Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.031319
Article

Tech Science Press

Copy Move Forgery Detection Using Novel Quadsort Moth Flame Light Gradient Boosting Machine

R. Dhanya[1,*] and R. Kalaiselvi[2]

[1]Department of Computer Applications, Noorul Islam Centre for Higher Education, Nagercoil, 629180, India
[2]Department of Computer Science Engineering, Noorul Islam Centre for Higher Education, Nagercoil, 629180, India
*Corresponding Author: R. Dhanya. Email: rdhanya2022@gmail.com
Received: 14 April 2022; Accepted: 08 June 2022

Abstract: A severe problem in modern information systems is Digital media tampering along with fake information. Even though there is an enhancement in image development, image forgery, either by the photographer or via image manipulations, is also done in parallel. Numerous researches have been concentrated on how to identify such manipulated media or information manually along with automatically; thus conquering the complicated forgery methodologies with effortlessly obtainable technologically enhanced instruments. However, high complexity affects the developed methods. Presently, it is complicated to resolve the issue of the speed-accuracy trade-off. For tackling these challenges, this article put forward a quick and effective Copy-Move Forgery Detection (CMFD) system utilizing a novel Quad-sort Moth Flame (QMF) Light Gradient Boosting Machine (QMF-Light GBM). Utilizing Borel Transform (BT)-based Wiener Filter (BWF) and resizing, the input images are initially pre-processed by eliminating the noise in the proposed system. After that, by utilizing the Orientation Preserving Simple Linear Iterative Clustering (OPSLIC), the pre-processed images, partitioned into a number of grids, are segmented. Next, as of the segmented images, the significant features are extracted along with the feature's distance is calculated and matched with the input images. Next, utilizing the Union Topological Measure of Pattern Diversity (UTMOPD) method, the false positive matches that took place throughout the matching process are eliminated. After that, utilizing the QMF-Light GBM visualization, the visualization of forged in conjunction with non-forged images is performed. The extensive experiments revealed that concerning detection accuracy, the proposed system could be extremely precise when contrasted to some top-notch approaches.

Keywords: Borel transform based wiener filter (BWF); orientation preserving simple linear iterative clustering (OPSLIC); keypoint features; block features; outlier detection

1 Introduction

In recent days, as several images have been encountered in each day of life, digital media has evolved into a vital aspect of living. In a broad range of industries, images play a vital role in communicating valuable information [1]. It is extremely intricate to express complex information (scenes) in words, whereas the images are influential and effective to pass intricate information [2]. In these days, as there is a huge amount of image processing and editing software, images are easily handled and modified. The mechanism of moulding, modifying, or copying images and documents is proffered as Forgery [3]. For exemplar, a critical power accident occurs in the power maintenance sector owing to the forgery of related images. During the court trials, the criminals do not get the punishment they deserve for the crime in forgery of scene images, which creates a great impact on the social harmony and steadiness [4]. Copy-move, splicing, and retouching are several forms of image forgery. In the image splicing technique, to create the forged images, specific portions from diverse images are spliced. To execute image retouching, the geometric transformation techniques say flipping, rotation, skewing, stretching, and scaling are incorporated [5]. An image's particular section is copied from one location, along with pasted, in a dissimilar location on to the same image in Copy-Move Forgery (CMF) [6,7]. An image's one section is copied together with background and moved to the same image's different areas in CMF [8]. Prior to pasting elsewhere, the duplicated portion may undergo some alterations, say scaling and modifications in brightness [9]. By the procedure of copy-move, an image may be counterfeited to hide or alter its sense [10,11]. It is complex to examine CMF with the naked eye if done carefully despite being one of the prevalent image modifications, along with it is intricate to prove the images' legitimacy for the forensic analyst [12,13]. So owing to these problems, the security and trustworthiness of media information are damaged [14]. There is an acute requirement for creating methodologies that are competent in checking and certifying digital original images owing to the rising prevalence of image falsification [15]. To discover image misrepresentation, numerous developments have been researched and proposed. Keypoint-based and Block-based approaches are the classifications of the CMFD technique. A general processing frame followed by the '2' CMFD classifications are; Feature Extraction (FE), Feature Matching (FM) and post-processing [16]. Images are then separated into several image blocks in block-based CMFD techniques, from which features are extracted. From an image, the key points are retrieved, and after that, the descriptors utilize these key points for FM in key-based CMFD techniques [17]. But, particularly in the FM stage, the time intricacy is large for prevailing methodologies, and the place of forged sections is not precise to meet practical needs. This work designed an efficient CMFD methodology utilizing the QMF-Light GBM, to subdue the prevailing difficulties.

The remaining work is structured as: The prevailing work associated with CMFD is represented in Section 2. The forgery detection method proposed is illustrated in Section 3. The outcomes along with discussions are analysed in Section 4. The conclusion is illustrated in the last section.

2 Literature Survey

Meena et al. [18] suggested a CMFD scheme, utilizing the Tetrolet transform. Primarily, image inputted is separated into intersecting blocks. After that Tetrolet transform extracted '4' low-pass coefficients along with '12' high-pass coefficients from every extracted segments. Following that, the feature vectors were lexicographically arranged, and associated blocks were discovered by correlating with the retrieved Tetrolet features. Though the copied portions had undergone certain post-processing activities, the method discovered and located the counterfeited portions correctly, as revealed by the experiment's outcomes. But, a large computational difficulty was the limitation here.

Meena et al. [19] presented a cross method by combining the block-based and keypoint-based method utilizing Fourier-Mellin Transform (FMT) and Scale Invariant Feature Transform (SIFT) respectively.

Primarily, image under investigation was split into texture and smooth portions. The SIFT was then utilized to extract key points as of the image's texture area. After that, for the image's smooth portion, the FMT was adapted. Next, to discover the image's copied portions, the extracted characters were matched. On comparing the prevailing CMFD methodologies, the investigation outcomes showed that this system performed better. For greatly distorted images, this system was ineffective. Wang et al. [20] created a rapid and effectual CMFD utilizing adaptive keypoint extraction along with processing, initiating the quick tough invariant feature, and screening out incorrect duals. Initially, using quick approximated LoG filter and uniformity processing, the constant distribution keypoints were retrieved flexibly as of the forged image. The rapid robust invariant attribute was then utilized for defining the image keypoints, which were then matched via the Rg2NN algorithm. Ultimately, by utilizing optimized mean-residual normalized production correlation, the duplicated portions were localized, and the incorrectly matched pairs were eliminated by implementing the segmentation-based candidate clustering. When analogized with the prevailing methodologies, the comprehensive research exhibited the method's efficiency. Yet, in smooth images, the approach was unsuccessful in identifying forgery.

