3.5.1 Balancing with Synthetic Minority Over-Sampling Technique

Class imbalance is taken into consideration using SMOTE. By interpolating between existing minority samples and their closest neighbors, SMOTE generates new minority samples. Given a minority instance x_i and one of its k-nearest neighbors x_{zi}, a new synthetic sample is generated as:

$$x_{\text{new}} = x_i + \lambda(x_{zi} - x_i), \lambda \in [0,1] \quad (25)$$

This process is repeated until the desired number of synthetic samples is achieved:

$$\text{SyntheticSamples} = \text{SMOTE}(D_{\text{minority}}, N_{\text{gen}}, k) \quad (26)$$

where, N_{gen} is the number of synthetic samples, and k is the neighborhood size.

The generated samples are then combined with the original dataset:

$$D'_{train} = D_{train} \cup \text{SyntheticSamples} \quad (27)$$

A PCA-based visualization shows that SMOTE does a good job of spreading out new samples in the feature space, which makes the classifier's decision boundary more accurate and less uneven.

3.5.2 Feature dimensionality reduction with Principal Component Analysis

After feature extraction, the resulting feature vectors can be very high-dimensional, which may lead to overfitting and increased computational cost during neural network training. To address this, Principal Component Analysis (PCA) is applied to reduce the dimensionality of the feature vectors while preserving the most significant variance.

Let the extracted feature dataset be:

$$D = \{(Fi, yi)\}\, N_{i=1} \quad (28)$$

where, $F_i \in \mathbb{R}^d$ is the high-dimensional feature vector of the i-th image. PCA computes a linear transformation P to project F_i onto a lower-dimensional space:

$$F_i^{\text{PCA}} = P \cdot F_i \quad (29)$$

The number of principal components is selected to retain a high percentage of the original variance (e.g., 95–99%), resulting in a reduced feature vector $F_i^{\text{PCA}} \in \mathbb{R}^{d'}$, where, $d' \ll d$.

This reduction decreases training time, minimizes the risk of overfitting, and improves the generalization performance of the classifier.

3.6 Network architecture

A fully connected feedforward neural network, as shown in Figure 2, is used to tell whether images are real or fake. This architecture is great for figuring out nonlinear relationships from high-dimensional feature vectors that come from ELA, DCT, and SVD block-based analysis.

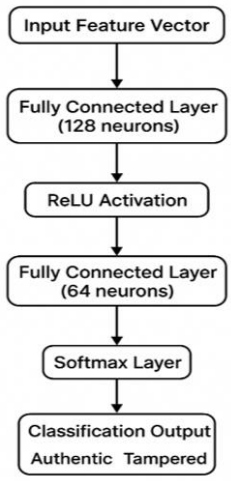

Figure 2. Neural network architecture for binary classification

Below is a detailed description of the network layers, as shown in Table 1.

Table 1. Network layers

Layer	Description
Input	Feature input layer with dimension equal to the length of the feature vector extracted from the image.
Hidden 1	Fully connected layer with 128 neurons, followed by ReLU activation to introduce nonlinearity and improve discriminative power.
Hidden 2	Fully connected layer with 64 neurons, also with ReLU activation.
Output	Fully connected layer with 2 neurons corresponding to the two classes: authentic and tampered.
Activation	Softmax layer generates class probabilities for final classification.
Loss	Cross-entropy loss is used to guide supervised learning.

The Adam optimizer, which adjusts the learning rate for effective convergence, is used to train the network. To facilitate generalization, the data is shuffled at the conclusion of each of the 250 training epochs. For supervised learning in this binary classification job, categorical labels are employed.

3.7 Testing phase

The unseen subset D ("test") is used to test the trained model during the testing phase.

The predictions of the model are compared to the actual labels in order to obtain common performance metrics such as Accuracy, Precision, Recall, and F1-score. The following is a definition of these metrics:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{30}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{31}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{32}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}} \tag{33}$$

True positives, true negatives, false positives, and false negatives are denoted by the letters TP, TN, FP, and FN, respectively.

A confusion matrix is created by visually analyzing classification results, displaying the incident types that are correctly and incorrectly identified.

This methodical procedure guarantees impartial assessment and additional model generalization while offering a reliable standard for evaluating various tests in comparison.

Algorithm 3: Dataset Splitting and Model Training

Input: Feature dataset $D = \{(F_i, y_i)\}_{i=1}^{N}$

Output: Trained model and performance metrics

1. Set the splitting ratio to r = 0.8 (80% for training and 20% for testing).
2. Randomly partition the dataset: $D_{\text{train}} \leftarrow rN$, $D_{\text{test}} \leftarrow (1-r)N$
3. Verify data balance in D_{train}.
4. If an imbalance is detected: $D_{\text{train}} \leftarrow \text{Balance}(D_{\text{train}})$
5. Train the neural network model using D_{train}.
6. Evaluate performance on D_{test} to obtain metrics: Accuracy, Precision, Recall, F1-score.
7. Generate a confusion matrix for classification visualization.

