Int. J. Nonlinear Anal. Appl.
Volume 12, Special Issue, Winter and Spring 2021, 1947–1955
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2021.24642.2786

Copy-move forgery detection using a bat algorithm with mutation

Ehsan Amiri[a], Ahmad Mosallanejad[b,*], Amir Sheikhahmadi[a]

[a]Department of Computer Engineering, Sanandaj Branch, Islamic Azad University, Sanandaj, Iran.
[b]Department of Computer Engineering, Sepidan Branch, Islamic Azad University, Sepidan, Iran.

(Communicated by Madjid Eshaghi Gordji)

Abstract

One of the challenges today is capturing fake images. One type of image forgery is a copy-move forgery. In this method, a part of the image is copy and placed at the most similar point. Due to the existing algorithms and processing software, it is not easy to detect forgery areas and has created challenges in various applications. Based on the Bat algorithm, the proposed method has tried to help detect fake images by finding forgery areas. The proposed method includes a simple image segmentation and detection of forgery areas with the help of the BAT Algorithm with Mutation. According to the proposed algorithm, the image is first grayed out, then divided into 100 pieces. The optimal Bat algorithm randomly selects some component image and performs a similarity search. The mutation operator is used to avoid getting stuck in the local optimization. The proposed algorithm does not get stuck in the local optimization with the help of the mutation operator and can find forgery areas with a precision of about 81.39% for the IMD dataset and about 81.04% for the MICC-F600 dataset.

Keywords: Image Forgery, Copy-Move Forgery, BAT Algorithm, Mutation.

1. Introduction

Image forgery is the process [1] of deliberately manipulating an image to alter the information transmitted from it. This manipulation can be done by adding, removing [2], Detecting [3], or modifying [2] any feature of the image or content, leaving no trace of the resulting change. Due to the availability of many free image editing tools and software, image forgery has become easier [4],

*Corresponding author
Email addresses: e.e.amiri@gmail.com (Ehsan Amiri), ahmad.upm@gmail.com (Ahmad Mosallanejad), asheikhahmadi@iausdj.ac.ir (Amir Sheikhahmadi)

Received: August 2021 *Accepted:* December 2021

difficult to detect, and has eroded confidence in the accuracy and integrity of an image. Therefore, robust algorithms for automatic detection of forgery are essential and are an important research problem in image processing [5]. Detection of image forgery [6] is one of the major challenges in digital image processing and forensics [4]. Image mixing [7], image retouching [8], and area duplication or copying [9] are the most common types of image forgery. Among the types of image forgery, copy-move forgery, also called cloning, is the most common model [8]. In this type of image forging, part of an image is copied and pasted into another part of the same image. The main purpose of this type of forgery is to hide unwanted objects, copy some aspects of the image or enhance the visual impact. Copied areas can be of any size and shape and can be pasted one or more times in different places in the same image (Figure 1) [10].

Figure 1: Example of copy-move forgery [10]

Copy Move Forgery Detection (CMFD) aims at both detecting the copy move tampered images and localizing specific tampered regions [11]. Given the importance of this issue in image forgery detection, many researchers have focused on CMFD and have obtained very good results. In general, two types of block based methods [12] and keypoint based methods [13] are examined in this area.

In block-based CMFD methods, most images are divided into several rectangular or circular blocks. The required attributes are obtained according to the selected blocks of the image. Vega et al. (2021) have proposed the CMFD method based on discrete cosine transform (DCT) [14]. After obtaining the required blocks, several different properties are provided to describe the blocks. For example, Hilal et al. (2018) used principal component analysis (PCA) [15] to describe blocks of low complexity, and in (2013), Li et al. Extracted uniform local binary patterns (LBP) from circular blocks [16]. Keypoint methods Extract a large number of key points from the image and perform operations on these points. The scale-invariant feature transform (SIFT) [17] is used in many studies as a key and descriptive point in CMFD. For example, Amerini et al. (2011) proposed copy-move detection based on the SIFT feature and provided a good start in forgery detection [16]. Despite the suitability of the SIFT method, various methods have improved its performance. In [18], the JLinkage algorithm is used to classify matching key points. Another feature, the Speed Up Robust Feature (SURF), is introduced in [9]. In addition, many feature detection [30] methods such as deep neural networks [31], wavelet [32], and optimization methods [33] in fraud detection have been introduced that have been generally considered. In this article, an optimal way to identify different and similar parts of an image is presented. The method will be based on a bat algorithm with mutation (BAM). Due to the challenges in detecting fake points, the proposed method has tried to get the most connection between forgery pixels.

