Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 561 – 582
© TÜBİTAK
doi:10.3906/elk-2001-138

Efficient hybrid passive method for the detection and localization of copy-move and spliced images

Navneet KAUR, Neeru JINDAL, Kulbir SINGH*
Electronics and Communication Engineering Department, Thapar Institute of Engineering and Technology,
Patiala, Punjab, India

Received: 29.01.2020 • **Accepted/Published Online:** 12.10.2020 • **Final Version:** 30.03.2021

Abstract: Digital passive image forgery methods are extensively used to verify the authenticity and integrity of images. Splicing and copy-move are the most common types of passive digital image forgeries. Several approaches have been proposed to detect these forgeries separately, but very few approaches are available that can detect them simultaneously. However, a more efficient method is still in demand to meet the day-to-day challenges to detect these forgeries at the same time. So, a passive hybrid approach based on discrete fractional cosine transform (DFrCT) and local binary pattern (LBP) is proposed to detect copy-move and splicing forgeries simultaneously. The extra parameter i.e. fractional parameter of DFrCT is utilized to enhance the accuracy and LBP is used to highlight the tampering artifacts effectively. Then, a support vector machine (SVM) is employed to categorize the images into authentic, copy-move, and spliced images. Next, localization is performed on both the copy-move and spliced images to localize the duplicated areas in the image. Experiments on six benchmark datasets, namely, CASIA v1.0, GRIP, CASIA v2.0, IMD, COVERAGE, and Columbia, attain accuracy rates of 99.67%, 99.23%, 99.76%, 98.81%, 95%, and 98.17%, respectively. To validate the effectiveness of the proposed method, comparative analysis has been performed with existing methods in terms of ROC, precision, recall, F_1 score, F_2 score, and accuracy. Moreover, the robustness of the proposed work is tested under rotation attack and better results are attained in comparison to the existing techniques.

Key words: Local binary pattern, discrete fractional cosine transform, copy-move, splicing

1. Introduction

In the modern era, digital images are primary sources of information sharing as they are used to interact among people all over the world. However, with the development of photo editing software such as GIMP and Photoshop, images can be easily counterfeited at a low cost, thus placing the authenticity of images at risk. Several procedures such as intrusive and nonintrusive have been introduced to identify fake images. Intrusive techniques identify tampering by authenticating the truthfulness of a watermark or signature. In contrast, nonintrusive procedures have a broad range as they only rely on examining the properties of images. Image splicing and copy-move forgery (CMF) detection are the two most common types of nonintrusive techniques. Splicing is a tampering technique in which one or more than one part of a source image is duplicated and pasted into the target image. However, in CMF, some parts of the target image are duplicated and pasted into an equivalent image to cause misinterpretation or generate misleading results [1, 2].

This paper is organized as follows: Section 2 discusses the existing splicing and copy-move procedures.

*Correspondence: ksingh@thapar.edu

Section 3 describes the proposed approach. Section 4 presents the simulation results and comparative analysis. Finally, Section 5 concludes the study.

2. Related work

This section discusses the latest procedures for detecting image forgery. In CMF, matching regions are identified in images, whereas feature irregularities are identified in splicing forgery. Several techniques have been proposed that can independently resolve the problem of copy-move and splicing forgeries. However, very few techniques can detect both the forgeries in the same image [1, 3]. Thus, developing a technique to identify forgery that meets day-to-day challenges is challenging.

Splicing forgery is difficult to detect as compared to CMF detection. In splicing forgery, a portion of the image is generally blurred, resampled, and double compressed to generate a tampered image. However, owing to the variety of splicing, several approaches have been introduced in recent years. Prakash et al. [1] extracted features through a block discrete cosine transform (BDCT) and enhanced threshold method. He et al. [4] combined Markov features in a discrete wavelet transform (DWT) and discrete cosine transform (DCT) domain. Although their paper verified the legitimacy of the Markov features, the accuracy rate was not enhanced. Zhang et al. [5] extracted the Markov features in the contourlet transform and the DCT domain to detect splicing forgery. Agarwal et al. [6] extracted internal statistical properties by using rotation invariant cooccurrence of the local binary pattern (LBP) operator. El-Alfy et al. [7] extracted Markov features in the DCT domain for forgery detection. Sheng et al. [8] detected splicing by extracting Markov features in the discrete octonion cosine transform (DOCT) domain using a support vector machine (SVM) classifier.

CMF detection is generally performed using block-based or keypoint-based procedures. Most of the block-based methods are based on DCT, zernike moments, DWT, and discrete fractional wavelet transform (DFrWT) [9, 10] for extracting features. In contrast, keypoint-based techniques include scale invariant feature transform (SIFT), speeded up robust features (SURF), and oriented FAST and rotated BRIEF (ORB) [9, 11–15]. Christlein et al. [16] examined the detection performance of various methods; among them, zernike moments proved to be efficient owing to its small memory. Li et al. [17] extracted block features using polar cosine transform (PCT) and Cozzolino et al. [18] proposed an efficient CMF detection technique using the PatchMatch algorithm, which is a fast nearest-neighbor search technique. Emam et al. [19] used discrete polar complex exponential transform (DPCET) for block feature extraction and locality sensitive hashing (LSH) for analogous block detection. However, this technique was inefficient and slow. Zandi et al. [20] detected CMF by employing the benefits of keypoint-based and block-based procedures to locate tampered areas; however, the method was not robust.

Several approaches [3, 21–23] used the combination of LBP and other techniques and achieved improved accuracy rates. However, a more accurate method is required to meet the day-to-day forgery challenges. Alamadi et al. [3] proposed a forgery detection technique based on LBP and DCT. Whereas, the proposed method has used LBP and DFrCT, as it utilizes the flexibility of an extra parameter 'fractional order' in the DFrCT. Technique in [3] only aims to categorize the image as authentic and forged. Whereas, the proposed method is used to categorize as well as localize tampered areas for the two forgeries i.e. copy-move and splicing. The primary contributions of the proposed scheme are as follows:

- The proposed work is focused to detect copy-move and splicing forgeries simultaneously and efficiently. Moreover, the localization of forged images is performed to detect the tampered areas in both types of forgeries.

- Several combinations of LBP and other approaches have been used in the literature; a more accurate approach is still required. The proposed scheme combines DFrCT and LBP, which is used for the first time according to the best knowledge of authors. It utilizes the flexibility of an extra parameter in the DFrCT i.e. fractional parameter, and LBP is used to capture the forgeries in the images by highlighting the tampering artifacts efficiently.
- The efficacy of the proposed scheme is validated by performing extensive simulations on six datasets that provide improved results in terms of various performance metrics such as precision, recall, F_1 score, F_2 score, accuracy, etc. Also, the quantitative performances of localized images have been evaluated.

3. Proposed work

The primary aim of the proposed scheme is to discover whether the given image is forged or not. If tampering is detected, then the presence of copy-move and splicing is tested using the SVM classifier. Further processing is performed to locate the forged regions in both spliced and copy-move images. Figure 1 demonstrates a detailed framework of the proposed algorithm.

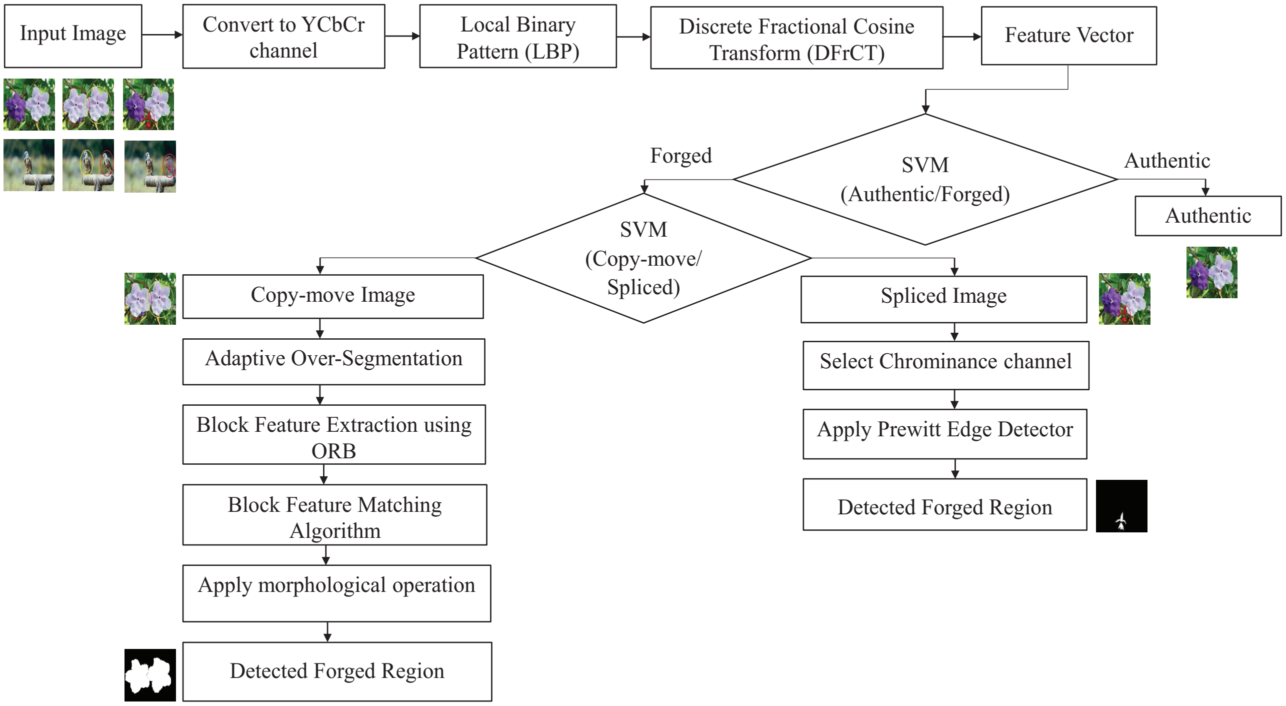

Input Image
Convert to YCbCr channel
Local Binary Pattern (LBP)
Discrete Fractional Cosine Transform (DFrCT)
Feature Vector
Forged
SVM (Authentic/Forged)
Authentic
Authentic
SVM (Copy-move/ Spliced)
Copy-move Image
Adaptive Over-Segmentation
Block Feature Extraction using ORB
Block Feature Matching Algorithm
Apply morphological operation
Detected Forged Region
Spliced Image
Select Chrominance channel
Apply Prewitt Edge Detector
Detected Forged Region

