International Journal of Safety and Security Engineering
Vol. 16, No. 4, April, 2026, pp. 723-730
Journal homepage: http://iieta.org/journals/ijsse

Lightweight Convolutional Neural Network-Based Forgery Detection with Parallelized Feature Extraction

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.160402

Received: 6 March 2026
Revised: 20 April 2026
Accepted: 28 April 2026
Available online: 30 April 2026

Keywords:
lightweight Convolutional Neural Network, image forgery detection, digital image forensics, parallel feature extraction

ABSTRACT

Being able to tell the difference between real and fake digital photos is an important part of multimedia forensics. This study proposes a hybrid architecture that integrates handcrafted and deep learning features to accurately detect tampering. The preprocessing pipeline uses block-wise Discrete Cosine Transform-Singular Value Decomposition (DCT-SVD) feature extraction, multi-quality Error Level Analysis (ELA), grayscale conversion, and Local Binary Patterns (LBP). A lightweight Convolutional Neural Network (CNN) with convolutional layers, batch normalization, global average pooling, and fully connected layers with softmax classification uses these features to make inputs with more than one channel. We used two large datasets, CASIA1 and CASIA2, to test the framework we made. We set aside 20% of the data for testing and 80% for training. Our system was right 98.13% of the time on CASIA1. It was also highly effective at finding the right things, with a precision of 98.09% and a recall of 98.17%. This meant that it was very well balanced between the two, with an F1-score of 98.12%. It still worked well, getting it right 96.68% of the time when we tested it on the bigger CASIA2 dataset. It was still highly effective at finding everything, with a recall rate of 97.49%, but it wasn't quite as precise, with a rate of 96.54%. We used several computers to get features at the same time and a special kind of computer chip to speed up the neural network part. Because of this, we could almost always work on problems as they came up.

1. INTRODUCTION

One big problem is how easy it is to fake digital photos. It's can be easily changing photos and make them look real because so many people use digital cameras and editing software. But this could be very bad because fake pictures can change what people think, change what we see on social media and in the news, and even change what we see in court. So, it's very important to find out if an image has been changed. There are two main types of fake photos. One way to do this is to copy and paste a part of the image onto another part of the same image. The second is putting something from another picture into the original picture [1, 2].

Error Level Analysis (ELA) [3] and Local Binary Patterns (LBP) [4] to find artifacts in the texture, frequency, and image compression domains. These techniques are effective in certain situations, but they are ineffective when dealing with complex adjustments or post-event chores like JPEG compression, scaling, or adding noise. In the past, people looked for handmade items that had flaws caused by tampering to tell fake goods from real ones. Many people have used block-based Discrete Cosine Transform-Singular Value Decomposition (DCT-SVD) analysis [5].

Convolutional Neural Networks (CNNs) have demonstrated great potential for tasks like identifying phoney images, and deep learning has advanced significantly [6, 7]. While hand-crafted approaches may miss subtle variations in spatial and contextual information, CNNs are capable of automatically learning hierarchical features from raw photos. If the quality or usage of the photos changes, deep learning models may not perform well [8]. On their own, they might also need a lot of data and computational power.

Recent studies indicate that optimal outcomes from both paradigms can be achieved through the integration of handcrafted features with deep learning. These techniques, such as domain-specific descriptors and data-driven feature learning, make it more difficult and unclear to create many types of images [9, 10]. It has been demonstrated on benchmark datasets that the integration of CNNs with ELA and LBP features enhances detection accuracy. Block-wise DCT features, on the other hand, add to learned features by capturing local frequency differences [11]. This paper proposes a hybrid framework for image fraud detection that integrates a CNN-based classifier with multi-quality ELA, LBP, and block-wise DCT-SVD features, motivated by the aforementioned results. The method's effectiveness for practical forensic applications is validated by its superior accuracy, precision, recall, and F1-score when evaluated on the CASIA 1 and CASIA 2 datasets.

2. RELATED WORK

In recent years, significant attention has been devoted to detecting forged digital images. Existing approaches can be broadly categorized into three groups: handcrafted feature-based methods, deep learning-based methods, and hybrid approaches.

Handcrafted feature-based methods

Early research focused on handcrafted features to identify inconsistencies introduced during image manipulation. Farid [3] proposed JPEG ghost analysis to reveal compression inconsistencies caused by image splicing. LBP, a dependable texture descriptor created by Ojala et al. [4], is frequently used to identify structural issues in images. Popescu and Farid [5] improved resampling-based detection techniques to identify copy-move forgeries by looking at block-wise artefacts in transform domains like DCT and SVD.

