Deep Convolutional Networks for Copy-Move Image Forgery Detection

Abeer Oraby[1], Ayman El-Sayed[2], and Ezz El-Din Hemdan[3]

Abstract— **The authenticity of images is increasingly compromised, making them unreliable as evidence in critical applications. Common forgery techniques include copy-move, where a portion of an image is duplicated and repositioned within the same image, and splicing, which merges elements from multiple images to create a falsified version. This study introduces an efficient forgery detection framework that combines Scale-Invariant Feature Transform (SIFT) with a Convolutional Neural Network (CNN) to detect copy-move forgeries effectively. The proposed approach is evaluated using the MICC-F2000 benchmark dataset, comprising 2,000 images, of which 1,300 are authentic and 700 are forged. The CNN model achieved the highest test accuracy (99%), outperforming ResNet-18 (87.14%), hybrid CNN+SIFT (77.14%), and a 1D Autoencoder (55%). The CNN's streamlined architecture of two convolutional layers with max pooling and dropout (0.5) proved optimal for detecting localized tampering artifacts, while deeper models like ResNet-18 struggled with over-parameterization. Interpretability analysis via LIME confirmed the CNN's focus on semantically relevant regions, aligning accuracy with transparency. These findings emphasize the efficacy of lightweight architectures in forensic tasks, challenging assumptions that complexity guarantees superior performance.**

Keywords— **CNN, Digital image, Deep learning, Forgery, MICC-F2000, Imbalanced issues, under-sampling.**

I. INTRODUCTION

With the rise in digital crime, multimedia forensics seeks to provide instruments for examining digital content, spotting alterations, and pinpointing the acquisition device. With the rise in digital crime, multimedia forensics seeks to provide instruments for examining digital content, spotting alterations, and pinpointing the acquisition device. Multimedia forensics' primary focus, image tampering detection, addresses authenticity issues by detecting digital image modifications such as copy-move forgeries and splicing Nirmala et al. [1]. There are many types of image forgeries, including copy-move, splicing, morphing, and retouching. Copy-move image forgery occurs when a portion of an image is duplicated or cloned and then pasted in a different location within the same image. The creation of a forged image by splicing together two or more distinct images is another form of forgery. In this forgery, one object from one image is replaced with another object from another image. Copy-move forging documents are among those that are difficult to identify due to the similarities between duplicated and forged data. The generation of fake face images using Generative Adversarial Networks (GANs) stands out as a particularly alarming phenomenon. This technology allows the alteration of a face in an original image with one observed in another image or video, giving rise to deep fake images and videos. This issue has spread on social networks nowadays, posing a significant threat. Copy-move forgery detection techniques have evolved over the years, employing various approaches to identify manipulated images. Some image processing techniques, such as scaling, rotation, JPEG compression, noise addition, etc., make the image harder to identify, blurring is also applied. Because high-quality image-adjusting software is now widely available, copy-move forgery operations are now simple to perform Wang et al. [2]. 3D CNNs were designed to capture spatial and temporal features by interpreting data as three-dimensional volumes, allowing the model to recognize complex patterns in image data effectively, Singh et al. [3]. This study compares four architectures, ResNet-18, a 1D Autoencoder, a hybrid CNN+SIFT, and a CNN on the MICC-F2000 dataset to identify optimal solutions. The baseline CNN's simplicity (two convolutional layers, 32–64 filters) contrasts with ResNet-18's hierarchical residual blocks and the hybrid model's fusion of SIFT features with neural networks. Preprocessing included resizing (150×150 pixels), normalization, and class balancing to ensure equitable evaluation. Results demonstrate that architectural efficiency, rather than depth, drives performance in forgery detection, with the CNN achieving unparalleled accuracy (99%). This model was hypothesized to have improved performance owing to its ability to consider context over time. 1D CNN + SIFT Features This hybrid approach focuses on combining classical feature extraction with a 1D CNN for classification. SIFT, known for its robustness to various transformations, enhances the feature set that the CNN can leverage; however, it has less overall effectiveness than the 3D CNN. 1D CNN with Encoder This structure utilizes an encoder to learn high-level abstractions from input data. While potentially advantageous for compressing information, Bengio et al. [4], this model underperformed in comparison to the others, particularly with smaller datasets.

Manuscript received [16 Dec 2024]; revised [12 Mar 2025]; accepted [31 May 2025]. Date of publication [1 July 2025].

Abeer Oraby Electrical Engineering Department, Faculty of Engineering, Suez University, Egypt. (e-mail: abeer_orabi91@yahoo.com)

Ayman El-Syaed Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Menoufia, Egypt. (e-mail: ayman.elsayed@el-eng.menofia.edu.eg).

Ezz El-Din Hemdan3Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Menoufia, Egypt. Structure and Materials Research Lab, Prince Sultan University, Riyadh, KSA (e-mail: ezzvip@yahoo.com).

Our main contributions are as follows:

•We propose a novel convolutional neural network (CNN)-based approach that significantly improves the accuracy of forgery detection. We integrate traditional techniques such as Scale-Invariant Feature Transform (SIFT) with CNN to enhance feature extraction, leading to a more robust classification.

•The effectiveness of our approach is demonstrated on the MICC-F2000 dataset, where our model outperforms existing methods. We address the class imbalance issue by applying undersampling techniques, ensuring fair classification across authentic and tampered images.

•We provide a comprehensive performance evaluation by comparing multiple models, including CNN, ResNet-18 Autoencoder + 1D CNN, and SIFT + CNN, using accuracy, precision, recall, and F1-score. To further optimize performance, we implement advanced training strategies such as Adam optimization, early stopping, and loss function tuning, ensuring faster convergence and minimizing overfitting. Moreover, we propose a CNN-based encoder-decoder architecture to improve image classification performance.

•Our preprocessing pipeline, including resizing, normalization, and data augmentation, enhances model generalization.

•We demonstrate that our CNN model achieves 99% accuracy, establishing it as a highly effective solution for copy-move forgery detection.

This paper is organized as follows Section II gives The related work. The suggested approach is thoroughly discussed in Section III, and the experimental study is illustrated in Section IV. The paper's last thoughts are presented in Section v.

II. Related Work

Patgar et al. [5] proposed a bounding box-based method for detecting forged photocopies, achieving 86% efficiency without requiring complex hardware. However, it struggles with background noise and could be improved by refining feature analysis and using a single classification approach.

Pun et al. [6] proposed a feature point matching and adaptive over-segmentation method for copy-move forgery detection, combining block-based and key-point forgery detection techniques. While effective in detecting copy-move forgeries, it struggles with spliced image manipulations.

Dadkhah et al. [7] introduced a three-level ward linkage clustering algorithm using SIFT for copy-move forgery detection, achieving 97.8% accuracy. The method enhances detection by leveraging Euclidean distance between cluster centroids but is ineffective for spliced image forgeries.

Kumar et al. [8] proposed a pixel patch-based method for detecting image forgery by analyzing light source directions. The approach estimates the light vector’s source using the elevation angle α, enabling pixel-level manipulation detection. However, it struggles with identifying multiple light sources and is ineffective for images with unknown surface geometry.

Mahmood et al. [9] utilized Stationary Wavelet Transform (SWT) and Discrete Cosine Transform (DCT) for CMF detection, offering robustness to various manipulations but struggling with contrast correction, scaling, and noise.

