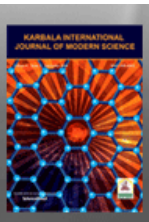

Karbala International Journal of Modern Science

Volume 8 | Issue 4 Article 3

RDLNN-based Image Forgery Detection and Forged Region Detection Using MOT

Akram Hatem Saber
Department of Electronics Engineering, Aligarh Muslim University, alasmr.2a@gmail.com

Mohd Ayyub Khan
Department of Electronics Engineering, Aligarh Muslim University, Aligarh, Uttar Pradesh

Basim Galeb Mejbel
Department of Computer Technician Engineering, AL-Esraa University, Baghdad

RDLNN-based Image Forgery Detection and Forged Region Detection Using MOT

Abstract

Image forgery detection TEMPhas become an emerging research area due to the increasing number of forged images circulating on the internet and other social media, which leads to legal and social issues. Image forgery detection includes the classification of an image as forged or authentic and as well as localizing the forgery wifin the image. In this paper, we propose a Regression Deep Learning Neural Network (RDLNN) based image forgery detection followed by Modified Otsu Thresholding (MOT) algorithm to detect the forged region. The proposed model comprises five steps that are preprocessing, image decomposition, feature extraction, classification and block matching. In the preprocessing step, the RGB images are converted to YCbCr color format. Then, the images are decomposed using the new Polar Dyadic Wavelet Transform (PDyWT), followed by the extraction of important features. The classification phase called RDLNN effectively classifies the normal image and the forged image. For localization of the forgery, the forged image is divided into a number of blocks, and then Genetic Three Step Search (GTSS) algorithm is exploited to identify the dissimilar blocks. To get the exact forged region in the image, the dissimilar blocks are analyzed by the Modified Otsu Thresholding (MOT) algorithm. The proposed algorithm is compared wif widely used image forgery detection algorithms. The results show that the proposed method improves the forgery detection accuracy and precision by at least 6.04% and 3.77%, respectively, as compared to the already existent techniques such as ANFIS, KNN, ANN, and SVM. Moreover, the training time of the proposed network is lower by at least 64.3 % than the above existing techniques

Keywords

Image forgery detection; Image forensics; Image decomposition; regression deep learning neural network; Polar dy-adic wavelet transform; Genetic Three Step Search algorithm (GTSS); Modified Otsu Thresholding (MOT) algorithm.

RESEARCH PAPER

RDLNN-Based Image Forgery Detection and Forged Region Detection Using MOT

Akram H. Saber [a,*], Mohd A. Khan [a], Basim G. Mejbel [b]

[a] Department of Electronics Engineering, Aligarh Muslim University, Aligarh, Uttar Pradesh, India
[b] Department of Computer Technician Engineering, AL-Esraa University, Baghdad, Iraq

Abstract

Image forgery detection has become an emerging research area due to the increasing number of forged images circulating on the internet and other social media, which leads to legal and social issues. Image forgery detection includes the classification of an image as forged or authentic and as well as localizing the forgery within the image. In this paper, we propose a Regression Deep Learning Neural Network (RDLNN) based image forgery detection followed by Modified Otsu Thresholding (MOT) algorithm to detect the forged region. The proposed model comprises five steps that are preprocessing, image decomposition, feature extraction, classification and block matching. In the preprocessing step, the RGB images are converted to YCbCr color format. Then, the images are decomposed using the new Polar Dyadic Wavelet Transform (PDyWT), followed by the extraction of important features. The classification phase called RDLNN effectively classifies the normal image and the forged image. For localization of the forgery, the forged image is divided into a number of blocks, and then Genetic Three Step Search (GTSS) algorithm is exploited to identify the dissimilar blocks. To get the exact forged region in the image, the dissimilar blocks are analyzed by the Modified Otsu Thresholding (MOT) algorithm. The proposed algorithm is compared with widely used image forgery detection algorithms. The results show that the proposed method improves the forgery detection accuracy and precision by at least 6.04% and 3.77%, respectively, as compared to the already existent techniques such as ANFIS, KNN, ANN, and SVM. Moreover, the training time of the proposed network is lower by at least 64.3% than the above existing techniques.

Keywords: Image forgery detection, Image forensics, Image decomposition, Regression deep learning neural network, Polar dyadic wavelet transform, Genetic Three Step Search algorithm (GTSS), Modified Otsu Thresholding (MOT) algorithm

1. Introduction

Recently, there has been a multi-fold increase in sharing of digital images over the internet for various purposes. As a reasonable number of multimedia tools are available [1], anyone with little knowledge about image editing can easily alter images using these tools [2], which leads to image forgery developing legal and social issues. In photography, the widespread presence of these forged images destroys integrity and threatens national security, commerce, media, etc. As a result, image forensics has evolved to regain trust in photography [3,4], and thus image forgery detection (IFD) is an important area to authenticate the images circulating or available on the internet. Image can tamper with different techniques such as copy-move, image splicing, and retouching [5]. In the Copy-move forgery technique, a small image segment is copied and pasted at a different location in the same image [6,7]. In image splicing, a certain part is selected from an image and then pasted into another suitable image [8]. Image retouching is a methodology in which some information in an image is removed, and that part is smoothened using filters along with the properties in the

Received 11 March 2022; revised 29 June 2022; accepted 4 July 2022.
Available online 10 November 2022

* Corresponding author at:
E-mail address: alasmr.2a@gmail.com (A.H. Saber).

https://doi.org/10.33640/2405-609X.3260

neighborhood. The images are tempered in such a way that a human eye cannot identify the forgery [9].

In order to detect the image forgery, preprocessing, Feature Extraction (FE), and classification are performed in series [10]. The input data is processed into a structured format in the preprocessing stage. After that, the images are decomposed into feature vectors [11], and important significant features are extracted from the decomposed features. With the extracted features, the forged images are differentiated efficiently using a classifier [12,13].

1.1. Motivation

A large number of algorithms have been proposed recently; however, many of the algorithms are targeted for copy-move type of forgery, such as the blind copy-move image forgery detection approach that utilizes the Undecimated Dyadic Wavelet Transform (DyWT) [14]. The experiment's results focus on copy-move forgery only. We introduced (Polar DyWT) which was more efficient. Similarly, ANFIS [15] identifies the forgery in the case of splicing only, and this classifier still needs to be explored for further research.

Moreover, the complexity of the proposed algorithms is still large. In low processing devices and real-time situations, complexity matters greatly and needs to be reduced. Therefore, there is still a need to research and explore this area to improve image forgery detection (IFD) for low-processing devices.

