Digital Image Authentication and Analysis: Unmasking Copy-move Forgery in Digital Images through Combined DCT and GLCM Features with Block Matching Technique

Prabhu Bevinamarad and Prakash H. Unki

***Abstract*—In the contemporary era, images hold an increasingly valuable trove of information and have become indispensable to current digital systems. Conversely, the accessibility of cost-effective electronic devices and advanced image editing tools has empowered many individuals to manipulate an image's meaningful content, fabricating counterfeit images to deceive society and government entities. This paper introduces a robust block-matching methodology that leverages composite features crafted through the Discrete Cosine Transform (DCT) and Gray-Level Co-occurrence Matrix (GLCM). The proposed approach uses the Stationary Wavelet Transform (SWT) to decompose an input image, effectively capturing intricate frequency details across scales and positions. Subsequently, the block-tiling procedure divides the approximation subband into overlapping blocks. Each of these blocks undergoes the extraction of DCT and GLCM image features, which are then amalgamated to form composite image features. Finally, the block-matching process is applied to find the suspected image blocks to effectively classify the input image as authentic or forged while concurrently identifying copy-move forgery regions within the query image. The effectiveness of this method was assessed on both image and pixel levels using the widely accessible CoMoFoD standard dataset by considering image samples of basic copy-move forgery and image forgery subjected to diverse post-processing operations. Based on the evaluation outcomes achieved at both image and pixel levels, the proposed method produces results with a precision of 97.50%, a recall of 92.26%, an F1-score of 95.12% and a precision of 93.40%, a recall of 90.18% and an F1-score of 92.86% for image level and the pixel level evaluation respectively.**

***Index Terms*—Block matching, Copy-move forgery, Image forensics, Image splicing, Stationary wavelet transform, and Post-processing operations.**

I. Introduction

NOWADAYS, multimedia information and digital gadgets have become ubiquitous and indispensable for individuals worldwide. The availability of well-equipped electronic devices in terms of hardware and software capabilities facilitates the end-user to capture, reproduce, manipulate, and transmit digital content such as text, images, audio, and video within no time. According to the statistics, over 300 million multimedia content is uploaded and circulated daily over social media to propagate information worldwide. In addition, many governments and private sector departments have launched various e-service facilities that require multimedia data as input to reduce the processing time and replace the traditional systems with automated systems with security measures[1]. However, many individuals in society are using advanced multimedia editing tools to damage the meaningful content of an image and practice unethical activities[2] instead of taking advantage of various graceful digital facilities provided by information technology and e-platforms. The sophisticated multimedia editing software is just like a two-edged sword. On the one hand, it encourages people to beautify and add creative artifacts to the multimedia content to express their innovative ideas. On the other hand, it makes Forger create tampered images, audio, and video by modifying its original content without much effort and leaving any noticeable trace despite many tampering detection techniques[3], [4].

In recent years, image exploitation has become more common. Digital images can be tampered with in various ways: copy-move, splicing, and image retouching[5]. Nevertheless, copy-move image forgery has become the most common, widespread, and challenging. It implies copying some part of the image region and pasting the cloned part in another area on the source image to corrupt its original meaning. Generally, the copy-move forged has no perceptive visual traces on its surface to check its authenticity and requires an intense searching technique to identify the correlated pixels. The forgery practitioners often apply extra post-processing steps (such as rotation, noise addition, blurring, etc.) to hide image counterfeiting and make the detection hard. Besides, the advances in image tampering techniques have made it easier for the forger to create realistic fake images and encourage others to misuse the tools.

Hence, a skillful image tampering detection technique is needed to search and identify manipulated images transmitted over the internet and other social platforms to prevent them from being used for improper purposes even when images are post-processed with significant post-operations performed. Our proposed approach uses DCT, GLCM combined image features, and a block matching technique to localize the forgery regions for basic copy-move forgery and forgery with post-processing operations incorporated. Our work's contribution can be summed up as follows:

- The proposed approach uses SWT for image decomposition that helps capture the most similar and discrim-

Manuscript received March 11, 2024; revised October 24, 2024.

Prabhu Bevinamarad is an Assistant Professor in the Department of Computer Science and Engineering, BLDEA's V.P. Dr. P.G. Halakatti College of Engineering and Technology (Affiliated to Visvesveraya Technological University, Belagavi-590018, Karnataka), Vijayapura, Karnataka 586103, India (E-mail: prabhubev@gmail.com).

Dr. Prakash H. Unki is a Professor and Head in the Department of Information Science and Engineering BLDEA's V.P. Dr. P.G. Halakatti College of Engineering and Technology (Affiliated to Visvesveraya Technological University, Belagavi-590018, Karnataka), Vijayapura, Karnataka 586103, India (Email:prakashhunki@gmail.com).

inative image characteristics and detect tampered areas in regular and irregularly shaped objects.
- The composite features formed using DCT coefficients and GLCM properties improved the forgery detection region under various post-processing attacks such as blurring, contrast augmentation, brightness change, and color dithering.
- The proposed technique uses different threshold parameters to segregate similar feature vectors effectively.
- We conducted a comparative analysis to assess the effectiveness of our proposed method considering plain image forgery images and forgery images with various post-processing operations.

The latter part of this paper proceeds as follows: Section II discusses the related work connected to copy-move forgery detection techniques. Section III explains the proposed approach with essential block and flow diagrams, Section IV details the setup for the experiment and evaluation, and Section V concludes the proposed method.

II. RELATED WORK

This section describes the efforts made by many researchers to develop various techniques to address the hurdles related to copy-move forgery. Below, we summarize techniques developed during the last decade to identify and mark the copy-move forgery practice. The first block-matching technique was developed by [6]; this paper employed a quantized DCT coefficient representation of each image block pixel to identify copy-move forged region present in an image. In[7], a technique based on Principal Component Analysis(PCA) was developed to reduce the image dimension and extract a feature to detect duplicate regions present in an image efficiently. In [8], the author has proposed a method based on blur moment invariants and the Principal Component Transform(PCT) to extract image features and detect duplicated regions. In the paper[9], the author represented relevant image features by employing Discrete Wavelet Transform(DWT), Singular Value Decomposition(SVD) and lexicographical sorting technique to reduce the time taken to find duplicate image regions. In the paper[10], the author exploited DCT and circular shapes to represent image blocks to extract the features and find duplicated regions present in an image. A similar technique is implemented in [11] where the author considered fixed-sized overlapping blocks and applied DCT and quantized by the quantization matrix. Finally, the quantized matrix is subdivided into non-overlapping blocks, and SVD features are extracted to identify region duplication in an image. In[12], the author presented a method to detect copy-move forgery by using features extracted from SWT and SVD to address the issue related to edges blurring and noise addition to forged images. In the paper[13], the author employs polar coordinate system representation to extract representative features for individual blocks. The primary part highlighted in this study is the block's frequency, which is determined through the Fourier transform. In[14], the author introduced a forgery detection approach that relies on both the DWT and the DCT for feature extraction and reduction. In[15], the author employed SWT for feature extraction, and SVD was used for feature reduction to detect forgery regions. In [16], use Discrete Stationary Wavelet Transform(DSWT) with Multi-Dimensional Scaling to identify recognizable instances of copy-move image tampering. The paper[17] used the Intensity Coherence Vector(ICV) feature extraction technique to detect free-form forgery images. The article[18] suggested Fractional Quaternion Cosine Transforms(FrQCT) for copy move forgery detection for color images. In[19], the DCT extracts features from each block of Cellular Automata(CA) to construct feature vectors based on the sign information of the DCT coefficients to detect a forgery. The author in[20]combined DCT and SVD for feature extraction and reduction, followed by Support Vector Machines(SVM) K-means clustering algorithm for forgery detection. [21]proposed a method for JPEG-compressed test images suspected of tampering by employing the DCT. The paper [22] presents a way to combine Speeded Up Robust Features(SURF) and Binary Robust Invariant Scalable Keypoints(BRISK) descriptors for the detection of Copy-move Forgery(CMF). The authors in [23] introduced low-dimensional DCT and DWT-based features for forgery detection. In[24] Scale Invariant Features Transform(SIFT) keypoints and DCT are combined for forgery detection. In [25], the authors employed DCT and CA to extract features from the image blocks. In paper [26], the double matching and region localizing processes are developed using Local Intensity Order Pattern(LIOP) key points and Density-Based Spatial Clustering of applications with noise for forgery detection. In the paper[27], the authors employed DWT and inverse-DWT for image forgery detection. In [28], the authors used a Binary Discriminative Feature(BDF) descriptor for feature extraction technique and a Color Histogram(CH) to detect forgery images and regions. The authors in the paper[29] utilized the Steerable Pyramid Transform(SPT) to decompose the suspected image, and GLCM features were extracted from each orientation to find the image forgeries. The paper[30] utilizes the Polar Complex Exponential Transform(PCET) to obtain the features from each overlapping block, followed by the Gradient Direction Pattern(GDP) histogram technique to reduce the dimension of extracted features to find the image forgery. The review of the most relevant existing forgery detection approaches is tabulated in Table VI by highlighting the feature extraction technique and significant remarks.

After reviewing various existing techniques, we have understood that the current approach employs diverse features constituting an extensive feature set, resulting in tedious matching and high computational complexity. Many passive forgery detection methods are limited in detecting forgeries in images without addressing various post-processing attacks, i.e., these methods struggle to identify forgery regions where forged images undergo substantial post-processing operations. Moreover, some of these methods could be more resilient to introducing noise and exhibit unsatisfactory performance. So, to tackle these issues, we propose a block-matching technique that primarily emphasizes constructing a composite feature vector by leveraging the DCT[6], [15] and GLCM[29] features, preceded by pre-processing using SWT. This innovative approach forms a robust image feature set and efficiently identifies copy-move forgery regions.

