Information Security Journal: A Global Perspective

ISSN: 1939-3555 (Print) 1939-3547 (Online) Journal homepage: www.tandfonline.com/journals/uiss20

ForgNetwork: An adaptive forgery detection and localization framework using stationary wavelet transform and dilated adaptive VGG16 with optimization strategy

Prabhu Bevinamarad & Prakash Unki

To cite this article: Prabhu Bevinamarad & Prakash Unki (07 Nov 2025): ForgNetwork: An adaptive forgery detection and localization framework using stationary wavelet transform and dilated adaptive VGG16 with optimization strategy, Information Security Journal: A Global Perspective, DOI: 10.1080/19393555.2025.2570170

To link to this article: https://doi.org/10.1080/19393555.2025.2570170

INFORMATION SECURITY JOURNAL: A GLOBAL PERSPECTIVE
https://doi.org/10.1080/19393555.2025.2570170

Taylor & Francis
Taylor & Francis Group

ForgNetwork: An adaptive forgery detection and localization framework using stationary wavelet transform and dilated adaptive VGG16 with optimization strategy

Prabhu Bevinamarad[a] and Prakash Unki[b]

[a]Department of Computer Science and Engineering, B.L.D.E.A's V.P.Dr.P.G. Halakatti College of Engineering and Technology, Vijayapura, India; [b]Department of Information Science and Engineering, B.L.D.E.A.'s V.P.Dr. P.G. Halakatti College of Engineering and Technology, Vijayapura, India

ABSTRACT

Recently, the ability to alter images has become remarkably accessible, allowing for effortless manipulation of an image's semantic content using widely available editing tools and techniques. These techniques are also called journaling tools, and it is used to effectively detect the changes in images. In the image processing technique, the deviations in images are determined by using typically square, slide regular, and artifacts techniques. But, the determination of these changes in images is a tedious procedure. To determine the forgery activities, a suitable method should be developed with the help of wavelet transform and deep learning techniques. Initially, images are gathered from various online sources. These collected images are then processed using Adaptive SWT, where the parameters of SWT are optimized using a HTS-BESO. The adaptive SWT splits the entire image into patches for each sub-band. Following this, the DA-VGG16Net framework is employed to extract deep features from these split patches. The parameters of DA-VGG16Net are also optimized using HTS-BESO. Finally, feature matching is conducted using multi-similarity checking to recognize and localize forgeries within the images. The experimental results are compared to various existing forgery detection models to ensure the efficiency of the model by considering various performance measures.

KEYWORDS

Adaptive stationary wavelet transform; Dilated Adaptive VGG16; forgery detection and localization; hybridized tuna swarm with bald eagle search optimization

1. Introduction

The introduction of the latest technologies in digital processing has led to alterations in original images (Zhuo et al., 2022). However, these modifications often leave behind subtle clues in the surrounding environment. These changes in images cause problems in medical records in the medical field and court evidence (Gu et al., 2022). To determine the originality of images, various authentication methods are employed (Li et al., 2017). These authentication methods are further divided into two parts: one is the active authentication technique, and the other one is the passive authentication technique. In the active authentication process, the modified images are tested by signature code, and this signature code is available in image capturing (Aloraini et al., 2021). Here, the digital mark is marked on the multimedia content. Therefore, changes in multimedia content can be easily identified (Yang et al., 2021). However, all image-capturing devices do not have digital mark and signature codes, so this method is not apt in all places (Barni et al., 2021). In the passive authentication technique, the integrity of an image is assessed by identifying duplicated regions within the image. However, these duplicated regions often do not contain any explicit clues or information about their authenticity (Lee et al., 2022). Passive authentication techniques consist of various techniques. Here, splicing and forgery detection are the most effective processes. The discovery of forgery is a tedious process because it cannot be able to find the difference between the original and duplicated images.

An image can be easily changed by using advanced editing tools. An image can be changed by using two techniques, and they are content-preserving and content-changing (Rao et al., 2020). In the content-preserving system, techniques such as contrast enhancement, blurring, and

CONTACT Prabhu Bevinamarad prabhu.rb@bldeacet.ac.in Department of Computer Science and Engineering, B.L.D.E.A's V.P.Dr.P.G. Halakatti College of Engineering and Technology, Vijayapura, Karnataka 586103, India

compression are applied to the post-processed image. These modifications do not affect the original cognitive content of the image (Kadam et al., 2021). So, it is a less destructive method. The second method, namely content changing system, copy-move, splicing, and object removal techniques, are carried out in the processed image (Huang et al., 2022). It may cause changes in the semantic content of the image. In recent years, several works have been developed for forgery detection. To find the duplicated image at various locations, forensic clues like square detection windows are used (Niu et al., 2021). The image processing is done using the sliding-window paradigm, which is considered an effective method for image processing (Yan & Pun, 2017). However, this method has certain disadvantages (Alipour & Behrad, 2020). The square shape of the window is not suitable for detecting human limbs (Zhou et al., 2022). Additionally, it may cause errors when processing heterogeneous images, and it is not suitable for all pixels of the image.

Most state-of-the-art image tampering categorization methods make use of frequency domain features and/or statistical characteristics of an image (Cristin & Cyril Raj, 2017). Artifact analysis by many JPEG compressions is also used to identify tampered images. It is suitable only for JPEG formats. Noise is added to the JPEG compressed image to increase the performance of similarity detection. Visual recognition tasks such as semantic separation, object identification, and scene classifications are done with the help of deep learning techniques (Meena & Tyagi, 2020). Deep learning-based methods such as Stacked Auto-Encoders (SAE) and Convolutional Neural Networks (CNN) are also used to identify duplicate images. Particular tampering methods such as splicing and copy-move are used to identify the forgery activities in media (Vinolin & Sucharitha, 2021). Thus, one method may not hold well over other types of manipulation. Furthermore, it looks unrealistic to presume that the type of tampering will be identified in advance.

The main contributions of the proposed deep learning-based forgery detection mechanism are listed as follows.

- To design an effective advanced deep networks-based forgery detection mechanism to identify the forgery activities in images and prevent the forgery activities with low cost.
- To increase the performance of the adaptive forgery identification and localization framework, a new HTS-BESO algorithm is developed. By using this HTS-BESO, the parameters from Adaptive SWT and the parameters from Dilated VGG16 are optimized to provide more accurate detection results.
- To decompose the image using SWT, which helps to detect the most similar and discriminative features from the decomposed images, where the parameters like start level, wavelet type, and norm are optimized with the help of the suggested HTS-BESO algorithm to improve the decomposition performance.
- To suggest a DA-VGG16Net model for detecting the forgeries from the objects by checking the multi-similarity, where the parameters like epochs and hidden neuron count in VGG 16 are optimized with the help of suggested HTS-BESO algorithm to maximize the accuracy and precision.
- The output of the suggested HTS-BESO-DA-VGG 16 Net-based forgery detection scheme is compared with various existing algorithms and recently developed techniques to find the effectiveness of the planned method.

The proposed HTS-BESO-based optimization algorithm to prevent fraud activities is explained in the below sections. Section II deals with the related work of the existing forgery detection methods. Section III discusses the developed forgery detection mechanism and data collection. In section IV, SWT and Adaptive SWT are described in detail. In section V, Dilated Adaptive VGG16 and Multi-similarity methods for detecting forgeries are briefly explained. Section VI depicts the result discussion of the offered forgery detection scheme. Section VII gives details about the conclusion of the proposed HTS-BESO-DA-VGG16Net-based forgery detection scheme.

2. Literature survey

2.1. Related works

Dua et al. (2020) have proposed a thorough procedure for looking into JPEG compressed test

images that were thought to have been altered either by splicing or copy-move forgery. The image plane was split into 8×8non-overlapping pixel units for JPEG compression. To determine whether the image was genuine or fake and to locate the altered area in fake photos, a unified technique based on the block-processing of JPEG images has been presented. The system was able to distinguish between spliced and copy-moved forgeries of images. Once the presence of tampering has been established, the next stage involves localizing the falsified region according to the type of forgery detected. By identifying and delineating the altered region, forensic investigators can focus their analysis and efforts on understanding the extent and nature of the forgery, which is essential for accurate assessment and potential legal proceedings. Using block-wise correlation maps of un-estimated coefficients of DCT and its recompressed form at various quality levels, the tampered region of spliced JPEG images was located. Using all possible combinations of quality variables, the technique was able to spot the spliced section of images that had been altered by pasting an uncompressed one. In the case of copy-move forgeries, the duplicated regions were found instead of utilizing each block's extremely localized phase congruency properties. The concert of the proposed technique was compared to other state-of-the-art techniques, and experimental findings were presented to support the theoretical notion.

Bappy et al. (2019) suggested a high-confidence tampering localization structure that separated tampered regions from un-tampered ones using resembling features using Long Short-Term Memory (LSTM) cells and an encoder-decoder network. Artifacts such as JPEG, upsampling, rotation, quality degradation, downsampling, and shearing were captured using resembling features. Lastly, the mapping from low-resolution feature maps to pixel-wise predictions for image tamper localization was then learned by the decoder network. End-to-end training was conducted utilizing the ground-truth masks and the anticipated mask offered by the final layer of the proposed architecture to learn the network parameters. Moreover, a sizable image-splicing dataset was added to direct the training procedure. Extensive testing on three different datasets showed that the suggested strategy was capable of precisely localizing image modifications at the pixel level.

Lin and Li (2020) explored a segmentation-based forgery detection method that uses the local uniformity of visually unnoticeable clues to change the limitations of existing segmentation methods that were merely based on visually perceptible content and suggested a forgery localization method depending on Photo-Response Non-Uniformity (PRNU). The method for multi-orientation localization incorporated the forgery probability discovered by picture segmentation and windows with many orientations. To localize object insertion and object removal frauds, specialized techniques are employed to accurately detect and highlight the manipulated regions within an image. The suggested forgery detection approaches surpassed current PRNU-based forgery localizers concerning both the region and border F1 scores, according to experimental results on a publicly available realistic tampering picture dataset.

Ganeshan et al. (2022) offered a Generative Adversarial Network based on Autoregressive Elephant Herding Optimization (A-EHO-based GAN). The Risk by Regression-based Conditional Autoregressive Value Quantiles (CAViaR) and EHO approaches have been combined to create the proposed A-EHO approach. For each foreground item, first, features such as Local Optimal Oriented Pattern (LOOP) and CNN features were retrieved. The forgery score is calculated by features of the GAN. Based on the feature vector and the forgery score, the Ridge NN classifier identified the fake image in this case. The outcome was that the implemented technique performed better in terms of TNR, detection rate, TPR, and ROC.