1.2. Contribution

The following are the primary implications of this paper: First, we developed an efficient feature extraction method based on the novel Polar Dyadic Wavelet Transform (PDyWT). The image size in DyWT remains constant at different levels. To increase the performance of the DyWT approach, we included the polar form in this DyWT. Second, this study developed a new regression DLNN-based image forgery detection method. The classifier is trained to recognize forged or authentic images and also identify the forgery types (copy move, spliced). Accuracy and training time have both improved. Ultimately, we proposed the GTSS (genetic three-step search) algorithm. This method distinguishes between similar and dissimilar blocks. Finally, using the Modified Otsu Binarization approach, which properly displays the forged regions, the non-similar blocks (forged regions) are clearly discovered. The results show that the proposed method has improved the performance over already existing algorithms by at least 64.3%.

The rest of the paper is organized as follows: Section 2 provides the literature review; Section 3 gives the proposed methodology of RDLNN-based image forgery detection and localization of the forged region using the MOT method; Section 4 illustrates the results and discussion of the proposed method based on performance metrics; finally, Section 5 concludes the paper.

2. Related works

Several image forgery detection algorithms have been proposed in the recent decade. This section provides a brief overview of existing approaches. The methodologies mostly used in forgery detection are characterized into two domains: intrusive and non-intrusive [16–18]. In the intrusive method, also known as the non-blind method, the scope is limited because it requires a certain quantity of digital information to be embedded in the original image [19]. In the non-intrusive technique, also called the blind method, embedded information is not needed. The already available IFD techniques [20,21] have limitations like high complexity while using a larger size of feature vectors. In an improved IFD system, the limitations are handled using several methods such as Convolution Neural Network (CNN) [22] and Artificial Neural Network (ANN) [23]. Some of the methods only establish the effectiveness in splicing forgery but cannot expose all types of image forgery.

Sondos Fadl. [27] established inter-frame forgeries, namely detection system frame deletion, frame in [24]. the Discrete Wavelet Transform (DWT) based decomposition is exploited. The forgery detection speed relies on the position of the copy-move. The detection process should be repeated into smaller blocks to locate the region of the copy-move if the copy-move is localized between two blocks.

Some research papers propose the use Scale-Invariant Feature Transform (SIFT) methods for copy-move forgery detection [25]. However, the SIFT techniques have high complexity in forgery detection and localization.

K Kunj Bihari Meena. [26] presented a novel copy-IFD technique regarding the Tetrolet transform. This was the first time Tetrolet was used in the image forgery detection field. The Tetrolet transform is then used to derive four low-pass coefficients and twelve high-pass coefficients from each block. The results displayed that in this method, the forged portions in the images were

detected and localized precisely in the copied regions. However, the image preserving authenticity was a major complication in this technique. When changing the averaging filter from 3 to 7, the performance metric reduces. The proposed technique cannot detect forgery beyond the scaling range of [80%–135%] and also failed to detect spliced forgery.

Insertion, and frame duplication, by 2D-CNN of spatiotemporal information. The fusion aimed at deep automatic FE (Feature Extraction) and Gaussian RBF (Radial Base Function) Multi-Class Support Vector Machine (RBF-MSVM) was employed for the classification procedure. The technique efficiently detected the entire inter-frame forgeries in forged videos, even after post-processing operations, namely Gaussian noise, brightness modifications, Gaussian blurring, and compression. However, the forgery detection is time hungry because of the deployment of large-size feature vectors.

Khizar Hayat. [28] proffered a forgery detection technique based on the Discrete Wavelet Transform (DWT) and the Discrete Cosine Transform (DCT) for feature reduction. The results exhibited that the technique surpassed the other modern methods with respect to accuracy rates. But its usability is restricted if the copy-move where localized between two blocks. Similarly, if the duplicated region is significantly resized or rotated, the approach produces unsatisfactory results.

Sreenivasu Tinnathi. [29] produced the Copy-Move Forgery Detection (CMFD) method regarding the adaptive segmentation and the hybrid FE method. The proficiency of the CMFD method was enhanced, and the computational complexity was minimized by the partition of the tampered image into non-overlapped segments. The method capability was enhanced by using HWHT (Hybrid Wavelet Hadamard Transform) to take out the segment features and attain a strong result by implementing geometrical deviations within the image. However, the drawbacks of this technique were high computational complexity and the difficulty in identifying the shape region.

Bin Xiao [30] introduced a splicing forgery detection technique using two methods: Coarse-to-Refined CNN (C2RNet) and diluted adaptive clustering. The experiment's results revealed that the detection method attained the best outcome even under different attack conditions compared to the novel splicing forgery detection techniques. However, only a single tampered region of an image was focused on in this technique; hence, the post-processing method was restricted.

Ghulam Muhammad. [14] deployed a blind copy-move image forgery detection approach that utilized the Undecimated Dyadic Wavelet Transform (DyWT). The experiment's results showed that, in comparison with other techniques, this technique is more capable with the use of discrete wavelet transform DWT and the LL1 or HH1 sub-bands only. The method's efficiency in copy-move image forgery alone was proved, and it had not identified all the types of the image forgery.

To address these challenges, the presented work established RDLNN-based Image Forgery Detection and the identification of a forged region using the MOT technique. The novelty of this work is to identify the exact forged region in a forged image and reduce the computational complexity.

3. Materials and methods

The probability of image forgery has been increased with the improvement of high-resolution digital cameras along with photo editing software and their enhanced features. In forensics investigation and numerous other fields, an image is considered legal evidence, so spotting image manipulation is significant. Hence, an RDLNN-based IFD and forgery region detection using an MOT algorithm is introduced in this method.

The proposed method contains five steps: preprocessing, decomposition, feature extraction, detection, and localization. In the beginning, the publicly available input data are preprocessed, wherein the image's color is converted into YCbCr mode. Therefore, the image can be recognized by the machine easily. After that, the PDyWT is employed for decomposing the preprocessed image. The significant features are extracted from the decomposed images, and these features are inputted into the classifier called RDLNN. Now, the classifier differentiates between the forged and original image more efficiently. When the image is identified as a forged image, it is partitioned into a number of blocks. Then, using the Genetic Three Step Search (GTSS) approach, the block matching function is executed for every block to detect similar and dissimilar blocks. The dissimilar blocks are evaluated using the MOT algorithm, and the forged region of an image is detected more precisely. Figure 1 presents the block diagram of the proposed technique.