Hegazi et al. [21] designed an enhanced method for keypoint-based CMFD. Density-oriented clustering and the Guaranteed Outlier Removal algorithm were utilized to build the approach. The faked patch was recognized more precisely whilst lowering time along with space difficulty by exploiting density-based grouping algorithm. To minimize wrong matches more efficiently, the Guaranteed Outlier Removal (GORE) algorithm was used along with the RANdom SAmple Consensus (RANSAC) algorithm. Given multiple challenging situations, the methodology transcended the prevailing methods and this was revealed from the outcomes that were executed on several benchmark datasets. The RANSAC method was utilized to eradicate the wrong matches, but it only decreased the wrong matches and failed to produce accurately matched keypoint pairs. Zhu et al. [22] used Adaptive attention and Residual Refinement Network (AR-Net) to investigate end-to-end neural network. Particularly, the adaptive attention method was employed to amalgamate the position along with channel attention attributes, to completely get context details together with enhancing the feature's representation. Next, to gauge the self-correlation among feature maps, along with Atrous Spatial Pyramid Pooling (ASPP), deep matching was employed and correlation maps scaled were merged to construct the abrasive mask. Lastly, residual refinement segment was used to optimize the abrasive mask that preserved object's boundary structure. It was evident from the experimental findings that the AR-Net outperformed the prevailing algorithms. But, the technique took a long time for computation.

Qadir et al. [23] described a passive block-based technique for identifying CMF in images, which were distorted by post-processing threats like compression and sound. The approach functioned by separating the image into intersecting blocks, and then extracting features of these individual blocks utilizing Discrete Cosine Transform (DCT). Then, to build feature vectors regarding the DCT coefficient's sign information, the Cellular Automat was utilized. Lastly, to discover the forged parts, matching of feature vectors was done utilizing the KD-tree-based nearest-neighbour searching technique. Though an image was extremely damaged by post-processing attacks, the outcomes illustrated that the suggested method performed extraordinarily well when analogized with the prevailing techniques. But, owing to computation of feature vectors and more images blocks being matched, the method took high cost for computation.

3 Proposed Copy-Move Forgery Detection System

Owing to advancement of image editing software, there has been a rapid increase in image manipulation; in addition, with this enhancement, the image falsification is performed devoid of degrading its quality or leaving any traces. The procedure of copying an image's particular portion and fixing it in the same image's some other portion is mentioned as CMF, which is an extensively utilized forgery model. The

major intention of such tampering is to misinform people by hiding some valuable data or by replicating things. To identify CMF, a huge number of methodologies have been presented, which primarily focussed on two categories: (i) centred on the block feature and (ii) centred on the key-point feature. Initially, the forged image is separated into overlapping or non-overlapping blocks in the block-centric model. Afterwards, the FE along with FM is executed. Conversely, the keypoint-centred methodology is reliant on the identification of higher-entropy image parts. In an image, the candidate key points are the pixels' local maxima along with minima being extracted. There are few cons to the prevailing forgery detection methodologies. Therefore, producing an effectual CMFD model to detect along with to locate the forgery in digital images is highly essential. Consequently, for effectual CMFD, the QMF Light GBM model is proposed here. In the proposed framework, for the FM process, the block along with key-point features are extracted to make out the images being forged; then, to take away the false positive matches, which advanced in the FM, the outliers are detected utilizing the UTMOPD; finally, to visualize if the image is forged or not, the visualization is done utilizing the QMF-Light GBM. Fig. 1 demonstrates the proposed model's block diagram.

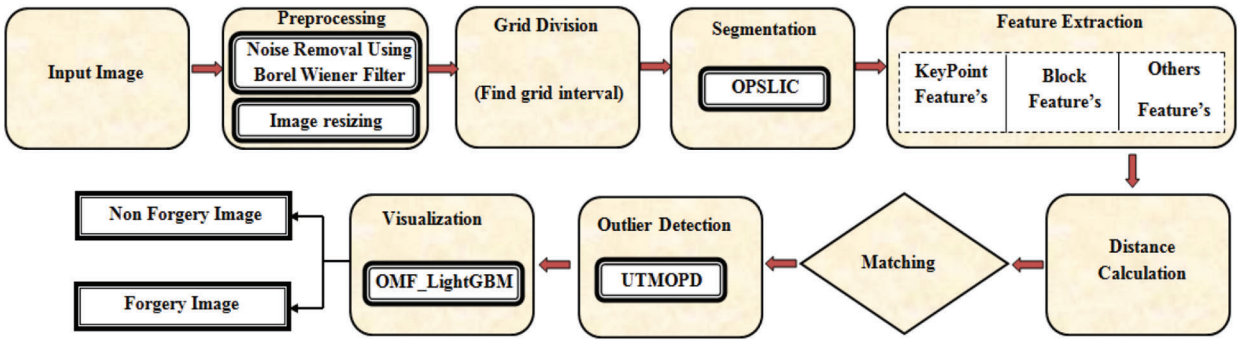

Figure 1: Proposed QMF_light GBM methodology block diagram

3.1 Pre-processing

Enhancing image's quality by suppressing unnecessary distortions ameliorating certain features is the intention of pre-processing; thus, the image can be evaluated effectively. The pre-processing includes conversion, image resizing, filtering, together with brightness transformation. Noise removal and image resizing are the two steps utilized for pre-processing the input images, which are initialized as,

$$E_n = \{E_1,\ E_2,\ E_3,\ \ldots\ldots\ldots E_N\} \tag{1}$$

where, the number of input images is specified as E_n.

Then, in the noise removal process, the noise is removed utilizing the BWF. The received signal together with the estimated noise-removal signals is analogized to alleviate the noise by the Wiener Filter. The frequency components degraded by the noise aren't reconstructed, and also the filters are not capable of restoring components in the conventional Wiener Filter. The Fourier transform is replaced by the BT in the BWF to trounce the aforementioned issues. The noise removed image $E_{n(NR)}$ is specified as,

$$E_{n(NR)} = WF(E_n).RS(E_n) \tag{2}$$

where, the received signal is specified as $RS(E_n)$. The Wiener filter adopted for the input image $WF(E_n)$ is signified as,

$$WF(E_n) = \frac{BT''(E_n)}{|BT(E_n)|^2 + \frac{\Delta_s(E_n)}{\Delta_N(E_n)}} \tag{3}$$

where, the BT is specified as $BT(\bullet)$, the point spread function's inverse BT is signified as $BT(\bullet)$, the power spectrum of signal process is symbolized as Δ_s, together with the power spectrum of noise process is notated as Δ_N. The BT is signified as,

$$BT(E_n) = \int_0^\infty T(u)e^{-E_n u}du \tag{4}$$

where, the complex function attained utilizing the complex variable u is denoted as T. Next, the noise-removed images $E_{n(NR)}$ are resized to a regular size of 512×512, which is provided as,

$$E_{n(NR)} \overset{resizeto512\times512}{\longrightarrow} E_{n(512\times512)} \tag{5}$$

where, the resized image is notated as $E_{n(512\times512)}$.