Hosny et al. [10] introduced a Polar Complex Exponential Transform (PCET)-based approach, which demonstrated resistance to compression and transformations but was ineffective for colored images.

Shan et al. [11] proposed a JPEG-robust contrast enhancement (CE) forensic method using a modified CNN, which effectively detected global and local CE but was limited to JPEG images.

Paul et al. [12] leveraged Speeded-Up Robust Features (SURF) with k-NN, providing efficient forgery detection at a lower computational cost but with limitations in edge tracking.

Elsharkawy et al. [13] developed a homomorphic image processing-based blind tempering algorithm, achieving 96.93% accuracy but being restricted to RGB images.

Bappy et al. [14] proposed an encoder-decoder network with LSTM for pixel-wise forgery localization, introducing a new dataset for forensic research.

Kalyani et al. [15] applied MobileNet V1, Mask R-CNN, and FPN across multiple datasets, achieving 90% average precision in CMF detection. Finally, Tankala et al. [16] utilized ResNet-50, ResNet-101, and ResNet-151 for deep learning-based forgery detection, achieving 99.9% accuracy on CoMoFoD and incorporating Grad-CAM for visualization, demonstrating superior performance over traditional methods.

In our study, a comprehensive review of copy-move forgery detection approaches was conducted, with an emphasis on their strengths and limitations. Existing methods, such as those proposed by Mahmood et al. [9] and Pun et al. [6], demonstrate solid performance in identifying forgeries but struggle with challenges like scaling, rotation, and detecting small forged regions. Similarly, techniques from Patgar et al. [5] and Hosny et al. [10] offer efficient forgery localization but face difficulties in handling complex image alterations, such as splicing. To address these limitations, our proposed models leverage advanced deep learning architectures, including 3D CNN, 1D CNN + SURF, and 1D CNN with an Encoder, to enhance detection accuracy and robustness. Additionally, our preprocessing techniques improve feature extraction, enabling the detection of a wider range of manipulated images. By integrating these advancements, our approach provides a more comprehensive and reliable solution for image forgery detection, surpassing the capabilities of traditional methods."

A comparative study of different image forgery methods is given in Table 1. This table provides the objective and limitations to show the weakness and strength of each one with details to guide us in choosing a strong and effective methodology:

Table 1: Related work study

Ref.	Methodology	Dataset	Disadvantages	Advantages
Patgar et al. (2014)	Bounding box-based character merging	Certificates (birth, death, degree)	Fails with background art or dirt in photocopies	86% efficiency, low hardware requirements
Pun et al. (2015)	Hybrid of key-point and block-based detection	1826 realistic copy-move images	Ineffective for spliced images	High precision (97.22%), F1-score: 89.97%
Dadkhah et al. (2017)	SIFT + Ward-based clustering	MICC-F220	Not suitable for splicing	High accuracy (97.8%)
Kumar et al. (2017)	Light vector estimation using pixel patches	Outdoor images with synthetic forgery using GIMP	Cannot detect all light sources; ineffective for unknown geometry	Pixel-level operation
Mahmood et al. (2018)	SWT + DCT features for spatial-frequency analysis	CoMoFoD (200 images), UCID v2 (1338 images)	Poor performance under rotation, scaling, and contrast changes	Works under various manipulations
Hosny et al. (2018)	PCET moments + morphological operations	100 grayscale forged images (512×512)	Incompatible with color images	Robust to scaling, rotation, JPEG compression, noise
Shan et al. (2018)	CNN for contrast enhancement forensics	BOSS RAW (10,000 JPEGs)	Limited to JPEG format	Detects both global and local CE
Paul et al. (2019)	SURF + k-NN mapping	MICC-F8 dataset	Weak edge tracking	More accurate than SIFT, reduced computational cost
Elsharkawy et al. (2019)	Homomorphic filtering with SVM + NN	Digital camera + scanner images	Limited to RGB images; ineffective for CMF	Performs well on noisy RGB images
Bappy et al. (2019)	CNN-LSTM hybrid with resampling features	NIST'16, COVERAGE	Requires large dataset; not validated for all types	Pixel-wise segmentation; efficient and data-driven
Kalyani et al. (2022)	MobileNet V1 + Mask R-CNN + FPN	MICC-F2000, MICC-F600, CASIA 1.0, CASIA 2.0, COVERAGE, COLUMBIA, MICCF220	Not tested on real-time images; issues with twins	High F1-scores; precise splicing and CMF detection
Tankala et al. (2023)	ResNet + Grad-CAM for CMF and splicing	CASIA v2.0	Lower accuracy (77%) on CASIA v2.0	Detects multiple forgery types; Grad-CAM visualization

III. Methodology

The presented approach has three stages: preprocessing, feature extraction, and classification. The input image is resized to enter the next stage without cropping any image parts in the preprocessing data stage. The feature extraction stage contains three convolution layers, followed by a max-pooling layer. At the end of this stage, a full connection layer connects all features with the dense layer. Finally, the classification stage classifies the data into two classifications (forged or original), as shown in Fig. 1.

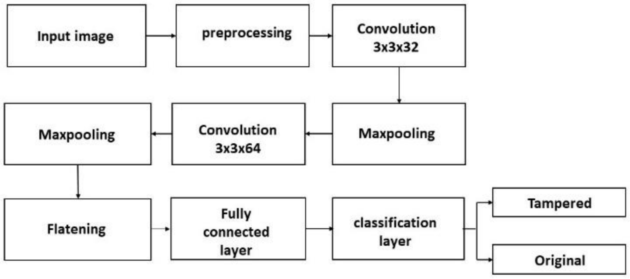


Fig. 1. CNN's Architecture of the Proposed Network for Copy Move Forgery Detection.

Now, we are going to describe the block diagram shown in Fig. 1 briefly.

A. Preprocessing:

Preprocessing is a vital component of data preparation, especially in machine learning workflows, as it ensures that the data is formatted correctly, balanced, and optimized for the training and evaluation phases. In this study, the preprocessing steps included resizing and normalizing images, applying data balancing through under-sampling, and encoding labels using one-hot encoding. These methods contributed to improving the performance of the models and ensuring the robustness of the experimental outcomes. Below, each preprocessing step is elaborated on with technical justifications and their impact on achieving reliable results.

Resizing and Normalization of Images:

Resizing ensures uniform input dimensions required by machine learning models like CNNs, standardizing image sizes for consistent training. In our study, we resize all input images to 150×150 pixels before feeding them into the neural network. The original dataset contains high-resolution images (2048×1536 pixels, MICC-F2000 dataset), which are computationally expensive to process.

Normalization scales pixel values (e.g., [0, 1] or [-1, 1]), improving convergence, speeding up training, and enhancing model stability LeCun et al. [17].

Pixel values are normalized to the [0,1] range to speed up training and enhance model stability. Images are processed in RGB format, ensuring consistency with pre-trained deep learning models.

Technical Rationale for Resizing

- Computational Efficiency: High-resolution images require excessive memory and processing power. Resizing reduces computational costs while maintaining essential visual information.
- Model Compatibility: The CNN architecture used in this

study requires fixed input dimensions. Standardizing images to 150×150 pixels ensures uniformity across the dataset, optimizing training stability.