Figure 1. Detailed framework of the proposed algorithm

3.1. Preprocessing

Herein, the YCbCr color channel is used, which is a part of the RGB color channel, in which Y represents the luminance component and Cb and Cr represent the chrominance component. Most image contents are better preserved in the Y channel as compared to Cb and Cr. Human eyes are more sensitive toward luminance as compared to the chrominance channel. Although the tampered image cannot be easily identified by human eyes, few tampering artifacts are left behind in the Cr channel. Therefore, the Cr channel is used to identify the tampering artifacts. The chrominance part is obtained by subtracting the luminance part from blue (Cb=B-Y) and red (Cr=R-Y) [7]. The YCbCr image is obtained from the RGB image (whose value uses 8 bits, ranging from 0 to 255) as shown below:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \lambda \begin{bmatrix} 65.48 & 128.55 & 24.96 \\ -37.79 & -74.20 & 112 \\ 112 & -93.78 & -18.21 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \tag{1}$$

where λ is the scaling factor and B, G and R are blue, green, and red channels of the image, respectively. Figure 2 depicts the example of YCbCr, luminance, and chrominance components. The figure shows that the Cr channel of the chrominance component preserves most of the tampering artifacts. The flower's contour in the sixth column of Figure 2 is prominent compared to the other items in the image. Thus, edges that are disturbed by tampering can be easily identified in the Cr channel.

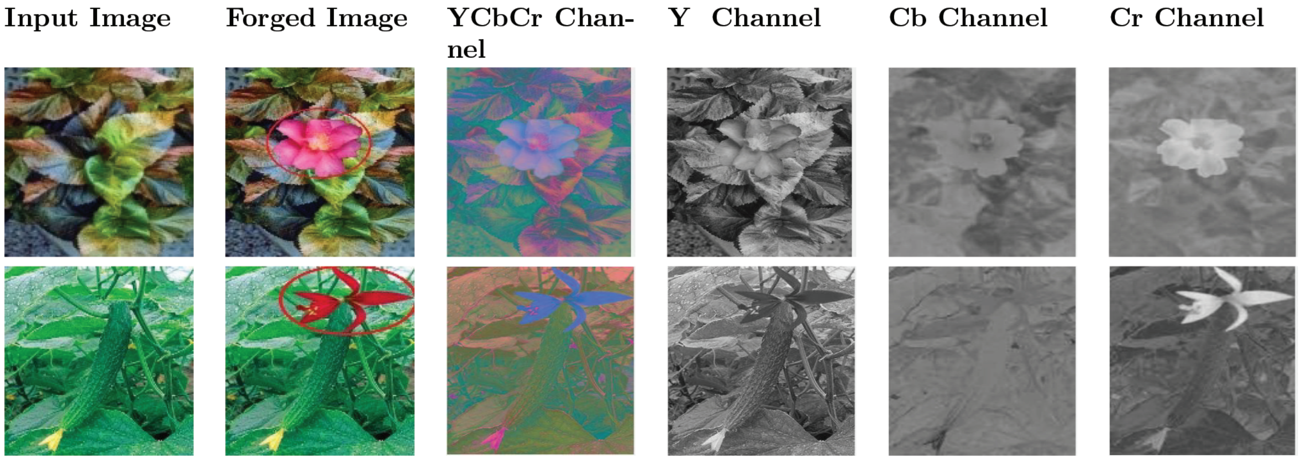


Figure 2. Example of YCbCr color channel and respective image components (Y, Cb, Cr) column-wise.

3.2. Feature extraction

Local binary pattern (LBP) is used to capture the statistical changes such as edges that occur during the copy-paste operation. The edges of the pasted area change, thus producing discontinuity along the edges of the pasted region. Consequently, there is a variation in the distribution of local frequency. Moreover, there is no relationship between the pixels of the image present in the area. Thus, capturing the statistical fluctuations is an important phase in detecting image tampering. Therefore, LBP is appropriate for highlighting tampering artifacts [3, 7, 22]. The LBP operator $(LBP_{M,N})$ is applied to each image which is defined as follows:

$$LBP_{M,N} = \sum_{i=0}^{M-1} T(m_i - m_c)2^i \tag{2}$$

where M denotes the total m_i pixels on the circular neighborhood of the present pixel m_c, N is the radius of neighborhood and $T(m_i - m_c)$ is the threshold function as represented below:

$$T(m_i - m_c) = \begin{cases} 1, & m_i - m_c \geq 0 \\ 0, & m_i - m_c < 0 \end{cases} \tag{3}$$

Further, to capture the changes in local frequency distribution of the LBP image, it is converted into the frequency domain through discrete fractional cosine transform (DFrCT). DFrCT is a generalized form of DCT containing an extra free parameter "α" which is used in every function, where DCT is beneficial [24, 25]. DCT for sequence $y[p], 0 \leq p \leq P-1$ is defined as follows:

$$Y(q) = \alpha(q) \sum_{p=0}^{P-1} y[p] cos \left[\frac{(2p+1)\pi q}{2P} \right], for \ \ 0 \leq q \leq P-1 \tag{4}$$

$$\alpha(q) = \begin{cases} \frac{1}{\sqrt{P}} & for \ q = 0 \\ \sqrt{\frac{2}{P}} & for \ \ 1 \leq q \leq P-1 \end{cases} \tag{5}$$

Because of the orthogonal sequence, the inverse DCT (IDCT) can be recovered as

$$y[p] = \sum_{q=0}^{P-1} \alpha(q) Y(q) cos \left[\frac{(2p+1)\pi q}{2P} \right], \ \ 0 \leq p \leq P-1 \tag{6}$$

The eigen decomposition of the DCT kernel is used in DFrCT and the even Hermite-Gauss eigenvectors of the Fourier matrix D_P^a are used to obtain the exclusive eigenvectors in the cosine case. In order to calculate DFrCT coefficients, the kernel matrix of N-point DFrCT is defined in [24] as:

$$K_{\mathrm{P},\alpha} = V_p D_P^a V_P^T = V_p D_P^{2\alpha/\pi} V_P^T = V_P \begin{bmatrix} 1 & & 0 \\ e^{-2j\alpha} & \ddots & \\ 0 & & e^{-j2(P-1)\alpha} \end{bmatrix} V_P^T \tag{7}$$

, where $\alpha = a\pi/2$ is the rotation angle, $V_P = [v_0|v_1| \cdots |v_{2P-2}]$, and D_P is the diagonal matrix, in which the diagonal entries have the same eigen values corresponding to the column eigenvectors of matrix V_P. In image processing applications, two-dimensional DFrCT is used. So, one dimensional DFrCT (row-wise and column-wise) is used two times to form two-dimensional DFrCT. In this, two rotation angles i.e. α and β are taken separately in two dimensions. For each image, five features i.e. mean, variance, standard deviation, skewness and kurtosis have been extracted from DFrCT coefficients for dimensionality reduction and generating a feature vector. Thus, the size of feature vector is five for each image.

3.3. SVM classification

In this section, the SVM classifier with radial basis function (RBF) kernel has been applied to classify the images into authentic, copy-move, and spliced. Some binary classification issues do not have a fundamental hyper plane as a valuable separating criterion. For those issues, there is a variation of the mathematical approach that holds all the simplicity of an SVM separating hyper plane. SVM is a standard classifier depending on the knowledge of hyperplane. The proposed work consists of two SVM classifiers that seek a decision boundary with the maximum margin for the training set and is applied in its essential nature as a binary classifier. Moreover, the computational load of the proposed scheme is reduced by using two binary SVM classifiers for three types of classes i.e. authentic, copy-move, and spliced. The SVM classifier training is performed by using the LIBSVM with a RBF (Gaussian) kernel as it provides better accuracy. The RBF kernel [26, 27] is defined as:

$$\phi(x_i, y_i) = exp(-\gamma ||x_i - y_i||^2) \quad (\gamma > 0) \tag{8}$$

The grid search method with five-fold cross validation is performed to find the optimal value γ. Figure 3 shows the two-class SVM classifier using the RBF kernel model. It comprises of a training phase and testing phase in which they are separated by a hyperplane. Let $\{x_i, z_i\}$ signifies the training set, where z_i is the target output for training data x_i and the test input vector is represented by y_i. As shown in Figure 1, the first SVM classifier is used to classify the images as authentic or forged. If the test image is authentic, then it is not further processed. The purpose of the second SVM classifier is to discriminate the forged images and classify them into two categories: copy-move and splicing forgery images [26, 27]. The forgery detection algorithm is given in Algorithm 1.

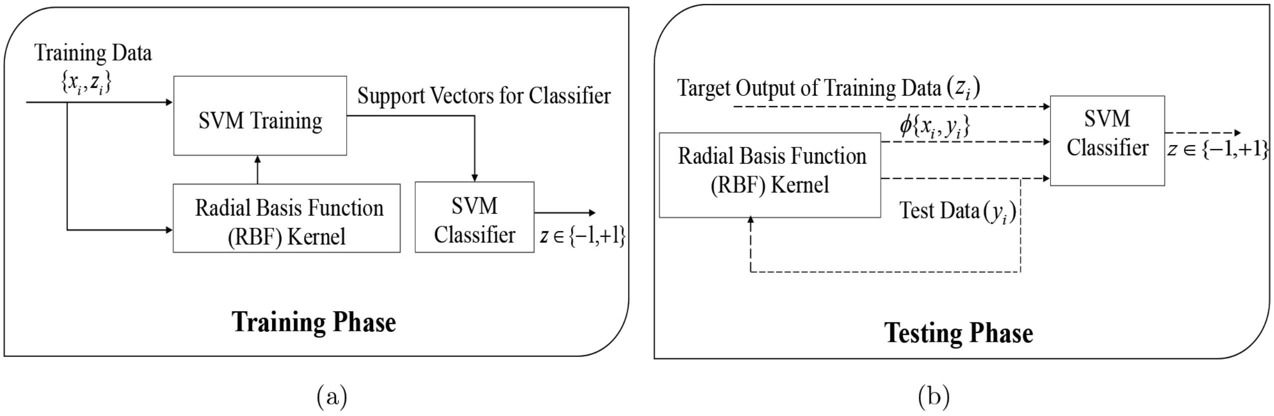


Figure 3. Block diagram of (a) training phase (b) testing phase of two-class SVM classifier using RBF kernel.

