Multimedia Tools and Applications (2023) 82:4517–4532
https://doi.org/10.1007/s11042-022-13610-8

Unveiling digital image forgeries using Markov based quaternions in frequency domain and fusion of machine learning algorithms

Savita Walia[1] · Krishan Kumar[1] · Munish Kumar[2]

Received: 20 March 2021 / Revised: 22 October 2021 / Accepted: 18 July 2022 /
Published online: 29 July 2022

Abstract

Nowadays, digital images have been manipulated for various fraudulent purposes from time to time. Therefore, proposing a detection algorithm for such manipulations has become a key interest for investigators. Digital image splicing and copy-move are among the most commonly performed forgeries and their detection is problematic. In spite of the number of accessible potential solutions for manipulation detection, the increasing number of forgery cases and their complexity necessitate the need for improved detection methods. The technique put forward in this article aims to deliver a reliable remedy for image manipulation detection. In the presented article, a novel procedure for discovering forgery in digital images is proposed with the use of quaternions that expands the distinction between probability distributions of authentic and forged images. Quaternions help in representing the image in its entirety as a vector field and discrete cosine transform is applied on the quaternion that is constructed from the color components of the image. After that, Markov features are originated from the transitional probabilities of the inter-block and intra-block correlation amongst the coefficients of the blocks along four directions. Finally, a feature vector of 648 dimensions is classified using a fusion of various classifiers. The performance of the approach is presented in terms of seven different performance metrics. The experiments are performed on three benchmark datasets viz. CoMoFoD dataset, CASIA version 1.0 and CASIA version 2.0 datasets. The proposed approach beats existing approaches over the CoMoFoD dataset and yields competitive outcomes for the CASIA version 2.0 dataset.

Keywords Image forensics · Decision fusion · Discrete transforms · Forgery detection · Fusion of classifiers · Majority voting · Markov process · Quaternions

✉ Munish Kumar
munishcse@gmail.com

Extended author information available on the last page of the article

1 Introduction

With the swift growth of the internet and technology, digital images are vulnerable to fraudulent manipulations. Digital images are the simplest and commonly used media to share and transfer information and they are even used in various fields as visual proof of things. Hence, manipulation detection in digital images has become the need of the hour [38]. Investigators have proposed several methods in the last decade that can be characterized into two major types: Active and Passive blind methods [37]. Active methods are employed when there is little knowledge about the image available beforehand [13]. Image watermarking and digital signatures are the techniques that come under active methods of forgery detection. On the other hand, passive methods of forgery detection work on intrinsic changes made in the image that is used for corroborating the genuineness of the images. Passive methods attracted much more attention in recent times as they are more practical as compared to active methods. In a practical scenario, only a single image is handed over to a forensic team with no aforementioned information. In such cases, passive blind methods are more appropriate. There are two types of forgery that commonly operate on digital images: copy-paste and splicing/ image composites. For uncovering copy-paste forgery, many algorithms are proposed by researchers that follow a simple procedure of feature extraction followed by a feature vector matching [1, 10, 22, 36]. On the other hand, image splicing introduces some intrinsic changes in the images that further instigate inconsistencies. The major goal of any forgery detection method is to determine such inconsistencies in the images.

In the last couple of years, various methods of detection of image composites, or splicing have also been proposed. Farid [11] has proposed a method to unveil artificially higher-order correlations in the image due to forgery operation by analyzing the interactions between wave patterns in the frequency domain. A 266-dimensional feature vector was created by using wavelet transform and Markov features and resulted in a success rate of 91.8% for the Columbia dataset developed in the Digital Video and Multimedia (DVMM) laboratory at Columbia University [31]. The method used statistical features based on one-dimensional and two-dimensional moments which improve a similar methodology used in steganalysis [30]. Hsu and Chang [17] have proposed a novel method based on the function of camera response and geometrical invariants in a semi-automatic routine. Image segmentation was incorporated to make it fully automatic [18]. Dong et al. [6] have suggested a method based on run-length to distinguish between original and fake images. This method was later surpassed by He et al. [15]. Chen et al. [4]have reached a detection rate of 82.32% for the Columbia DVMM splicing assessment dataset by comparing the weighted alignment of the components of the Fourier transform on the image with the summation of the components using Support Vector Machine (SVM) classifier. The Gray Level Co-occurrence Matrix (GLCM) of edge image was also used for colored images to detect splicing forgery [39]. The edges of the chrominance component of the image were exhibited as a Markov chain. The significant limitation of the method was that only a single-color channel has been exploited for further processing. The feature dimensionality is kept low and an accuracy of 95.6% was attained on CASIA v2.0. Further, it has been noticed that CASIA v2.0 has some drawbacks and the detection rate deteriorated to 78% with Markov features after rectification of the dataset [32].

It has been observed from literature that Markov features can unveil the dependencies among adjoining pixels whenever there is any small change owing to splicing operation [9, 14, 16, 40]. Markov features are extracted using transitional probabilities of the current and future states of the image blocks. The sturdy capability of this matrix of transition probabilities is to indicate the correlation between the pixels. Hence, the method based on Markov-based features on the color characteristics of an image is proposed. Quaternions in the discrete cosine transform are used in various fields [12, 29] due to their applications for processing color images. In the case of image tampering detection, the maximum of the procedures transforms the color image into a grayscale image before the actual processing. Such algorithms do not take the color characteristics of the image into account. However, these color characteristics are important to distinguish between forged and original images as splicing operation modifies the color characteristics of an image. To make use of the information from all the three channels of the image, the concept of quaternions can be used. Quaternions are helpful in removing the redundancy and can provide a quantitative analysis of the image. With the implementation of quaternions, the detection accuracy of the composite image forgery detection algorithms can be improved by exploiting the color perversion. Thus, an algorithm for uncovering forgery in digital images using quaternions in the frequency domain is proposed in this article with very convincing results. The following are the two major reasons for proposing an approach based on Block-QDCT (BQDCT):

1. Diverse regions in the image have distinctive content information and textural information. We propose to apply overlapping Block-QDCT on the input image, and it can maintain a strategic distance from the effects of inconformity produced by diverse regions of natural image amid the process of regular DCT.
2. There are continuity, consistency, smoothness, periodicity and regularity between the pixels of an unaltered image [21], and this relationship will be altered gradually when processed for around 12 to 18 pixels, so the procedure of Markov-based Block-QDCT can diminish the impact presented by distant pixels on the proposed statistical features.