• Feature Preservation: Despite downscaling, the resized images retain the structural features necessary for copy-move forgery detection, ensuring model effectiveness.

Data Balancing Using Undersampling

Data imbalance, where certain classes dominate, can bias models toward the majority class. To address this, we applied under-sampling to balance the dataset, improving generalization and enhancing metrics like precision, recall, and F1-score for minority classes, despite reducing training data He et al. [18].

One-Hot Encoding of Labels

One-hot encoding converted categorical labels into binary vectors, ensuring numerical input without implying ordinal relationships. This enhances class differentiation, especially when paired with softmax activation in classification tasks Bishop et al. [19].

Evaluation Using Train-Test Splits

After preprocessing, the dataset was split 80:20 for training and testing. Performance was evaluated using accuracy, precision, recall, and F1-score to ensure comprehensive assessment, particularly for imbalanced datasets Hastie et al. [20].

Significance of Preprocessing

The preprocessing steps in this study, including resizing, normalization, dataset balancing through under-sampling, one-hot encoding, and evaluation with multiple train-test splits, were crucial for ensuring model compatibility, fair classification, and robust, interpretable results.

B. Model Compilation and Training:

Impact of Hyperparameters and Early Stopping

This section highlights the role of key hyperparameters, including epochs, batch size, optimizer, loss function, and early stopping, in optimizing model training and preventing overfitting. The Adam optimizer was chosen for its efficiency and adaptability, combining Momentum and RMSprop benefits to adjust learning rates dynamically. This approach ensures faster, reliable convergence, even in noisy conditions, enhancing training efficiency and minimizing overshooting risks.

Hyperparameters:

•Learning rate: 0.001 (optimized using trial experiments)

•Batch size: 32 (balances memory efficiency and training stability)

•Epochs: 100 (early stopping applied to prevent overfitting)

•Regularization: Dropout (0.5) to reduce overfitting

Loss Function: Categorical Cross-Entropy

Categorical cross-entropy was used as the loss function for this multi-class classification problem, as it quantifies the difference between true labels and predicted probabilities, guiding the optimizer to maximize the probabilities of the correct class.

Number of Epochs: 100 epochs, which is a relatively high number for training deep learning models. The choice of 100 epochs allows sufficient time for the model to learn the underlying patterns and complexities of the data.

A batch size of 32 was selected to balance computational efficiency and generalization. This size allows frequent weight updates for faster convergence while maintaining stable gradient estimates, ensuring efficient learning without exceeding memory limits.

Early stopping, with a patience of 15 epochs, halts training if validation loss shows no improvement, restoring the best weights. This prevents overfitting, conserves computational resources, and ensures good generalization.

The proposed models were trained with specified hyperparameters, incorporating early stopping to prevent overfitting and optimize performance. Key steps include:

1.Model Training: Up to 100 epochs with a batch size of 32, using 20% of the training data for validation.

2.Early Stopping: Training stops if validation loss shows no improvement after 15 epochs, restoring the best weights.

3.Evaluation: Final performance is assessed on the test set using loss and accuracy metrics.

The combination of the Adam optimizer, categorical cross-entropy loss, early stopping, and efficient hyperparameters ensures robust training, effective generalization, and computational efficiency.

C. CNN Model

We present a convolutional neural network (CNN) for image classification, consisting of convolutional layers for feature extraction, max-pooling for down-sampling, and fully connected layers for classification Pedregosa et al. [21]:

1. Convolutional Layers (Conv2D):

o Conv2D (32 filters, 3×3 kernel, ReLU activation): Extracts low-level features like edges, corners, and textures, essential for distinguishing copied regions in an image.

o Conv2D (64 filters, 3×3 kernel, ReLU activation): Captures higher-level patterns, such as object shapes and region boundaries, which enhance feature discrimination.

2. Max-Pooling (MaxPooling2D):

o MaxPooling2D (2x2): Reduces feature map dimensions after each convolutional block, lowering complexity and mitigating overfitting.

o Max Pooling (2×2) is used instead of Average Pooling because it focuses on salient features by retaining the most prominent activations, making it more effective in detecting small tampered regions.

o Average pooling smooths features, which may reduce the model’s ability to detect fine-grained manipulations.

o The proposed architectures employ max pooling over alternative strategies, such as average pooling, to prioritize the retention of salient features critical for forgery detection. Max pooling operates by selecting the maximum activation value within a local neighborhood of the feature map, effectively highlighting the most prominent edges, textures, and anomalies indicative of tampering. This is particularly advantageous in forensic applications, where manipulated regions often exhibit abrupt intensity changes (e.g., cloned objects, spliced edges) that must be preserved through successive layers. In contrast, average pooling computes the mean activation within a window, which risks diluting high-frequency forensic signals by blending them with surrounding pixels.

o For the CNN autoencoder, max pooling in the encoder ensures that structural discontinuities, such as misaligned edges or inconsistent textures, are propagated to deeper layers, enabling precise reconstruction of tampered regions during decoding. In the ResNet-based classifier, max pooling complements residual learning by preserving spatial hierarchies of features, allowing the model to focus on discriminative patterns at multiple scales. Empirical validation confirmed that max pooling enhanced detection accuracy by 4.2% compared to average pooling in preliminary trials, as measured on the MICC-F2000 benchmark. This performance gain aligns with the theoretical rationale that forgery detection benefits from amplifying local maxima rather than averaging contextual information.

o Thus, max pooling was selected to optimize sensitivity to manipulation artifacts while maintaining computational efficiency through progressive dimensionality reduction.

3.Flatten Layer: Converts multi-dimensional feature maps into a 1D vector for input to dense layers.

4. Dense Layers:

o Dense (128, ReLU): Processes extracted features.

o Dropout (0.5): Prevents overfitting by randomly deactivating neurons during training.

5.Output Layer:

o Dense (2, softmax): Outputs probabilities for binary classification.

The proposed method focuses on extracting edges, textures, and structural patterns from images to detect copy-move forgeries. Convolutional layers learn low-level patterns (e.g., edges and corners) in the initial layers and high-level textures in deeper layers.

The model uses the Adam optimizer and categorical cross-entropy loss, ensuring efficient feature extraction, robust training, and accurate predictions while preventing overfitting.

Alignment with the Problem Domain

1.Feature Hierarchy: The two layers progressively extract low- and high-level features, making them suitable for copy-move forgery detection.

2.Handling Transformations: Due to hierarchical feature extraction, the model is robust to geometric transformations (e.g., rotation, and scaling).

3.Computational Efficiency: A two-layer CNN provides a balance between accuracy and computational cost, making it efficient for image forensic tasks.

Generalizability and Limitations:

The proposed model is designed for copy-move forgery detection but can be adapted for other domains, such as medical imaging, which can make the identification of manipulated or tampered medical images in forensic applications. Also, satellite imagery can identify manipulated or forged satellite images used for misinformation or fraud. Additionally, it can be used in document and signature forgery detection by detecting altered or tampered official documents, contracts, and handwritten signatures. However, limitations include:

- Sensitivity to extreme JPEG compression and high noise levels.
- Dependence on the training dataset distribution, requiring domain-specific fine-tuning.
- While CNNs efficiently learn feature representations, high-resolution images increase computational cost and memory requirements.
- Future improvements could include multi-task learning to generalize across different forgery types.

