Evolutionary Intelligence (2025) 18:127
https://doi.org/10.1007/s12065-025-01117-8

RESEARCH PAPER

An optimized CNN model for copy-move forgery detection and localization in digital images using particle swarm and Grey wolf optimization algorithms

Prabhu Bevinamarad[1] · **Prakash H. Unki**[1]

Received: 28 May 2024 / Revised: 8 January 2025 / Accepted: 12 November 2025

Abstract
Image data is increasing daily at a rapid speed, and at the same time, image forgery has become widespread with readily available and accessible software tools. The image manipulations can be performed in different ways. But, copy-move image forgery is one of the prominent types of malicious forgery operation aimed at hiding sensitive information within the same image and making it challenging for human eyes to detect the forgery and authenticate its contents. The literature reported many deep learning techniques to detect the copy-move forgery images and enhance the detection accuracy. However, these techniques struggle with hyperparameter optimization issues, and fine-tuning hyperparameters and optimizing networks pose significant challenges. Hence, this study proposes an approach to implement optimized Convolutional Neural Network (CNN) model by integrating metaheuristic optimization strategies such as Particle Swarm Optimization (PSO) and Gray Wolf Optimization (GWO) algorithms to identify optimal CNN parameters for automated fine-tuning of CNN configurations. Further, the optimized CNN model is used to extract the significant image features and construct efficient feature maps to classify copy-move forgery images and accurately localize the forgery regions present within the images. To evaluate the performance of the proposed model, we used three publicly available standard datasets: CoMoFoD, CMFD, and MICC-F600. The CoMoFoD dataset is used to validate the performance against plain copy-move forgery images and forgery images with global postprocessing attacks such as image blurring, contrast reduction, noise addition, and JPEG compression. The CMFD and MICC-F600 datasets are used to test plain forgery images and forgery images with local postprocessing attacks such as rotation and scaling. The evaluation results show that the suggested approach effectively detects and localizes forgery, performing well against plain and post-processed copy-move forgery images.

Keywords Convolution neural network · Copy-move forgery · Digital forensics · Gray wolf optimization · Hyperparameter tuning · Particle swarm optimization

1 Introduction

Images have become the most prominent entities used in various digital systems, such as hospitals, banks, courtrooms, etc., for document verification and information dissemination to speed up the process and provide hassle-free service to humankind. The number of images used in these places is increasing daily as legal evidence. On the other hand, the availability and accessibility of advanced image editing tools have led to a surge in crimes involving digital image manipulations and sharing them on social media sites like Facebook, Instagram, Telegram, and WhatsApp without knowing its implications on society. It also requires an individual to put more effort into understanding

Prakash H. Unki contributed equally to this work.

✉ Prabhu Bevinamarad
prabhubev@gmail.com

Prakash H. Unki
prakashunki@gmail.com

[1] Department of Computer Science and Engineering, BLDEA's V.P. Dr. P.G. Halakatti College of Engineering and Technology (Affiliated to Visvesvaraya Technological University, Belagavi-590018, Karnataka), Vijayapura, Karnataka 586103, India

Published online: 26 November 2025

and discerning between real and fake photos. In addition, it also raises concerns about the authenticity of image content. Image alteration detection can be broadly divided into active and passive methods. The active method incorporates a digital signature or watermark when generating the image. Later, the image is examined using these embeddings to see if it has been altered. The passive method cannot depend on pre-embedded data, such as a watermark implanted to identify image counterfeiting. Since no further information is available to detect image forgeries, this process is called the blind image forgery detection method. The blind image forgery is further classified into dependent and independent.

The independent method identifies resampling and compression manipulations, while the dependent method uncovers splicing and copy/move alterations. The hierarchical relationship of these methods is illustrated in Fig. 1. Copy-move forgery is a significant tampering technique to enhance visual content or conceal crucial information within images. This technique involves placing a duplicated image segment over a specific area with or without additional image processing steps. When duplication occurs without postprocessing, the patches look identical; However, in postprocessing operations, they diverge. Consequently, the replaced patch may mimic the original image's texture and pattern, making it nearly indistinguishable from the naked eye. Therefore, forensic tools are essential to establish an image's integrity and authenticity.

The first copy-move forgery detection technique in a digital image was introduced by Fridrich et al. [1] in 2003 by utilizing statistical features. In this method, the image is initially segmented into patches and significant image features are extracted and paired using the matching technique. This approach functions effectively when no alterations are made to the duplicated area. However, it becomes ineffective when subjected to geometric transformations like rotation and scaling applied to the copied region after pasting it elsewhere. Over the past two decades, several methods based on block matching and key-point-based techniques have been reported in the literature. However, these methods face challenges when dealing with small and duplicated regions. Also, these existing forgery detection methods often rely on manually crafted features sensitive to post-processing operations and struggle to detect copy-move forgery regions and forgery with post-processing accurately. In recent years, several Copy-move Forgery Detection (CMFD) methods leveraging deep learning techniques have been introduced to address the various issues related to copy-move forgery detection. The deep learning networks draw inspiration from biological neurons in human networks, featuring multiple nonlinear layers to process objects in parallel. Nevertheless, many of these methods are limited to classification, require more effort in hyperparameter tuning and exhibit inadequate performance when confronted with different local and global post-processing operations. Given these challenges, we present a forgery detection system utilizing the CNN capability by integrating a metaheuristic optimization algorithm to optimize various hyperparameters automatically and extract significant image features to detect copy-move forgery images and localize the forgery region present in the image. This system overcomes the limitations of both traditional methods and current deep learning techniques, offering an efficient solution for detecting copy-move forgery in images while effectively handling the complexities of real-world scenarios.

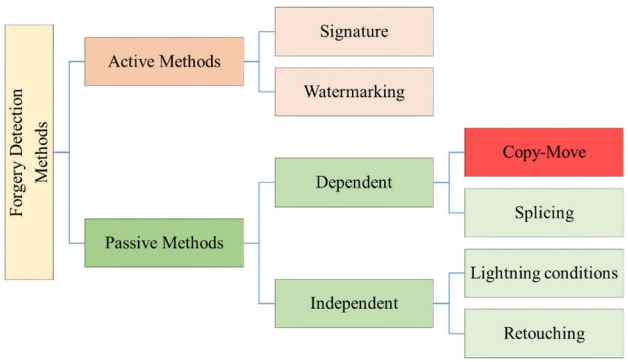

Fig. 1 Classification of image forgery detection techniques

The reminder section of this paper is structured as follows: Section 2 presents a literature review on copy-move forgery detection. Section 3 delves into the details of the implementation of the proposed model. Section 4 covers the experimental setup, evaluation, results, and discussion. Finally, Sect. 5 concludes by summarizing the outcomes of the proposed work.

2 Review of related literature work

Since 2003, many copy-move forgery detection methods have been implemented to detect copy-move forgeries in digital images utilizing different approaches. These approaches are classified into Block-based, Keypoint-based, and Deep Learning based forgery detection techniques. Block-based techniques are the primary categories of forgery detection techniques initially introduced. Critical steps include block tiling, feature extraction, feature filtering, and feature matching. These techniques partition the image into overlapping blocks of fixed size (b $\times$ b). Key image characteristics are extracted from each block, forming a feature matrix with feature vectors for the overlapping blocks. Then, lexicographically sorting the feature vector enables feature filtering and matching. Finally, identical patches representing duplicated areas are identified. Many block-based techniques have been developed to determine

the copy-move forgeries and highlight the forged regions as discussed in [2–11]. Nevertheless, these methods suffer from significant downsides, including their high computational demands and the need for more excellent stability in handling geometric transformations.

Keypoint-based techniques are another type of forgery detection approach. The primary steps include preprocessing, extracting keypoint features from the image, and feature matching. In 2011, Amerini et al. [12] introduced a CMFD technique based on key points. This approach involves extracting Scale Invariant Features Transform (SIFT) key points from images, focusing on 180 features. Instead of relying on direct Euclidean distance, the method calculates a ratio of distance pairs and utilizes a generalized 2NN approach to create clusters of nearest neighbors. These clusters then match key points and identify duplicated regions within the images. Later, several keypoint-based techniques are developed to identify copy-move forging images and identify the forged areas [2, 13–16]. The limitation of this technique is their decreased accuracy in handling small duplicated areas, which arises from the insufficient availability of key points. In parallel, several hybrid approaches are introduced in the literature, such as combining block and keypoint-based approaches. The author Mehta et al. [4] suggested a hybrid method by combining the Oriented FAST and Rotated BRIEF (ORB) features with keypoint-based feature sets (Discrete Cosine Transform (DCT)) derived from image blocks. Several hybrid techniques have been proposed, including those presented in [17–23].

The integration of block and keypoint techniques offers the potential for more accurate detection results. However, challenges like computational inefficiencies and the need for robustness against various image processing methods limit the overall effectiveness of these approaches. In recent years, the adoption of deep learning techniques has gained significant traction among researchers, demonstrating promising results in forgery detection and classification while enhancing the accuracy of CMFD attempted by previous methods. In these models, a deep learning model is trained using some training data, enabling the prediction image as either forged or original [8, 24–28]. Some approaches also utilized ground truth images during training and allowed the model to classify forgery images and localize the forgery regions [29–38] present in the image.

Earlier researchers have proposed various traditional and CNN architectures to improve performance in classifying and localizing copy-move forgery images. However, fine-tuning hyperparameters to obtain optimal parameter values to construct an efficient CNN network and regularizing the parameters remains challenging. Consequently, many studies have suggested integrating metaheuristic optimization algorithms to dynamically fine-tune the hyperparameters of CNN models during feature extraction, training, and the search for similar feature maps will enhance the efficiency of forgery detection methods. Therefore, our approach introduces a novel technique that integrates Particle swarm optimization and Gray wolf optimization algorithms to efficiently optimize the hyperparameters of a CNN model, enabling the extraction of significant image features for detecting copy-move image forgery. Table 1 provides a concise summary of recent copy-move forgery detection techniques relevant to our work.

Based on the remarks presented in the Table 1 above, the image forgery is simplified by the availability of robust image editing tools without much effort. Despite implementing various forgery detection methods targeting copy-move forgeries to enhance detection accuracy and forgery images post-processed with different global and local postprocessing operations. Yet, these approaches come with several limitations, some of which are mentioned here: (1) Specific image quality issues like noise, distortion, and compression artifacts may cause some approaches to become less accurate and lead to false positives or negatives. (2) The forgery images exhibiting significant variations in texture blur, brightness alterations, color reduction, and contrast adjustments affect the forgery detection process. (3) Several techniques suffer from high complexity, computation time, and less accuracy. (4) Forgery detection in rotation and scaling postprocessing operations needs significant improvement. Hence, these issues serve as a motivation for introducing this innovative concept. The main contributions of this work are outlined as follows:

- The proposed approach constructs an optimized Convolution Neural Network model by auto-tuning hyperparameters using hybrid PSGW optimization to extract significant image features effectively.
- Hybrid PSGW optimization enhances the CNN model to classify copy-move forgery images and localize the forgery regions accurately.
- Compare the quantitative detection results of the proposed model with the different state-of-the-art approaches evaluated on three distinct benchmark datasets.
- The proposed approach effectively addresses the plain image forgery and forgery images with global and local post-processing attacks such as image blurring, contrast reduction, JPEG compression, scaling and rotation postprocessing operations.
- An ablation test is performed by considering each dataset to highlight the substantial impact of optimization algorithms and significant contribution within the proposed system.