Further, classification plays a major role in various digital image processing tasks. Various data-driven approaches like machine learning or deep learning have demonstrated remarkable results in most image processing or classification problems [3, 23, 28]. Different classifiers can be fused together such that the outcome of the fusion of these classifiers can be more accurate as compared to the single classifiers, comprising the best one [5, 24]. The elementary notion behind fusing the results of various classifiers is that while making a decision regarding classification, the final outcome should not be dependent on the results of a single classifier but instead all the classifiers contribute in the decision making. In this paper, majority voting has been used to process the outcomes of the classifiers. The contributions of the article are as follows:

3. It proposes an unprecedented forgery detection approach that uses Markov based transitional probabilities on quaternions in frequency domain, i.e. the discrete cosine transform as feature extraction from color images.
4. Markov-based features help in detecting the intrinsic changes that occur due to forgery operations.
5. The extracted features are classified using fusion of various classifiers to obtain the best results.
6. Evaluation of proposed approach with eight state-of- the-art manipulation detection approaches has been displayed over three benchmark datasets.
7. Seven different performance metrics are calculated for three benchmark datasets using the proposed approach and established that it either beat the existing methods or gives promising outcomes in terms of accuracy.

The remaining article is structured as follows. Section 2 explains the proposed methodology. Section 3 gives the details about the experimental outcomes obtained from the simulations. At last, concluding notes and findings of the experimental results are given in Section 4.

2 Proposed methodology

The block illustration of the proposed methodology is given in Fig. 1. Intra-block and inter-block differences are utilized to formulate the feature vector in vertical, horizontal, and diagonal directions on the quaternion discrete cosine transformation of an RGB image. Also, the correlations between and within the blocks are considered along with major diagonal and anti-diagonal quaternion discrete cosine transform (DCT) coefficients. The final feature vector is created by calculating the correlations along every possible direction on DCT coefficients of the image. Further, the feature vector is provided to the various combinations of machine learning classifiers to perform the distinction between authentic and fake image.

2.1 Constructing the quaternions

The notion of quaternions is very widely used in pure and applied mathematics. The quaternions are employed in three-dimensional computer graphics, texture analysis and computer vision applications. In color image processing, quaternions have proved to be very efficient as they consider all the three color channels and with the help of quaternions, a colored image can be considered as a vector field holistically. A quaternion is an extended complex quadruple having one real part and three imaginary numbers and is of the form as shown in (1).

$$a + bi + cj + dk \tag{1}$$

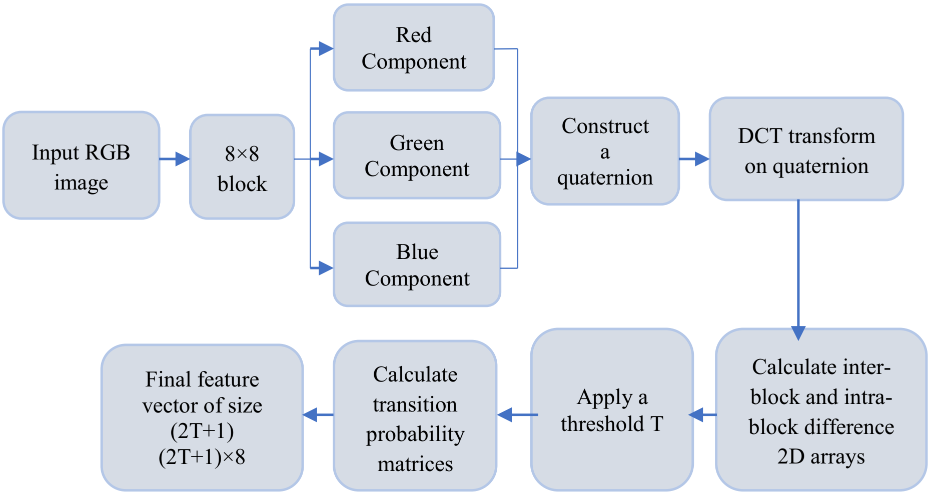
Input RGB image
8×8 block
Red Component
Green Component
Blue Component
Construct a quaternion
DCT transform on quaternion
Calculate inter-block and intra-block difference 2D arrays
Apply a threshold T
Calculate transition probability matrices
Final feature vector of size (2T+1)(2T+1)×8

Fig. 1 Block diagram of proposed methodology

a, *b*, *c* and *d* are real numbers where *a* is the non-imaginary part of the quaternion and *i*, *j* and *k* are the basic quaternion units which they satisfy the Hamilton rule [2]:

$$j = k, jk = i, ki = jji = -k, kj = -i, ik = -j \tag{2}$$

It also assures that $i^2 = j^2 = k^2 = ijk = -1$. Quaternions are non-commutative for multiplication and other basics of quaternions can be found in [2]. A quaternion can be created from two complex numbers with the help of Clay-Dickson theorem [20]. All the three quaternion units are orthogonal to each other. Let us assume $m, n \varepsilon C, m = a + bi, n = c + di, a, b, c, d \varepsilon R$, then

$$q = m + nj, q \varepsilon Q, j^2 = -1 \tag{3}$$

Hence,

$$q = a + bi + (c + di)j = a + bi + cj + dij = a + bi + cj + dk \tag{4}$$

The transformation given in (4) is used to create the quaternion from complex numbers. The quaternions are also called as hyper-complex numbers [20]. More details on how to construct a quaternion can be found in [20, 27]. Assume μ_1, μ_2 are the two axes of the unit quaternion and they are mutually perpendicular, after that q can be disintegrated into two different complex coordinates in the direction of μ_1 and μ_2.