3.1. Preprocessing

First, the inputted images are directed into the preprocessing step. In this step, the input data is

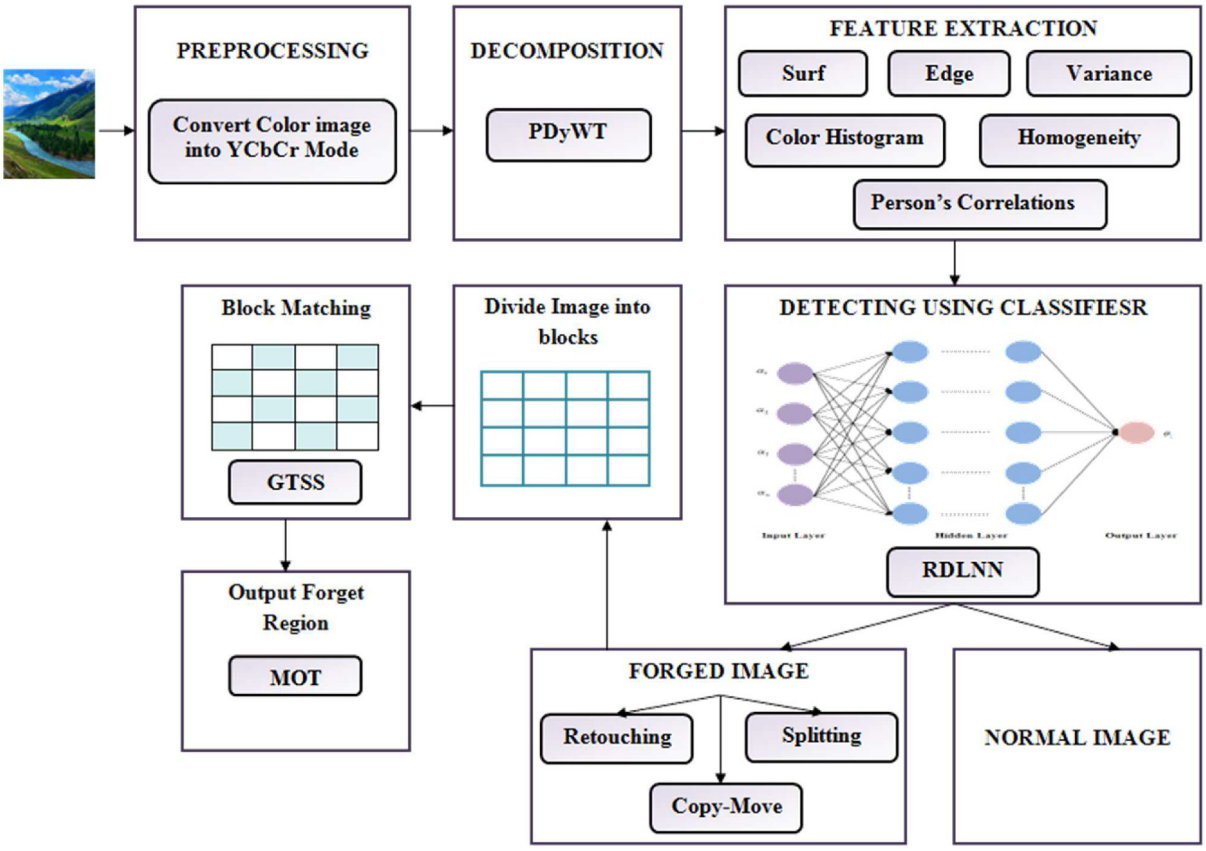

Fig. 1. The block diagram for the proposed methodology.

converted into a structured format after processing. The following expression represents the pre-processing function.

$$\delta = \delta_p\left[R_{p_x}\right] \tag{1}$$

wherein δ implies the preprocessing function's output; R_{p_x} specifies the input data; δ_p signifies the preprocessing. In the present work, the input color image is converted into YCbCr color mode in the preprocessing step. In YCbCr format, Y implies the luma components, Cb signifies the blue-difference chroma components, and Cr indicates the red-difference chroma components. The color image's conversion as the YCbCr color mode function is arithmetically equated as,

$$\aleph_{cr} = \varsigma_Y\left[R_{p_x}\right] \tag{2}$$

wherein $\aleph_{cr}$ implies the conversion function's result of the color image as YCbCr color mode and ς_Y signifies the YCbCr color conversion function.

3.2. Decomposition

The decomposition phase is performed after pre-processing. The new PDyWT separates the complex image into individual components. In DyWT, the image size remains the same at a disparate level. It is decomposed into '4' sub-images at each level, labeled as LL: The upper left quadrant; HL: The lower left; LH: the upper right blocks of an image; and HH: The lower right quadrant. Most data are concentrated in the LL sub-image, which is considered the image approximation. Merely partial data concerning an image is rendered by the DyWT. The polar form (P) is merged with DyWT to enhance its performance. Therefore, a large amount of information and coordinate values are rendered by the PDyWT via analyzing the complete image. The PDyWT steps and their following equation are given as: Presume $\boldsymbol{\delta}$ as the image to be decomposed, $\kappa(\hbar)$ signifies the wavelet function, $v(\hbar)$ implies shifted through a translation time via the polar coordinates χ_1 and χ_2. An image PDyWT can well be gauged as

$$\partial_{\varsigma}(\chi_1,\chi_2=\gamma*\cos\theta\, I_{\chi_1})=\frac{1}{\sqrt{\chi_1}}\int_{-\infty}^{\infty}\kappa(\hbar)v^{*}\left(\frac{\hbar_{-\chi_2}}{\chi_1}\right)d\hbar \quad (3)$$

wherein $v^{*}(\hbar)$ signifies the complex conjugate of $v(\hbar)$, which is compressed or expanded upon $\boldsymbol{\chi}_1$. The polar coordinates are mathematically represented by

$$\boldsymbol{\chi}_1=\gamma*\cos\boldsymbol{\theta} \quad (4)$$

$$\boldsymbol{\chi}_2=\boldsymbol{\chi}*\sin\boldsymbol{\theta} \quad (5)$$

wherein, $\boldsymbol{\chi}_1$ and $\boldsymbol{\chi}_2$ signifies the polar coordinates, which renders the required data and the coordinate values centered on sinθ and cosθ.