These techniques don't demand a lot of processing power and are simple to learn. However, they frequently struggle to notice minor adjustments or manage post-processing operations like compression and noise.

Deep learning-based approaches

CNNs have showed great promise in identifying phoney photographs as deep learning has advanced. A constrained CNN architecture was proposed by Bayar and Stamm [6] to display modification traces while reducing the quantity of content in an image. CNN and LSTM networks were coupled by Bappy et al. [10] to identify patterns of tampering that occur in both space and time. Multi-domain feature fusion in CNN frameworks improves detection accuracy on a variety of datasets, as demonstrated by Aminu et al. [12].

Using JPEG recompression artefacts to reduce expenses, Ali et al. [13] developed a lightweight CNN model that achieved 92.23% accuracy on the CASIA 2.0 dataset. A CNN-based system utilising block-level processing and transform-domain features was also created by Raghavendra et al. [14]. It achieved 97.7% accuracy on CASIA2 with an 80/20 train-test split. By integrating CNN with ELA and metadata features, Kumar et al. [15] were able to achieve 91% accuracy. This demonstrates the value of combining cues based on compression.

Although deep learning techniques are typically quite successful, they may not perform well on novel forms of manipulation or extensive post-processing, and they require a large amount of labeled data.

Hybrid approaches

Hybrid approaches combine learnt and hand-made elements to overcome the limitations of single methods. To make it more difficult to copy, relocate, or splice forgeries, Walia et al. [9] integrated CNN features with texture descriptors. A hybrid strategy that combines SURF and DCT was proposed by Ojeniyi et al. [16]. DCT enhances frequency representation, while SURF identifies keypoints. This approach is more effective against many attacks, such as compression, rotation, and noise.

A multiscale feature fusion framework that combines pretrained CNN features with traditional descriptors, such as HOG and color moments, was proposed by Badar et al. [17]. When clustering and spatial verification are applied, the approach yields very accurate results (92–95%) and is robust to geometric aberrations. To identify tiny compression artefacts, multi-stage feature extraction and ELA have also been extensively employed. Sari and Fahmi [18] and Singh et al. [19] demonstrated how multi-level ELA combined with texture and frequency-domain characteristics could enhance detection performance.

Transformer-based methods

Transformer-based designs have recently been applied in photo forensics. A system that finds and sorts fake areas using the Segment Anything Model (SAM) and Vision Transformers (ViT) was proposed by Pawar et al. [20]. With an accuracy rate of up to 98% on CASIA datasets, their technique proved effective for a variety of forgeries.

To put it briefly, handmade techniques are straightforward and effective, but they are less effective for complex modifications. Although deep learning techniques are more precise, they require large amounts of data and may not perform well with new data [21, 22]. Because they incorporate the best aspects of both methods, hybrid frameworks that blend deep and handcrafted features are a suitable option. The proposed approach makes picture forgery detection robust and accurate by combining CNN classification with DCT-SVD features, multi-quality ELA, and LBP. This is the suggested methodology.

3. PROPOSED METHODOLOGY

A basic CNN that can automatically distinguish between actual and fraudulent images serves as the foundation for the suggested method. By combining various feature representations, such as ELA, LBP, and DCT-SVD, the model creates a multi-channel input that functions for CNN-based learning. Training and testing are the two key components of the entire organization. Using the data it collects during training, the model learns to distinguish between authentic and fraudulent photos. In order to determine how well the trained model performs, it is tested with data that it has never seen before. An overview of the recommended training and testing process is illustrated in Figure 1.

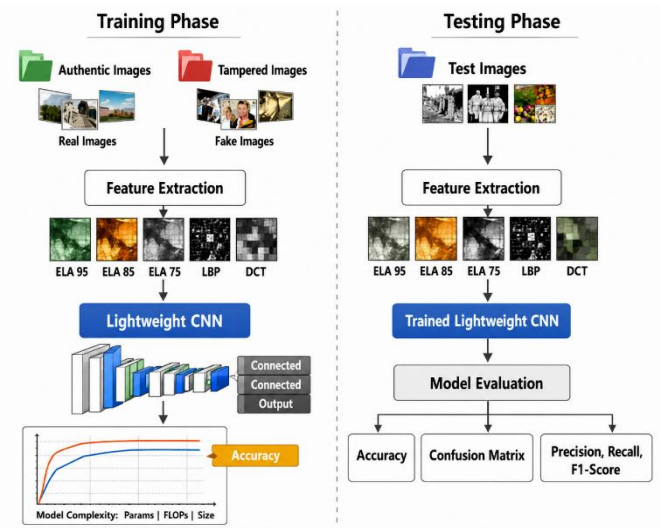

Figure 1. An overview of the proposed system that use a lightweight Convolutional Neural Network (CNN) and parallelised feature extraction to detect image manipulation

3.1 Preparing the dataset

A carefully selected dataset of authentic and modified images from publicly accessible locations is used in the proposed method:

• Authentic Images: Unaltered photos.