The rest of the article is organized as follows: Section 2 presents the bat algorithm with the mutation method. Section 3 provides a copy-move detection algorithm. Section 4 presents the experiments and Section 5 conclusions.

2. Bat algorithm with mutation (BAM)

The bat algorithm is a swarm intelligence optimization method, in which the search algorithm is inspired by the social behavior of bats and the phenomenon of echolocation to sense distance. In BA, each bat is defined by its position x_t (Eq. 2.1), velocity v_t (Eq. 2.2), frequency f_i (Eq. 2.3), loudness Ai and the emission pulse rate r_i in a d-dimensional search space [19, 20].

$$x_t = x_{t-1} + v_t \tag{2.1}$$

$$v_t = v_{t-1} + (x_t - x^*)f_i \tag{2.2}$$

$$f_i = f_{min} + (f_{max} - f_{min}) \times \beta \tag{2.3}$$

where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution. Here x* is the current global best location (solution) which is located after comparing all the solutions among all the n bats [20]. For the local search part, once a solution is selected among the best solutions, a new solution for each bat is generated locally using random walk. It is calculated by Eq. 2.4.

$$x_{new} = x_{old} + \sigma A_t \tag{2.4}$$

where $\sigma \in [-1, 1]$ is a scaling factor which is a random number, and A_t and A_t are shown as Eq. 2.5 and 2.6.

$$A_{t+1} = \alpha A_t \tag{2.5}$$

$$r_{t+1} = r_0[1 - \exp(-\omega t)] \tag{2.6}$$

Where α and ω are constants. In essence, we set $\alpha = \omega = 0.9$ in this work. A candidate substitutes the parent only if it has better fitness and a greedy selection scheme that often overtakes traditional EAs. The advantages of DE are easy implementation, simple structure, speed, and robustness [21]. As the search relies entirely on random walks, the bat algorithm cannot guarantee fast convergence. In [22], the main modification of adding mutation operator is made to the BA, including two minor modifications, which are made to speed up convergence, thus making the method more practical for a wider range of applications without losing the attractive features. Like BA, in BAM [21], each bat is defined by its position x_i, velocity v_i, the emission pulse rate r_i and the fixed frequency f, loudness A in a d-dimensional search space. The second modification is to add a mutation operator to increase the diversity of the population to improve the search efficiency and speed up the convergence to the optimum. For the local search part, once a solution is selected among the current best solutions, a new solution for each bat is generated locally using a mutation operator in DE, updating the new solution to increase the diversity of the population to improve the search efficiency [22]. X is shown in Eq. 2.7.

$$x_{new} = x_{r1} + F(x_{r2} - x_{r3})) \tag{2.7}$$

Where F is the mutation weighting factor, while r_1, r_2, r_3 are uniformly distributed random integer numbers between 1 and NP.

3. Proposed method

This section proposes a novel matching based on BAM, as shown in Fig. 2. The image is first transferred to the gray surface. A 10 × 10 segmentation is applied to the image method, and the resulting sections are provided to the Bat with mutation algorithm. By detecting similar areas, the copy-move forgery detection operation is performed.