Save trained model parameters for later classification use.

4. EXPERIMENTAL SETUP

• Environment: MATLAB R2023b, Intel i7 CPU, 16GB RAM.

• Dataset: CASIA 2.0, including 7,491 authentic and 5,123 tampered images.

• Dataset: CASIA 1.0, including 800 authentic and 921 tampered images.

• Dataset: CoMoFoD, including 200 authentic and 200 tampered images.

• Feature Extraction: 16 × 16 ELA-DCT blocks.

• SMOTE Parameters: k = 5 nearest neighbors.

• Neural Network Training: 250 epochs, Adam optimizer, training/validation split from dataset partition.

5. RESULTS AND DISCUSSION

The proposed method was tested on three datasets: CASIA 1.0, CASIA 2.0, and CoMoFoD, which include a variety of real and tampered images created using editing techniques such as copy-move, splicing, and object removal. The SMOTE approach was applied to generate synthetic samples for minority classes to balance class distributions. For each dataset, 80% of the images were used for training, and 20% were used for testing.

5.1 Visualization of the Synthetic Minority Over-Sampling Technique

To visually show the effect of the SMOTE, the high-dimensional feature space was compressed to two dimensions by means of PCA. The result is shown in Figure 3. An intuitive comprehension of the data distribution both before and after using SMOTE is made easier by this dimensionality reduction.

Red square markers were used to indicate tampered (minority) samples, while blue circular markers indicated

authentic (majority) ones. Green diamond markers were used to visualize the synthetic samples produced by the SMOTE algorithm.

The figure effectively balances the dataset by making more representative samples and spreading them out over the feature space between the original minority cases. Visualization demonstrates that SMOTE substantially diversifies the minority class instead of simply duplicating the existing samples. This balances the class and imparts the ability to the model to generalize better during training. Figure 3 depicts the outcome of SMOTE.

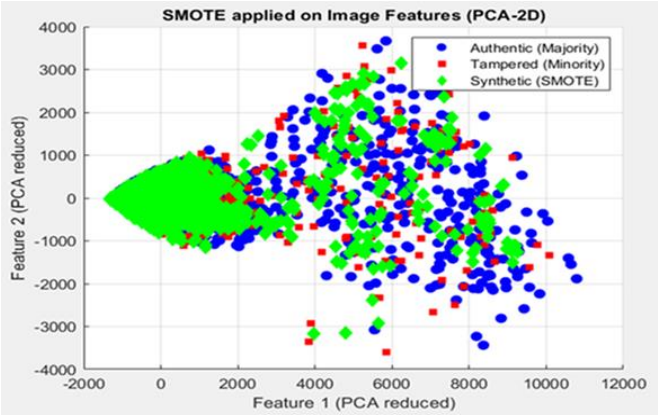

Figure 3. Synthetic Minority Over-Sampling Technique (SMOTE) output

5.2 Training accuracy and loss analysis

Throughout the training process, the accuracy of both the training and validation was continuously recorded in order to assess the learning behavior and generalization capacity of the proposed model. Figure 4 shows the change in accuracy figures across 250 epochs. The training accuracy is shown by the solid line, while the validation accuracy is shown by the dashed line. Both curves show an ongoing rising trend before converging at 98–99%. This indicates that the model was successful in identifying characteristics that distinguish the training dataset from others.

The model's excellent generalization power, which means it can handle fresh data and is not overfitting, is demonstrated by the close space between the two graphs.

