Received: September 28, 2023. Revised: November 26, 2023.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Hybrid Feature Selection for Effective Copy-Move Forgery Detection

Manish Shankar Kaushik[1]* **Aditya Bihar Kandali[1]**

[1]*Department of Electrical Engineering, Jorhat Engineering College, Assam, India*
* Corresponding author's Email: mainamanish@gmail.com

Abstract: In today's digital landscape, the threat of digital image forgery is on the rise, necessitating the development of advanced detection and authentication techniques. Copy-move image forgery, a prevalent form of manipulation, involves duplicating a portion of an image and inserting it elsewhere with malicious intent. Detecting such forgeries is of utmost importance. This research introduces an innovative approach that combines convolutional neural network (CNN)-based deep features and gray-level co-occurrence matrix (GLCM) features. These hybrid features are carefully selected using neighborhood component analysis (NCA) to optimize their discriminative power. Subsequently, a support vector machine (SVM) classifier is employed to classify these refined hybrid features, resulting in exceptional detection performance. The proposed method achieves remarkable accuracy in forgery detection. In the CASIA 1 dataset, the hybrid features-NCA-SVM method outperforms other techniques with an accuracy of 97.62%. Similarly, on the MICC-F220 dataset, the hybrid features-NCA-SVM approach attains the highest accuracy of 98.00%. These results underscore the robustness and versatility of the proposed method in detecting copy-move forgeries across different datasets.

Keywords: Convolutional neural network, Cumulative distribution function, Gray-level co-occurrence matrix, Neighborhood component analysis, Support vector machine.

1. Introduction

The widespread availability of image editing software and the ease of digital image manipulation have amplified concerns about image forgery. Among the various forms of image manipulation, copy-move image forgery stands out as a particularly challenging problem in the realm of image authentication. In this deceptive technique, a section of an image is duplicated and placed in another area, compromising the integrity and authenticity of the image. Detecting such manipulations is of paramount importance, with applications spanning from forensic investigations and journalism to ensuring the trustworthiness of digital archives.

The ubiquity of digital cameras has elevated the role of digital media in our daily lives. However, the ready availability of powerful digital image manipulation tools, like 3D Max and Photoshop, has raised significant doubts about the authenticity of digital content, especially when used as evidence in legal cases, insurance claims, or scientific research. Alarming reports reveal a disturbing prevalence of manipulated images within approved manuscripts, underscoring the urgent need for robust forgery detection methods.

In this context, one specific type of forgery, known as copy-move forgery (CMF), has garnered considerable attention in both research and practical applications. CMF involves the duplication of a portion of an image and its insertion elsewhere within the same image. The emerging field of image forensics, particularly in the domain of CMF detection, has become an imperative response to these challenges.

The proposed method presented in this study addresses the complexities of CMF detection by incorporating innovative features and approaches. This introduction will further elaborate on the distinctive features of the proposed method and emphasize its main advantages over existing techniques, positioning it as a formidable solution for image forgery detection. Following are the new

features of the proposed method:

- ***Integration of hybrid features:*** The proposed method introduces a novel approach by combining two different types of features – convolutional neural network (CNN)-based deep features and gray-level co-occurrence matrix (GLCM) features. This fusion of texture-based and deep features enhances the overall discriminative power of the model.
- ***Feature selection with NCA:*** The inclusion of neighborhood component analysis (NCA) is a key feature of the method. NCA optimizes the feature set, reducing dimensionality while preserving classification accuracy. This fine-tuning contributes to more efficient and accurate forgery detection.

2. Literature review

In a recent investigation [1], scholars introduced an upgraded strategy for detecting copy-move forgery (CMF) by employing a salient keypoint selection technique. This innovative approach incorporates the utilization of keypoint features like scale-invariant feature transform (SIFT) and KAZE, effectively bolstering the model's resilience. Meanwhile, the authors of a separate study [2] presented a deep learning (DL) methodology for CMF detection. This novel approach leverages the convolutional block attention module (CBAM) for feature extraction, image segmentation, and the precise localization of manipulated regions. It further harnesses deep matching for self-correlation of feature maps and employs atrous spatial pyramid pooling (ASPP) to amalgamate scaled correlation maps into a coarse mask. To ensure consistency, bilinear upsampling is subsequently deployed to resize the estimated results to align with the input image dimensions. In a distinct investigation [3], the utilization of two DL models, namely, the smaller VGGNet and MobileNetV2, for CMF detection is explored. These models are esteemed for their efficiency and adaptability, particularly for deployment on embedded devices. A modified iteration of MobileNetV2, in particular, demonstrated remarkable efficiency in CM detection. Moreover, in a separate research endeavor [4], an advanced fake image-feature network (AFIFN) grounded in DL techniques is devised. This intricate model incorporates preprocessing techniques such as YCrCb and discrete cosine transformation (DCT)-based image manipulation. The AFIFN model operates within a two-layered network architecture and is engineered to process a pair-wise dataset as its primary input. In another investigative effort [5], simultaneous examination of splicing and CMF recognition is carried out utilizing the CASIA v1.0 and CASIA v2.0 datasets. The investigative process involves the extraction of pertinent features from suspicious images through block discrete cosine transform (BDCT) and an enhanced threshold method, which greatly simplifies the detection of image manipulation. The subsequent phase involves image classification by a support vector machine (SVM), categorizing the images as either CMF or splicing forgeries. Furthermore, the researchers in [5] address the issue of copy-move image forgery detection through a meticulous analysis of discrete cosine transform (DCT) coefficients. The initial step involves the transformation of RGB images into grayscale variants, adhering to conventional image processing techniques. Subsequently, a comprehensive two-dimensional DCT coefficient analysis is executed, generating a feature vector through systematic zig-zag scanning across all image blocks. This feature vector is then subjected to lexicographic sorting, ultimately leading to the identification of duplicated blocks based on euclidean distance. Additionally, in [7-9], a pioneering hybrid approach is introduced, seamlessly merging block-based technology with fourier-mellin transform (FMT) and keypoint-related strategies employing scale-invariant feature transform (SIFT). This innovative approach initially segregates input images into smooth and textured regions, consequently facilitating the extraction of keypoints from the texturized portions through the utilization of SIFT descriptors. Simultaneously, fourier mellin transformation (FMT) is employed on the smoother regions of the images. Subsequently, the extracted features undergo comprehensive matching to unearth duplicated image segments. In yet another in-depth study [10], the researchers confront the intricate challenge of identifying CMF, ultimately proposing an effective and dependable passive-blind detection technique. Furthermore, an intricate keypoint-based image forensics system, relying on superpixel segmentation and the Helmert transformation, is unveiled in [11]. The primary objective of this system is to identify CMF-infected images while simultaneously gathering crucial forensic data. Moreover, [12] introduces an innovative W-Net system-based approach, explicitly engineered for detecting and precisely localizing regions of video forgery through the application of the CMF approach. Remarkably, this particular approach showcases impressive proficiency in the domain of recognizing