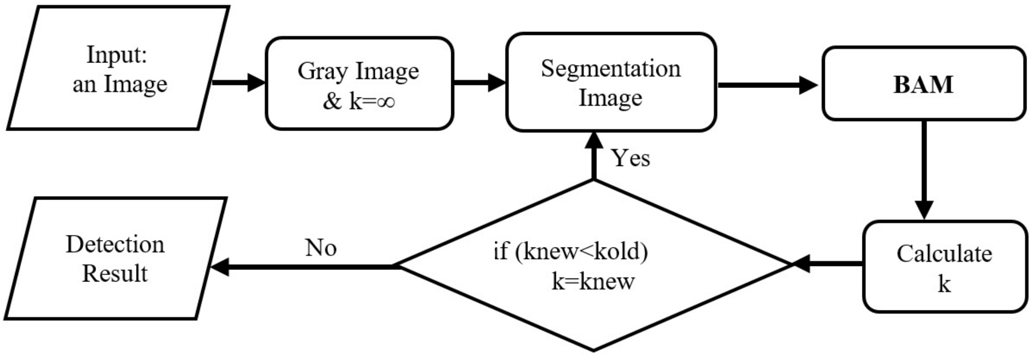

Figure 2: Copy-Move forgery detection with BAM

Almost all digital images are RGB-based color images. First, convert them into a gray-scale by combining all the three channels using Eq. 3.1.

$$Y = 0.298R + 0.582G + 0.117B \tag{3.1}$$

Where R, G, and B indicate the pixel intensities of red, green, and blue channels, respectively.

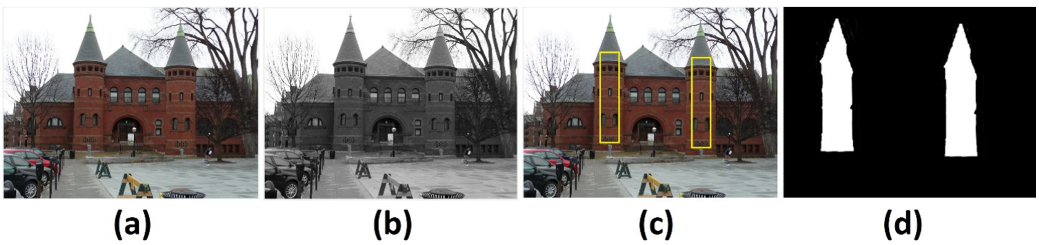

Figure 3: Copy Move with BAM, (a) Main Image, (b) Gray Image, (c) Image Matching with BAM, (d) Detection Result

The proposed algorithm performs copy-move detection with the help of bat algorithm. Swarm algorithms must have random initial values to execute and optimize. In the proposed method, set the number of pixels(k) with infinity, and the image will be divided into 100 sections in the second stage, including ten rows and ten columns. The reason for choosing this type of division is the equality of parts and the simplicity of its implementation. Randomly five sections will be considered as the input of the bat algorithm. The BAM algorithm [22] shown in Figure 2 receives the random parts of the input and performs the similarity check with the help of the fitness function introduced in Eq. 3.2.

$$fitness(x, y) = \sum_{i=1}^{M2} \sum_{j=1}^{N2} (|I_1(i + x - 1, i + y - 1) - I_2(i, j)|) \tag{3.2}$$

Where (x, y) represents the position in the original section, I_1 and I_2 are the pixel values for the original section and object section, respectively. Among many definitions for fitness (x, y), the best fitness (the minimum fitness) is the matching point. For timely implementation, it needs to calculate (M1-M2+1) *(N1-N2+1) fitness values, and it is so time-consuming to compute that it cannot satisfy the real-time application. Therefore, the paper uses a bat algorithm with mutation (BAM) to accelerate searching speed. Updates each round according to the maximum similarity between the speed and location parameters. A section compares all the sections in the image and returns the section that has the most similarity. Performs this process for the five initial selection sections. Suppose the BA algorithm is used and the initial random segments are not forged segments. In that case, the algorithm (Table. 1) will never work because no similar segments will be obtained for the initially selected segments. In the proposed model, if the first five parts do not find similarity, the mutation operation will be performed according to Eq. 7. The other five random parts will be selected. In a way, the BAM algorithm will move towards the answer by jumping from the selected parts. The number of steps of this algorithm is considered to be 100 rounds. After completing all the steps, there are two possible scenarios. In the first case, similar sections are found. In this case, the number of pixels that make up the selected part is calculated (knew). Moreover, compared to the number of previous pixels. Suppose it is less than the number of previous pixels. In that case, it indicates a change in the selected part, and as a result, the model's sensitivity to selecting correct forgery pixels increases with this process.