3.3. Feature extraction

The Feature Extraction (FE) is carried out subsequent to the Image Decomposition (ID). As for the decomposed image, the best suitable and informative features are extracted. Furthermore, the redundant data is reduced. In the proposed work, the vital features, say Pearson's correlation, Speeded Up Robust Feature (SURF), edge, color histogram, homogeneity, with variance features, are extracted. The FE process is expressed as

$$\Gamma^{t}_{ex}=\Gamma^{t}_{ex}\{\partial_{\varsigma}(\chi_1,\chi_2)\} \quad (6)$$

Here, $\partial_{\varsigma}(\chi_1,\chi_2)$ signifies the decomposed image; Γ^{t}_{ex} implies the FE function given by

$$\boldsymbol{\Gamma}^{t}_{ex}=\left[\boldsymbol{\Gamma}^{t}_{sf},\boldsymbol{\Gamma}^{t}_{ed},\boldsymbol{\Gamma}^{t}_{pc},\boldsymbol{\Gamma}^{t}_{ch},\boldsymbol{\Gamma}^{t}_{ho},\boldsymbol{\Gamma}^{t}_{va}\right] \quad (7)$$

wherein Γ^{t}_{sf} signifies the SURF Γ^{t}_{ed} implies the edge feature; Γ^{t}_{pc} signifies the Pearson's correlation; Γ^{t}_{ch} implies the color histogram; Γ^{t}_{ho} implies homogeneity; the variance is implied as Γ^{t}_{va}.

3.4. Detection using classifier

The extracted features extracted from the decomposed images are inputted to the RDLNN classifier, which identifies whether the inputted image is authentic or forged. The three layers are comprised of the Deep Learning Neural Network (DLNN), namely the Input Layer (IL), Hidden Layer (HL), and Output Layer (OL). The inputted data is acquired by the IL and passed into the classifier. The HL is also named as a dense layer. The dense layer is accountable for executing the function of adding the product of the inputted value and the weight vector of every input node linked to it. OL is accountable for producing the classifier's outcome. But, the Activation Function (AF) doesn't actively support the multi-layers in the DLNN's HL. The performance degradation is exhibited by the AF in the case of large inputs offered into the classifier. A chance of offering irrelevant results exists. To manage that, the Regression activation (R) function is utilized in the DLNN. Therefore, better outputs are effectively provided by the RDLNN with no error. Figure 2 exhibits the DNN's general structure: The steps that are incorporated in the RDLNN are enlisted below:

Step 1: The image's features are offered as the input to the classifier in an initial step, and the corresponding weight values are presented as:

$$\Gamma_i=\{\Gamma_1,\Gamma_2,\Gamma_3,\Gamma_4,.....,\Gamma_n\} \quad (8)$$

$$\lambda_i=\{\lambda_1,\lambda_2,\lambda_3,\lambda_4,....,\lambda_5\} \quad (9)$$

wherein $\boldsymbol{\Gamma}_i$ signifies the inputs of the classifier and $\boldsymbol{\lambda}_i$ indicates the weight values.

Step 2: The outcome from the IL is offered to the HL. Herein, the multiplication of the provided inputs with the weight vectors is done. After that, the bias vectors are chosen arbitrarily and summed up together. The IL is mathematically indicated as:

$$\vartheta_i=\sum_{i=1}^{n}\Gamma_i\lambda_i+\beta_i \quad (10)$$

Step 3: The HL's outcome is executed, and so is the AF. The HL outcome's arithmetical illustration is produced as,

$$\varphi_i=f\left(\sum\vartheta_i\lambda_i+\beta_i\right) \quad (11)$$

wherein $f(.)$ signifies the AF.

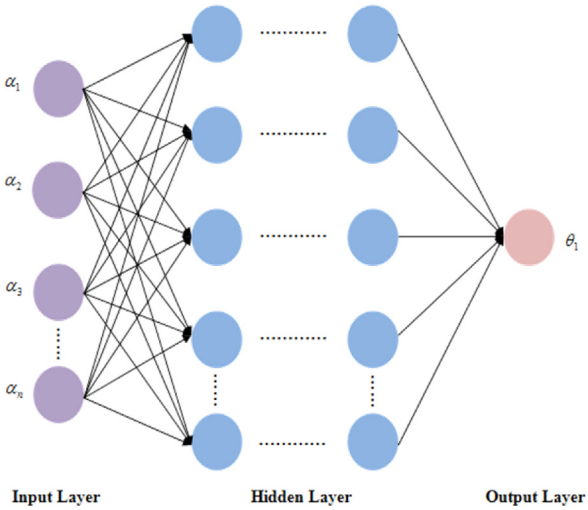

Fig. 2. : General structure of the DNN.

Step 4: In the work proposed, the regression AF is utilized that effectively supports the classifier's multi-layers and offers the ideal result without any error. The regression AF is formulated as,

$$f(\tilde{I}_r) = \beta_i + \sum_{f=1}^{M} \left((\varphi_i)_{f+1} - (\varphi_i)_f \right) . \psi \left((\lambda_i)_{f+1}, (\lambda_i)_f \right) \tag{12}$$

wherein β_i signifies the bias function; ϕ_i indicates the HL; λ_i symbolizes the weight value; ψ indicates the kernel function of the weight value.

Step 5: The three steps mentioned above are executed for every stage in RDLNN. Finally, the output unit is computed by adding up every weight of the input features.

$$\Theta_i = f_a \left(\sum \varphi_i \lambda_i + \beta_i \right) \tag{13}$$

wherein the output unit is signified as $\boldsymbol{\Theta_i}$, the HL's weights are signified as $\boldsymbol{\lambda_i}$, ϕ_i indicates the layer's value that precedes the OL, and f_a signifies the SoftMax AF.

Step 6: Finally, the classifier's outcome is contrasted with the target output value. The error value is the variation betwixt these '2' values. The error value is mathematically specified as:

$$E_r = A_i - \Theta_i \tag{14}$$

wherein E_r symbolizes the error value, A_i signifies the target output values, and Θ_i indicates the classifier's current output value. The model offers the precise value of the error value $E_r = 0$. The back-propagation is done by updating the weights if the error value $E_r \neq 0$. Finally, two forms of output are offered by the classifier's output; one is the normal image, and the other is the forged image.