3.2 Segmentation

The 512×512 images $E_{n(512\times512)}$ are partitioned into a number of grids following the completion of pre-processing. Next, by utilizing the OPSLIC, the images are segmented. SLIC is broadly adopted in super-pixel clustering in which by employing the requisite number of equivalent-sized super-pixels, various boundaries in the image are well-preserved. A group of pixels possessing the same pixel intensity values is mentioned as super-pixel. Every single pixel is consigned with a label in segmentation. Certain characteristics are shared by pixels with the same label. The covariance matrix's volatility is a difficulty faced by the earlier SLIC. In the OPSLIC, the orientation preserving distance is utilized to conquer this issue.

Firstly, the input image is transformed to the CIELAB Color Space (CS) in which by utilizing all pixels of image along with the magnitude of super-pixels, the image's grid interval is calculated. A uniform CS is provided by CIELAB by illustrating all perceivable colors mathematically in '3' dimensions namely L for lightness, A as well as B for color opponents. Thus, for the provided image, the grid image is calculated as,

$$\tau = \sqrt{\frac{M}{N}} \tag{6}$$

where, the grid interval is denoted as τ, the number of super-pixels is illustrated as M, the total number of pixels in the image is symbolized as N. By sampling pixels at regular grid intervals, the cluster centres are initialized as,

$$\phi_M = (L_M,\ A_M,\ B_M,\ X_M,\ Y_M) \tag{7}$$

where, the pixel coordinates in the image are specified as X, Y, the cluster centre is indicated as ϕ_M. Regarding the lower gradient position in the 3×3 neighbourhood, the cluster centres are moved to the seed locations. The gradients are calculated as,

$$Gd(X,\ Y) = ||E(X+1,\ Y) - I(X-1,\ Y)||^2 + ||E(X,\ Y+1) - I(X,\ Y-1)||^2 \tag{8}$$

After that, the distance betwixt every single pixel in the search region $2\tau \times 2\tau$ with its corresponding cluster centre is measured as,

$$R = R_{LAB} + \frac{l}{\tau} R_{XY} \tag{9}$$

$$R_{XY} = \sqrt{(X_M - X_n)^2 + (Y_M - Y_n)^2} \tag{10}$$

$$R_{LAB} = 1\sqrt{n}||(LAB \in (L_M,\ A_M,\ B_M) - LAB \in (L_n,\ A_n,\ B_n)||^z,\ z = 2\tau \times 2\tau \tag{11}$$

where, distance betwixt coordinates of every single pixel and the cluster centre is exhibited as R_{XY}, the orientation preserving distance betwixt pixel and the cluster centre is depicted as R_{XY}.

Every single pixel n is connected with the closest cluster centre regarding the distance value. Next, the new cluster centre is calculated as the mean $(L_M,\ A_M,\ B_M,\ X_M,\ Y_M)$ of the entire pixels in the cluster. Then, regarding the variation betwixt the preceding centres, the residual error is calculated until the error doesn't go beyond the threshold. Fig. 2 illustrates the proposed OPSLIC algorithm's pseudo-code.

Input: *preprocessed images* $E_{n(512\times512)}$
Output: *Segmented image* $E_{seg(n)}$

Begin

Initialize Cluster Centres $\varphi_M = (L_M, A_M, B_M, X_M, Y_M)$
Move φ_M to lower gradient position
For each cluster centres φ_M **do**
{
Compute distance between each pixel in $2\tau \times 2\tau$ search region
Assign pixels to the nearest cluster centre using $R = R_{LAB} + \frac{l}{\tau} R_{XY}$
} **End for**
Compute new cluster centre
Compute residual error
Until *residual error* $\leq$ *Threshold*
Return the segmented image $E_{seg(n)}$
End

Figure 2: Pseudo-code of the OPSLIC algorithm

The basic steps in the OPSLIC are elaborated in Fig. 2. In this, by grouping pixels, which share the same characteristics into regions, the segmented images are generated together with it is specified as $E_{seg(n)}$.

3.3 Feature Extraction

Following segmentation, the FE is executed wherein the segmented images $E_{seg(n)}$ are utilized. The features needed to detect the copy-move images are obtained with the assist of FE. Key point features, block features, and other features are the '3' categories of features being extracted.

- *Key Point Features:* Information about the image content is termed as a key point feature. It provides data about the image's structures like edges, points, or objects. These features are calculated by pondering a region of certain pixel intensities in the region of it. Corner, edge, Difference of Gaussian (DoG), Laplacian of Gaussian (Log), dense-field features, Determinant of Hessian (DoH), Oriented Fast and Rotated Brief (ORB), sparse-field features, Binary Robust Independent Elementary Features (BRIEF) are keypoint features being extracted. Consequently, the feature is specified as,

$$\nabla_{n(kpf)} = \{\nabla_1,\ \nabla_2,\ \nabla_3,\ \ldots\ldots\ldots\ldots,\ \nabla_n\} \tag{12}$$

- *Block Features:* The block features comprise the linear amalgamation of actual features. Discrete Wavelet Transform (DWT), DCT, Singular Value Decomposition (SVD), and Principal Component

Analysis (PCA) are the block features being extracted. The block features being extracted are indicated as,

$$\nabla_{n(bf)} = \{\nabla_1,\ \nabla_2,\ \nabla_3,\ \ldots\ldots\ldots\ldots,\ \nabla_n\} \tag{13}$$

- *Other Features*: Illumination features, multiscale features, content-centric image features, and geometric features like scaling along with rotation, SIFT are the other features being extracted. These features are signified as,

$$\nabla_{n(of)} = \{\nabla_1,\ \nabla_2,\ \nabla_3,\ \ldots\ldots\ldots\ldots,\ \nabla_n\} \tag{14}$$

Lastly, the features being extracted are formulated as,

$$\nabla_{n(E)} = \{\nabla_{n(kpf)},\ \nabla_{n(bf)},\ \nabla_{n(of)}\} \tag{15}$$

where, the total number of features including keypoint features $\nabla_{n(kpf)}$, block features $\nabla_{n(Bf)}$, and other features $\nabla_{n(of)}$ extracted as of the segmented image $E_{n(seg)}$ is symbolized as $\nabla_{n(E)}$.

By utilizing the distance measure, the similarity betwixt every single feature descriptor is measured following the completion of FE. After that, to identify the forged region, the copy-move along with actual input images' distance value is matched. After FM, there may occur some false positive matches that specify an image as a forgery image even when the image is a non-forgery image. Consequently, by utilizing the UTMOPD, the outlier detection is conducted further to avoid such false matches termed outliers.

3.4 Outlier Detection

By utilizing the UTMOPD, the outlier detection is performed for the image $E_{matched(n)}$ with the respective matched pixel values. TMOPD is a local technique centred on the pattern (sample labels) diversity measure in a set of topological neighbourhoods of every single sample. '3' major steps are included in this to make out the outliers. In the 1st step, by utilizing the graph, which denotes the samples' mutual k-nearest neighbour, the samples' local structure is encoded. In the 2nd step, the graph communities are gauged. As of the communities, the outliers are recognized in the 3rd step. The problems rising as of a huge number of matching points are not addressed by the traditional TMOPD. The graph construct step is substituted by the union operation in the UTMOPD to conquer the aforementioned problems.