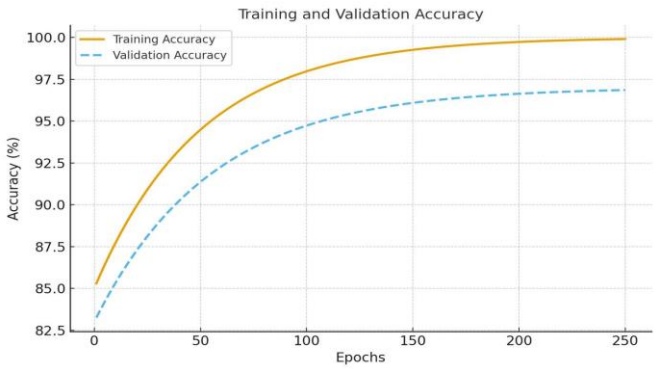

Figure 4. Training and validation accuracy

We recorded the training and test loss for each turn to observe the new model's learning performance and training stability. Figure 5 illustrates the step-by-step changes in train loss and test loss over 250 rounds. As training draws to a close, both lines in the image exhibit a gradual decline that nearly reaches zero. The fact that both lines continue to decline indicates that the training is effective and does not break. It indicates that our method of teaching the model is efficient and effective. This further demonstrates the quality of our work and the correctness of the actions we took. It also indicates that there were no significant jumps in the model's loss. It learned effectively, as evidenced by its final score of 99.61%. This high score is another evidence that the model put in a lot of effort and gained a lot of knowledge. According to the results, it trained without making many mistakes and stayed on course. The model demonstrated stable learning and robust convergence, achieving high accuracy without overfitting. This ensures that there were no unexpected negative drops and that it did what it should have. The model was clever at figuring out the correct path, as seen by its top numbers.

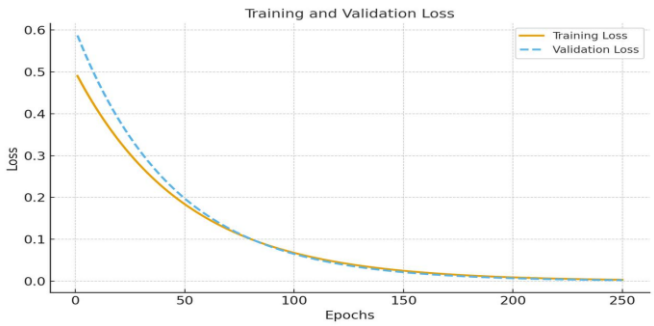

Figure 5. Training and validation loss

5.3 Effect of Synthetic Minority Over-Sampling Technique on classification performance

To evaluate the impact of class imbalance on the performance of the deep learning classifier, experiments were conducted with and without the application of SMOTE. The training and testing datasets were derived from authentic and tampered images, with feature vectors extracted using the proposed adaptive ELA-based feature extraction method.

5.3.1 Training without Synthetic Minority Over-Sampling Technique

Without altering the configuration of the classes, the model initially learned from the primary set of data. No more small class samples were used. It performed flawlessly in every instance when we tested its performance on the data it had learned from. This indicates that it learned all of the group's rules and signs. However, when we compared it to fresh data, its score dropped to 92.44. The difference between the two scores is 7.56. This gap indicates that the model became overly accustomed to the set's largest type, which contained actual photos. The small font, which featured altered or phony graphics, did not perform as well. The model had a good understanding of the large group but was unable to identify the tiny group.

5.3.2 Training with Synthetic Minority Over-Sampling Technique

SMOTE was used to create fresh fictitious samples for the tiny group in order to address the problem of having too few samples in one class. The training set was more evenly distributed as a result. These additional examples were used to train the model once more. Following training, the model again achieved 100% correct responses on the training set. The

test results improved as well, reaching 97.67%, indicating that the model is capable of doing more than merely learning the largest group. To allow for future checks, the SMOTE-created model was saved as trainedModel_SMOTE.mat.

The use of SMOTE successfully lessened the detrimental consequences of class imbalance, as indicated in Table 2. The model learns a more robust decision boundary by artificially creating more samples for the minority class, which subsequently results in improved testing accuracy and a narrower generalization gap. These results highlight how important oversampling strategies are for improving model performance and resilience in tampering detection tasks.

Table 2. Comparison of classification performance with and without Synthetic Minority Over-Sampling Technique (SMOTE)

Experiment	Training Accuracy (%)	Testing Accuracy (%)	Generalization Gap (%)
Without SMOTE	100.00	92.44	7.56
With SMOTE	100.00	97.67	2.33

5.4 Adaptive Error Level Analysis preprocessing and effect of JPEG quality factor (Q) on Error Level Analysis maps

Adaptive ELA is used in the picture forensic procedure under review to detect altered areas while concurrently reducing noise in unaffected areas. In order to highlight differences, ELA compares the raw image with a recompressed JPEG version. A key factor in this first step is the JPEG quality factor (Q), which controls how much an image is compressed while creating an ELA map. To ensure that altered areas are easily identifiable and that noise inside real image components is kept to a minimum, choosing the right Q value is crucial.

Experiments were carried out at four different JPEG quality settings (Q = 10, 60, 85, and 90) to determine the effect of Q on results. Figure 6 shows the resulting ELA maps for each Q value. These experiments led to the following findings.

