



## Copy-Move Forgery Detection and Classification Using Improved Butterfly Optimization Algorithm-based Convolutional Neural Network

Shashikala S<sup>1\*</sup> Ravikumar G K<sup>2</sup>

<sup>1</sup>Department of Computer Science and Engineering, BGS Institute of Technology, Visvesvaraya Technological University, Belagavi:590018, Karnataka, India

<sup>2</sup>Department of Computer Science, BGS College of Engineering and Technology, Visvesvaraya Technological University, Belagavi-590018, Karnataka, India

\* Corresponding author's Email: shashiresearch.2021@gmail.com

---

**Abstract:** Copy-move image forgery involves digitally modifying an image by copying and pasting content, hiding foreground objects, or highlighting them through duplication. However, as digital image forgery can become an extremely unsafe and challenging technique to classify images effectively, therefore understanding the detection and classification of real and forgery images is essential. In this research, an improved butterfly optimization algorithm-based convolutional neural network (IBOA-CNN) is proposed for copy-move forgery detection (CMFD), enhancing accuracy and convergence speed by expanding the iteration memory. This proposed approach is used to detect and classify images as original or fake accurately and effectively using deep learning (DL). Initially, the image is obtained by the MICC-F220, MICC-F600, MICC-F2000, and CASIA 2.0 datasets and then image pre-processing is performed by converting Red Green Blue (RGB) into a grayscale image. The Local Binary Pattern (LBP), Wavelet Features (DWT), and ResNet-50 are utilized to extract the features from the images, IBOA is used for feature selection and finally, CNN is employed for the classification to classify CMFD as an original or fake image. Existing methods such as Stacked Sparse Denoising Autoencoder- Spotted Hyena Optimizer- Grasshopper Optimization Algorithm (SSDAE-SHO-GOA), Deep CNN using ResNet-101, and CNN are compared with the IBOA-based CNN approach using MICCF-2000 dataset. The proposed IBOA-based CNN achieves a better accuracy of 97.59%, 99.20%, 99.83%, and 98.92% for MICC-F220, MICC-F600, MICC-F2000, and CASIA 2.0 datasets compared with the existing methods like SSDAE-SHO-GOA, Deep CNN using ResNet-101, and CNN.

**Keywords:** Convolutional neural network, Copy-move forgery, Classification, Detection, Deep learning.

---

### 1. Introduction

Forgery detection of a digital image is a significant procedure in digital life due to images are easily modified. Digital images are currently used on the print, internet, and in digital media [1]. Digital images are crucial data that are utilized in a variety of applications, including as evidence in court, forensics, social networks, computer-aided medical diagnosis systems, and the military. Due to the importance of their contexts, it is vital to confirm their authenticity and keep them tamper-proof. Many computer tools allow users and standard people to forge digital images, making it difficult for the eye to recognize

false images [2]. The basic technologies employed to detect and localize forgeries in digital images include hash codes, cryptography, digital signatures, watermarking, and image integrity authentication [3]. Active and passive are the two different kinds of image authentication methods [4, 5]. Active methods that depend on the previous information of the images include digital signatures and digital watermarks. Since previous information about the images may not be available in certain situations, passive approaches can be used to determine the validity of images [6]. Different passive image forgeries are there, including image splicing, copy-move, resampling, etc [7, 8].

CMF is the common popular kind of digital image forgery detected by passive methods [9]. CMF

imaging which involves copying an element of the image and pasting it over the original image used to be considered a special type of forgery [10]. As a result, image forensics combined with CMF identification made it extremely important in the network-based community [11]. If a modified image is used in a criminal investigation without the use of a qualified forensic tool, prosecutors may be misled. As a result, a strong image forensics tool for detecting and localizing copied movement is required [12]. Because of the uniform characteristics region of the source and target, image copy-move forgery technology produces a good visual effect and a believable fake result with basic manipulations such as noise addition, JPEG compression, scaling, rotating, and blurring [13, 14]. Therefore, forged image identification is explored to provide efficient solutions utilizing deep learning algorithms [15]. However, digital image forgery can become an extremely unsafe and challenging technique to classify images effectively. In this research, an IBOA-CNN is proposed for CMFD, enhancing accuracy and convergence speed by expanding the iteration memory. This proposed approach is used to detect and classify images as original or fake accurately and effectively using DL

The primary contribution of this research is as follows:

- An IBOA is presented to increase the performance of BOA's outcome accuracy and global optimization by integrating fractional order and chaotic mapping with the classic BOA.
- The chaotic mapping is employed to increase starting population diversity and hence enhance the BOA's capability of global search. The fractional derivative is used to increase the capacity of memory and BOA convergence and improve the searchability and memorability of iterative processes, with the goal of preventing the BOA from entering a local optimum.
- The proposed IBOA-CNN approach is evaluated using MICC-F220, MICC-F600, MICC-F2000, and CASIA 2.0 benchmark datasets. The efficacy of IBOA-based CNN is analyzed by using performance evaluation of precision, accuracy, f1-score, and recall.

The rest of the paper is organized in the following manner: Section 2 presents the literature survey. The block diagram of the proposed method is discussed in section 3. Results are illustrated in section 4. Section 5 discussed the conclusion of this paper.

## 2. Literature survey

CMFD is used by researchers to detect and classify images as authentic or fake. Some of the notable research in determining the CMFD was reviewed in this section.