manipulated videos. Lastly, [13] pioneers an inventive system tailored for the detection of copy-move image forgery, predominantly founded on a texture feature descriptor termed local tetra pattern (LTrP). This sophisticated system facilitates block-level image comparison to pinpoint localized tampered regions with remarkable precision.

In the literature review, several methods for copy-move forgery detection (CMF) are discussed, each with its own set of strengths and limitations. Here, a clear problem definition is provided, and the drawbacks of these conventional techniques are highlighted:

1. **Salient keypoint selection technique [1]:** This method relies on keypoint features like SIFT and KAZE to enhance resilience, but it may not be robust enough to handle complex manipulations, such as those involving deep learning-based forgery techniques.
2. **Deep learning methodology with CBAM and ASPP [2]:** Deep learning methods often require extensive computational resources and large datasets, making them impractical for resource-constrained environments. Furthermore, the method's complexity may limit its generalizability.
3. **Utilization of smaller DL models (VGGNet, MobileNetV2) [3]:** Smaller DL models may achieve efficiency but might lack the capability to handle more intricate forgery scenarios. They may not be suitable for comprehensive forgery detection across various manipulation types.
4. **Advanced fake image-feature network (AFIFN) [4]:** While AFIFN employs DL techniques, it may have limitations in handling specific preprocessing techniques, and its performance may vary depending on the input data. The model's complexity might also hinder its deployment.
5. **Simultaneous splicing and CMF recognition [5]:** Simultaneous recognition of splicing and CMF may introduce complexity, making it challenging to adapt to specific forgery detection requirements. Furthermore, the method might be less effective for individual CMF detection.
6. **DCT coefficient analysis [5]:** This method focuses primarily on DCT coefficients, which may limit its ability to detect more sophisticated forgeries that do not leave distinct DCT patterns. It might not provide a comprehensive solution for CMF detection.
7. **Hybrid approach with SIFT and FMT [7-9]:** The hybrid approach may introduce increased computational overhead due to the combination of multiple techniques. It might be less efficient in real-time applications.
8. **Passive-blind detection technique [10]:** The specific implementation details and potential limitations of this passive-blind technique are not mentioned in the literature, making it challenging to assess its suitability for various forgery scenarios.
9. **Keypoint-based image forensics system [11]:** The effectiveness of this system for CMF detection and its adaptability to different forgery types are not thoroughly discussed, leaving its limitations unclear.
10. **W-Net system for video forgery detection [12]:** This method is specialized for video forgery, and its applicability to image forgery detection, particularly CMF, is not apparent.
11. **Local tetra pattern (LTrP) system [13]:** The LTrP system focuses on texture feature descriptors and block-level image comparison. It might not cover the entire spectrum of forgery techniques, especially those involving content manipulation.

While the aforementioned research has contributed significantly to copy-move forgery detection, the integrated approach presented here offers several distinct advantages. By amalgamating both texture-based GLCM features and deep features, a more comprehensive and informative feature set is constructed, enriching the discriminative potential. The incorporation of NCA ensures that the feature set is finely tuned for the specific task, effectively reducing dimensionality while preserving classification accuracy. Furthermore, the SVM classifier, well-suited for high-dimensional feature spaces, contributes to the heightened accuracy of forgery detection. In sum, the proposed method presents a robust and efficient solution that effectively overcomes the limitations of existing approaches, thus showcasing superior performance in the intricate task of image forgery detection.

The remainder of this paper is organised as follows: Third section gives an overview of the materials and methods employed in this work. The fourth section outlines the suggested approach for identifying image forgeries. Section five summarizes the results and discussion. Section six concludes with final observations.