$$q = m^{'} + n^{'}\mu_2, \mu_2^2 = -1 \tag{5}$$

where $m' = a' + b'\mu_1, n = c' + d'\mu_1, a', b', c', d' \varepsilon R$ then, $q = a' + b'\mu_1 + c'\mu_2 + d'\mu_3$. Here, $\mu_3 = \mu_1\mu_2$ and μ_3 is perpendicular to μ_1 and μ_2. For the problem being addressed in this article, the coordinates of image *(b, c, d)* will be transformed into the coordinates *(a', b', c', d')* under the three axes i.e.μ_1, μ_2 and μ_3.

2.2 Discrete cosine transform on quaternions

In literature, most of the techniques based on the discrete cosine transform separate the color channels of the image. For example, the image is changed to YCbCr color space, only Y component is chosen for detection procedure [19, 21]. This is not helpful in exploiting the correlation between all the color channels. On the other hand, QDCT can handle all the three channels simultaneously and colored images can be handled in an integrated manner. A RGB image can be denoted using quaternion matrix as shown in (6).

$$f_q(m,n) = f_r(m,n)i + f_g(m,n)j + f_b(m,n)k \tag{6}$$

Where $f_r(m, n), f_g(m, n)$ *and* $f_b(m, n)$ are red, green and blue color components of the image [19]. In the proposed approach, forward quaternion discrete cosine transform (FQDCT) has been used. FQDCT can be categorized into two types: left-handed and right-handed since quaternion multiplication follows non-commutative rule.

2.3 Markov features of each block from discrete cosine transform on quaternions

The steps to calculate the Markov features are provided in [16] and it has been observed that the Markov chain features in DWT and DCT domain have performed considerably well. In our method, the primary stage to extricate the features differs from [16]. The original color images are split into sliding blocks of size 8 × 8. The block obtained after segmentation is still a colored sub-image. The quaternion is formed from the R, G and B component of the sub-image. Further, DCT is applied to obtained quaternion matrix. The coefficients of the transformation are assembled in a matrix and the square root of real and the imaginary part is computed. Thus, the final matrix is obtained by arranging the blocks according to the block location. To compute Markov features from the QDCT coefficients, round the coefficients and the absolute values for further processing. Next, compute vertical, horizontal, major diagonal and minor diagonal distances (Q_v, Q_h, Q_d and Q_{-d} within blocks by applying (7) to (10).

$$Q_v\ (u,v) = Q\ (u,v) – Q\ (u,v+1) \tag{7}$$

$$Q_h\ (u,v) = Q\ (u,v) – Q\ (u+1,v) \tag{8}$$

$$Q_d\ (u,v) = Q\ (u,v) – Q\ (u+1,v+1) \tag{9}$$

$$Q_{-d}\ (u,v) = Q\ (u+1,v) – Q\ (u,v+1) \tag{10}$$

where *Q (u, v)* is the matrix containing the rounded off QDCT coefficients. Similarly compute the vertical, horizontal, major diagonal and anti-diagonal differences between the blocks, i.e. inter-block distances (R_v, R_h, R_d and R_{-d} using (11) to (14).

$$R_v\ (u,v) = Q\ (u,v) – Q\ (u,v+8) \tag{11}$$

$$R_h\ (u,v) = Q\ (u,v) – Q\ (u+8,v) \tag{12}$$

$$R_d\ (u,v) = Q\ (u,v) – Q\ (u+8,v+8) \tag{13}$$

$$R_{-d}\ (u,v) = Q\ (u+8,v) – Q\ (u,v+8) \tag{14}$$

Since, difference values calculated in (7) to (14) are obtained in the form of integers and contain a broad range. So, it is desirable to look for some methods like rounding and threshold. For this, a threshold T is given which is a positive integer. If the value of an entity after rounding in a different array obtained in (7) to (14) is more than T or less than -T, then the value will be substituted with T or -T correspondingly using (15).

$$A_{new} = \begin{cases} -T, & A_{old} < -T \\ T, & A_{old} > T \\ A_{old}, otherwise. \end{cases} \tag{15}$$

Finally, calculate the transitional probabilities of the above obtained inter-block and intra-block matrices using the equations given in (16) to (23).

$$PM1_h(i,j) = \frac{\sum_{u=1}^{m-2}\sum_{v=1}^{n}\delta(Q_h(u,v) = i, Q_h(u+1,v) = j)}{\sum_{u=1}^{m}\sum_{v=1}^{n}\delta(Q_h(u,v) = i)} \tag{16}$$

$$PM1_v(i,j) = \frac{\sum_{u=1}^{m-1}\sum_{v=1}^{n-1}\delta(Q_h(u,v) = i, Q_h(u,v+1) = j)}{\sum_{u=1}^{m-1}\sum_{v=1}^{n-1}\delta(Q_h(u,v) = i)} \tag{17}$$

$$PM2_h(i,j) = \frac{\sum_{u=1}^{m-1}\sum_{v=1}^{n-1}\delta(Q_v(u,v) = i, Q_v(u+1,v) = j)}{\sum_{u=1}^{m-1}\sum_{v=1}^{n-1}\delta(Q_v(u,v) = i)} \tag{18}$$

$$PM2_v(i,j) = \frac{\sum_{u=1}^{m-1}\sum_{v=1}^{n-1}\delta(Q_v(u,v) = i, Q_v(u,v+1) = j)}{\sum\limits_{u=1}^{m-1}\sum\limits_{v=1}^{n-1}\delta(Q_v(u,v) = i)} \tag{19}$$