Table 1 Summary of recent forgery detection techniques

Ref#	Technique	Remarks
[39]	Block+Keypoint with Adaptive Galactic Swarm Optimization (AGSO)	This approach addresses the post-processing operations like scaling, rotation, noise addition, and JPEG compressions. However, the method must be extended for other operations such as image blurring, contrast reduction, etc
[40]	Block based with Adaptive Harris Hawk Optimization (AHHO)	Attains improved outcomes for forgery detection compared with the existing approaches. However, the method is not efficient for images with complicated backgrounds and textures
[41]	CNN with Sealion Customized Firefly (SCFF)	This method can address the plain copy-move, JPEG compression, noise inconsistency, and splicing type of image forgeries. It should be extended to address the other types of post-processing attacks, such as scaling, rotation and blend-off
[42]	Machine Learning (ML) with Hybrid Artificial Bee Colony with African Buffalo Optimization (HABC-ABO)	This method only addresses the scaling and rotation type of post-processed operations. The model's ability must be enhanced to address other sophisticated global post-processing operations
[43]	Autoregressive Elephant Herding (AEHO) based Generative Adversarial Network (GAN)	This method only addresses the plain copy move forgery attack. The process needs to address the various post-processing operations, such as rotation, scaling, and image blurring, which are essential to build a robust system
[44]	FMC clustering with Emperor Penguin Optimization (EPO)	Attains better performance against plain, forgery with brightness, contrast, scaling and rotation post-processed forgery images. This method cannot be applied to over-compressed images and the blend of post-processing operations
[45]	Deep Belief Network (DBN) with Adaptive Crow Search Algorithm (ACSA)	Able to identify single as well as multiple forgeries efficiently. However, forgery attacks like JPEG compression, texture and noise effects must be addressed
[46]	DL based SSDAE with Grasshopper Optimization Algorithm (GOA) and Spotted Hyena Optimizer (SHO)	Determines the best parameter combination solution and the improved SSDAE with the optimum input weights and hidden layers. However, the method does not address the various post-processing operations on forgery images
[47]	VGG16 with Hybrid Tuna Swarm with Bald Eagle Search Optimization (HTS-BESO) technique	This method efficiently addresses the copy-move forgery with various post-processing operations. However, it does not address the multiple forgery operations
[48]	EfficientNet with Seagull Pelican Optimization Algorithm (SPOA)	This method only classifies authentic and tampered images. The technique needs to be extended to locate the tampered regions
[37]	RESNET18 with PSO	This method only identifies the manipulated images containing Gaussian and salt and pepper distortions. Other post-processing operations, such as image blurring, JPEG compression, rotation and scaling, must be considered
[49]	Gaussian Mixture Model (GMM) with PSO	The research was only verified through experiments with six image forgery techniques, and additional enhancements are needed for thorough experimental outcomes
[50]	Deep Maxout Network (DMN) with SHGSO	Enhanced image forgery detection for legal and social issues. Improved accuracy over existing forgery detection techniques
[51]	CNN with Sequential Minimal Optimization (SMO)	Difficulty in differentiating original and forged images with software accessibility
[52]	Adaptive Neuro-Fuzzy Inference System (ANFIS)	This method is achieved success in detecting copy-move and splicing forgeries in digital images. But it does not address the localization of forgery regions and the post-processing attacks
[11]	DCT and Grey Level Co-occurrence Matrix (GLCM)	This approach achieved good results on both plain and forged images when subjected to global post-processing attacks. However, it does not account for local post-processing attacks such as scaling and rotation

3 Implementation details of proposed work

Deep Learning models have recently gained significant traction across various computer vision and image processing applications. This progress has inspired us to develop an optimized CNN model using a hybrid PSGW metaheuristic optimization technique to optimize various hyperparameters of the CNN model. Further, the optimized CNN architecture is used to extract significant image features and construct efficient feature maps to detect copy-move forgeries in digital images. The proposed model consists of five main phases: (1) Loading image dataset and preprocessing, (2) Optimized CNN model construction via hyperparameter optimization, (3) Feature extraction, (4) Investigation of patch similarity, and (5) Forgery detection. The proposed forgery detection and localization framework is depicted in Fig. 2, and the model design pseudo-code is illustrated in Algorithm 1.

3.1 Loading image dataset and preprocessing

In this step, the image dataset $I_{Dataset}$, which contains all N images comprising both forgery and authentic images, is loaded and fed to the optimized CNN model for the feature extraction process. This is shown in the Eq. 1 as follows,

$$I_{dataset} = \{I_1, I_2, I_3 .... I_k .. I_N\} \tag{1}$$

Where I_k indicates the k^{th} image in the database and I_N signifies the N^{th} image in the input dataset. The preprocessing aims to prepare data and make other processing operations easier. Therefore, during the preprocessing stage, each image in the dataset is resized to $256 \times 256 \times 3$ to ensure uniform input dimensions, which is necessary for deep learning models, particularly convolutional neural networks (CNNs), to process the image effectively.

3.2 CNN model construction and hyperparameter optimization

CNN stands out as a favored option in computer vision and image classification applications because of its adeptness in processing images efficiently, thanks to its layered structure. Among different layers of the CNN model, the convolution layer automatically selects the significant features from the image. The pooling layer produces reduced features. The dense layer is responsible for classification. Each layer holds significance in the CNN model, with the kernel size and quantity defining the number of features. The kernel weights are initially set randomly and then fine-tuned during model training. The ReLu layer processes the convolution layer's output to maintain values within a specific range, and a dropout layer prevents overfitting. The pooling layer reduces the size of the feature map, commonly employing max pooling to capture low-level features such as edges. Higher-level layers are generally fully connected (FC) layers that process the pooled output to make classification decisions. The final layer applies a softmax function to generate probabilities for multiclass classification.

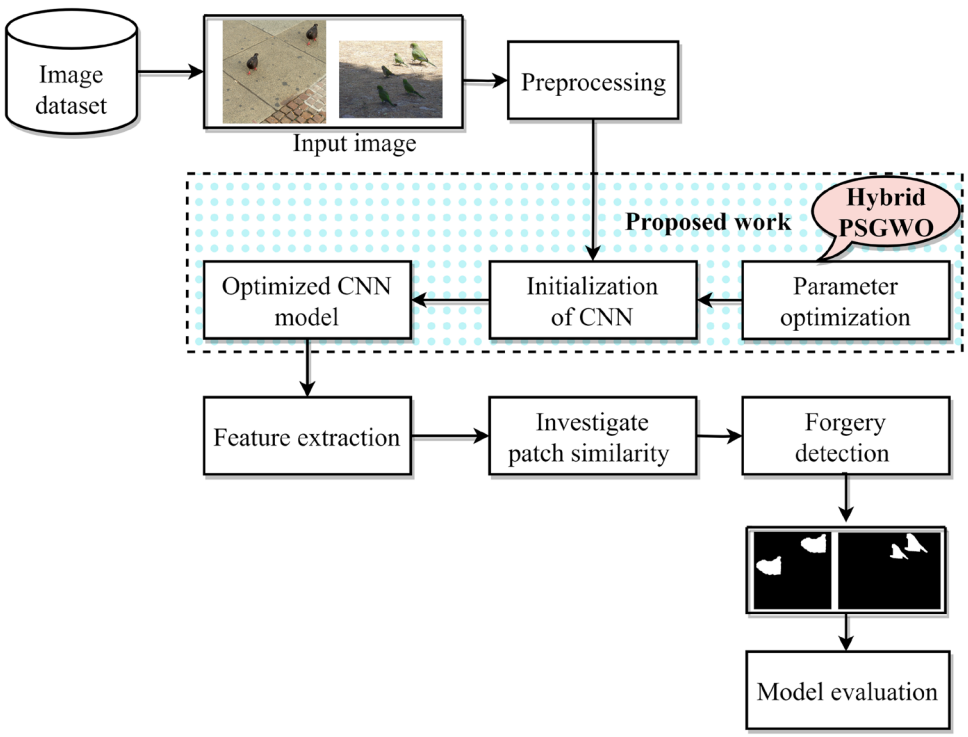

Fig. 2 Framework of proposed forgery detection methodology

Algorithm 1 Pseudo-code of proposed optimized CNN model design

INPUT: Image(Authentic or Forgery)
OUTPUT: i) Detect whether the image is authentic or forged.
ii) Highlight forgery regions.

Load the input image dataset
Count dataset images
Initialize Total_img=0
Check for file extensions .jpg,.jpeg,.png
if file has an image extension **then**
    Total_img= Total_img+1
    Repeat steps 4-6 for all files in directory and subdirectory
end if
Perform resize operation for each image
for i=1 to Total_img **do**
    Read the image
    Resize each input image into 256 × 256 × 3
end for
Initialize CNN model
init_CNNmodel = Sequential()
init_CNNmodel.add 2 CL(channels=64, kernal size=3 × 3)
init_CNNmodel.add 1 MPL(pool window= 2 × 2, strid=2 × 2)
init_CNNmodel.add 2 CL(channels=128, kernel size= 3 × 3)
.....
.....
init_CNNmodel.add 1 MPL(pool window=2 × 2, strid 2 × 2)
Apply hybrid PSGW optimization algorithm to optimize the hyperparameters of the CNN model using algorithm 2
Optimized_CNN=hybrid_PSGW(init_CNN)
Apply the Optimized_CNN model to extract image features
feature_maps= Optimized_CNN()
Investigation of feature similarity using equation(15)
Perform forgery detection to localize the forgery regions present in the image

The CNN model's hyperparameters include kernel length, number and types of kernels, stride within the kernel, activation function, and learning rate in the convolution layer are crucial factors linked to each layer's performance and functionality, and these hyperparameters significantly impact the result of the CNN model. Identifying a CNN's nearly ideal hyper-parameter configuration is a complex task, requires a lot of effort, and is costly when considering every possible combination. Consequently, appropriate CNN hyper-parameter refinement is viewed as an optimization problem to improve the CNN model's overall performance. The suggested approach optimizes the CNN model's hyperparameters by combining the strengths of both PSO and GWO optimization algorithms to construct an optimized CNN model to detect and localize the forgery images and regions efficiently.

3.2.1 Particle swarm optimization algorithm

The collective behavior of animal swarms inspires the PSO approach like bird flocks or fish schools as described by Eberhart and Kennedy [53]. The PSO is proficient at addressing both minimization and maximization problems. Each individual in the population is called a particle and is defined by three vectors and two actual values: one indicating the particle's position and the other representing its velocity change. The third vector is a record of the particle's best-found position. As a part of the PSO algorithm, each particle navigates the solution space based on its current velocity, the best solution (Pbest) and the overall best position in the population (Gbest). In the search space, the algorithm consistently retains many potential solutions. It refines these candidates iteratively by repeating the following three steps (optimizing the goal function) until the halting condition is satisfied:

1. Evaluate the suitability of every component.
2. Update the best fitness values on a local and global scale and the relevant rankings.
3. Adjust each particle's position and velocity as necessary.

This behavior is modeled using the Eqs. (2) and (3) as follows,

$$V_i(k+1) = w * V_i(k) + C_1 r_1(Pbest_i(k) - X_i(k)) + C_2 r_2(Gbest_i(k) - X_i(k)) \quad (2)$$

$$X_i(k+1) = X_i + V_i(k+1) \quad (3)$$

Where $V_i(k+1)$ and $X_i(k+1)$ define velocity parameters and the position of the i^{th} particle at instant k + 1. w signifies the inertia weight constant used to balance between global exploration and local exploitation. The parameters C1 and C2 are the self and group learning weights. r1 and r2 are random numbers uniformly distributed between [0, 1]. The positions of the known best particle in the total population are represented by variable *Pbest* and *Gbest*, respectively.

3.2.2 Grey wolf optimization algorithm

Grey wolf optimization was created by Mirjalili et al. [54] in 2014. It emulates the leadership structure of wolf packs known for their collaborative hunting. It categorizes search agents into Alpha, Beta, Delta, and Omega based on fitness. Alphas lead decision-making and hunting, Betas assist and discipline, Deltas manage Omega while scouting, and Omegas serve as the lowest rank. In optimization problems, the Alpha decision often proves the most effective. Swarm intelligence methods step in when there is no continuous supervision, with GWO enabling grey wolves to self-lead, inspired by their natural behavior. The GWO can tackle image recognition and classification and is incorporated into the swarm intelligence technique. The three stages of the search process include searching for prey, encircling, and attacking the prey, which is intended to resemble the hunting behavior of grey wolves. The exploration takes place within the first two stages, while exploitation is covered in the final stage. The mathematical model of GWO hunting behavior is shown below.