Ruchi Gupta [16] implemented a DL-based approach on the stacked sparse denoising autoencoder (SSDAE) model to detect and classify the images as fake or legitimate. The hidden layers bias and input weight of the SSDAE approach were enhanced by employing the spotted hyena optimizer (SHO) and grasshopper optimization algorithm (GOA). The hybrid model of SSDAE-GOA-SHO has been evaluated by using four datasets named MICC-F220, CASIA 2.0, MICC-F2000, and MICC-F600. The SSDAE method was used to solve the statistical analysis and runtime analysis by evaluating the performance. However, the computer complexity was decreased by the large block size but duplicated images were unrecognized in the SSDAE-SHO-GOA approach.

Sharma Vaishali & Singh Neetu [17] presented a deep CNN (DCNN) utilizing ResNet-101 deep layers to address the issue of disappearing and exploding gradients. The ResNet-101 approach was tuned by employing a cyclical learning rate (CLR) hyperparameter to achieve the best performance. The CLR was an efficient technique that enhanced model accuracy with a minimal rise in the number of epochs. However, training the approach with a small sample size causes overfitting, which prevents the technique from extracting features while the approach was in training.

Saboor Koul [18] introduced a CNN for automatic CMFD in images by using the MICC-F2000 dataset. CNN was used to automatically learn features and provide them to a classifier. The feature extractor was a component of CNN's training phase, which includes an activation function, convolutional layers, fully-connected layers, pooling layers, and a classification layer. CNN has achieved greater performance with minimal computational power. However, training CNN needs a high amount of labeled data which was challenging to acquire for various and realistic forgery scenarios.

Navneet Kaur [19] implemented a deep-learning CMF detection system that employs contrast-limited adaptive histogram equalization (CLAHE) and a CNN to classify images as original or fake. Initially, image pre-processing was accomplished by the use of image scaling and the CLAHE approach. The CNN model extracts visual features and generates feature maps from them. Data augmentation approaches like vertical and horizontal flips were used to reduce bias,

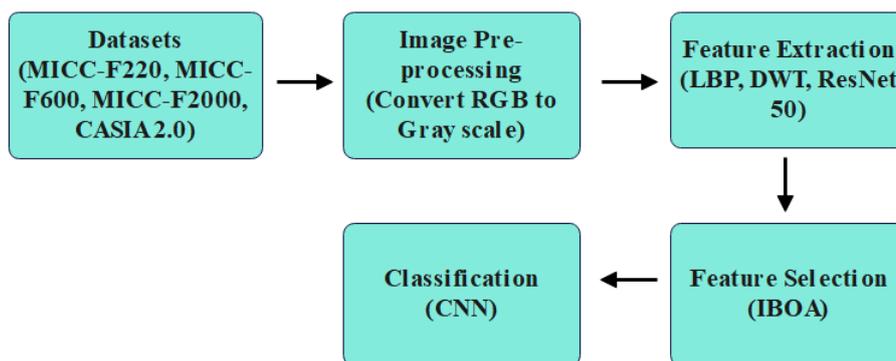


Figure. 1 Block diagram for the proposed method

improve generality, and compensate for a lack of training data. However, the implemented approach fails to classify the digital images accurately and effectively.

Nidhi Goel [20] presented a DL technique for image CMFD by using the MICC-F2000 dataset and this approach applied a dual-branch CNN to classify the images as original or fake. By using distinct kernel sizes in each branch, the dual-branch CNN neural network extracts multi-scale information. The presented approach was lightweight and achieved good accuracy in prediction and the computation time, and performance scores were efficient compared to an existing model. However, the Dual branch CNN approach solves only the detection issue and therefore comparison scope was limited to the classification phase.

Baraa Tareq Hammad [21] introduced a Conventional CMFD utilizing the AlexNet DL approach to extract image features, while the ReliefF feature selection was employed to obtain efficient features. Then, the logistic classifier was fed specific features to identify whether images were authentic or fake. The introduced approach was evaluated on the benchmark datasets MICC-F2000 and MICC-F600. The logistic classifying of the features obtained from the AlexNet approach was effective and simple. However, deep learning like AlexNet struggled to effectively exploit spatial correlations between objects and information in images, which limits their ability to detect complicated geometric transformations.

These existing approaches have different applications and also suffers from limitations. As a result, by introducing the IBOA-based CNN approach, these limitations can be overcome in this research.

### 3. Proposed methodology

In this research, an IBOA-based CNN approach is proposed for CMFD using deep learning. Initially, the proposed HCFS-BOA approach is evaluated using the MICC-F220, MICC-F600, MICC-F2000, and CASIA 2.0 benchmark dataset for detecting image forgery. Next, the image pre-processing is performed by converting the color space of the image from RGB to grayscale. The LBP, DWT, and ResNet 50 approaches are used for feature extraction and the extracted features were selected by using IBOA for CMFD. Finally, the CNN classification is performed to classify the images as authentic or fake. Fig. 1 shows the block diagram of the proposed method.

#### 3.1 Datasets

In this research, the proposed HCFS-BOA approach is evaluated using the MICCF-220, MICCF-600, MICCF-2000, and CASIA 2.0 benchmark datasets for detecting image forgery.

##### 3.1.1. MICCF-220 dataset

MICCF-220 dataset includes 220 images, 110 of that are operated and 110 of which are authentic. The ranges of resolution images from 722 x 480 to 800 x 600 pixels and fake patch obtains an 1.2% average for the whole image.

##### 3.1.2. MICCF-600 dataset

MICCF-600 dataset includes 600 images with a resolution of about 3888 x 2592 pixels. It has 152 manipulated images and 448 authentic images.

##### 3.1.3. MICC-F2000 dataset

MICC-F2000 dataset [22] includes 2000 images, 700 of which have been forged and 1300 are original. It is made up of 600 images with a resolution of approximately 3888 x 2592 pixels. It includes 152 manipulated images and 448 real images. After

collecting the dataset, data are pre-processed by using the image pre-processing.