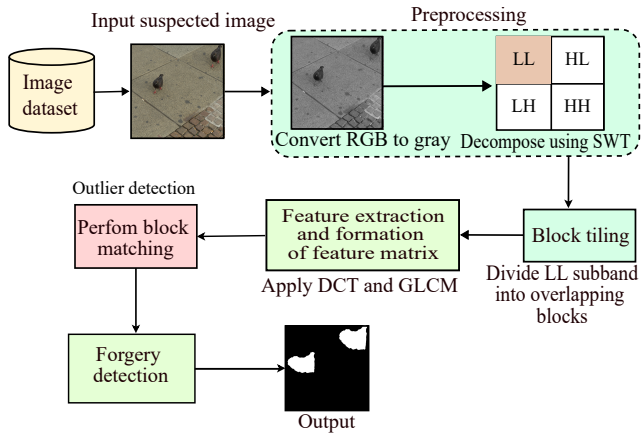


Fig. 1: Proposed Copy-move forgery detection approach

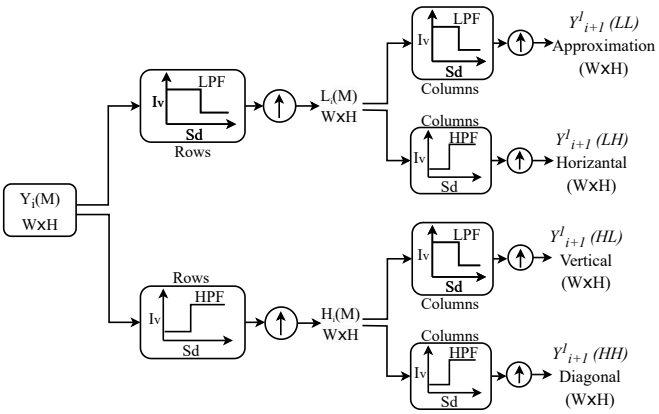


Fig. 2: First-level image decomposition using stationary wavelet transform (redrawing based on [15])

III. PROPOSED METHODOLOGY

The proposed approach utilizes an effective block-matching technique for identifying copy-move forgeries in digital images. It employs DCT[15] and GLCM[29] for feature extraction, creating composite feature vectors that serve as crucial artifacts for detecting forgery regions. The system's operation is divided into five main phases: 1) Pre-processing, 2) Block titling, 3)Feature extraction and formation of feature matrix, 4) Block matching, and 5) Forgery detection. A visual representation of the entire forgery detection system, encompassing these phases, can be found in Fig. 1. Additionally, Algorithm 1 provides a step-by-step pseudo code of the proposed forgery detection approach.

A. Pre-processing

The pre-processing step involves two steps; firstly, the input RGB query image is transformed into YCbCr color space to enhance image data and suppress distortion to extract the appropriate image features. Because the luminance channel (Y_i) is more sensitive to the human eye and carries detailed spatial information than other color spaces. The mathematical equations (1, 2, and 3) illustrate how to convert RGB to YCbCr color space and extract the luminance channel (Y_i) of an RGB image.

$$Y_i = 16 + \left(\frac{65.738R}{256}\right) + \left(\frac{129.057G}{256}\right) + \left(\frac{25.064B}{256}\right) \quad (1)$$

$$Cb_i = 128 - \left(\frac{37.945R}{256}\right) - \left(\frac{74.494G}{256}\right) + \left(\frac{112.439B}{256}\right) \quad (2)$$

$$Cr_i = 128 + \left(\frac{112.439R}{256}\right) - \left(\frac{94.154G}{256}\right) - \left(\frac{18.285B}{256}\right) \quad (3)$$

In the second step of our process, we employ the stationary wavelet transform to partition the input image into several frequency bands. Unlike the DWT, where the input image undergoes a halving of resolution at each level due to decomposition using both low and high pass filter banks, the SWT maintains the input image's resolution undecimated at each level, resulting in wavelet coefficients [31] that match the input size. Consequently, this resolves the issue of shift-invariance, meaning the coefficients remain unchanged even if the signal is shifted. This approach yields remarkable results, especially when forgery images undergo diverse post-processing operations. Therefore, in our approach, we apply the SWT to the luminance channel (Y_i) and execute a first-level decomposition, resulting in four frequency subbands: approximation($Y^1{}_{i+1}(LL)$), horizontal($Y^1{}_{i+1}(LH)$), vertical($Y^1{}_{i+1}(HL)$), and diagonal($Y^1{}_{i+1}(HH)$) and the dimension of each subband remains identical to that of the input image. The equations (4, 5, 6, and 7) illustrate the j^{th} level decomposition using SWT, and Fig. 2 depicts the input image's first-level SWT decomposition process and corresponding subbands. The x and y axes represent the spatial domain (Sd) and intensity values (Iv), respectively, with low-pass (LPF) and high-pass (HPF) filter banks applied during the decomposition process.

$$LL_{j+1}(U,V) = \sum_{x=-\infty}^{+\infty}\sum_{y=-\infty}^{+\infty} l_x^j l_y^j LL_j(U+x, V+y) \quad (4)$$

$$LH_{j+1}(U,V) = \sum_{x=-\infty}^{+\infty}\sum_{y=-\infty}^{+\infty} l_x^j h_y^j LL_j(U+x, V+y) \quad (5)$$

$$HL_{j+1}(U,V) = \sum_{x=-\infty}^{+\infty}\sum_{y=-\infty}^{+\infty} h_x^j l_y^j LL_j(U+x, V+y) \quad (6)$$

$$HH_{j+1}(U,V) = \sum_{x=-\infty}^{+\infty}\sum_{y=-\infty}^{+\infty} h_x^j h_y^j LL_j(U+x, V+y) \quad (7)$$

Where $LL_{j+1}, LH_{j+1}, HL_{j+1}, and\ HH_{j+1}$ represents four frequency subbands of stationary wavelet transform decomposition.

B. Block Tiling

Since the $Y^1{}_{i+1}(LL)$ subband contains a smoother version of the image and provides a global description along with directional features, the $Y^1{}_{i+1}(LL)$ subband is segregated into several 8×8 overlapping blocks by keeping step size=1 and repeating the block tiling process. Thus, the total blocks (TOB) that are overlapping are defined in equation (8) as follows,

$$TOB = (W - B_w + 1) * (H - B_h + 1) \quad (8)$$

Where the terms W and H define the number of rows and columns of the $Y^1{}_{i+1}(LL)$ subband, B_w and B_h define an

overlapping block's row and column size. The block B_i represents an overlapping block during the process, where i = 1, 2,. . . , TOB.

C. Feature Extraction and Formation of Composite Feature Matrix

In this step, the composite features are formed by extracting DCT coefficients and GLCM features from each overlapping block. It is known that the DCT features prove to be highly efficient. Therefore, initially, we apply DCT on each overlapping block B_i to extract DCT features and the corresponding 2-D DCT coefficient matrix is calculated for each overlapping image block B_i of size $B_w \times B_h$ is shown in equation(9) as follows.

$$B(p,q) = \frac{1}{4}C(p)C(q)\sum_{w=0}^{7}\sum_{h=0}^{7}B_i(w,h)\times \cos\left(\pi\frac{(2w+1)*p}{16}\right)\times\cos\left(\pi\frac{(2h+1)*q}{16}\right), \quad 0 \le p \le 7 \; and \; 0 \le q \le 7 \tag{9}$$

$$where\ C(p) = \begin{cases} \frac{1}{\sqrt{2}} & \text{p=0} \\ \frac{2}{\sqrt{8}} & 0 \le p \le 7 \end{cases} and$$

$$C(q) = \begin{cases} \frac{1}{\sqrt{2}} & \text{q=0} \\ \frac{2}{\sqrt{8}} & 0 \le q \le 7 \end{cases}$$

Through the 2D-DCT transformation, these overlapping blocks are reshaped so that most of their energy becomes concentrated in the initial low-frequency DCT coefficients. Consequently, high-frequency components can only be eliminated by surrendering valuable image information. After applying DCT, the zigzag scanning from the starting top left corner, i.e., DC coefficient to other AC coefficients, is performed to select the most informative frequency coefficients (first 06 coefficients out of 64 features) to construct a part of the feature vector instead of considering every coefficient of an overlapping image block[15]. Hence, the DCT features of each overlapping block are defined as $f_iDCT = [f_{i^1}, f_{i^2}, f_{i^3}, f_{i^4}, f_{i^5}, f_{i^6}]$.

In copy-move forgery, the region copied and pasted at a different location on the same image often has repeated patterns. Therefore, we capture the spatial correlations between pixel values in an image using a texture analysis method known as the GLCM. The GLCM considers the associations between pixel values separated by a specific distance and orientation, i.e., if two pixels with similar values commonly occur close to one another, the GLCM will record data on how pixels are arranged and organized inside an image. Hence, the suggested method employs GLCM to extract the texture features of each overlapping block to capture spatial relationships between pixel values and achieve improved results. The GLCM begins by computing the co-occurrence matrix and subsequently derives multiple features, including contrast, entropy, energy, homogeneity, variance, and correlation. Among all these features, we have considered only the principal statistical metrics that describe the spatial relationships between pixels in an image, such as contrast, correlation, and energy and computation of these properties are defined in equations (10, 11 and 12) as follows,

$$f_{cont} = \sum_{r,c=0}^{N-1}(r-c)^2(Pi)_{rc} \tag{10}$$

$$f_{corr} = \sum_{r,c=0}^{N-1}\frac{(r-\mu_r)(c-\mu_c)Pi(r,c)}{\sigma^2} \tag{11}$$

$$f_{ene} = \sum_{r,c=0}^{N-1}(Pi_{rc})^2 \tag{12}$$

The Pi signifies the GLCM matrix with elements r and c. The N indicates the number of gray levels in the overlapping blocks, and μ_r and μ_c specify the mean of elements of r^{th} row and c^{th} column of GLCM matrix. Finally, σ^2 indicate variance respectively.