Each swarm agent's encircling behavior is illustrated through mathematical Eqs. (4) and (5) as follows.

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}_{(t)}| \quad (4)$$

$$\vec{D}(t+1) = |\vec{X}_p(t) - \vec{A} \cdot \vec{D}| \quad (5)$$

Where D is the encircling behavior of each agent, the parameter 't' indicates the current iteration; the pray position is defined by X_p, and the position of the Gray wolf is stated by X. The variables A and C are the coefficient vectors computed by using Eqs. (6) and (7) mentioned below,

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r1} - \vec{a} \quad (6)$$

$$\vec{C} = 2 \cdot \vec{r2} \quad (7)$$

Where 'a' is linearly dropped from 2 to 0, consequently, in many iterations, random vectors such as r1 and r2 whose values are selected within an interval of [0, 1]. The Grey Wolf's location is updated concerning the pray's location using Eq. (8) as follows.

$$\begin{aligned} \vec{D}_\alpha &= |C1 \cdot \vec{X}_\alpha - \vec{X}(t)|, \\ \vec{D}_\beta &= |C2 \cdot \vec{X}_\beta - \vec{X}(t)|, \\ \vec{D}_\delta &= |C3 \cdot \vec{X}_\delta - \vec{X}(t)| \end{aligned} \quad (8)$$

Differentiating between A and C will allow for the grasp of multiple areas surrounding the ideal search agent concerning the current location. The hunting process of the grey wolf can be modeled using the mathematical Eqs. (9) and (10) as follows,

$$\begin{aligned} \vec{X}_1 &= |\vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha)|, \\ \vec{X}_2 &= |\vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta)|, \\ \vec{X}_3 &= |\vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta)| \end{aligned} \quad (9)$$

$$\vec{X} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (10)$$

Where, D_α, D_β, and D_δ are distance vectors. X_α, X_β, and X_δ are the positions vectors of α, β and δ wolves respectively. The parameters A1, A2, A3, C1, C2, and C3 are coefficient vectors. The position of the Grey Wolves can be updated as random values ranging from [−2a to 2a], and the selected value is compared to a gap value. If $|A| \leq 1$ indicates an attack, then the focus turns to the prey's search and attack capabilities, guiding the population to approach the prey while avoiding divergence.

3.2.3 Optimizing CNN model by hybrid PSGW optimization algorithm

The hybrid optimization approach is implemented by integrating Particle swarm optimization and Grey wolf optimization techniques to optimize the CNN hyperparameters. These two optimization algorithms are combined to reduce the risk of getting stuck in local minima during the computation of optimal parameters. While PSO occasionally moves particles to random spots to escape these minima, its success rate is limited. So, to address this issue, the GWO's exploration capacity directs some particles to areas enhanced by GWO instead of random locations, thus boosting the method's effectiveness. This integration of PSO and GWO resulted in the hybrid PSGWO algorithm, which optimizes the algorithm's runtime by combining the strengths

Algorithm 2 Hybrid PSGW optimization process

```
Initialize the following parameters
a. Gray wolf population
Xi= 1,2,3,. . . .n
b. A, a, C and w
c.'n' wolves' places randomly for an agent ϵ[0, 1]
Determine α, β, and δ solutions based on the fitness function and agent's fitness level using equation 14.
while i ≤ M_iter do
    for each population do
        Update the velocity and the position of agents using equation 11 and equation 12, respectively
    end for
    Modify the parameters A, a, C and w
    Assess all particles using the objective function
    Revise the positions of the three best agents α, β, and δ
    Increment the iteration counter: i = i + 1
end while
```

of both techniques as formulated below in Eqs. (11), (12), and (13).

$$\begin{aligned}\vec{D}_\alpha &= |C1 \cdot \vec{X}_\alpha - w \cdot \vec{X}(t)|,\\ \vec{D}_\beta &= |C2 \cdot \vec{X}_\beta - w \cdot \vec{X}(t)|,\\ \vec{D}_\delta &= |C3 \cdot \vec{X}_\delta - w \cdot \vec{X}(t)|\end{aligned} \tag{11}$$

The PSO and GWO variations are combined using Eqs. (12) and (13) to update the velocities as,

$$\begin{aligned}V_i(k+1) = w * V_i(k) + C_1 r_1 (X_1 - X_i(k))\\ + C_2 r_2 (X_2 - X_i(k)) + C_3 r_3 (X_3 - X_i(k))\end{aligned} \tag{12}$$

$$X_i(k+1) = X_i(k) + V_i(k+1) \tag{13}$$

The fitness function is designed to optimize the Rosen-Brock function efficiently, which is crucial in identifying the optimal value. This fitness function is derived specifically to enhance the optimization process in the proposed work. The Eq. (14) defines a compatible coordinate system, departing from the traditional approach of building local approximation models like f(x) or depending on gradient information.

$$f(x) = \sum_{i=1}^{N-1} [100(X_{i+1} - X_i^2) + (1 - X_i)^2] \tag{14}$$

Algorithm 2 presents the pseudo-code for the hybrid Particle Swarm Gray Wolf Optimization (PSGWO) algorithm. The critical parameters for optimizing the network architecture include the number of convolution layers, kernel sizes used in each convolution operation, batch size, and the number of filters employed for feature map extraction. The CNN model is initialized to engage in collaborative

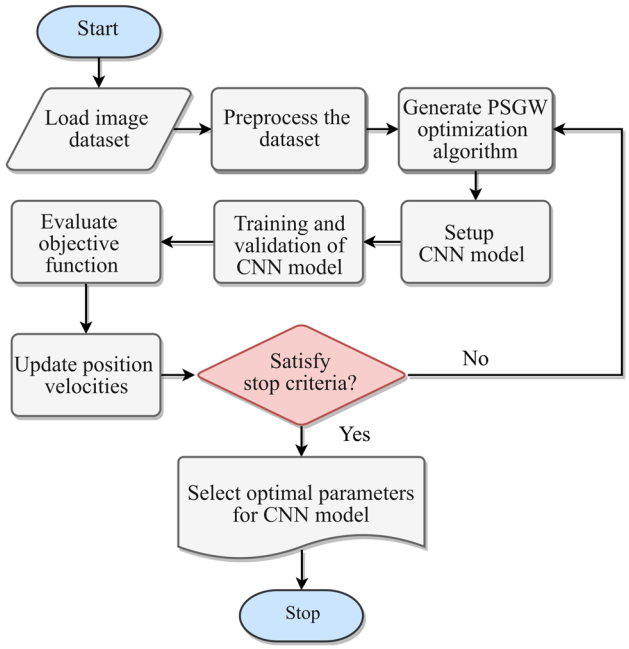

Fig. 3 Flow diagram of hybrid PSGW optimization process

parameter optimization using the hybrid PSGW algorithm. The hybrid PSGW is configured based on specified execution parameters leading to the generation of particles. Each agent embodies a complete CNN training representing a potentially optimal solution with each position containing a parameter value targeted for optimization. Figure 3 illustrates a flow diagram of the CNN optimization process using the hybrid PSGW optimization technique.

3.3 Feature extraction

During this phase, the focus shifts to feature extraction from the pre-processed image, leveraging the optimized CNN

model. The initial CNN model undergoes fine-tuning using a hybrid PSO and GWO algorithm to ascertain pertinent parameter values crucial for the image feature extraction. Subsequently, the fully connected nodes are eliminated from the network's conclusion. The altered architecture of the optimized CNN network is illustrated in Fig. 4, showing the removal of fully connected layers and their replacement with the final pool layer. As a result, the output of the last max-pooling layer serves as a feature extraction component of the model, presenting itself as feature vectors or feature maps. These features are subsequently forwarded to the next stage, where they are investigated for similarity to detect forgery images and regions within the input images.

3.4 Investigation of patch similarity

The feature extraction module produces 512, 16 × 16 feature patches based on the suggested framework. This step investigates the correlation distribution of neighboring feature vectors to acquire the possible duplicated and relocated patch pairs. The correlation consistency is crucial for evaluating how compatible these feature vectors are with one another and helps to identify tampering characteristics. We employ the Pearson Correlation Coefficient(PCC), as shown in Eq. (15), to quantify the feature similarity. The patch similarity module generates 256 patches sized 16 × 16, identifying pairs of similar patches based on their correlation coefficient. A higher correlation coefficient signifies a more remarkable similarity between the patch pairs.

$$PCC = \frac{\sum_{j=1}^{N}(X_j - \bar{x})(y_j - \bar{y})}{\sqrt{\sum_{j=1}^{N}(X_j - \bar{x})^2(y_j - \bar{y})^2}} \tag{15}$$

Where x_j and y_j denote the extracted features and $\bar{x}$ and $\bar{y}$ signifies the mean of each x_j and y_j respectively.

3.5 Forgery detection

The patch similarity module generates a 16 × 16 feature block at a lower resolution than the input image. A high-dimensional output is essential to accurately localize the forged region within the image. Traditional max-pooling layers significantly reduce the image size, making precise localization challenging. To address this, an upsampling process is employed to enlarge the smaller feature map to a larger size. Unlike max-pooling filters, which condense information, upsampling filters replicate rows and columns from the previous layer's output based on a specified stride. Within the forgery detection module, simple bilinear upsampling is applied twice combined with BN-Inception before passing through an additional upsampling block. The resulting upsampled 256 × 256 feature block is then processed through the BN-Inception network again [31]. Finally, a detection map is produced using a 1 × 1 convolution layer followed by a sigmoid activation function.

To assess the accuracy of pixel classification, the model employs the binary cross-entropy loss function. This function is a metric for evaluating how well the model fits the provided data. When the model's predictions significantly differ from the actual values, the loss function yields a significant value; conversely, when the projections align closely with reality, the loss function returns a smaller value. The Eq. (16) defines the loss function as follows,

$$T_{loss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i^t \ln(y_i) + (1 - y_i^t)\ln(y_i)] \tag{16}$$

Where y_i^t is the true map of the i^{th} image, y_i is the output of the network of the i^{th} image, N is the number of training images, and T_{loss} denotes the loss function.

4 Experimental setup

The proposed model is developed using the PyCharm IDE with Python 3.0, utilizing libraries such as TensorFlow, OpenCV, Matplotlib, and NumPy. The hardware support includes 11^{th} generation Core i7 processor, 128GB SSD storage, 16GB DDR4 RAM, and a 4GB NVIDIA GeForce GTX 1050Ti GPU.