3.5. Block matching

After detecting the classifier's outcome as the forged image, it is inputted to the block matching function to discover the exact forged region of an image. Usually, the forged image encompasses '2' classes:

- Copy-move image
- Splicing

The forged areas are detected by dividing them into manifold blocks. Next, GTSS performs the block matching function on the divided blocks. Generally, the three-step search block matching algorithm uses the Sum of Absolute Differences (SAD). It is costly and computationally intricate to evaluate the SAD. Furthermore, the block matching procedure is believed to be the utmost consuming operation. The proposed work utilizes the genetic algorithm (GA) in combination with the Three Step Search (TSS) algorithm (called GTSS) to avoid the enumeration of numerous search locations. Therefore, similar and non-similar blocks are detected by the GTSS. The GTSS's algorithmic steps are:

Step 1: The equivalent block of the current frame's block should be found in the reference frame. Then, the search window around it should be defined. Generally, the number of steps that are necessarily aimed at the provided search window ϖ is rendered by

$$\ell = \left[log_{\,2}(\varpi + 1) \right] \tag{15}$$

Step 2: The step size v_{sz} is defined. The distance betwixt pixels in a search space for n^{th} a step is mathematically written as:

$$v_{sz}(n) = 2^{\ell - n} \tag{16}$$

Step 3: Next, the SAD value is enumerated aimed at a diverse number of blocks, including the central block. In general, a large number of SAD values are produced by the TSS, which in turn brings about high computational intricacy. The genetic algorithm is utilized to deal with that, which aids in an effectual assortment of SAD values.

Step 4: Initialize the number of SAD values ξ_a, which are signified as:

$$\xi_a = \{\xi_1, \xi_2, \xi_3, ..., \xi_n\} \tag{17}$$

Step 5: The fitness is assessed aimed at every SAD value. Next, the fitness f can well be calculated as the Mean Absolute Difference (MAD), which is mathematically signified as,

$$\boldsymbol{M_{ad}} = \frac{1}{\boldsymbol{\mu\omega}} \sum_{i=1}^{\mu} \sum_{j=1}^{\omega} |v_s(i,j) - v_{s+1}(i,j)| \tag{18}$$

wherein v_s signifies the original block under consideration v_{s+1} implies the block identified at the destination frame subsequent to transformation and $(\boldsymbol{\mu}, \boldsymbol{\omega})$ implies the block's dimensions.

Step 6: The fitness value is set with a certain level of threshold $\boldsymbol{TH_{id}}$. If it lies within the threshold value, it is regarded as the finest SAD value, or else the iteration recurs until the best SAD value is obtained.

Step 7: For crossover α_{co} estimation, the SAD values selected for crossover are taken into the next generation after swapping one or more arbitrary values.

Step 8: The SAD values that are chosen for mutation α_{ma} are swapped with uniformly distributed arbitrary values aimed at the centroid, angle, scale, shear, and squeeze. Until the best SAD value is found, the crossover along with the mutation process recurs.

Step 9: The block encompassing the optimal SAD value is taken as the best-matched block that can be utilized as the Centre block. Next, half the step size is pondered, and the genetic algorithm gauges the SAD values. Then, the block encompassing the optimal SAD is selected as the best match, which is employed as the Centre block aimed at the subsequent step. Until a step size of '1' is attained, this iteration continues by updating the best SAD values as the Centre block. Thus, the GTSS algorithm easily identifies similar blocks and non-similar blocks.

3.6. Pseudocode of the GTSS algorithm

The GTSS algorithm pseudocode is explicated in Algorithm 1. The non-similar blocks as of the block matching function are inputted into the identification stage. Here, the non-similar blocks that signify the image's forged region can be clearly identified by utilizing the MOT methodology. Generally, the Otsu Thresholding (OT) method examines the total image; however, a few pixels are left uncovered. Aiming to resolve the issue, the proposed methodology utilizes a MOT wherein the image's X, Y, and Z-axis are examined and cover the entire pixels inside the image. Therefore, the MOT methodology efficiently partitions the non-similar and similar patches. The processes engaged in the MOT methodology are explicated in brief as follows:

- Generally, OT methodology suggests that the image comprises just two entities, the foreground and then the background. Otsu fixes the threshold aimed at decrementing the class distributions' overlapping.
- Normally, Otsu's methodology partitions the image as two regions, i.e., light and dark regions that are signified as Φ_1 and Φ_2 articulated as:

$$\Phi_1 = \{0, 1, 2, \ldots\ldots, t_{h_c}\} \tag{19}$$

$$\Phi_2 = \{t_h, t_h + 1, \ldots\ldots, \ell_d - 1, \ell_d\} \tag{20}$$

wherein t_h signifies the threshold value and implies the image's maximal grey level at instance 256.

Algorithm 1. Pseudocode of the GTSS algorithm.

Algorithm 1 Pseudocode of the GTSS algorithm

Input: Number of divided blocks of the forged image
Output: Identifying the similar and dissimilar blocks of forged image

Begin
  Initialise: number of divided blocks
  Initialise: SAD values ξ_i
  Compute the fitness for each SAD value as
  If $f(\xi_i) \le TH_{ld}$
    Update the SAD value $\xi_{i_{best}}$
  Else
    Evaluate the cross over function a_{co}
    Evaluate the mutation function a_{ma}
  End if
    Calculate the window size, ℓ by computing, $[log, (\varpi + 1)]$
    Calculate the step-size, v_{sz} by computing, 2^{l-n}
    While step-size=1
    Break the iteration
  Else
    The iteration continues
  End if
  Return the similar and dissimilar blocks
End

The optimal t_h is defined by decrementing the weighted group variances' summation, wherein the weights are computed as of the corresponding groups' possibility. The histogram probability $\wp(i)$ of the observed grey value.
$\mathbf{i} = 1, 2, 3, \ldots, \mathbf{M}$ is equated as,

$$\wp(i) = \frac{n\{(a_1, a_2) | img(a_1, a_2) = i\}}{(A_1, A_2)} \tag{21}$$

wherein the index aimed at an image's row and column is denoted as a_1 and a_2, correspondingly.

- The weight $\omega_b(t_h)$, geometric mean $v_b(t_h)$, and variance $\sigma_b^2(t_h)$ of the class Φ_1 comprising (0 to t_h) intensity value range are arithmetically equated as,

$$\omega_b(t_h) = \sum_{i=1}^{t_h} \wp(i) \tag{22}$$

$$v_b(t_h) = \frac{\sum_{i=1}^{t_h} i * \wp(i)}{\omega_b(t)} \tag{23}$$

$$\sigma_b^2(t_h) = \frac{\sum_{i=1}^{t_k} (i - v_b(t_h))^2 * \wp(i)}{\omega_b(t_h)} \tag{24}$$

The weight $\omega_g(t_h)$, geometric mean $v_g(t_h)$, and variance $\sigma_g^2(t_h)$ of a class Φ_2 comprising $(t_h + 1 to I)$ intensity value ranges are arithmetically articulated as:

$$\omega_g(t_h) = \sum_{i=1}^{t_k} \wp(i) \tag{25}$$

$$v_g(t_h) = \frac{\sum_{i=1}^{t_k} i * \wp(i)}{\omega_g(t_h)} \tag{26}$$

$$\sigma_g^2(t_h) = \frac{\sum_{i=1}^{t_h} \left(i - v_g(t_h)\right)^2 * \rho(i)}{\omega_g(t_h)} \tag{27}$$

$$\sigma_\omega^2 = \omega_b(t_h) * \sigma_b^2(t_h) + \omega_g(t_h) * \sigma_g^2(t_h) \tag{28}$$

- The total variance can be attained via summarizing the within-class and between-class variance, which is equated as,

$$\sigma_Y^2 = \sigma_\omega^2(t_h) + \sigma_b^2(t_h) \tag{29}$$

herein σ_Y^2 implies the image's total variance, which doesn't rely on the threshold. Hence, the MOT methodology precisely detects an image's forged regions.