3.4. Localization of CMF

If CMF is detected, then a procedure is implemented to locate the tampered portions. The proposed work integrates a block-based and keypoint-based method for CMF detection. Initially, adaptive oversegmentation is performed on the input image using DWT and simple linear iterative clustering (SLICO) to split the image in nonoverlaying and uneven blocks. Later, ORB is implemented on each block for extracting the feature points. Then, matching is executed among the block features and finally, the forged area is detected by applying morphological operation [28–30].

In the adaptive oversegmentation technique, initially, a three-level DWT is applied for analyzing the frequency distribution of the input image. The percentage of low-frequency distribution is measured based on low-frequency and high-frequency energies, which is further used to calculate the size of super-pixels as presented in Algorithm 2. Furthermore, the SLICO algorithm segments the input image into nonoverlaying areas of

Algorithm 1 Detection of image forgery

```
Input Test image (authentic/copy-move/spliced)
Output Detection outcome whether the image is authentic or copy-move or spliced
procedure
    Convert test image $(I)$ into $YC_bC_r(Im)$
    for Each image component $YC_bC_r, Y, C_b, C_r$ do
        $Im_{LBP} \leftarrow$ Apply LBP on each image component
        $Im_{DFrCT} \leftarrow$ Apply DFrCT $(Im_{LBP})$
    end for
Combine the extracted features to attain feature vector
Apply SVM to categorize (Authentic/Forged)
    if Forged then
        Apply SVM to classify (Copy-move/Spliced)
        if Copy-move then
            $Im_x \leftarrow$ Apply DWT $(Im)$
            $Im_y \leftarrow$ Apply SLICO $(Im_x)$
            $f_1 \leftarrow$ Apply ORB $(Im_y)$
            $f_2 \leftarrow$ Feature matching $(f_1)$
            $f_3 \leftarrow$ Matched blocks $(f_2)$
            $f_4 \leftarrow$ Apply morphological operation $(f_3)$
            Detected region
        end if
        if Spliced then
            $C_r \leftarrow$ Select chrominance channel $(Im)$
            $Im_p \leftarrow$ Apply Prewitt edge detector $(C_r)$
            Detected region
        end if
    else
        Authentic
    end if
end procedure
```

unequal shape. This algorithm is an adaption of the k-means clustering technique for the effective generation of superpixels. The SLICO procedure splits the input image to achieve image blocks with a calculated size of superpixels [28, 29].

Then, the proposed approach chose ORB as the feature point extraction technique for extracting block features from the image blocks. Since ORB is efficient and faster than the existing techniques such as SIFT and SURF, it extricates the features from each image block. ORB is the fusion of feature detection and extraction methods such as FAST detector and BRIEF descriptor owing to their advantages such as low cost, good performance, and invariance to illumination and blur. Initially, the FAST detector is used for determining the keypoints. The intensity centroid (IC) approach is employed for adding an orientation component to FAST for precisely measuring the corner orientation [31]. The $(w+v)^{th}$ order moment of keypoints with varying intensity $I(w,v)$ is defined as:

$$mm_{ab} = \sum_{w,v} w^a v^b I(w,v) \tag{9}$$

Algorithm 2 Adaptive oversegmentation

```
procedure
    Input image $I$ of size $W \times V$
    Compute the low-frequency energy $L_{FE}$ and high-frequency energy $H_{FE}$
            $L_{FE}=\sum|AC_3|$
            $H_{FE}=\sum_k(\sum|DC_k|+\sum|HC_k|+\sum|VC_k|),\ k=1,2,3$
    where, $AC_3$ is the approximation coefficient of DWT and $DC_k$, $HC_k$, and $VC_k$ are the detailed coefficients
    of DWT
    Evaluate the percentage of low-frequency coefficients using the equation:
            $L_F=(L_{FE}/(L_{FE}+H_{FE}))\times 100\%$
    Compute size of superpixels $S_P$ after the evaluation of $L_F$ as given below:
      if $L_F > 50\%$, then $S_P$ is measured as:
            $S_P=((1/50)\times W\times V)^{1/2}$
      end if
      if $L_F \leq 50\%$, then $S_P$ is measured as:
            $S_P=((1/100)\times W\times V)^{1/2}$
      end if
end procedure
```

The centroid (C_e) is obtained from the moments of keypoints as represented by the following equation:

$$C_e=\left[\frac{mm_{10}}{mm_{00}},\frac{mm_{01}}{mm_{00}}\right] \tag{10}$$

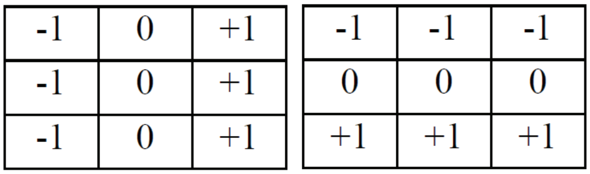
Then, a path from the center O to centroid $O\vec{C_e}$ gives the orientation ψ of keypoints.

$$\psi=atan\left[\frac{mm_{01}}{mm_{00}}\bigg/\frac{mm_{10}}{mm_{00}}\right]=atan(mm_{01},mm_{10}) \tag{11}$$

where $atan(\cdot)$ is the arctangent function. Subsequently, ORB uses the r-BRIEF; an improved form of the steered BRIEF descriptor in combination with an appropriate learning step. In the proposed procedure, features of the block are paired with other blocks for computing the correct matches among all blocks. Initially, the total paired feature points are evaluated and then a correlation coefficient map is generated. Thus, two patches are created and the two keypoints corresponding to the patches are calculated. Then, the keypoint threshold is evaluated, which localizes the matched block pairs. Finally, similar points present in the paired blocks are extricated and considered to localize the location of the doubted forgery region. The labeled feature points can locate the forged regions. The superpixels can locate the position of image forgery and divide the input image well. The morphological procedure is incorporated to identify forgery areas of the image. Finally, a binary image with the detected forged region is obtained [28, 31].

3.5. Localization of spliced forgery

If a spliced image is detected by the classifier, then further processing is performed for locating the forged region. Firstly, the Cr channel is selected from the YCbCr color channel because it preserves most of the tampering artifacts. The tampered part in the Cr channel is prominent than the other objects in the image, which is easily identified by human eyes. Then, edge detection is employed. As discussed earlier, the edges of the tampered part are different from the other parts of the input image. Therefore, edge detection plays a significant role in locating tampered parts. A Prewitt edge detector is used to compute the magnitude and orientation of the image. Prewitt is extensively used to detect vertical and horizontal edges of an image to identify parts in the image where the intensity changes quickly [32]. The Prewitt edge detector contains a couple of 3×3 convolution kernels as shown in Figure 4.



-1	0	+1
-1	0	+1
-1	0	+1

-1	-1	-1
0	0	0
+1	+1	+1

Figure 4. Prewitt operator's 3×3 mask

The highest response of all the kernels for the pixel location is used to compute the local edge gradient magnitude as follows:

$$|E| = max(|E_j|, j = 1 : s) \tag{12}$$

where E_j is the response of kernel j at a suitable position of the pixel, and s is the number of convolution kernels. The horizontal and vertical gradients are calculated and merged. Then, the threshold is applied by setting the threshold value. Consequently, the tampered portion is highlighted in the image.

4. Performance Analysis

4.1. Description of datasets

In the experimentation, the efficiency of the proposed approach is demonstrated by using six datasets, namely, CASIA v1.0, GRIP, CASIA v2.0, IMD, COVERAGE, and Columbia. CASIA v1.0 comprises 800 authentic and 921 forged images. CASIA v2.0 is a larger dataset, which comprises 7491 authentic and 5123 forged images. Both CASIA v1.0 and CASIA v2.0 comprise splicing and copy-move images [33]. Further, the Columbia dataset consists of 183 authentic and 180 tampered images with image sizes ranging from 757×568 to 1152×768 pixels. The GRIP dataset comprises 80 authentic and 80 tampered images of the same size, i.e., 768×1024. The image manipulation dataset (IMD) is composed of 48 authentic and 48 forged images with an average resolution of approximately 3000×2300. The COVERAGE dataset has 100 authentic and 100 forged images of average resolution 400×486 [15, 17, 34, 35].

4.2. Performance metrics

The effectiveness of the proposed scheme is calculated using various performance metrics at image level, such as detection accuracy (DA), recall (R) or true positive rate (TPR), F_1 score (F_1), F_2 score (F_2), precision (P), true negative rate (TNR), and Mathews correlation coefficient (MCC), informedness (Inf), markedness (Mkd). DA is the proportion of summation of true positives and true negatives to the overall images used in the experiment. Sensitivity (also called TPR or R) is the possibility of classifying a forged image as forged. Specificity (also known as TNR) is the possibility of classifying an authentic image as authentic. Precision is the possibility that detected image is truly forged. F_1 score merges both recall and precision in a single value. F_2 score is an average of recall and precision. Informedness states the possibility that a classifier is informed about the condition and markedness enumerates the possibility that the condition is marked by the classifier. Moreover, MCC is the correlation coefficient between authentic and predicted classes for the classifier [7, 36].