3. Materials and methods

The attempt to detect copy-move image forgery

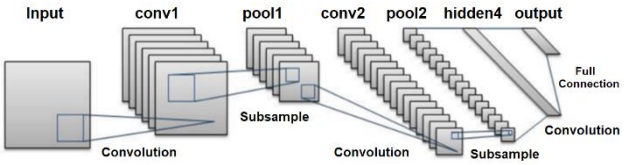

Figure. 1 Architectural representation of a deep convolutional neural network [2]

constitutes a multifaceted challenge, demanding the extraction and scrutiny of distinguishing features from manipulated images. This section elaborates on a comprehensive approach aimed at addressing this intricate issue. It combines the robust attributes of convolutional neural network (CNN)-based deep features with gray-level co-occurrence matrix (GLCM) features, resulting in a multifaceted methodology for detecting image forgeries.

3.1 CNN-based deep features

Deep learning has ushered in a transformative era within computer vision by enabling the automatic derivation of intricate and abstract image representations. Convolutional neural networks (CNNs) stand out as a foundation of this advancement, notably excelling in feature extraction and representation learning. In the approach outlined herein, a pre-trained CNN architecture is harnessed for the extraction of deep features from input images. These deep features are designed to encapsulate both low-level and high-level image characteristics, furnishing the methodology with the capacity to effectively distinguish between authentic and manipulated regions.

The convolution operation: At its core, a convolution is an operation applied to two functions, typically involving real numbers as arguments. Mathematically, the convolution operation can be defined as:

$$s(t) = \int x(a)w(t-a)da \quad (1)$$

In the context of CNNs, this operation is often represented as:

$$s(t) = (x * w)(t) \quad (2)$$

Here, the first function (x) is referred to as the input, and the second function (w) is the kernel. The output of this convolution operation is termed a feature map. When dealing with discrete data in the context of computing, continuous functions are approximated as a sum of "discrete" functions. This takes the form:

$$S(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \quad (3)$$

In the domain of deep learning, the input is typically a multi-dimensional vector (tensor), while the kernel is often a multi-dimensional parameter vector that is adjusted through the learning process. For example, when using an image (I) as input data, a two-dimensional kernel (K) is commonly employed, denoted as:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n) \quad (4)$$

3.2 GLCM features

GLCM operates by capturing the spatial correlations between pixel intensities in an image, furnishing insights into texture and patterns. The integration of GLCM features within the methodology aims to augment the discriminative capabilities of the feature set, particularly when handling images replete with intricate textures.

Mathematical formulation: Given an image x_i with pixel intensities represented as $I(x_i)$, GLCM computes a matrix $P_{d,\theta}(x_i)$ that encodes the joint probability of pixel pairs at a certain distance d and angle θ. These GLCM matrices capture texture information.

3.3 Hybrid feature selection with NCA

To ensure that the feature set encompasses the most pertinent and discriminative information, the methodology employs neighborhood component analysis (NCA), a potent feature selection technique. NCA is designed to optimize the feature space by preserving essential characteristics that distinguish authentic regions from manipulated ones. By amalgamating CNN-based deep features and GLCM features into a unified hybrid feature set, the approach constructs a multifaceted representation of the image content. NCA is subsequently employed to discern and retain the most informative features from this hybrid set, a process that effectively reduces dimensionality while preserving or improving classification accuracy.

Mathematical formulation: Given a dataset D with features X and corresponding labels Y, NCA aims to learn a transformation T of the feature space such that the neighborhood information is preserved:

$$D' = T(X) \quad (5)$$

Where:

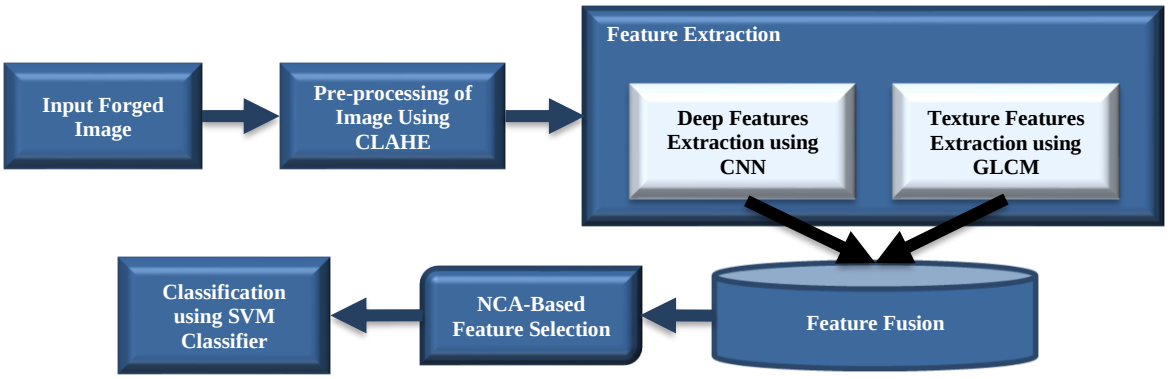

Figure. 2 Block diagram of proposed approach for detecting copy-move forgery

- D' represents the transformed feature space.
- T is the learned transformation.
- The selection of informative features is intrinsically embedded in the transformation learned by NCA.

3.4 SVM classifier

The final phase of the proposed methodology involves classification via a support vector machine (SVM) classifier. SVMs have attained renowned for their effectiveness in binary classification tasks and exhibit particular prowess in handling high-dimensional feature spaces. The deployment of an SVM classifier is driven by the objective of harnessing the discriminative potential of the streamlined hybrid feature set to accurately and reliably detect instances of copy-move image forgeries.