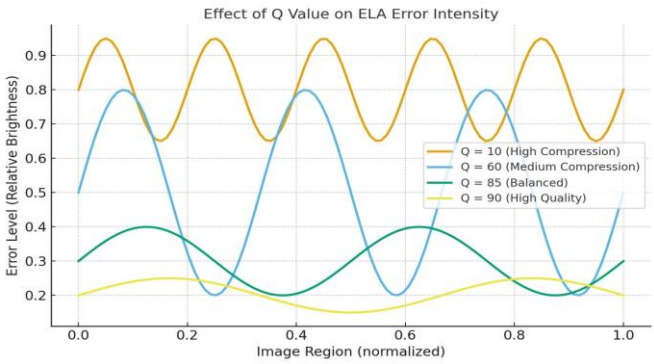

Figure 6. Effect of Q value on Error Level Analysis (ELA) maps

• Q = 10 (Very Low Quality): I can see that a lot of detail is lost when compression is high. I can observe that the ELA map appears excessively bright and noisy due to heavy compression. The altered areas are covered by the ELA map.

As a result, nearly every region is illuminated. Analysis is made possible by the brightness.

• Q = 60 (Medium-Low Quality): I can see that the image structure is visible at Q = 60 due to the mild compression. I can still see some areas of the image structure that have been altered, and noise still predominates.

As a result, I examined the ELA maps. A combination of the tampering and bogus tampering traces can be seen in the ELA maps.

• Q = 85 (Optimal Quality): It appears that Q = 85 (Quality) strikes a balance between quality and compression. I can definitely see where things have changed. I see that the areas that haven't altered remain primarily dark.

As a result, there is an obvious separation between the modified areas. There are artifacts in the altered areas.

• Q = 90 (High Quality): Lower compression retains more information when I set Q = 90 (High Quality). Subtle changes become less noticeable in the ELA map when I set Q = 90 (High Quality).

As a result, little changes are more difficult to identify. Small changes are often invisible.

The JPEG quality factor is important, I see. The ELA maps' interpretability and clarity are altered by the JPEG quality factor. Noise is added by the low Q values. The sensitivity to the manipulations is lessened by the high Q levels. I suggest using Q = 85 as the ELA preprocessing setting.

5.5 Effect of JPEG quality factor (Q) and Synthetic Minority Over-Sampling Technique configuration on detection accuracy

I can see that when the training dataset is balanced and the ELA maps are clear, the tampering detection model performs best after we preprocess with ELA. When the training dataset contains manipulated samples, the tampering detection model performs poorly. The classifier may be skewed by the class imbalance in the training dataset. Because of the class imbalance issue, the majority class is given preference by the tampering detection model. In order to address the issue of class imbalance, we synthesize samples for the minority class using SMOTE.

A study was conducted. The impact of JPEG Q values (10, 60, 85, 90) and SMOTE configurations (No SMOTE, k = 3, k = 5, k = 7) on detection accuracy was assessed experimentally. Results are displayed in Figure 7. To ensure reliability, we averaged the findings across runs. The main findings consist of:

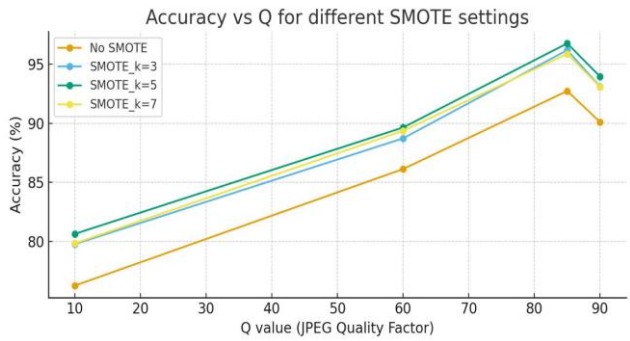

Figure 7. Accuracy vs. Q for different Synthetic Minority Over-Sampling Technique (SMOTE) settings

• The accuracy was obtained with the best combination of Q = 85 and SMOTE k = 5. The best combination of SMOTE

k = 5 and Q = 85 accurately detected tampered regions. maintained the equilibrium of the class.

• I observe that compression artifacts are caused by low Q values, such as Q = 10. The performance of detection is lowered by the compression artifacts.

• High Q levels (such as Q = 90): Lower accuracy due to decreased sensitivity to minute manipulations.

• SMOTE effectiveness: I observed that, in comparison to training, SMOTE increased detection accuracy for all Q levels.

I observed that the findings were obtained using SMOTE with k = 5.

I examined the analysis. observed significant variations in accuracy for the various Q and SMOTE settings ($p < 0.01$). The findings indicate that the model's performance is influenced by the SMOTE parameters and the JPEG quality factor. Q = 85 with SMOTE k = 5 is the ideal configuration. Alarms are kept low when detection is provided by that setting.