These parameters are represented by the following equations:

$$DA = \frac{T_N + T_P}{T_N + T_P + F_N + F_P} \tag{13}$$

$$Sensitivity = TPR = R = \frac{T_P}{T_P + F_N} \tag{14}$$

$$Specificity = TNR = \frac{T_N}{T_N + F_P} \tag{15}$$

$$P = \frac{T_P}{T_P + F_P} \tag{16}$$

$$F_1 = 2\frac{P \cdot R}{P + R} \tag{17}$$

$$F_2 = 5\frac{P \cdot R}{4 \cdot P + R} \tag{18}$$

$$MCC = \frac{T_P \times T_N - F_P \times F_N}{\sqrt{((T_P + F_P)(T_P + F_N)(T_N + F_P)(T_N + F_N))}} \tag{19}$$

$$Inf = TPR + TNR - 1 \tag{20}$$

$$Mkd = \frac{T_P}{T_P + F_P} + \frac{T_N}{T_N + F_N} - 1 \tag{21}$$

where F_P represents the images that are inaccurately identified as forged, T_P represents accurately detected forged images, T_N represents accurately detected authentic images, and F_N represents mistakenly identified authentic images.

4.3. Simulation results

As discussed, the tampering artifacts present in the forged image are highlighted by the LBP operator. Then, DFrCT transforms the LBP image into the frequency domain for capturing the local frequency fluctuations produced by these artifacts. The LBP operator is applied to the image. There are two parameters of LBP: M is total pixels in the circular neighborhood, and N is its radius [22, 37]. In LBP, for 3×3 image blocks, each central pixel (m_c) is compared with its eight neighbors ($m_i, i = 0 : M - 1$). If the value of neighbors have lesser value than that of the central pixel, then it will hold binary digit '0', and if other neighbors have value equal to or more than that of the central pixel, then it will hold binary digit '1'. For each given central pixel, binary code is obtained by concatenating all these binary digits in a clockwise manner, which starts from the one of its top-left neighbor. The central pixel value is replaced by the generated binary code and the LBP code is decimal value of that binary code. The calculation of LBP code is given in Figure 5.

The extensive experiments have been performed on CASIA v1.0 database (dataset) with different LBP parameters (M, N). The various combinations of M and N i.e. (8, 1), (16, 2) and (24, 3) are considered in the experimentation as shown in Figure 6 and it is observed that the best performance is attained by using M=8 and

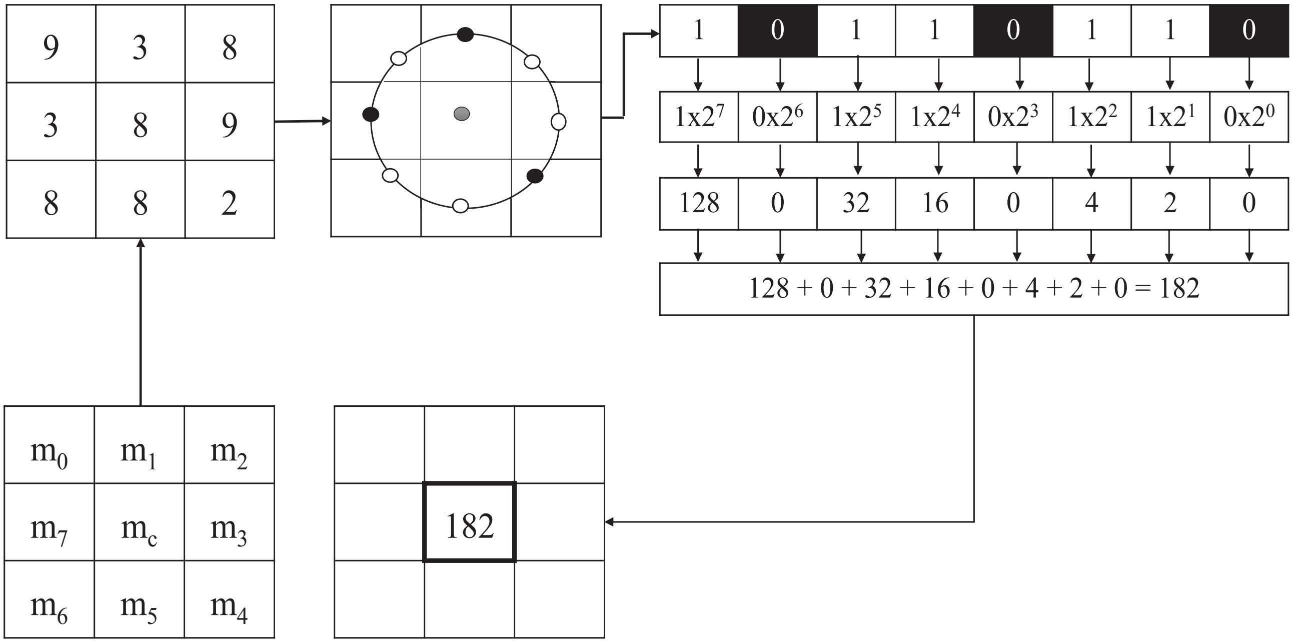


Figure 5. Calculation of LBP code

N=1. Also, it should be noted that the accuracy rate decreases with the increase in LBP parameters (M,N). The reason that higher values of LBP parameters do not give better performance is that when higher values are chosen, small scale features which are highly discriminative are ignored and thus, performance accuracy decreases. Moreover, the number of LBP codes becomes unmanageable since it increases exponentially with M; the number of LBP codes is 2^M if (M,N) is used. For example, when M=8, the value of LBP code becomes 256. Furthermore, it is also confirmed from [3, 22, 37] that the best performance is achieved using (8, 1) LBP parameters. As a result, the next simulations are executed using same optimal values of LBP parameters i.e. (8, 1) for other databases as well.

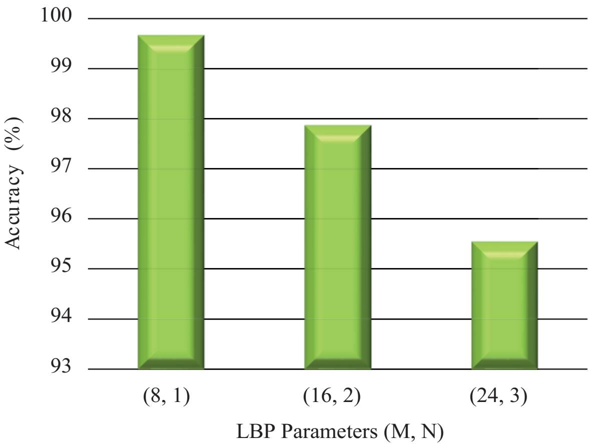
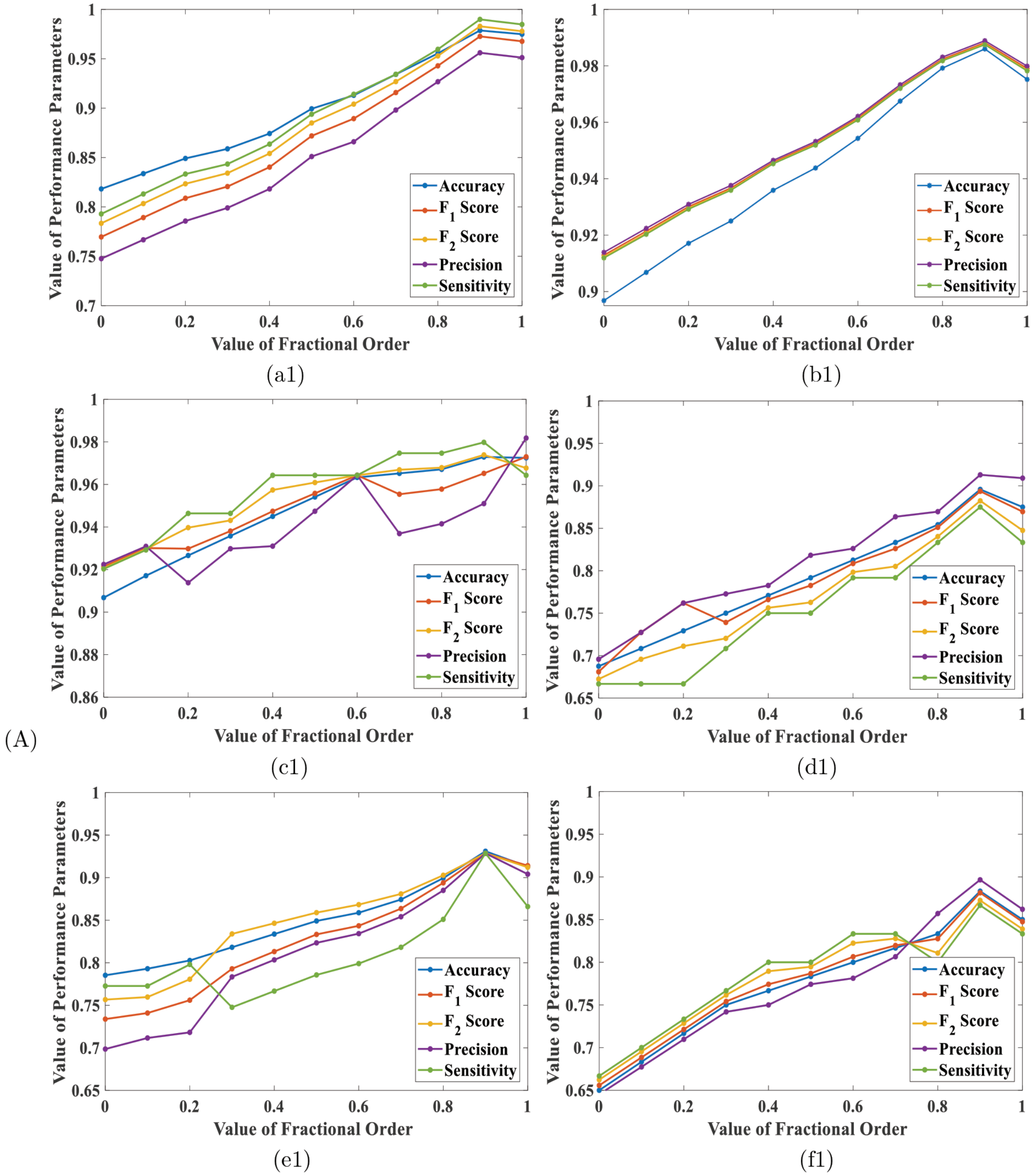


Figure 6. The effect of LBP parameters (M,N) on the accuracy

Moreover, the effect of fractional order values is calculated as it produces different DFrCT features. Figure 7 shows the performance of various parameters such as sensitivity, F_1 score, precision, accuracy, and F_2 score with respect to fractional order "α". Initially, the fractional order is varied from 0 to 1 in the steps of 0.1 as shown in Figure 7 (A). It has been observed that the value of performance parameters is better for fractional order between 0.9 and 1. Further, fractional order is varied from 0.9 to 1 in the steps of 0.01 as shown in Figure 7 (B). It has been noted that for CASIA v1.0 and CASIA v2.0 datasets, the proposed technique provides the best results at fractional order α=0.98. For Columbia, GRIP, COVERAGE, and IMD datasets, the best results are obtained at fractional order α=0.97, 0.99, 0.94 and 0.96, respectively.