• Tampered images: are ones that have been changed by copy-move, splicing, or adding content.

To make it easier to extract features while keeping important tampering artifacts, all photos are turned into grayscale. By changing the size of each image to a fixed resolution of 128 × 128 pixels, the CNN's input dimensions are standardized:

$$I_{\text{resized}} = \text{resize}(I_{\text{original}}, 128 \times 128) \quad (1)$$

where, $I_{original}$ is the input image.

To split the dataset into training and testing sets, use Standard Split. 80% of the data is for training and 20% is for testing.

3.2 Feature extraction

To enhance the robustness of the proposed system, three complementary feature types are extracted and combined into a unified multi-channel representation: ELA, LBP, and DCT-SVD features. These features capture compression artifacts, local texture information, and frequency-domain characteristics, respectively. To improve computational efficiency, feature extraction is performed in parallel.

We put the three types of features into a 128 × 128 × 5 multi-channel tensor for CNN input. Parallel execution makes it possible to analyze data almost in real time on systems with GPUs, which cuts down on processing time by a lot.

3.2.1 Error Level Analysis

ELA reveals compression issues that may indicate the image has been altered. To determine ELA, this study used several JPEG quality levels (Q = 95, 85, and 75). Based on prior experience, these settings were selected to display both positive and negative recompression artefacts. While lower quality levels (e.g., 75) highlight compression discrepancies and make it easier to identify places that have been altered, higher quality levels (e.g., 95) preserve minute details.

The absolute difference between the original image and the recompressed image is the *ELA* map for each greyscale image (I_g):

$$ELA(I_g, Q) = |I_g - J(I_g, Q)| \quad (2)$$

The recompressed image at quality Q is $J(I_g, Q)$, and the absolute difference indicates regions that could be fabricated.

3.2.2 Local Binary Patterns

To identify texture changes in a limited area, LBP are utilized. In this study, LBP is computed using a radius (R = 1) and (P = 8) neighboring pixels. Uniform LBP patterns are adopted to reduce dimensionality while preserving discriminative texture information. The LBP value at each pixel is computed as:

$$\text{LBP}(i,j) = \sum_{n=0}^{7} s(P_n - P_c) \cdot 2^2 \quad (3)$$

where:

P_c is the brightness of the central pixel.

P_n = the intensity of the n-th neighbor.

$s(x)$ = 1 if $x \geq 0$; otherwise, 0.

LBP captures micro-texture differences that help identify fake artifacts.

3.2.3 Discrete Cosine Transform (DCT)-Singular Value Decomposition (SVD) features

Each image is divided into non-overlapping (8 × 8) blocks. The DCT is applied to each block to extract frequency-domain information. Subsequently, SVD is applied to each DCT block, and the largest singular value is retained as a representative feature.

This process captures subtle variations in frequency components that may result from tampering operations.

Finally, the extracted ELA, LBP, and DCT-SVD feature maps are resized (if necessary) and concatenated to form a multi-channel tensor of size (128 × 128 ×5), which is used as input to the CNN model.

3.3 Lightweight Convolutional Neural Network architecture

The model receives as input a multichannel feature tensor generated through the fusion of ELA, LBP, and DCT-SVD features, depending on the selected feature configuration, the input tensor has one of two possible forms. In the grayscale ELA case, the tensor consists of three channels (H × W × 3), formed by combining one ELA channel, one LBP channel, and one DCT-SVD channel. In the color ELA case, the tensor consists of five channels (H × W × 5), where three channels correspond to RGB-based ELA features in addition to the LBP and DCT-SVD channels.

To accommodate these configurations, the input layer of the CNN is defined according to the number of channels in the fused tensor.

The network begins with an input layer and proceeds via a sequence of convolutional blocks, as shown in Figure 2. Each block consists of a convolutional layer, batch normalisation, and a ReLU activation function. By progressively extracting valuable forensic patterns from the fused feature tensor, these layers enable the detection of minute alterations brought about by image manipulation.