The second case is when no similarities are seen. In this case, since the number of initial pixels (k) has not changed, the selection and re-sorting operation is performed. The proposed algorithm could not find a suitable section on 78% of the images in the first round based on the investigations. However, in other rounds, this operation was completed, and similar sections were found. If, after five rounds of the whole algorithm, the same part is not found or the number of k pixels does not change, the algorithm stops. The main drawback of the proposed method is the selection of the number of revolutions k, which is tried to be addressed in the next articles and researches, and as a result, to obtain better precision and recall.

4. Experimental setup

4.1. Data set

Here, will examine series of data to copy-move forgery detection. The first database contains the IMD (Image Manipulation Dataset) public image data set (Figure. 4) [13] that has been used to evaluate the proposed method. The IMD dataset, sometimes known as CoMoFoD, includes 48 different simple images, rotating images, JPEG compression images, and noise images. The largest image in this dataset is about 3000 × 2300 pixels. In this dataset, about 10% of the areas of each image are manipulated.

Figure 4: Example results of the BAM forgery detection algorithm on the IMD dataset. (a1) and (b1) Original image [13]. (a2) and (b2) Detected forged region

The second database is known as MICC-F600 [23], which contains 1440 images (Figure. 5). This data set has been used to construct test images with more types of area manipulation. The size of the images in this dataset varies from 800 $\times$ 533 to 3888 x 2592 pixels. This set includes (1) single copies: forged areas are reproduced once. (2) Multiple copies: Forgery areas have been duplicated two or three times.

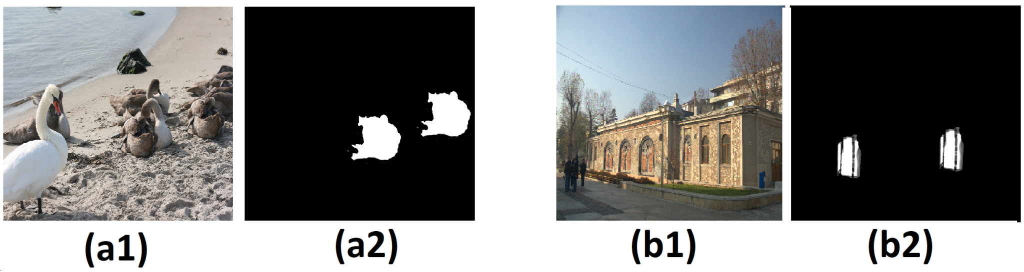

Figure 5: Example results of the BAM forgery detection algorithm on the MICC-F600 dataset(a1) and (b1) Original image. (a2) and (b2) Detected forged region.

4.2. Performance measures

For certain, CMFD aims to promote detection precision and recall its best to find all the pixels belonging to the tampered region. The performance of the CMFD schemes is tested at two levels: the image level and the pixel level. At the image level, whether an image has been tampered with or not is emphasized, while pixel-level focuses on correctly locating the tampered regions. Generally, three commonly used indexes, precision(Eq. 4.1), recall(Eq. 4.2) and F1(Eq. 4.3), represent the effect of different aspects, which are also applied to our method evaluation. They are calculated as [24]:

$$Precision = (A \cap B)/|A| \tag{4.1}$$

$$Recall = |A \cap B|/|B| \tag{4.2}$$

$$F1 = 2(Precision\text{"."}Recall)/(Precision + Recall) \tag{4.3}$$

To calculate these parameters, two factors, A and B, are defined, A as the detected images by the method and B as the forged images of the data set. At the image level, precision is computed as the ratio of the number of correctly detected forged images to the number of totally detected forged images, as shown in Eq. (4.1) and, recall is computed as the ratio of the number of correctly detected forged images to the total number of forged images in the dataset, as shown in Eq. (4.2). F1 combines both precision and recall as a weighted average measure. The score is called the F1 Score because it gives equal weights to both precision and recall, as shown in Eq. (4.3).

4.3. Comparison results and analysis

The results of quantitative analysis on the images are taken according to the proposed model, which includes the detection of forgery with the help of BAM. The Keypoint method automatically detects fake images, but the results are not complete and accurate. precision in detecting Copy-Move forgery is the possibility of identifying real forgery points, and recall is the possibility of detecting forged images.