(a1)

(b1)

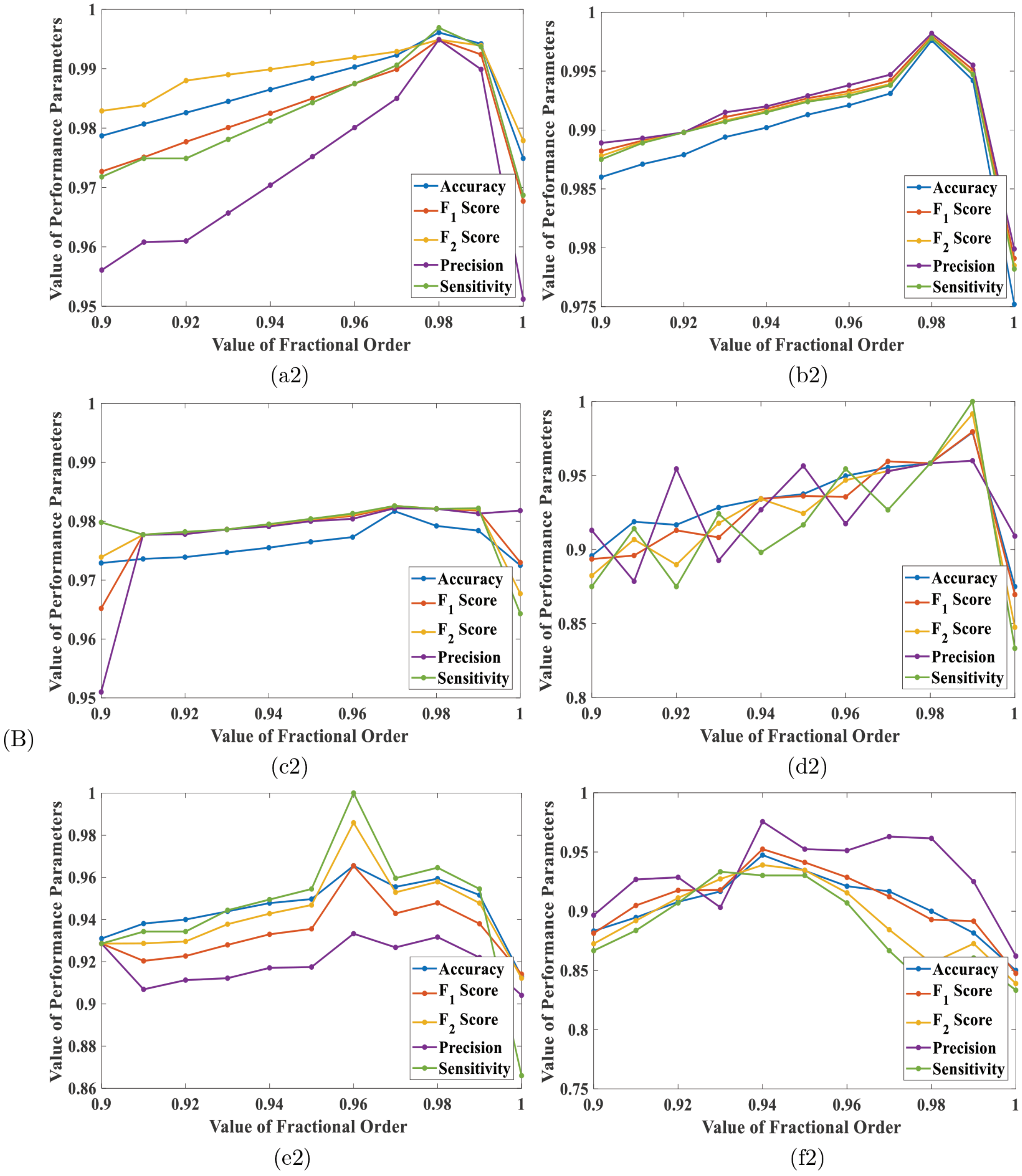
(A)

(c1)

(d1)

(e1)

(f1)



(a2)

(b2)

(c2)

(d2)

(e2)

(f2)

Figure 7. Value of performance parameters for different fractional orders varying from (A) 0 to 1 and (B) 0.9 to 1 for CASIA v1.0 (a1-a2), CASIA v2.0 (b1-b2), Columbia (c1-c2), GRIP (d1-d2), IMD (e1-e2), COVERAGE (f1-f2) datasets

Also, separation and aggregation of various color channels, i.e., Y, Cb, Cr, and YCbCr are evaluated. Table 1 shows different performance parameters for all six datasets for different color channels. It is observed that the values of performance parameters for Cr channel are greater as compared to Y, Cb, and YCbCr color channels, indicating that the Cr channel performs the best for the proposed scheme. In contrast, the Y channel performs worst in comparison to all the color channels (YCbCr, Cb, and Cr) in terms of various performance parameters.

Table 1. Performance parameters (%) for different color channels on various datasets

Datasets	Color Channel	Accuracy	TPR	TNR	Precision	F_1 Score	F_2 Score	MCC	Inf	Mkd
CASIA v1.0	YCbCr	99.42	99.49	99.37	98.99	99.24	99.39	98.77	98.87	98.68
	Y	98.84	98.99	98.74	98.00	98.49	98.79	97.55	97.73	97.37
	Cb	99.22	98.99	99.37	98.99	98.99	98.99	98.36	98.36	98.36
	Cr	99.67	99.56	99.74	99.56	99.56	99.56	99.30	99.30	99.30
CASIA v2.0	YCbCr	99.71	99.73	99.68	99.78	99.76	99.74	99.40	99.41	99.39
	Y	99.55	99.60	99.48	99.64	99.62	99.61	99.07	99.08	99.06
	Cb	99.60	99.64	99.55	99.69	99.67	99.65	99.18	99.19	99.17
	Cr	99.76	99.78	99.74	99.82	99.80	99.79	99.51	99.52	99.50
Columbia	YCbCr	97.25	96.43	98.11	98.18	97.30	96.77	94.51	94.54	94.58
	Y	95.41	96.43	94.34	94.74	95.58	96.09	90.83	90.77	90.89
	Cb	96.33	96.43	96.23	96.43	96.43	96.43	92.65	92.65	92.65
	Cr	98.17	98.21	98.11	98.21	98.21	98.21	96.33	96.33	96.33
IMD	YCbCr	96.55	100	93.33	93.33	96.55	98.59	93.33	93.33	93.33
	Y	89.66	92.86	86.67	86.67	89.66	91.55	79.52	79.52	79.52
	Cb	93.10	100	86.67	87.50	93.33	97.22	87.08	86.67	87.50
	Cr	98.81	100	97.14	98.00	98.99	99.59	97.57	97.14	98.00
GRIP	YCbCr	97.92	100	95.83	96.00	97.96	99.17	95.92	95.83	96.00
	Y	93.75	95.83	91.67	92.00	93.88	95.04	87.58	87.50	87.65
	Cb	95.83	100	91.67	92.31	96.00	98.36	91.99	91.67	92.31
	Cr	99.23	100	98.44	98.51	99.25	99.70	98.47	98.44	98.51
COVERAGE	YCbCr	93.33	90.00	96.67	96.43	93.10	93.27	86.86	86.67	87.05
	Y	90.00	83.33	96.67	96.15	89.29	85.62	80.72	80.00	81.45
	Cb	91.67	86.67	96.67	96.30	91.23	88.44	83.75	83.33	84.18
	Cr	95.00	96.67	93.33	93.55	95.08	96.03	90.05	90.00	90.10

4.4. Localization results

The uniqueness of the proposed algorithm is that localization of the tampered part is performed on both splicing and CMF after identifying the presence of forgery in the image. To quantitatively evaluate the performance of localization of the images, three pixel-level metrics are calculated for forged regions of a detected forged image i.e. precision (P_p), recall (R_p) and F_1 score (F_p). These pixel-level metrics are beneficial for evaluating the general localization performance of the algorithm [38]. At pixel-level, the precision is defined as the ratio of the number of correctly detected forged pixels to the number of totally detected forged pixels and recall is defined as the ratio of the number of correctly detected forged pixels to the number of forged pixels in the ground-truth forged image.

$$P_p = \Omega/\Omega_1 \tag{22}$$

$$R_p = \Omega/\Omega_2 \tag{23}$$

where, Ω is number of correctly detected forged pixels, Ω_1 is total detected forged pixels, and Ω_2 is number of forged pixels in the ground-truth forged image. F_p merges both recall and precision in a single value as given below:

$$F_p = 2\frac{P_p \cdot R_p}{P_p + R_p} \tag{24}$$

The value of these metrics and detection results are given in Figure 8 and 9 for copy-move and spliced images, respectively.