3.4.1 Graph Construction

The adjacency matrix of the set of samples' mutual k-nearest neighbour is plotted in the graph, which is signified as,

$$g(E_{matched(n)}) = (J,\ B) \tag{16}$$

where, the set of $k-$ nearest neighbour of samples is denoted as J, the adjacency matrix is indicated as B. J as well as B are proffered as,

$$J = \{a_i,\ lb(a_i)\} where,\ mkNN(a_i) = \{a_j such that a_j \in kNN(a_i),\ a_i \in kNN(a_j)\} \tag{17}$$

$$B = (b(a_i,\ a_j)_{i,j} = b_{ij}) \tag{18}$$

where, set of labelled points a_i in the feature space is depicted as $lb(a_i)$, the distance of the labelled points, and the mutual $k-$ nearest neighbor of a_i is notated by the relation of $a_j \in kNN(a_i),\ a_i \in kNN(a_j)$. After that, the graph's encoding is performed regarding,

$$b_{ij} = \begin{cases} \dfrac{1}{\cup(a_i,\ a_j)+1} & if\,(a_j \in mkNN(a_i) \\ 0 & otherwise \end{cases} \tag{19}$$

where, the union of $k-$ nearest neighbours is indicated as $\cup$. The graph edges are coded as 1 whilst $\cup$ is maximum, otherwise, they are coded as 0 in the encoding process. By this process, the number of neighbours for every node a_i is associated.

3.4.2 Computation of Graph Communities

A group of nodes, which have additional links connecting nodes of the same group, is mentioned as communities. Here, regarding the maximal union of adjacent $k-$ nearest groups, the initial communities are calculated. Let P denotes the set comprising all sets of nodes, which belongs to any of the discovered communities; in addition, the sub-graphs generated by P and T are symbolized as $g_P,\ g_T$. After that, the function of the degree of community $\Re$ proffers every single node's belongingness to the g_T community. It is notated as,

$$\Re(g_T) = \frac{dg_{g_T}\{a_i\}}{dg_{g_P}\{a_i\}} \tag{20}$$

As of the degree function, the internal connectivity of g_T is calculated as,

$$ic(g_T) = \sum\nolimits_{a_i \in g_T} \Re(g_T) \tag{21}$$

Next, the connectivity betwixt the isolated node and g_T is gauged as,

$$con(g_T),\ a_{iso(i)}) = \sum\nolimits_{a_i \in g_T} \frac{\Re(g_T)}{d(a_i,\ a_{iso(i)})+1} \tag{22}$$

Consequently, the isolated node is appended to the primarily identified community when it meets the criterion,

$$con(g_T,\ a_{iso(i)}) \geq \underset{i,iso(i)}{max} \frac{1}{d(a_i,\ a_{iso(i)})+1}(ic(g_T)) \tag{23}$$

where, the term $\underset{i,iso(i)}{max} \frac{1}{d(a_i,\ a_{iso(i)})+1}$ is the tolerance parameter. Like so, the sets of nodes that belong to a particular community with a common function regarding the internal connectivity are recognized.

3.4.3 Classification of Inliers and Outliers

In this, the inliers along with the outliers are distinguished and mapped to the square function's various corners. It is proffered utilizing the local measure of irregularity as of '2' probabilistic measures gauged in its set of neighborhoods. Class outliers and attribute outliers are the '2' types of outliers. Attribute outliers are specified as the nodes not belonging to any of the identified communities. Class outliers are the nodes that are expected to belong to communities with a higher heterogeneity in nodes label or belong to the majority of nodes with a label varied as of the class outlier label. The heterogeneity of the community that sample belongs to is gauged as,

$$\Omega = \{ \begin{matrix} J \rightarrow [0,\ 1] \times [0,\ 1] \\ a_i,\ lb(a_i) \rightarrow (\Omega_1(a_i,\ lb(a_i)),\ \Omega_2(a_i,\ lb(a_i)) \end{matrix} \tag{24}$$

where, heterogeneity of community labels utilizing the probability of g_T is quantified as Ω_1, the number of nodes whose label varies from the sample label utilizing the probability of $lb(a_i)$ is quantified as Ω_2.

$\Omega_1 \in (0,\ 1)$ (When the complete communities possess heterogeneous labels, then $\Omega_1 = 0$ and when the label within every single community is the same for the whole nodes in the community, then $\Omega_1 = 1$). $\Omega_2 \in (0,\ 1)$ (When no community is reliable with the sample label, then $\Omega_2 = 0$ and when the whole communities possess nodes with labels equivalent to the sample label, then $\Omega_2 = 1$). The inliers are mapped to $(1,\ 1)$, outliers are mapped to $(0,\ 0)$ and the class outliers are mapped to $(0,\ 1)$ regarding the belongingness of every single node to a respective community. Consequently, after removing the outliers, the output image is illustrated as $E_{OR(n)}$.

3.5 Visualization

The images $E_{OR(n)}$ are given to the visualization phase after filtering the outlier. In this, by utilizing the QMF-Light GBM, the input image is visualized whether forged or not. Light GBM is a gradient boosting algorithm. In this, a tree-centric learning model is utilized in which the weak learner develops sequentially. In Light GBM, the 1st tree finds out how to fit the target variable; the 2nd tree learns from the 1st tree along with it also determines how to fit the residual; the subsequent tree finds how to mitigate the residual and fit the residual as of the preceding tree together with it continues until the residual doesn't alter. Here, the updation procedure relies on the loss function. The training is continued if a minimum loss function is obtained; conversely, if it is maximum, then to mitigate the training time along with to ameliorate the accuracy, the optimized weight parameter is generated utilizing the QMF. Fig. 3 depicts the QMF-Light-GBM's architecture.