Hence, the final GLCM features are defined as $f_iGLCM = [f_{i^{cont}}, f_{i^{corr}}, f_{i^{ene}}]$. Finally, both DCT and GLCM extracted features are combined to form a composite feature vector i.e. $CFV_i = [f_{i^1}, f_{i^2}, f_{i^3}, f_{i^4}, f_{i^5}, f_{i^6}, f_{i^{cont}}, f_{i^{corr}}, f_{i^{ene}}]$ and placed row wise pattern to form a composite feature matrix(CFM). Each row (f_i) of CFM pertains to the composite feature vector of each overlapping block B_i where, $f_{i^1} - f_{i^6}$ corresponds to DCT feature and $f_{i^{cont}}, f_{i^{corr}}$, and $f_{i^{ene}}$ signifies GLCM features corresponds to contrast, correlation and energy respectively. Fig. 3 illustrates a pictorial representation of feature extraction and formation of a composite feature matrix. Later, the CFM is sorted lexicographically to position the feature vectors corresponding to similar blocks adjacent to each other to avoid overlapping blocks that are considered similar and reduce computational time during the feature-matching process. We also record each overlapping block $B_{i^{th}}$ row and column number with composite feature vector for further processing the sorted composite feature matrix.

D. Block Matching and Outlier Detection

In the case of copy-move type of image forgery, it is often believed that the copied region is not placed at the exact location, i.e., the blocks tampered with are non-intersecting with the copied region. So, to efficiently classify and mark the copy-pasted region, we have imposed Block Distance (Bd_{th}) and Block Similarity threshold (Bs_{th}) constraints to group similar overlapping blocks and filter out non-similar blocks present in an image.

1) Block Distance Threshold: This criterion is applied to filter out the neighboring blocks and to match the identical block pairs located at different parts of the feature matrix likely to have been tampered with in the image. The minimum Block Distance(BD) is calculated between two blocks using equation (13).

$$BD(B_i, B_j) = \sqrt{(p-l)^2 + (q-m)^2} \tag{13}$$

Where (p,q) and (l,m) represent coordinates of the top left corner of image blocks B_i and B_j, $BD(B_i, B_j)$ signifies the computed block distance used to compare with block distance threshold(Bd_{th}) value for its consideration, i.e., if $BD(B_i, B_j) > Bd_{th}$ is satisfied. We assume that the block B_i and B_j are likely to have been tampered with and are considered candidates forgery detection.

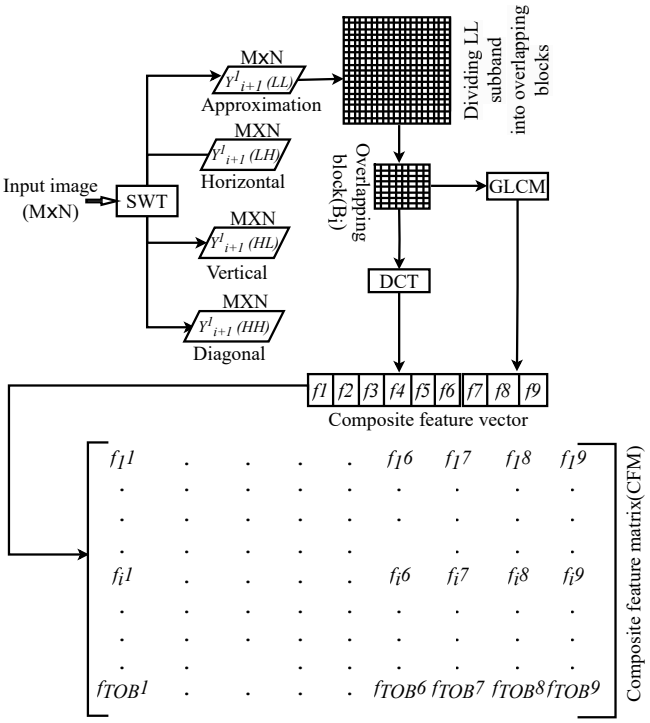


Fig. 3: Composite feature extraction and formation of feature matrix

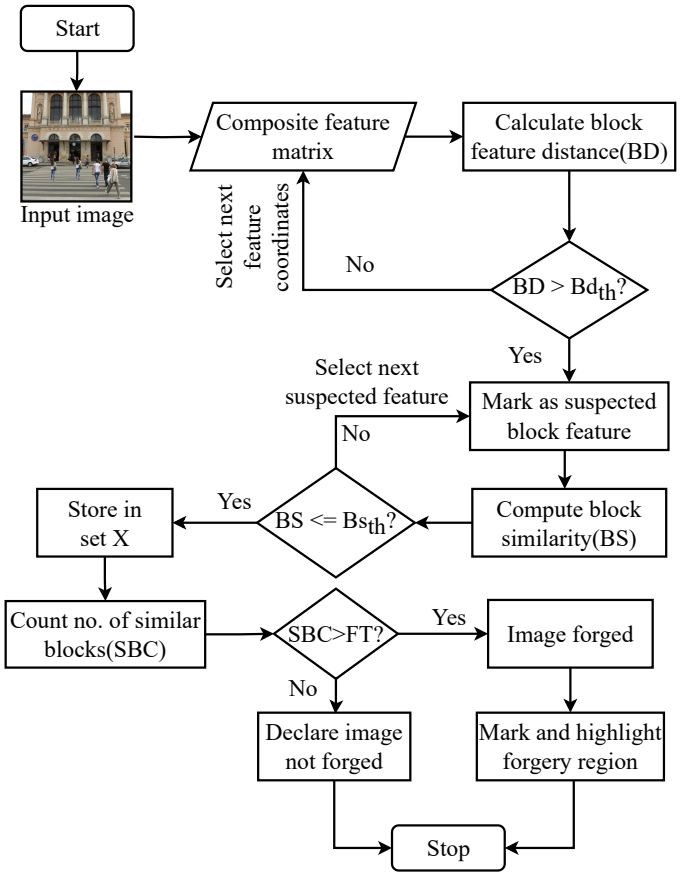


Fig. 4: Flow diagram of feature matching and outlier detection process

2) Block Similarity Threshold: This criteria is adopted to correctly check whether the feature vectors corresponding to the block represent an image's forgery region or not, and this is achieved by comparing the block similarity threshold value with the block similarity(BS) distance computed between two composite feature vectors, i.e. if $BS(B_{\mathrm{i}}, B_{\mathrm{j}}) < Bs_{\mathrm{th}}$ then the $B_{i^{th}}$ and the $B_{j^{th}}$ block features are successfully matched. Respective block indices are copied into set X to indicate the selected pair of blocks is likely to have been forged. Otherwise, it is removed from the composite feature matrix (CFM), indicating no match has been found. The proposed block-matching technique has adopted the Euclidean distance algorithm to estimate the similarity between two feature vectors corresponding to image blocks.Let $\mathrm{f_i}$ and $\mathrm{f_j}$ denote the $\mathrm{i^{th}}$ and $\mathrm{j^{th}}$ feature vectors corresponding to block $\mathrm{B_i}$ and $\mathrm{B_j}$. The Euclidean distance between these feature vectors is performed using equation $BS(B_{\mathrm{i}}, B_{\mathrm{j}}) = \sqrt{\sum_{k=1}^{L} [f_{i(k)} - f_{j(k)}]^2}$, where L is the length of the feature vector. The block-matching process begins from the first row and continues till the end of all feature vectors of a feature matrix. The threshold values used during the block matching are chosen empirically through experiments. Fig. 4 depicts the process of block matching and outlier detection.

E. Forgery Detection

Despite the rigorous block-matching process, due to various post-processing operations, the set X may sometimes contain falsely matched feature vectors that often affect the detection result. Therefore, to remove falsely matched block features and improve the detection result, we employed the Frequency Threshold (FT) constraint to separate most likely forged block pairs from outliers. The FT defines the number of matching blocks having the same mutual positions. Thus, the FT is used to identify the image level evaluation by verifying against several similar block counts (SBC) and copy-pasted regions when a specified number of identical block pairs with the same mutual block positions are detected. Finally, to visualize the resultant image with the detected forgery region, we have employed a binary mapping technique where all matching pairs of blocks are assigned as ones in the neighborhood of 8×8 pixels to display the same image blocks as white. Further morphological operations, such as opening and closing, are applied to eliminate the isolated blocks. Table III tabulates the visual detection results for plain and copy-move forgery with post-processing operations.

IV. EXPERIMENTAL SETUP

The proposed methodology utilizes Intel(R) Core(TM) i5-10500H $CPU@2.50GH_z$ speed, RAM of size 8.00 GB system configuration, and the MATLAB version 2015a image processing toolbox to implement the concept and simulate the experiments.