$$PM3_d(i,j) = \frac{\sum_{u=1}^{m-16}\sum_{v=1}^{n-16}\delta(R_d(u,v) = i, R_d(u+8,v+8) = j)}{\sum_{u=1}^{m-16}\sum_{v=1}^{n-16}\delta(R_d(u,v) = i)} \tag{20}$$

$$PM3_{-d}(i,j) = \frac{\sum_{u=1}^{m-16}\sum_{v=1}^{n-16}\delta(R_{-d}(u+8,v) = i, R_{-d}(u,v+8) = j)}{\sum_{u=1}^{m-16}\sum_{v=1}^{n-16}\delta(R_{-d}(u,v) = i)} \tag{21}$$

$$PM3_h(i,j) = \frac{\sum_{u=1}^{m-8}\sum_{v=1}^{n-8}\delta(R_v(u,v) = i, R_v(u+8,v) = j)}{\sum_{u=1}^{m-8}\sum_{v=1}^{n-8}\delta(R_v(u,v) = i)} \tag{22}$$

$$P\mathrm{M3}_v(i,j) = \frac{\sum_{u=1}^{m-8}\sum_{v=1}^{n-8}\delta(R_h(u,v) = i, R_h(u,v+8) = j)}{\sum_{u=1}^{m-8}\sum_{v=1}^{n-8}\delta(R_h(u,v) = i)} \tag{23}$$

Where *i, j* ε {−T, −T + 1, …, 0,… T-1, T}, *m* and *n* signify the number of rows and columns respectively in the authentic image. δ(.) = 1, only if it satisfies all the arguments otherwise δ (.) = 0. The obtained final count of the feature vector is given by (2 T + 1) × (2 T + 1) × 8. In our experiments T is taken as 4. Hence, 648-dimensional feature vector is obtained. The experiments were also run with thresholds 2, 3, 4 and 5. At threshold 2 and 3, the accuracy is decreasing. On the other hand, when the threshold is taken greater than 4, there is no significant change in the accuracy. Moreover the computational expense increases. So, T = 4 is chosen for further experimental work.

2.4 Classification

The features are labeled as forged and authentic and are fed to classifiers fused in various combinations. Decision level fusion is performed on the outputs of the classifiers as shown in Fig. 2. For that, the majority voting algorithm is used in which the decisions from the classifiers are preserved directly. No probabilistic modeling of the outputs is performed. That is why combining the classifiers is such manner is simple and inexpensive [5]. In decision level fusion, each classifier is allowed to make their decision independently. If there are N classifiers for two-class classification problem, the ensemble output will be accurate if at-least $\lfloor N/2 + 1 \rfloor$ classifiers select the right class. Now, let's undertake that each classifier has a probability P_i of making a right decision. Then, the probability of ensemble combination of giving a right decision follows a binomial distribution. For experimental work, 80% data is considered for training and other 20% data is used for testing. The pseudo-code for the ensemble classification is shown in Algorithm 1.

Algorithm 1 Majority voting based Ensemble method.

```
**Representation :**
M: Number of the classifiers, N: Classifiers, D: Total number of test instances, +1:
Class label for forged images, -1: Class label for original images, $Y_D$: It represents
predicted output of the ensemble model
**Input:** M, N and D
**Procedure Enmseble():**
for i = 1 to D do
        for j = 1 to M do
                Final_prediction$_i$ = Majority(Prediction$_{ij}$)
        end for
end for
end procedure
**Output:** $Y_D$
```

3 Experimental outcomes

The experimentations have been implemented on Windows 10–64 bit personal computer with 16GB of RAM and Intel(R) Core-i7 processor. The code is written using Python programming language and to speed up the process of feature extraction, CUDA API is used. The details of the datasets and various experiments explicated to confirm the accuracy of the proposed method are given below:

3.1 Datasets used

Three standard datasets for image tampering detection have been used for assessment of the proposed approach which are available freely. The datasets considered in this work are Copy-move forgery detection (CoMoFoD) [34] dataset, CASIA version 1.0 [7] and CASIA version 2.0 [7] datasets. The two datasets viz. CASIA version 1.0 and CASIA version 2.0 are delivered by Chinese Academy of Sciences. Several manipulations are performed viz. rotation, translation, scaling, distortion and various combinations of such geometrical operations. Various

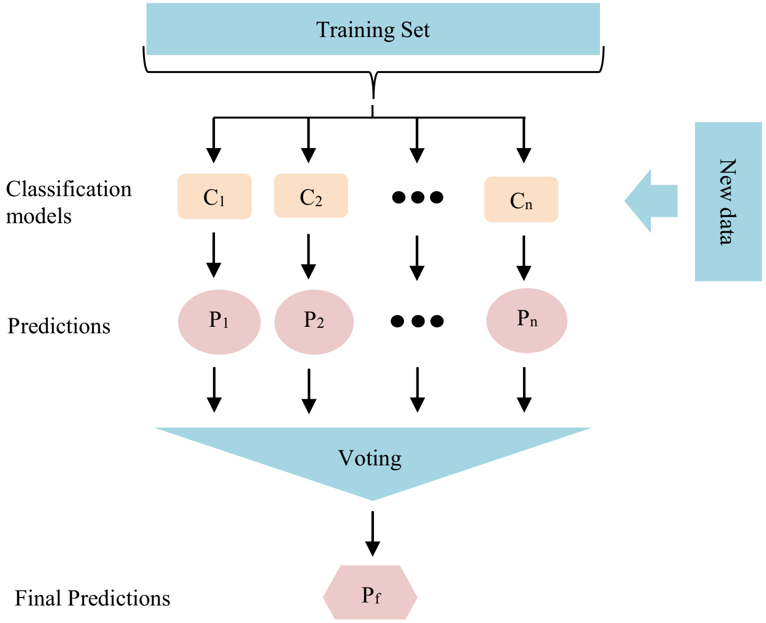

Fig. 2 Fusion of classifiers

post-processing methods, like blur, noise, color reduction, JPEG compression, etc. are also incorporated to the images of these datasets. The particulars of the datasets are mentioned in Table 1.