The CNN model's starting parameters are established statically, such as the activation function of Relu and Sigmoid

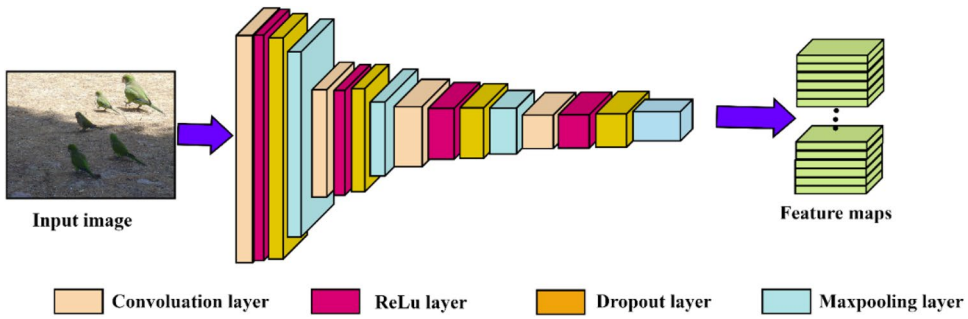

Fig. 4 Optimized CNN network for feature extraction

for optimizing as an Adam and epochs=100. Similarly, the number of particles, iterations, inertia weight upper bound, lower bound, threshold and accelerator factor (C1, C2, C3) for hybrid PSGW are set as [5, 0.5, 100, 100, 10, 0.5, 2]. The dynamic parameters that hybrid PSGW optimizes for CNN are the number of hidden layers, the size of the filters used in each hidden layer, the number of filters, and the batch size. This section also details the datasets used for evaluation, the evaluation metrics considered for model assessment, the results and discussion, and the ablation study to understand the contribution of optimization techniques

4.1 Dataset description

The proposed approach utilizes three diverse benchmark datasets consisting of copy-move forgery images, namely, CoMoFoD [55], CMFD [56], and MICC-F600 [57] to experiment. The CoMoFoD dataset comprises 200 authentic and 200 plain copy-move forgery images and 600 images for each forgery image subjected to postprocessing operations such as noise addition, brightness adjustment, contrast modification, color reduction, and image blurring performed at three different levels. The dataset includes 1,800 images altered by JPEG compression with 10 varying compression factors.

The CMFD dataset comprises copy-move forgery images of medium-sized BMP format images measuring 1000 × 700 or 700 × 1000 pixels. It consists of 50 original and 50 plain forgery images, with 320 and 600 forgery images exposed to rotation and scaling postprocessing attacks.

The MICC-F600 dataset comprises images featuring plain copy-move forgery and forgery images post-processed with geometric transformations. This dataset includes 600 high-resolution images of about 800 × 532 pixels, of which 448 are original, and 152 are manipulated using copy-move forgery and post-processed with rotation, scaling, and multiple cloning attacks. The forged images in the dataset exhibit varying sizes and shapes in the tampered regions, making this dataset particularly challenging.

The CoMoFoD dataset is specifically used to detect forgery images that have undergone global postprocessing operations such as image blurring, contrast reduction, JPEG compression etc, where the modifications are applied to the entire image rather than being limited to the copy-move forgery region. In contrast, the CMFD and MICC-F600 datasets are used for detecting copy-move forgery images that have undergone local postprocessing operations, such as scaling and rotation, where the modifications are applied solely to the copy-move forgery region instead of the entire image. Each tampered image has its matching ground truth image, often used to evaluate detection results. To execute the requirements, we have separated the dataset images into training, validation, and testing at a ratio of 75:15:10 for experimentation.

		Predicted	
		Mask image pixel as a 1(Forged)	Mask image pixel as a 0(Authentic)
Actual	Ground truth mask pixel as a 1(Forged)	TP	FN
	Ground truth mask pixel as a 0(Authentic)	FP	TN

Fig. 5 Simplification of TP, FP, FN and TN for pixel level evaluation

4.2 Evaluation of the model and result discussion

The confusion matrix is used to assess the model performance at the pixel and image levels. The resultant image pixels produced by forgery detection module are compared to the matching ground truth mask pixel at the exact location in the forged image to assess the pixel level. The simplification of the parameters of the confusion matrix to evaluate at the pixel level is depicted in Fig. 5. The parameter TP signifies the count of accurately identified forged pixels, FP is the tally of pixels mistakenly identified as forged, FN is the number of falsely overlooked forged pixels, and TN is indeed detected unforged pixels, respectively. Based on these parameters, the following evaluation metrics are formulated and expressed in the form of Eqs. (17–23) as follows,

$$Precision\,(P) = \frac{T^p}{(T^p + F^p)} \tag{17}$$

$$Recall\,(R) = \frac{T^p}{(T^p + F^n)} \tag{18}$$

$$F1 - score\,(F1) = 2 * \frac{(P * R)}{(P + R)} \tag{19}$$

$$Accuracy\,(A) = \frac{(T^p + T^n)}{(T^p + F^n) * (F^p + T^n)} \tag{20}$$

$$True\ Negative\ Rate\,(TNR) = \frac{T^n}{(T^n + F^p)} \tag{21}$$

$$False\ Negative\ Rate\,(FNR) = \frac{F^n}{(F^n + T^p)} \tag{22}$$

$$MCC = \frac{(T^p * T^n) - (F^p * F^n)}{\sqrt{(T^p + F^p)(T^p + F^n)(T^n + F^p)(T^n + F^p)}} \tag{23}$$

The performance analysis of the suggested model was gauged on images collected from three diverse datasets using

Table 2 Detection results obtained for CoMoFoD dataset against plain and postprocessed forgery images

Forgery attacks	Levels	P	R	A	F1	TNR	FNR	MCC
Plain forgery (PF)	–	0.97	0.97	0.95	0.97	0.86	0.02	0.83
JPEG compression (JC) (Compression factor)	20	0.96	0.97	0.94	0.96	0.81	0.02	0.80
	30	0.96	0.97	0.94	0.96	0.81	0.02	0.80
	40	0.96	0.97	0.94	0.96	0.81	0.02	0.80
	50	0.98	0.98	0.96	0.98	0.90	0.01	0.88
	60	0.97	0.97	0.95	0.97	0.86	0.02	0.83
	70	0.97	0.97	0.95	0.97	0.86	0.02	0.83
	80	0.96	0.97	0.94	0.96	0.82	0.02	0.81
	90	0.96	0.88	0.94	0.91	0.82	0.02	0.81
	100	0.98	0.98	0.96	0.98	0.90	0.01	0.89
Noise addition(NA) ($\mu = 0$ and σ^2)	0.009	0.98	0.98	0.96	0.98	0.85	0.01	0.83
	0.005	0.98	0.98	0.96	0.98	0.86	0.01	0.84
	0.0005	0.93	0.92	0.96	0.92	0.86	0.01	0.84
Image blurring (IB) (Variance)	0.009	0.97	0.97	0.95	0.97	0.85	0.02	0.82
	0.005	0.97	0.97	0.95	0.97	0.85	0.02	0.82
	0.0005	0.97	0.89	0.95	0.92	0.86	0.02	0.83
Brightness change (BC) (Brightness level)	(0.01,0.95)	0.97	0.98	0.96	0.97	0.87	0.01	0.86
	(0.01,0.9)	0.97	0.97	0.95	0.97	0.87	0.02	0.84
	(0.01,0.8)	0.97	0.97	0.95	0.97	0.91	0.02	0.89
Color reduction(CR) (Intensity level per color channel)	32	0.98	0.97	0.95	0.97	0.87	0.02	0.82
	64	0.98	0.97	0.95	0.95	0.88	0.02	0.83
	128	0.98	0.97	0.95	0.97	0.88	0.02	0.84
Contrast adjustment (CA) (Contrast range)	(0.01,0.95)	0.97	0.97	0.95	0.97	0.85	0.02	0.82
	(0.01,0.9)	0.97	0.97	0.95	0.97	0.86	0.02	0.83
	(0.01,0.8)	0.97	0.97	0.95	0.97	0.86	0.02	0.83

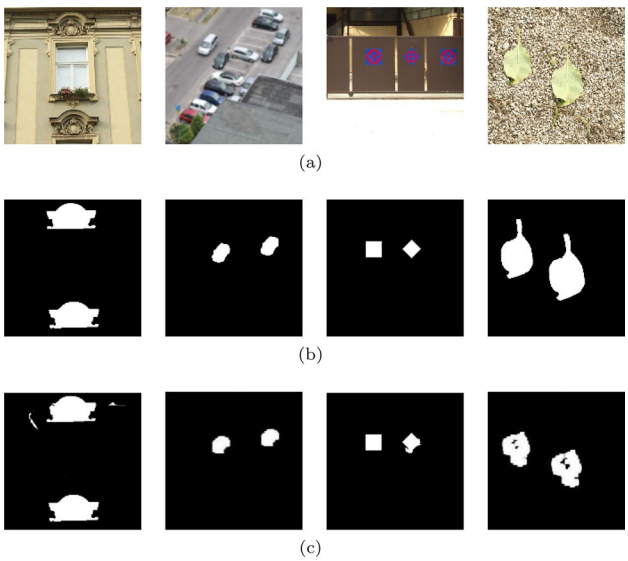

Fig. 6 Forgery detection results obtained for CoMoFoD dataset. (**a**) Forgery image, (**b**) Ground truth, (**c**) Forgery detection result

the evaluation metrics mentioned above. The model is tested using 10% of the excluded images from each dataset, while the remaining images were utilized during the training and validation. After that, the precision, recall, Accuracy, TNR, FNR, F1-score, and MCC values are computed for plain and forgery images with different post-processing applied on images. Table 2 tabulates statistical results obtained for the CoMoFoD dataset for plain forgery and forgery images subjected to different post-processing attacks and Fig. 6 shows the visualization of the detection results.

To understand the model performance the obtained results tested using CoMoFoD dataset are compared with DCT+GLCM [11], BDF+CH [58], Tetrolet Transform [59], DCT+PH [60], and Adaptive Oversegmentation [61] methods to validate the robustness of proposed work against Image blurring, Brightness change, Color reduction and Contrast adjustment post-processing attacks. Figures 7, 8, and 9 illustrate these post-processing attacks comparative analysis in terms of precision, recall and F1-score with existing approaches.

Furthermore, we utilized the CMFD dataset to evaluate the performance of the proposed model against post-processing operations such as translation, rotation, and scaling. Specifically, we considered plain forgery images and forgery images subjected to rotation and scaling. The rotation range was set to $-25°$ and $25°$ with a step size of $5°$, while the scaling factor ranged from 0.25 to 2 with increments of 0.25. Table 3 and Fig. 10 shows the detect results and visualization of detection results for plain forgery and forgery images with rotation and scaling post-processing attacks.

To represent the diversity of forgery detection scenarios and demonstrate a comprehensive assessment of the

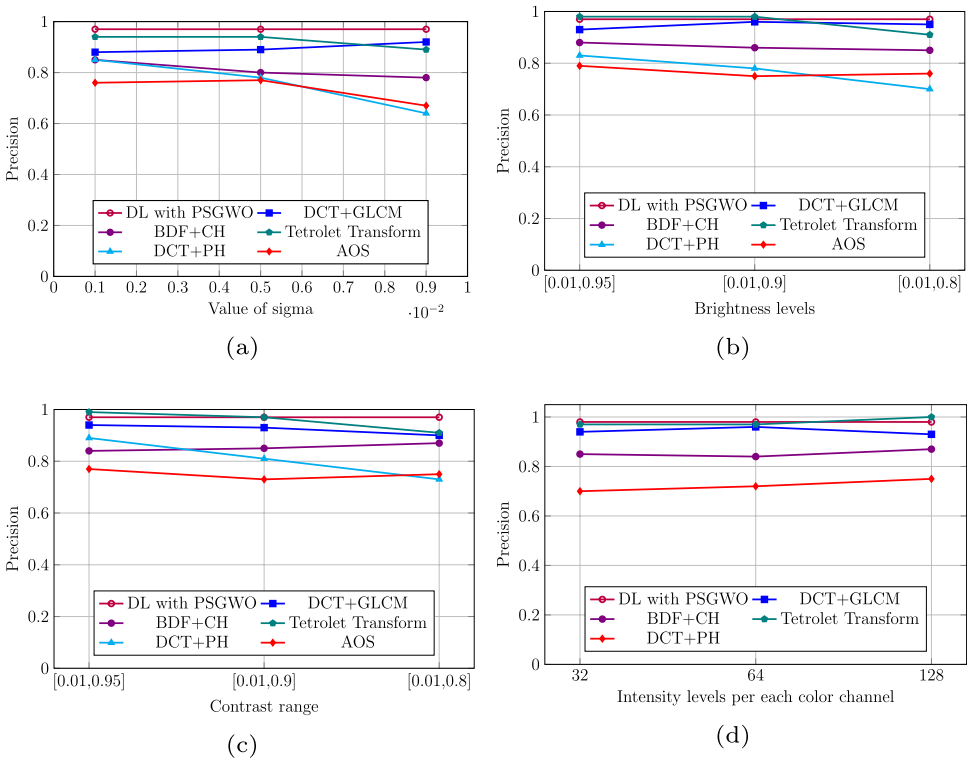

Fig. 7 Precision analysis for (**a**) Image blurring, (**b**) Brightness change, (**c**) Color reduction, and (**d**) Contrast adjustment post-processing attack