Abdalla et al. (2019) offered a technique to identify the forgeries and a deep convolution learning technique. They have been demonstrated to be very successful in combating image forgeries produced by GAN. In this type of algorithm, the image was altered to closely resemble the original, making it difficult for an untrained human eye to distinguish it as a fake. The goal of the current study was to examine the processing model by using copy-move forgery detection; it is made up of an adversarial model and a deep convolution model. The results demonstrated a very high detection accuracy

performance evaluated by the CNN and discriminator forgery detectors. The network was created using a fusion module and two-branch architecture. CNN and GAN, the two types, were used to localize and identify copy-move forgery locations.

Mahmood et al. (2018) proposed a reliable method for the identification and detection of CMF in digital photos. To reveal image forgeries, the technique retrieved SWT-based characteristics. SWT's remarkable localization capabilities in the spectral and spatial domains led to its adoption. More particular, the SWT approximation subband was used since it included the most data that was most suitable for forgery detection. Applying Discrete Cosine Transform (DCT) has decreased the feature vectors' dimension. They conducted experiments using two common datasets, CoMoFoD and UCID, to evaluate the proposed technique. The evaluation output showed that the offered strategy performed better in terms of true and false identification rate than the methods already in use. As a result, the suggested forgery detection method might be used to identify the altered areas and provide advantages in image forensic applications.

Bilal et al. (2020) offered a reliable CMF identification method to address the aforementioned issues. The suggested method combined Binary Robust Invariant Scalable Key Points (BRISK) and Speeded up Robust Features (SURF) descriptors. SURF characteristics stood up well to various post-processing assaults like blurring, rotation, and additional noise. Nevertheless, the scale-invariant forged portions and poorly located key points of the objects within the forged image were judged to be difficult to detect using conventional methods. The second nearest neighbor and hamming distance were utilized to match the fused features. The remaining erroneous matches were eliminated from the clusters using the random sample consensus technique. The fabricated regions were found and localized after some post-processing. Using three common datasets, the performance of the proposed CMFD approach was evaluated. In terms of true and false detection rates, the proposed strategy outperformed the cutting-edge methods employed for CMF detection.

Koul et al. (2022) explored a forgery detection method based on deep learning. In this digital monarchy, copy-move and image splicing to produce fake images were commonplace. The former used merging two images to dramatically alter the real image and produce a new fake one, while the latter involved copy-move, which entailed resembling one element of the image and pasting it to another part of the image. Here, a novel approach utilizing CNNs has been proposed for the identification of forgeries. A dataset called MICC-F2000, which included 2000 photographs, of which 1300 were genuine, and 700 were forgeries, was taken into consideration for the experimental work. The experimental results showed that the suggested model worked better than other established techniques for detecting copy-move forgeries.

2.2. Problem statement

The authenticity of images is questionable when the usage of the widespread availability of digital devices, commercially available image editing tools, and open-source tools. Hence, the localization process is very important for identifying the tampered areas to provide higher authenticity. Therefore, several forgery detection approaches are developed using deep learning techniques. The features and demerits related to the existing forgery detection methods are given in Table 1. DCT (Dua et al., 2020) effectively identified the duplicate regions using the phase congruency features. In addition, it localized the tampered regions with high accuracy. However, it struggles to get the phase congruency features from the multiple orientations in the covariance matrix. CNN-LSTM (Bappy et al., 2019) segments different types of manipulations such as copy-move and, object removal, splicing. Therefore, it effectively handles the large dimensional dataset to localize tampers. Yet, there is a chance for missing useful information during the splicing of large dimensional datasets. Furthermore, it is slightly affected by noise. PRNU (Lin & Li, 2020) highly exploits the local homogeneity from indiscernible clues. For instance, it performs very well on forgery localization of both boundary and region in terms of the F1 score. However, it cannot perform multi-orientation detection among both the soft and hard boundary forgeries. Moreover, it increases the computational complexity while localizing hard boundary forgeries. CNN (Ganeshan et al., 2022)

Table 1. Features and challenges of previous forgery detection approaches.

Author [citation]	Methodology	Features	Challenges
Dua et al., (2020)	DCT	• It effectively identified the duplicate regions using the phase congruency features. • It localized the tampered regions with high accuracy.	• It struggles to get the phase congruency features from the multiple orientations in the covariance matrix.
Bappy et al., (2019)	CNN-LSTM	• It segments different types of manipulations such as object removal, copy-move, and splicing. • It effectively handles the large dimensional dataset to localize tampers.	• There is a chance for missing useful information during the splicing of large dimensional datasets. • It is slightly affected by noise.
Lin & Li, (2020)	PRNU	• It highly exploits the local homogeneity from indiscernible clues. • It performs very well on forgery localization of both boundary and region in terms of the F1 score.	• It cannot perform multi-orientation detection among both the soft and hard boundary forgeries. • It increases the computational complexity while localizing hard boundary forgeries.
Ganeshan et al., (2022)	CNN	• It accurately captures the copied image portion in the same image. • It is very simple to implement, and the detection rate is higher than other approaches.	• It is more complex when the features from the copied portion are unique to the other portions in the same image.
Abdalla et al., (2019)	GAN-CNN	• It constantly updates the learning ability of trained data. • It provided high sensitivity in forgery detection.	• It needs more time to train the CNN model to produce efficient results over forgery detection.
Mahmood et al., (2018)	SWT-DCT	• The tampered detection areas are effectively detected by using this developed model. • It detects the forgeries by concealing the replication of desirable objects to enhance the performance of the model concerning false and true detection rates.	• It is obscured utilizing several factors like additive noise, larger scaling, contrast adjustment, painting, and blindness. Therefore, post-processing approaches are required to solve these issues.
Bilal et al., (2020)	DBSCAN Clustering	• It provides accurate and robust results for multiple and single forged regions. • It provides time-efficient results during the detection of scale-invariant forged regions.	• It struggles to detect post-processing attacks like brightness change, smoothening, and excessive scaling.
Koul et al., (2022)	CNN	• It provides improved forgery detection accuracy than the other methods. • It is helpful to alleviate the derelictions.	• It loses some information during merging. • It is affected by white Gaussian noise.

accurately captures the copied image portion in the same image. Furthermore, it is very simple to implement, and the detection rate is higher than other approaches. Yet, it is more complex when the characteristics from the copied portion are unique to the other portions in the same image. GAN-CNN (Abdalla et al., 2019) constantly updates the learning ability of trained data. On the other hand, it offers high sensitivity in forgery detection. However, it requires additional time to train the CNN model effectively for optimal results in forgery detection. In SWT-DCT (Mahmood et al., 2018), the tampered detection areas are effectively detected by using this developed model. Meanwhile, it detects the forgeries by concealing the replication of desirable objects and enhances the production of the model in terms of false and true detection rates. However, it is obscured utilizing several factors like additive noise, larger scaling, contrast adjustment in painting, and blindness. Therefore, post-processing approaches are required to solve these issues. DBSCAN Clustering (Bilal et al., 2020) provides precise and robust results for the multiple and single tampered regions. In addition, it provides time-efficient results during the detection of scale-invariant forged regions. Nevertheless, it struggles to detect post-processing attacks like brightness change, smoothening, and excessive scaling. CNN (Koul et al., 2022) provides improved forgery detection accuracy than the other methods. Moreover, it helps to alleviate derelictions. It is the chance to lose information from the original data. In addition, it is affected by white Gaussian noise. These issues that arise from the existing forgery identification methods are resolved using the newly developed advanced deep structure-based forgery localization and detection system.

3. ForgNetwork: schematic representation and dataset description of proposed adaptive forgery detection and localization framework

3.1. Developed a forgery detection mechanism

Images are the important thing that is used as evidence in criminal investigation and forensic

investigations. They are also used in the medical field to ensure our health issues. But, changes in medical documents like X-rays cause serious problems in our body's health. In recent years, image editing tools have occupied more space in our daily lives. They are Photoshop, Photo Plus mobile editing apps, etc. Therefore, the image can be easily altered by using these editing tools. Because of these editing tools, the semantic content of the image can be easily altered. Visual media is a platform that plays a crucial role in our daily lives. At the same time, image manipulation is a problem associated with visual media. These manipulations cause forgery activities in crime investigations. And it may also lead to false evidence in a judicial court. But, we cannot be able to determine the alteration caused in the original image because they are so minute, and we can't be able to see with our eyes. Several methods are developed to detect the difference between the original image and the manipulated image. However, they have several limitations like low precision, small robustness, and high false alarm rate. Existing methods are highly prone to fuzzy attacks, so they must be performed in a separate room. By utilizing existing techniques, distinguishing between malicious retouching and innocent retouching can be challenging. These limitations are effectively addressed by the proposed HTS-BESO-DA-VGG16Net-based forgery detection scheme. A diagrammatic illustration of the developed forgery detection mechanism using deep learning is shown in Figure 1.

The advanced deep learning-based forgery detection mechanism is employed to detect discrepancies between manipulated images and their originals. This method is cost-effective for identifying forgery activities, making it suitable for developing a novel approach to detect forgeries at a low cost. Initially, the required images for the detection of forgeries are collected from two different datasets. Then these images are decomposed using the SWT technique. In SWT decomposition, the parameters like start level, wavelet type, and norm are optimized with the help of the suggested HTS-BESO algorithm. After the process of decomposition, the images are split into several patches. Here, the patches are split from the decomposed images.

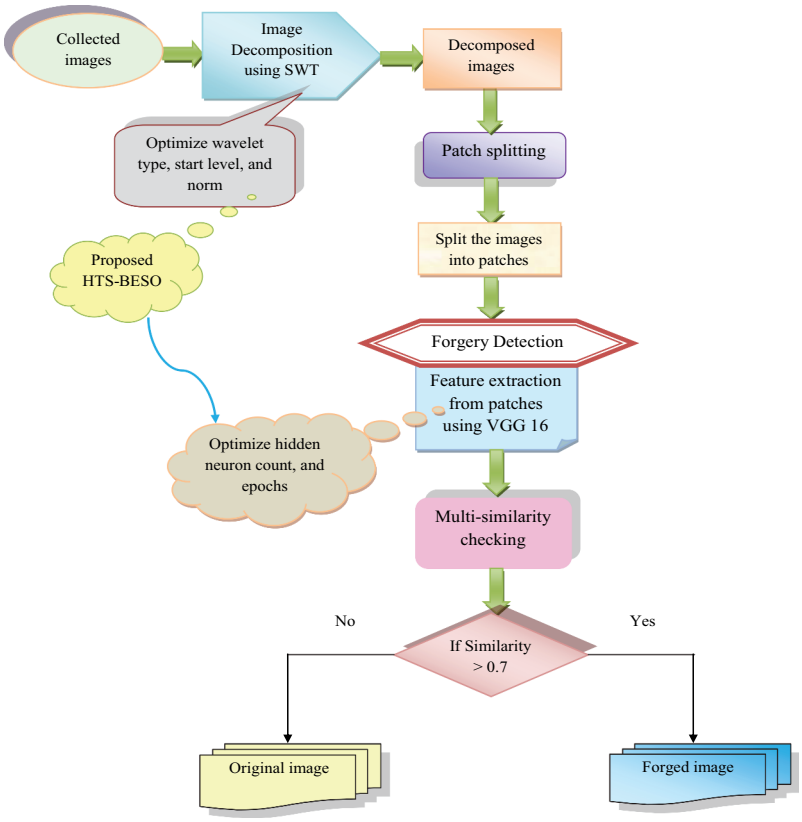

Figure 1. Diagrammatic illustration of developed forgery detection mechanism.