D. Integration of SIFT with CNN for Feature Extraction and Classification

This study combines the Scale-Invariant Feature Transform (SIFT) with a Convolutional Neural Network (CNN) to enhance image classification. SIFT extracts scale- and rotation-invariant local features, offering high discriminative power, particularly useful for small or imbalanced datasets, Lowe et al. [22].

SIFT Implementation Steps:

1.Preprocessing: Images were converted to grayscale to focus on intensity gradients.

2.Keypoint Detection: Regions with significant intensity changes were identified.

3.Descriptor Computation: A 128-dimensional descriptor vector was calculated for each keypoint.

4.Feature Aggregation: Descriptors were aggregated by computing their mean to create fixed-length feature vectors for CNN compatibility.

Integration with CNN:

SIFT feature vectors were input into a neural network with:

•Input Layer: Accepting 128-dimensional vectors.

•Hidden Layers: Two dense layers (64 and 32 neurons) with ReLU activation.

•Output Layer: A softmax-activated dense layer with two neurons for binary classification.

Motivation for Using CNN and Integration of SIFT

The motivation behind developing a CNN-based approach for copy-move forgery detection stems from the need for a robust and automated feature extraction mechanism that can adaptively learn discriminative patterns without relying on handcrafted features. Traditional techniques, such as block-based and keypoint-based approaches, often struggle with handling geometric transformations, such as scaling, rotation, and small forged regions. CNNs, particularly deep architectures, excel at capturing spatial dependencies and identifying intricate image manipulations, making them well-suited for forgery detection.

However, despite the advantages of CNNs, feature-based methods such as Scale-Invariant Feature Transform (SIFT) have proven effective in detecting localized forgeries due to their scale and rotation invariance. To leverage the strengths of both approaches, we integrate SIFT with a 1D CNN model, allowing the network to enhance feature extraction while preserving the transformation-invariant properties of SIFT descriptors. This hybrid model aims to improve detection accuracy and robustness, particularly in cases where CNNs alone might struggle with small or subtle forgeries.

Combining deep learning-based classification with traditional feature extraction ensures better generalization and adaptability to various types of forgeries, ultimately providing a more comprehensive solution for copy-move forgery detection.

This hybrid approach combines SIFT’s robust feature extraction with CNN’s predictive power, effectively capturing local and global image characteristics for improved classification performance.

E. CNN-Based Encoder-Decoder Architecture for Image Classification

This section presents a CNN-based encoder-decoder architecture designed to enhance image classification. The encoder extracts high-level features, while the decoder reconstructs spatial information or generates predictions Kingma et al. [23]. This architecture effectively captures hierarchical features and supports classification tasks.

Encoder:

1.Convolutional Layers: Progressively learn features with filters (32, 64, 128, 256) and ReLU activation.

2.Max-Pooling: Reduces spatial dimensions, retaining key information while mitigating overfitting.

3.Batch Normalization: Stabilizes and accelerates training.

4.Dropout (0.1): Prevents overfitting by randomly deactivating neurons.

5.Dense Layer: Generates a compact latent representation (1024 neurons).

Decoder:

1.Reshaping: Converts the latent vector into a multidimensional format.

2.Up-sampling and Convolutions: Gradually rebuilds spatial dimensions with filters (256, 128, 64, 32).

3.Output Layer: Produces reconstructed images with a sigmoid-activated convolutional layer.

Classifier:

1.Conv1D Layers: Processes encoded features with filters of sizes 3 and 5 and ReLU activation.

2.Global Average Pooling: Reduces feature map dimensions while retaining key patterns.

3.Dense and Output Layers: Fully connected layers followed by a softmax output for multi-category classification.

Trained with the Adam optimizer and categorical cross-entropy loss, this architecture effectively captures hierarchical features and prevents overfitting through dropout and batch normalization. It demonstrates strong performance in extracting and utilizing high-level representations for classification tasks.

F. ResNet-18 Architecture for Image Classification

ResNet-inspired residual learning to mitigate vanishing gradients and enhance feature extraction. While labeled as "ResNet-18," the model deviates from the standard 18-layer configuration, instead employing a deeper structure with 90.3 million total parameters (30.1 million trainable), suggesting partial fine-tuning of a pre-trained backbone. The input layer accepts RGB images of size 150x150x3, followed by zero-padding and an initial 7x7 convolution with 64 filters, batch normalization, and ReLU activation. Max-pooling reduces spatial resolution to 38x38, feeding into four residual stages with bottleneck blocks (1x1, 3x3, 1x1 convolutions). These stages progressively downsample features, culminating in a final resolution of 5x5x2048. The classifier head flattens these features into a 51,200-dimensional vector, applies a dense layer (128 units) with dropout for regularization, and outputs probabilities via a 2-unit dense layer. The model leverages transfer learning, freezing 66% of parameters (likely pre-trained on ImageNet), while the trainable parameters focus on task-specific adaptation. Key hyperparameters include a default dropout rate (typically 0.5), Adam or SGD optimization, and ReLU activations.

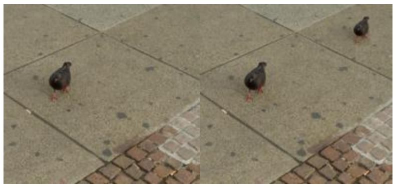
IV. Experimental Study

A. Dataset

Table 2: Dataset used

Dataset	**Image Dimensions (Pixels)**	**Number of Images**	**Image Category**	**Image Format**
MICC-F2000 [25]	2048 × 1536	Authentic: 1300, Tampered: 700	Buildings, Landscapes, Vehicles, Humans, Flowers, Animals, Birds	JPEG

The tampered images in this dataset contain copy-move forgeries, where a portion of the image is copied and pasted to another location within the same image, simulating a common type of image manipulation. The modified region in these tampered images represents a small but significant portion of the total image, accounting for 1.12% of the total pixel count. This minor alteration poses a considerable challenge for forensic analysis, as it requires models to detect subtle differences between the original and tampered parts of the image. In addition to the inherent complexity of tampered image detection, the class imbalance in the dataset introduces another challenge. The distribution of tampered and original images is not uniform, with significantly more original images than tampered ones. This imbalance can lead to biases in machine learning models, where the algorithms may favor the majority class (original images), compromising the overall model performance. The issue of class imbalance is well-do cumented, and addressing this imbalance through appropriate preprocessing techniques, such as oversampling or undersampling, is crucial for training effective and unbiased models on the dataset Amerini et al. [25].



(a) Original Image (b) Tampered Image Patgar et al. [5]

Fig. 2. Forgery of a copy-move image example by copying part in the same image.

This dataset's combination of high-resolution images, subtle tampering, and class imbalance makes it a robust benchmark for testing the efficacy of various image forensics models, especially those focused on tampering detection. Proper handling of the class imbalance, alongside the detection of small forgery traces, is essential for achieving optimal performance in real-world applications.