A. Dataset Used

To evaluate the proposed system, we considered a copy-move forgery image sample from the CoMoFoD dataset[32]. The CoMoFoD dataset primarily contains 200 images of size 512×512. Based on different geometrical transformations and post-processing operations, the CoMoFoD dataset images are divided into five classes such as translation, rotation, scaling, distortion, and combination, and each image group consists

TABLE I: DESCRIPTION OF EVALUATION DATASET

Attacks	Parameters	Levels	Values	Forgery image count
Plain Forgery(PF)	–	–	–	200
Image Blurring(IB)	Value of sigma	l1	0.009	600
		l2	0.005	
		l3	0.0005	
Brightness Change(BC)	Brightness levels (Lower bound, Upper bound)	l1	[0.01, 0.95]	600
		l2	[0.01, 0.9]	
		l3	[0.01, 0.8]	
Color Reduction (CR)	Intensity levels per each color channel	l1	32	600
		l2	64	
		l3	128	
Contrast Adjustment (CA)	Contrast range (Lower bound, Upper bound)	l1	[0.01, 0.95]	600
		l2	[0.01, 0.9]	
		l3	[0.01, 0.8]	

TABLE II: STATISTICAL RESULTS OF PROPOSED FORGERY DETECTION APPROACH

Attacks	Levels	Evaluation metrics		
		P(%)	R(%)	F1(%)
PF	-	94.10	93.82	93.96
IB	l1	85.21	89.10	89.87
	l2	89.77	87.44	88.59
	l3	86.66	83.44	89.67
BC	l1	93.23	95.06	94.14
	l2	96.54	94.88	96.54
	l3	95.88	96.39	95.13
CR	l1	94.77	94.34	97.81
	l2	90.47	89.77	93.68
	l3	93.58	92.33	92.95
CA	l1	94.65	93.80	94.22
	l2	93.86	92.81	93.33
	l3	90.66	89.77	92.10

of six post-processed images, such as JPEG compression, image blurring, noise addition, brightness change, color reduction, and contrast adjustments. As a result, there are 10,400 image samples available, including ground truth and original images, in the CoMoFoD dataset. Since this dataset contains more forged images produced by post-processing operations, it is more helpful in evaluating the robustness of the algorithms against post-processing operations. Table I summarizes the dataset images used to test the proposed methodology.

B. Parameter Setup

To experiment with the proposed system, we have assumed the following values for the parameter: Overlapping block size (B_wxB_h) = 8×8, No. of neighboring features vectors to be compared = 5, Block distance threshold (Bd_{th})= 40,Block similarity threshold(Bs_{th})=0.00001, and Frequency threshold(FT)=12.

C. Performance Metric

The proposed system is gauged at two levels. Firstly, at the image level, we inspect whether the suspected input image has tampered with copy-move forgery, and it does not require any ground truth image during evaluation. Secondly, we estimate how accurately we can identify the tampered pixels of the copy-move forgery region. Thus, evaluation requires both tampered and ground truth images. The following are the three major benchmark parameters, namely Precision(P), Recall(R), and F1-score(F1) are used to evaluate the forgery detection techniques[33], [34] and are expressed in equations (14), (15), and (16) as follows,

$$P = \frac{True\ positives(t_p)}{True\ positives(t_p) + False\ positives(f_p)} \quad (14)$$

$$R = \frac{True\ positives(t_p)}{True\ positives(t_p) + False\ negatives(f_n)} \quad (15)$$

$$F1 - Score(F1) = \frac{2 * (PR)}{(P + R)} \quad (16)$$

Where t_p, f_p and f_n are clearly described as follows,

1) **t_p (True positives)**: It quantifies the correct identification of forged images forged for image-level evaluation and forged pixels forged for pixel-level evaluation.
2) **f_p(False positives)**: It defines the misidentification of the forged image as un-forged for image-level evaluation and forged pixels as un-forged for pixel-level evaluation.
3) **f_n(False negatives)**: It indicates unidentified un-forged images for image-level evaluation and the number of unidentified un-forged pixels for pixel-level evaluation.

Hence, from the above definition, we can conclude that the evaluation parameters are directly proportional to the performance of any detection technique.

D. Evaluation of Results and Discussion

This section discusses the experimental approach and records the results obtained for basic copy-move forgery and forgery with different post-processing operations. In the former, some parts of the image region are copied and placed at a different location without applying post-processing operations. Here, we have considered 80 images, including 40 original and 40 forged (001_F to 040_F), to evaluate the performance at the image level. Additionally, we have taken the corresponding ground truth images for assessing at the pixel level. As per the evaluation results, the proposed technique can correctly detect 39 forgery images forged out of 40 forgery images and 37 original images out of 40 authentic images. However, the proposed approach erroneously identified three original images as forged and one as original, which was forged. Hence, the method produces results with a precision of 97.50%, recall of 92.26%, an F1-score of 95.12% and a precision of 93.40%, recall of 90.18% and an F1-score of 92.86% for image level and the pixel level evaluation respectively.

E. Performance Evaluation for Post-Processed Images

The assessment of copy-move forgery image performance across various post-processing scenarios involves the analysis of 40 forgery images that have undergone post-processing, including image blurring, brightness change, color reduction,

and contrast adjustment operations. Specifically, each of these 40 images is subjected to three post-processing levels, resulting in 12 post-processed forgery variations for each image. Consequently, the evaluation encompasses a comprehensive set of 480 forgery images that have undergone diverse post-processing operations and their corresponding ground truth images sourced from the CoMoFoD dataset.

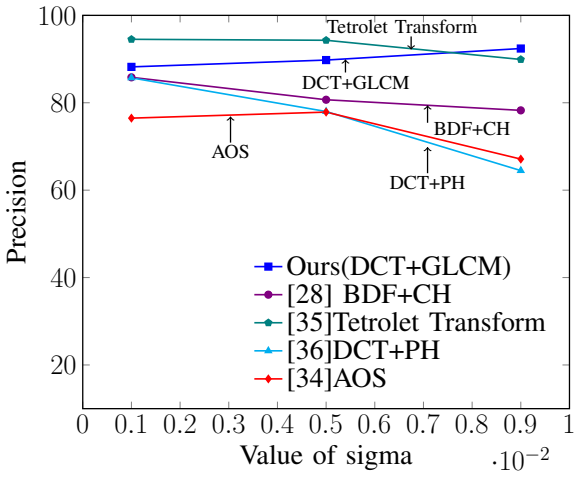
1) Blurring Operation: To evaluate the robustness of the proposed image-blurring post-processing operation, we selected 40 images from the CoMoFoD dataset that underwent post-processing with three distinct levels [0.009, 0.005, 0.0005], generating 120 altered images. Subsequently, we utilized these images to validate the method's performance against image-blurring attacks. The outcomes of this assessment, encompassing precision, recall, and F1-score metrics, can be found in Table II. Furthermore, visual representations of the results are provided in the third row of Table III.

2) Brightness Change Operation: To assess the robustness against the brightness change post-processing operations for forgery images, we chose 40 images from the CoMoFoD dataset. These chosen images were post-processed using three separate brightness levels such as [(0.01, 0.95), (0.01, 0.9), (0.01, 0.8)]. The range [0.01, 0.95] of brightness change does not significantly affect the image's visual effect, whereas the brightness level [0.01, 0.8] yields a much brighter image, resulting in 120 modified images. The evaluation results are presented in Table II, and the detection results are in the fourth row of Table III.

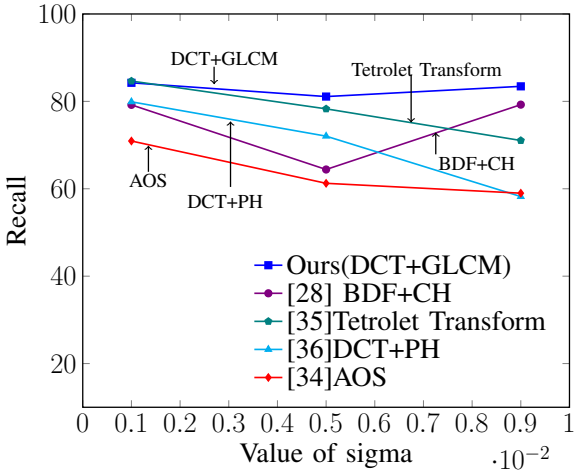
3) Color Reduction and Contrast Adjustment: To evaluate the robustness of the proposed technique against color reduction, we analyzed 120 images from the CoMoFoD dataset. These images underwent post-processing with various intensity levels per color channel, including 32, 64, and 128. Additionally, we examined image samples that underwent contrast adjustment post-processing with three distinct contrast levels: [(0.01, 0.95), (0.01, 0.9), (0.01, 0.8)]. Among these, the contrast change range of [0.01, 0.95] had minimal impact on the image's visual appearance, while the contrast level [0.01, 0.8] resulted in a significantly brighter image. The detection results are presented in Table II, and a visual representation of the outcomes can be found in Table III, fifth and sixth rows for color reduction and contrast adjustment post-processing operations, respectively.

F. Overall Performance Analysis

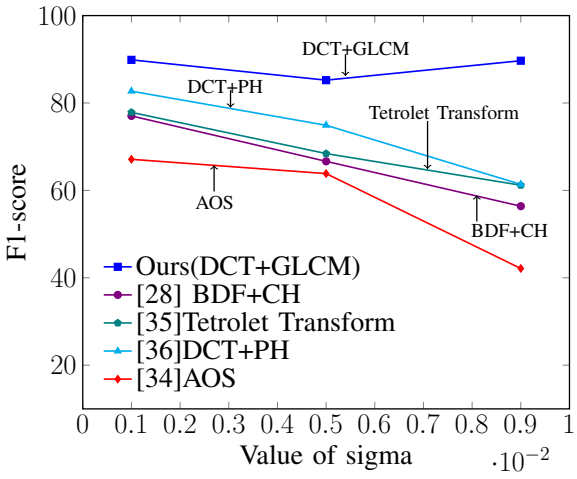
The primary aim of any copy-move forgery detection technique is to determine whether the given input image is forged and precisely mark the forgery regions. Therefore, the obtained results are compared with existing approaches to understand the capability and correctness of our proposed method. Table VII contains a comparative study of detection results with an existing system regarding precision, recall and F1-score for plain copy move forgery and forgery with different post-processing operations. Fig. 5 to Fig. 8 shows corresponding graphs. Based on the comparative study tabulated in Table VII, the proposed method shows the results are progressive compared to existing techniques. The results of the proposed approach are also compared with well-known current methods to validate the overall performance. As per the obtained results tabulated in Table IV and the