The feature maps are then reduced in size by an average pooling layer, speeding up computations without sacrificing crucial data. To further enhance feature representation, an additional convolutional layer is implemented after the pooling stage. The input dimensionality disparities between the 3-channel and 5-channel configurations are compensated for by this layer.

After features are removed, spatial data is combined into a compact feature vector using a Global Average Pooling (GAP) layer. By eliminating the need for large, fully connected layers, this approach reduces the number of trainable parameters and the risk of overfitting.

Finally, the resulting feature vector is passed to a softmax classification layer, which outputs the probability of the input image belonging to either the authentic or tampered class. Overall, the architecture shown in Figure 2 maintains a balance between computational efficiency and classification performance while supporting flexible input representations of varying channel dimensions.

Training Hyperparameters:

- Optimizer: Adam
- Learning rate: 3×10^{-4}
- Epochs: 80
- Mini-batch size: 16
- Early stopping: Training stops if validation accuracy reaches 99% or shows no improvement for 5 consecutive epochs.

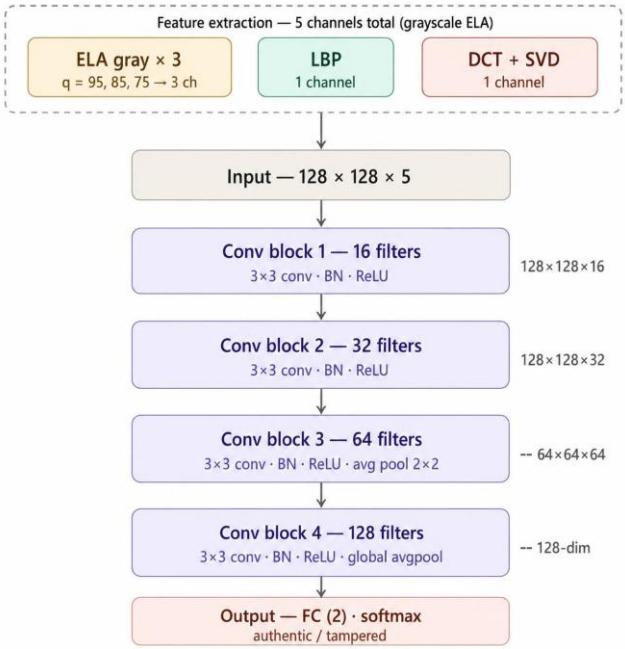

Figure 2. Structure of the proposed lightweight Convolutional Neural Network (CNN) architecture

3.4 Training and testing workflow

The overall workflow for training and testing the CNN-based image forgery detection system is divided into two phases: training and testing. The training phase focuses on feature extraction, model learning, and optimization, while the testing phase evaluates the trained model on unseen images to measure its performance. Algorithm 1 (Training Workflow) and Algorithm 2 (Testing Workflow) show the steps in detail.

3.4.1 The training phase

During training, you extract features, add to the data, train the CNN, and keep an eye on how well it works. Algorithm 1 gives a brief overview of the whole process.

Algorithm 1: Training Workflow

Input: B is the batch size, α is the learning rate, E is the number of epochs, and D_train is the training dataset.
Output: The trained CNN model M, along with graphs of accuracy and training loss.

(1) Feature Extraction: Get multi-channel features like RGB, LBP, and ELA for each image in D_train.
(2) Data augmentation: Use methods like flipping, rotating, and scaling to make the data more interesting.
(3) Training the Model:
  (a) Configure the CNN settings.
  (b) For every era between 1 and E:
  i. For any interval between 1 and E: i. Sort the data into B-sized groups.
  ii. Assign each batch CNN features.
  iii. Use the loss function you picked to find the loss.
  iv. Use the Adam optimiser to adjust the weights.
(4) Keeping track of progress: For each epoch, write down the accuracy and training loss curves.
(5) Saving the model: Save the trained model M so you can look at it again later.
(6) Timing Analysis: Keep track of how long each epoch takes on average and how long the whole training takes.

3.4.2 The testing phase

To assess the model's overall performance, images that it has never seen before are used. Algorithm 2 provides all the information you require regarding how things operate.