### 3.1.5. CASIA 2.0 dataset

CASIA 2.0 dataset includes 12,614 images, certain of that are uncompressed TIFF and BMP files, while others are the files of JPEG with Q varying factors ranging from 320 x 240 to 800 x 600 pixels. It includes 5123 altered images and 7491 genuine images.

The images from the aforementioned datasets are pre-processed by converting the image from RGB to Greyscale.

## 3.2 Image pre-processing

After the collection of images, pre-processing is performed by converting the color space of the image from RGB to grayscale. Initially, the input image is pre-processed since color features require the original true color image, but LBP texture features require a grayscale version of the original image. In this case, the original RGB color image will be converted into a grayscale image by using Eq. (1)

$$Grayscale = 0.289 \times R + 0.587 \times G + 0.114 \times B \quad (1)$$

The acquired image is then passed as input to the feature extraction.

## 3.3 Feature extraction

After pre-processing the acquired image, the LBP, DWT, and ResNet 50 approaches are used for feature extraction.

### 3.3.1. Local binary pattern (LBP)

LBP is an effective local descriptor with invariant features for rotation and conversion of grayscale. The LBP [23] is employed to generate a vector of features that is partially invariant to translation, rotation, and scaling. The LBP code is calculated by analyzing the pixel to the nearby pixel cell. The suggested method employs a uniform LBP family to extract fifty-nine LBP features (F84-F142). The pre-processed image is segmented into 3 x 3 cells with 8 neighbors and a radius of one in this method. By using Eq. (1), the LBP is computed for each pixel in the cell which is expressed in Eq. (2)

$$LBP_{(p,r)}^u$$

$$= \sum_{p=0}^{p-1} s(g_p - g_c)2^p, s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2)$$

Where  $g_c$  – pixel centre

$g_p$  – circle neighbors with  $r$  radius

$(p, r)$  – subscript denotes  $p$  neighbor with  $r$  radius

$u$  – uniform local pattern

The histogram is then produced for each cell using Eq. (3)

$$H_k = \sum_{i,j} I[f(i, j) = k], i = 0, 1, 2 \dots n - 1 \quad (3)$$

To obtain a coherent description, the histogram is now normalized by using Eq. (4)

$$N_k = \frac{H_k}{\sum_{i=0}^{n-1} H_i} \quad (4)$$

### 3.3.2. Wavelet features (DWT)

It is a frequency-based texture feature grouping approach and the energy of a signal is captured by a specific wavelet coefficient. For the timescale representation of the digital signal in DWT, a digital filtering approach is employed. The images have been divided into small wavelets that correspond to the sub-frequency bands low-high (LH), low-low (LL), high-low (HL), and high-high (HH) [24]. The DWT's 32 features (F52-F83) are considered in this research. Initially, the pre-processed image was scaled to 256 x 256 pixels, and then the approximate coefficient was split up into four sub-bands until it reached the fourth-level approximation coefficient. For the 16 x 16 size, the coefficient means and standard deviations of every row are derived for 32 features. From the pre-processed data, the mean and standard deviation are expressed by using the Eqs. (5) and (6)

$$Mean = \frac{1}{n} \sum_{j=1}^n |LL^4(i, j)| \quad (5)$$

$$SD = \sqrt{\frac{1}{(n-1)} \sum_{j=1}^n |LL^4(i, j) - Mean(LL^4)|^2} \quad (6)$$

### 3.3.3. ResNet 50

The purpose of utilizing ResNet-50 in feature extraction for CMFD is to improve forgery detection accuracy and effectiveness by using DL approach's ability to learn and represent complex image data. The Residual network includes five stages. The first stage is the input stage, which has only one convolutional layer with batch normalization and

generates the initial feature map using the activation function [25]. Identity and convolutional blocks are used in the remaining states. A convolutional layer with activation functions and further batch normalization is included in these two blocks. The input layer has an additional bridge to the output layer to enhance the residuals of the convolutional blocks. The residual block on ResNet 50 is expressed in Eq. (7)

$$y = F(x, W + x) \tag{7}$$

Where  $x$  and  $y$  denotes input and output layer respectively, Function  $F$  refers to the residual map. When the input and output values on the ResNet 50 are identical, the residual block has been processed. After the feature extraction, the image features are selected using an optimization process.

### 3.4 Feature selection

After feature selection, the IBOA approach is implemented for feature selection in CMFD. Using an IBOA for feature selection in CMFD aids in optimizing the detection process, making it more efficient, accurate, and adaptive to various forgery conditions. A suitable set of features is chosen to provide high-quality detection results, and BOA can help with this process. The BOA is a nature-inspired optimization approach inspired by butterfly collective behavior. BOA is created for a global search of the solution space, which means it may examine a large range of potential solutions. As a result, it is well suited for locating global optima in complicated and multi-modal optimization issues. Every butterfly in the ecosystem is regarded as a self-sufficient search individual in the BOA, generating a specific strength of fragrance. As a result, when an individual butterfly goes from one area to another with the searching process, the motion fitness changes, and fragrance distributes throughout the process of moving. Every butterfly detects different butterflies' fragrances however, the fragrance diminishes over time and the butterfly goes to the area with the highest fragrance. This is the key difference among other metaheuristics and the butterfly algorithm.