4.3.1. Results on IMD

In this section, the identified results are compared with some of the advanced CMFD methods. These methods include: SIFT [23], KAZE [25], LIOP [26], PCET [27], and MSA [28]. In this case, the results are shown in the simple copy subset in Table 2.

Table 2 shows that the BAM method has the highest precision (81.39%), followed by 80.77% in SIFT and 75.48% in MSA. However, the goal of the CMFD method is to detect as much as possible of all manipulated images. It is more important to detect fake images for a set of images containing real and image forgery.

4.3.2. Results on MICC-F600

This section compares the identified results with some of the advanced CMFD methods on the MICC-F600 dataset. The methods of introduction in this section, as in the previous section, are: These methods include: SIFT [23], KAZE [25], PCET [27], MSA [28] and DAMF [29]. In this case, the results are shown in the simple copy subset in Table 3.

This table shows that the proposed method is more accurate than the other methods presented in Table 3. According to the recall column, it is clear that the number of image forgery detected by this method is much higher than other methods. The precision of other methods is high because they detect images forgery correctly, but the proposed method, in addition to increasing the precision, detects more image forgery.

5. Conclusion and future work

Copy-move is the most common method of image manipulation in which image areas are copied internally. The proposed method focuses on detecting copy-move forgery using the BAM model. Experimental analysis proved the effectiveness of the proposed method in detecting forgery and transmission of forgery. This method offers a higher detection rate and precision. Results show a significant improvement in the precision and value of the F1 Score compared to other algorithms. It has also shown relatively good results for call rates. The results show that the proposed method detects copy-move counterfeiting and achieves a precision of about 81.39% for the IMD dataset and about 81.04% for the MICC-F600 dataset. Future work will focus on improving the localization precision of the area and expanding the method for detecting other types of image forgery.

References

[1] A. Roy, R. Dixit, R. Naskar and R. S. Chakraborty, *Copy-Move Forgery Detection in Digital Images—Survey and Accuracy Estimation Metrics,* In Digital Image Forensics, (2020) 27-56.

[2] A. Novozámský and M. Šorel, *Detection of Copy-Move Image Modification Using Jpeg Compression Model,* Forensic Sci. Inter., (2018) 47-57.

[3] E. Amiri, Z. Roozbakhsh, S. Amiri and M.H. Asadi, *Detection of Topographic Images of Keratoconus Disease Using Machine Vision,* Int. J. Eng. Sci. Appl., 4 (4) (2020) 145-50.

[4] P. M. Raju and M. S. Nair, *Copy-Move Forgery Detection Using Binary Discriminant Features,* J. King Saud Univ.-Comput. Inf. Sci., (2018).

[5] B. Mahdian and S. Saic, *Detection of Copy–Move Forgery Using a Method Based on Blur Moment Invariants,* Forensic Sci. Int., 171 (2-3) (2007) 180-89.

[6] I.Amerini, L. Ballan, R. Caldelli, A.D. Bimbo, L.D. Tongo and G. Serra, *Copy-Move Forgery Detection and Localization by Means of Robust Clustering with J-Linkage,* Signal Process.: Image Commun., 28 (6)(2013) 659-69.

[7] W. Abd, N. Bakiah, A. W. A. Wahab, M. Y. I. Idris, R. Ramli, R. Salleh, S. Shamshirband and K. R. Choo, *Copy-Move Forgery Detection: Survey, Challenges and Future Directions,* J. Network Comput. Appl., 75 (2016) 259-278.

[8] J. C. Lee, *Copy-Move Image Forgery Detection Based on Gabor Magnitude,* J. Visual Commun. Image Represent., 31 (2015) 320-334.

[9] B. Shivakumar and S. S. Baboo, *Detection of Region Duplication Forgery in Digital Images Using Surf,* Int. J. Comput. Sci. Issues (IJCSI), 8 (4) (2011) 199.