Algorithm 2: Testing Workflow

INPUT: A trained CNN model M and a test dataset D_test
Results: Evaluation metrics and Y_pred forecasts

(1) Feature Extraction: Obtain the same multi-channel features that were utilised for training from each image I in D_test.
(2) Enter the features you obtained from the trained CNN model M to obtain the predicted labels Y_pred. Check the fit between the Y_true and Y_pred labels.
(3) Determine how to quantify the assessment:
  - The number of photographs that were correctly classified is known as accuracy.
  - Divide the number of actual positives by the number of anticipated positives to determine the precision.
  - Divide the total number of positives by the number of genuine positives to determine recall.
  - The F1-score is obtained by taking the harmonic mean of recall and precision.Confusion Matrix: To look at things more closely, make a table with true positives, false positives, true negatives, and false negatives.
(4) Outcomes of Reporting: Give a general idea of the metrics and, if you want, use charts or graphs to help people understand them.
(5) Timing Analysis: Record how long it takes to test each image and how long it takes to test all of them to determine how well the model performs in real life.

4. EXPERIMENT EVALUATION

4.1 Constructing the experiment

To speed up training and inference, we used GPU acceleration in MATLAB (Deep Learning Toolbox) for all of our testing. The proposed lightweight CNN uses features like ELA, LBP, and DCT-SVD to integrate data from several sources and support forensic investigators. Two popular benchmark datasets in digital image forensics were used to test the system:

• CASIA1
• CASIA2

To save money on computing and keep things the same, all images were turned to black and white and shrunk to 128 × 128 pixels before feature extraction.

We used the following tests to rate performance: how well the computer worked and how well it classified things:

• Total Accuracy
• Precision
• Recall
• F1-score
•The Confusion Matrix
• Time Taken to Extract Features
• Time to Classification
• The total duration of training

Several distinct experimental trials were performed to assess stability and reproducibility.

4.2 Experimental results on CASIA1

Five independent experiments were conducted on the CASIA1 dataset to evaluate the stability, repeatability, and generalization capability of the proposed method, as presented in Table 1.

• During trials, the accuracy consistently nears 98%.

• The proposed lightweight CNN multi-feature fusion has a strong ability to tell the difference between things, and this ability is not affected by changes in training. Figure 3 shows the confusion matrix for CASIA1. It shows that most real and altered images are correctly identified, with only a few being misclassified.

4.3 Experimental results on CASIA2

Five independent experiments were conducted on the CASIA2 dataset to assess the stability, repeatability, and generalization capability of the proposed method, as reported in Table 2.

• Accuracy, which stays above 96% all the time, shows that generalization is

strong.

• CNN inference on a GPU is very fast (about 0.013 seconds per image), but parallelized feature extraction takes up most of the processing time (about 0.17 seconds per image).

• It is clear that the proposed method is suitable for large forensic analysis pipelines.

Table 1. Overall performance on CASIA1

Test	Accuracy (%)	Macro Precision (%)	Macro Recall (%)	Macro F1-Score (%)	Total Testing Time (s)
Test 1	98.26	98.22	98.29	98.25	17.78
Test 2	98.26	98.29	98.21	98.25	19.97
Test 3	97.97	97.90	98.06	97.96	19.05
Test 4	98.11	98.05	98.18	98.11	18.64
Test 5	98.04	97.98	98.12	98.05	18.92
Average	98.13	98.09	98.17	98.12	18.87

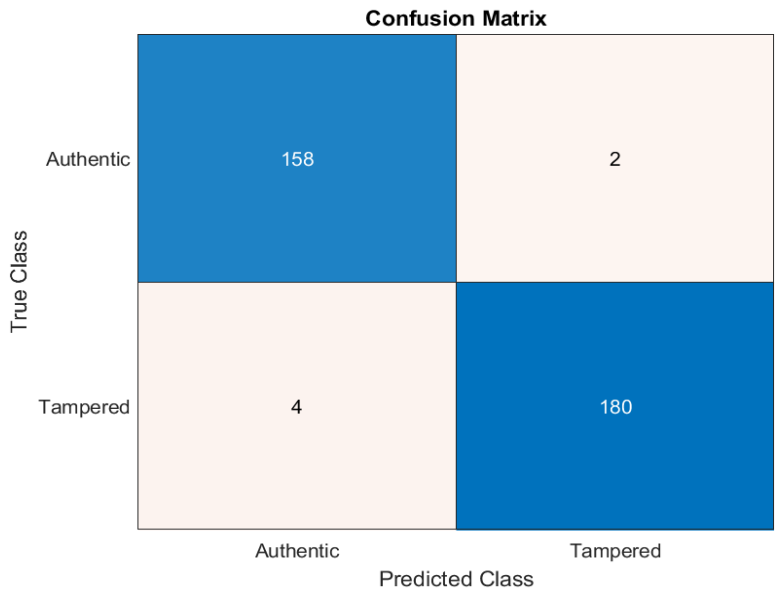

Figure 3. Confusion matrix of the proposed model on the CASIA1 dataset

Table 2. Overall performance on CASIA2

Test	Accuracy (%)	Macro Precision (%)	Macro Recall (%)	Macro F1-Score (%)	Total Testing Time (s)
Test 1	96.63	96.49	97.53	97.50	369.99
Test 2	96.81	96.65	97.60	97.63	372.10
Test 3	96.55	96.42	97.35	97.38	368.44
Test 4	96.92	96.78	97.66	97.72	374.26
Test 5	96.48	96.35	97.29	97.32	367.85
Average	96.68	96.54	97.49	97.51	370.53