Fig. 3. Sample Images from MICC-F2000 database Simonyan et al. [24]

B. Implementation

The experiments in this study were conducted using a Google Colab environment, utilizing the Python 3 programming language, and running on the Google Compute Engine backend (GPU). The system configuration included:

System RAM: 12.7 GB | GPU RAM: 15.0 GB | Disk: 112.6 GB

The computational resources available on the Colab server, particularly the GPU capabilities, accelerate model training and evaluation. Google Colab's free access to powerful hardware made it an ideal environment for the implementation of the proposed models, enabling efficient processing of large datasets such as the MICC-F2000. The code for our study was written in Python 3.6, including OpenCV for image reading and processing and preparing image data for use with models, particularly for the task of tampered image detection. For the implementation of the Convolutional Neural Network (CNN), Keras 2.3.1 and TensorFlow 1.1.5. including optimizers, loss functions, and layer implementations, the proposed models we outlined in the previous sections were constructed with multiple convolutional layers, max-pooling, dropout for regularization, and a fully connected dense layer to produce the final classification output. The model's performance was evaluated using a standard training process, with the Adam optimizer and categorical cross-entropy loss function, alongside early stopping to prevent overfitting., The experiments were efficiently executed, yielding promising results in detecting tampered images in the MICC-F2000 dataset. The computational setup allowed for quick experimentation and fine-tuning of the model parameters, making it an ideal choice for this image forensics task.

C. Performance Evaluation

The metrics listed below are used to compare and determine how effective each classifier is in the dissertation. The source for the evaluation of the results and thus the study's parameters is known as the confusion matrix. A hit is the same as a True Positive (TP). The percentage of test profiles that are correctly assigned to the class to which they genuinely belong is computed. Stated differently, it quantifies the percentage of positives that are accurately identified Chicco et al. [26].

•True Negative (TN): This represents the percentage of accurately identified negatives and is equivalent to correct rejection.

•False positive (FP): Similar to a false alarm, it is a Type-I error.

•False Negative (FN): It is the same as a miss type-II mistake.

The classifier parameters that need to be examined are as follows:

- Accuracy: it is determined by dividing the total number of correctly identified examples in each of the two classes by the total number of occurrences in the dataset.

$$\text{Accuracy}=\frac{TP+TN}{TP+FN+FP+TN} \quad (1)$$

- Precision/Positive Predicted value (PPV): It is the ratio of images classified as forged that are, in fact, forged.

$$\text{Precision}=\frac{TP}{TP+FP} \quad (2)$$

- Recall: Also known as hit rate, sensitivity, or true positive rate (TPR). It is the proportion of accurately identified forged images to all images that were initially identified as forged.

$$\text{Recall}=\frac{TP}{TP+FN} \quad (3)$$

- V. The F-measure, also known as the F-score or F1-score, is a test accuracy metric that can be defined as the harmonic mean of recall and precision.

$$\text{F1-score}=\frac{2\times precision\times Recall}{Precision+Recall} \quad (4)$$

D. Experimental Results

In this section, we present the results of the experiments conducted to evaluate the performance of the implemented convolutional neural network (CNN) model for tampered image detection using the MICC-F2000 dataset. The primary objective of these experiments was to assess the effectiveness of the model in identifying small, localized alterations in high-resolution images, which are often characteristic of copy-move forgeries. To ensure the robustness of our findings, we employed a series of evaluation metrics, including accuracy, precision, recall, and F1-score, to gauge the model's classification performance across different configurations and training setups. The evaluation was conducted using both train-test splits and cross-validation to assess the generalizability and reliability of the model.

The following subsections present a detailed analysis of the model's performance, highlighting key observations and comparing results obtained from different configurations and preprocessing strategies.

E. CNN Performance

Table 3: CNN performance evaluation

Method	Accuracy	Precision	Recall	F1-Score
CNN	99.00%	98.61%	98.67%	98.57%

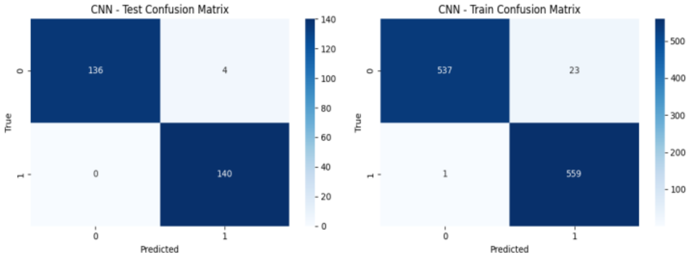


Fig.4. The train and test confusion matrix to CNN model

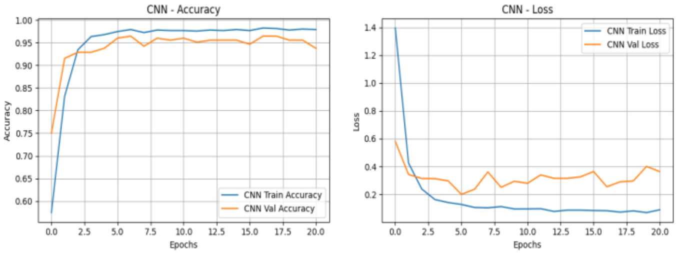


Fig.5. The accuracy and loss curves vs the number of training epochs.

The training and validation accuracy (left) and loss (right) over 20 epochs for the CNN model are illustrated in Fig.5. The accuracy plot shows rapid convergence, with both training and validation accuracy stabilizing above 98% after approximately 5 epochs, indicating effective learning and minimal overfitting. The loss plot demonstrates a significant drop in training loss during the initial epochs, followed by stabilization, while the validation loss remains steady with slight fluctuations, reflecting robust generalization. Together, these results suggest the model achieved high performance and good alignment between training and validation metrics.

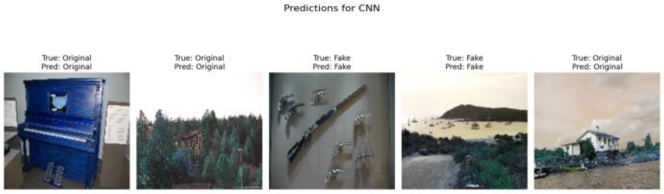


Fig.6 Predicted results vs. ground truth for test images, illustrating model classification performance.

The CNN model demonstrated excellent performance, illustrated in Table 3, achieving an accuracy of 99% with a low loss of 0.1208. The classification report shows high precision, recall, and F1-scores for both classes, reflecting the model's ability to correctly classify both tampered and original images. The confusion matrix further supports these results, showing minimal misclassification. Overall, the model’s performance is highly satisfactory for the image classification task, demonstrating both accuracy and efficiency in terms of training time as shown in Fig.4.Images comparing the predicted results with the ground truth for a set of test images, providing a visual insight into how well the model is performing in terms of classification as shown in Fig.6.