The fragrance concentration  $f$  in the BOA is calculated using three parameters: the modality of the sensor  $c$ , power index  $a$ , and the stimulus intensity  $I$ . The butterfly's perceptual morphology is its fragrance, that is an algorithm initialization constant and is typically utilized as a parameter of optimization. The fitness function is used to calculate stimulus intensity  $I$ . Furthermore, the power index

parameter  $a$  is a constant that varies between 0 and 1. The BOA is represented in its basic and simple form which is expressed in Eq. (8)

$$f = cI^a \tag{8}$$

BOA's primary stages are characterized as follows:

- a) The initialization step, where the objective function is defined to address the drawbacks of standard BOA, such as low convergence and the ease of falling into a local optimum in multi-objective optimizations. Therefore, chaotic mapping and fractional differentiation were employed using the time-trajectory planning's time-jerk criterion. The trajectory is then improved by adapting the control points of the non-uniform rational B-spline (NURBS). To accomplish this, the modality of a sensor, switching probability, initial population, and power index must be defined, and the associated fitness value must be calculated.
- b) Iteration step, where butterflies' positions in the solution space are reallocated, and thus the fragrance and fitness value for every butterfly are recomputed. At this phase, either a local or global search is required. The butterfly passes to the  $g^*$  butterfly with the highest fragrance value in the global search. This procedure which is expressed in Eq. (9)

$$x_i^{t+1} = x_i^t + (r^2 \times g^* - x_i^t) \times f_i \tag{9}$$

Where  $x_i^{t+1}$  and  $x_i^t$  are the outcomes for the  $i^{th}$  iterations  $t + 1$  and  $t$ ,  $r$  is a random value between  $[0, 1]$ . The JADE-GL tuned butterfly approach's global search is employed to execute a greater global search. This approach is expressed in Eq. (10)

$$x_i^{t+1} = x_i^t + (r(g_i^* - x_i^t) + (1 - r)(x_{r1}^t - x_{r2}^t)) \times f_i \tag{10}$$

Where  $x_i^{t+1}$  indicates the outcome related to the  $i^{th}$  butterfly in the  $i^{th}$  iteration,  $g$  represents ideal outcomes for the present iteration,  $f_i$  indicates the butterfly's fragrance, and random number  $r$  represents the range  $(0,1)$ .

A butterfly undertakes a random walk if it cannot detect the fragrance presented by other butterflies. At this time, the BOA is stated in terms of local search which is expressed in Eq. (11)

$$x_i^{t+1} = x_i^t + (r^2 \times x_i^t - x_k^t) \times f_i \tag{11}$$

Where  $x_i^{t+1}$  and  $x_k^t$  represent the outcomes for the  $j^{th}$  and  $k^{th}$  butterflies in the iteration  $i^{th}$  correspondingly.

c) The termination step, is where the most effective solution is determined. This scenario happens when the produced outcome matches the criterion or when the number of iterations enters a particular threshold.

### 3.4.1. Improved BOA (IBOA)

An IBOA [26] is presented to increase the performance of BOA’s outcome accuracy and global optimization by integrating fractional order and chaotic mapping with the classic BOA.

#### 3.4.1.1. Chaotic mapping

It is a function of multivariate nonlinear that is utilized for nonlinear probabilistic forecasting of time series information to increase starting population diversity and hence enhance the BOA’s capability of global search. To maximize the initial population in the BOA approach, circle chaotic mapping is employed which is expressed in Eq. (12).

$$x_{k+1} = x_k + b - (p - 2\pi) \sin(2\pi x_k) \text{ mod } (1) \tag{12}$$

Where  $p = 0.5$  and  $b = 0.2$  which are the control parameter. Eq. (11) produces a chaotic mapping distribution with a range of (0,1).

#### 3.4.1.2. Fractional derivative

It is frequently employed to increase the capacity of memory and BOA convergence, improve the searchability and memorability of iterative processes, and enhance the outcome accuracy, with the goal of avoiding the BOA from entering a local optimum. In this way, the present research optimizes BOA using fractional differentiation of Grunwald-Letnikov. The gamma function is employed to convert the integral of integer order (G-L) to fractional order. The fractional differential of  $\alpha$  - order is denoted in Eq. (13).

$$D_t^\alpha \cdot \eta(t) = \lim_{h \rightarrow 0} h^{-\alpha} \sum_{m=0}^{\lfloor \frac{t-\alpha}{h} \rfloor} (-1)^m \frac{\Gamma(\alpha+1)}{\Gamma(m+1)\Gamma(\alpha-m+1)} \eta(t - mh) \tag{13}$$

Where  $D$  and  $\alpha$  represent fractional derivative and fractional derivative order accordingly. When  $\alpha > 0, G - L$  indicates differentiation, whereas  $\alpha < 0$  represents integration. When associated with the

BOA, the derivative fraction which is expressed in Eq. (14).

$$D_t^\alpha \cdot x(t) = \lim_{h \rightarrow 0} h^{-\alpha} \sum_{m=0}^{\lfloor \frac{t-\alpha}{h} \rfloor} (-1)^m \frac{\Gamma(\alpha+1)}{\Gamma(m+1)\Gamma(\alpha-m+1)} x(t - mh) \tag{14}$$

To integrate with the discrete time period in trajectory planning, Eq. (14) is simplified in the below form which is expressed on Eq. (15).

$$D_t^\nu \cdot x(t) = T^{-\nu} \sum_{m=0}^{\zeta} (-1)^m \frac{\Gamma(\nu+1)}{\Gamma(m+1)\Gamma(\nu-m+1)} x(t - mT) \tag{15}$$

Where  $\zeta$  and  $T$  represent the truncation order and sampling period respectively. From Eq. (11) the global search is written in below form which is expressed in Eq. (16).

$$x_i^{t+1} - x_i^t = (r(g_i^* - x_i^t) + (1 - r)(x_{r1}^t - x_{r2}^t)) \times f_i \tag{16}$$

Consider  $\nu = T = 1$ , Eq. (15) is re-expressed in Eq. (17).