From the patched images, the most relevant features are retrieved by using Dilated Adaptive VGG16. The parameters from the VGG16, like hidden neuron count and epochs, are optimized using the proposed HTS-BESO algorithm to maximize the accuracy and precision values. Finally, one patch is kept constant, and the similarity value for the remaining patches is determined. If the similarity value is greater than 0.7, then the image is said to be a forged image. Finally, the final results are contrasted with the various existing algorithms to determine the effectiveness of the offered HTS-BESO-DA-VGG 16-based forgery detection scheme.

3.2. Dataset collection

The dataset 1 is taken from (Seidita et al., 2016) and also it consists of the names of the universities and surgical colleges and their departments. The section Diatics consists of several categories like doctorates, master and advanced courses, and state exams.

The dataset 2 is taken from (Tralic et al., 2013) and also it contains details about the CMFD technique. It consists of a total of 260 images, and the images are duplicated in nature. Here, the images are classified into large and small sizes. The sizes of small and large size images are $512 \times 512$and 3000x2000, respectively.

The input image collected from the online sources is indicated as In_a^{col}, where $a = 1, 2, 3 \ldots, A$ is the total number of images.

3.3. Proposed HTS-BESO

3.3.1. Purpose

The proposed HTS-BESO algorithm aims to enhance the performance of a deep learning-based forgery detection scheme by optimizing various parameters critical to its operation. Specifically, it optimizes parameters from the SWT, such as wavelet type, level, start level, and norm. Additionally, parameters from the VGG16 model, like hidden neuron count and epochs, are also optimized using this algorithm. The proposed HTS-BESO algorithm is derived from the conventional Tuna Swarm Optimization (TSO) and Bald Eagle Search Optimization (BESO).

TSO harnesses the collective intelligence seen in tuna swarms, simulating their cooperative food-searching behavior. However, achieving optimal performance with TSO necessitates precise tuning of parameters like swarm size, inertia weights, and the balance between exploration and exploitation. On the other hand, BESO is inspired by the hunting prowess of bald eagles, emphasizing strategic hunting techniques and acute vision to efficiently locate prey. Yet, effectively managing constraints such as feasibility and bounds in optimization tasks presents challenges, often requiring specialized adaptations or hybrid approaches with constraint-handling techniques.

3.3.2. Novelty

The shortcomings of the existing algorithm are overcome with the help of the proposed HTS-BESO algorithm. Because of this algorithm, the accuracy and precision values are highly maximized over the detection of forgeries. The suggested HTS-BESO algorithm is implemented with the help of updating the uniform random number based on the fitness value. Based on the updated random parameter, the best solution is obtained. The condition to be satisfied to update the position of the candidates is written in Eq. (1).

$$if \quad c < \frac{BF}{WF} \tag{1}$$

Here, c is represented as a uniform random number and the range is $[0, 1]$. The terms BF and WF are signified as best and worst fitness values, respectively. If the condition $if \quad c < \frac{BF}{WF}$ is met, TSO is updated otherwise BESO is updated. This approach aims to achieve a highly optimized solution within the problem space. By integrating this conditional update strategy, the method effectively enhances the convergence rate of the optimization system, as observed in HTS-BESO.

TSO (Xie et al., 2021): Tuna is one type of sea fish. Another name for tuna is called as Thunini. Tuna is a type of marine predator, and it uses a different technique for swimming. The tuna fish carry out two types of forging techniques.

Mathematical model: The mathematical model of the TSO is given as follows.

Initialization: Here, the optimization practice is initialized with the aid of creating the random population at free space, and it is given in Eq. (2).

$$Y_j^I = rand\,(a-b), \quad j = 1, 2, \ldots, MQ \tag{2}$$

Here, a uniformly distributed random vector is represented as *rand* and it ranges between [0, 1], *OQ* is denoted as several tuna population, upper and lower bound are indicated as *a*, *b* respectively, the initial individual in j^{th} direction is represented as Y_j^I.

Spiral foraging: If a small fish like sardines changes the direction of swimming, then tuna fish carry out spiral foraging techniques. The mathematical form of spiral foraging is given in Eq. (3).

$$Y_j^{u+1} = \begin{cases} \beta_1 \cdot \left(Y^u_{premi} + \chi \cdot |Y^u_{premi} - Y^u_i|\right) + \beta_2 \cdot\ Y^u_j, j = 1, \\ \beta_1 \cdot \left(Y^u_{premi} + \chi \cdot |Y^u_{premi} - Y^u_i|\right) + \beta_2 \cdot Y^u_{j-1},\ j = 2, 3, \ldots, MQ \end{cases} \tag{3}$$

$$\beta_1 = b + (1-b) \cdot \frac{u}{u_{\max}}, \tag{4}$$

$$\beta_2 = (1-b) - (1-b) \cdot \frac{u}{u_{\max}}, \tag{5}$$

$$\chi = e^{cm} \cdot \cos(2\pi c), \tag{6}$$

$$m = e^{3\cos((u\max+1/u)-1)\pi)}, \tag{7}$$

Here, β_1 and β_2 are signified as weight coefficients and they control the displacement of individual value, maximum iteration is represented as $u_{\max}$, the current optimal individual is indicated as Y^u_{premi}, uniform random number is indicated as *c* and its ranges in between [0, 1], j^{th} individual of the $u+1$ iteration is represented as Y_j^{u+1}.

The optimal individual does not follow the foraging group. Therefore, the orientation point is accidentally created in search space with the help of random coordinates. Because of this reference point, the foraging process is carried out more easily. The random coordinate is determined by using the following Eq. (8).

$$Y_j^{u+1} = \begin{cases} \beta_1 \cdot \left(Y^u_{rand} + \chi \cdot |Y^u_{rand} - Y^u_i|\right) \\ \quad +\beta_2 \cdot\ Y^u_j, j = 1, \\ \beta_1 \cdot \left(Y^u_{rand} + \chi \cdot |Y^u_{rand} - Y^u_i|\right) \\ +\beta_2 \cdot\ Y^u_{j-1}\ ,\ j = 2, 3, \ldots, MQ \end{cases} \tag{8}$$

Thus, the random reference point at search space is signified as Y^u_{rand}.

This algorithm is mainly used for global exploration. Later, this global exploration is changed to local exploration. So the reference point of spiral space is changed from accidental individual to best individual. The expression of spiral foraging is given in Eq. (9).

$$Y_j^{u+1} = \begin{cases} \beta_1 \cdot \left(Y^u_{rand} + \chi \cdot \left|Y^u_{rand} - Y^u_i\right|\right) & \\ +\beta_2 \cdot\ Y^u_j, j = 1, & \\ \beta_1 \cdot \left(Y^u_{rand} + \chi \cdot \left|Y^u_{rand} - Y^u_i\right|\right) & \\ +\beta_2 \cdot\ Y^u_{j-1}\ ,\ j = 2, 3, \ldots, MQ & if\ rand\ < \frac{u}{u_{\max}} \\ \beta_1 \cdot \left(Y^u_{premi} + \chi \cdot \left|Y^u_{premi} - Y^u_i\right|\right) & if\ rand\ \geq \frac{u}{u\max} \\ +\beta_2 \cdot\ Y^u_j, j = 1 & \\ \beta_1 \cdot \left(Y^u_{premi} + \chi \cdot \left|Y^u_{premi} - Y^u_i\right|\right) & \\ +\beta_2 \cdot\ Y^u_{j-1}\ ,\ j = 2, 3, \ldots, MQ & \end{cases} \tag{9}$$

Parabolic Foraging: Tunas also form a parabolic forge for their hunting process. Here, food is taken as a reference point, and also they search for food around themselves. The expression of parabolic foraging is given in Eq. (10).

$$Y_j^{u+1} = \begin{cases} Y^u_{premi} + rand \cdot (Y^u_{premi} - Y^u_j) \\ +UG \cdot q^2 \cdot (Y^u_{premi} - Y^u_j), & if\ rand < 0.5, \\ UG \cdot q^2 \cdot Y^u_j & if\ rand \geq 0.5, \end{cases} \tag{10}$$

$$q = \left(1 - \frac{u}{u_{\max}}\right)^{(u/u_{\max})} \tag{11}$$

Here, a random number is indicated as *rand*, and the value is between $[-1, 1]$.

Therefore, tuna follows parabolic foraging and spiral foraging practices for its hunting process. The first step in the optimization process is to create a random population in the search space. In every iteration process, any one of the foraging techniques is selected to create search space. This depends on the probability of x. The value x is determined using the parameter setting simulation experiments.

Bald Eagle Search Optimization (BESO) (Ferahtia et al., 2022): This algorithm is developed based on the hunting process of an eagle. This method consists of three stages. The first stage is to select the space for hunting the second stage is to select its food, and the third is called swooping.

In the swooping stage, the new position is selected by the following Eq. (12).

$$SW_{np}(j) = SW_{premi} + \delta \cdot rand \cdot (SW_m - SW(j)) \tag{12}$$

Thus, the best position is signified as SW_{premi}, the random number is represented as *rand* and its value is ranges between [0, 1], the new position is indicated as $SW_{np}(j)$, control gain is represented as δ and it ranges between [1.5, 2], best position is denoted as SW_{premi}.