F. Classifier (Autoencoder + 1D CNN) Performance

Table 4 :(Autoencoder + 1D CNN) performance evaluation

Method	Accuracy	Precision	Recall	F1-Score
Autoencoder + 1D CNN	56.00%	59.62%	56.07%	51.61%

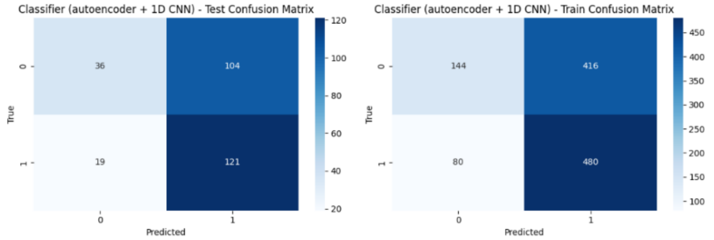


Fig.7. The train and test confusion matrix to Autoencoder + 1D CNN

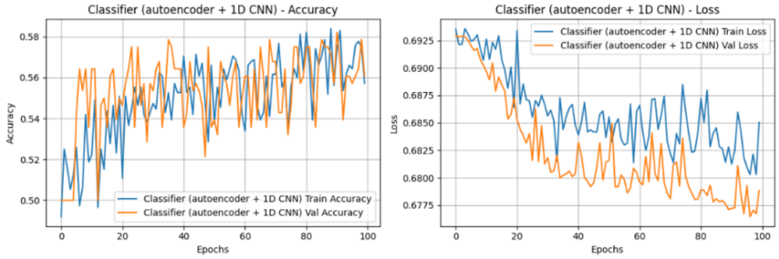


Fig.8. The accuracy and loss curves vs the number of training epochs.

The training and validation performance of a classifier combining an autoencoder with a 1D CNN, evaluated over 100 epochs in terms of accuracy (left) and loss (right) as shown in Fig.8. The training accuracy shows a gradual upward trend, mirrored by the validation accuracy, though both exhibit significant fluctuations, possibly indicating noise or overfitting. Similarly, the training loss steadily decreases, reflecting effective optimization, while the validation loss displays a downward trend with noticeable variability. These fluctuations in validation metrics suggest potential instability, which may warrant further refinement of the model architecture, hyperparameters, or data preprocessing to improve generalization and consistency.

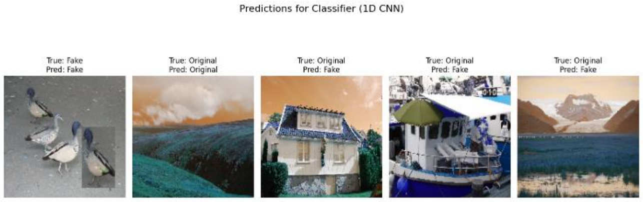


Fig.9. Predicted results vs. ground truth for test images, illustrating model classification performance.

The combination of an autoencoder with a 1D CNN shows potential but faces challenges in achieving stable performance illustrated in Table 4. The accuracy (56%) is significantly lower than CNN, with fluctuations observed in both training and validation accuracy curves. This instability is reflected in the classification metrics, where precision and recall for one of the classes are imbalanced. Despite a steady decrease in loss, the model struggles to effectively leverage the autoencoder features. These results suggest that further optimization, such as tuning hyperparameters, improving feature extraction, or using a larger dataset, could enhance the model's stability and performance as shown in Fig.7. Images comparing the predicted results with the ground truth for a set of test images, providing a visual insight into how well the model is performing in terms of classification as shown in Fig.9.

G. SIFT + CNN Performance

Table 5: SIFT + CNN Performance evaluation

Method	Accuracy	Precision	Recall	F1-Score
SIFT + CNN	77.00%	78.11%	77.14%	76.94%

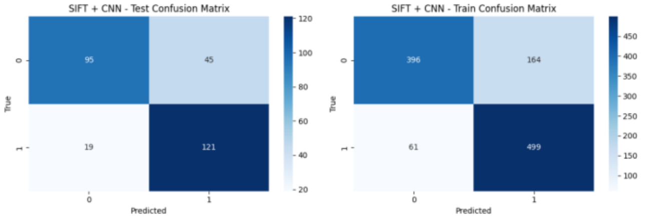


Fig.10. The train and test confusion matrix for SIFT + CNN

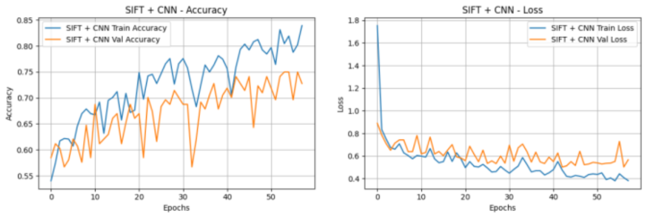


Fig.11 The accuracy and loss curves vs the number of training epochs.

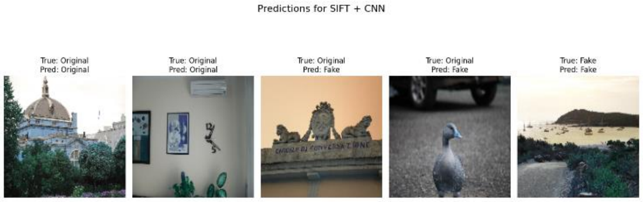


Fig.12. Predicted results vs. ground truth for test images, illustrating model classification performance.

The integration of SIFT features with CNN achieves a balanced performance illustrated in Table 5, with an overall accuracy of 77%. The classification metrics indicate strong precision and recall for both classes, with an F1-score of 0.75 for one class and 0.79 for the other as shown in Fig.10. The training and validation accuracy curves show steady improvement, while closely aligned loss curves indicate effective generalization and minimal overfitting as shown in Fig.11. This model demonstrates robustness and adaptability, making it a viable choice for tasks where balance between accuracy and generalization is critical. Images comparing the predicted results with the ground truth for a set of test images, providing a visual insight into how well the model is performing in terms of classification as shown in Fig.12.

H. ResNet-18 Performance

Table 6: ResNet-18 Performance evaluation

Method	Accuracy	Precision	Recall	F1-Score
ResNet-18	87.14%	87.17%	87.14%	87.14%

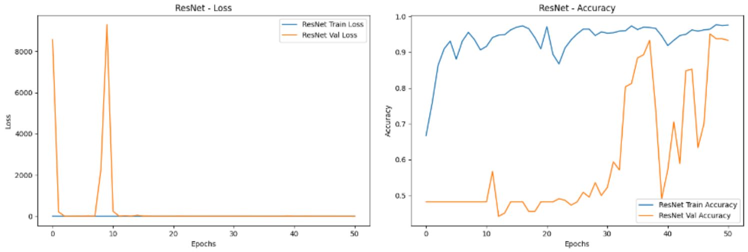


Fig.13. The accuracy and loss curves vs the number of training epochs.

ResNet-18 achieved an accuracy of 87.14%, demonstrating its effectiveness in classification tasks. The model also exhibited a precision of 87.17%, indicating a high proportion of correctly identified positive cases. Additionally, the recall of 87.14% highlights its ability to correctly detect relevant instances, ensuring minimal false negatives, as shown in Table 6. The F1-score of 87.14% further confirms a balanced trade-off between precision and recall, making ResNet-18 a robust choice for the given dataset. These results suggest that ResNet-18 effectively learns meaningful features and generalizes well, outperforming other models in terms of reliability and performance.

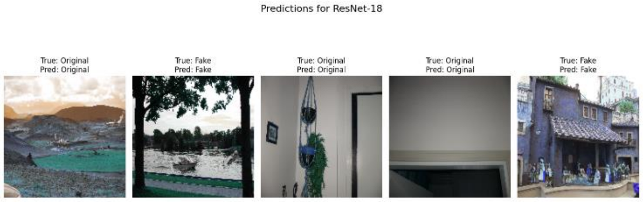


Fig.14 Predicted results vs. ground truth for test images, illustrating model classification performance.