$$D_t^\nu \cdot [x_i^{t+1}] = (r(g_i^* - x_i^t) + (1 - r)(x_{r1}^t - x_{r2}^t)) \times f_i \tag{17}$$

The first four terms of Eq. (17) are expressed in Eq. (18).

$$x_i^{t+1} = vx_i^t + \frac{1}{2}v(1 - v)x_i^{t-1} - \frac{1}{6}v(1 - v)(2 - v)x_i^{t-2} + \frac{1}{24}v(1 - v)(2 - v)(3 - v)x_i^{t-3} + (r(g_i^* - x_i^t) + (1 - r)(x_{r1}^t - x_{r2}^t)) \times f_i \tag{18}$$

Furthermore, a solution is chosen at random as an individual in the present situation. The solution and its inverse outcomes are then used in a secondary local search. A local search is presented using the quasi-Newton approach. It is expressed in Eq. (19) to calculate the new position

$$x_{id}^{t+1} = lb + (ub - lb) \times x_{k+1} \tag{19}$$

Where  $ub$  and  $lb$  are the NURBS curve control point adaptation of upper and lower limits, and  $x_{id}^{t+1}$  is the outcome for the  $id^{th}$  butterfly in the iteration  $t + 1$ . Thus, the acquired data is passed into the feature selection phase which is sufficient for the classification of CMFD.

### 3.5 Classification

The selected features are classified using the CNN model, which produces enormous results in domains such as natural language processing (NLP) and image processing. In contrast to multi-layer perceptron (MLP), CNN reduces the number of neurons and parameters, resulting in rapid adaptability and minimal complexity. The CNN model offers an extensive number of clinical classification applications. CNN models are a subset of feed-forward neural network (FFNN) and deep learning models. The convolution operations convention is constant, which implies the filter is independent in function that reduces the parameter amount. Pooling, convolution, and fully connected layers are the three types of layers used in the CNN method. These layers are required for performing feature extraction, dimensionality reduction, and classification. The filter is slid on the computers through the forward pass of convolution operation, and the input capacity of activation map that assesses the point-wise result of every score is added to obtain the activation. The sliding filter is employed by linear and convolution operator, it is stated as quick distribution of dot product. Consider  $w$  is the kernel function,  $x$  is the input,  $(x \times w)(a)$  on time  $t$  is formulated as Eq. (20).

$$(x \times w)(a) = \int x(t)w(a - t)da \quad (20)$$

Where,  $a$  is  $R^n$  for each  $n \geq 1$ . The parameter  $t$  is the discrete which is presented in Eq. (21).

$$(x \times w)(a) = \sum_a x(t)w(t - a) \quad (21)$$

The 2D image  $I$  as input,  $K$  is a 2D kernel and convolution is formulated as Eq. (22).

$$(I \times K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (22)$$

To improve the non-linearity, two different activation functions are utilized like ReLU and softmax. The ReLU is represented as Eq. (23).

$$ReLU(x) = (0, x) \quad x \in R \quad (23)$$

The gradient  $ReLU(x) = 1$  for  $x > 0$  and  $ReLU - (x) = 0$  for  $x < 0$ . The ReLU convergence ability is better than the sigmoid non-linearities. The next layer is softmax, it is preferable when the result required to included two or more classes which is mathematically formulated as Eq. (24).

$$softmax(x_i) = \frac{exp(x_i)}{\sum_j exp(x_j)} \quad (24)$$

The pooling layers are applied to result a statistic of input and rescale the structure of output without losing essential information's. There are various types of pooling layer, this paper utilized highest pooling which individually produce large values in rectangular neighborhood of individual points  $(i, j)$  in 2D information for every input feature correspondingly. The fully connected (FC) layer is a last layer with  $m$  and  $n$  output and input are illustrated. The parameter of output layer is stated as weight matrix  $\in M_{m,n}$ . Where  $m$  and  $n$  is an rows and columns and the bias vector  $b \in R^m$ . Consider as an input vector  $x \in R^n$ , the fully connected layer output with an activation function  $f$  is formulated as Eq. (25).

$$FC(x) := f(Wx + b) \in R^m \quad (25)$$

Where the  $Wx$  is the matrix product while function  $f$  is employed as component. This fully connected layer is applied for classification difficulties. The FC layer of CNN is commonly involved at topmost level. The CNN production is compressed and displayed as a single vector. Thus, the classification phase is performed using the most significant and pertinent features.

## 4. Experimental results

In this research, the IBOA based CNN is simulated by using a Python environment with the system configuration of RAM:16GB, Processor: Intel core i7 and Operating System: Windows 10. The parameters like accuracy, precision, recall, and f1-score are utilized to estimating the performance of model. The mathematical representation of these parameters is shown in Eqs. (26), (27), (28), and (29)

- Accuracy – Accuracy is the proportion of accurate predictions to all input samples and it is calculated using the below equation

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (26)$$

- Precision - The precision measures the percentage of actual data records versus expected data records. The performance of the classification model is greater if the precision is higher.

$$Precision = \frac{TP}{TP+FP} \quad (27)$$

Table 1. Performance of feature selection using MICC-F2000

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
GOA	88.20	84.61	87.11	88.23
WOA	90.12	83.40	86.63	89.17
SCSO	91.57	89.77	90.57	90.25
BOA	93.17	91.82	91.86	92.37
<b>IBOA</b>	<b>94.97</b>	<b>93.13</b>	<b>92.72</b>	<b>94.92</b>