The position of an eagle in the search space is adjusted by using the BESO algorithm, and it is done after the selection of the best position SW_{premi}. The updated model is given in Eq. (13).

$$SW_{np}(j) = SW(j) + o(j).(SW(j) - SW(j+1)) + n(j).(SW(j) - SW_m) \tag{13}$$

Here, directional coordinates for j^{th} position is signified as

o, m and they are described in Eq. (14).

$$\begin{aligned} n(j) &= \frac{n\,rand(j)}{\max(|n\,rand|)};\ nrand(j) \\ &= rand(j).\ \sin(\zeta(j)) \end{aligned}$$

$$\begin{aligned} o(j) &= \frac{o\,rand(j)}{\max(|o\,rand|)};\ o\,rand(j) \\ &= rand(j).\ \cos(\zeta(j)) \end{aligned} \tag{14}$$

$$\zeta(j) = b.\pi.rand;\ rand(j) = \zeta(j).S.rand$$

Here, the number of search cycles is represented as S and its value ranges between [0.5, 2], the control parameter is denoted as b and it is used to determine the curve between the points of search and it ranges between [5, 10]

In this stage, the eagle targets its food from the best place. The hunting model is given in Eq. (15).

$$SW_{np}(j) = rand \cdot SW_{premi} + n1(j) \cdot (SW(j) - d1 \cdot SW_m) + o1(j) \cdot (SW(j) - d2 \cdot SW_{premi}) \tag{15}$$

Here, random number is represented as $d1$ and $d2$, respectively, and directional coordinates are signified as $n1$ and $o1$, respectively. Finally, the values of directional coordinates are determined using Eq. (16).

$$\begin{aligned} n1\,(j) &= \frac{n\,rand(j)}{\max(|n\,rand|)};\ nrand(j) \\ &= rand(j).\ \sin\ i\,(\zeta(j)) \end{aligned}$$

$$\begin{aligned} o1\,(j) &= \frac{o\,rand(j)}{\max(|o\,rand|)};o\,rand(j) \\ &= rand(j).\ \cos\ i\,(\zeta(j)) \end{aligned} \tag{16}$$

$$\zeta(j) = b.\pi.rand;\ rand(j) = \zeta(j)$$

Here, directional coordinates are represented as m and o, respectively; also the term b is signified as a control parameter. The algorithm for the Proposed HTS-BESO is given in Algorithm 1.

Algorithm 1: Proposed HTS-BESO

Initialize the value of the number of population $Npop$
Initialize maximum iteration value $u_{\max}$
Find the value of BF and TF then
Find the value of the uniform random number c
For $t = 1$ to $u_{\max}$
For $j = 1$ to $Npop$
If *if* $c <$ (BF/WF)
Find the value of Y_{rand}^{u}
Update the position using TSO using Eq. (10)
Else
Find the value of the best position
Update the position using BESO by Eq. (13)
End if
End for
End for

4. Image decomposition and patch splitting using hybridized tuna swarm with bald eagle search optimization

4.1. Stationary wavelet transform

In this technique, images collected from the dataset In_a^{col} are given as input. SWT (Nason & Silverman, 1995) is the sub-section of Dynamic Wavelet Transform (DWT). However, DWT has certain disadvantages. SWT has high redundancy, so it will maintain the dyadic value of the sample images. SWT is used to find the data fusion in images, and also it is used for the edge detection process. However, these operations are not to be performed in DWT. The expression for transformation in SWT is given in Eq. (17).

$$d_{k,l} = \sum_{m \in X} y(m)\omega'_{k,l}(m) \tag{17}$$

Here, a discrete wavelet is represented as $\omega'_{k,l}(m)$, and its value is determined using Eq. (18).

$$\omega'_{k,l}(m) = 2^{-(k/2)}\omega_{0,0}(2^{-k}(m-l)), \tag{18}$$

$$db_{1.l}(m) = \sum\nolimits^{i} 1(m-v)y(v)$$

$$de_{1.l}(m) = \sum\nolimits^{h} 1(m-v)y(v) \tag{19}$$

The above equations are generalized and written in Eq. (20).

$$db_{k,l}(m) = \left[\uparrow 2^{k-1}[i_1] * db_{k-1.l}\right] = \sum\nolimits^{i^k} (m-v)db_{k-1,l}(v)$$

$$de_{k,l}(m) = \left[\uparrow 2^{k-1}[h_1] * de_{k-1.l}\right](l) = \sum\nolimits^{h^k} (m-v)be_{k-1,l}(v) \tag{20}$$

Here, $db_{k,l}$ and $de_{k,l}$ are the approximate coefficient and detailed coefficients, respectively. These coefficients are created with the help of signal sequence, and it is represented as $y(m)$, i_1 and j_1 is the adaptive size of the high and low pass filter, respectively, oversampling of high pass and low pass filters at coefficient values of $i^{k-1}(m)$ and $h^{k-1}(m)$ is indicated as $\uparrow 2^{k-1}[i_1] = i^k(m)$ and $\uparrow 2^{k-1}[h_1] = h^k(m)$ the expression is used to find the values and it is written in Eq. (21).

$$\begin{cases} h^j(2m) = h^{j-1}(m) \\ h^j(2m+1) = 0 \end{cases} \quad \begin{cases} i^j(2m) = i^{j-1}(m) \\ i^j(2m+1) = 0 \end{cases} \tag{21}$$

The above equation gives very accurate output signals by utilizing the DWT method. The output obtained from this process is denoted as D_b^{SWT}.

4.2. Adaptive SWT-based image decomposition

Here, the SWT decomposition technique is used to decompose the input image into many portions. Because of this decomposition, the images are easily partitioned based on the fixed kernel size value. Here, parameters like the norm, start level, level, and wavelet type are optimized using the newly developed HTS-BESO to get high accuracy and precision values over the detection of forgeries from the original images. This adaptive SWT-based wavelet decomposition preserves the fine details of the image that may help to extract the best discriminative features from the decomposed images. Therefore, the detection reliability of the developed model is highly improved using the Adaptive SWT-based decomposition. The illustration of adaptive SWT-based image decomposition is shown in Figure 2.

4.3. Patch splitting

Here, the decomposed image D_b^{SWT} is given as input. Patch means rectangular-shaped small box. It is composed of several pixels. Because of this

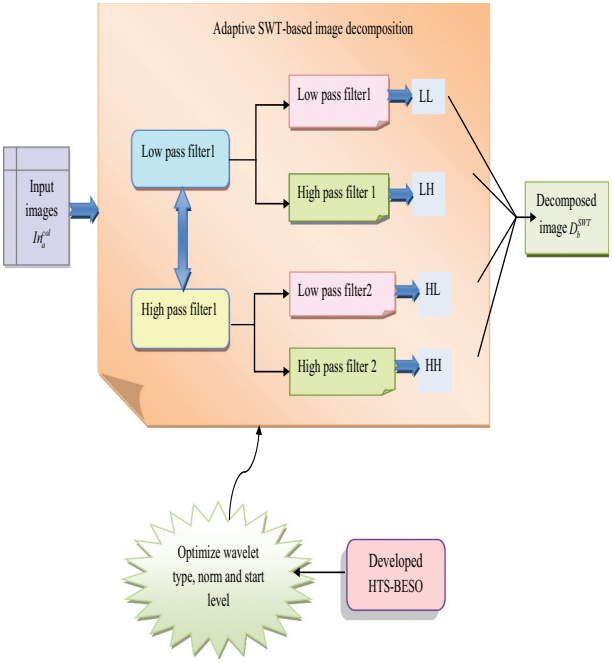

Figure 2. Illustration of adaptive SWT-based image decomposition.

technique, the image processing algorithms are easily carried in the patched images. This method reduces the complexities in image processing. In this technique, the first process is to mark the seed points. After that, find the movement of this point using the small gradient. To divide the image into small patches, then we have to increase the number of seed points. In the developed forgery detection model, the decomposed images are split into nine patches. This patch splitting helps to identify the similarity values, which is most helpful in detecting the forged images with high accuracy. Finally, the split patches are denoted as P_c^{sp}.

5. Dilated adaptive VGG16 network for detecting forgeries and multi-similarity checking for localization

5.1. Dilated VGG16

5.1.1. VGG 16 (Tammina, 2019)

Here, the split patch image P_c^{sp} is given as input. This method is used for the huge image recognition process. VGG 16 is one type of VGG model. It consists of 16 layers, namely convolutional and fully connected layers. Here, each layer is stacked with a convolutional layer on the top side. Therefore, the process will easily be carried out. In this structure, the initial and second layers are made with the help of a kernel filter of size 3×3. The first step in this process is to send the input signal to the first two convolutional layers. After the application of this input signal, the size of the signal is changed. After that, the output signals are sent to the next layer, and its name is called as max pooling layer. These layers consist of the third and fourth convolution layers made up of kernel filters of size 3×3. After the second stage, the size of the output signal was reduced to the previous one. Furthermore, the signal was sent to the next layer. These layers also consist of the convolutional layer, which is made up of a kernel filter. The final layers are connected with hidden layers. Finally, the output of the sixteenth layer was obtained and it has 1000 unit values.

Conventional VGG 16 consumes more devices for computation. To overcome this, dilated adaptive VGG 16 is used. Dilation is performed by replacing the normal VGG 16 method. The dilated models are formed by replacing the dilated convolution kernel instead of the conventional kernel. Dilated VGG 16 consists of a convolution layer of number 2, and it also consists of several connected layers. This method easily processes the image without any complexity. Therefore, we can achieve the best possible results. Here, convolution kernels are replaced with dilated convolution kernels for easy operation. Dilated convolutions are used to store more input images, and they do not require pooling for this operation, and it does not change the dimension of the input image. The characteristics of the input image are extracted using VGG 16.

5.2. Proposed dilated adaptive VGG16-based forgery detection

The forged image detection is done with the help of the developed DA-VGG16Net, where the parameter optimization is carried out to enhance the detection efficiency. By using this network, the most relevant features are effectively retrieved from the nine patches. These important features are supportive of the discovery of forgeries with high accuracy. Here, the parameters like hidden neuron count and the number of epochs are optimized with the assistance of the proposed HTS-BESO. Therefore, the accuracy and precision values are highly maximized. Hence, the overall effectiveness of the system is also

increased. The objective function of the proposed deep learning-based forgery detection scheme is signified as DF_1 and is written in Eq. (22).

$$DF_1 = \underset{\left\{W_T^{SWT}, L_l^{SWT}, S_L^{SWT}, N_n^{SWT}, H_N^{VGG16}, E_e^{VGG16}\right\}}{\arg\min} \left(\frac{1}{A+P}\right) \tag{22}$$

Thus, accuracy and precision is represented as A, and P, respectively, $W_T^{SWT}, L_l^{SWT}, S_L^{SWT}, N_n^{SWT}$ are the optimized parameters of SWT, and they are wavelet type and it is ranges between $[0, 4]$, level and its value is ranges between $[5, 15]$, start level and the value is ranges between $[1, 4]$, norm and its value is ranges between $[0, 1]$. And H_N^{VGG16}, E_e^{VGG16} are the optimized parameters of VGG 16 and they are hidden neuron count; its value ranges between $[5, 255]$ and epochs and its value ranges between $[5, 50]$ respectively. The value of accuracy and precision is determined by using Eq. (23) and Eq. (24).

$$A = \frac{(a+b)}{a+b+x+y} \tag{23}$$

$$P = \frac{b}{a+x} \tag{24}$$

Here, true positives and false positives are signified as a and x respectively, b and y is represented by true and false negatives respectively. The diagrammatic representation of dilated adaptive VGG-16-based forgery detection is represented in Figure 3.