(a)



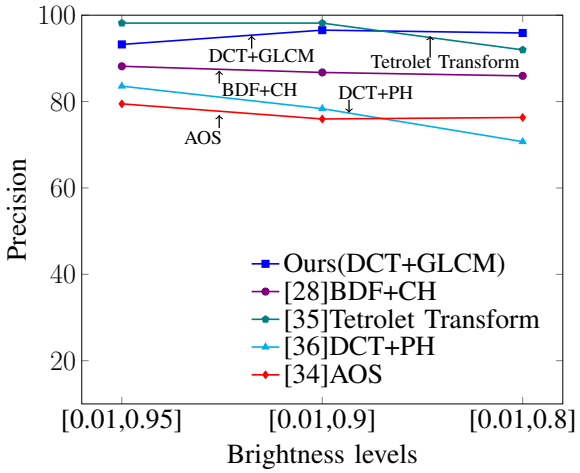
(b)



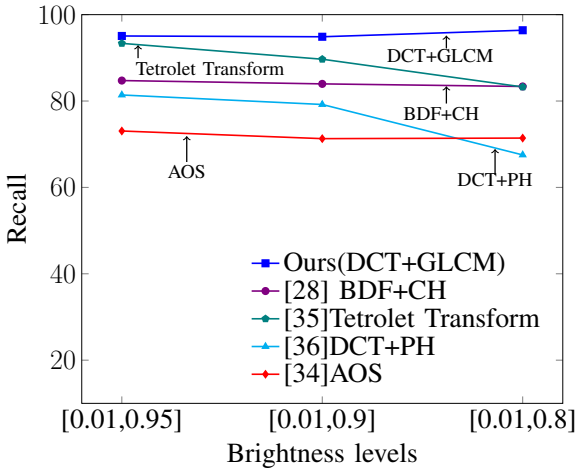
(c)

Fig. 5: Comparison of forgery detection results in terms of (a) Precision, (b) Recall, and (c) F1-score with an existing system for image blurring operation

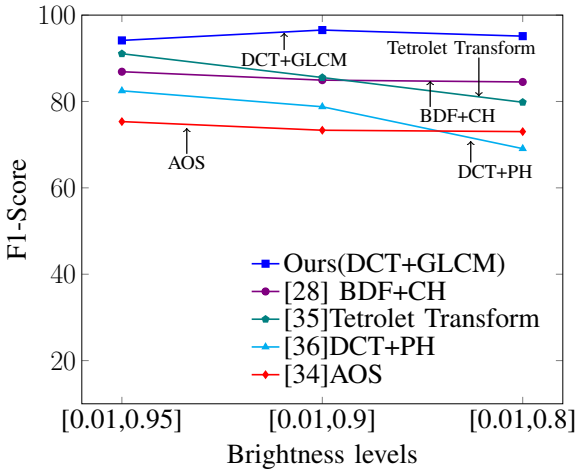
corresponding graph shown in Fig. 11, the precision and recall are marginally less compared with the techniques[37], [19]. Still, the recall compared to Lin et al. 2019[38] method differs more by about 9.82%. However, the F1-score is much more significant compared to other methods. Hence, the overall results demonstrate the state-of-the-art in detecting



(a)

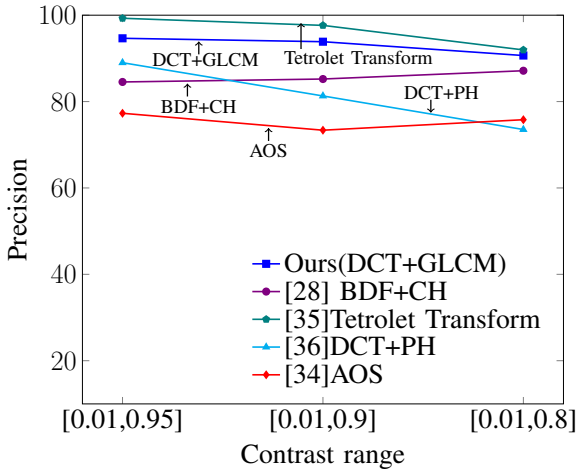


(b)

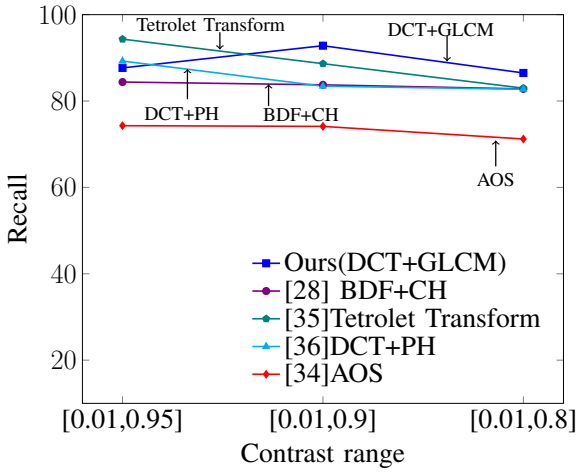


(c)

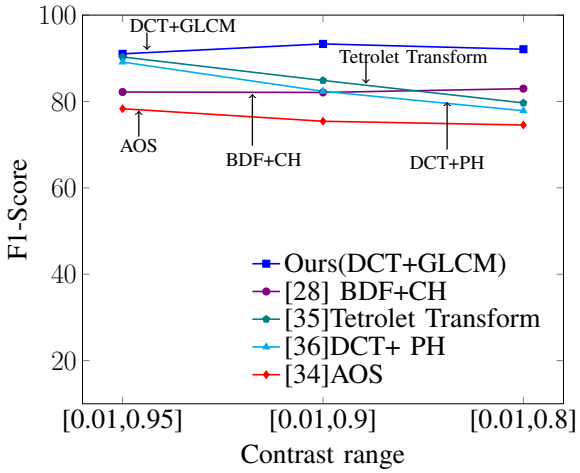
Fig. 6: Comparison of forgery detection results in terms of (a) Precision, (b) Recall, and (c) F1-score with an existing system for brightness change operation



(a)



(b)



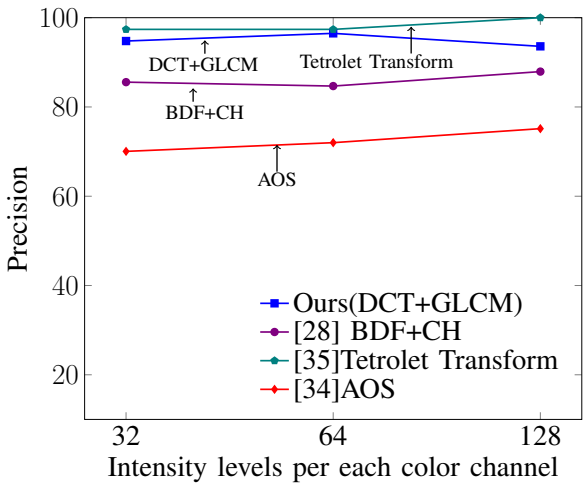
(c)

Fig. 7: Comparison of forgery detection results in terms of (a) Precision, (b) Recall, and (c) F1-score with an existing system for contrast adjustment operation

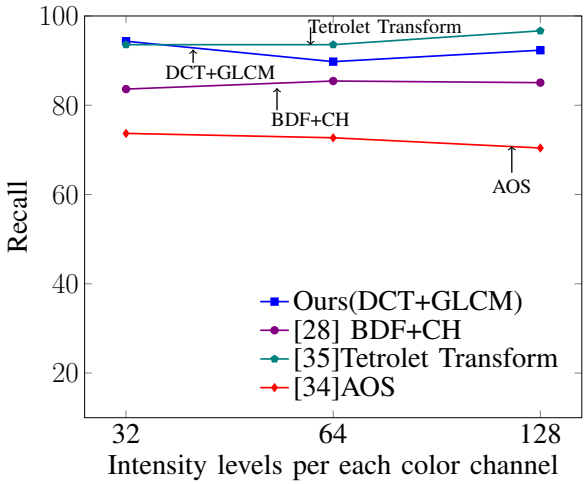
forgery and forgery regions in digital images.

G. Analysis of the Impact of Block Size on Performance

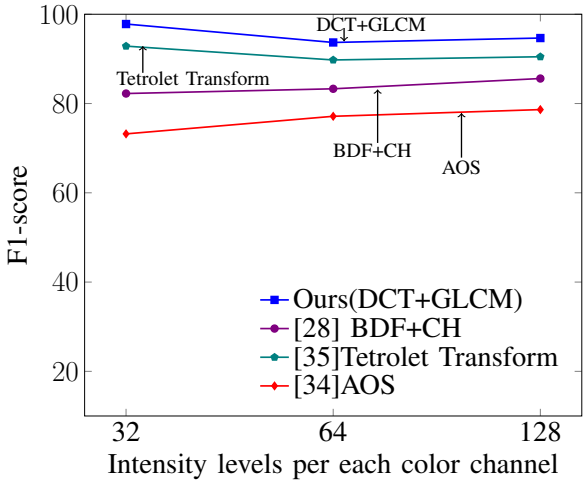
To understand the most suitable overlapping block size to get better detection performance. We have analyzed the proposed methodology by changing the overlapping block size from 4×4 to 16×16, gradually increasing the block size by 2×2, and recording the effect of the increase in block size on the performance metrics. As per the graph in Fig. 9, the Precision generally improves as the block size increases, suggesting that larger block sizes reduce the number of false positives. Consequently, it increases the



(a)



(b)



(c)

Fig. 8: Comparison of forgery detection results in terms of (a) Precision, (b) Recall, and (c) F1-score with an existing system for color reduction operation