3.2 Performance metrics

The performance of the detection method is established by how appropriately the suggested method is capable of classifying the images into authentic and manipulated class. Different performance metrics are calculated for validating the proposed approach. All the metrics are described below:

Accuracy: It is the ratio of accurately identified images with total number of identified images. It is calculated using (24).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100 \tag{24}$$

where TP is True Positives, TN is True Negatives, FP is False Positives and FN is False Negatives.

Table 1 Details of datasets used for evaluation

Dataset	Number of original images	Number of modified images	Image resolution	Image format
CoMoFoD	200	10,200	512×512	JPEG, PNG
CASIA v1.0	800	925	384×256	JPEG
CASIA v2.0	7491	5123	240×160 to 900×600	JPEG, BMP, TIFF

Precision: Precision is given as the number of image feature vectors correctly categorized as forged upon total number of image feature vectors labelled as forged and is given by (25).

$$Precision = \frac{TP}{TP + FP} \tag{25}$$

Root Mean Squared Error (RMSE): For finding out the classifier ensemble with low error-rate, RMSE metric has been calculated as shown in (26). For binary classification problems, RMSE helps to shed light on how distant the classification model is from giving the correct answer.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{\left(\widehat{y}_i - y_i\right)^2}{n}} \tag{26}$$

where $\widehat{y}_i$ is the predicted outcome, y_i is the actual outcome and n is the total number of images tested using a particular classifier.

True Positive Rate (TPR): TPR reflects the classification model's ability to detect the image features belonging to the forged class. Hence, higher the TPR, better is the classification model.

$$TPR = \frac{TP}{TP + FN} \tag{27}$$

False Rejection Rate (FRR): FRR determines the frequency at which the classification model makes a mistake by classifying a forged image as authentic. It is calculated using (28).

$$FRR = \frac{FN}{FN + TP} \tag{28}$$

F1-score: It is calculated using (29) and the highest possible value of F1-score is 1. The highest value indicates the perfect classification model. It indicates the balance between recall and precision by taking into account both the false negatives and false positives in contrast to accuracy.

$$F1{-}score = \frac{2TP}{2TP + FP + FN} \tag{29}$$

Area Under the Curve (AUC): AUC is the measurement metric for indicating performance of classification models at numerous threshold settings. Receiver Operating Characteristics (ROC) curve is obtained by plotting TPR and FPR at diverse points. The area under ROC curve hence gives a value that determines how correctly the model classifies forged images as forged and original images as original.

3.3 Classification

Various experimentations are performed on three benchmark datasets. Firstly, individual classifier model viz. k-Nearest Neighbor, Linear Support Vector Machine (SVM), Decision Tree, Multi-Layer Perceptron (MLP) and Random Forest have been constructed and tested on features from the above mentioned datasets. The results can be seen in Table 2. The values of the metrics obtained from this experimentation are used to highlight the improvements achieved by the proposed method with ensemble classifiers. For k-NN, 5 nearest neighbor

Table 2 Performance of individual classifier on benchmark datasets

Dataset	Classifier	Accuracy	Precision	F1-Score	TPR	FRR	AUC
CASIA v1	k-NN	63.74	62.63	0.616	66.59	0.393	0.640
	Linear SVM	80.56	77.43	0.786	81.21	0.203	0.760
	Decision Tree	72.34	68.88	0.698	73.59	0.292	0.730
	MLP	82.04	85.29	0.817	85.88	0.216	0.770
	Random Forest	74.54	70.43	0.715	71.85	0.320	0.730
CASIA v2	k-NN	89.24	91.11	0.897	90.24	0.116	0.893
	Linear SVM	90.47	91.16	0.901	91.77	0.109	0.901
	Decision Tree	82.46	82.28	0.824	82.41	0.175	0.820
	MLP	92.05	90.36	0.920	90.42	0.063	0.928
	Random Forest	90.70	89.69	0.907	89.69	0.083	0.901
CoMoFoD	k-NN	93.12	91.41	0.935	92.71	0.065	0.930
	Linear SVM	94.97	92.06	0.921	91.78	0.017	0.990
	Decision Tree	90.84	91.52	0.913	90.55	0.089	0.890
	MLP	92.51	92.21	0.930	91.12	0.063	0.908
	Random Forest	93.84	92.20	0.939	91.98	0.043	0.938

are considered and the distance metric used is Euclidean distance. For Linear SVM, linear kernel function is used and the kernel scale is set to 1. For construction of Decision Tree model, maximum number of splits is taken as 100 and Gini's diversity index is used as the split criterion. Multi-Layer Perceptron is constructed using 20 hidden neurons with sigmoidal function as the activation function in hidden layer and softmax function at output layer. For Random Forest classifier, number of trees is taken as 50. Further to test the features extracted from these datasets, four ensemble combinations of different classifiers mentioned above are structured whose details are given in Table 3. Ensemble classifiers are beneficial as they result in lower error and less over-fitting. Various performance metrics are used for evaluation which is discussed in previous section. The experimental results on CASIA v2 dataset, CASIA v1 dataset and CoMoFoD dataset are given in Tables 4, 5 and 6 respectively. Accuracy, RMSE and AUC for different datasets under consideration are graphically shown in Figs. 3, 4 and 5, respectively. It can be observed from the results that the ensemble classifiers obtained better results as compared to individual classifier in case of all three benchmark datasets.