4.4 Computational efficiency and model complexity analysis

This section examines the effectiveness of the suggested framework and the complexity of the lightweight CNN model's architecture to provide a complete picture of how well it would function in actual digital forensics scenarios.

The average processing time for each image, including feature extraction and classification, is displayed in Table 3. Features are extracted using block-wise DCT-SVD, ELA, and LBP. The data is classified using the lightweight CNN model.

The primary computational barrier in the multi-stage processing pipeline is feature extraction, as Table 3 illustrates. Specifically, compared to the CNN inference stage, ELA, LBP, and DCT-SVD processes require higher processing resources. However, because the network is light and the GPU

accelerates it, classification is really quick. Each photograph simply takes a few milliseconds.

The architecture of the suggested CNN, which provides a structural perspective on efficiency, is displayed in Table 4.

With just 97,986 parameters and a size of 0.36 MB, Table 4 demonstrates how little the suggested model is in terms of memory and parameter size. Strong representation through multi-feature input fusion is made possible by this small design, which also makes deployment and storage considerably simpler.

Because of GPU acceleration and improved convolution processes, the model's 1.56 GFLOP cost is still affordable. When paired with quick inference time, the suggested system achieves a decent balance between accuracy and computational cost.

Overall, the findings in Tables 3 and 4 demonstrate the computational efficiency of the proposed approach at both the feature-processing and model-architecture levels. This implies that it can be applied to large-scale image forgery detection programs that must operate in real-time or almost real-time.

Table 3. The suggested method's cost per image

Dataset	Feature Extraction Time (s/image)	Classification Time (s/image)
CASIA1	0.05	0.002–0.003
CASIA2	0.177	0.013

Table 4. Details regarding the suggested lightweight CNN model's architecture and computational complexity

Feature	Lightweight CNN Model
Input Channels	5 (ELA + LBP + DCT-SVD)
Convolution Layers	5 layers (16, 32, 64, 128 filters)
Pooling	Average Pooling (1 layer)
Fully Connected Layers	1 layer (2 classes)
Total Parameters	97,986
Model Size	0.36 MB
FLOPs	1,561

Note: Convolutional Neural Network (CNN); Floating Point Operations (FLOPs)

4.5 Conversation

The findings of the experiment demonstrate that the proposed technique is robust and adaptable, since it performs well on both the CASIA1 and CASIA2 datasets. On the simpler CASIA1 dataset, the model's average accuracy is 98.13% ± 0.13%, whereas on the more difficult CASIA2 dataset, it is 96.68% ± 0.18%. The slight variations between runs demonstrate that the model is stable and unaffected by modifications to the distribution or setup of the training data.

On both datasets, the recommended approach maintains high values for accuracy, macro precision, macro recall, and macro F1-score. The model performs well on both actual and manipulated images and does not favour one over the other, as evidenced by the balance between precision and recall. Because it reduces the percentage of forgeries that go undetected, the model's strong recall values demonstrate that it can identify manipulated photographs, which is crucial for digital forensics.

When compared to alternative approaches, the suggested model either surpasses or is comparable to both conventional CNN-based and hybrid CNN + ELA techniques. While some approaches demonstrate comparable accuracy on CASIA2, they don't necessarily perform well across all evaluation criteria or datasets. However, the recommended method is effective and reliable, demonstrating its strength and versatility.

Despite using multi-feature fusion, the method is nonetheless computationally efficient. CASIA2 requires more time to test because to its larger dataset. The total processing time is still adequate for real-world forensic applications where accuracy and reliability are more crucial than minute variations in runtime. Because the standard deviation is not particularly big when repeating the same experiment, the proposed framework is even more dependable.

The findings demonstrate that integrating various feature representations improves the model's ability to distinguish between items. Because the approach employs many features, it is more effective at identifying both structural faults and compression artefacts. Consequently, the suggested approach is effective against a variety of picture forgeries, including splicing and copy-move attacks. It can therefore be applied to practical digital forensic tasks.