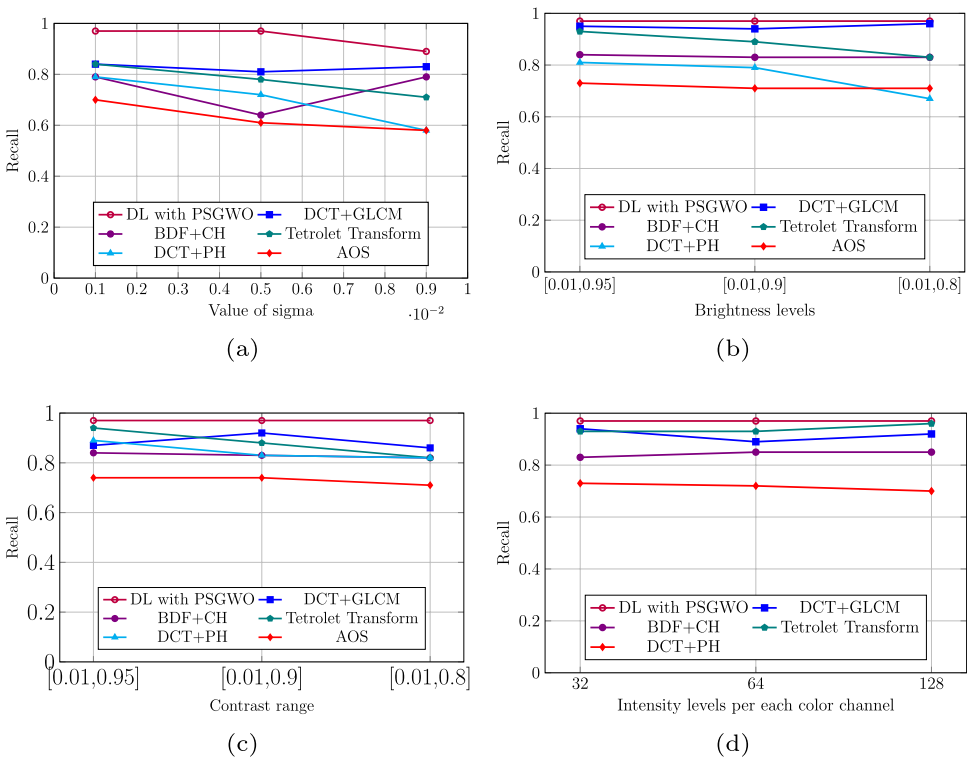

Fig. 8 Recall analysis for (**a**)Image blurring, (**b**) Brightness change, (**c**) Color reduction, and (**d**) Contrast adjustment post-processing attack

model's functionality and capacity to generalize across varied scenarios. The proposed work is validated by using a MICC-F600 dataset, and obtained results are compared with Block+keypoint with AGSO [39], Segmentation [17], SIFT+NMS [2], Feature point [18] and Hybrid feature based [19] approaches which are tested against Rotation, Scaling, Additive noise and JPEG compression post-processing attack. The rotation angle is changed between 2° to 10° with the step of 1°, and in scaling post-processing attack, the forged region is varied with the scale ratio of 1.01 to 1.09

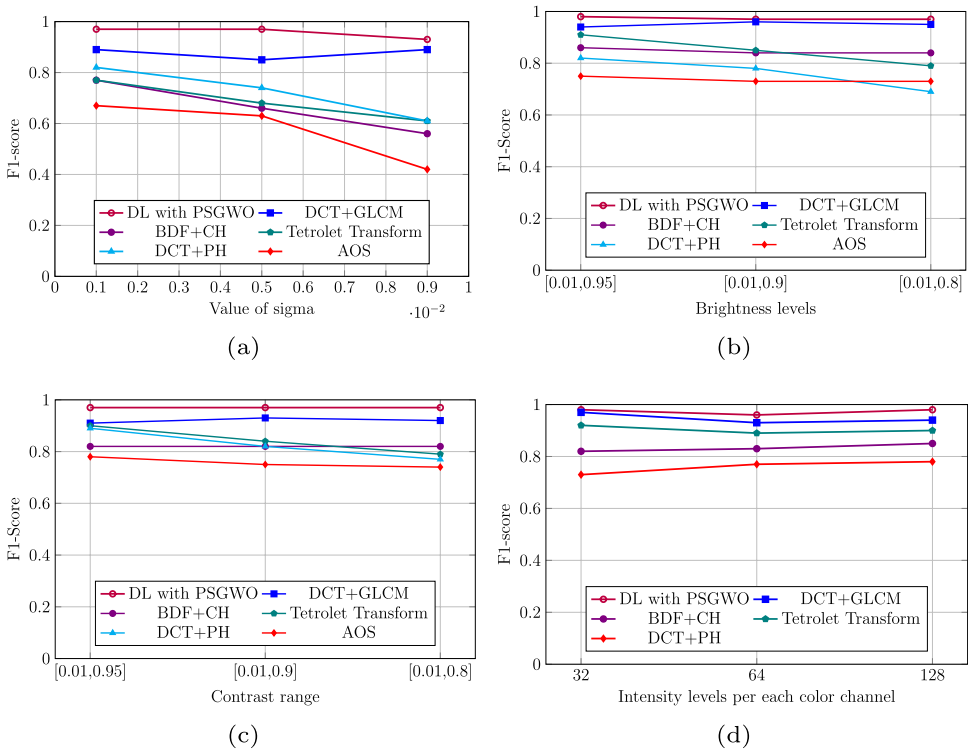

Fig. 9 F1-score analysis for (**a**) Image blurring, (**b**) Brightness change, (**c**) Color reduction, and (**d**) Contrast adjustment post-processing attack

Table 3 Detection results obtained for CMFD dataset

Evaluation metrics							
Forgery attacks	P	R	A	F1	TNR	FNR	MCC
Translation	0.97	0.97	0.95	0.97	0.66	0.02	0.66
Rotation	0.97	0.92	0.95	0.95	0.84	0.02	0.81
Scaling	0.97	0.97	0.95	0.97	0.80	0.02	0.77

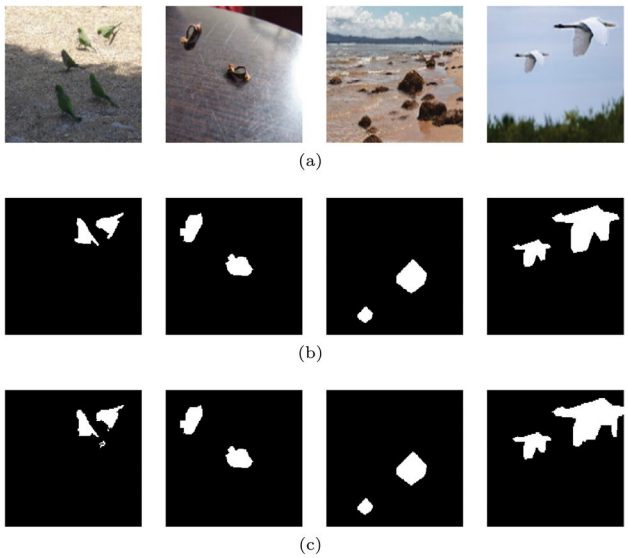

Fig. 10 Forgery detection results obtained for CMFD dataset. (**a**) Forgery image, (**b**) Ground truth, and (**c**) Forgery detection result

with the step of 0.01. Furthermore, the image is degraded with Gaussian noise by varying the standard deviation from 0.02 to 0.10. Also, to validate the robustness against JPEG compression, the image quality factor has been changed between 20 and 100 with the step of 10. Figures 11, 12 and 13 illustrate the performance comparison in terms of precision, recall and F1-score and Fig. 14 depicts the sample results obtained for the MICC-F600 dataset.

The suggested approach produces better results for typical copy-move forgery procedures than the state-of-the-art methods in nearly all test cases. It remains unchanged when subjected to geometric modifications like rotation and scaling. Furthermore, the technique is unaffected by post-processing actions such as color correction, contrast enhancement, noise addition, brightness enhancement, JPEG compression, and image blurring.

The comparisons and outcomes discussed above are analyzed at the pixel level. Pixel-level analysis can divide an image's pixels into forged and un-forged categories. As a result, a pixel-level analysis can identify the location of a forgery region in an image. Another type of research entails looking at images individually to ascertain whether images

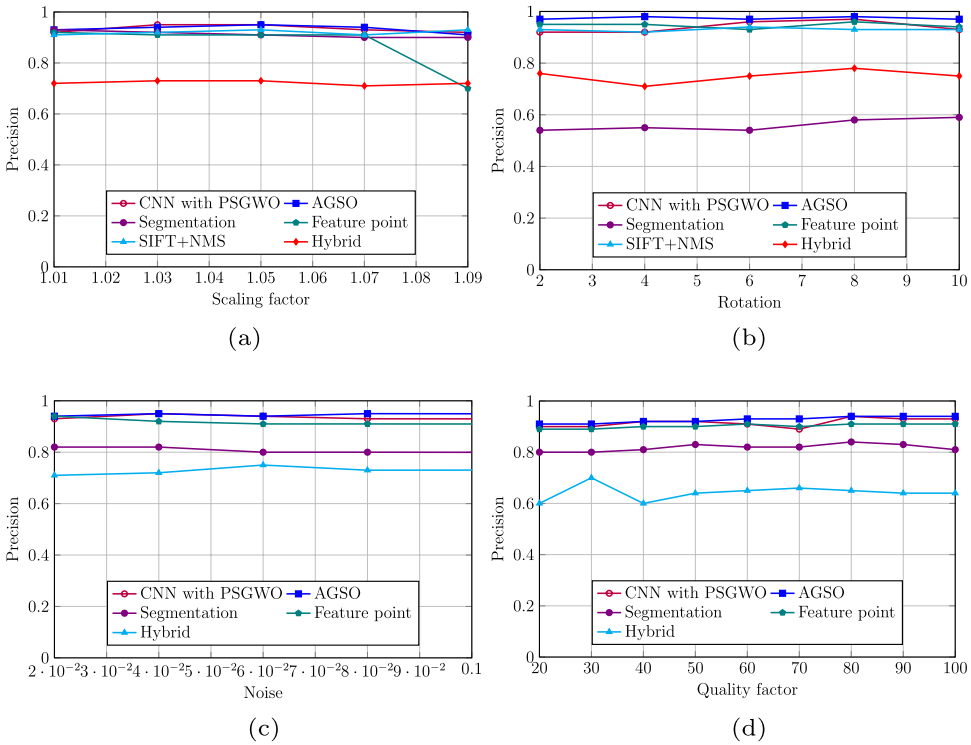

Fig. 11 Precision analysis for (**a**) Scaling, (**b**) Rotation, (**c**) Noise, and (**d**) JPEG compression attack