[10] H. A. Alberry, A. A. Hegazy and G. I. Salama, *A Fast Sift Based Method for Copy Move Forgery Detection,* Future Comput. Inf. J., 3 (2) (2018) 159-165.

[11] K. Liu, W. Lu, C. Lin, X. Huang, X. Liu, Y. Yeung and Y. Xue, *Copy Move Forgery Detection Based on Keypoint and Patch Match,* Multimedia Tools Appl., 78 (22) (2019) 31387-31413.

[12] Y. Sun, R. Ni and Y. Zhao, *Nonoverlapping Blocks Based Copy-Move Forgery Detection,* Secur. Commun. Networks, 2018 (2018).

[13] E. Ardizzone, A. Bruno and G. Mazzola. *Copy–Move Forgery Detection by Matching Triangles of Keypoints.* IEEE Trans. Inf. Forensics Secur., 10 (10) (2015) 2084-2094.

[14] E. A. Armas. Vega, E. G. Fernández, A. L. Sandoval. Orozco and L. J. García. Villalba, *Copy-Move Forgery Detection Technique Based on Discrete Cosine Transform Blocks Features,* Neural Comput. Appl., 33 (10) (2021) 4713-4727.

[15] A. Hilal and S. Chantaf, *Uncovering Copy–Move Traces Using Principal Component Analysis, Discrete Cosine Transform and Gabor Filter,* Analog Integr. Circuits Signal Process., 96(2) (2018) 283-91.

[16] L. Li, S. Li, H. Zhu, S.C. Chu, JF. Roddick and J.S. Pan, *An Efficient Scheme for Detecting Copy-Move Forged Images by Local Binary Patterns,* J. Inf. Hiding Multim. Signal Process., 4 (1) (2013) 46-56.

[17] D. G. Lowe, *Object Recognition from Local Scale-Invariant Features*, Int. J. Comput. Vision, 60(2) (2004) 91-110.

[18] G. Jin and X. Wan, *An Improved Method for Sift-Based Copy–Move Forgery Detection Using Non-Maximum Value Suppression and Optimized J-Linkage,* Signal Process.: Image Commun., 57 (2017) 113-125.

[19] Y. Wang, P. Wang, J. Zhang, Z. Cui, X. Cai, W. Zhang and J. Chen, *A Novel Bat Algorithm with Multiple Strategies Coupling for Numerical Optimization,* Mathematics, 7 (2) (2019) 135.

[20] M.A. Al-Betar and MA. Awadallah, *Island Bat Algorithm for Optimization,* Expert Sys. Appl., 107 (2018) 126-45.

[21] W.H. Bangyal, J. Ahmad, H.T. Rauf and S. Pervaiz, *An Overview of Mutation Strategies in Bat Algorithm,* Int. J. Adv. Comput. Sci. Appl. (IJACSA), 9 (8) (2018) 523-34.

[22] J.W. Zhang and G.G. Wang, *Image Matching Using a Bat Algorithm with Mutation,* Paper presented at the Applied Mechanics and Materials, 2012.

[23] I. Amerini, L. Ballan, R. Caldelli, A.Del. Bimbo and G. Serra, *A Sift-Based Forensic Method for Copy–Move Attack Detection and Transformation Recovery,* IEEE Trans. Inf. Forensics Secur., 6(3) (2011) 1099-110.

[24] Q. Lyu, J. Luo, K. Liu, X. Yin, J. Liu and W. Lu, *Copy Move Forgery Detection Based on Double Matching,* J. Visual Commun. Image Represent., 76 (2021) 103057.

[25] F. Yang, J. Li, W. Lu and J. Weng, *Copy-Move Forgery Detection Based on Hybrid Features,* Eng. Appl. Artif. Intell., 59 (2017) 73-83.

[26] C. Lin, W. Lu, X. Huang, K. Liu, W. Sun, H. Lin and Z. Tan, *Copy-Move Forgery Detection Using Combined Features and Transitive Matching,* Multimedia Tools Appl., 78 (21) (2019) 30081-30096.

[27] M. Emam, Q. Han and X. Niu, *Pcet Based Copy-Move Forgery Detection in Images under Geometric Transforms,* Multimedia Tools Appl., 75 (18) (2016) 11513-11527.