4. Results and discussion

The proposed method's experimental analysis is presented here. To state its effectiveness, the proposed technique's performance analysis and comparative analysis are executed. The proposed IFD system is applied in MATLAB. The input images are taken from a publicly accessible dataset, Institute of Automation Chinese Academy of Science CASIAv2 (https://ieeexplore.ieee.org/.) [31]. The following performance metrics are used to compare the efficacy of the proposed technique: sensitivity, specificity, accuracy, precision, recall, F-Measure, False-Negative Rate (FNR), False-Positive Rate (FPR), Matthews Correlation Coefficient (MCC), and training time.

The proposed RDLNN's performance analysis is compared with different existent techniques, like ANFIS, KNN, ANN, and SVM, to state its effectiveness. Table 1 exhibits the proposed RDLNN's performance analysis concerning different performance metrics, namely sensitivity, specificity, and accuracy. The sensitivity at the rate of 94.57%, specificity of 97.83%, and 96.2% accuracy are obtained by the proposed RDLNN. The sensitivity at an average of 84.92%, specificity at 63.58% average, and accuracy at 73.06% average are obtained by existent techniques, like ANFIS, KNN, ANN, and SVM, which are comparatively low when analogized to the proposed RDLNN. Figure 3 depicts the graphical illustration of Table 1. The proposed RDLNN is relatively examined with different existing techniques like ANFIS, KNN, ANN, and SVM. From the comparative study, it is obvious that the proposed technique outperforms the other top-notch methods by yielding the maximal rate of sensitivity of 94.57%, specificity of 97.83%, and accuracy of 96.2%, which is comparatively higher when contrasted to the prevailing techniques. Therefore, the normal image and the forged image are efficiently distinguished by the proposed RDLNN with no misclassification error.

Table 2 tabulates the proposed RDLNN's performance analysis with different existing techniques concerning the precision, recall, and F-Measure. The number of positive class predictions of the design is quantified by precision and recall, and both precision and recall values are balanced by the F-Measure. Thus, the model's robustness is signified by the higher value of precision, recall, and F-Measure. In the proposed architecture, the precision of 97.75%, recall of 94.57%, and F-Measure of 96.13%

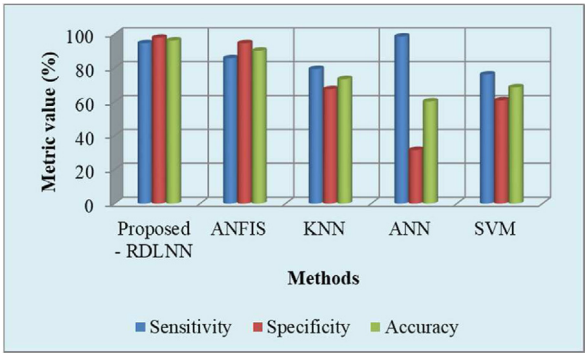

Fig. 3. Comparative analysis of proposed RDLNN based on sensitivity, specificity, and accuracy.

Table 1. Performance analysis of proposed RDLNN based on sensitivity specificity and accuracy.

Performance metrics (%)/ Techniques	Sensitivity	Specificity	Accuracy
Proposed - RDLNN	94.57	97.83	96.2
ANFIS [15]	85.71	94.57	90.16
KNN [32]	79.35	67.39	73.37
ANN [33]	98.55	31.52	60.25
SVM [32]	76.09	60.87	68.48

Table 2. Performance analysis of proposed RDLNN based on precision, recall, and F-Measure.

Performance metrics (%)/ Techniques	Precision	Recall	F-Measure
Proposed - RDLNN	97.75	94.57	96.13
ANFIS	93.98	85.71	89.66
KNN	70.87	79.35	74.87
ANN	51.91	98.55	68
SVM	66.04	76.09	70.71

are obtained. In already existing techniques ANFIS, KNN, ANN, and SVM, the precision ranges between 66.04% and 93.98%, Recall ranges between 85.71% and 76.09%, and F-Measure values range between 68% and 89.66% are attained by the prevailing techniques. The proposed RDLNN is showing better performance as compared to existing techniques. Figure 4 offers a clear view of tabulation 2. The figure also illustrates the edge in the precision, recall, and F-Measure values of the proposed RDLNN over the other techniques.

The Performance of the proposed RDLNN in terms of FPR, FNR and MCC is shown in Table 3. The work's reliability is revealed by the lower value of FPR and FNR rates. As per the statement, less FPR and FNR values are attained by the proposed RDLNN when analogized to the existent works. The FPR at the rate of 2.17% and FNR at the rate of 5.43% are attained by the proposed work, while the existing works, like ANFIS, KNN, ANN, and SVM, attain an FPR value that ranges between 1.45 and 68.48%. Moreover, the work proposed is also assessed concerning the MCC metric. In contrast to the FPR and FNR rates, the model's effectiveness is signified by the higher value of MCC. Herein, the proposed RDLNN's MCC value is 92.44%, but the MCC value that overall ranges between 37.39% and 80.63% are attained by the existing techniques. Thus, the proposed RDLNN is a less error-prone model and provides more accurate results with less misprediction. The proposed RDLNN's graphical analysis with diverse existent techniques, namely ANFIS, KNN, ANN, and SVM, is exhibited in Fig. 5.

The proposed design's efficacy is exhibited by the comparative study by examining the false prediction values. The false prediction is decremented by the technique proposed, and the classification rate is improvised by evading misclassification and staying effective against the prevailing methods. Hence, the existing techniques are outperformed by the proposed RDLNN by attaining low false prediction values and higher MCC scores.