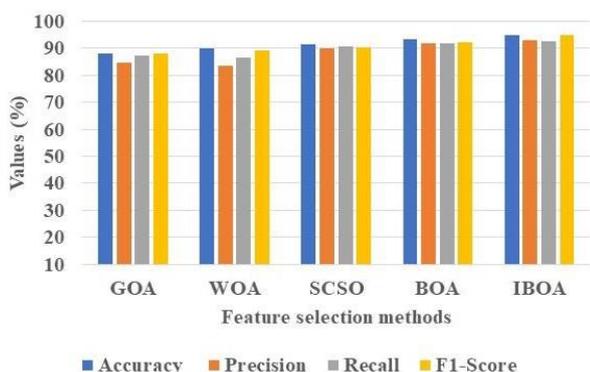


Figure. 2 Graphical representation of feature selection performances

- Recall – Recall is calculated as the sum of the true positives and the positive class images.

$$Recall = \frac{TP}{TP+FN} \tag{28}$$

- F1-Score – It is also known as the harmonic mean, which seeks a balance between recall and precision.

$$F1 - Score = \frac{2TP}{2TP+FP+FN} \tag{29}$$

#### 4.1 Quantitative and qualitative analysis

This section shows the quantitative and qualitative analysis of proposed IBOA based CNN model with respect to precision, accuracy, f1-score, and recall are presented in Table 1, 2 and 3. Table 1 illustrates the performance of feature selection using MICC-F2000 dataset. The performance of Whale Optimization Algorithm (WOA), GOA, Sand Cat Swarm Optimization (SCSO), Butterfly Optimization Algorithm (BOA), and IBOA are measured and matched with the proposed IBOA-based CNN. Fig 2 represents that the graphical representation of feature selection methods. The obtained result shows that the proposed IBOA algorithm attains accuracy of 94.97% precision of 93.13%, recall of 92.72%, and f1-score of 94.92% which is better when compared to existing optimization algorithms.

Table 2. Performance of classification with default features using MICC-F2000 dataset

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ANN	86.23	87.02	86.85	85.37
KNN	88.67	86.61	87.36	89.95
DT	90.78	87.25	89.81	90.23
SVM	91.32	90.78	91.49	91.90
<b>CNN</b>	<b>93.96</b>	<b>91.09</b>	<b>92.83</b>	<b>92.74</b>

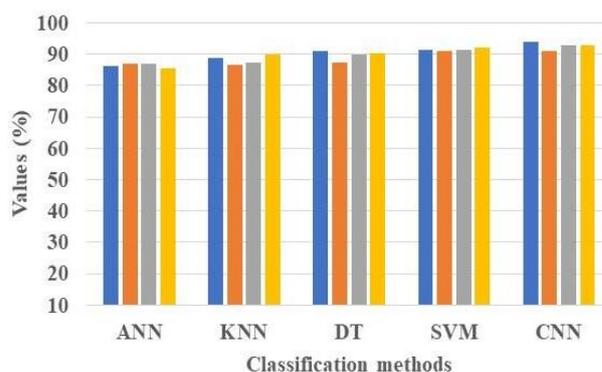


Figure. 3 Graphical representation of classification performances

Table 2 illustrates the performance of classification with default features using MICC-F2000 dataset. The performance of K-Nearest Neighbor (KNN), Artificial Neural Network (ANN), Decision Tree (DT), and Support Vector Machine (SVM) are measured and matched with the proposed IBOA-based CNN. Fig. 3 represents that the graphical representation of classification performances. The obtained result shows that the proposed CNN algorithm attains accuracy of 93.96% precision of 91.09%, recall of 92.83%, and f1-score of 92.74% which is better when compared to existing optimization algorithms.

Table 3 illustrates the performance of feature selection-based classification after feature selection using MICC-F2000 dataset. The performance of ANN, KNN, DT, SVM, and CNN are measured and matched with the proposed IBOA-based CNN. Fig. 4 illustrates that the graphical representation of optimized features for different classifier performances. The obtained result shows that the CNN algorithm attains accuracy of 99.83% precision of 99.32%, recall of 98.93%, and f1-score of 97.99% which is better when compared to existing optimization algorithms.

Table 4. Comparative analysis with existing methods

Methods	Datasets	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
SSDAE-GOA-SHO [16]	MICC-F220	97.45	98.75	98.25	98.55
	MICC-F600	98.92	88.45	85.21	91.41
	MICC-F2000	99.12	99.25	91.14	85.32
	CASIA 2.0	98.02	96.03	97.74	97.48
DCNN using ResNet-101 [17]	MICC-F220	96.09	N/A	N/A	N/A
	MICC-F600	97.63	N/A	N/A	N/A
	MICC-F2000	96.87	N/A	N/A	N/A
CNN [18]	MICC-F2000	97.52	97	96	97
Proposed IBOA-CNN	MICC-F220	97.59	98.92	98.71	98.97
	MICC-F600	99.20	90.26	86.35	91.66
	MICC-F2000	99.83	99.32	98.93	97.99
	CASIA 2.0	98.92	97.74	98.12	98.31

Table 3. Performance of classification with optimized feature for different classifier

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ANN	94.75	93.29	93.72	94.96
KNN	95.18	94.16	93.98	93.87
DT	96.26	94.75	95.23	95.68
SVM	97.64	96.40	97.75	97.06
CNN	<b>99.83</b>	<b>99.32</b>	<b>98.93</b>	<b>97.99</b>

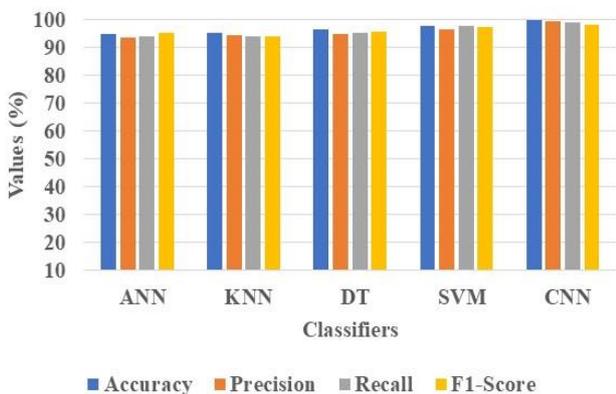


Figure. 4 Graphical representation of optimized features for different classifier performances