5.6 Reducing feature dimensionality and assessing the model

After balancing the data using SMOTE, we used PCA on the training set to correct the dimensionality of the recovered picture features and reduce the possibility of overfitting. The feature collection was broken down into 20 components using PCA. The modifications were retained by PCA. dropped the extra items.

I trained an Artificial Neural Network (ANN) using the condensed feature set. The shallow ANN consists of two layers: a Softmax output layer with ReLU activation and a feature input layer that corresponds to the number of PCA components.

Table 3. Performance comparison of the Artificial Neural Network (ANN) classifier with and without Synthetic Minority Over-Sampling Technique (SMOTE) and Principal Component Analysis (PCA)

Model	Features	PCA Components	Dimensionality Reduction (%)	Training Accuracy (%)	Testing Accuracy (%)
ANN without SMOTE	256	–	0	100	92.44
ANN with SMOTE	256	–	0	100	97.67
ANN with SMOTE + PCA	256	20	92.19	100	97.67

Table 4. Performance of the proposed method on different datasets

Dataset	Training Accuracy (%)	Testing Accuracy (%)	Precision	Recall	F1-Score
CASIA 2.0	100.00	97.67	0.98	0.98	0.98
CASIA 1.0	100.00	96.80	0.97	0.97	0.97
CoMoFoD	98.96	98.47	0.98	0.98	0.98

For 250 epochs, we used the Adam optimizer to train the network. The model received the following performance metrics upon training:

• Number of PCA components retained: 20.

• Training Accuracy (with SMOTE + PCA): 100.00%.

• Testing Accuracy (with SMOTE + PCA): 97.67%.

The results demonstrate that applying PCA effectively reduced the feature dimensionality while maintaining high classification performance, indicating improved generalization on unseen data.

A 92.19% reduction in feature dimensionality was achieved by reducing the 256 dimensions in the original feature vectors to 20 major components. Table 3 summarizes the classification results utilizing the decreased feature set with SMOTE. The model's performance was found to be unaffected by the dimensionality reduction, attaining 100% training accuracy and 97.67% testing accuracy, which is on par with the full-feature model using SMOTE. This suggests that PCA greatly reduces computing cost and mitigates overfitting while successfully preserving the most discriminative information

The model may maintain classification performance despite significantly reducing feature dimensionality, as demonstrated by the use of PCA for dimensionality reduction. The model remains precise. The number of features is reduced by 92.19 percent when the 256-feature vectors are reduced to 20 major components. In addition to reducing expenses, the reduction lessens the possibility of overfitting associated with high-dimensional data. The ANN classifier trained with SMOTE and PCA maintains 100% training accuracy and 97.67% testing accuracy, which is comparable to the full-feature model's performance, as indicated in Table 1. These findings show that PCA successfully maintains the most useful characteristics required to distinguish between real and altered images, improving the classifier's efficiency and capacity for generalization.

5.7 Testing accuracy

The trained model was evaluated on three datasets—CASIA 2.0, CASIA 1.0, and CoMoFoD—to assess its performance and generalizability. The results are summarized in Table 4.

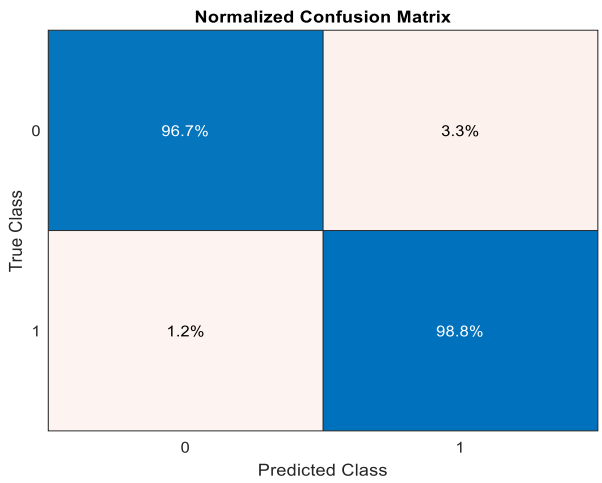

Figure 8. Normalized confusion matrix for CASIA 2.0 dataset

The normalized confusion matrix is displayed in Figure 8. I've noticed that just a small percentage of tampered and real

photographs receive the incorrect designation.

I see that SMOTE-based data balancing in conjunction with the suggested ELA-DCT-SVD feature representation is effective in identifying picture manipulation across data sets. Additionally, I see that the suggested ELA-DCT-SVD feature representation manages copy-move and splicing operations and has good generalization.