Mathematical formulation: Let X' denote the feature set obtained after NCA-based feature selection. The SVM classification can be defined as follows:

$$\hat{y} = \text{sign}(w \cdot X' + b) \tag{6}$$

Where:

- $\hat{y}$ represents the predicted class label.
- w denotes the weight vector.
- b is the bias term.

The SVM aims to find an optimal hyperplane defined by w and b that best separates the classes in the transformed feature space X'. The classification is based on the sign of the decision function, which is determined by the dot product of w and X'.

4. System modelling

4.1 Image pre-processing

Enhancing image contrast: It is a critical preprocessing step in copy-move forgery detection, as it can improve the visibility of manipulated regions and enhance the overall discriminative power of feature extraction methods. This section discusses the use of contrast limited adaptive histogram equalization (CLAHE) for this purpose, providing both a detailed description and relevant mathematical formulations. CLAHE is a widely used technique for enhancing the contrast of an image while preserving local details. It's particularly beneficial when dealing with images that have non-uniform lighting or regions of varying contrast. In the context of copy-move forgery detection, CLAHE can help in making subtle alterations more discernible and highlight inconsistencies introduced by the forgery. The key idea behind CLAHE is to divide an image into small tiles and apply histogram equalization separately to each tile. This adaptive approach ensures that local contrast is enhanced while avoiding over-amplification of noise in uniform regions. Let's break down the mathematical formulations for CLAHE:

Image division into tiles:

- The input image I is divided into N non-overlapping tiles of size $T_x \times T_y$, where T_x and T_y are the dimensions of each tile.
- This division is performed to create localized histograms for adaptive equalization.

Histogram calculation:

- For each tile T_i, a histogram H_i is computed. The histogram represents the distribution of pixel intensities within that specific tile.
- The histogram H_i is calculated using:

$$H_i(k) = number\ of\ pixels\ with\ intensity\ k\ in\ tile\ T_i \tag{7}$$

Histogram equalization:

- After calculating the histograms, each tile's cumulative distribution function (CDF) C_i is computed.
- The CDF C_i is calculated as:

$$C_i(k) = \begin{cases} C_i(k) & if\ C_i(k) \le ClipLimit \\ ClipLimit & Otherwise \end{cases} \quad (8)$$

Here, $ClipLimit$ is a user-defined parameter that controls the extent of contrast limiting.

Intensity transformation:

- Finally, an intensity transformation is applied to each pixel in the original image, mapping it to the enhanced image.
- The intensity transformation function T is calculated as:

$$T(x,y) = C_{lim}\big(I(x,y)\big) \quad (9)$$

Here, $I(x,y)$ represents the pixel intensity at position (x,y) in the original image.

4.2 CNN-based deep features extraction

Generalized CNN-based deep features extraction involves configuring a convolutional neural network (CNN) architecture to capture relevant patterns and information for this specific task.

4.2.1. CNN architecture selection

Selecting an appropriate CNN architecture is the first step. Popular choices include VGG, ResNet, Inception, and custom-designed architectures. The selection should be based on factors like the complexity of the forgery patterns, the availability of pre-trained models, and computational resources.

4.2.2. Input preprocessing

Preprocessing ensures that input images are suitable for the CNN model. It typically includes resizing images to a consistent resolution, normalization of pixel values, and augmentation (e.g., rotations, flips) to enhance the model's robustness.

4.2.3. Transfer learning

Utilizing pre-trained convolutional neural network (CNN) models trained on extensive image datasets, such as ImageNet, has the potential to expedite the training process while also potentially augmenting overall performance. Fine-tuning the pre-trained models by adjusting the top layers to match the problem's output requirements is a common practice.

4.2.4. Feature extraction

Feature extraction using a CNN involves feeding the input images through the network and extracting features from one or more intermediate layers. These features capture hierarchical information, from low-level textures to high-level semantic content.

4.2.5. Mathematical formulation

Let's define the key components mathematically:

CNN model: A CNN model can be represented as a function f_{CNN} that takes an input image I and produces feature maps at a specific layer L:

$$F_L = f_{CNN}(I) \quad (10)$$

Here, F_L is the set of feature maps at layer L and I is the input image.

Feature extraction: The feature extraction process involves selecting one or more feature maps from layer L to represent the input image. These feature maps are flattened into a vector, which forms the extracted deep features:

$$F_{extracted} = Flatten(F_L) \quad (11)$$

Here, $F_{extracted}$ is the vector of extracted deep features.

4.3 Texture features extraction using GLCM

GLCM features are computed by analyzing the co-occurrence of pairs of pixel intensities within a specified neighborhood in an image. The basic idea is to measure how often different pairs of gray levels appear together at a certain distance and direction within an image. These features capture textural information related to patterns, structures, and variations in an image. Let's delve into the mathematical formulation of GLCM features extraction:

4.3.1. Define the gray-level co-occurrence matrix (GLCM)

The GLCM, denoted as $P(i,j|d,\theta)$, represents the joint probability of two pixels with intensity values i and j occurring at a certain relative displacement d and angle θ in the image. It is computed by sliding a window of a specified size over the image and counting the occurrences of pixel pairs that meet the displacement and angle criteria.

$$P(i,j|d,\theta) = \sum_x \sum_y \delta(I(x,y) - i)\delta(I(x + d\cos(\theta), y + d\sin(\theta) - j) \quad (12)$$

Where:

- $P(i,j|d,\theta)$ is the GLCM at displacement d and angle θθ for pixel values i and j.
- δ is the Kronecker delta function, which is 1

if the condition inside is true and 0 otherwise.
- $I(x, y)$ is the pixel value at position (x, y) in the image.
- d is the displacement (the distance between two pixels).
- θ is the angle at which we are analyzing the co-occurrence (typically 0°, 45°, 90°, and 135°).