5.3. Multi-similarity-based image localization

It is the process of locating similar objects in images. This process is done after the splitting of patches. Multi-similarity checking is done for all selected patches. Here, one patch is kept constant, and the remaining patches are compared with the constant patch. Finally, this value is compared with the condition. If the similarity value exceeds 0.7, the image is classified as a forged image; otherwise, it is classified as an original image. A graphical representation of multi-similarity-based image localization is given in Figure 4.

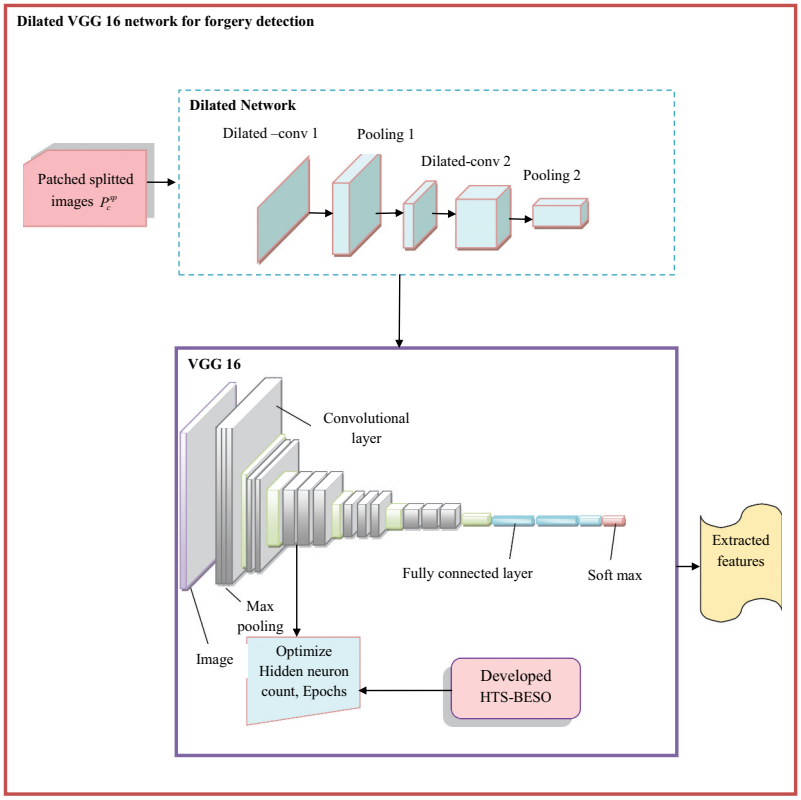

Figure 3. Diagrammatic representation of dilated adaptive VGG16-based forgery detection.

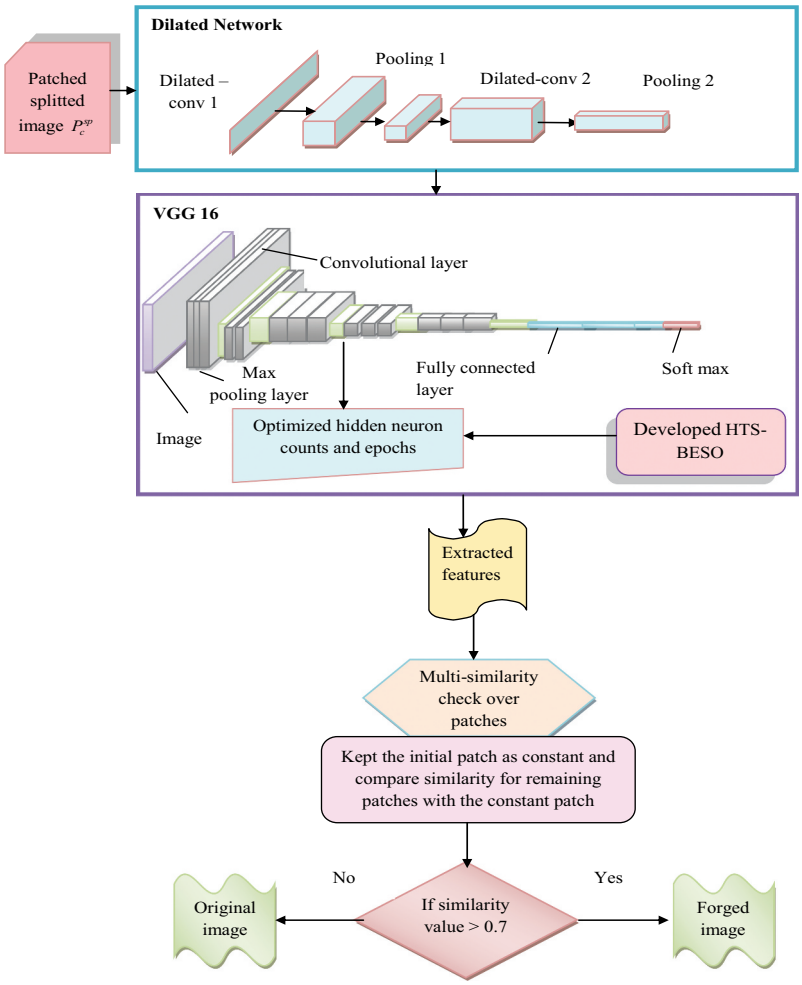

Figure 4. Graphical representation of multi-similarity-based image localization.

Images	Result images
	Dataset 1
Forged images	
Proposed HTS-BESO-DA-VGG16 based forged image	
	Dataset 2
Forged image	
Proposed HTS-BESO-DA-VGG16 based forged image	

Figure 5. Output images obtained from HTS-BESO-DA-VGG16-based forgery detection scheme.

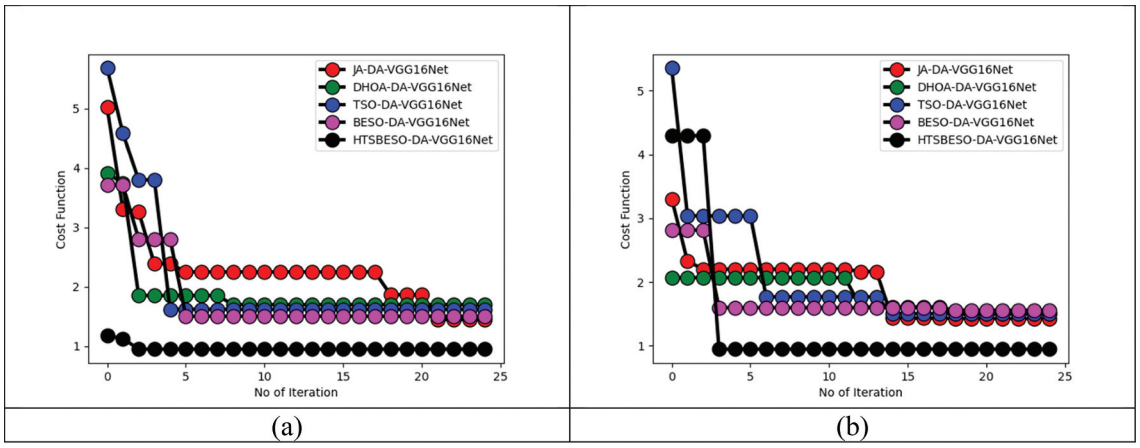

Figure 6. Performance estimation of the proposed deep networks-based forgery detection scheme through the existing algorithms in terms of convergence analysis for (a) dataset 1 and (b) dataset 2.

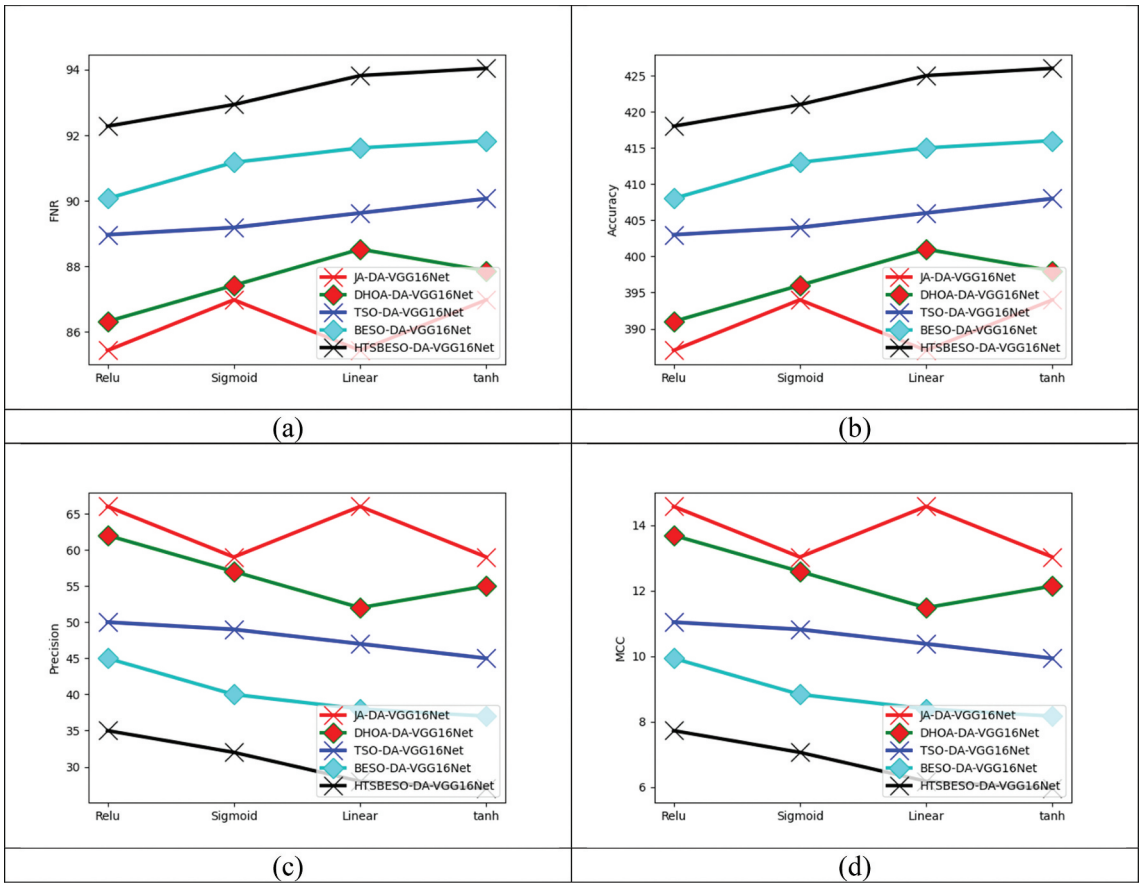

Figure 7. Performance estimation of the proposed deep networks-based forgery detection scheme through the existing algorithms based on (a) FNR, (b) accuracy, (c) precision, (d) MCC.