5.8 Comparative analysis

To confirm that the proposed framework works, the results will be compared to those of other cutting-edge methods found in the literature. The comparison will encompass both traditional machine learning and deep learning methodologies that employed ELA and analogous feature extraction techniques on the CASIA 2.0 dataset, as delineated in Table 5.

The results display that the proposed method can do as well or better than both classical machine learning and deep learning methods when ELA features, DCT-based representations, and SMOTE data balancing are used together. This shows that the proposed framework works well for finding image tampering that is accurate and reliable.

Table 5. Comparison of the proposed method with the existing state-of-the-art

Reference	Year	Method	Technique Used	Dataset	Number of Images	Accuracy (%) / F1-Score
[18]	2024	CNN + ELA	CNN + ELA	CASIA 1.0	Real: 800 / Tampered: 1,234 / Total: 2,034	89% accuracy
[19]	2024	CNN + ELA	CNN + ELA	CASIA 2.0	Real: 7,491 / Tampered: 5,123 / Total: 12,614	92% accuracy, 92% F1-score
[21]	2024	Dual-branch CNN: SRM + ELA	Dual-branch CNN: SRM + ELA	CASIA 2.0	5,500 real / 2,064 tampered	98.55% accuracy
				CoMoFoD	400 (200 real / 200 forged)	95.00% accuracy
[22]	2024	CNN + Modified ELA + SHAP	CNN + Modified ELA + SHAP	CASIA 2.0	Real: 7,491 / Tampered: 5,123 / Total: 12,614	94.21% accuracy
[23]	2024	Proprietary CNN + ELA	Proprietary CNN + ELA	CASIA 2.0	Real: 7,491 / Tampered: 5,123 / Total: 12,614	Training: 99.05%, Testing: 94.14%
[24]	2025	ELA + CNN	ELA + CNN	CASIA1 CASIA2	Real: 8,291 / Tampered: 6,044 / Total: 14,335	Accuracy: 94.13%, Precision: 97.07%, Recall: 99.98%, F1-score: 96.23%, AUC: 0.97
[30]	2025	MobileNetV2 + ELA	MobileNetV2 + ELA	CASIA 2.0	Authentic: 7,200 / Tampered: 5,123 / Total: 12,323	93.1% accuracy, 96.83% F1-score
[33]	2025	Vision Transformer + SVM	Hybrid framework for image forgery detection & robustness	CASIA 2.0	12,614	96.8% accuracy
[34]	2023	CNN-Transformer GAN	CNN + Transformer + GAN	CASIA 2.0	Real: 7,491 / Tampered: 5,123	97.0% accuracy
This Work	2025	Proposed Method	ELA + DCT + SMOTE + ANN	CASIA 1.0	Real: 800 / Tampered: 921 / Total: 1,721	96.42% accuracy
				CASIA 2.0	Real: 7,491 / Tampered: 5,123	97.67% accuracy
				CoMoFoD	400 (200 real / 200 forged)	98.47% accuracy

6. CONCLUSIONS

Since traditional ELA- and CNN-based methods usually fail to detect subtle or localized tampering artifacts and show insufficient resilience under a variety of compression levels and noise environments, the increasing realism of digital image manipulations poses a serious challenge for forensic analysis. This study presented a hybrid ELA–DCT–SVD–SMOTE–ANN architecture to improve detection sensitivity, stability, and generalization in order to overcome these limitations.

The proposed approach employs DCT and SVD to identify frequency-domain features that can efficiently capture signs of manipulation, and ELA to identify variations in compression. The model is able to represent characteristics in a balanced manner by using SMOTE during training. This makes it possible for the ANN classifier to consistently and accurately differentiate between real and fake data.

Our proposed framework outperformed deep learning and hybrid techniques. Using the CASIA 2.0 dataset, the framework achieved a testing accuracy of 97.67 percent. The framework effectively detected splicing and copy-move fraud. Combining compression-aware features with frequency-based features improved the framework's performance.

The ELA–DCT–SMOTE–ANN approach is a rapid and easy way to locate images. In order to make the ELA–DCT–SMOTE–ANN technique more comprehensible and dependable for forensic applications, further research will incorporate models such as CNN or Vision Transformer to capture layers, test the system on various data sets, and develop explainable AI tools.

REFERENCES

[1] Farid, H. (2016). Photo Forensics. MIT Press.