TABLE III: VISUALIZATION OF DETECTION RESULTS FOR PLAIN AND POST-PROCESSED FORGERY IMAGES

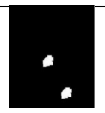
Forgery attacks	Original image	Forged image	Ground truth	Detected result
Plain forgery				
Image blurring				
				
Brightness change				
				
Color reduction				
				
Contrast adjustment				
				

TABLE IV: COMPARISON OF FORGERY DETECTION TECHNIQUES WITH EXISTING APPROACHES

Existing techniques	Feature name	P(%)	R(%)	F1(%)
Pun et al. 2015[34]	SIFT	81.26	84.01	82.34
Emam et al. 2016[37]	PCET	93.50	81.00	88.10
Lin et al. 2019[38]	LIOP	67.60	100.00	80.70
Gani and Qadir 2020[19]	DCT-Cellular Automata (CA)	83.00	90.30	86.50
Gani and Qadir 2021[39]	Thresholding CA	90.00	87.00	87.80
Yang et al. 2021[40]	SIFT and Two-Stage Filtering (TSF)	55.30	77.30	63.80
Raju and Nair 2022[28]	BDF + CH	90.60	86.70	88.40
Ours	DCT + GLCM	93.40	90.18	92.86

accuracy performance of Precision for forgery detection. Conversely, the smaller block size tends to have higher Recall, indicating fewer false negatives due to increased block similarities. However, as the block size increases, Recall decreases significantly, and the F1-score is higher at block sizes 6×6 to 8×8, where both Precision and Recall are relatively high. A good balance exists between false positives and false negatives. As the block size increases beyond this range, the F1-score and Recall drop gradually,

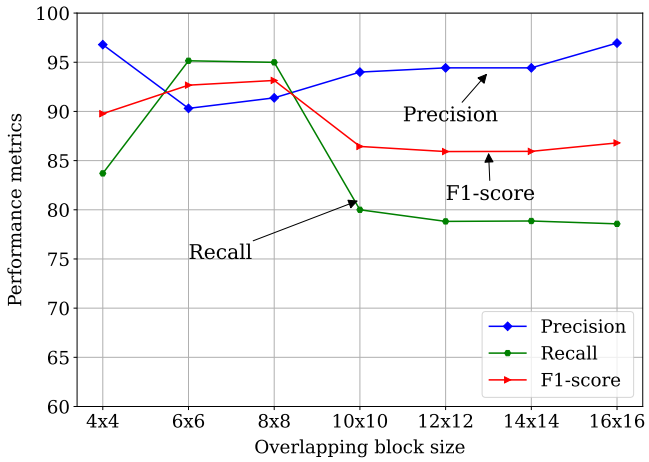


Fig. 9: Effect of overlapping block size on performance metrics

reflecting the increased false negatives. Therefore, there is always a tradeoff between block size and performance.

Hence, selecting an optimal block size is essential to achieve better detection results and attain a balanced performance. Therefore, our analysis has successfully navigated this tradeoff to identify the optimal block size. As per the experimental results illustrated in Fig. 9, the block sizes between 6×6 and 8×8 seem ideal, and they balance minimizing false positives and negatives and improved F1 score, especially the 8×8 block size. Therefore, our proposed approach sets an overlapping block size 8×8.

H. Computational Complexity

Usually, in any copy-move forgery detection technique, identifying correlated pixels is a challenging task and often needs an extensive feature-matching process. Our proposed method uses stationary wavelet transformation as a part of preprocessing and DCT followed by GLCM to extract image features. The size of the feature dimension is reduced by considering only the first six highly potential DCT features and combined with texture elements to represent feature vectors to facilitate fast feature matching and reduce computational overhead. We have noted the time the proposed system took at two stages and compared it with other techniques to analyze the computational burden. Table V tabulates the time (in seconds) taken to process a single image of size 512×512 for the feature extraction and matching process. The time recorded in Table V and the graph depicted in Fig. 10 reveal variations in time consumption between different feature extraction and matching techniques utilized in various existing techniques. Some of these techniques exhibit shorter processing times for feature extraction, primarily due to the dimensionality of the features. Conversely, they require more execution time for feature matching. Notable examples include Tralic[41], Mahmood[42], Wang[36], and Wang[43]. This extended processing time can be attributed to the rigorous computation of similarity, the calculation of shift vectors between a block and its nearest neighbors, and the segregation of suspected block features. Usually, the number of nearest neighbors to be checked increases the

TABLE V: COMPARISON OF AVERAGE RUNNING TIME (IN SECONDS) OBSERVED AT TWO STAGES WITH EXISTING TECHNIQUES

Existing techniques	Feature extraction	Feature matching	Total
D. Tralic et al. [41]	37	225	262
T. Mahmood et al.[42]	2.8	175	177.8
H. Wang and H. Wang [36]	5.6	150	155.6
Y. Wang et al. [43]	4.1	165	169.1
G. Gani and F. Qadir [39]	305	16	321
G. Gani and F. Qadir[25]	31.5	0.8	32.3
Ours	21.5	12.09	33.59

time required for feature matching. In contrast, Gani [39], and Qudir[25] employ a more intensive feature extraction process, resulting in longer extraction times but shorter matching times. However, the proposed technique strikes a balance by offering relatively less feature extraction and matching time. However, the feature-matching process in the proposed method takes slightly longer than in Qudir[25] due to different threshold parameters. It's important to highlight a trade-off between the running time and the similarity threshold value. Increasing the similarity threshold can enhance the method's robustness against post-processing manipulations. However, this improvement comes at the expense of improved matching and post-processing time. Because many feature vectors meet the matching and filtering criteria, even when they are located outside the copy-pasted regions. Therefore, selecting an appropriate similarity threshold value is crucial to minimize the resulting matched vectors.

V. CONCLUSION

The image copy-move forgery is a widely used tampering method. Our proposed technique uses composite features formed using DCT and GLCM image features. To reduce the size of the feature dimension, we have considered only the first six potential DCT coefficients and combined them with GLCM features to facilitate fast feature matching and reduce computational overhead. The experimental results obtained in Section IV reveal that the proposed technique yields better precision, recall, and F1-score for both basic copy move forgery and a forgery with various post-processing operations and confirms significant improvement over the different existing techniques. Nevertheless, image forgery can be concealed by applying other post-processing operations such as increasing or decreasing the size of the forgery region, rotation, adding white Gaussian noise, and JPEG compression and a blend thereof. In this case, detecting forgery is much more complex and challenging. Along with this, selecting the significant value of the threshold is crucial. Therefore, there is a demand for more sophisticated algorithms to address various issues related to the blend of post-processing operations and a new approach to select dynamic threshold values that are essential for accurately classifying and detecting copy-move forgery images and regions.

TABLE VI: SUMMARIZATION OF EXISTING TECHNIQUES

Ref#	Feature extraction	Remarks
S. M. Fadl and N. A. Semary [13]	Fast Fourier Transform (FFT)	The exhaustive search and GPU parallel computing increase the execution time and detection. The precision needs to be improved for geometric and intensity modifications.
K. Hayat and T. Qazi[14]	DWT and DCT	Underperform in the presence of occlusion and images with recurring patterns.
T. Mahmood et al.[15]	SWT and DCT	Significant scaling, rotation, inpainting, additive noise, contrast adjustments, or combinations need to be addressed efficiently.
V. Thirunavukkarasu et al.[16]	DSWT	It does not detect the tampered region with some geometric transformations.
Emre Gurbuz[17]	ICV	Unable to ascertain which of the matched copy-move region pairs is the original and which is the duplicate. The parameter values are selected based on an empirical study. Need to develop dynamic parameter selection technique.
Beijing Chen et al.[18]	FrQCT	Not effectively addressing the forgery images with large scaling and rotation.
G. Gani and F. Qadir [19]	Sign of DCT coefficients	Improved methods need to be derived to address rotation and scaling post-processing attacks.
G. S. Priyanka and K. Singh[20]	DCT and SVD	Difficult in the detection of small-sized forged regions. Appropriate clustering by the K-means and selecting optimal threshold values pose significant challenges.
Dua, J. Singh, and H. Parthasarathy[21]	DCT	Experiences slightly higher rates of false detections, mainly when there is a change in illumination and blurring. Extracting phase congruency features from multiple orientations in the covariance matrix poses a challenge.
M. Bilal et al. [22]	DWT, SURF and BRISK	Complex post-processing attacks include large scaling, smoothening, and brightness change need to be addressed.
S. P. Jaiprakash et al.[23]	DCT and DWT	Hard in detecting images with varying resolutions, blurring attacks, and multiple forgery regions.
Gul Tahaoglu[24]	DCT	The method must address other post-processing attacks, such as brightness change, color reduction, and contrast adjustment, applied to the entire forgery image.
Gulnawaz Gani and Fasel Qadir[25]	DCT and CA	Geometric transformations and other arbitrary manipulation types need to be addressed, and feature extraction time needs to be reduced.
Q. Lyu et al.[26]	LIOP keypoints	Precision decreases due to the extended triangles and increased number of keypoint pairs resulting from the double-matching process.
T. Qazi, et al.[27]	DWT and Inverse DWT	The method fails to detect when the patch is not taken from the host image and has high JPEG compression.
P. M. Raju and M. S. Nair[28]	BDF and CH	This method can detect copy-move forgery regions, but still, an improvement is needed for the accuracy and detection of other types of forgery images.
S. B. G. T. Babu and C. S. Rao [29]	SPT and GLCM	This approach cannot discriminate between authentic and computer-generated images.
S. B. G. T. Babu and C. S. Rao [30]	PCET and Gradient Direction Pattern (GDP)	This approach does not address the image blurring, color reduction, and brightness change post-processing attacks. Also, the method for threshold value calculation needs to be improved.