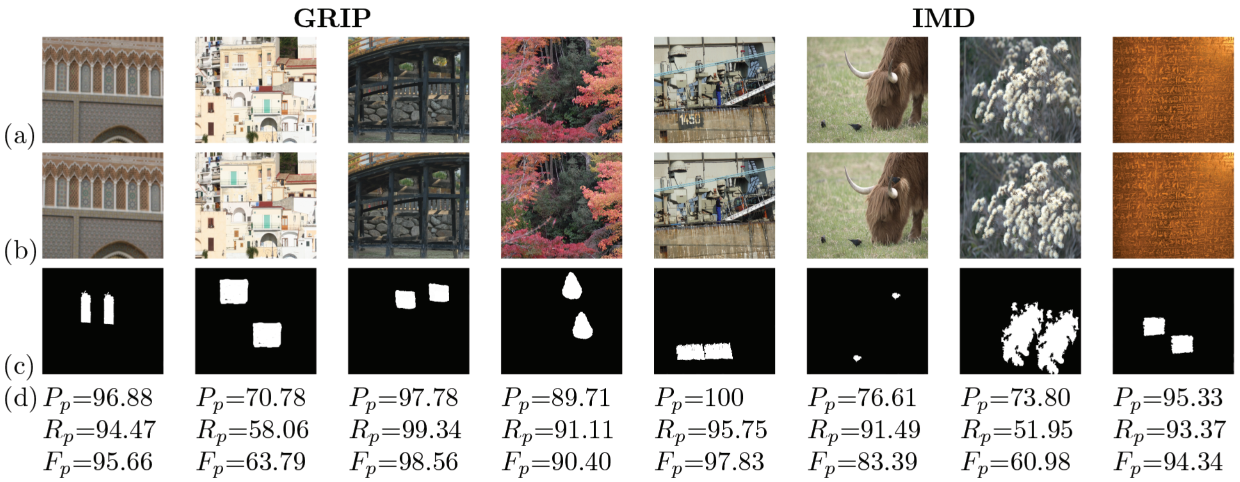

GRIP
IMD
(a)
(b)
(c)
(d) P_p=96.88 R_p=94.47 F_p=95.66
P_p=70.78 R_p=58.06 F_p=63.79
P_p=97.78 R_p=99.34 F_p=98.56
P_p=89.71 R_p=91.11 F_p=90.40
P_p=100 R_p=95.75 F_p=97.83
P_p=76.61 R_p=91.49 F_p=83.39
P_p=73.80 R_p=51.95 F_p=60.98
P_p=95.33 R_p=93.37 F_p=94.34

Figure 8. CMF detection results of the proposed method for GRIP, and IMD datasets: (a) authentic images, (b) forged images, and (c) detection results, and (d) performance metrics of each localized copy-move image

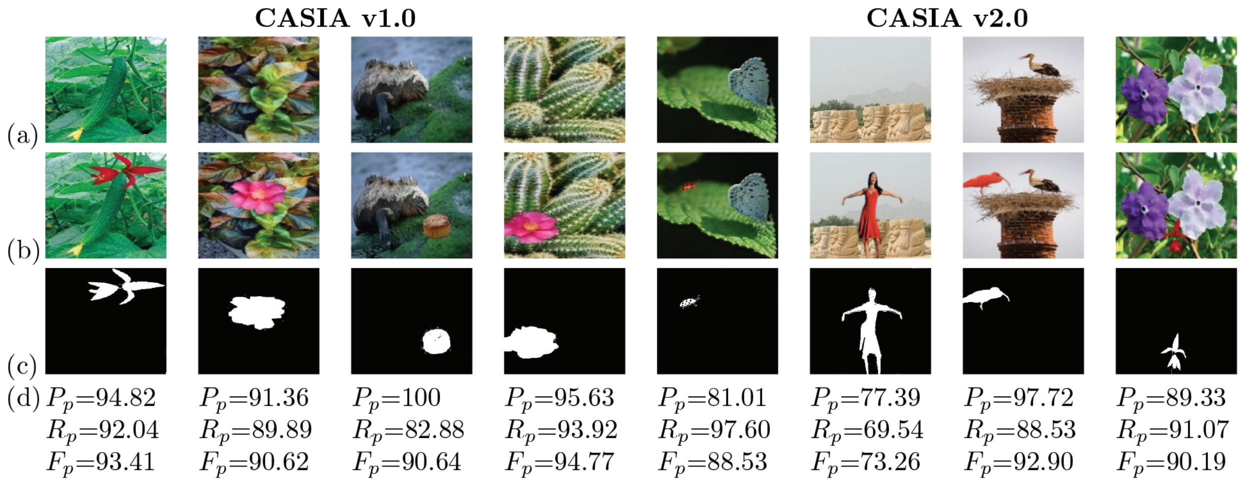

CASIA v1.0
CASIA v2.0
(a)
(b)
(c)
(d) P_p=94.82 R_p=92.04 F_p=93.41
P_p=91.36 R_p=89.89 F_p=90.62
P_p=100 R_p=82.88 F_p=90.64
P_p=95.63 R_p=93.92 F_p=94.77
P_p=81.01 R_p=97.60 F_p=88.53
P_p=77.39 R_p=69.54 F_p=73.26
P_p=97.72 R_p=88.53 F_p=92.90
P_p=89.33 R_p=91.07 F_p=90.19

Figure 9. Splicing localization of the proposed method for CASIA v1.0 and CASIA v2.0 datasets: (a) authentic images, (b) forged images, (c) detection results, and (d) performance metrics of each localized spliced image

4.5. Comparative analysis

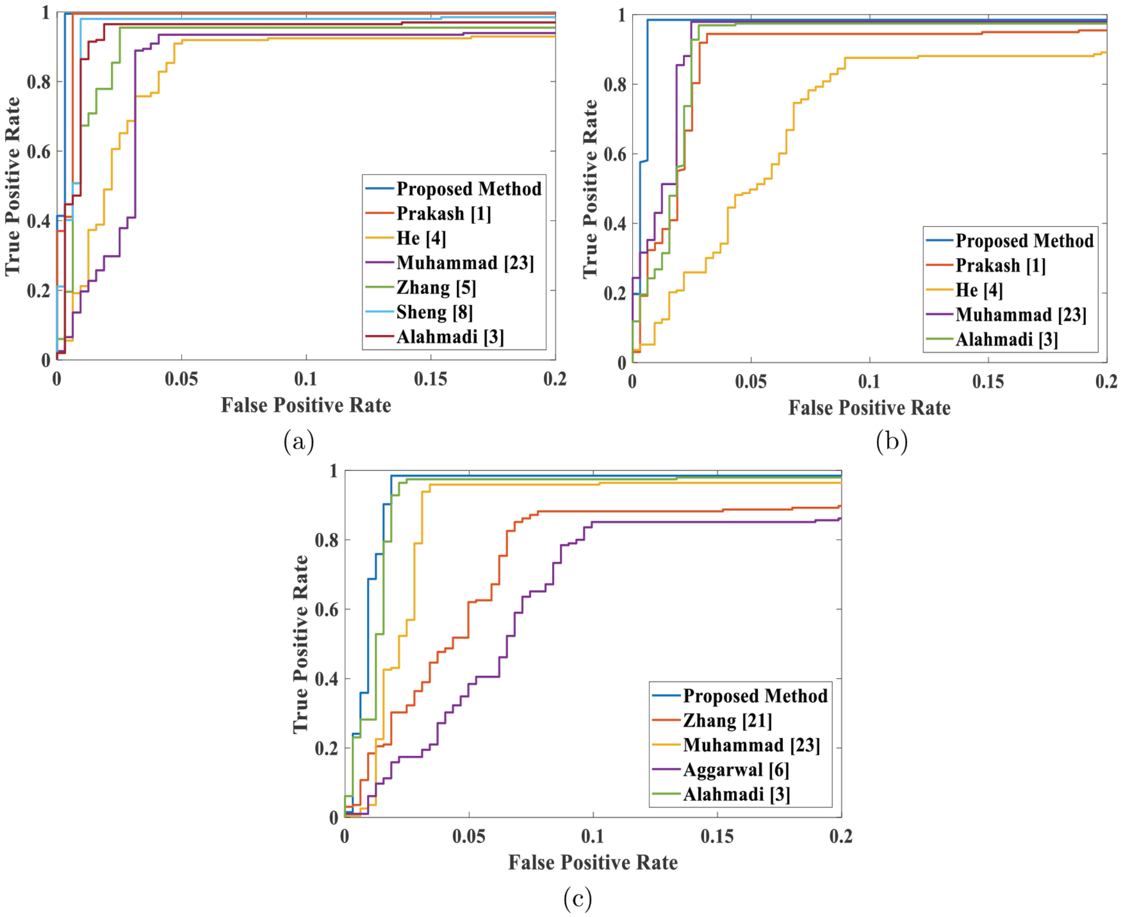
To determine the efficacy of the proposed work, a comparative analysis is performed with the existing algorithms [1, 3–8, 10, 21–23] as illustrated in Table 2. For CASIA v1.0, the proposed technique achieves 99.67% accuracy, 99.56% TPR, 99.74% TNR, and 99.30% Inf, indicating that the technique is accurate for authentic and forged image detection. Similarly, the proposed scheme outperforms the existing schemes for CASIA v2.0 with 99.76% accuracy, 99.78% TPR, and 99.52% Inf. However, the TNR value of the proposed algorithm is slightly lower than that of the algorithm proposed by El-Alfy [7]. Moreover, the accuracy rate of the proposed scheme is higher in comparison to Lamba [10] for CASIA v1.0 and CASIA v2.0 datasets. Also, the proposed scheme has less computational time in comparison to Lamba [10] as given in Section 4.7. Furthermore, the experimental results for the Columbia dataset achieved an accuracy of 98.17%, TPR of 98.21%, TNR of 98.11% and Inf of 96.33%. It is observed from Table 2 that the proposed work achieves better results than the existing techniques.

Table 2. Comparison results with existing methods on CASIA v1.0, Columbia and CASIA v2.0 datasets.