6. Results and Discussions

6.1. Experimental setup

The suggested HTS-BESO-DA-VGG16Net-based forgery detection scheme was implemented with the help of Python. The result of the offered method was contrasted with the existing algorithm. The analysis of this method was carried out with the help of the following parameters. Here, the number of population was taken as 10. The value of maximum iteration and chromosome length was taken as 50 and 4, respectively.

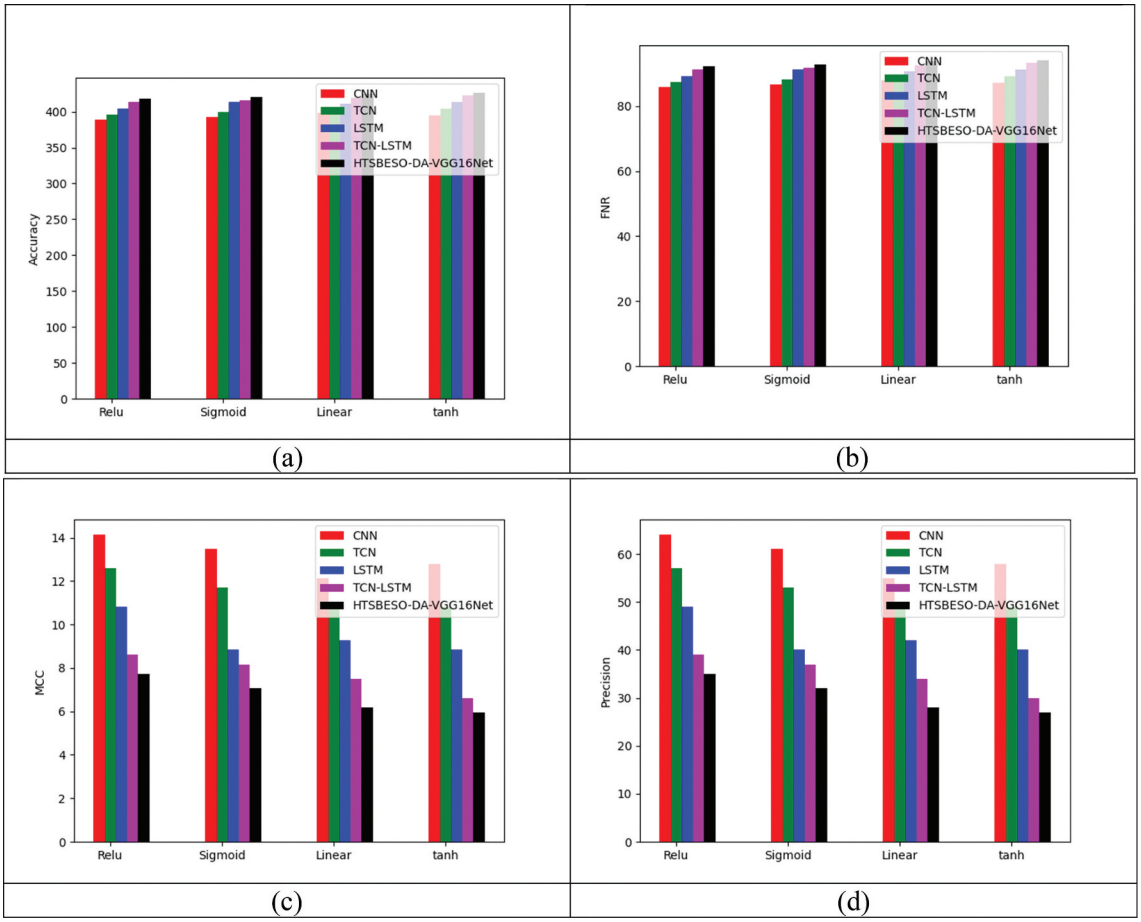

Figure 8. Performance estimation of the proposed deep network-based forgery detection scheme through the existing technique in terms of (a) accuracy (b) FNR, (c) MCC, (d) precision.

The existing algorithms Jaya Algorithm (JA) (Zitar et al., 2022), Deer Hunting Optimization Algorithm (DHOA) (Brammya et al., 2019), TSO (Xie et al., 2021), and BESO (Ferahtia et al., 2022) were used to analyze the efficiency of the offered method.

6.2. Result images

The following Fig. 5 consists of tamper images and the ground truth images from the collected dataset.

6.3. Performance metrics

The comparative examination is done with the help of performance metrics, and they are listed below.

Specificity: It is calculated by Eq. (25).

$$V_{spci} = \frac{b}{b+y} \tag{25}$$

NPV: The value of NPV is determined using Eq. (26).

$$V_{NPV} = \frac{b}{b+x} \tag{26}$$

Sensitivity: Sensitivity is determined by using the following Eq. (27).

$$V_{sen} = \frac{a}{a+x} \tag{27}$$

FPR: The following Eq. (28) is used to determine the value of FPR.

$$V_{FPR} = \frac{x}{x+b} \tag{28}$$

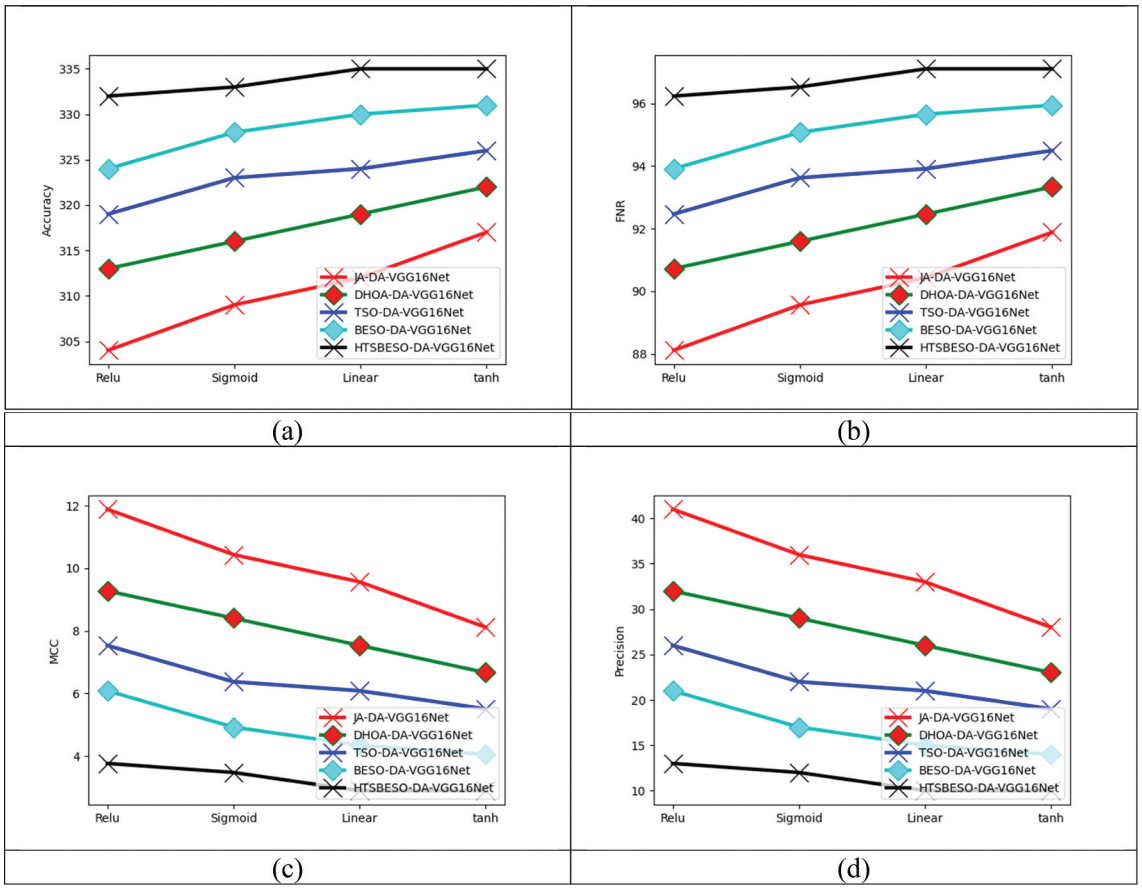

Figure 9. Performance estimation of the proposed deep network-based forgery detection scheme through the existing algorithms concerning (a) accuracy, (b) FNR, (c) MCC, (d) precision.

FDR: The value of FDR is determined using Eq. (29).

$$V_{FDR} = \frac{y}{y + a} \tag{29}$$

FNR: The FNR is determined using Eq. (30).

$$V_{FNR} = \frac{x}{x + a} \tag{30}$$

MCC: The value of MCC is determined by using Eq. (31).

$$V_{MCC} = \frac{a \times b - x \times y}{\sqrt{(a + y)(a + x)(b + y)(b + x)}} \tag{31}$$

6.4. Performance assessment on convergence

Fig. 6 shows the performance evaluation of the suggested HTS-BESO-DA-VGG16Net-based forgery detection scheme in terms of convergence analysis. The analysis results show that, at the 15th iteration, the cost function of the proposed deep networks-based forgery detection system converges 40% more than JA-DA-VGG16Net, 50% more than TSO-DA-VGG16Net, 60% more than BESO-DA-VGG16Net and hence it provides more accuracy and precision. From the analysis, the output shows that the convergence rate of the proposed method is increased when compared to the existing algorithms.