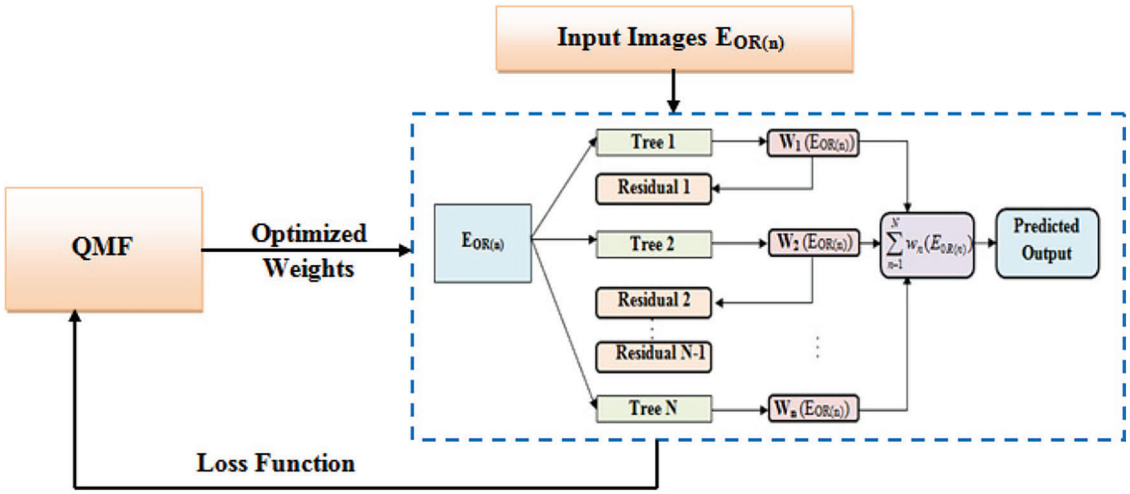

Figure 3: Architecture of QMF-Light GBM

Numerous regression trees are incorporated to make a Light GBM as,

$$D_m(t) = \sum_{n=1}^{N} W_n(E_{ol(n)}) \tag{25}$$

where, the prediction after a regression tree W_n included is specified as $D_m(t)$. The weighted sum of the predictions made by the preceding tree is the final ensemble model's predictions. A new regression tree is appended by minimizing the objective in every iteration.

$$W_n = \underset{W_n}{argmin} \sum_{m=1}^{M} \psi(D_m,\ D_m(t)) + \square(W_n) \tag{26}$$

where, the loss function gauged as the difference betwixt the prediction $D_m(t)$ and target D_m is signified as $\psi(\bullet)$, the regularization function is exhibited as Φ. The regularization function is proffered as,

$$\Phi(W_n) = \chi h + \frac{\varpi}{2} \sum_{l=1}^{L} \theta_l^2 \tag{27}$$

where, the parameter that penalizes the number of leaves h is denoted as χ, the weight of leaves h is exhibited as θ. The squared error loss function is notated as,

$$\psi = [res - W_n(E_{ol(n)})]^2 \tag{28}$$

where, the residual employed to fit the decision tree W_n is demonstrated as $res = (D_m(t) - D_m(t-1))$. By utilizing the QMF, the weight values are optimized if the maximum loss function is achieved.

Moth Flame Optimization (MFO) is a population-centred meta-heuristic algorithm motivated by the moths' navigation behaviour in light. In MFO, by exchanging the position vectors, moths can fly in a hyper dimension space. Creating the initial population, updating the moth positions, and updating the number of flames are the '3' major steps included in the MFO. Quick-sort is utilized by the prevailing MFO. The Quick-sort is extremely slow; thus, it is replaced by Quad-sort. Quad-sort is faster than Quick-sort for random data and for ordered data, it slightly faster than Tim-sort. Consequently, the algorithm is called as QMF.

3.5.1 Population Initialization

The moths' populace (i.e., the weight of leaves to be optimized) is initialized as,

$$\theta = \begin{bmatrix} \theta_{1,1}\theta_{1,2} \ldots..\theta_{1,t} \\ \theta_{2,1}\theta_{2,2} \ldots..\theta_{2,t} \\ \vdots\vdots\vdots \\ \theta_{l,1}\theta_{l,1} \ldots..\theta_{l,t} \end{bmatrix} \tag{29}$$

where, the number of moths is indicated as l, the number of parameters is specified as t. After that, regarding the fitness value, all moths are sorted as,

$$\delta(\theta) = \begin{bmatrix} \delta(\theta_1) \\ \delta(\theta_2) \\ \vdots \\ \delta(\theta_l) \end{bmatrix} \tag{30}$$

where, the fitness function is denoted as δ. Just like the population matrix, the flames which are mentioned as the QMF's components is illustrated as,

$$\wp = \begin{bmatrix} \wp_{1,1}\wp_{1,2} \ldots..\wp_{1,t} \\ \wp_{2,1}\wp_{2,2} \ldots..\wp_{2,t} \\ \vdots\vdots\vdots \\ \wp_{l,1}\wp_{l,1} \ldots..\wp_{l,t} \end{bmatrix} \tag{31}$$

Moreover, regarding the fitness value, the flames are sorted as,

$$\delta(\wp) = \begin{bmatrix} \delta(\wp_1) \\ \delta(\wp_2) \\ \vdots \\ \delta(\wp_l) \end{bmatrix} \tag{32}$$

where, the flames are indicated as $\wp$. In this, the moths are the search agents, which move around the Search Space (SS), and the flames are the moths' best position achieved until now in the SS.

3.5.2 Moth's Position Updation

'3' significant functions are deployed here and are given as,

$$QFM = \{\alpha,\ \beta,\ \gamma\} \tag{33}$$

where, the moths' random population along with fitness value is specified as α, the moths' motion in the SS is determined as β, the function verified to complete the searching process is signified as γ. It can be expressed as,

$$\alpha = \varsigma \rightarrow \{\theta,\ \wp(\theta)\} \tag{34}$$

$$\beta = \theta \rightarrow \theta \tag{35}$$

$$\gamma = \theta \rightarrow T,\ F \tag{36}$$

The significant function that moves the moths around the SS is specified as β; it also updates every single moth's position until the γ returns T. Consequently, the β function updates every single moth's position regarding the flame as,

$$\theta_l = \mathfrak{Z}(\theta_i,\ \wp_j) \tag{37}$$

where, the spiral function is symbolized as $\mathfrak{Z}$. It permits a moth to fly "around" a flame and not in the space betwixt them. It is proffered as,

$$\Im(\theta_i,\ \wp_j) = C_j.exp(fs).cos(2\pi s) + \wp_j \tag{38}$$

where, the distance betwixt the i^{th} moth and j^{th} flame is represented as $C_j = |\theta_i - \wp_j|$, a constant is denoted as $f, s \in (1,\ -1)$ is the random number, which proffers how much the moth's subsequent position should be nearer to the flame. The balance betwixt exploitation and exploration is guaranteed by the moth's spiral motion close to the flame in the SS. In every single iteration, for the list of flames, the fitness value is analysed and it is sorted by utilizing the Quad-sort methodology regarding the fitness matrices. It is specified as,

$$\wp(QFM) = \wp(q(Quadsort(\wp) + \wp(\theta_l) \tag{39}$$

After that, by utilizing the fitness matrices, moths update their positions. This prevents from getting into the trap of local optima.