Datasets	**Techniques**	**Accuracy(%)**	**TPR(%)**	**TNR(%)**	**Inf(%)**
CASIA v1.0	He [4]	94.29	—	—	—
	Muhammad [23]	94.89	95.15	93.91	89.06
	Zhang [5]	96.69	98.05	95.31	93.36
	Alahmadi [3]	97.50	96.75	98.24	94.99
	Sheng [8]	98.77	—	—	—
	Lamba [10]	99.65	—	—	—
	Prakash [1]	99.45	99.37	99.50	98.87
	Proposed method	**99.67**	**99.56**	**99.74**	**99.30**
CASIA v2.0	He [4]	89.76	—	—	—
	Muhammad [23]	97.33	98.50	96.53	95.03
	Alahmadi [3]	97.50	98.31	96.88	96.88
	El-Alfy [7]	99.73	99.70	**99.76**	99.46
	Sheng [8]	97.59	—	—	—
	Lamba [10]	99.01	—	—	—
	Prakash [1]	96.68	95.77	97.52	93.29
	Proposed method	**99.76**	**99.78**	99.74	**99.52**
Columbia	Zhang [21]	91.38	—	—	—
	Alahmadi [22]	96.60	—	—	—
	Muhammad [23]	96.39	—	—	—
	Agarwal [6]	93.81	—	—	—
	Alahmadi [3]	97.77	—	—	—
	Proposed method	**98.17**	**98.21**	**98.11**	**96.33**

"—" indicates not calculated by the author

Further, Figure 10 shows the comparison of receiver operating characteristics (ROC) curves for CASIA v1.0, Columbia, and CASIA v2.0 datasets. The ROC curves for the datasets are zoomed for better visualization.The ROC curve visualizes the classifier's performance. Similarly, it is used to define the advancement of TPR and the false positive rate (FPR). TPR represents the number of tampered images that are accurately categorized as tampered. Similarly, FPR signifies tampered images that are mistakenly categorized as authentic. It is observed that the ROC curve of the proposed algorithm is closer to the upper left corner, which depicts that it attains a better accuracy rate in comparison to the existing schemes.



(a)

(b)

(c)

Figure 10. Comparison of ROC curves for (a) CASIA v1.0, (b) CASIA v2.0, and (c) Columbia datasets

Further, Table 3 depicts the comparative analysis of the proposed algorithm with other algorithms [9, 11–20] on GRIP, COVERAGE, and IMD datasets. For the IMD dataset, the sensitivity of 100%, precision of 98%, F_1 score of 98.99%, and 99.59% F_2 score are achieved. For the COVERAGE dataset, the proposed scheme attains a sensitivity of 96.67%, the precision of 93.55%, F_1 score of 95.08%, and 96.03% F_2 score. Furthermore, the GRIP dataset achieves 100% sensitivity, 98.51% precision, 99.25% F_1 score, and 99.70% F_2 score. Thus, the results reveal that the proposed method outperforms other existing methods. However, the proposed scheme achieves a slightly lower value of F_1 score than Li [14] on the GRIP dataset. This is because the technique in [14] has solved matching problem over a huge number of keypoints, but at the cost of large computational load. Also, Li [14] only deals with CMF, however, the proposed technique deals with the detection and localization of two types of forgeries i.e. CMF and splicing.

4.6. Detection results under rotation attack

Also, the robustness of the proposed scheme has been computed against geometrical attack i.e. rotation attack. In this case, fake images are created by using 48 images of the IMD dataset and the copied regions are attacked by rotation attack. The copied regions are rotated with the rotation angle of 2^o to 10^o. In this case, a

Table 3. Comparison with existing methods on IMD, GRIP and COVERAGE datasets

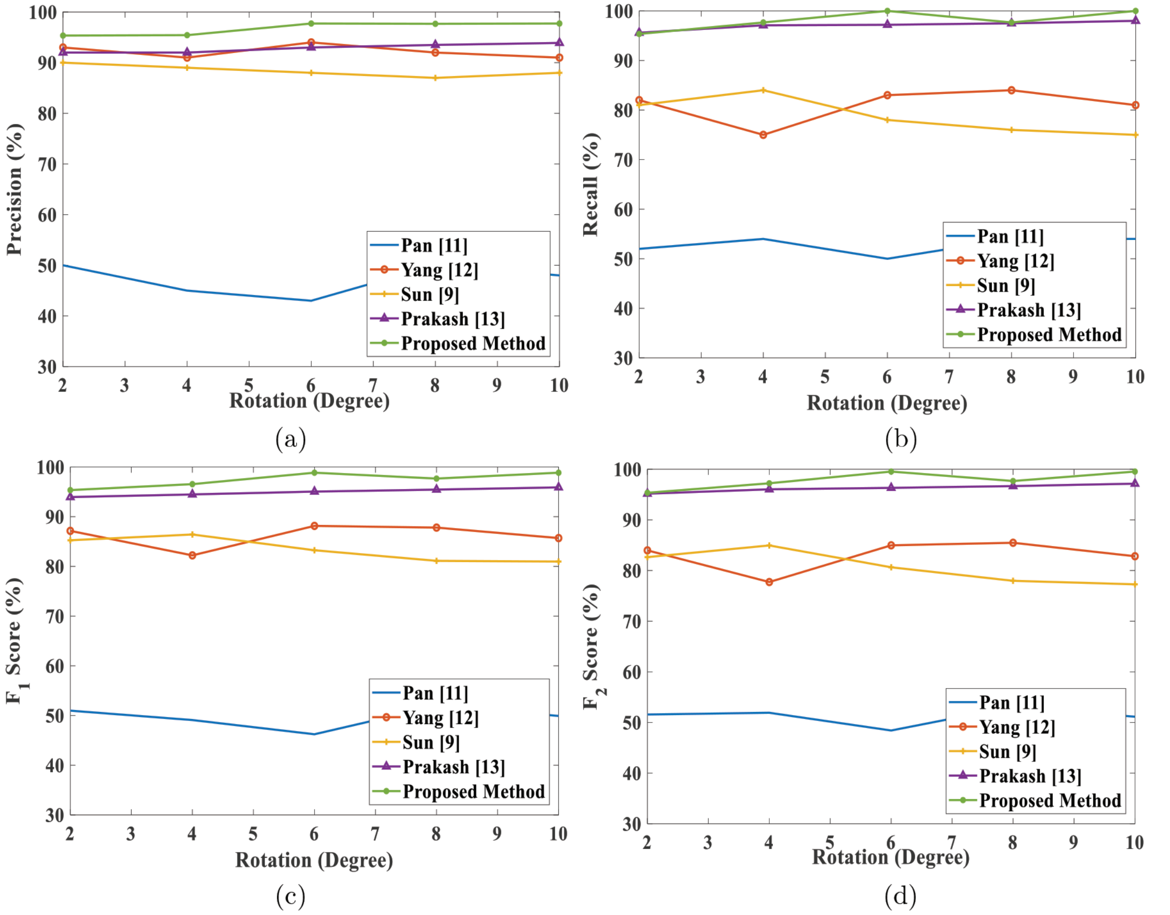
Datasets	Techniques	Sensitivity(%)	Precision(%)	F_1 Score (%)	F_2 Score (%)
IMD	Pan [11]	79.17	88.37	83.52	80.92
	Amerini [15]	79.20	88.40	83.54	80.88
	Emam [19]	87.50	92.70	90.02	88.49
	Yang [12]	78.61	90.27	84.04	80.69
	Sun [9]	83.33	90.91	86.96	84.74
	Li [14]	100	—	98.97	—
	Prakash [13]	87.80	92.30	89.98	88.65
	Proposed method	**100**	**98.00**	**98.99**	**99.59**
GRIP	Amerini [15]	70.00	77.56	73.68	71.39
	Christlein [16]	100	74.76	85.56	93.67
	Li [17]	83.75	70.52	76.57	80.72
	Cozzolino [18]	98.75	91.85	95.18	97.28
	Li [14]	100	—	**100**	—
	Zandi [20]	100	74.76	85.56	93.67
	Proposed method	**100**	**98.51**	99.25	**99.70**
COVERAGE	Amerini [15]	85.71	40.43	54.95	70.02
	Christlein [16]	46.15	75.00	57.14	49.99
	Li [14]	80.22	—	72.28	—
	Cozzolino [18]	59.34	61.97	65.45	67.72
	Proposed method	**96.67**	**93.55**	**95.08**	**96.03**

"—" indicates not calculated by the author

test is performed on a total of 48 × 5=240 images. Figure 11 shows the graphical illustration of the detection results under rotation attack for precision, recall, F_1 score, and F_2 score. In this figure, the rotation degree is represented along the x-axis. It is observed from the graphs, that the value of performance parameters of the proposed scheme is much better in comparison to other techniques.

4.7. Computational load

In this section, computational load has been calculated and compared with existing methods. Computational load is the time taken by the algorithm to execute images. Likewise, the large feature vector length possesses high probability of increasing the computational load. Lamba [10] has compared block sizes 4×4, 8×8, 16×16 and achieved better accuracy with block size of 16×16 with feature vector length 14 on CASIA dataset. Our proposed technique use five features (almost one-third of [10]), and reduced the computation time approximate 6-7 times as given in Table 4. However, limitation of proposed technique is that block size is not considered as DFrCT gives blocking artifacts in comparison to DFrWT used by [10]. Table 5 shows the average computational time comparison of proposed method and Li [14] on IMD, GRIP and COVERAGE datasets. It has been observed that the proposed method is computationally more efficient.



(a)

(b)

(c)

(d)

Figure 11. Detection results under rotation attack for (a) precision (b) recall (c) F_1 score and (d) F_2 score

Table 4. Comparison of computational load on CASIA v1.0 dataset

Techniques	Computational time(sec)	Feature length	Block size
Lamba [10]	140-150 sec depending on image size	14	16 ×16
Proposed method	**7-20 sec depending on image size**	**5**	—

Table 5. Comparison of average computational load(sec) on IMD, GRIP and COVERAGE datasets

Techniques	IMD	GRIP	COVERAGE
Li [14]	86.6	13.9	2.3
Proposed method	**33.2**	**9.4**	**1.9**

5. Conclusion

This study proposed an image tampering detection algorithm that can detect both copy-move and splicing forgery together. At the outset, an input image is converted into the YCbCr color channel. Then, for each image, the local binary pattern is calculated and transformed in the frequency domain using DFrCT to capture the local frequency distributions. Further, five features i.e. mean, variance, standard deviation, skewness and kurtosis have been extracted from DFrCT coefficients to produce a feature vector. Subsequently, the system is trained with original and tampered images after attaining the feature vector. Then, SVM is applied to categorize the images. After identification of spliced and copy-move images, localization is performed to detect the tampered part in the image. Adaptive over-segmentation and ORB are used to locate the forged region in copy-move images. The Prewitt edge operator is applied to locate the tampered region in spliced images. The simulation results reveal that the Cr channel extricates the features in the proposed algorithm; as it performs better than other color channels and preserves most of the tampering artifacts. The proposed scheme is intensively evaluated on six standard datasets, namely, CASIA v1.0, GRIP, CASIA v2.0, IMD, COVERAGE and Columbia and accuracy rates of 99.67%, 99.23%, 99.76%, 98.81%, 95%, and 98.17%, respectively, are achieved. Furthermore, the proposed algorithm outperforms the existing schemes with regard to different performance parameters. Also, the simulation results show that the proposed scheme can detect tampering areas even in the presence of rotation attack. In future, other types of geometrical attacks like scaling, JPEG compression, etc., can be performed to validate the robustness of the proposed scheme.