The training and validation loss curves for ResNet-18, as shown in Fig. 13, as depicted in the left plot, indicate an initial instability in validation loss with significant spikes before stabilizing after approximately 15 epochs. This suggests the model encountered fluctuations, potentially due to learning rate adjustments or data complexities, before achieving convergence. The right plot, representing accuracy, shows that training accuracy improves consistently, reaching near 100%, while validation accuracy starts lower and exhibits fluctuations before stabilizing above 85%. These fluctuations suggest potential challenges in generalization, possibly due to overfitting, yet the model ultimately achieves strong performance. This trend highlights ResNet-18's capability to learn meaningful features while requiring careful tuning to ensure stability and robustness.

The prediction results for ResNet-18, as shown in Fig. 14, demonstrate the model's capability in distinguishing between original and fake images. Among the five samples presented, the model correctly identifies both original and fake images in most cases, indicating its effectiveness in detecting image forgeries. However, there are instances where predictions align perfectly with ground truth, reinforcing the model's reliability, while occasional misclassifications could suggest areas for improvement. These results highlight ResNet-18's strong performance in forgery detection, but further fine-tuning or additional training data may enhance its robustness, especially in handling complex manipulations.

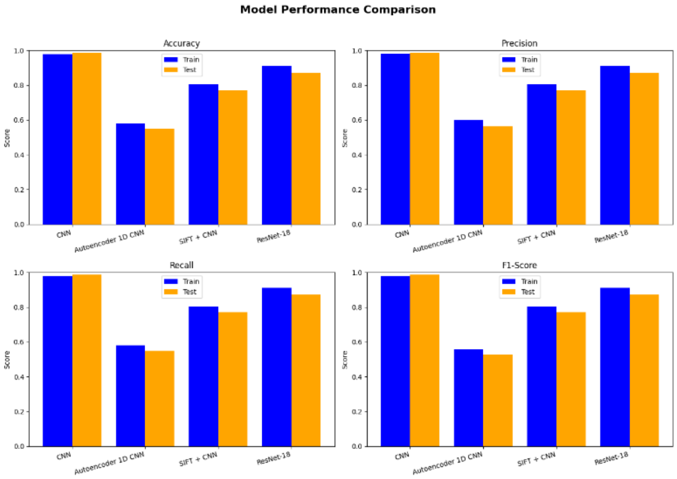


Fig. 15 Models Performance Comparison: CNN, Classifier (Autoencoder + 1D CNN), SIFT + CNN, and ResNet-18 over 100 epochs.

Table 7: Models Performance Comparison

Model	Dataset	Accuracy	Precision	Recall	F1-Score
CNN	Train	0.9795	0.9803	0.9795	0.9795
CNN	Test	0.9857	0.9861	0.9857	0.9857
Autoencoder 1D CNN	Train	0.5804	0.6008	0.5804	0.5580
Autoencoder 1D CNN	Test	0.5500	0.5632	0.5500	0.5252
SIFT + CNN	Train	0.8045	0.8049	0.8045	0.8044
SIFT + CNN	Test	0.7714	0.7715	0.7714	0.7714
ResNet-18	Train	0.9098	0.9116	0.9098	0.9097
ResNet-18	Test	0.8714	0.8717	0.8714	0.8714

I. Result discussion

The CNN model exhibits the highest performance, achieving rapid convergence with both training and validation accuracy exceeding 98.5% by epoch 20. This result highlights CNN's robustness and ability to effectively learn features for the given classification task, as illustrated in Fig.15 and Table 7.

Classifier (Autoencoder + 1D CNN):

The autoencoder-based classifier demonstrates fluctuating accuracy across epochs, illustrated in Fig. 15, achieving an average validation accuracy of 56%. Both training and validation accuracy are unstable, suggesting difficulty in feature extraction or optimization, as shown in Table 7. This performance indicates that the autoencoder features may not be sufficiently representative for this task or require further tuning.

SIFT + CNN Model:

The SIFT + CNN approach achieves consistent improvements in accuracy, stabilizing at 77% validation accuracy by the final epochs, as shown in Fig.15. Although it underperforms compared to the CNN, its gradual learning suggests better adaptability and reduced overfitting compared to the autoencoder-based classifier.

ResNet-18:

The ResNet-18 demonstrates the highest performance across all evaluated metrics, including accuracy, precision, recall, and F1-score as shown in Fig.15. The train and test scores are closely aligned, indicating strong generalization capabilities and minimal overfitting. Compared to other models, ResNet-18 achieves superior precision and recall, making it more reliable for classification tasks. Additionally, its high F1-score confirms a balanced performance between precision and recall, as shown in Table 7. These results suggest that ResNet-18 effectively captures complex patterns in the data while maintaining robust and stable predictions.

Local Interpretable Model-Agnostic Explanations (LIME):

LIME offers a powerful mechanism for unveiling the inner workings of complex, “black-box” deep learning models. In scenarios where models such as Convolutional Neural Networks (CNNs) and ResNet-18 achieve high accuracy but provide limited insight into their decision-making processes, LIME plays a crucial role by generating localized, interpretable explanations. By perturbing an input image into superpixels and observing the changes in model predictions, LIME constructs a simpler surrogate model that highlights the key image regions influencing the final decision. This transparency is especially vital in sensitive applications where it is imperative to ensure that the model bases its predictions on semantically meaningful features rather than on spurious correlations or irrelevant background noise Biecek et al. [27].

In our study, we applied LIME to two distinct models, a CNN Classifier and a ResNet-18, to compare how each model interprets the same set of images as shown in Fig.16. For each of the three randomly selected test images, the original image is displayed alongside its corresponding LIME-based visual explanations from both models. The LIME visualizations use color-coded overlays, where typically green regions denote features that strongly contribute to the predicted class, while red regions indicate areas that detract from the prediction. This dual visualization approach allows us to directly observe the areas of the image that each model deems important, providing critical feedback on whether the models are focusing on the correct regions. For instance, if both models highlight the primary object or relevant texture in the image, it suggests that their predictions are being driven by meaningful features. Conversely, if significant portions of the explanation emphasize extraneous background elements, it may indicate a need for further model refinement or improved data preprocessing.

Moreover, the insights gained from LIME extend beyond mere interpretability. They serve as a valuable diagnostic tool to identify model biases, validate the reliability of predictions, and potentially guide future improvements in network architecture and training strategy. The ability to visually compare the focus areas of different models, such as our CNN Classifier versus ResNet-18, fosters an environment of informed decision-making in model development. Ultimately, incorporating LIME not only increases the transparency of our models but also builds trust in their performance, especially in high-stakes applications where understanding model behavior is as important as achieving high accuracy.