### 4.2 Comparative analysis

This section provides the comparative analysis of proposed IBOA-based CNN model with evaluation metrics like precision, accuracy, f1-score, and recall as shown in Table 4. The existing methods such as SSDAE-GOA-SHO [16], DCNN using ResNet-101 [17], and CNN [18] are employed to assess IBOA based CNN performance. The results obtained shows that proposed IBOA-CNN attains better performance compared to the existing methods. The proposed approach achieves better accuracy of 97.59%, 99.20%, 99.83%, and 98.92% for MICC-F220, MICC-F600, MICC-F2000, and CASIA 2.0 datasets respectively.

### 4.3 Discussion

This section discusses the advantage of proposed method and the limitations of existing methods. The existing approaches has some limitation such as SSDAE-GOA-SHO [16] in that the computer complexity was decreased by the large block size but duplicated images were unrecognized in the model. DCNN using ResNet-101 [17] model training the approach with a small sample size causes overfitting, which prevents the technique from extracting features while the approach was in training. The CNN [18] needs a high amount of labeled data which was challenging to acquire for various and realistic forgery scenarios. The proposed IBOA based CNN model overcome the existing model limitations. The IBOA optimizes the feature selection process by iteratively searching for the most informative subset of features. IBOA often outperforms traditional optimization techniques in terms of convergence speed. This means that it can identify effective solutions to optimization problem faster, which is useful for real-time applications when computational time is limited. By leveraging the exploitation and exploration of IBOA, the technique efficiently explores the feature space, selecting relevant features that contribute to accurate forgery classification. This enhances the performance of CNN by focusing on the most discriminative features. By combining IBOA with CNN for feature selection in forgery classification, the proposed model obtains better result in terms of 97.59%, 99.20%, 99.83%, and 98.92% accuracy for MICC-F220, MICC-F600, MICC-F2000, and CASIA 2.0 datasets respectively.

### 5. Conclusion

In this research, the IBOA-based CNN approach is proposed for CMFD to detect and classify images

as original or fake accurately and effectively using deep learning. The proposed method mainly comprises five stages: dataset, image pre-processing, feature extraction, feature selection, and classification. First, the image is obtained from MICC-F220, MICC-F600, MICC-F2000, and CASIA 2.0 datasets and image pre-processing is performed by converting RGB into grayscale which makes forgery detection more effective. The LBP, DWT, and ResNet-50 are utilized to extract the features from the images and IBOA is used for feature selection to increase the BOA's outcome accuracy and performance of global optimization by integrating fractional order and chaotic mapping with the classic BOA. Finally, the CNN classification to classify CMFD as original or fake accurately and effectively. The proposed IBOA-based CNN achieves a better accuracy of 97.59%, 99.20%, 99.83%, and 98.92% for MICC-F220, MICC-F600, MICC-F2000, and CASIA 2.0 datasets compared with the existing methods like SSDAE-GOA-SHO, DCNN using ResNet-101, and CNN. In the future, detection and localization of the CMF will be focused utilizing various machine learning methods.

**Notation description**

Symbol	Description
$g_c$	pixel centre
$g_p$	circle neighbors with $r$ radius
$u$	uniform local pattern
$x$ and $y$	input and output layer
$x_i^{t+1}$ and $x_i^t$	outcomes for the $i^{th}$ iterations $t + 1$ and $t$
$r$	random value between $[0, 1]$
$x_i^{t+1}$	outcome related to the $i^{th}$ butterfly in the $i^{th}$ iteration
$g$	ideal outcomes for the present iteration
$f_i$	butterfly's fragrance
$x_i^{t+1}$ and $x_k^t$	outcomes for the $j^{th}$ and $k^{th}$ butterflies in the iteration $i^{th}$ correspondingly
$D$	fractional derivative
$\alpha$	fractional derivative order
$\zeta$	truncation order
$T$	sampling period
$ub$ and $lb$	NURBS curve control point adaptation of upper and lower limits
$x_{id}^{t+1}$	outcome for the $id^{th}$ butterfly in the iteration $t + 1$
$Wx$	matrix product
$f$	function component

**Conflicts of interest**

The authors declare no conflict of interest.

**Author contributions**

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, have been done by 1<sup>st</sup> author. The supervision and project administration, have been done by 2<sup>nd</sup> author.