4.3.2. Normalize the GLCM

Normalization ensures that GLCM values lie in the range [0, 1] and makes the features invariant to changes in image contrast and brightness. Normalization is typically done by dividing the GLCM by the sum of all its elements:

$$P_{normalized}(i, j|d, \theta) = \frac{P(i, j|d, \theta)}{\sum_i \sum_j P(i, j|d, \theta)} \quad (13)$$

4.3.3. Compute GLCM features

Once the GLCM is normalized, it can compute various texture features from it. Common GLCM features include:

- ***Contrast:*** Measures the local variations in pixel intensity values.

$$Contrast = \sum_i \sum_j (i - j)^2 P_{normalized}(i, j|d, \theta) \quad (14)$$

- ***Energy (angular second moment):*** Reflects the uniformity or homogeneity of texture.

$$Energy = \sum_i \sum_j [P_{normalized}(i, j|d, \theta)]^2 \quad (15)$$

- ***Entropy:*** Captures the randomness or complexity of the texture.

$$Entropy = -\sum_i \sum_j P_{normalized}(i, j|d, \theta) \log[P_{normalized}(i, j|d, \theta)] \quad (16)$$

- ***Correlation:*** Describes the linear dependency between pixel pairs.

$$Correlation = \frac{\sum_i \sum_j (i-\mu)(j-\nu) P_{normalized}(i, j|d, \theta)}{\sigma_i \sigma_j} \quad (17)$$

Where:

- μ and ν are the means of the marginal distributions of i and j, respectively.
- σ_i and σ_j are the standard deviations of the marginal distributions of i and j, respectively.

4.4 Combining deep features and GLCM features

Combining deep features extracted from a CNN with GLCM features is a powerful approach for image analysis tasks, including copy-move image forgery detection. The goal is to leverage both texture information captured by GLCM and semantic information captured by deep features to improve the overall performance of the detection system.

4.4.1. Feature combination process

The process of combining deep features and GLCM features involves merging these two sets of features into a single, unified feature vector. The following steps outline the process:

Deep feature extraction:
- Deep features are derived from the input images through the utilization of either a pre-trained CNN model or a custom-made architecture precisely created for the particular undertaking.
- The CNN extracts features that represent high-level semantic information in the images.

GLCM feature extraction:
- GLCM features are computed by analyzing the spatial relationships between pixel values, capturing textural information in the images.
- Multiple GLCMs may be computed for different orientations and displacements.

Feature fusion:
- The extracted deep features and GLCM features are concatenated or combined in some way to create a hybrid feature vector.
- Concatenation is a common approach, resulting in a feature vector that contains deep features followed by GLCM features.

4.4.2. Mathematical formulation of feature combination

Let D represent the deep features extracted from the CNN, and GG represent the GLCM features. The combined feature vector F can be expressed as:

$$F = [D, G] \quad (18)$$

Where, $[D, G]$ denotes the concatenation of deep features D and GLCM features G into a single vector.

4.5 NCA-based feature selection

Neighborhood component analysis (NCA) is a

dimensionality reduction technique that aims to select a subset of features to improve the performance of a classification or clustering task. In the context of combined deep and GLCM features, NCA can be used to select the most informative features from the hybrid feature vector.

4.5.1. Objective function

NCA aims to maximize a stochastic objective function that measures the quality of feature selection based on the task's goals. The objective function is defined as:

$$J(f) = \sum_{i=1}^{N} p_i \sum_{j \neq i} p_j 1(y_i = y_j) \, exp\left(-\left\| f(x_i) - f(x_j) \right\|^2\right) \quad (19)$$

Where:

- f represents the feature selection function.
- x_i and x_j are data samples.
- y_i and y_j are the corresponding labels.
- $1(y_i = y_i)$ is an indicator function that equals 1 if $y_i = y_j$ (samples belong to the same class) and 0 otherwise.
- p_i is the probability of selecting sample x_i for optimization.

4.5.2. Optimization

NCA uses optimization techniques, such as stochastic gradient descent (SGD), to find the optimal feature selection function f that maximizes the objective function $J(f)$. The optimization process iteratively updates the feature selection based on the task's goals.

4.5.3. Feature selection result

The final feature selection result is a subset of the combined feature vector F, consisting of the most informative features according to the NCA optimization. These selected features are retained for further processing, such as classification for copy-move image forgery detection.

4.6 Classification using support vector machine

SVM is a robust classification algorithm used to identify copy-move forgeries in digital images. In this context, SVM plays a crucial role in distinguishing between authentic images and those containing copy-move forgeries, utilizing features selected through NCA.

4.6.1. SVM formulation

The core formulation for the binary classification problem addressed by SVM can be described as follows:

Given a training dataset:

$$Training\ Dataset = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \quad (20)$$

Where:

- x_i represents a feature vector (NCA-selected features) of dimension d for the i^{th} sample.
- y_i signifies the corresponding class label, where $y_i \in \{-1, +1\}$ (typically, -1 for authentic and $+1$ for forgery).