6.5. Performance assessment on data set 1

The following Figures 7 and 8 show the performance evaluation of the suggested HTS-BESO-DA-VGG16Net-based forgery detection scheme. This system is compared with various techniques and existing algorithms. The comparison results show that all the existing algorithms do not give proper results when finding the forgery detection. Here, the linear stage is taken in the x-axis. The linear, sigmoid, tanh, and Relu activation functions

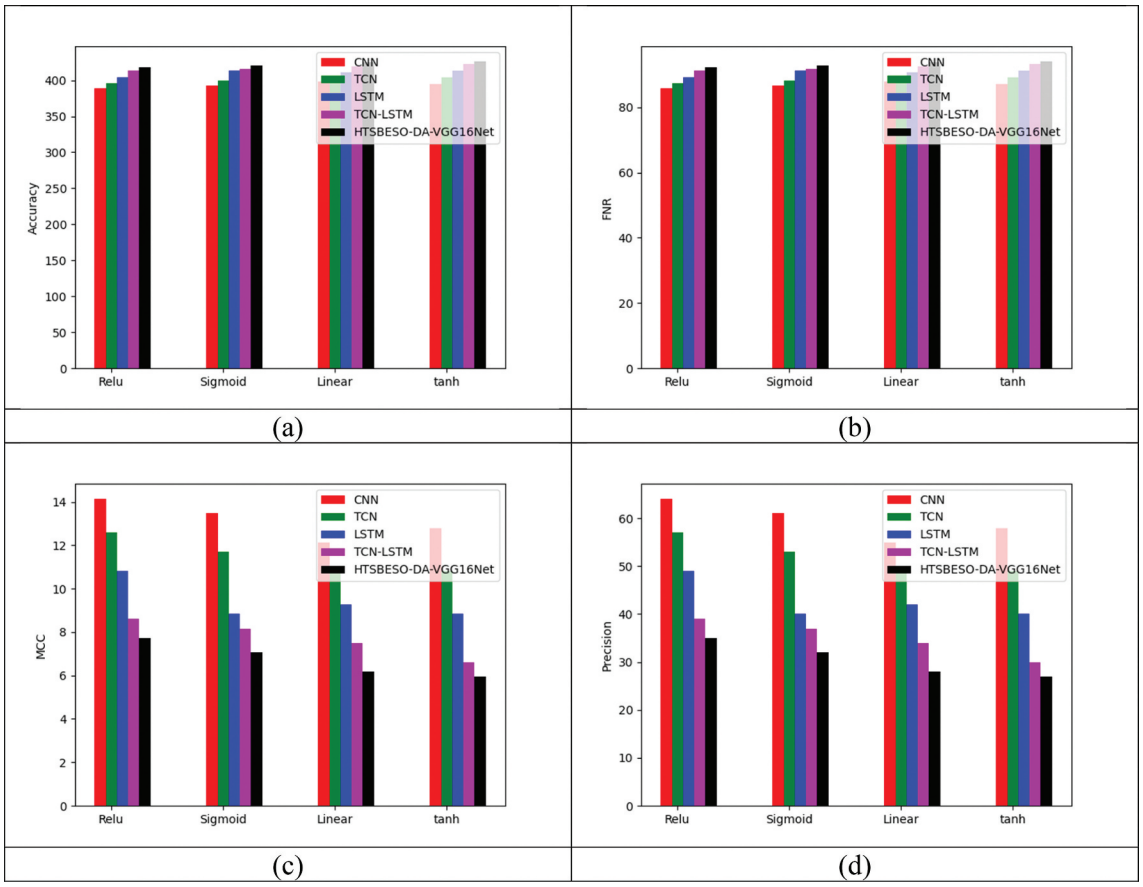

Figure 10. Performance estimation of the proposed deep network-based optimization algorithm through the existing techniques concerning (a) accuracy, (b) FNR, (c) MCC, (d) precision.

Table 2. Statistical analysis of the developed forgery detection model among existing algorithms.

Dataset 1					
TERMS	JA Zitar et al. (2022)	DHOA Brammya et al. (2019)	TSO Xie et al. (2021)	BESO Ferahtia et al. (2022)	HTS-BESO
Worst	5.025627	3.920323	5.684171	3.71733	1.176679
Best	1.451256	1.702118	1.624979	1.507752	0.951784
Mean	2.282944	1.909305	2.080122	1.840696	0.968015
Median	2.253147	1.702118	1.624979	1.507752	0.951784
Std	0.716849	0.57129	1.087437	0.694974	0.05539
Dataset 2					
Worst	3.306382	2.069596	5.353756	2.817412	4.294403
Best	1.428268	1.499351	1.505756	1.561018	0.9479
Mean	1.908893	1.799575	2.050778	1.734979	1.34948
Median	2.165806	1.609794	1.769636	1.599674	0.9479
Std	0.475451	0.262449	0.887536	0.400072	1.087485

are considered for the analysis of performance, by using these functions, the performance of the suggested model is visualized very clearly. Here, the performance of the proposed HTS-BESO-DA-VGG16Net-based forgery detection scheme is better than the JA-DA-VGG 16 Net, DHOA-DA-VGG 16 Net, TSO-DA-VGG 16 Net and BESO-DA-VGG 16 Net with the percentage value of 9.60%, 6.36%, 5.40%, and 2.87% at the linear stage. Therefore, the output shows that the proposed method achieves a high FNR rate.

6.6. Performance assessment on data set 2

The following Figures 9 and 10 show the performance assessment of the suggested HTS-BESO-DA -VGG16Net-based forgery detection scheme along with various existing techniques and algorithms.

Table 3. Overall performance of the developed HTS-BESO-DA-VGG16-based forgery detection scheme among various algorithms and classifiers.

Dataset 1					
Algorithm Comparison					
TERMS	JA Zitar et al. (2022)	DHOA Brammya et al. (2019)	TSO Xie et al. (2021)	BESO Ferahtia et al. (2022)	HTS-BESO
Accuracy	87.28261	88.58696	90.54348	92.5	94.02174
Sensitivity	86.9936	88.69936	90.1919	92.32409	94.02985
Specificity	87.58315	88.47007	90.90909	92.68293	94.0133
Precision	87.93103	88.88889	91.16379	92.91845	94.23077
FPR	12.41685	11.52993	9.090909	7.317073	5.986696
FNR	13.0064	11.30064	9.808102	7.675906	5.970149
NPV	86.62281	88.27434	89.91228	92.07048	93.80531
FDR	12.06897	11.11111	8.836207	7.081545	5.769231
F1-Score	87.45981	88.79402	90.67524	92.62032	94.1302
MCC	0.745653	0.771663	0.810885	0.84998	0.880396
Classifier Comparison					
TERMS	CNN Abdalla et al. (2019)	LSTM Bappy et al. (2019)	TCN Hewage et al. (2020)	TCN-LSTM Bappy et al. (2019)	HTS-BESO
Accuracy	87.93478	89.13043	91.19565	93.36957	94.02174
Sensitivity	88.0597	88.91258	91.04478	93.60341	94.02985
Specificity	87.80488	89.35698	91.35255	93.12639	94.0133
Precision	88.24786	89.67742	91.6309	93.40426	94.23077
FPR	12.19512	10.64302	8.64745	6.873614	5.986696
FNR	11.9403	11.08742	8.955224	6.396588	5.970149
NPV	87.61062	88.57143	90.7489	93.33333	93.80531
FDR	11.75214	10.32258	8.369099	6.595745	5.769231
F1-Score	88.15368	89.29336	91.3369	93.50373	94.1302
MCC	0.758615	0.782592	0.823886	0.867337	0.880396
Dataset 2					
Algorithm Comparison					
TERMS	JA Zitar et al. (2022)	DHOA Brammya et al. (2019)	TSO Xie et al. (2021)	BESO Ferahtia et al. (2022)	HTS-BESO
Accuracy	91.58621	93.24138	94.62069	96.27586	97.37931
Sensitivity	91.57303	93.25843	94.66292	96.34831	97.47191
Specificity	91.59892	93.22493	94.57995	96.20596	97.28997
Precision	91.31653	92.9972	94.39776	96.07843	97.19888
FPR	8.401084	6.775068	5.420054	3.794038	2.710027
FNR	8.426966	6.741573	5.337079	3.651685	2.52809
NPV	91.84783	93.47826	94.83696	96.46739	97.55435
FDR	8.683473	7.002801	5.602241	3.921569	2.80112
F1-Score	91.4446	93.12763	94.53015	96.21318	97.3352
MCC	0.831682	0.864794	0.892388	0.9255	0.947576
Classifier Comparison					
TERMS	CNN Abdalla et al. (2019)	LSTM Bappy et al. (2019)	TCN Hewage et al. (2020)	TCN-LSTM Bappy et al. (2019)	HTS-BESO
Accuracy	91.72414	92.96552	94.2069	95.31034	97.37931
Sensitivity	91.57303	92.69663	94.10112	95.22472	97.47191
Specificity	91.86992	93.22493	94.30894	95.39295	97.28997
Precision	91.57303	92.95775	94.10112	95.22472	97.19888
FPR	8.130081	6.775068	5.691057	4.607046	2.710027
FNR	8.426966	7.303371	5.898876	4.775281	2.52809
NPV	91.86992	92.97297	94.30894	95.39295	97.55435
FDR	8.426966	7.042254	5.898876	4.775281	2.80112
F1-Score	91.57303	92.827	94.10112	95.22472	97.3352
MCC	0.83443	0.859261	0.884101	0.906177	0.947576

For sigmoid function, the precision of the proposed deep networks-based forgery detection system is 35.2% more than JA-DA-VGG16Net, 28.5% more than TSO-DA-VGG16Net, 14.34% more than BESO-DA-VGG16Net. From the analysis, the output shows that the convergence rate of the proposed method is increased when compared to the existing algorithms.

6.7. Statistical analysis of the proposed model

A statistical examination of the proposed HTS-BESO-DA-VGG16Net-based forgery detection scheme is given in Table 2. Here, the best, mean, median, and standard deviation measures are calculated to describe the performance of the developed forgery detection model. The analysis

outcome shows that the proposed HTS-BESO-DA-VGG16Net-based forgery detection scheme achieves the best value of 52.34%, 78.61%, 70.20%, and 57.59% than JA, DHOA, TSO, and BESO. Therefore, the recommended scheme gives superior performance in the statistical examination when compared to the presented algorithms.

6.8. Overall examination of the suggested model

The overall examination of the planned HTS-BESO-DA-VGG16Net-based forgery recognition proposal is given in Table 3. The accuracy and precision values are used to find out the fitness value. Fitness is the main parameter and it is used to value the value of a random number. The proposed HTS-BESO-DA-VGG16Net-based forgery detection scheme obtained a sensitivity value of 8.08% than JA, 6% than DHOA, 4.24% than TSO, and 1.84% than BESO. Therefore, the analysis results showed that the proposed HTS-BESO-DA-VGG16Net-based forgery detection scheme attained better sensitivity than the presented algorithms.