The total training time consumed by the proposed RDLNN technique, along with different existent techniques like ANFIS, KNN, ANN, and SVM, is exhibited in Table 4. The proposed method attained the training time of 0.6711 s, while the training time ranges between 1.881 seconds and 41.703 s are attained by the existent techniques like ANFIS, KNN ANN, and SVM. Hence, the proposed technique takes less time to finish the training of the network faster than the existing methodologies. Figure 6 depicts the comparative examination of the training time taken by the RDLNN classifier and other prevailing techniques, like ANFIS, KNN, ANN, and SVM. It is understood from the graph that 0.671 s is taken by the proposed RDLNN while about 1.881 s, 3.873 s, 7.285 s, and 41.70 s correspondingly are required by the existing techniques, i.e., ANFIS, KNN, ANN, and SVM, aimed at training

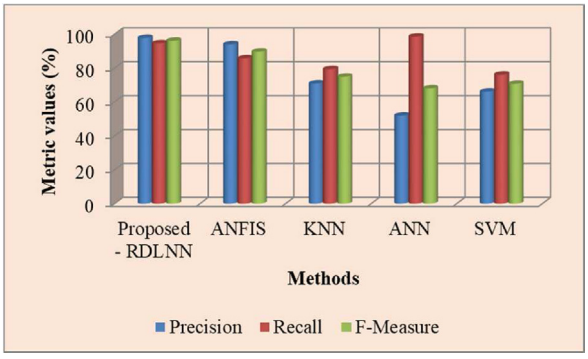

Fig. 4. Comparative analysis of proposed RDLNN based on precision, recall, and F-Measure.

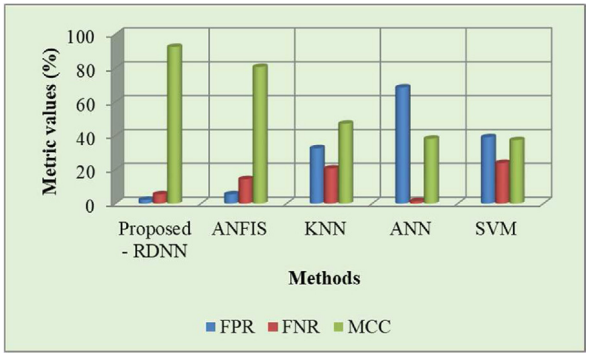

Fig. 5. Comparative analysis of proposed RDLNN based on FPR, FNR, and MCC.

Table 3. Performance analysis of proposed RDLNN based on FPR, FNR, and MCC.

Performance metrics (%)/Techniques	FPR	FNR	MCC
Proposed - RDNN	2.17	5.43	92.44
ANFIS	5.43	14.29	80.63
KNN	32.61	20.65	47.08
ANN	68.48	1.45	38.22
SVM	39.13	23.91	37.39

Table 4. Performance analysis of proposed RDLNN based on Based on Training Time.

Training time (sec)/Techniques	Training Time
Proposed - RDLNN	0.671198
ANFIS	1.881727
KNN	3.873886
ANN	7.285279
SVM	41.70318

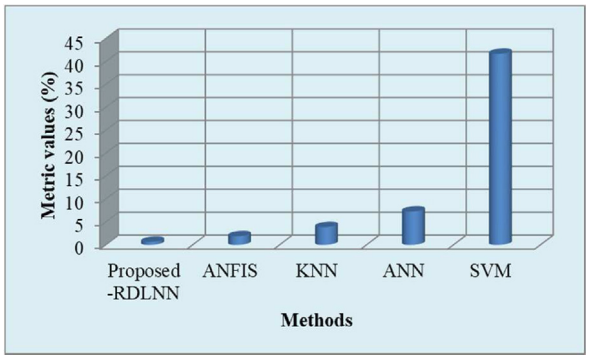

Fig. 6. Comparative analysis of proposed RDLNN based on training time.

the data. Thus, the classification process is completed by the proposed RDLNN as rapidly as possible compared to the existing works.

5. Conclusions

The RDLNN-based Image Forgery Detection has been proposed in this paper. It discovers the forged region utilizing the MOT methodology. The technique engages numerous operations, like preprocessing, image decomposition, feature extraction, identification, and forged image block matching to detect the forged region of the image. After that, the work's end result is examined, wherein the performance examination and the comparative assessment of the proposed and existent methodologies are executed regarding a few performance metrics aimed at validating the proposed methodology's efficacy. The developed methodology efficiently detects the forged image in diverse conditions. The publicly prevalent datasets are employed in the examination wherein the RDLNN proposed stands with a higher accuracy rate as compared to the existing techniques. Specifically, the proposed RDLNN-based image forgery detection identifies the forged region with 96.2% accuracy, which is at least 6.04% higher than the accuracy reported by the already existing state-of-the-art image forgery techniques such as ANFIS and KNN.

Conflict of interest

No conflict of interest.