4.6 A comparison with alternative approaches

We evaluated the effectiveness of the proposed image forgery detection algorithm against the state-of-the-art techniques using common CASIA datasets. The datasets, experimental setup, and performance metrics—accuracy, precision, recall, F1-score, and runtime—are listed in Table 5.

Table 5. Comparison of the performance of proposed and existing methods on CASIA datasets

Dataset	Total # Images	# Images Used	Dataset Details	Technique	Accuracy (%)	Macro Precision (%)	Macro Recall (%)	Macro F1-Score (%)	Run Time / Testing Time
CASIA1	1725	345	750 Real / 975 Tamper	ViT [20]	98.00	-	-	-	-
CASIA2	12614	2522	7491 Real / 5123 Tamper	ViT [20]	98.00	-	-	-	-
CASIA2	12614	2522	7491 Real / 5123 Tamper	CNN [17]	92.30	85	97	91	-
CASIA2	12614	4795	1701 Real / 3274 Tamper	CNN [14]	97.70	-	-	-	-
CASIA2	12614	2523	1498 Real / 1025 Tamper	CNN + ELA [19]	91.00	-	-	-	-
CASIA1	1725	1725	750 Real / 975 Tamper	Proposed method	98.13 ± 0.13	98.09 ± 0.16	98.17 ± 0.09	98.12 ± 0.13	18.87 ± 0.80 s
CASIA2	12614	12614	7491 Real / 5123 Tamper	Proposed method	96.68 ± 0.18	96.54 ± 0.17	97.49 ± 0.15		

The results show that the suggested method works well on the CASIA1 and CASIA2 datasets. The proposed strategy achieved an accuracy of 98.13% ± 0.13% on the CASIA1 dataset, surpassing previous ViT-based methods (98.00%) and CNN approaches (92.30%). The proposed method achieved a score of 96.68% ± 0.18% on CASIA2, comparable to the CNN approach (97.70%) and superior to the CNN + ELA method (91.00%). Also, the suggested method is very stable when it comes to macro accuracy, recall, and F1-score, with values above 96% in both datasets. The model does a great job of telling the difference between real and fake photos, as shown by the AUC values of 0.9823 (CASIA1) and 0.9875 (CASIA2).

The larger dataset size on CASIA2 means that it takes longer to run (370.53 ± 2.61 s), but it is still useful for real-world applications where accuracy is very important.

The results of the comparison show that the proposed method is more accurate and reliable than traditional CNN and CNN + ELA methods. It also works well on a variety of datasets and types of forgery, such as splicing and copy-move.

The comparison in Table 4 is conducted on the same benchmark datasets with consistent dataset splits and the same number of images, ensuring a fair evaluation in terms of data partitioning. However, it should be noted that some referenced methods may apply different preprocessing or feature extraction pipelines depending on their original implementations. Despite this, the comparison remains valid as all methods are evaluated on standard CASIA benchmarks using widely accepted evaluation metrics.

5. CONCLUSION

We suggested a reliable CNN-based method for identifying fake photos, improved by ELA and multi-channel feature extraction. Large benchmark datasets like CASIA1 and CASIA2 were used to train and evaluate the system. Finding several fakes, including copy-move and splicing, was a great success.

The proposed approach outperformed the majority of current approaches, such as CNN and ViT models. It also has great recall, accuracy, precision, and F1-score. The Adam optimizer's capacity to enhance training, data augmentation, and multi-channel feature extraction all contributed to improved detection. You can also perform training, testing, and timing analysis in an organised manner by following the methods in Algorithms 1 (Training) and 2 (Testing).

The experimental findings demonstrate that, even when the images are altered, the proposed technique can consistently distinguish between authentic and fraudulent photos. The model may be helpful for media verification, digital forensics, and law enforcement because of its excellent accuracy, particularly on CASIA2 (96.68%).

REFERENCES

[1] Fridrich, J., Soukal, D., Lukáš, J. (2003). Detection of copy-move forgery in digital images. In Proceedings of Digital Forensic Research Workshop, pp. 652-663.

[2] Ng, T.T., Chang, S.F. (2004). A Markov random field model for image splicing detection. Proceedings of ICIP.