References

[1] Prakash CS, Kumar A, Maheshkar S, Maheshkar V. An integrated method of copy-move and splicing for image forgery detection. Multimedia Tools and Applications 2018; 77 (20): 26939-26963. doi: 10.1007/s11042-018-5899-3

[2] Teerakanok S, Uehara T. Copy-Move Forgery Detection: A State-of-the-Art Technical Review and Analysis. IEEE Access 2019; 7 (1): 40550-40568. doi: 10.1109/ACCESS.2019.2907316

[3] Alahmadi A, Hussain M, Aboalsamh H, Muhammad G, Bebis G et al. Passive detection of image forgery using DCT and local binary pattern. Signal, Image and Video Processing 2017; 11 (1): 81-88. doi: 10.1007/s11760-016-0899-0

[4] He Z, Lu W, Sun W, Huang J. Digital image splicing detection based on Markov features in DCT and DWT domain. Pattern Recognition 2012; 45 (12): 4292-4299. doi: 10.1016/j.patcog.2012.05.014

[5] Zhang Q, Lu W, Weng J. Joint image splicing detection in DCT and Contourlet transform domain. Journal of Visual Communication and Image Representation 2016; 40 (1): 449-458. doi: 10.1016/j.jvcir.2016.07.013

[6] Agarwal S, Chand S. Image forgery detection using Markov features in undecimated wavelet transform. In: 2016 Ninth International Conference on Contemporary Computing (IC3); Noida, India; 2016. pp. 1-6.

[7] El-Alfy ES, Qureshi MA. Robust content authentication of gray and color images using lbp-dct markov-based features. Multimedia Tools and Applications 2017; 76 (12): 14535-14556. doi: 10.1007/s11042-016-3855-7

[8] Sheng H, Shen X, Lyu Y, Shi Z, Ma S. Image splicing detection based on Markov features in discrete octonion cosine transform domain. IET Image Processing 2018; 12 (10): 1815-1823. doi: 10.1049/iet-ipr.2017.1131

[9] Sun Y, Ni R, Zhao Y. Nonoverlapping blocks based copy-move forgery detection. Security and Communication Networks 2018; 2018 (1): 1-11. doi: 10.1155/2018/1301290

[10] Lamba AK, Jindal N, Sharma S. Digital image copy-move forgery detection based on discrete fractional wavelet transform. Turkish Journal of Electrical Engineering & Computer Science 2018; 26 (3): 1261-1277. doi: 10.3906/elk-1701-275

[11] Pan X, Lyu S. Region duplication detection using image feature matching. IEEE Transactions on Information Forensics and Security 2010; 5 (4): 857-867. doi: 10.1109/TIFS.2010.2078506

[12] Yang F, Li J, Lu W, Weng J. Copy-move forgery detection based on hybrid features. Engineering Applications of Artificial Intelligence 2017; 59 (1): 73-83. doi: 10.1016/j.engappai.2016.12.022

[13] Prakash CS, Panzade PP, Om H, Maheshkar S. Detection of copy-move forgery using AKAZE and SIFT keypoint extraction. Multimedia Tools and Applications 2019; 78 (16): 23535-23558. doi: 10.1007/s11042-019-7629-x

[14] Li Y, Zhou J. Fast and effective image copy-move forgery detection via hierarchical feature point matching. IEEE Transactions on Information Forensics and Security 2018; 14 (5): 1307-1322. doi: 10.1109/TIFS.2018.2876837

[15] Amerini I, Ballan L, Caldelli R, Del Bimbo A, Serra G. A sift-based forensic method for copy–move attack detection and transformation recovery. IEEE Transactions on Information Forensics and Security 2011; 6 (3): 1099-1110. doi: 10.1109/TIFS.2011.2129512

[16] Christlein V, Riess C, Jordan J, Riess C, Angelopoulou E. An evaluation of popular copy-move forgery detection approaches. IEEE Transactions on Information Forensics and Security 2012; 7 (6): 1841-1854. doi: 10.1109/TIFS.2012.2218597

[17] Li J, Li X, Yang B, Sun X. Segmentation-based image copy-move forgery detection scheme. IEEE Transactions on Information Forensics and Security 2014; 10 (3): 507-518. doi: 10.1109/TIFS.2014.2381872

[18] Cozzolino D, Poggi G, Verdoliva L. Efficient dense-field copy–move forgery detection. IEEE Transactions on Information Forensics and Security 2015; 10 (11): 2284-2297. doi: 10.1109/TIFS.2015.2455334

[19] Emam M, Han Q, Niu X. PCET based copy-move forgery detection in images under geometric transforms. Multimedia Tools and Applications 2016; 75 (18): 11513-11527. doi: 10.1007/s11042-015-2872-2

[20] Zandi M, Mahmoudi-Aznaveh A, Talebpour A. Iterative copy-move forgery detection based on a new interest point detector. IEEE Transactions on Information Forensics and Security 2016; 11 (11): 2499-2512. doi: 10.1109/TIFS.2016.2585118

[21] Zhang Y, Zhao C, Pi Y, Li S. Revealing image splicing forgery using local binary patterns of DCT coefficients. In: Communications, Signal Processing, and Systems; Springer, New York, NY; 2012. pp. 181-189.

[22] Alahmadi AA, Hussain M, Aboalsamh H, Muhammad G, Bebis G. Splicing image forgery detection based on DCT and Local Binary Pattern. In: IEEE 2013 Global Conference on Signal and Information Processing; Austin, TX, USA; 2013. pp. 253-256.

[23] Muhammad G, Al-Hammadi MH, Hussain M, Bebis G. Image forgery detection using steerable pyramid transform and local binary pattern. Machine Vision and Applications 2014; 25 (4): 985-995. doi: 10.1007/s00138-013-0547-4

[24] Jindal N, Singh K. Image and video processing using discrete fractional transforms. Signal, Image and Video Processing 2014; 8 (8): 1543-1553. doi: 10.1007/s11760-012-0391-4

[25] Singh K, Saxena RG. Performance of discrete fractional Fourier transform classes in signal processing applications, PhD, Thapar Institute of Engineering and Technology, Patiala, Punjab, India, 2006.

[26] Rhee KH. Median filtering detection based on variations and residuals in image forensics. Turkish Journal of Electrical Engineering & Computer Science 2017; 25 (5): 3811-3826. doi:10.3906/elk-1606-410

[27] Chang CC, Lin CJ. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent systems and technology 2011; 2 (3): 1-27. doi: 10.1145/1961189.1961199

[28] Pun CM, Yuan XC, Bi XL. Image forgery detection using adaptive oversegmentation and feature point matching. IEEE Transactions on Information Forensics and Security 2015; 10 (8): 1705-1716. doi: 10.1109/TIFS.2015.2423261

[29] Han CY. Improved SLIC imagine segmentation algorithm based on K-means. Cluster Computing 2017; 20 (2): 1017-1023. doi: 10.1007/s10586-017-0792-9

[30] Jindal N, Singh K. Digital image forensics-gateway to authenticity: Crafted with observations, trends and forecasts. In: Singh AK, Mohan A (editors). Handbook of Multimedia Information Security: Techniques and Applications, Springer, Cham, 2019, pp. 681-701.

[31] Rublee E, Rabaud V, Konolige K, Bradski GR. ORB: An efficient alternative to SIFT or SURF. In: IEEE 2011 International Conference on Computer Vision; Barcelona, Spain; 2011. pp. 2564-2571.

[32] Sujatha P, Sudha KK. Performance analysis of different edge detection techniques for image segmentation. Indian Journal of Science and Technology 2015; 8 (14): 1-6. doi: 10.17485/ijst/2015/v8i14/72946

[33] Dong J, Wang W, Tan T. Casia image tampering detection evaluation database. In: IEEE 2013 China Summit and International Conference on Signal and Information Processing; Beijing, China; 2013. pp. 422-426.

[34] Hsu YF, Chang SF. Detecting image splicing using geometry invariants and camera characteristics consistency. In: IEEE 2006 International Conference on Multimedia and Expo; Toronto, Ont., Canada; 2006. pp. 549-552.

[35] Wen B, Zhu Y, Subramanian R, Ng TT, Shen X et al. COVERAGE-A novel database for copy-move forgery detection. In: IEEE 2016 International Conference on Image Processing; Phoenix, AZ, USA; 2016. pp. 161-165.

[36] Bozkurt I, Bozkurt MH, Ulutas G. A new video forgery detection approach based on forgery line. Turkish Journal of Electrical Engineering & Computer Sciences 2017; 25 (6): 4558-4574. doi: 10.3906/elk-1703-125

[37] Alahmadi A, Hussain M, Aboalsamh HA, Zuair M. PCAPooL: unsupervised feature learning for face recognition using PCA, LBP, and pyramid pooling. Pattern Analysis and Applications 2019; 23 (1) : 673–682. doi: 10.1007/s10044-019-00818-y

[38] Pham NT, Lee JW, Kwon GR, Park CS. Hybrid image-retrieval method for image-splicing validation. Symmetry 2019; 11 (83): 1-15. doi: 10.3390/sym11010083