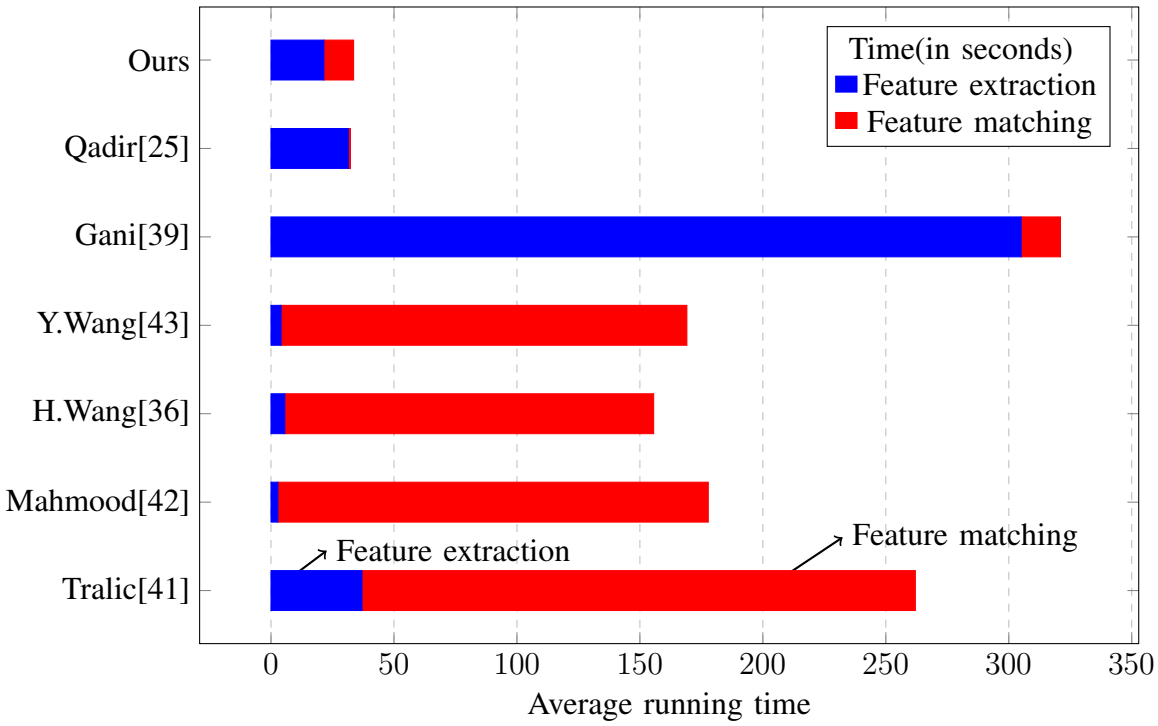


Fig. 10: Comparison of average running time with existing techniques

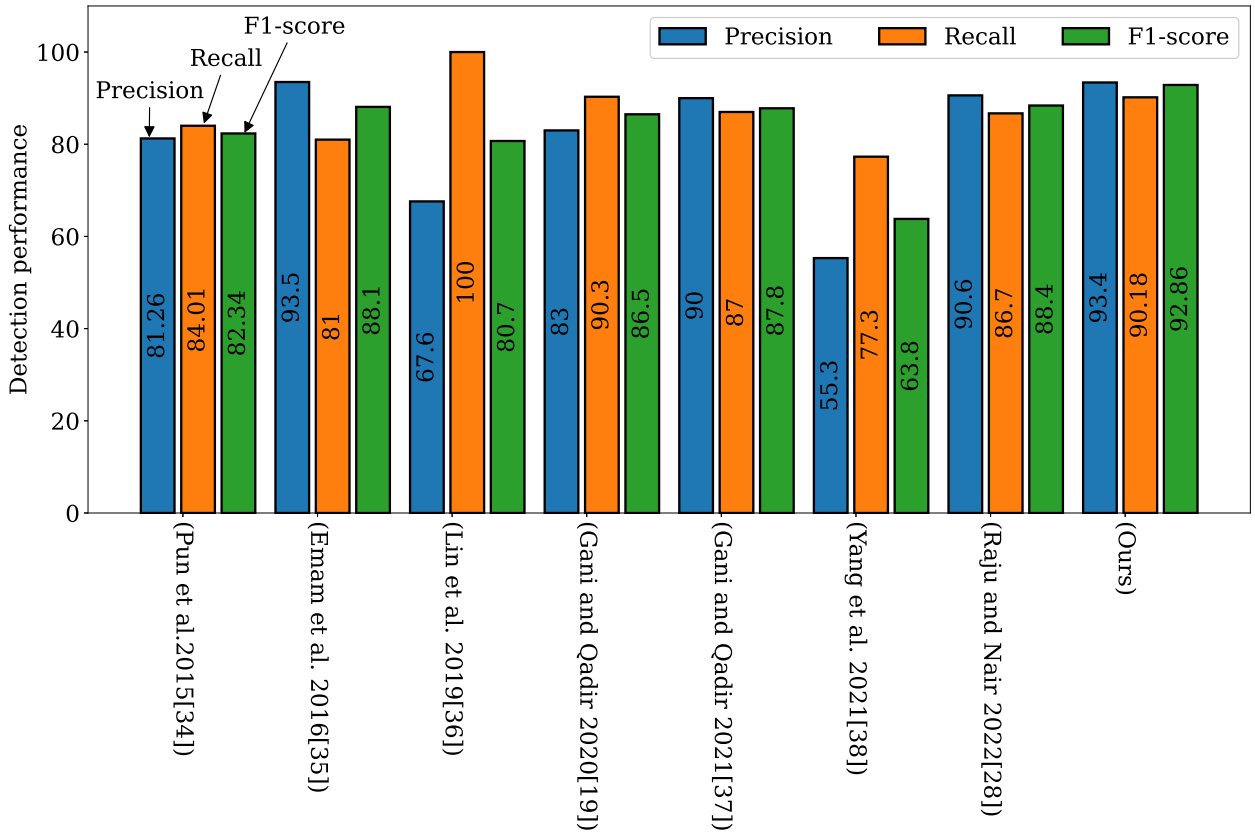


Fig. 11: Comparison of detection results with existing techniques

TABLE VII: COMPARISON OF STATISTICAL RESULTS WITH EXISTING FORGERY DETECTION TECHNIQUES

Attacks	Levels	Ours			BDF+Color Histogram(CH)[28]			Tetrolet Transform[35]			DCT+Perceptual Hashing(PH)[36]			Adaptive Oversegmentation(AOS)[34]		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
PF	-	94.10	93.82	93.96	90.55	86.74	88.35	99.20	92.16	95.64	90.02	90.10	90.06	81.26	84.01	82.34
IB	l1	88.21	84.25	89.87	85.82	79.20	77.04	94.52	84.65	77.85	85.70	79.90	82.70	76.48	70.93	67.11
	l2	89.77	81.10	85.22	80.69	64.41	66.65	94.30	78.29	68.42	78.00	72.04	74.90	77.86	61.25	63.84
	l3	92.41	83.44	89.67	78.26	79.25	56.38	89.91	71.05	61.18	64.90	58.22	61.38	67.11	58.99	42.13
BC	l1	93.23	95.06	94.14	88.17	84.74	86.89	98.17	93.35	91.06	83.57	81.40	82.47	79.47	73.06	75.34
	l2	96.54	94.88	96.54	86.74	83.97	84.94	98.17	89.68	85.55	78.36	79.20	78.78	75.97	71.29	73.34
	l3	95.88	96.39	95.13	85.95	83.37	84.52	91.97	83.26	79.82	70.71	67.52	69.08	76.32	71.42	73.02
CR	l1	94.77	94.34	97.81	85.56	83.62	82.24	97.38	93.57	92.86	-	-	-	70.05	73.69	73.20
	l2	96.49	89.77	93.68	84.68	85.43	83.30	97.38	93.57	89.76	-	-	-	72.00	72.71	77.14
	l3	93.58	92.33	94.67	87.92	85.06	85.60	100.00	96.67	90.48	-	-	-	75.14	70.42	78.64
CA	l1	94.65	87.67	91.03	84.54	84.40	82.19	99.29	94.33	90.31	89.00	89.26	89.13	77.29	74.29	78.32
	l2	93.86	92.81	93.33	85.22	83.75	82.09	97.64	88.65	84.87	81.30	83.43	82.35	73.37	74.13	75.42
	l3	90.66	86.50	92.10	87.14	82.81	82.97	91.96	82.98	79.67	73.53	82.72	77.85	75.80	71.21	74.55

Algorithm 1: Pseudo code of the proposed forgery detection approach

Data: Image as an input.
Result: i. Identify whether the input image is forged or authentic.
ii. Mark and display the forgery region present in the forgery image.