3.5.3 Number of Flames Updation

The number of flames is reduced as shown in Eq. (40) to ameliorate the QMF's exploitation.

$$num(\wp) = round\left((L - q) * \frac{L - q}{q_{max}}()\right) \tag{40}$$

where, the maximum number of flames is denoted as L, the current iteration is specified as q, the maximum number of iterations is signified as q_{max}. The SS's exploration and exploitation is balanced by the gradual decrement in the number of flames. Similarly, to mitigate the loss function, the QMF moves the Light GBM's weights along with the search for the weights, which are superior to the random solutions. Consequently, the optimal solution acquired as of the QMF is notated as,

$$\theta_{l(opt)} = \begin{bmatrix} \theta_{opt(1,1)}\theta_{opt(1,2)} \ldots\ldots\theta_{opt(1,t)} \\ \theta_{opt(2,1)}\theta_{opt(2,2)} \ldots\ldots\theta_{opt(2,t)} \\ \vdots\vdots\vdots \\ \theta_{opt(l,1)}\theta_{opt(l,1)} \ldots\ldots\theta_{opt(l,t)} \end{bmatrix} \tag{41}$$

where, the optimal weight values utilized to train the Light GBM is symbolized as $\theta_{l(opt)}$. Thus, by utilizing the weight values, the loss function is minimized to achieve the new tree. Subsequently, the objective function becomes,

$$W_n = \underset{W_n}{argmin} \sum\nolimits_{m=1}^{M} \vartheta_m W_n(E_{ol(n)}) + \frac{1}{2}\vartheta_m W_n^2(E_{ol(n)})) + \chi h + \frac{\varpi}{2}\sum\nolimits_{l=1}^{L} \theta_{l(opt)}^2 \tag{42}$$

where, the 1st and 2nd order gradient statistical outcomes of ψ are indicated as ϑ_m. Furthermore, every single node is split by the decision tree with the highest data gained. In this, to split and grow the tree, the leaf that mitigates the loss is selected. Every single node's splitting is signified as,

$$splt(y) = \frac{1}{L_o}\left\{ \frac{\left(\sum_{E_{ol(n)} \in o \le y} \vartheta_m\right)^2}{L_{l|o}(y)} + \frac{\left(\sum_{E_{ol(n)} \in o > y} \vartheta_m\right)^2}{L_{l|o}(y)} \right\} \tag{43}$$

where, the point at which the node is split is denoted as y, the number of samples on a fixed node is indicated as o. This process is repeated until the residual remains constant. Lastly, the predicted output verifies whether the input image is forged or not.

4 Result and Discussion

To check the efficiency of the proposed model, several experimentations are executed here. With 4 GB RAM and 3.20 GHz Intel i5/core i7 CPU, the entire experiment has been conducted. The PYTHON is employed as the working platform to execute the proposed CMFD utilizing QMF-Light GBM.

4.1 Database Description

By utilizing two image datasets, MICC-F220 and MICC-F600, the proposed methodology's performance is assessed. MICC-F220 dataset contains 220 images, of which 110 are tampered and 110 of which are originals, with sizes from 722 × 480 to 800 × 600 pixels. The MICC-F600 dataset contains 440 original images, 160 tampered images, and 160 ground truth images with sizes ranging from 800 × 532 to 3888 × 2592 pixels. For training and testing, data of 80% and 20% are utilized in the proposed model. In Figs. 4 and 5, the dataset's sample images along with the outcome images of further processing are depicted.

The sample images' single copy-paste detection is addressed in Fig. 4. Dataset's input images are depicted in Fig. 4a, the pre-processed images are exhibited in Fig. 4b, the segmented images are demonstrated in Fig. 4c, and in Fig. 4d matching points together with detected copy paste section is presented. Multiple copy-paste detection in the sample image is interpreted in Fig. 5. The input image is

elucidated in Fig. 5a, the pre-processed image is portrayed in 5(b), the segmented image is illustrated in 5(c), and in Fig. 5d the matching points along with copy-paste portions are depicted.

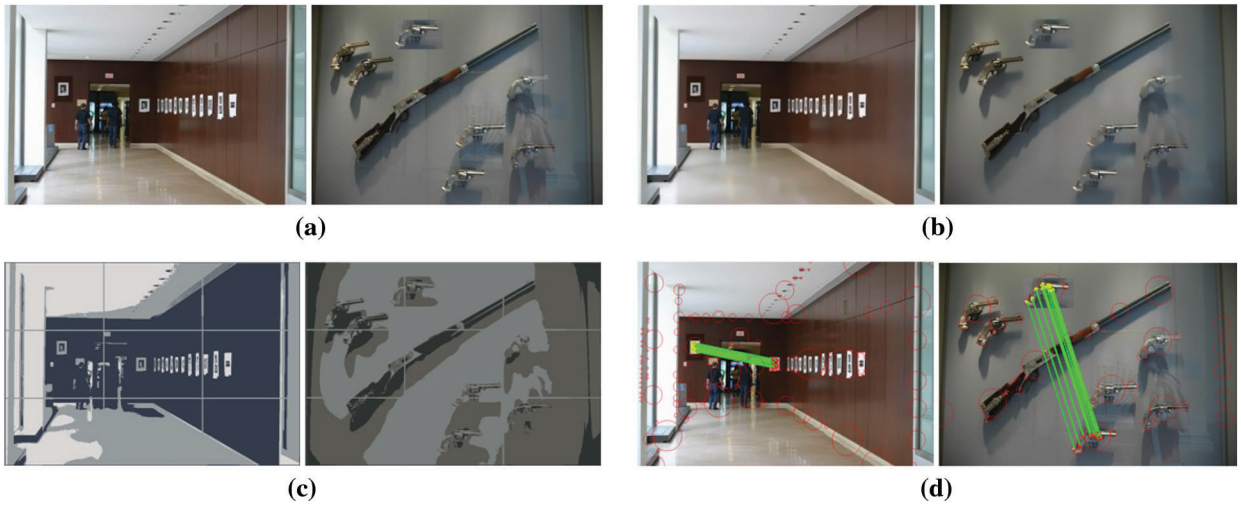

Figure 4: Sample images of single copy-paste (a) input images (b) pre-processed images (c) segmented images (e) images with matching points and detected copy paste part

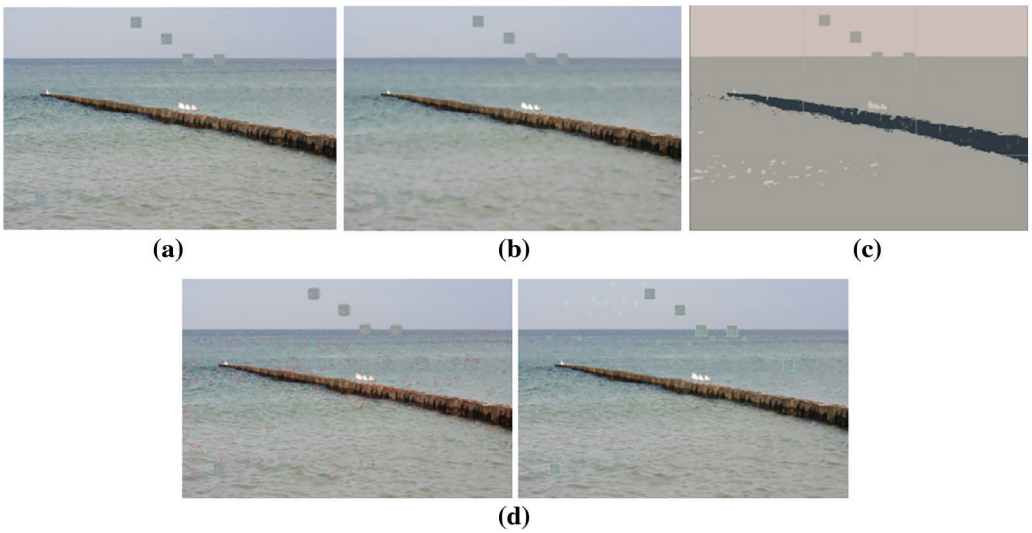

Figure 5: Sample images of multiple copy-paste (a) input image (b) pre-processed image (c) segmented image (d) images with matching points and detected copy paste part

4.2 Performance Analysis

By analogizing with the prevailing Recurrent Neural Network (RNN), Artificial Neural Network (ANN), Convolution Neural Network (CNN), along with Deep Neural Network (DNN), the QMF-Light GBM is evaluated here. To examine the proposed methodologies' performance, the parameters like specificity, sensitivity, accuracy, Negative Predictive Value (NPV), F-measure, False Positive Rate (FPR), Mathews Correlation Coefficient (MCC), False Negative Rate (FNR), as well as False Recognition Rate (FRR) are utilized.