Table 3 Different fusion combinations used for experiments

Ensemble	Algorithm 1	Algorithm 2	Algorithm 3
Ensemble 1	Linear SVM	Decision Tree	Random Forest
Ensemble 2	Linear SVM	KNN	Random Forest
Ensemble 3	Linear SVM	MLP	Random Forest
Ensemble 4	k-NN	Linear SVM	Decision Tree

Table 4 Results of decision fusion on CASIA V2 Dataset

CASIA Version 2	Accuracy	Precision	RMSE	TPR	FRR	AUC
Ensemble 1	93.80	93.9	24.89	93.8	0.60	0.938
Ensemble 2	93.57	93.6	25.34	93.6	0.60	0.936
Ensemble 3	93.57	93.6	25.34	93.6	0.60	0.936
Ensemble 4	76.74	76.8	48.22	76.7	0.23	0.765

Table 5 Results of decision fusion on CASIA V1 dataset

CASIA Version 1	Accuracy	Precision	RMSE	TPR	FRR	AUC
Ensemble 1	81.97	82.0	42.45	82.0	0.18	0.817
Ensemble 2	77.03	77.1	47.92	77.0	0.23	0.768
Ensemble 3	82.55	82.5	41.76	82.6	0.18	0.823
Ensemble 4	76.74	76.3	48.22	76.7	0.23	0.765

Table 6 Results of decision fusion on CoMoFoD dataset

Copy Move Data	Accuracy	Precision	RMSE	TPR	FRR	AUC
Ensemble 1	98.56	98.6	11.98	98.6	0.10	0.986
Ensemble 2	99.28	99.3	8.47	99.3	0.07	0.993
Ensemble 3	99.28	99.3	8.47	99.3	0.07	0.993
Ensemble 4	99.04	99.1	9.78	99.0	0.09	0.991

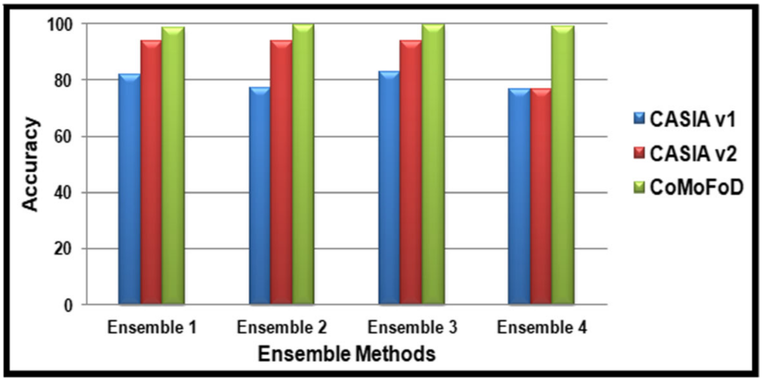

Fig. 3 Accuracy using different ensemble method

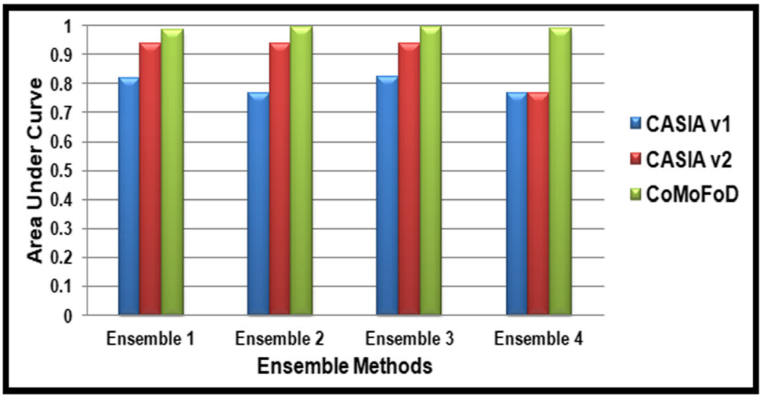

Fig. 4 RMSE using different ensemble method

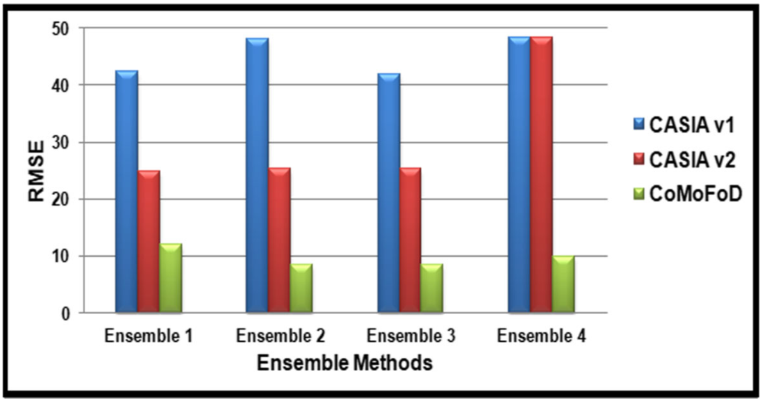

Fig. 5 Area Under the Curve using different ensemble method

3.4 Evaluation in comparison with state-of-art approaches

The results obtained from the experiments are compared and analyzed with state-of-the-art methods on various datasets. The method proposed in this paper outperformed [16, 19] in terms of accuracy on CASIA v1.0 dataset as shown in Table 7. The method in [16] processed color images, whereas [19] used the chrominance component of YCbCr model and our method uses all the three color components of an RGB image. The comparison of previous algorithms on CASIA v2.0 is shown in Table 8. The results obtained CoMoFoD dataset are also compared with state-of-art methods as shown in Table 9. In [8], the statistical properties of

Table 7 Comparison of accuracy on CASIA version 1.0 dataset

Method	Dimensionality	Accuracy
He et al. [16]	100	89.76%
Hussain et al. [19]	–	80.71%
Proposed method	648	82.55%

Table 8 Comparison of accuracy on CASIA version 2.0 dataset

Method	Dimensionality	Accuracy
Shi et al. [31]	266	84.86%
Li et al. [26]	972	92.38%
He et al. [16]	100	93.42%
Proposed method	648	93.80%

Table 9 Comparison of accuracy on CoMoFoD

Method	Dimensionality	Accuracy
Du et al. [8]	–	98.0%
Lee et al. [25]	12	98.8%
Ustubioglu et al. [35]	–	94.0%
Thirunavukkarasu [33]	–	99.0%
Proposed method	648	99.28%

AC coefficients of JPEG compressed image are used with a standard deviation of non-zero DCT coefficients to give an accuracy of 98.0%. In [25], histogram of oriented Gabor magnitude is utilized to extract features, and then the compact features are served to SVM for classification.