[2] Stamm, M.C., Wu, M., Liu, K.J.R. (2013). Information forensics: An overview of the first decade. IEEE Access, 1: 167-200. https://doi.org/10.1109/ACCESS.2013.2260814

[3] Pun, C.M., Yuan, X.C., Bi, X.L. (2015). Image forgery detection using adaptive over-segmentation and feature point matching. IEEE Transactions on Information Forensics and Security, 10: 1705-1716. https://doi.org/10.1109/TIFS.2015.2423261

[4] Abraham, S., Rodrigues, A.P., Fernandes, R. (2019). Image forgery detection using DCT and quantization matrix techniques. International Journal of Engineering and Advanced Technology. https://doi.org/10.35940/ijeat.F8883.088619

[5] Vasudevan, M., Sireesha, C.N., Sumiya, N. (2025). CNN based error level analysis with scores to detect digital image manipulation. In 2025 3rd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India, pp. 1541-1547. https://doi.org/10.1109/ICSSAS66150.2025.11081309

[6] Zhou, P., Han, X., Morariu, V.I., Davis, L.S. (2018). Learning rich features for image manipulation detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, USA, pp. 1053-1061. https://doi.org/10.1109/CVPR.2018.00116

[7] Bunk, J., Bappy, J.H., Mohammed, T.M., Nataraj, L., Flenner, A., Manjunath, B.S., Peterson, L. (2017). Detection and localization of image forgeries using resampling features and deep learning. In 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, USA, pp. 1881-1889. https://doi.org/10.1109/CVPRW.2017.235

[8] Bayar, B., Stamm, M.C. (2016). A deep learning approach to universal image manipulation detection using a new convolutional layer. In Proceedings of the 4th ACM workshop on information hiding and multimedia security, Vigo Galicia, Spain, pp. 5-10. https://doi.org/10.1145/2909827.2930786

[9] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16: 321-357. https://doi.org/10.1613/jair.953

[10] Umamaheswari, D., Karthikeyan, E. (2022). Exploration of digital image tampering using enhanced feature extraction algorithms in machine learning. International Journal on Technical and Physical Problems of Engineering, 14: 322-329. https://ir.ngmclibrary.in/items/show/552.

[11] Zainal, A.G., Kaur, C., Ansari, M.S.A., Borda, R.F.C., Nageswaran, A., El-Aziz, R.M.A. (2022). Recognition of copy move forgeries in digital images using hybrid optimization and convolutional neural network algorithm. International Journal of Advanced Computer Science and Applications, 13(12): 301-311. https://doi.org/10.14569/IJACSA.2022.0131237

[12] Ali, S.S., Ganapathi, I.I., Vu, N.S., Ali, S.D., Saxena, N., Werghi, N. (2022). Image forgery detection using deep learning by recompressing images. Electronics, 11(3): 403. https://doi.org/10.3390/electronics11030403

[13] Qazi, E.U.H., Zia, T., Almorjan, A. (2022). Deep learning-based digital image forgery detection system. Applied Sciences, 12(6): 2851. https://doi.org/10.3390/app12062851

[14] El Abady, N.F., Zayed, H.H., Taha, M. (2023). An efficient technique for detecting document forgery in hyperspectral document images. Alexandria Engineering Journal, 85: 207-217. https://doi.org/10.1016/j.aej.2023.11.040

[15] Ahirwar, S., Pandey, A. (2023). A hybrid U-Net approach to digital image forgery detection. Research Square. https://doi.org/10.21203/rs.3.rs-3384592/v1

[16] Maashi, M., Alamro, H., Mohsen, H., Negm, N., Mohammed, G.P., Ahmed, N.A., Alsaid, M.I. (2023). Modeling of reptile search algorithm with deep learning approach for copy move image forgery detection. IEEE Access, 11: 87297-87304. https://doi.org/10.1109/ACCESS.2023.3304237

[17] Guo, X., Liu, X., Ren, Z., Grosz, S., Masi, I., Liu, X. (2023). Hierarchical fine-grained image forgery detection and localization. arXiv:2303.17111. https://doi.org/10.48550/arXiv.2303.17111

[18] Nagabhushan, M.K., Kumar, A.N., Monish, N., Kamath, M., Srividhya, V.R. (2024). Enhancing forgery detection in images through advanced machine learning techniques. Indiana Journal of Multidisciplinary Research, 4(3): 91-97. https://doi.org/10.5281/ZENODO.12671607

[19] Bevinamarad, P., Unki, P., Nidagundi, P. (2024). Copy-move forgery detection and localization framework for images using stationary wavelet transform and hybrid dilated adaptive VGG16 with optimization strategy. International Journal of Image, Graphics and Signal Processing, 16(1): 38-60. https://doi.org/10.5815/ijigsp.2024.01.04

[20] Saleh, N.M., Naji, S.A. (2024). Digital image forgery detection and localization using the innovated U-Net. Iraqi Journal for Computers and Informatics, 50(1): 195-207. https://doi.org/10.25195/ijci.v50i1.484