Discussion: Regarding certain quality metrics, the proposed and prevailing technique's performance is demonstrated in Tab. 1. The fraction of accurate image detection at which an image suffering from forgery is predicted correctly as the image is forged is measured by sensitivity, accuracy, and specificity. These values must be high, to attain enhanced performance. By having greater specificity, sensitivity, and accuracy values, the proposed model acquires enhanced performance and it is demonstrated in Tab. 1. Sensitivity of 98.3857, specificity of 93.9134, and accuracy of 97.2222 are achieved by the proposed model. The prevailing approaches, however, have lesser sensitivity, accuracy, and specificity than the proposed methodology. As a result of the investigation, the proposed model is incredibly accurate in finding forged areas. Fig. 6 reflects the pictorial representation of the aforesaid analysis.

Table 1: Comparative analysis based on sensitivity, specificity, and accuracy

Methods	Sensitivity	Specificity	Accuracy
Proposed QMF-Light GBM	98.3857	93.9134	97.2222
CNN	96.4025	91.9168	95.5835
ANN	94.7632	89.4635	92.6962
RNN	92.4924	87.3415	90.2434
DNN	88.8432	84.9431	87.1521

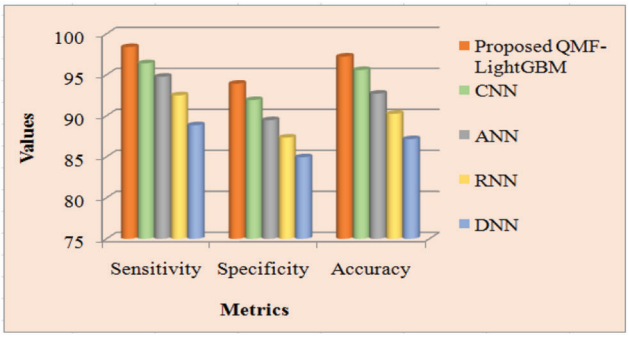

Figure 6: Performance analysis of proposed and existing methods

Discussion: The proposed along with the prevailing technique's performance is shown in Tab. 2. To determine analysis accuracy F-measure is utilized, which is a weighted average of precision and recall. The number of true negatives is stated by the NPV, which is the likelihood that the image is actually non-forged after a negative outcome. For two classes, the MCC assesses the visualization's quality. The greater the metrics' values, the more precise the technique is. Tab. 2 displays that the proposed system achieves larger MCC, F-measure, and NPV than the prevailing techniques. The proposed technique has an F-measure of 94.36557, but the prevailing CNN, ANN, RNN, and DNN approaches have F-measures of 92.74325, 89.84532, 88.29857, and 86.48322 respectively. The proposed framework has F-measure superior to the prevailing techniques. Regarding NPV and MCC, the proposed system outperforms the prevailing techniques by a greater value of 94.36557 and 92.9973 respectively. Therefore, when analogized to the prevailing techniques, the proposed model performs superior. Fig. 7 depicts the findings of the examination of Tab. 1 graphically.

Table 2: Comparative analysis based on F-measure, NPV, and MCC

Methods	F-measure	NPV	MCC
Proposed QMF-Light GBM	94.36557	94.36557	92.9973
CNN	92.74325	92.74325	90.7829
ANN	89.84532	89.84532	89.1975
RNN	88.29857	88.29857	88.4461
DNN	86.48322	86.48322	82.2158

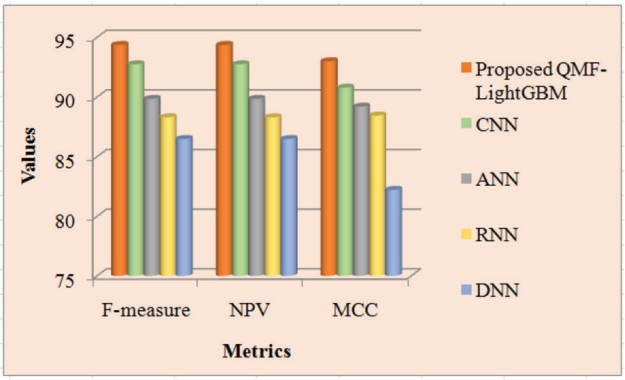

Figure 7: Performance comparison of proposed and existing methods

Discussion: The proposed and prevailing methodologies' FPR, FNR, and FRR are inspected in Tab. 3. FNR is the false-negative cases' appraisal, while FPR is the fraction of forged images wrongly observed as non-forged images. The metric that indicates how likely the system wrongly discards the forged images is termed FRR. It is claimed that the technique is highly accurate if it acquires lower values for these criteria. Thus, with the lesser values of FPR, FNR, and FRR, the proposed strategy produces superior outcomes, which is apparent from Tab. 1. The proposed methodologies' FNR and FRR are equal to and lesser than the prevailing techniques that are 0.02124, and it is revealed from Tab. 3. The proposed model offers a value of 0.01197 for FPR. However, the prevailing methodologies' FNR is 0.21101, 0.29859, 0.34262, and 0.51479 for CNN, ANN, RNN, and DNN respectively. The investigation reveals that the proposed approach has significantly lower FNR, FPR, and FRR values. Therefore, the acquired results revealed that the proposed system outperforms the prevailing CNN, ANN, RNN, and DNN approaches. Fig. 8 depicts a visual representation of the aforesaid analysis.