References

[1] J.L. Zhong, C.M. Pun, Y.F. Gan, Dense moment feature index and best match algorithms for video copy-move forgery detection, Inf Sci 537 (2020) 184–202, https://doi.org/10.1016/j.ins.2020.05.134.
[2] M.M. Isaac, M. Wilrscy, Image forgery detection based on Gabor wavelets and local phase quantization, Procedia Comput Sci 58 (2015) 76–83, https://doi.org/10.1016/j.procs.2015.08.016.
[3] S.N. Youseph, R.R. Cherian, Pixel and edge-based illuminant color estimation for image forgery detection, Procedia Comput Sci 46 (2015) 1635–1642, https://doi.org/10.1016/j.procs.2015.02.099.
[4] Y. Wang, X. Kang, Y. Chen, Robust and accurate detection of image copy-move forgery using PCET-SVD and histogram of block similarity measures, J Inf Secur Appl 54 (2020) 1–11, https://doi.org/10.1016/j.jisa.2020.102536.
[5] D.C. Jeronymo, Y.C.C. Borges, L.S. Coelho, Image forgery detection by semi-automatic wavelet soft thresholding with error level analysis, Expert Syst Appl 85 (2017) 348–356, https://doi.org/10.1016/j.eswa.2017.05.044.
[6] V. Schetinger, M. Iuliani, A. Piva, M.M. Oliveira, Image forgery detection confronts image composition, Comput Graph 68 (2017) 152–163, https://doi.org/10.1016/j.cag.2017.08.014.
[7] A.H. Saber, M.A. Khan, B.G. Mejbel, Digital Image forgery Detection by utilize combined feature extraction techniques, in: 2021 international conference of advanced technology and engineering (ICOTEN) IEEE, 2021, pp. 1–7. https://ieeexplore.ieee.org/document/9493527.
[8] S. Dua, J. Singh, H. Parthasarathy, Image forgery detection based on statistical features of block DCT coefficients, Procedia Comput Sci 171 (2020) 369–378, https://doi.org/10.1016/j.procs.2020.04.038.
[9] C.M. Pun, B. Liu, X.C. Yuan, Multi-scale noise estimation for image splicing forgery detection, J Vis Commun Image Represent 38 (2016) 195–206, https://doi.org/10.1016/j.jvcir.2016.03.005.
[10] B. Diallo, T. Urruty, P. Bourdon, C.F. Maloigne, Robust forgery detection for compressed images using CNN supervision, Foren Sci Int Rep 2 (2020) 1–11, https://doi.org/10.1016/j.fsir.2020.100112.
[11] J.L. Zhong, C.M. Pun, Two-pass hashing feature representation and searching method for copy-move forgery detection, Inf Sci 512 (2020) 675–692, https://doi.org/10.1016/j.ins.2019.09.085.
[12] S. Farooq, M.H. Yousaf, F. Hussain, A generic passive image forgery detection scheme using local binary pattern with rich models, Comput Electr Eng 62 (2017) 459–472, https://doi.org/10.1016/j.compeleceng.2017.05.008.
[13] T. Mahmood, A. Irtaza, Z. Mehmood, M.T. Mahmood, Copy-move forgery detection through stationary wavelets and local binary pattern variance for forensic analysis in digital images, Forensic Sci Int 279 (2017) 8–21, https://doi.org/10.1016/j.forsciint.2017.07.037.
[14] G. Muhammad, M. Hussain, G. Bebis, Passive copy-move image forgery detection using undecimated dyadic wavelet transform, Digit Invest 9 (2020) 49–57, https://doi.org/10.1016/j.diin.2012.04.004.
[15] H.G. Hadigheh, G.B. Sulong, Splicing forgery detection based on neuro fuzzy fusion, Life Sci J 15 (2018) 5–88. http://www.lifesciencesite.com/lsj/life150518/.
[16] A.H. Saber, M.A. Khan, B.G. Mejbel, A survey on image forgery detection using different forensic approaches, Adv Sci Technol Eng Syst J 5 (2020) 361–370, https://doi.org/10.25046/aj050347, 3.
[17] F. Casino, T.K. Dasaklis, G. Spathoulas, M. Anagnotopoulos, A. Ghosal, I. Borocz, Research trends, challenges, and emerging topics in digital forensics: a review of reviews, IEEE Access IEEE Access 10 (2022) 25464–25493. https://ieeexplore.ieee.org/document/9720948.
[18] J.C. Lee, C.P. Chang, W.K. Chen, Detection of copy-move image forgery using histogram of orientated gradients, Inf Sci 321 (2015) 250–262, https://doi.org/10.1016/j.ins.2015.03.009.
[19] A.V. Malviya, S.A. Ladhake, Pixel-based image forensic technique for copy-move forgery detection using auto color correlogram, Procedia Comput Sci 79 (2016) 383–390, https://doi.org/10.1016/j.procs.2016.03.050.

[20] S. Devi Mahalakshmi, K. Vijayalakshmi, S. Priyadharsin, Digital image forgery detection and estimation by exploring basic image manipulations, Digit Invest 8 (2012) 215–225, https://doi.org/10.1016/j.diin.2011.06.004.
[21] R.S. Oommen, M. Jayamohan, S. Sruthy, Using fractal dimension and singular values for image forgery detection and localization, Proc Technol 24 (2016) 1452–1459, https://doi.org/10.1016/j.protcy.2016.05.176.
[22] B. Liu, C.M. Pun, Locating splicing forgery by fully convolutional networks and conditional random field, Signal Process Image Commun 66 (2018) 103–112, https://doi.org/10.1016/j.image.2018.04.011.
[23] E.S. Gopi, Digital image forgery detection using artificial neural network and independent component analysis, Appl Math Comput 194 (2007) 540–543, https://doi.org/10.1016/j.amc.2007.04.055.
[24] T. Mahmood, Z. Mehmood, M. Shah, T. Saba, A robust technique for copy-move forgery detection and localization in digital images via stationary wavelet and discrete cosine transform, J Vis Commun Image Represent 53 (2018) 202–214, https://doi.org/10.1016/j.jvcir.2018.03.015.
[25] B. Yang, X. Sun, H. Guo, Z. Xia, X. Chen, A copy-move forgery detection method based on CMFD SIFT, Multimed Tool Appl 77 (2020) 837855, https://doi.org/10.1007/s11042-016-4289-y.
[26] K.B. Meena, V. Tyagi, A copy-move image forgery detection technique based on tetrolet transform, J Inf Secur Appl 52 (2020) 1–9, https://doi.org/10.1016/j.jisa.2020.102481.
[27] S. Fadl, Q. Han, Q. Li, CNN spatiotemporal features and fusion for surveillance video forgery detection, Signal Process Image Commun 90 (2021) 1–32, https://doi.org/10.1016/j.image.2020.116066.
[28] K. Hayat, T. Qazi, Forgery detection in digital images via discrete wavelet and discrete cosine transforms, Comput Electr Eng 62 (2017) 448–458, https://doi.org/10.1016/j.compeleceng.2017.03.013.
[29] S. Tinnathi, G. Sudhavani, An efficient copy-move forgery detection using adaptive watershed segmentation with AGSO and hybrid feature extraction, J Vis Commun Image Represent 74 (2021) 1–19, https://doi.org/10.1016/j.jvcir.2020.102966.
[30] B. Xiao, Y. Wei, X. Bi, W. Li, J. Ma, Image splicing forgery detection combining coarse to refined convolutional neural network and adaptive clustering, Inf Sci 511 (2020) 172–191, https://doi.org/10.1016/j.ins.2019.09.038.
[31] J. Dong, W. Wang, T. Tan, Casia image tampering detection evaluation database, in: 2013 IEEE China summit and international conference on signal and information processing, 2013, pp. 422–426. https://ieeexplore.ieee.org/document/6625374.
[32] L. Almawas A. Alotaibi, K. Heba, Comparative performance study of classification models for image-splicing detection, Procedia Comput Sci 175 (2020) 278–285, https://doi.org/10.1016/j.procs.2020.07.041.
[33] S. Ranjan, G. Prayati, B. Anupama, A. Monika, M. Anu, Framework for image forgery detection and classification using machine learning, in: 2018 2nd international conference on trends in electronics and informatics (ICOEI) IEEE, 2018, pp. 1–9. https://ieeexplore.ieee.org/abstract/document/8553924.