[3] Farid, H. (2009). Exposing digital forgeries from JPEG ghosts. IEEE Transactions on Information Forensics and Security, 4(1): 154-160. https://doi.org/10.1109/TIFS.2008.2012215

[4] Ojala, T., Pietikainen, M., Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7): 971-987. https://doi.org/10.1109/TPAMI.2002.1017623

[5] Popescu, A.C., Farid, H. (2005). Exposing digital forgeries by detecting traces of resampling. IEEE Transactions on Signal Processing, 53(2): 758-767. https://doi.org/10.1109/TSP.2004.839932

[6] Bayar, B., Stamm, M.C. (2016). A deep learning approach to universal image manipulation detection using a constrained 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

[7] de Carvalho, T.J., Riess, C., Angelopoulou, E., Pedrini, H., de Rezende Rocha, A. (2013). Exposing digital image forgeries by illumination color classification. IEEE Transactions on Information Forensics and Security, 8(7): 1182-1194. https://doi.org/10.1109/TIFS.2013.2265677

[8] Man, Q., Gee, S.J., Cho, Y.I. (2026). Multi-domain perception transformer for generalized forgery image detection. Applied Sciences, 16(1): 533. https://doi.org/10.3390/app16010533

[9] Walia, S., Kumar, K., Kumar, M., Gao, X.Z. (2021). Fusion of handcrafted and deep features for forgery detection in digital images. IEEE Access, 9: 99742-99755. https://doi.org/10.1109/ACCESS.2021.3096240

[10] Bappy, J.H., Simons, C., Nataraj, L., Manjunath, B.S., Roy-Chowdhury, A.K. (2019). Hybrid LSTM and encoder–decoder architecture for detection of image forgeries. IEEE Transactions on Image Processing, 28(7): 3286-3300. https://doi.org/10.1109/TIP.2019.2895466

[11] Alkawaz, M.H., Sulong, G., Saba, T., Rehman, A. (2018). Detection of copy-move image forgery based on discrete cosine transform. Neural Computing and Applications, 30: 183-192. https://doi.org/10.1007/s00521-016-2663-3

[12] Aminu, A.A., Agwu, N.N., Adeshina, S., Ahmed, M.K. (2022). Detection of image manipulation with convolutional neural network and local feature descriptors. TELKOMNIKA (Telecommunication, Computing, Electronics and Control), 20(3): 629-637. https://doi.org/10.12928/telkomnika.v20i3.23318

[13] 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

[14] Raghavendra, C., Dodda, R., Sake, M., Nimmala, S. (2025). Advancements in CNN-based techniques for robust image forgery detection: Challenges and future directions. E3S Web of Conferences, 616: 02009. https://doi.org/10.1051/e3sconf/202561602009

[15] Kumar, S., Lodhi, S., Kamini. (2025). Image forgery detection using deep learning. In 2025 3rd International Conference on Disruptive Technologies (ICDT), Greater Noida, India, pp. 238-243. https://doi.org/10.1109/ICDT63985.2025.10986517

[16] Ojeniyi, J.A., Adedayo, B.O., Ismaila, I., Abdulhamid,

S.M. (2018). Hybridized technique for copy-move forgery detection using discrete cosine transform and speeded-up robust feature techniques. International Journal of Image, Graphics and Signal Processing, 10(4): 22-30. https://doi.org/10.5815/ijigsp.2018.04.03

[17] Badar, P., Geetha, G., Mahesh, T.R. (2025). Multiscale feature fusion for robust copy-move forgery detection in digital images. Journal of Information Systems Engineering and Management, 10(15s): 579-587. https://doi.org/10.52783/jisem.v10i15s.2496

[18] Sari, W.P., Fahmi, H. (2021). The effect of error level analysis on the image forgery detection using deep learning. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 6(3): 187-194. https://doi.org/10.22219/kinetik.v6i3.1272

[19] Singh, T., Goel, Y., Yadav, T., Seniaray, S. (2023). Performance analysis of ELA-CNN model for image forgery detection. In 2023 4th International Conference for Emerging Technology (INCET), Belgaum, India, pp. 1-6. https://doi.org/10.1109/INCET57972.2023.10170007

[20] Pawar, D., Gowda, R., Chandra, K. (2025). Image forgery classification and localization through vision transformers. International Journal of Multimedia Information Retrieval, 14: 8. https://doi.org/10.1007/s13735-025-00358-8

[21] Sun, S.Y., Meng, P.C. (2026). A lightweight conditional diffusion model for restoring turbulence-degraded facial images. Acadlore Transactions on AI and Machine Learning, 5(1): 1-10. https://doi.org/10.56578/ataiml050101

[22] Du, J., Fu, W., Zhang, Y., Wang, Z. (2024). Advancements in image recognition: A Siamese network approach. Information Dynamics and Applications, 3(2): 89-103. https://doi.org/10.56578/ida030202