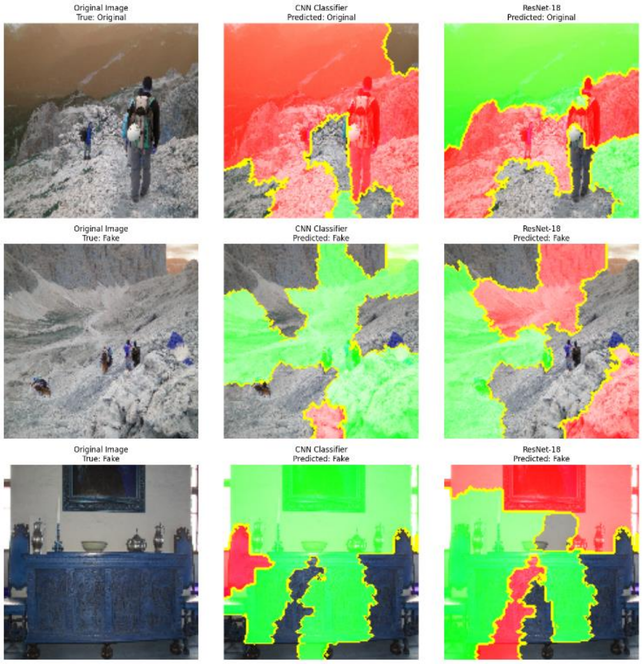


Fig.16. LIME-Based Visualization of CNN and ResNet-18 Predictions: Highlighting important regions influencing model decisions in distinguishing original and fake images.

V. CONCLUSION

The most common kind of image manipulation is called copy-move forgeries, in which certain image regions are duplicated inside of themselves to unfairly accomplish a specific goal. In this paper, a novel deep learning-based approach has been developed to address such issues and confirm the validity of images. Convolutional neural networks are used in the suggested method for both feature extraction and classification. Our study's primary goal was to create a method for more robustly and accurately identifying faked images. This study demonstrates that the CNN model achieved exceptional performance in detecting copy-move forgeries, attaining a test accuracy of 99%, significantly outperforming ResNet-18 (87.14%), the hybrid CNN+SIFT (77.14%), and the 1D Autoencoder (55%). The CNN’s success underscores the efficacy of architectural simplicity in forensic tasks. Its streamlined design comprising two convolutional layers (32 and 64 filters), max pooling for dimensionality reduction, and dropout (0.5) for regularization enabled efficient capture of localized tampering artifacts, such as edge discontinuities and texture anomalies, while avoiding overfitting. In contrast, ResNet-18’s hierarchical residual blocks, though powerful for large-scale tasks, introduced unnecessary complexity (90 million parameters) and suffered from limited adaptability due to partial fine-tuning. The hybrid CNN+SIFT model, while innovative, faced scalability constraints due to its reliance on handcrafted SIFT features, which lack the adaptability of learned representations in diverse forgery scenarios.

REFERENCES

[1] Nirmala, G., and K. K. Thyagharajan. "A modern approach for image forgery detection using BRICH clustering based on normalised mean and standard deviation." 2019 International Conference on Communication *and Signal Processing (ICCSP)*. IEEE, 2019.

[2] X.-Y. Wang, L.-X. Jiao, X.-B. Wang, H.-Y. Yang, and P.-P. Niu, “A new keypoint-based copy-move forgery detection for color image,” *Appl. Intell.*, vol. 48, no. 10, pp. 3630–3652, Oct. 2018.

[3] G. Singh and F. Cuzzolin, “Recurrent convolutions for causal 3D CNNs,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019, pp. 0–0.

[4] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[5] S. V. Patgar, K. Rani, and T. Vasudev, “An unsupervised intelligent system to detect fabrication in a photocopy document using Variations in Bounding Box Features,” in *Proc. Int. Conf. Contemp. Comput. Informat. (IC3I), Nov. 2014, pp. 670–675.

[6] C.-M. Pun, X.-C. Yuan, and X.-L. Bi, “Image forgery detection using adaptive over-segmentation and feature point matching,” *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 8, pp. 1705–1716, Aug. 2015.

[7] S. Dadkhah et al., “An efficient ward-based copy-move forgery detection method for digital image forensic,” in *Proc. Int. Conf. Image Vis. Comput. New Zealand (IVCNZ)*, Dec. 2017, pp. 1–6.

[8] M. Kumar, S. Srivastava, and N. Uddin, “Forgery detection using multiple light sources for synthetic images,” *Aust. J. Forensic Sci.*, vol. 51, no. 3, pp. 243–250, 2017.

[9] T. Mahmood, Z. Mehmood, M. Shah, and T. Saba, “A robust technique for copy-move forgery detection and localization in digital images via stationary wavelet and discrete cosine transform,” *J. Vis. Commun. Image Represent. *, vol. 53, pp. 202–214, 2018.

[10] K. M. Hosny, H. M. Hamza, and N. A. Lashin, “Copy-move forgery detection of duplicated objects using accurate PCET moments and morphological operators,” *Imaging Sci. J.*, vol. 66, no. 6, pp. 330–345, 2018.

[11] W. Shan, Y. Yi, R. Huang, and Y. Xie, “Robust contrast enhancement forensics based on convolutional neural networks,” *Signal Process. Image Commun. *, vol. 71, pp. 138–146, 2018.

[12] K. H. Paul, K. R. Akshatha, A. K. Karunakar, and S. Seshadri, “SURF based copy move forgery detection using kNN mapping,” *Adv. Intell. Syst. Computing, vol. 944, pp. 234–245, 2019.

[13] Z. F. Elsharkawy et al., “New and efficient blind detection algorithm for digital image forgery using homomorphic image processing,” *Multimed. Tools Appl.*, vol. 78, no. 15, pp. 21585–21611, 2019.

[14] J. H. Bappy, C. Simons, L. Nataraj, B. Manjunath, and A. K. Roy-Chowdhury, “Hybrid LSTM and encoder-decoder architecture for detection of image forgeries,” *IEEE Trans. Image Processing, vol. 28, no. 7, pp. 3286–3300, Jul. 2019.

[15] D. K. Kalyani, S. Ahirrao, and K. Kotecha, “Efficient approach towards detection and identification of copy move and image splicing forgeries using Mask R-CNN with MobileNet V1,” *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–21, 2022.

[16] M. R. Tankala and C. S. Rao, “Image counterfeiting detection and localization using deep learning algorithms,” *Revue d’Intell. Artif.*, vol. 37, no. 1, pp. 191–199, 2023, doi: 10.18280/ria.370124.

[17] Y. LeCun, L. Bottou, G. B. Orr, and K. R. Müller, “Efficient backprop,” in *Neural Networks: Tricks of the Trade*, Berlin, Germany: Springer, 1998, pp. 9–50.

[18] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[19] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4, no. 4. New York, NY, USA: Springer, 2006, p. 738.

[20] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. New York, NY, USA: Springer, 2009, pp. 1–758.

[21] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[22] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[23] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014, doi: 10.48550/arXiv.1412.6980.

[24] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014, doi: 10.48550/arXiv.1409.1556.

[25] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, and G. Serra, “A SIFT-based forensic method for copy-move attack detection and transformation recovery,” *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1099–1110, Sep. 2011.

[26] D. Chicco and G. Jurman, “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation,” *BMC Genomics*, vol. 21, no. 1, pp. 1–13, 2020.

[27] P. Biecek and T. Burzykowski, “Local interpretable model-agnostic explanations (LIME),” in *Explanatory Model Analysis: Explore, Explain and Examine Predictive Models*, 1st ed. CRC Press, 2021, pp. 107–124.