[28] E. Silva, T. Carvalho, A. Ferreira and A. Rocha, *Going Deeper into Copy-Move Forgery Detection: Exploring Image Telltales Via Multi-Scale Analysis and Voting Processes,* J. Visual Commun. Image Represent., 29 (2015) 16-32.

[29] J. Deng, J. Yang, S. Weng, G. Gu and Z. Li, *Copy-Move Forgery Detection Robust to Various Transformation and Degradation Attacks,* KSII Trans. Internet Inf. Syst.(TIIS) 12 (9) (2018) 4467-4486.

[30] S. Amiri, A. Mosallanejad and A. Sheikhahmadi, *Medical images fusion based on equilibrium optimization and discrete wavelet,* Int. J. Nonlinear Anal. Appl., 12, (Special Issue) (2021) 1337-1354.

[31] A. Kazemi, M. E. Shiri, A. Sheikhahmadi and M. Khodamoradi, *A new parallel deep learning algorithm for breast cancer classification,* Int. J. Nonlinear Anal. Appl., 12 (Special Issue) (2021) 1269-1282.

[32] E. Amiri, M. Rahmanian, S. Amiri and H. Yazdani Praee, *Medical images fusion using two-stage combined model DWT and DCT,* Int. Adv. Res. Eng. J., 5 (3) (2021) 344-351.

[33] N. khaledian, F. Mardukhi, *CFMT: a collaborative filtering approach based on the nonnegative matrix factorization technique and trust relationships,* J. Ambient Intell. Hum. Comput., (2021) 1-17.

Table 1: Copy Move detection algorithm using BAM

Algorithm execution procedure
Begin
I=input image;
I1=gray image(I) by Eq. 8; $k = \inf$; I1 is selected section
for j=1:5
I1=segmentation section in 10×10
Begin BAM algorithm
Step 1: Initialization. Set the generation counter t = 1;
Initialize the population of NP bats section image randomly
and each bat corresponding to a potential solution to the given problem;
define pulse frequency Q; set loudness A_i, the initial velocities v_i
and pulse rate $r_i(i = 1, 2, , NP)$; set weighting factor F.
Step 2: Evaluate the fitness for each bat in section image by Eq. 9
Step 3: while The halting criteria is not satisfied or t ¡ MaxGeneration do
Sort the population of bats P from best to worst by order of fitness for each bat;
for i=1:NP (all bats) do
Select uniform randomly $r1 \neq r2 \neq r3 \neq i$
$r_4 = [NP * rand]$
$v_i^t = v_i^{(}t - 1) + (v_i^t - x_*)Q$
$x_i^t = x_i^{(}t - 1) + v_i^t$
if (rand ¿ r) then
$x_u^t = x_* + \alpha A^t$
else
$x_u^t = x_(r_1)^t + F(x_(r_2)^t - x_(r_3)^t)$
end if
Evaluate the fitness for the offsprings x_u^t, x_i^t
end for i
Evaluate the threat cost for each bat in image section by Eq. 9.
Sort the population of bats P from best to worst by fitness order for each bat; t = t+1;
Step 4: end while
Step 5: Inversely transform the coordinates in final optimal path
into the original coordinate, and output
End BAM algorithm
Calculate k pixel
if knew¡kold then
k=knew
end if
End for j;
End.

Table 2: Results of IMD dataset

Methods	Precision (%)	Recall (%)	F1 (%)
SIFT [23]	80.77	43.75	56.75
KAZE [25]	71.43	83.33	76.93
LIOP [26]	73.44	75.41	74.42
PCET [27]	73.65	62.77	67.69
MSA [28]	75.48	73.28	74.36
Proposed method	81.39	83.79	82.19

Table 3: Results of MICC-F600 dataset

Methods	Precision (%)	Recall (%)	F1 (%)
SIFT [23]	77.55	42.21	54.67
KAZE [25]	68.40	51.40	58.70
LIOP [26]	71.14	66.34	67.69
PCET [27]	64.58	72.45	68.00
MSA [28]	73.86	73.28	74.00
Proposed method	81.04	81.36	81.15