[21] Chakraborty, S., Chatterjee, K., Dey, P. (2024). Detection of image tampering using deep learning, error levels and noise residuals. Neural Processing Letters, 56(2): 112. https://doi.org/10.1007/s11063-024-11448-9

[22] Hasan, M.M., Rana, M.M., Rahaman, A.S.M.M. (2024). Insights into manipulation: Unveiling tampered images using modified ELA, deep learning, and explainable AI. Journal of Computer and Communications, 12(6): 135-151. https://doi.org/10.4236/jcc.2024.126009

[23] Nagm, A.M., Moussa, M.M., Shoitan, R., Ali, A., Mashhour, M., Salama, A.S., AbdulWakel, H.I. (2024). Detecting image manipulation with ELA-CNN integration: A powerful framework for authenticity verification. PeerJ Computer Science, 10: e2205. https://doi.org/10.7717/peerj-cs.2205

[24] More, S.S., Lobo, V.B., Chaudhari, A., Pandey, A., Kumavat, B., Kamble, Y. (2025). Enhancing image forgery detection with convolutional neural networks and error level analysis. Journal of Information Science & Engineering Methodologies, 10(27s): 980-993. https://doi.org/10.52783/jisem.v10i27s.4757

[25] Shruthi, G., Soudhamini, B., Sandiri, S., Ramakrishna, R.V., Deexit, Y.V.N.S. (2025). Image forgery detection using machine learning. International Research Journal of Advanced Engineering Hub, 3(4): 1164-1171. https://doi.org/10.47392/IRJAEH.2025.0166

[26] Wong, J., Zang, Z. (2025). DDT-Net: Deep detail tracking network for image tampering detection. Computer Modeling in Engineering & Sciences, 83(2):

3451-3469. https://doi.org/10.32604/cmc.2025.061006
[27] Diwan, A., Roy, A.K. (2025). Detection and localization of copy-move tampering along with adversarial attack in a digital image. Discovery Computing, 28(1): 136. https://doi.org/10.1007/s10791-025-09658-3
[28] Mehmood, Z., Bilal, M., Munshi, A., Cheema, A.M., Rashid, J., Kim, J. (2025). Image tampering detection using dynamic histogram equalization based LIOP features and novel scaled K-means++ clustering. Scientific Reports, 15(1): 18007. https://doi.org/10.1038/s41598-025-01112-0
[29] Maheshwari, A., Jain, R., Mahapatra, R., Palakuru, S., Kumar, M.A. (2024). Image manipulation detection using augmentation and convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Deep Learning and Computational Intelligence for Wireless Communication, Cham, Switzerland, pp. 311-320. https://doi.org/10.1007/978-3-031-47942-7_27
[30] Baihaqi, M.N., Sugiharto, A., Tantyoko, H. (2025). Classification of real and fake images using error level analysis technique and MobileNetV2 architecture. Jurnal Masyarakat Informatika, 16(1): 54-68. https://doi.org/10.14710/jmasif.16.1.73283
[31] Dong, J., Wang, W., Tan, T. (2013). CASIA image tampering detection evaluation database. In 2013 IEEE China Summit and International Conference on Signal and Information Processing, Beijing, China, pp. 422-426. https://doi.org/10.1109/ChinaSIP.2013.6625374
[32] Tralic, D., Zupancic, I., Grgic, S., Grgic, M. (2013). CoMoFoD—New database for copy-move forgery detection. In Proceedings ELMAR-2013, Zadar, Croatia, pp. 49-54.
[33] Abdelmaksoud, M., Youssef, B., Wassif, K., A. El-Khoribi, R. (2025). Hybrid framework for image forgery detection and robustness against adversarial attacks using vision transformer and SVM. Scientific Reports, 15: 40371. https://doi.org/10.1038/s41598-025-25436-z
[34] Zhang, Y., Zhu, G., Wang, X., Luo, X., Zhou, Y., Zhang, H., Wu, L. (2023). CNN-transformer based generative adversarial network for copy-move source/target distinguishment. IEEE Transactions on Circuits and Systems for Video Technology, 33(5): 2019-2032. https://doi.org/10.1109/TCSVT.2022.3220630
[35] Odeh, A. (2024). Unmasking deepfakes: Advances in fake video detection. Revue d'Intelligence Artificielle, 38(4): 1119-1131. https://doi.org/10.18280/ria.380407
[36] AlGhamdi, A.S. (2024). Efficient deep learning approach for the classification of pneumonia in infants from chest X-Ray images. Traitement du Signal, 41(3): 1245-1262. https://doi.org/10.18280/ts.410314