**References**

- [1] R. D. Sushir, D. G. Wakde, and S. S. Bhutada, "Enhanced blind image forgery detection using an accurate deep learning based hybrid DCCAE and ADFC", *Multimedia Tools and Applications*, pp. 1-28, 2023.
- [2] K. M. Hosny, A. M. Mortda, M. M. Fouda, and N. A. Lashin, "An efficient CNN model to detect copy-move image forgery", *IEEE Access*, Vol. 10, pp. 48622-48632, 2022.
- [3] M. Z. Salim, A. J. Abboud, and R. Yildirim, "A visual cryptography-based watermarking approach for the detection and localization of image forgery", *Electronics*, Vol. 11, No. 1, p. 136, 2022.
- [4] E. U. H. Qazi, T. Zia, and A. Almorjan, "Deep learning-based digital image forgery detection system", *Applied Sciences*, Vol. 12, No. 6, p. 2851, 2022.
- [5] Abhishek and N. Jindal, "Copy move and splicing forgery detection using deep convolution neural network, and semantic segmentation", *Multimedia Tools and Applications*, Vol. 80, pp. 3571-3599, 2021.
- [6] N. Kaur, N. Jindal, and K. Singh, "An improved approach for single and multiple copy-move forgery detection and localization in digital images", *Multimedia Tools and Applications*, Vol. 81, No. 27, pp. 38817-38847, 2022.
- [7] P. Yu, J. Fei, Z. Xia, Z. Zhou, and J. Weng, "Improving generalization by commonality learning in face forgery detection", *IEEE Transactions on Information Forensics and Security*, Vol. 17, pp. 547-558, 2022.
- [8] S. S. Ali, I. I. Ganapathi, N. S. Vu, S. D. Ali, N. Saxena, and N. Werghi, "Image forgery detection using deep learning by recompressing images", *Electronics*, Vol. 11, No. 3, p. 403, 2022.
- [9] G. Tahaoglu, G. Ulutas, B. Ustubioglu, M. Ulutas, and V. V. Nabiyeu, "Ciratefi based copy

- move forgery detection on digital images”, *Multimedia Tools and Applications*, Vol. 81, No. 16, pp. 22867-22902, 2022.
- [10] H. Chen, C. Chang, Z. Shi, and Y. Lyu, “Hybrid features and semantic reinforcement network for image forgery detection”, *Multimedia Systems*, Vol. 28, No. 2, pp. 363-374, 2022.
- [11] M. Maashi, H. Alamro, H. Mohsen, N. Negm, G. P. Mohammed, N. A. Ahmed, S. S. Ibrahim, and M. I. Alsaid, “Modelling of Reptile Search Algorithm with Deep Learning Approach for Copy Move Image Forgery Detection”, *IEEE Access*, Vol. 11, pp. 87297-87304, 2023.
- [12] R. Ren, S. Niu, J. Jin, K. Xiong, and H. Ren, “ERINet: efficient and robust identification network for image copy-move forgery detection and localization: Efficient and robust identification network”, *Applied Intelligence*, Vol. 53, No. 12, pp. 16170-16191, 2023.
- [13] J. L. Zhong, J. X. Yang, Y. F. Gan, L. Huang, and H. Zeng, “Coarse-to-fine spatial-channel-boundary attention network for image copy-move forgery detection”, *Soft Computing*, Vol. 26, No. 21, pp. 11461-11478, 2022.
- [14] M. Sabeena and L. Abraham, “Convolutional block attention based network for copy-move image forgery detection”, *Multimedia Tools and Applications*, pp. 1-23, 2023.
- [15] F. Z. E. Biach, I. Iala, H. Laanaya, and K. Minaoui, “Encoder-decoder based convolutional neural networks for image forgery detection”, *Multimedia Tools and Applications*, Vol. 81, pp. 22611–22628, 2021.
- [16] R. Gupta, P. Singh, T. Alam, and S. Agarwal, “A deep neural network with hybrid spotted hyena optimizer and grasshopper optimization algorithm for copy move forgery detection”, *Multimedia Tools and Applications*, Vol. 82, No. 16, pp. 24547-24572, 2023.
- [17] S. Vaishali and S. Neetu, “Enhanced copy-move forgery detection using deep convolutional neural network (DCNN) employing the ResNet-101 transfer learning model”, *Multimedia Tools and Applications*, pp. 1-25, 2023.
- [18] S. Koul, M. Kumar, S. S. Khurana, F. Mushtaq, and K. Kumar, “An efficient approach for copy-move image forgery detection using convolution neural network”, *Multimedia Tools and Applications*, Vol. 81, No. 8, pp. 11259-11277, 2022.
- [19] N. Kaur, N. Jindal, and K. Singh, “A deep learning framework for copy-move forgery detection in digital images”, *Multimedia Tools and Applications*, Vol. 82, No. 12, pp. 17741-17768, 2023.
- [20] N. Goel, S. Kaur, and R. Bala, “Dual branch convolutional neural network for copy move forgery detection”, *IET Image Processing*, Vol. 15, No. 3, pp. 656-665, 2021.
- [21] B. T. Hammad, I. T. Ahmed, and N. Jamil, “An Secure and Effective Copy Move Detection Based on Pretrained Model”, In: *Proc. of 2022 IEEE 13th Control and System Graduate Research Colloquium (ICSGRC)*, pp. 66-70, 2022.
- [22] Dataset link:  
<http://lci.micc.unifi.it/labd/2015/01/copy-move-forgery-detection-and-localization/>
- [23] A. K. Jaiswal and R. Srivastava, “A technique for image splicing detection using hybrid feature set”, *Multimedia Tools and Applications*, Vol. 79, pp. 11837-11860, 2020.
- [24] A. H. Joaquín, G. M. Melendez, and R. Cumplido, “A secure DWT-based dual watermarking scheme for image authentication and copyright protection”, *Multimedia Tools and Applications*, Vol. 82, pp. 42739–42761, 2023.
- [25] F. M. Ghamry, H. M. Emar, A. Hagag, W. E. Shafai, G.M. E. Banby, M. I. Dessouky, A. S. E. Fishawy, N. A. E. Hag, and F. E. A. E. Samie, “Efficient algorithms for compression and classification of brain tumor images”, *Journal of Optics*, Vol. 52, No. 2, pp. 818-830, 2023.
- [26] P. Wu, Z. Wang, H. Jing, and P. Zhao, “Optimal Time-Jerk Trajectory Planning for Delta Parallel Robot Based on Improved Butterfly Optimization Algorithm”, *Applied Sciences*, Vol. 12, No. 16, p. 8145, 2022.