7. Conclusion

The developed deep network-based forgery detection scheme was implemented to determine the forgery activities in images. Initially, the desired images were collected from various online sources. With these collected images, an SWT was applied, where the parameters of the SWT were optimized using a developed HTS-BESO. By utilizing this adaptive SWT, the entire image was split into patches for each subband. Subsequently, the DA-VGG16Net framework was employed to extract deep features from the split patches. The parameters of the DA-VGG16Net were optimized using the same HTS-BESO. Finally, feature matching was performed using multi-similarity checking for the recognition and localization of forgeries. The experimental results were compared to various existing forgery detection approaches to ensure the efficiency of the developed model by considering various performance measures. The proposed HTS-BESO-based forgery detection technique obtained a sensitivity value of 8.08% than JA, 6% than DHOA, 4.24% than TSO, and 1.84% than BESO. Therefore, the analysis shows that the HTS-BESO-based forgery detection scheme achieves maximized accuracy and precision value in comparison to the existing algorithms. The model is primarily designed for image-based forgeries. Its performance and adaptability to other media types, such as videos or audio, are not addressed. Extending the model to handle multi-modal content includes videos, audio, and text, to detect forgeries across different types of media. This would enhance the model's versatility and applicability in a wider range of digital forensic investigations.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Dr. Prabhu Bevinamarad is working as assistant professor in the Department of Computer Science and Engineering, B.L.D.E.A's V.P.Dr.P.G. Halakatti College of Engineering and Technology, Vijayapura, Karnataka 586103, India from last 15 years, his expertise include image processing, video processing, natural language processing.

Dr. Prakash Unki is working professor and head in the Department of information Science and Engineering, B.L.D.E.A's V.P.Dr.P.G. Halakatti College of Engineering and Technology, Vijayapura, Karnataka 586103,India from last 25 years, his expertise include image processing, video processing, natural language processing.

References

Abdalla, Y., Iqbal, M. T., & Shehata, M. (2019). Copy-move forgery detection and localization using a generative adversarial network and convolutional neural network. *Information*, *10*(9), 286. https://doi.org/10.3390/info10090286

Alipour, N., & Behrad, A. (2020). Semantic segmentation of JPEG blocks using a deep CNN for non-aligned JPEG forgery detection and localization. *Multimedia Tools & Applications*, *79*(11–12), 8249–8265. https://doi.org/10.1007/s11042-019-08597-8

Aloraini, M., Sharifzadeh, M., & Schonfeld, D. (2021). Sequential and patch analyses for object removal video forgery detection and localization. *IEEE Transactions on Circuits and Systems for Video Technology*, *31*(3), 917–930. https://doi.org/10.1109/TCSVT.2020.2993004

Bappy, J. H., Simons, C., Nataraj, L., Manjunath, B. S., & Roy-Chowdhury, A. K. (2019). Hybrid LSTM and encoder-decoder architecture for detection of image forgeries. *IEEE Trans-*

actions on Image Processing, *28*(7), 3286–3300. https://doi.org/10.1109/TIP.2019.2895466

Barni, M., Phan, Q. T., & Tondi, B. (2021). Copy move source-target disambiguation through multi-branch CNNs. *IEEE Transactions on Information Forensics and Security*, *16*, 1825–1840. https://doi.org/10.1109/TIFS.2020.3045903

Bilal, M., Habib, H. A., 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. *Arabian Journal for Science & Engineering*, *45*(4), 2975–2992. https://doi.org/10.1007/s13369-019-04238-2

Brammya, G., Praveena, S., Ninu Preetha, N. S., Ramya, R., Rajakumar, B. R., & Binu, D. (2019). Deer hunting optimization algorithm: A new nature-inspired meta-heuristic paradigm. *The Computer Journal*. https://doi.org/10.1093/comjnl/bxy133

Cristin, R., & Cyril Raj, V. (2017). Consistency features and fuzzy-based segmentation for shadow and reflection detection in digital image forgery. *Science China Information Sciences*, *60*(8), 1–18. https://doi.org/10.1007/s11432-016-0478-y

Dua, S., Singh, J., & Parthasarathy, H. (2020). Detection and localization of forgery using statistics of DCT and Fourier components. *Signal Processing: Image Communication*, *82*, 115778. https://doi.org/10.1016/j.image.2020.115778

Ferahtia, S., Rezk, H., Abdelkareem, M. A., & Olabi, A. G. (2022). Optimal techno-economic energy management strategy for building's microgrids based bald eagle search optimization algorithm. *Applied Energy*, *306*, 118069. https://doi.org/10.1016/j.apenergy.2021.118069

Ganeshan, R., Muppidi, S., Thirupurasundari, D. R., & Kumar, B. S. (2022). Autoregressive-elephant herding optimization based generative adversarial network for copy-move forgery detection with interval type-2 fuzzy clustering. *Signal Processing: Image Communication*, *108*, 116756. https://doi.org/10.1016/j.image.2022.116756

Gu, A. R., Nam, J. H., & Lee, S. C. (2022). FBI-Net: Frequency-based image forgery localization via multitask learning with self-attention. *IEEE Access*, *10*, 62751–62762. https://doi.org/10.1109/ACCESS.2022.3182024

Hewage, P., Behera, A., Trovati, M., Pereira, E., Ghahremani, M., Palmieri, F., & Liu, Y. (2020). Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station. *Soft Computing*, *24*(21), 16453–16482. https://doi.org/10.1007/s00500-020-04954-0

Huang, Y., Juefei-Xu, F., Guo, Q., Liu, Y., & Pu, G. (2022). Fakelocator: Robust localization of GAN-based face manipulations. *IEEE Transactions on Information Forensics and Security*, *17*, 2657–2672. https://doi.org/10.1109/TIFS.2022.3141262

Kadam, K., Ahirrao, S., Kotecha, K., & Sahu, S. (2021). Detection and localization of multiple image splicing using MobileNet V1. *IEEE Access*, *9*, 162499–162519. https://doi.org/10.1109/ACCESS.2021.3130342

Koul, S., Kumar, M., Khurana, S. S., Mushtaq, F., & Kumar, K. (2022). An efficient approach for copy-move image forgery detection using convolution neural network. *Multimedia Tools & Applications*, *81*(8), 11259–11277. https://doi.org/10.1007/s11042-022-11974-5

Lee, S. I., Park, J. Y., & Eom, I. K. (2022). Cnn-based copy-move forgery detection using rotation-invariant wavelet feature. *IEEE Access*, *10*, 106217–106229. https://doi.org/10.1109/ACCESS.2022.3212069

Li, H., Luo, W., Qiu, X., & Huang, J. (2017). Image forgery localization via integrating tampering possibility maps. *IEEE Transactions on Information Forensics and Security*, *12*(5), 1240–1252. https://doi.org/10.1109/TIFS.2017.2656823

Lin, X., & Li, C. T. (2020). Prnu-based content forgery localization augmented with image segmentation. *IEEE Access*, *8*, 222645–222659. https://doi.org/10.1109/ACCESS.2020.3042780

Mahmood, T., Mehmood, Z., Shah, M., & Saba, T. (2018). A robust technique for copy-move forgery detection and localization in digital images via stationary wavelet and discrete cosine transform. *Journal of Visual Communication and Image Representation*, *53*, 202–214. https://doi.org/10.1016/j.jvcir.2018.03.015

Meena, K. B., & Tyagi, V. (2020). A hybrid copy-move image forgery detection technique based on Fourier-Mellin and scale invariant feature transforms. *Multimedia Tools & Applications*, *79*(11–12), 8197–8212. https://doi.org/10.1007/s11042-019-08343-0

Nason, G. P., & Silverman, B. W. (1995). The stationary wavelet transform and some statistical applications. *Wavelets and Statistics, 103*, 281–299. https://doi.org/10.1007/978-1-4612-2544-7_17.

Niu, Y., Tondi, B., Zhao, Y., Ni, R., & Barni, M. (2021). Image splicing detection, localization and attribution via JPEG primary quantization matrix estimation and clustering. *IEEE Transactions on Information Forensics and Security*, *16*, 5397–5412. https://doi.org/10.1109/TIFS.2021.3129654

Rao, Y., Ni, J., & Zhao, H. (2020). Deep learning local descriptor for image splicing detection and localization. *IEEE Access*, *8*, 25611–25625. https://doi.org/10.1109/ACCESS.2020.2970735

Seidita, V., Chella, A., & Carta, M. (2016). A biologically inspired representation of the intelligence of a university campus. *Procedia Computer Science*, *88*, 185–190. https://doi.org/10.1016/j.procs.2016.07.423

Tammina, S. (2019). Transfer learning using VGG-16 with deep convolutional neural network for classifying images. *International Journal of Scientific & Research Publications (IJSRP)*, *9*(10), 143–150. https://doi.org/10.29322/IJSRP.9.10.2019.p9420

Tralic, D., Zupancic, I., Grgic, S., & Grgic, M. (2013). [1] CoMoFoD - new database for copy-move forgery detection. *Proceedings ELMAR-2013*.

Vinolin, V., & Sucharitha, M. (2021). Dual adaptive deep convolutional neural network for video forgery detection

in 3D lighting environment. *The Visual Computer*, *37*, 2369–2390. https://doi.org/10.1007/s00371-020-01992-5.

Xie, L., Han, T., Zhou, H., Zhang, Z. R., Han, B., & Tang, A. (2021). Tuna swarm optimization: A novel swarm-based metaheuristic algorithm for global optimization. *Computational Intelligence and Neuroscience*, *2021*(1), 1–22. https://doi.org/10.1155/2021/9210050

Yan, C. P., & Pun, C. M. (2017). Multi-scale difference map fusion for tamper localization using binary ranking hashing. *IEEE Transactions on Information Forensics and Security*, *12*(9), 2144–2158. https://doi.org/10.1109/TIFS.2017.2699942

Yang, Q., Yu, D., Zhang, Z., Yao, Y., & Chen, L. (2021). Spatiotemporal trident networks: Detection and localization of object removal tampering in video passive forensics. *IEEE Transactions on Circuits and Systems for Video Technology*, *31*(10), 4131–4144. https://doi.org/10.1109/TCSVT.2020.3046240

Zhou, G., Tian, X., & Zhou, A. (2022). Image copy-move forgery passive detection based on improved PCNN and self-selected sub-images. *Frontiers of Computer Science*, *16*(4), 1–16. https://doi.org/10.1007/s11704-021-0450-5

Zhuo, L., Tan, S., Li, B., & Huang, J. (2022). Self-adversarial training incorporating forgery attention for image forgery localization. *IEEE Transactions on Information Forensics and Security*, *17*, 819–834. https://doi.org/10.1109/TIFS.2022.3152362

Zitar, R. A., Al-Betar, M. A., Awadallah, M. A., Doush, I. A., & Assaleh, K. (2022). An intensive and comprehensive overview of JAYA algorithm, its versions and applications. *Archives of Computational Methods in Engineering*, *29*(2), 763–792. https://doi.org/10.1007/s11831-021-09585-8