SVM aims to determine a hyperplane characterized by a weight vector w and a bias term b that effectively separates the data points. The decision function can be defined as:

$$f(x) = sign(w \cdot x + b) \quad (21)$$

Here:

- $f(x)$ serves as the decision function, predicting the class label for a given input feature vector xx.
- w represents the weight vector.
- b corresponds to the bias term.
- $(\cdot)$ denotes the dot product.

4.6.2. Soft margin SVM

In real-world scenarios, perfect linear separability of data is not always feasible. SVM accommodates this by introducing a soft margin, allowing for some classification errors. Slack variables ξ_i are introduced for each training sample, and the optimization objective becomes:

$$Optimization\ Objective = \min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \quad (22)$$

Subject to:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad i = 1,2,\ldots,N$$
$$\xi_i \geq 0, \quad i = 1,2,\ldots,N \quad (23)$$

Where:

- The symbol C denotes a regularization parameter, a pivotal component in machine learning, responsible for regulating the delicate balance between two critical objectives: maximizing the margin while

simultaneously minimizing classification errors.

- ξ_i stands for the slack variable associated with the i^{th} sample, allowing for some margin of misclassification.

Pseudo code for copy-move forgery detection with SVM and NCA-selected features

Step 1: *Feature extraction and selection*

Extract features from images using CNN and GLCM.

Select the most informative features using NCA-based feature selection.

Step 2: *Data preparation*

Split the dataset into training and testing sets.

Encode class labels (e.g., -1 for authentic, +1 for forgery).

Step 3: *SVM training*

Train an SVM classifier on the training data with NCA-selected features.

Choose appropriate SVM parameters (e.g., C and kernel type).

Step 4: *SVM testing*

Use the trained SVM model to predict class labels for the testing data.

Step 5: *Performance evaluation*

Evaluate the classification performance using metrics like accuracy, precision, recall, and F1-score.

Step 6: *Interpretation and reporting*

Examine the results to identify copy-move forgeries in the tested images.

Report the locations and characteristics of detected forgeries.

5. Results and discussion

5.1 Database

CASIA 1: The CASIA 1 [14] dataset includes a total of 1,000 digital images that are artificially tampered to create different types of image forgeries. These images are divided into 500 original images and their corresponding 500 forged images. The resolution of the images in the CASIA 1 dataset is relatively small, with an image size of 512×512 pixels. The images are in grayscale format, with a single channel representing the intensity values of the images.

MICC-F220: The MICC-F220 dataset is a benchmark dataset for image forgery detection, created by the media integration and communication center (MICC) at the University of Florence, Italy [15]. It contains a collection of 220 high-resolution digital images that are artificially tampered to create different types of image forgeries. MICC-F220 dataset consists of a total of 220 digital images. These images are divided into 110 original images and their corresponding 110 forged images, with each original image having one corresponding forged image. This provides a moderate-sized dataset for evaluating forgery detection methods.

(a) (b)

Figure. 3 Sample images from CASIA 1 dataset [14]: (a) Original image and (b) Forgery image

(a) (b)

Figure. 4 Sample images from MICC-F220 dataset [15]: (a) Original image and (b) Forgery image

Table 1. Evaluation parameters

TP (True Positive)	"Indicates instances of authentic regions correctly classified as authentic"
TN (True Negative)	"Indicates instances of manipulated regions correctly classified as manipulated"
FP (False Positive)	"Indicates instances of authentic regions incorrectly classified as manipulated"
FN (False Negative)	"Indicates instances of manipulated regions incorrectly classified as authentic"

5.2 Evaluation parameters

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (24)$$

$$Precision = \frac{TP}{TP+FP} \quad (25)$$

Table 2. Comparative analysis of results of CASIA 1 dataset

Parameters	GLCM-SVM	CNN-SVM	Hybrid Features-NCA-SVM
Accuracy	0.8182	0.9606	0.9762
Error	0.1818	0.0394	0.0238
Sensitivity	0.8182	0.9798	0.9748
Specificity	0.8182	0.9423	1
Precession	0.8182	0.9417	1
False Positive Rate	0.1818	0.0577	0
F1 Score	0.8182	0.9604	0.9872
Matthews Correlation Coefficient	0.6364	0.9219	0.8260
Kappa	0.6364	0.9212	0.8112

$$Sensitivity = \frac{TP}{TP+FN} \quad (26)$$

$$Specificity = \frac{TN}{TN+FN} \quad (27)$$

$$Error\ Rate = \frac{FP+FN}{TP+TN+FP+FN} \quad (28)$$

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP+TN} \quad (29)$$

$$F - Score = \frac{2TP}{2TP+FP+FN} \quad (30)$$

$$Matthews\ Correlation\ Coefficient\ (MCC) = \frac{(TP\times TN)-(FP\times FN)}{\sqrt{(TP+FN)(TP+FP)(TN+FN)(TN+FP)}} \quad (31)$$

$$Kappa\ Statistics = \frac{2(TP\times TN-FN\times FP)}{(TP+FP)\times(FP+TN)+(TN+FN)\times(FN+TN)} \quad (32)$$

5.3 Results

Table 2 provides a comparative analysis of the results obtained from three different methods applied to the CASIA 1 dataset for copy-move forgery detection. The first method utilizes GLCM-based texture features with SVM classification, resulting in an accuracy of 81.82%. The second method employs CNN-based deep features combined with SVM, achieving a significantly higher accuracy of 96.06%. The third method combines hybrid features selected through NCA with SVM, yielding the highest accuracy of 97.62%. These metrics, including error rate, sensitivity, specificity, precision, false positive rate, F1 score, Matthews correlation coefficient, and Kappa statistics, collectively evaluate the performance of each approach. Notably, the hybrid feature selection method stands out as it demonstrates superior performance across most metrics, indicating its effectiveness in detecting copy-move forgeries within the CASIA 1 dataset.