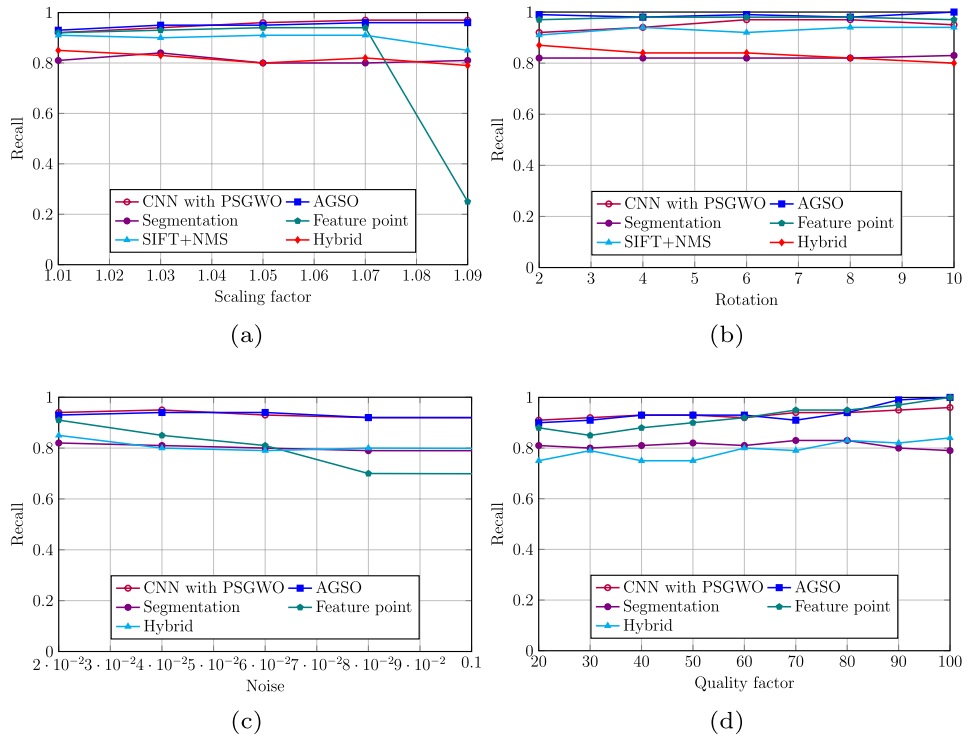

Fig. 12 Recall analysis for (**a**) Scaling, (**b**) Rotation, (**c**) Noise, and (**d**) JPEG compression attack

have been altered or not. The suggested model evaluates images based on true-negative, false-positive, and false-negative pixel counts. The trained model analyses original photos at the image level, using 100, 50 and 16 images from the CoMoFoD, CMFD and MICC-F600 datasets. When the model predicts a low number of false-positive and false-negative pixels and zero true-negative pixels, the image is considered non-tampered; if not, it is considered tampered. Figure 15 shows the confusion matrices that resulted from these classifications for three datasets. Next, based on these values, the suggested model's performance is computed, and the average results are tabulated in Table

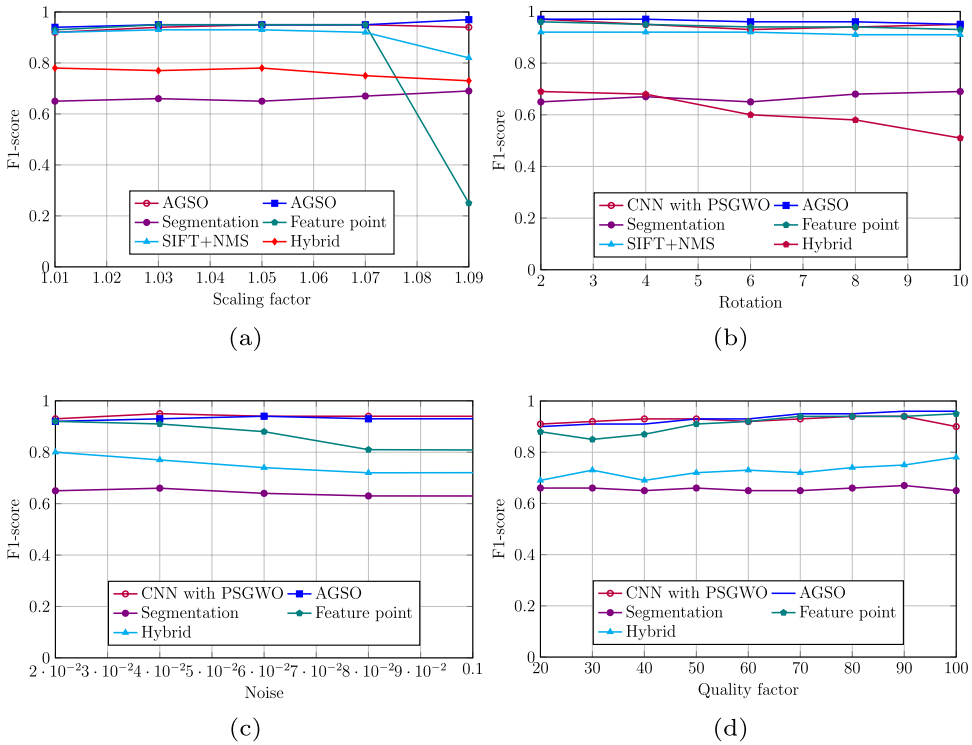

Fig. 13 F1-score analysis for (**a**) Scaling, (**b**) Rotation, (**c**) Noise, and (**d**) JPEG compression attack

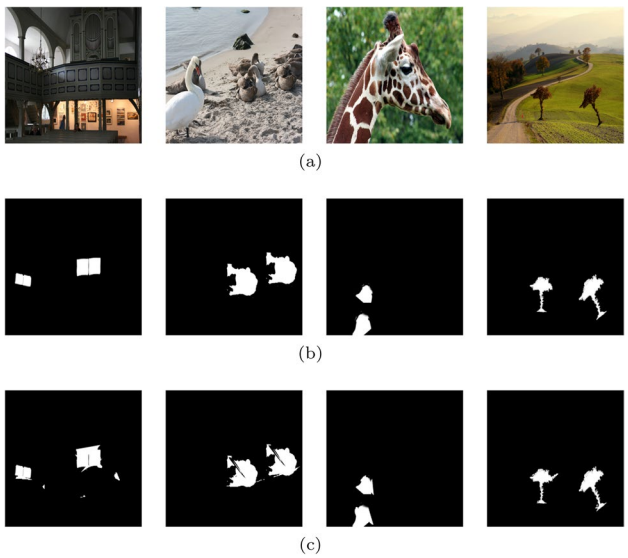

Fig. 14 Forgery detection results obtained for MICC-F600 dataset. (**a**) Forgery image, (**b**) Ground truth, and (**c**) Forgery detection result

4 for image and pixel-level evaluation for all three datasets. As per the results, the proposed approach demonstrated strong performance on the CoMoFoD and CMFD datasets at both the image and pixel levels. However, the performance on the MICC-F600 dataset was comparatively lower. This decline is attributed to complex alterations, variations in image quality, compression artifacts, and a limited number of forgery and authentic image samples.

Tables 5 tabulates a comparative analysis of performance metrics Precision(P), Recall(R), and F1-score across block, keypoint and deep learning with optimization techniques to understand the effectiveness of our proposed model. The block and keypoint-based method by [39] achieved moderate performance, with an F1-score of 92.75% and exhibited a robust balance between a precision of 92.45% and a recall of 93.63%. The method suggested by [9] demonstrated a comparatively lower F1-score of about 88.40%, indicating potential limitations in recall that constrained the method's overall effectiveness. The authors [11] approach showed improved results, with an F1-score of 92.86%, showing a well-rounded trade-off between precision of 93.40% and recall of 90.18%.

The Deep Learning approach suggested by [30] yields a lower F1-score of 69.62%, primarily due to a significant gap between precision 81.02% and recall 61.05%. These results highlight challenges in generalization and dataset representation compared to [31] and [33] approaches. The [31] method achieved an improved F1-score of 90.5%, although the precision of 99.5% was notably higher than recall 87.2%, suggesting potential overfitting or imbalanced predictions. The approach proposed by [33] showed well-balanced results over both the deep learning approaches mentioned above with a precision of 95.51%, recall of 93.21% and F1-score of 94.34%. The proposed approach significantly enhanced the detection performance over block-based and deep-learning-based methods by integrating advanced optimization techniques into deep-learning frameworks. The

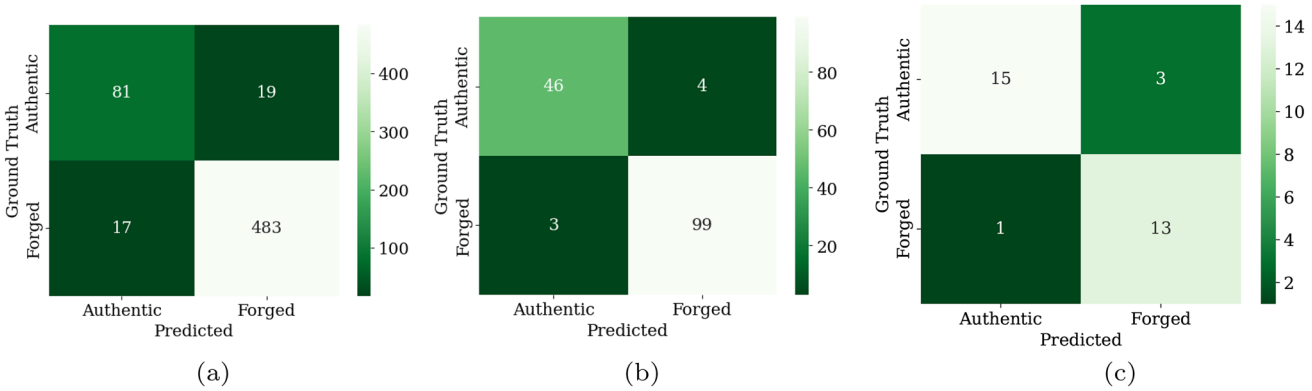

Fig. 15 Confusion matrix obtained for (**a**) CoMoFoD, (**b**) CMFD, and (**c**) MICC-F600 datasets

Table 4 Average detection results obtained at image and pixel levels for different datasets

Image level							
Evaluation metrics							
Dataset	P	R	A	F1	TNR	FNR	MCC
CoMoFoD	0.96	0.96	0.94	0.96	0.81	0.03	0.78
CMFD	0.96	0.97	0.95	0.96	0.92	0.02	0.89
MICC-F600	0.81	0.93	0.88	0.87	0.83	0.07	0.76
Pixel level							
Evaluation metrics							
Dataset	P	R	A	F1	TNR	FNR	MCC
CoMoFoD	0.97	0.96	0.95	0.96	0.86	0.02	0.84
CMFD	0.97	0.95	0.95	0.96	0.77	0.02	0.75
MICC-F600	0.92	0.94	0.93	0.94	0.87	0.04	0.81

Table 5 Comparison with existing approaches

Ref#	Methodology	Methods	P (%)	R (%)	F1 (%)
[39]	Block/Keypoint	Block+keypoint with AGSO	92.45	93.63	92.75
[9]		BDF and CH	90.60	86.70	88.40
[11]		DCT and GLCM	93.40	90.18	92.86
[30]	Deep learning	CNN	81.02	61.05	69.62
[31]		CNN	99.50	87.20	90.50
[33]		CNN	95.51	93.21	94.34
[41]	Deep learning with optimization	CNN with SCFF	96.93	97.69	97.05
[45]		DBN with ACSA	97.10	98.90	96.10
[46]		DL with GOA and SHO	95.62	93.08	93.19
Proposed		CNN with PSGWO	95.66	94.87	96.17

results demonstrated that our method achieved superior precision, reaching approximately 95.66%, outperforming all other approaches and marginally less in precision compared to deep learning models with optimization techniques compared to [31, 41, 45]. While its F1-score metrics showed improvement across all approaches slightly surpassed by the [41] method, the recall showed improvement over all other methods except for [31, 41, 45]. A comprehensive analysis of the findings confirms that the proposed approach delivers accurate and consistent results, effectively addressing the problem with more reliability than other methodologies.