4 Conclusion

In most of the detection algorithms proposed by researchers in recent times, the colored input image is changed into a grayscale image prior to any pre-processing. Most of the methods disregard the color statistics of the image. In the suggested method, a digitally colored image can be processed in terms of a quaternion by taking all the color channels as imaginary components of the quaternion. The detection rate of the image forgery detection method is upgraded by using the property of quaternions to eliminate the redundancy and to represent the digital image in its entirety as a vector. An algorithm based on transition probabilities on quaternions of the image DCT coefficients is put forward in this article. The crux of the article is the passive detection of color images by expanding the attributes of transition probability from the frequency domain to unveil the dependencies among adjoining pixels wherever there is even a tiny change in pixels of the image due to the malicious tampering. The feature-set obtained is fed to an ensemble of classifiers and the majority voting is used to get the final output of classification. The performance of the approach is presented in terms of seven different performance metrics. The experiments are performed on three benchmark datasets viz. CoMoFoD dataset, CASIA version 1.0 and CASIA version 2.0 datasets. The proposed approach beats existing approaches over the CoMoFoD dataset and yields competitive outcomes for CASIA version 2.0 dataset.

In future, the Markov-based features will be tested along-with deep features extracted from different pre-trained architectures in order to improve the detection accuracy for forged images. By using the fusion concept, different kinds of characteristics of the images can be included in the analysis which further helps in increasing the efficiency of an approach.

Acknowledgements This work was supported by the Technical Education Quality Improvement Project III (TEQIP III) of the Ministry of Human Resource Development (MHRD), Government of India, assisted by the World Bank (Sanctioned to UIET, Panjab University, Chandigarh, India) under Grant P154523.

Declarations

Conflict of interest All the authors declared that they have no conflict of interest in this work.

References

1. Ali M, Deriche M (2015) A bibliography of pixel-based blind image forgery detection techniques. Signal Processing : Image Communication 39:46–74. https://doi.org/10.1016/j.image.2015.08.008
2. Barbic J (2011) Quaternions and Rotations. University of Southern California CSCI 520:1–12
3. Chaudhary P, Gupta DK, Singh S (2021) Outcome prediction of patients for different stages of Sepsis using machine learning models. In: Hura G, Singh A, Siong Hoe L (eds) Advances in communication and computational technology, vol 668. Springer, Singapore, pp 1085–1098