Table 3. Comparative analysis of results of MICC-F220 dataset

Parameters	GLCM-SVM	CNN-SVM	Hybrid Features-NCA-SVM
Accuracy	0.9458	0.9557	0.9800
Error	0.0542	0.0443	0.0200
Sensitivity	0.9600	0.9700	0.9818
Specificity	0.9320	0.9417	0.9951
Precession	0.9320	0.9417	0.9800
False Positive Rate	0.0680	0.0583	0.0049
F1 Score	0.9458	0.9557	0.9799
Matthews Correlation Coefficient	0.8920	0.9117	0.9757
Kappa	0.8916	0.9113	0.9375

Table 4. Comparative analysis of proposed image forgery detection with previous research works

Authors	Dataset	Method	Accuracy %
[5]	CASIA 1	LBP	87.5%
[16]	CASIA 1	DCT and local binary pattern	96.3%
[17]	CASIA 1	DCT	92%
[18]	CASIA 1	SURF and template matching	97.5%
[19]	CASIA 1	CMFD-SIFT	93.1%
[20]	CoMoFoD dataset	Binary Discriminant Features	88.35%
[20]	IMD dataset	Binary Discriminant Features	89.31%
[21]	CMH dataset	LBPROT, SIFT	89.94%
Proposed Method	CASIA 1	CNN-SVM	96.06%
Proposed method	CASIA 1	Hybrid features-NCA-SVM	97.62%
Proposed Method	MICC-F220	CNN-SVM	95.57%
Proposed Method	MICC-F220	Hybrid Features-NCA-SVM	98%

Table 3 presents a comparative analysis of results obtained from three different methods applied to the MICC-F220 dataset for copy-move forgery detection. The first method utilizes GLCM-based texture features with SVM classification, achieving an

accuracy of 94.58%. The second method employs CNN-based deep features combined with SVM, resulting in an accuracy of 95.57%. The third method combines hybrid features selected through NCA with SVM, demonstrating the highest accuracy of 98.00%. These metrics, including error rate, sensitivity, specificity, precision, false positive rate, F1 score, Matthews correlation coefficient, and Kappa statistics, provide a comprehensive assessment of the performance of each approach. Notably, the hybrid feature selection method outperforms the others across most metrics, indicating its effectiveness in detecting copy-move forgeries within the MICC-F220 dataset. The high accuracy, low error rate, and strong agreement with the ground truth (as reflected in Kappa) highlight the robustness of this approach in the specific dataset context.

Table 4 offers a comprehensive comparative analysis of the proposed image forgery detection method alongside prior research works across various datasets. Previous research references [5, 16-19] explored various techniques primarily on the CASIA 1 dataset, achieving accuracy percentages ranging from 87.5% to 97.5%. In contrast, the proposed method, when applied to the CASIA 1 dataset, exhibits remarkable accuracy, with CNN-SVM achieving 96.06%, and the hybrid features-NCA-SVM approach outshining all previous methods with an accuracy of 97.62%. The proposed method's proficiency extends to the MICC-F220 dataset, where CNN-SVM achieves an accuracy of 95.57%, and the hybrid features-NCA-SVM method attains an impressive accuracy of 98%. These results underscore the superior performance of the proposed hybrid feature selection method compared to prior techniques, confirming its efficacy in image forgery detection across diverse datasets, while emphasizing that the references [20, 21] provide insights into alternative approaches.

6. Conclusion

This research paper introduces a robust and accurate approach for copy-move image forgery detection, addressing the growing threat of digital image manipulation in today's digital age. By combining CNN-based deep features and GLCM features, this novel approach achieves superior performance in detecting copy-move forgeries. The hybrid features selected using NCA optimize the discriminative power of the feature set. Subsequently, the SVM classifier effectively classifies the reduced hybrid features, ensuring accurate forgery detection. The results and discussion showcase the effectiveness of the proposed approach on two different datasets, CASIA 1 and MICC-F220. In the CASIA1 dataset, the hybrid features-NCA-SVM method outperforms other techniques, achieving an accuracy of 97.62%. Similarly, on the MICC-F220 dataset, the hybrid features-NCA-SVM approach attains the highest accuracy of 98.00%. These results indicate the robustness and versatility of the proposed method in detecting copy-move forgeries across different datasets. In terms of future scope, further research can explore the application of this approach to larger and more diverse datasets to assess its scalability and generalizability. Additionally, efforts can be made to enhance the efficiency of the detection process, making it more suitable for real-time applications. Furthermore, investigating the extension of this approach to detect other forms of digital image forgeries, such as splicing and retouching, could contribute to a more comprehensive image forensics toolkit. Overall, this research paves the way for improved digital image authenticity verification and forensics in an increasingly digital world.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Manish Shankar Kaushik is the principal author responsible for the study's conception and design, overseeing experimental procedures, conducting data analysis, and composing the manuscript. He adeptly executed data acquisition and analysis, generated graphical representations, and made substantial contributions to manuscript development. He was actively engaged in study design, offering invaluable insights during data interpretation, and precisely revising the manuscript. Dr. Aditya Bihar Kandali served as the project supervisor, providing critical assessment of the manuscript.