To comprehend the practicality and applicability of the results for real-world implementation. The proposed model's computational cost is analyzed and compared with alternative models, as shown in Fig. 16. The comparison includes the Adaptive Neuro-Fuzzy Inference System [52], Keypoint and Block and Keypoint-based methods [2, 39], Deep Learning techniques with optimization [33, 46], and Block-based approaches [11]. The results indicate that the proposed method requires slightly more execution time than ANFIS, Deep Learning with GOA and SHO, DL SLIC + SURF, and DCT + GLCM techniques due to the additional time required to optimize CNN parameters during feature extraction and perform matching operations across all feature maps in the forgery detection process. Despite this, the optimization process significantly improves the results regarding precision and F1-score compared to other existing methods.

Notably, the proposed approach substantially reduces execution time compared to SIFT-based, and Block & Keypoint with AGSO methods. Furthermore, it achieves superior accuracy compared to these methods. Hence, the

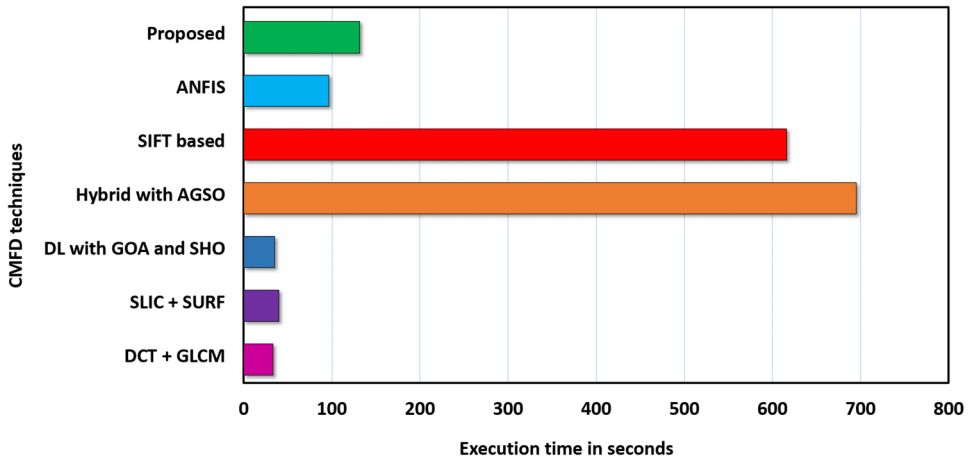

Fig. 16 Analysis of execution time

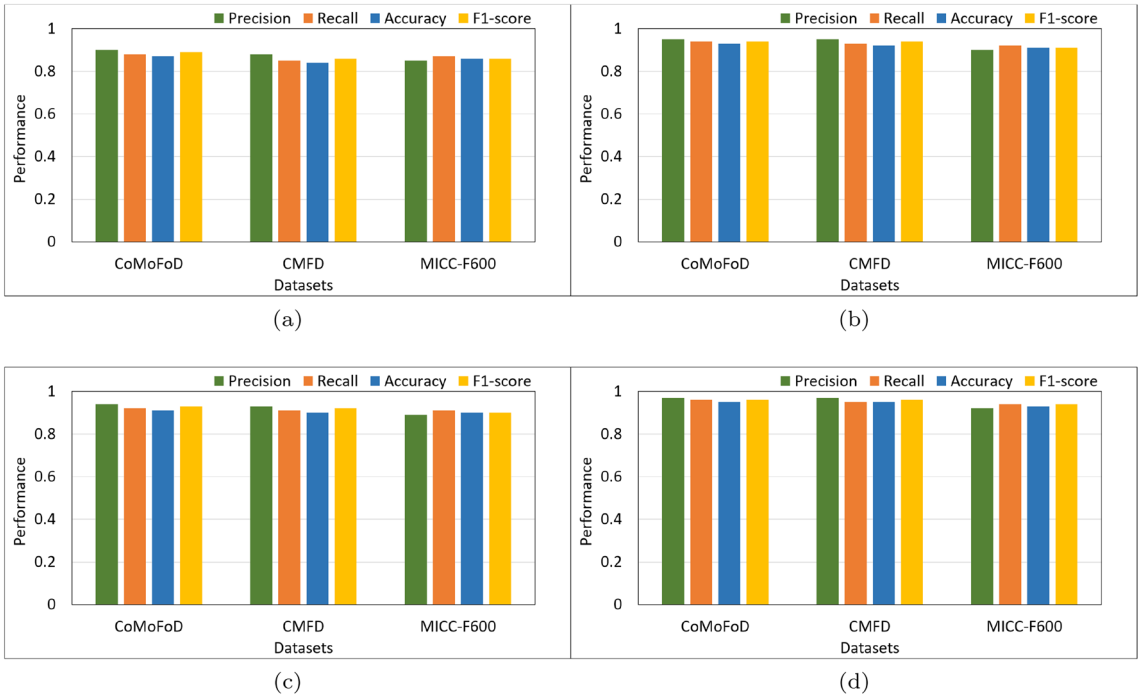

Fig. 17 Ablation analysis over different datasets. (**a**) CNN only, (**b**) Without GWO (CNN + PSO), (**c**) Without PSO (CNN + GWO), and (**d**) Full system (CNN + PSO + GWO)

proposed method can efficiently identify duplicated regions while maintaining reduced computational overhead.

4.3 Ablation study

The proposed model integrates Particle Swarm Optimization (PSO) and Gray Wolf Optimization (GWO) techniques for Convolutional Neural Network (CNN) architecture. These optimization algorithms play a critical role in fine-tuning the hyperparameters of the CNN model, thereby enhancing its performance. To evaluate the contribution of each component in the proposed method, we conducted a series of ablation experiments by incrementally adding the optimization modules to the base architecture and assessing the model's performance on three datasets.

The results of the ablation study are illustrated in Fig. 17. The baseline CNN model without PSO and GWO, as shown in Fig. 17a, exhibits a substantial decline in performance metrics. The standalone CNN struggles with threshold tuning and feature optimization, resulting in significantly lower scores across all metrics. When PSO is introduced to optimize the CNN hyperparameters, the model demonstrates notable improvements in recall, F1 score, and accuracy. However, the precision metric is adversely affected due to an imbalance between exploration and exploitation, increasing false positives, as shown in Fig. 17b.

Similarly, the incorporation of GWO alone for CNN hyperparameter optimization results in a slight decline in the recall, attributed to less effective global optimization capabilities, as depicted in Fig. 17c. The best performance is achieved by combining PSO and GWO to optimize the CNN hyperparameters, significantly enhancing detection performance across all metrics. The proposed model thus makes a crucial contribution to Copy-Move Forgery Detection (CMFD) performance, as demonstrated in Fig. 17d.

Particle Swarm Optimization (PSO) and Gray Wolf Optimization (GWO) are heuristic optimization techniques that inherently involve iterative computation and evaluation of candidate solutions. Hence, the computational costs may add further overhead due to the coordination between the two algorithms. However, the hybridization of PSO and GWO leverages the strengths of both algorithms—PSO's capability for global exploration and GWO's efficient local search mechanisms to achieve more robust and optimal CNN hyperparameters. The proposed optimized CNN model has significant potential in real-world applications such as forensic investigations to streamline the detection and localization of tampered images, aiding in criminal investigations, court evidence validation, and cybercrime analysis. Its high accuracy and localization precision make it valuable for identifying manipulated regions in digital evidence. Similarly, the model can be integrated into social media platforms, news agencies, and digital marketplaces to flag the manipulated content and ensure its automatic authenticity.

However, scalability and processing speed limitations could challenge real-time applications, especially with high-resolution images or large datasets and robustness may be reduced while dealing with advanced forgery techniques like adversarial attacks or deepfake manipulations. Also, false positives arise in scenarios with repetitive textures or patterns, necessitating careful calibration to balance sensitivity and specificity.

5 Conclusion

Copy-move forgery is a malicious image tampering operation that conceals information by duplicating and altering regions. This paper proposes an optimized CNN model with a hybrid PSGW optimization technique to detect and localize accurately forgeries, including plain and post-processed forgery images. The method is validated on three datasets to assess its effectiveness. As per the results, the proposed model performed better than other existing methods and showed an improved precision of 3.82%, recall of 2.97%, and F1-score of about 3.36% over block and keypoint-based approaches. The results observed over CNN models reveal that an increment of recall and F1-score of about 4.67 and 3.75% marginally decreased the precision of 0.75%, and the model showed improvement over CNN-based with optimization approaches of about 0.20% for precision and F1-score of about 0.72% with a slight decrease in recall. The recall is slightly reduced due to the model's memorized patterns that yield generalization of unseen data. As a result, it misses identifying true positives during the forgery detection phase.

In the future, we experiment with different thresholds to find an optimal balance between precision and recall. Also, a comprehensive dataset encompassing various forgery types, such as image splicing and copy-move forgery, will be combined to generalize the proposed model. Also, surrogate models can be adapted to approximate the objective function to reduce the computational burden.

Author contributions P.H. and P.B. Conceptualization; P.H. Methodology; P.H. validation; P.B. data curation; P.B. Software; P.H. investigation, P.H. and P.B. writing-original draft; P.H. and P.B. Supervision.

Funding This research received no external funding.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Fridrich J, Soukal D, Luk J (2003) Detection of Copy-move Forgery in Digital Images. In: Proceedings of Digital Forensic Research Workshop, Cleveland, Ohio
2. Jin G, Wan X (2017) An improved method for sift-based copy-move forgery detection using non-maximum value suppression and optimized j-linkage. Signal Process Image Commun 57:113–125. https://doi.org/10.1016/j.image.2017.05.010
3. Mahmood T, Mehmood Z, Shah M, Khan Z (2018) An efficient forensic technique for exposing region duplication forgery in digital images. Appl Intell 48(7):1791–1801. https://doi.org/10.1007/s10489-017-1038-5
4. Jaiswal AK, Srivastava R (2019) Copy-move forgery detection using shift-invariant SWT and block division mean features vol 524, pp 289–299. Springer, Cham. https://doi.org/10.1007/978-981-13-2685-1_28
5. Priyanka Singh G, Singh K (2020) An improved block based copy-move forgery detection technique. Multimedia Tools Appl 79(19–20):13011–13035. https://doi.org/10.1007/s11042-019-08354-x
6. Jaiprakash SP, Desai MB, Prakash CS, Mistry VH, Radadiya KL (2020) Low dimensional DCT and DWT feature based model for detection of image splicing and copy-move forgery. Multimedia Tools Appl 79(39–40):29977–30005. https://doi.org/10.1007/s11042-020-09415-2
7. Dua S, Singh J, Parthasarathy H (2020) Detection and localization of forgery using statistics of DCT and Fourier components.