4. W. Chen, Y. Q. Shi, and W. Su, "Image splicing detection using 2D phase congruency and statistical moments of characteristic function," in Proceedings of SPIE - The International Society for Optical Engineering, 2007, vol. 6505. https://doi.org/10.1117/12.704321.
5. Chitroub S (2010) Classifier combination and score level fusion: concepts and practical aspects. Int J Image Data Fusion 1(2):113–135. https://doi.org/10.1080/19479830903561944
6. Dong J, Wang W Tan T, Shi Y. (2009) "Run-length and edge statistics based approach for image splicing detection," in International Workshop on Digital Watermarking (IWDW), Lect Notes Comput Sci 5450, pp. 76–87
7. Dong J, Wang W, Tan T (2013) "CASIA Image Tempering Detection Evaluation Database (CAISA TIDE)", in 2013 IEEE China Summit and International Conference on Signal and Information Processing, pp. 422–426. https://doi.org/10.1109/ChinaSIP.2013.6625374
8. Dua S, Singh J, Parthasarathy H (2020) Image forgery detection based on statistical features of block DCT coefficients. Procedia Computer Science 171(2019):369–378. https://doi.org/10.1016/j.procs.2020.04.038
9. El-alfy EM, Qureshi MA (2015) Combining spatial and DCT based Markov features for enhanced blind detection of image splicing. Pattern Anal Applic 18:713–723. https://doi.org/10.1007/s10044-014-0396-4
10. Elaskily MA, Aslan HK, Elshakankiry OA, Faragallah OS (2017) "Comparative Study of Copy-Move Forgery Detection Techniques," in Intl Conf on Advanced Control Circuits Systems (ACCS) Systems & 2017 Intl Conf on New Paradigms in Electronics & Information Technology (PEIT), pp. 193–203
11. Farid H (2009) Image forgery detection. IEEE Signal Process Mag 26(2):16–25. https://doi.org/10.1109/MSP.2008.931079
12. Feng W, Hu B (2008) Quaternion Discrete Cosine Transform and its Application in Color Template Matching. Congress Image Signal Process:252–256. https://doi.org/10.1109/CISP.2008.61
13. Haouzia A, Noumeir R (2008) Methods for image authentication : a survey. Multimed Tools Appl 39(1):1–46. https://doi.org/10.1007/s11042-007-0154-3
14. Hashmir MF, Keskar AG (2015) Image forgery detection and classification using HMM and SVM classifier. International Journal of Security and its Applications 9(4):125–140
15. He Z, Sun W, Lu W, Lu H (2011) Digital image splicing detection based on approximate run length. Pattern Recogn Lett 32(12):1591–1597. https://doi.org/10.1016/j.patrec.2011.05.013
16. He Z, Lu W, Sun W, Huang J (2012) Digital image splicing detection based on Markov features in DCT and DWT domain. Pattern Recogn 45:4292–4299
17. Hsu Y-F, Chang S-F (2006) "Detecting Image Splicing Using Geometry Invariants And Camera Characteristics Consistency," in International Conference on Multimedia and Expo (ICME), pp. 549–552
18. Hsu Y-F, Chang S-F (2007) "Image Splicing Detection using Camera Response Function and Automatic Segmentation," in 2007 IEEE International Conference on Multimedia and Expo, pp. 28–31
19. Hussain M, Saleh SQ, Aboalsamh H, Muhammad G, Bebis G (2014) "Comparison between WLD and LBP Descriptors for Non-intrusive Image Forgery Detection," in Proc. of IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA), pp. 197–204
20. Kantor I, Solodovnikov A, Shenitzer A (1989) Hypercomplex numbers : an elementary introduction to algebras. Berlin: Springer: Springer-Verlag New York
21. Kaur M, Walia S (2016) Forgery detection using noise estimation and HOG feature extraction. Int J Multimed Ubiquitous Eng 11(4):37–48. https://doi.org/10.14257/ijmue.2016.11.4.05
22. Kaur A, Walia S, Kumar K (2018) "Comparative analysis of different keypoint based copy-move forgery detection methods," in 2018 Eleventh International Conference on Contemporary Computing (IC3), pp. 172–176
23. Kumar M, Gupta DK, Singh S (2021) Extreme event forecasting using machine learning models. In: Hura G, Singh A, Siong Hoe L (eds) Advances in communication and computational technology, vol 668. Springer, Singapore, pp 1503–1514. https://doi.org/10.1007/978-981-15-5341-7_115
24. Kuncheva LI, Whitaker CJ, Shipp CA, Duin RPW (2003) Limits on the majority vote accuracy in classifier fusion. Pattern Anal Applic 6(1):22–31. https://doi.org/10.1007/s10044-002-0173-7
25. Lee J-C (2015) Copy-move image forgery detection based on Gabor magnitude. J Vis Commun Image Represent 31:320–334. https://doi.org/10.1016/j.jvcir.2015.07.007
26. Li C, Ma Q, Xiao L, Li M, Zhang A (2017) Image splicing detection based on Markov features in QDCT domain. Neurocomputing 228:29–36. https://doi.org/10.1016/j.neucom.2016.04.068
27. Moxey CE, Sangwine SJ, Member S, Ell TA (2003) Hypercomplex correlation techniques for vector images. IEEE Trans Signal Process 51(7):1941–1953
28. Saxena A, Gupta DK, Singh S (2021) An animal detection and collision avoidance system using deep learning. In: Hura G, Singh A, Siong Hoe L (eds) Advances in communication and computational technology, vol 668. Springer, Singapore, pp 1069–1084. https://doi.org/10.1007/978-981-15-5341-7_81
29. Schauerte B, Stiefelhagen R (2012) "Quaternion-based Spectral Saliency Detection for Eye Fixation Prediction Related Work," in 12th European conference on Computer Vision, pp. 116–129

30. Y. Q. Shi, and D. Zou, and, W. Chen, and C. Chen, “Image steganalysis based on moments of characteristic functions using wavelet decomposition, prediction-error image, and neural network,” in IEEE International Conference on Multimedia and Expo, 2005, pp. 1–4. https://doi.org/10.1109/ICME.2005.1521412.
31. Shi YQ, Chen C, Chen W (2007) “A Natural Image Model Approach to Splicing Detection,” in 9th workshop on Multimedia & security, pp. 51–62
32. Sutthiwan P, Shi YQ, Zhao H, Ng TT, Su W (2011) Markovian Rake Transform for Digital Image Tampering Detection. Trans Data Hiding Multimed Sec VI Lect Notes Comput Sci 6730:1–17
33. Thirunavukkarasu V, Kumar JS, Chae GS, Kishorkumar J (2017) Non-intrusive forensic detection method using DSWT with reduced feature set for copy-move image. Wirel Pers Commun 98:1–19. https://doi.org/10.1007/s11277-016-3941-1
34. Tralic D, Zupancic I, Grgic I, Grgic M (2013) “CoMoFoD - New Database for Copy-Move Forgery Detection,” in 55th International Symposium ELMAR, pp. 25–27
35. Ustubioglu B et al (2016) Improved copy-move forgery detection based on the CLDs and colour moments. Imaging Sci J 64(4). https://doi.org/10.1080/13682199.2016.1162922
36. Walia S, Kumar K (2017) Pragmatical investigation of frequency-domain and spatial-structure based image forgery detection methods. Intern J Comput Intell IoT 1(2):240–245
37. Walia S, Kumar K (2018) An Eagle-Eye View of Recent Digital Image Forgery Detection Methods. Smart Innov Trends Next Gener Comput Technol 828:469–487. https://doi.org/10.1007/978-981-10-8660-1
38. Walia S, Kumar K (2019) Digital image forgery detection: a systematic scrutiny. Australian Journal of Forensic Sciences 51(5):488–526. https://doi.org/10.1080/00450618.2018.1424241
39. W. Wang, J. Dong, and T. Tan, “Image tampering detection based on stationary distribution of Markov chain,” in IEEE International Conference on Image Processing, 2010, pp. 2101–2104. https://doi.org/10.1109/ICIP.2010.5652660.
40. Zhao X, Wang S, Li S, Li J (2015) Passive image splicing detection by a 2-D noncausal Markov model. IEEE Transactions on Circuits and Systems for Video Technology 25(2):185–199

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Savita Walia[1] • **Krishan Kumar**[1] • **Munish Kumar**[2]

Savita Walia
s.walia24@pu.ac.in

Krishan Kumar
k.salujauiet@gmail.com

[1] University Institute of Engineering and Technology, Panjab University, Chandigarh 160014, India

[2] Department of Computational Sciences, Maharaja Ranjit Singh Punjab Technical University, Bathinda 151001, India