Acknowledgments

The authors would like to express their gratitude to Jorhat engineering college, Assam for all of their assistance and encouragement in carrying out this research and publishing this paper.

References

[1] N. Kumar and T. Meenpal, “Salient keypoint-based copy–move image forgery detection. Australian Journal of Forensic Sciences”, *Australian Journal of Forensic Sciences*, Vol. 55, No. 3, pp. 331-354, 2023.

[2] M. Sabeena and L. Abraham, “Convolutional

block attention-based network for copy-move image forgery detection", *Multimedia Tools and Applications*, pp. 1-23, 2023.

[3] M. N. Abbas, M. S. Ansari, M. N. Asghar, N. Kanwal, T. O. Neill, and B. Lee, "Lightweight deep learning model for detection of copy-move image forgery with post-processed attacks", In: *Proc. of 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pp. 000125-000130, 2021.

[4] M. Ananthi, P. Rajkumar, R. Sabitha, and S. Karthik, "A secure model on Advanced Fake Image-Feature Network (AFIFN) based on deep learning for image forgery detection", *Pattern Recognition Letters*, Vol. 152, pp. 260-266, 2021.

[5] C. S. Prakash, A. Kumar, S. Maheshkar, and V. Maheshkar, "An integrated method of copy-move and splicing for image forgery detection", *Multimedia Tools and Applications*, Vol. 77, pp. 26939-26963, 2018.

[6] M. H. Alkawaz, G. Sulong, T. Saba, and A. Rehman, "Detection of copy-move image forgery based on discrete cosine transform", *Neural Computing and Applications*, Vol. 30, pp. 183-192, 2018.

[7] M. A. Alohali, F. N. A. Wesabi, A. M. Hilal, S. Goel, D. Gupta, and A. Khanna, "Artificial intelligence enabled intrusion detection systems for cognitive cyber-physical systems in industry 4.0 environment", *Cognitive Neurodynamics*, Vol. 16, No. 5, pp. 1045-1057, 2022.

[8] K. B. Meena and V. Tyagi, "A hybrid copy-move image forgery detection technique based on Fourier-Mellin and scale invariant feature transforms", *Multimedia Tools and Applications*, Vol. 79, No. 11-12, pp. 8197-8212, 2020.

[9] A. M. Hilal, M. A. Alohali, F. N. A. Wesabi, N. Nemri, H. J. Alyamani, and D. Gupta, "Enhancing quality of experience in mobile edge computing using deep learning-based data offloading and cyberattack detection technique", *Cluster Computing*, pp. 1-12, 2021.

[10] Q. C. Yang and C. L. Huang, "Copy-move forgery detection in digital image", In: *Proc. of Advances in Multimedia Information Processing-PCM 2009: 10th Pacific Rim Conference on Multimedia*, Bangkok, Thailand, pp. 816-825, 2009.

[11] H. Y. Huang and A. J. Ciou, "Copy-move forgery detection for image forensics using the superpixel segmentation and the Helmert transformation", *EURASIP Journal on Image and Video Processing*, Vol. 2019, No. 1, pp. 1-16, 2019.

[12] B. Tokas, V. R. Jakkinapalli, and N. Singla, "Video Forgery Detection and Localization with Deep Learning Using W-NET Architecture", In: *Proc. of Computational Intelligence: Select Proceedings of InCITe*, Springer Nature Singapore, pp. 31-38, 2023.

[13] S. Ganguly, S. Mandal, S. Malakar, and R. Sarkar, "Copy-move forgery detection using local tetra pattern-based texture descriptor", *Multimedia Tools and Applications*, pp. 1-22, 2023.

[14] J. Dong, W. Wang, and T. Tan, "Casia image tampering detection evaluation database", In: *Proc. of China Summit and International Conference on Signal and Information Processing*, pp. 422-426, 2013.

[15] 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 Transactions on Information Forensics and Security*, Vol. 6, No. 3, pp. 1099-1110, 2011.

[16] A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, G. Bebis, and H. Mathkour, "Passive detection of image forgery using DCT and local binary pattern", *Signal, Image and Video Processing*, Vol. 11, pp. 81-88, 2017.

[17] S. Dua, J. Singh, and H. Parthasarathy, "Image forgery detection based on statistical features of block DCT coefficients", *Procedia Computer Science*, Vol. 171, pp. 369-378, 2020.

[18] B. Yang, X. Sun, H. Guo, Z. Xia, and X. Chen, "A copy-move forgery detection method based on CMFD-SIFT", *Multimedia Tools and Applications*, Vol. 77, pp. 837-855, 2018.

[19] A. Rani, A. Jain, and M. Kumar, "Identification of copy-move and splicing based forgeries using advanced SURF and revised template matching", *Multimedia Tools and Applications*, Vol. 80, pp. 23877-23898, 2021.

[20] P. M. Raju and M. S. Nair, "Copy-move forgery detection using binary discriminant features", *Journal of King Saud University-Computer and Information Sciences*, Vol. 34, No. 2, pp. 165-178, 2022.

[21] G. Tahaoglu, G. Ulutas, B. Ustubioglu, M. Ulutas, and V. V. Nabiyev, "Ciratefi Based Copy Move Forgery Detection on Digital Images", *Multimedia Tools and Applications*, Vol. 81, No. 16, pp. 22867-22902, 2022.