Table 3: Comparative analysis based on FPR, FNR, and FRR

Methods	FPR	FNR	FRR
Proposed QMF-Light GBM	0.01197	0.02124	0.02124
CNN	0.21101	0.21967	0.21967
ANN	0.29859	0.41102	0.41102
RNN	0.34262	0.49883	0.49883
DNN	0.51479	0.34678	0.34678

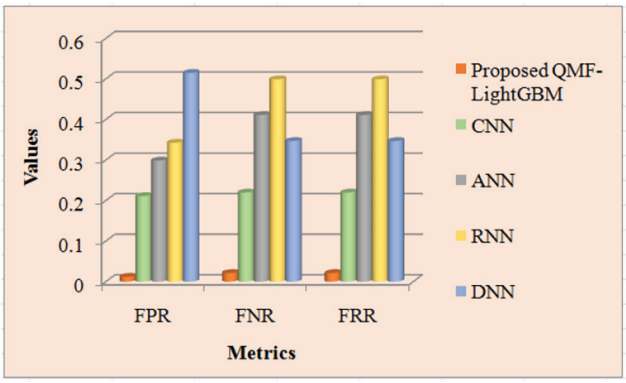

Figure 8: Performance analysis of proposed and existing methods

5 Conclusion

Since several transformations might be undergone by the forged region for making the forgery detection untraceable for human eyes, the CMF is highly complicated to identify. Therefore, a quick and precise CMFD is proposed for digital images. Detecting the forged parts efficiently utilizing a QMF-Light GBM is the paper's key intention. '6' stages namely, pre-processing, segmentation, FE, FM, outlier detection, and visualization are encompassed by the methodology. Concerning the performance metrics, the proposed QMF-Light GBM's performance is examined with the existent CNN, ANN, RNN, and DNN methodologies in the experiential assessment. The images gathered from MICC-F220 and MICC-F600 datasets are utilized by the proposed methodology for performance analysis. Experimental results revealed that copy-move forged regions could be very efficiently and accurately identified by the proposed model with an accuracy of 97.2222, which is higher contrasted to the prevailing methods. The method suffers still from high computation time for the higher number of key points detected in the process. For detecting CMF in videos, the proposed work could be extended in the upcoming future.

Acknowledgement: The authors with a deep sense of gratitude would thank the supervisor for his guidance and constant support rendered during this research.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] G. Singh and K. Singh, "An improved block-based copy-move forgery detection technique," *Multimedia Tools and Applications*, vol. 79, no. 1, pp. 13011–13035, 2020.

[2] A. Dixit and S. Bag, "Utilization of edge operators for localization of copy-move image forgery using WLD-HOG features with connected component labeling," *Multimedia Tools and Applications*, vol. 79, no. 3, pp. 26061–26097, 2020.

[3] R. Agarwal and O. P. Verma, "An efficient copy move forgery detection using deep learning feature extraction and matching algorithm," *Multimedia Tools and Applications*, vol. 79, no. 11, pp. 7355–7376, 2020.

[4] X. Tian, G. Zhou and M. Xu, "Image copy-move forgery detection algorithm based on ORB and novel similarity metric," *IET Image Processing*, vol. 14, no. 10, pp. 2092–2100, 2020.

[5] S. Tinnathi and G. Sudhavani, "An efficient copy move forgery detection using adaptive watershed segmentation with AGSO and hybrid feature extraction," *Journal of Visual Communication and Image Representation*, vol. 74, pp. 102966, 2021.

[6] M. Bilal, H. A. Habib, Z. Mehmood, R. M. Yousaf, T. Saba *et al.,* "A robust technique for copy-move forgery detection from small and extremely smooth tampered regions based on the DHE-SURF features and mDBSCAN clustering," *Australian Journal of Forensic Sciences*, vol. 53, no. 4, pp. 459–482, 2021.

[7] F. M. Al_Azrak, A. Sedik, M. I. Dessowky, G. M. El Banby, A. A. Khalaf *et al.,* "An efficient method for image forgery detection based on trigonometric transforms and deep learning," *Multimedia Tools and Applications*, vol. 79, no. 25, pp. 18221–18243, 2020.

[8] N. Goel, S. Kaur and R. Bala, "Dual branch convolutional neural network for copy move forgery detection," *IET Image Processing*, vol. 15, no. 3, pp. 656–665, 2021.

[9] P. Niyishaka and C. Bhagvati, "Copy-move forgery detection using image blobs and BRISK feature," *Multimedia Tools and Applications*, vol. 79, no. 35, pp. 26045–26059, 2020.

[10] A. Dixit and S. Bag, "A fast technique to detect copy-move image forgery with reflection and non-affine transformation attacks," *Expert Systems with Applications*, vol. 182, pp. 115282, 2021.

[11] S. Dua, J. Singh and H. Parthasarathy, "Image forgery detection based on statistical features of block DCT coefficients," *Procedia Computer Science*, vol. 171, pp. 369–378, 2020.

[12] E. A. Armas Vega, E. González Fernández, A. L. Sandoval Orozco and L. J. García Villalba, "Copy-move forgery detection technique based on discrete cosine transform blocks features," *Neural Computing and Applications*, vol. 33, no. 10, pp. 4713–4727, 2021.

[13] F. M. Al_azrak, Z. F. Elsharkawy, A. S. Elkorany, G. M. El Banby, M. I. Dessowky *et al.,* "Copy-move forgery detection based on discrete and SURF transforms," *Wireless Personal Communications*, vol. 110, no. 1, pp. 503–530, 2020.

[14] S. AlZahir and R. Hammad, "Image forgery detection using image similarity," *Multimedia Tools and Applications*, vol. 79, pp. 28643–28659, 2020.

[15] J. L. Zhong and C. M. Pun, "Two-pass hashing feature representation and searching method for copy-move forgery detection," *Information Sciences*, vol. 512, pp. 675–692, 2020.

[16] C. Wang, Z. Zhang, Q. Li and X. Zhou, "An image copy-move forgery detection method based on SURF and PCET," *IEEE Access*, vol. 7, pp. 170032–170047, 2019.

[17] H. Chen, X. Yang and Y. Lyu, "Copy-move forgery detection based on key point clustering and similar neighborhood search algorithm," *IEEE Access*, vol. 8, pp. 36863–36875, 2020.

[18] K. B. Meena and V. Tyagi, "A Copy-move image forgery detection technique based on tetrolet transform," *Journal of Information Security and Applications*, vol. 52, pp. 102481, 2020.

[19] K. B. Meena and V. Tyagi, "A hybrid copy-move image forgery detection technique based on Fourier-mellin and scale invariant feature transforms," *Multimedia Tools and Applications*, vol. 79, no. 11, pp. 8197–8212, 2019.

[20] X. Y. Wang, C. Wang, L. Wang, L. X. Jiao, H. Y. Yang *et al.,* "A fast and high accurate image copy-move forgery detection approach," *Multidimensional Systems and Signal Processing*, vol. 31, no. 3, pp. 857–883., 2020.

[21] A. Hegazi, A. Taha and M. M. Selim, "An improved copy-move forgery detection based on density-based clustering and guaranteed outlier removal," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 9, pp. 1055–1063, 2021.

[22] Y. Zhu, C. Chen, G. Yan, Y. Guo and Y. Dong, "AR-Net: Adaptive attention and residual refinement network for copy-move forgery detection," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6714–6723, 2020.

[23] G. Gani and F. Qadir, "A robust copy-move forgery detection technique based on discrete cosine transform and cellular automata," *Journal of Information Security and Applications*, vol. 54, pp. 102510, 2020.