Read the image Img_i where i=1,2,3,….N.
Pre-process the input image and apply SWT to utilize the approximation $({Y^1}_{i+1}(LL))$ subband.
$Img_g \leftarrow rgb2gray(Img_i)$;
$Img_{LL} = SWT(Img_g)$;
Divide (Img_{LL}) subband into overlapping blocks using block tiling process.
$TOB \leftarrow (W - B_{\mathrm{w}} + 1) * (H - B_{\mathrm{h}} + 1)$;
Extract DCT and GLCM features from each overlapping block B_{i} and construct a composite feature matrix(CFM).
while $i \neq len(TOB)$ **do**
    $f_i^{DCT} \leftarrow [f_{i^1}, f_{i^2}, f_{i^3}, f_{i^4}, f_{i^5}, f_{i^6}]$;
    $f_i^{GLCM} \leftarrow [f_{i^{cont}}, f_{i^{corr}}, f_{i^{ene}}]$;
    $CFV_i \leftarrow f_i^{DCT} + f_i^{GLCM}$;
    $CFM[row_i] = [CFV_i]$;
end
Lexicographically sort the composite feature vectors from CFM.
$CFM_{sorted} \leftarrow lexico\_sort(CFM)$;
Perform block matching to segregate similar feature blocks and drop the outliers.
for $i \leftarrow 1$ *to* $len(CFM_{sorted}) - 10$ **do**
    for $j \leftarrow 1$ *to* 10 **do**
        if $BD(CFM_{sorted}(i, 10:11), CFM_{sorted}(i+j, 10:11)) \geq Bd_{th}$ **then**
            if $SB(CFM_{sorted}(i, 1:9), CFM_{sorted}(i+j, 1:9)) \leq Bs_{th}$ **then**
                $X_{\mathrm{i}} \leftarrow CFV_i, CFV_j, (p,q) and (l,m)$;
            end
        end
    end
end
Detect the forgery image and mark the forgery region using suspected blocks count (SBC)
if $SBC \geq FT$ **then**
    i.Declare the input image is forged;
    ii. Perform binary mapping to mark the forgery region and visualize the result;
else
    Declare the input image is not forged(Authentic);
end

REFERENCES

[1] N. Arava, A. Bhuvaneshwari, and H. Fathima, "Dual information audio watermarking with modified wavelet based lsb technique," *Lecture Notes in Engineering and Computer Science*, vol. 2245, pp. 129–137, 2023.

[2] K. Zhou, C. Zhang, Y. Yu, S. Cong, and X. Yue, "Improving smote technology for credit card fraud detection category imbalance issues," *Engineering Letters*, vol. 31, no. 4, pp. 1780–1785, 2023.

[3] P. R. Bevinamarad and M. S. Shirldonkar, "Audio forgery detection techniques: Present and past review," in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, 2020, pp. 613–618.

[4] P. R. Bevinamarad and M. U. Mulla, "Review of techniques for the detection of passive video forgeries," *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, vol. 2, pp. 199–203, 2017.

[5] S. Singh and R. Kumar, "Image forgery detection: comprehensive review of digital forensics approaches," *Journal of Computational Social Science*, vol. 7, no. 1, pp. 877–915, 2024.

[6] J. Fridrich, D. Soukal, and J. Lukáš, "Detection of copy-move forgery in digital images," in *Proceedings of Digital Forensic Research Workshop*, Cleveland, Ohio, USA, 2003.

[7] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," Department of Computer Science, Dartmouth College, Hanover, New Hampshire, Tech. Rep. TR2004-515, 2004.

[8] B. Mahdian and S. Saic, "Detection of copy-move forgery using a method based on blur moment invariants," *Forensic Science International*, vol. 171, no. 2–3, pp. 180–189, 2007.

[9] L. Guohui, W. Qiong, T. Dan, and S. Shaojie, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on dwt and svd," in *Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, ICME 2007*, 2007, pp. 1750–1753.

[10] Y. Cao, T. Gao, L. Fan, and Q. Yang, "A robust detection algorithm for copy-move forgery in digital images," *Forensic Science International*, vol. 214, no. 1–3, pp. 33–43, 2012.

[11] J. Zhao and J. Guo, "Passive forensics for copy-move image forgery using a method based on dct and svd," *Forensic Science International*, vol. 233, no. 1–3, pp. 158–166, 2013.

[12] R. Dixit, R. Naskar, and S. Mishra, "Blur-invariant copy-move forgery detection technique with improved detection accuracy utilising swt-svd," *IET Image Processing*, vol. 11, no. 5, pp. 301–309, 2017.

[13] S. M. Fadl and N. A. Semary, "Robust copy–move forgery revealing in digital images using polar coordinate system," *Neurocomputing*, vol. 265, pp. 57–65, 2017.

[14] K. Hayat and T. Qazi, "Forgery detection in digital images via discrete wavelet and discrete cosine transforms," *Computers & Electrical Engineering*, vol. 62, pp. 448–458, 2017.

[15] T. Mahmood, Z. Mehmood, M. Shah, and T. Saba, "A robust technique for copy-move forgery detection and localization in digital images via stationary wavelet and discrete cosine transform," *Journal of Visual Communication and Image Representation*, vol. 53, pp. 202–214, 2018.

[16] V. Thirunavukkarasu, J. Satheesh Kumar, G. S. Chae, and J. Kishorkumar, "Non-intrusive forensic detection method using dswt with reduced feature set for copy-move image tampering," *Wireless Personal Communications*, vol. 98, no. 4, pp. 3039–3057, 2018.

[17] E. Gürbüz, G. Ulutas, and M. Ulutas, "Detection of free-form copy-move forgery on digital images," *Security and Communication Networks*, vol. 2019, no. 1, p. 8124521, 2019.

[18] B. Chen, M. Yu, Q. Su, and L. Li, "Fractional quaternion cosine transform and its application in color image copy-move forgery detection," *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8057–8073, 2019.

[19] 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, p. 102510, 2020.

[20] G. S. Priyanka and K. Singh, "An improved block based copy-move forgery detection technique," *Multimedia Tools and Applications*, vol. 79, pp. 13 011–13 035, 2020.

[21] S. Dua, J. Singh, and H. Parthasarathy, "Detection and localization of forgery using statistics of dct and fourier components," *Signal Processing: Image Communication*, vol. 82, p. 115778, 2020.

[22] M. Bilal, H. A. Habib, Z. Mehmood, T. Saba, and M. Rashid, "Single and multiple copy–move forgery detection and localization in digital images based on the sparsely encoded distinctive features and dbscan clustering," *Arabian Journal for Science and Engineering*, vol. 45, no. 4, pp. 2975–2992, 2020.

[23] S. P. Jaiprakash, M. B. Desai, C. S. Prakash, V. H. Mistry, and K. L. Radadiya, "Low dimensional dct and dwt feature based model for detection of image splicing and copy-move forgery," *Multimedia Tools and Applications*, vol. 79, no. 39–40, pp. 29 977–30 005, 2020.

[24] G. Tahaoglu, G. Ulutas, B. Ustubioglu, and V. V. Nabiyev, "Improved copy move forgery detection method via L*a*b* color space and enhanced localization technique," *Multimedia Tools and Applications*, vol. 80, no. 15, pp. 23 419–23 456, 2021.

[25] G. Gani and F. Qadir, "Copy move forgery detection using DCT, PatchMatch and cellular automata," *Multimedia Tools and Applications*, vol. 80, no. 21, pp. 32 219–32 243, 2021.

[26] Q. Lyu, J. Luo, K. Liu, X. Yin, J. Liu, and W. Lu, "Copy move forgery detection based on double matching," *Journal of Visual Communication and Image Representation*, vol. 76, p. 103057, 2021.

[27] T. Qazi, M. Ali, K. Hayat, and B. Magnier, "Seamless copy–move replication in digital images," *Journal of Imaging*, vol. 8, no. 3, pp. 1–15, 2022.

[28] P. M. Raju and M. S. Nair, "Copy-move forgery detection using binary discriminant features," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 2, pp. 165–178, 2022.

[29] S. B. G. T. Babu and C. S. Rao, "An optimized technique for copy–move forgery localization using statistical features," *ICT Express*, vol. 8, no. 2, pp. 244–249, 2022.

[30] S. B. G. T. Babu and C. S. Rao, "Efficient detection of copy-move forgery using polar complex exponential transform and gradient direction pattern," *Multimedia Tools and Applications*, vol. 82, no. 7, pp. 10 061–10 075, 2023.

[31] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation-based image copy-move forgery detection scheme," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 507–518, 2015.

[32] D. Tralic, I. Zupancic, S. Grgic, and M. Grgic, "Comofod - new database for copy-move forgery detection," in *Proceedings of Elmar - International Symposium Electronics in Marine*, 2013, pp. 49–54.

[33] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1841–1854, 2012.

[34] C. M. Pun, X. C. Yuan, and X. L. Bi, "Image forgery detection using adaptive oversegmentation and feature point matching," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1705–1716, 2015.

[35] 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, p. 102481, 2020.

[36] H. Wang and H. Wang, "Perceptual hashing-based image copy-move forgery detection," *Security and Communication Networks*, vol. 2018, p. 6853696, 2018.

[37] M. Emam, Q. Han, and X. Niu, "Pcet based copy-move forgery detection in images under geometric transforms," *Multimedia Tools and Applications*, vol. 75, no. 18, pp. 11 513–11 527, 2016.

[38] C. Lin *et al.*, "Copy-move forgery detection using combined features and transitive matching," *Multimedia Tools and Applications*, vol. 78, no. 21, pp. 30 081–30 096, 2019.

[39] G. Gani and F. Qadir, "A novel method for digital image copy-move forgery detection and localization using evolving cellular automata and local binary patterns," *Evolving Systems*, vol. 12, no. 2, pp. 503–517, 2021.

[40] J. Yang, Z. Liang, Y. Gan, and J. Zhong, "A novel copy-move forgery detection algorithm via two-stage filtering," *Digital Signal Processing*, vol. 113, p. 103032, 2021.

[41] D. Tralic, S. Grgic, X. Sun, and P. L. Rosin, "Combining cellular automata and local binary patterns for copy-move forgery detection," *Multimedia Tools and Applications*, vol. 75, no. 24, pp. 16 881–16 903, 2016.

[42] T. Mahmood, A. Irtaza, Z. Mehmood, and M. Tariq Mahmood, "Copy–move forgery detection through stationary wavelets and local binary pattern variance for forensic analysis in digital images," *Forensic Science International*, vol. 279, pp. 8–21, 2017.

[43] Y. Wang, X. Kang, and Y. Chen, "Robust and accurate detection of image copy-move forgery using pcet-svd and histogram of block similarity measures," *Journal of Information Security and Applications*, vol. 54, p. 102536, 2020.