Signal Process Image Commun 82:115778. https://doi.org/10.1016/j.image.2020.115778. (**2019**)
8. Qazi T, Ali M, Hayat K, Magnier B (2022) Seamless copy-move replication in digital images. J Imaging 8(3):1–15. https://doi.org/10.3390/jimaging8030069
9. Raju PM, Nair MS (2022) Copy-move forgery detection using binary discriminant features. J King Saud Univ Comput Inf Sci 34(2):165–178. https://doi.org/10.1016/j.jksuci.2018.11.004
10. Babu SBGT, Rao CS (2023) Efficient detection of copy-move forgery using polar complex exponential transform and gradient direction pattern. Multimedia Tools Appl 82(7):10061–10075
11. Bevinamarad P, Unki PH (2024) Digital image authentication and analysis: unmasking copy-move forgery in digital images through combined DCT and GLCM features with block matching technique. IAENG Int J Comput Sci 51(11):1672–1685
12. Amerini I, Ballan L, Caldelli R, Del Bimbo A, Serra G (2011) A SIFT-based forensic method for copy-move attack detection and transformation recovery. IEEE Trans Inf Forensics Secur 6(3 Part 2):1099–1110. https://doi.org/10.1109/TIFS.2011.2129512
13. Prakash CS, Panzade PP, Om H, Maheshkar S (2019) Detection of copy-move forgery using AKAZE and SIFT keypoint extraction. Multimedia Tools Appl 78(16):23535–23558. https://doi.org/10.1007/s11042-019-7629-x
14. Lyu Q, Luo J, Liu K, Yin X, Liu J, Lu W (2021) Copy move forgery detection based on double matching. J Vis Commun Image Represent 76:103057. https://doi.org/10.1016/j.jvcir.2021.103057
15. Niyishaka P, Bhagvati C (2020) Copy-move forgery detection using image blobs and BRISK feature. Multimedia Tools Appl 79(35–36):26045–26059. https://doi.org/10.1007/s11042-020-09225-6
16. Diwan A, Sharma R, Roy AK, Mitra SK (2021) Keypoint based comprehensive copy-move forgery detection. IET Image Process 15(6):1298–1309. https://doi.org/10.1049/ipr2.12105
17. Li J, Li X, Yang B, Sun X (2015) Segmentation-based image copy-move forgery detection scheme. IEEE Trans Inf Forensics Secur 10(3):507–518. https://doi.org/10.1109/TIFS.2014.2381872
18. Yu L, Han Q, Niu X (2016) Feature point-based copy-move forgery detection: covering the non-textured areas. Multimedia Tools Appl 75(2):1159–1176. https://doi.org/10.1007/s11042-014-2362-y
19. Lin C, Lu W, Huang X, Liu K, Sun W, Lin H (2019) Region duplication detection based on hybrid feature and evaluative clustering. Multimedia Tools Appl 78(15):20739–20763. https://doi.org/10.1007/s11042-019-7342-9
20. Agarwal S, Chand S (2018) Image forgery detection using co-occurrence-based texture operator in frequency domain. Adv Intelligent Syst Comput 518:117–122. https://doi.org/10.1007/978-981-10-3373-5_10
21. Meena KB, Tyagi V (2020) A hybrid copy-move image forgery detection technique based on Fourier-Mellin and scale invariant feature transforms. Multimedia Tools Appl 79(11):8197–8212. https://doi.org/10.1007/s11042-019-08343-0
22. Aloraini M, Sha L, Sharifzadeh M, Schonfeld D (2019) Dictionary learning and sparse coding for digital image forgery detection. IS and T International Symposium on Electronic Imaging Science and Technology 2019(5):1–7. https://doi.org/10.2352/ISSN.2470-1173.2019.5.MWSF-531
23. Bilal M, Habib HA, Mehmood Z, Saba T, Rashid M (2020) Single and multiple copy-move forgery detection and localization in digital images based on the sparsely encoded distinctive features and DBSCAN clustering. Arab J Sci Eng 45(4):2975–2992. https://doi.org/10.1007/s13369-019-04238-2
24. Elaskily MA, Elnemr HA, Sedik A, Dessouky MM, El Banby GM, Elshakankiry OA, Khalaf AAM, Aslan HK, Faragallah OS, Abd El-Samie FE (2020) A novel deep learning framework for copy-moveforgery detection in images. Multimedia Tools Appl 79(27–28):19167–19192. https://doi.org/10.1007/s11042-020-08751-7
25. Koul S, Kumar M (2022) An efficient approach for copy-move image forgery detection using convolution neural network. Content courtesy of Springer Nature, pp 11259–11277
26. Prem Kumar CD, Saravana Sundaram S (2023) Metaheuristics with optimal deep transfer learning based copy-move forgery detection technique. Intelligent Automation Soft Comput 35(1):881–899. https://doi.org/10.32604/iasc.2023.025766
27. Muniappan T, Bakiah N, Warif A, Ismail A, Atikah N, Abir M (2023) An evaluation of convolutional neural network (CNN) model for copy-move and splicing forgery detection. Int J Intelligent Syst Appl Eng 11(2):730–740
28. Sadanand VS, Janardhana SS, Purushothaman S, Hande S, Prakash R (2024) Convolutional neural network-based techniques and error level analysis for image tamper detection. Indones J Electr Eng Comput Sci 33(2):1100–1107. https://doi.org/10.11591/ijeecs.v33.i2.pp1100-1107
29. Chen B, Tan W, Coatrieux G, Member S, Zheng Y (2020) A serial image copy-move forgery localization scheme with source/target distinguishment. IEEE Trans Multimedia 23:3506–3517. https://doi.org/10.1109/TMM.2020.3026868
30. Jabeen S, Khan UG, Iqbal R, Mukherjee M, Lloret J (2021) A deep multimodal system for provenance filtering with universal forgery detection and localization. Multimedia Tools Appl 80(11):17025–17044. https://doi.org/10.1007/s11042-020-09623-w
31. Lee SI, Park JY, Eom IK (2022) CNN-based copy-move forgery detection using rotation-invariant wavelet feature. IEEE Access 10(September):106217–106229. https://doi.org/10.1109/ACCESS.2022.3212069
32. Chaitra B, Bhaskar Reddy PV (2023) An approach for copy-move image multiple forgery detection based on an optimized pre-trained deep learning model. Knowl Based Syst 269:110508. https://doi.org/10.1016/j.knosys.2023.110508
33. Diaa U (2024) A deep learning model to inspect image forgery on SURF keypoints of SLIC segmented regions. Eng Tech Appl Sci Res 14(1):12549–12555. https://doi.org/10.48084/etasr.6622
34. Khalil AH, Ghalwash AZ, Elsayed HAG, Salama GI, Ghalwash HA (2023) Enhancing digital image forgery detection using transfer learning. IEEE Access 11(August):91583–91594. https://doi.org/10.1109/ACCESS.2023.3307357
35. Jaiswal AK, Srivastava R (2022) Detection of copy-move forgery in digital image using multi-scale, multi-stage deep learning model. Neural Process Lett 54(1):75–100. https://doi.org/10.1007/s11063-021-10620-9
36. Qazi EUH, Zia T, Imran M, Faheem MH (2023) Deep learning-based digital image forgery detection using transfer learning. Intell Autom Soft Comput 38(3):225–240. https://doi.org/10.32604/iasc.2023.041181
37. Kasim Ö (2024) Deep learning-based efficient and robust image forgery detection. Multim Tools Appl 83(21):1–20. https://doi.org/10.1007/s11042-023-17946-7
38. Jaiswal AK, Srivastava R (2023) Fake region identification in an image using deep learning segmentation model. Multim Tools Appl. https://doi.org/10.1007/s11042-023-15032-6
39. Tinnathi S, Sudhavani G (2021) An efficient copy move forgery detection using adaptive watershed segmentation with AGSO and hybrid feature extraction. J Visual Commun Image Represent 74:102966. https://doi.org/10.1016/j.jvcir.2020.102966
40. Sabeena M, Abraham L (2021) Digital image forensic using deep flower pollination with adaptive Harris hawk optimization. Multim Tools Appl 80(17):26333–26355. https://doi.org/10.1007/s11042-021-10925-w

41. Ahmad M, Khursheed F (2022) A novel image tamper detection approach by blending forensic tools and optimized CNN: Sealion customized firefly algorithm. Multim Tools Appl 81(2):2577–2601. https://doi.org/10.1007/s11042-021-11529-0
42. Rao AV, Rao CS, Cheruku DR (2022) An enhanced copy-move forgery detection using machine learning based hybrid optimization model. Multim Tools Appl 81(18):25383–25403. https://doi.org/10.1007/s11042-022-11977-2
43. Ganeshan R, Muppidi S, Thirupurasundari DR, Kumar BS (2022) Autoregressive-Elephant Herding Optimization based Generative Adversarial Network for copy-move forgery detection with Interval type-2 fuzzy clustering. Signal Process Image Commun 108:116756. https://doi.org/10.1016/j.image.2022.116756
44. Agarwal R, Verma OP (2022) Robust copy-move forgery detection using modified superpixel based FCM clustering with emperor penguin optimization and block feature matching. Evol Syst 13(1):27–41. https://doi.org/10.1007/s12530-021-09367-4
45. Archana MR, Biradar DN, Dayanand J (2023) Image forgery detection in forensic science using optimization based deep learning models. Multim Tools Appl 83(15):45185–45206. https://doi.org/10.1007/s11042-023-17316-3
46. Gupta R, Singh P, Alam T, Agarwal S (2023) A deep neural network with hybrid spotted hyena optimizer and grasshopper optimization algorithm for copy move forgery detection. Multim Tools Appl 82(16):24547–24572. https://doi.org/10.1007/s11042-022-14163-6
47. Bevinamarad P, Unki P, Nidagundi P (2024) Copy-move Forgery detection and localization framework for images using stationary wavelet transform and hybrid dilated adaptive VGG16 with optimization strategy. Int J Image Graphic Signal Process 16(1):38–60. https://doi.org/10.5815/ijigsp.2024.01.04
48. Bhowal A, Neogy S, Naskar R (2024) Deep learning-based forgery detection and localization for compressed images using a hybrid optimization model. Multim Syst. https://doi.org/10.1007/s00530-024-01336-6
49. Liao L, Lei Y (2024) Image content forgery detection model combining PSO and SVM in electronic data forensics. Informatica 48(8):151–164. https://doi.org/10.31449/inf.v48i8.5897
50. Nirmala Priya G, Suresh Kumar K, Suganthi N, Muppidi S (2024) Squirrel Henry Gas Solubility Optimization driven Deep Maxout Network with multi-texture feature descriptors for digital image forgery detection. Concurr Comput Pract Exp 36(8):7965. https://doi.org/10.1002/cpe.7965
51. Manasa M, Chaudhari SS (2024) Spider monkey optimization-based image data forgery detection over vehicular cloud computing. In: Bhattacharyya S, Banerjee J.S, Köppen M. (eds) Human-centric smart computing, pp 333–343. Springer, Singapore
52. H Gedara TM, Loia V, Tomasiello S (2024) Detecting fake images using neuro-fuzzy inference systems: a brief comparative analysis. In: IEEE International Conference on Fuzzy Systems, pp 1–7. https://doi.org/10.1109/FUZZ-IEEE60900.2024.10612033
53. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, vol 4, pp 1942–19484. https://doi.org/10.1109/ICNN.1995.488968
54. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey Wolf optimizer. Advanc Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
55. Tralic D, Zupancic I, Grgic S, Grgic M (2013) CoMoFoD - New database for copy-move forgery detection. In: Proceedings Elmar - International Symposium Electronics in Marine (September), pp 49–54
56. Ardizzone E, Bruno A, Mazzola G (2015) Copy-move forgery detection by matching triangles of keypoints. IEEE Trans Inf Forensics Secur 10(10):2084–2094. https://doi.org/10.1109/TIFS.2015.2445742
57. Amerini I, Ballan L, Caldelli R, Del Bimbo A, Del Tongo L, Serra G (2013) Copy-move forgery detection and localization by means of robust clustering with J-Linkage. Signal Process Image Commun 28(6):659–669. https://doi.org/10.1016/j.image.2013.03.006
58. Raju PM, Nair MS (2022) Copy-move forgery detection using binary discriminant features. J King Saud Univ - Comput Inform Sci 34(2):165–178. https://doi.org/10.1016/j.jksuci.2018.11.004
59. Meena KB, Tyagi V (2020) A copy-move image forgery detection technique based on tetrolet transform. J Inform Security Appl 52:102481. https://doi.org/10.1016/j.jisa.2020.102481
60. Wang H, Wang H (2018) Perceptual hashing-based image copy-move Forgery detection. Security Commun Netw 2018(3):1–11. https://doi.org/10.1155/2018/6853696
61. Pun CM, Yuan XC, Bi XL (2015) Image forgery detection using adaptive oversegmentation and feature point matching. IEEE Trans Inf Forensics Secur 10(8):1705–1716. https://doi.org/10.1109/TIFS.2015.2423261