Dual System for Copy-move Forgery Detection using Block-based LBP-HF and FWHT Features

Badal Soni, *Member, IAENG,* Pradip K. Das and Dalton Meitei Thounaojam, *Member, IAENG*

***Abstract*—Copy-move forgery is a popular image tampering technique. In this paper, we propose two efficient block-based systems for detection of copy-move forgeries present in images. The first system is based on the extraction of Local Binary Pattern Histogram Fourier features from each overlapping block and forgery decision based on the matching of these block features using Euclidean similarity measure. The second proposed system is based on the extraction of Fast Walsh Hadamard Transform features from each overlapping block and forgery decision based on the matching of these block features using shift vectors analysis. Both systems are tested using tampered images of the *CoMoFoD* dataset. Experimental results show that both systems are not only able to accurately detect tampered regions but also are invariant to various post-processing operations such as: blur movement, contrast adjustment, brightness and color reduction. Proposed system-I is computationally more efficient than system-II. However, system-II is more robust to blurring post-processing operation.**

***Index Terms*—Copy-move, LBP, LBP-HF, FWHT, Block-matching.**

I. Introduction

Now a days, due to the availability of low cost or open source image handling software, manipulation in images becomes an easy task and on the other end determining whether an image has been manipulated or not becomes a grand challenge. Manipulating original images using image tampering tools or software is called digital image forgery. Copy-move forgery is one among many image manipulation techniques to manipulate an image. In copy-move forgery operation, a portion of the image is copied and pasted into one or more locations in the same image with the objective to hide or duplicate some objects or sub-portions of the image. In the copy-move operation, the copied region is taken from the same image and as a result, color palette, noise components, dynamic range and other properties will be compatible with the rest of the image. Hence, it is a challenging task to detect copy-move forgery in images. Generally, the copy-move forgery detection procedure is divided into the following steps:

- **Preprocessing**: Most of the methods operate on the grey scale images. Therefore, input tampered images need to be first converted from RGB to grey scale images.
- **Features extraction**: Features extraction is an important step in copy-move forgery detection. There are a lot of approaches for feature extraction. Selection of feature extraction approach, depending on the operating way of the particular technique. Block and key-points based techniques are mostly used in copy-move forgery detection. Block-based techniques, start by dividing the forged input image into overlapping or non-overlapping blocks. Blocks can be rectangular or circular. For a given image of size $M \times N$ pixels and size of block $B \times B$, overlapping rectangular blocks division is performed by sliding the block over the whole image, one pixel at a time, from left to right and top to bottom. Using this process, the total number of overlapping blocks for the given image is $\{(M-B+1)\times(N-B+1)\}$. The block-based methods can be divided into five categories. They are moment-based [1], [2], dimensionality reduction-based [3], frequency-based [4], intensity and texture-based [5]. In the keypoints-based approach, local features are extracted such as corners, blobs and edges from the tampered image. Each feature is represented as a set of descriptors. The descriptor increases the reliability of the features. Descriptors are matched to find the forged regions in the image. SIFT [6] and SURF [7] local features are widely used in the key-points based copy-move forgery detection techniques.
- **Matching**: Both block and keypoints based techniques use appropriate matching procedures for forgery decision. Similar feature vectors are matched using high similarity as an indicator for the presence of duplicated regions.
- **Post processing**: The goal of this step is to preserve and highlight the matched regions of the image that exhibited common properties.

In this paper, we propose and implement two different systems for copy-move tampering detection in digital images which are based on the Local Binary Pattern Histogram Fourier features (LBP-HF) [8] and Fast Walsh Hadamard Transform (FWHT) [9]. In the proposed system-I, the tampered image is divided into fixed size overlapping blocks. Thereafter, rotation invariant LBP-HF features are extracted from each overlapping block. The feature vectors are compared and duplicate regions of the image are located by covering the corresponding blocks of the tampered image as a result of high similarity. In system-II, Fast Walsh Hadamard Transform features are explored for block matching. Matching is performed among extracted LBP-HF and FWHT block features for system-I and system-II respectively.

Since the matching is performed among extracted transformed features in these systems, this reduces the matching time and ultimately decreases the overall computation time

Manuscript received July 18, 2017; revised October 07, 2017 and November 20, 2017, accepted December 24, 2017 .

Badal Soni is an assistant professor in the Department of Computer Science and Engineering, National Institute of Technology, Silchar, India, e-mail: soni.badal88@gmail.com, http://cs.nits.ac.in/badal/.

Dalton Meitei Thounaojam is an assistant professor in the Department of Computer Science and Engineering, National Institute of Technology, Silchar, India, e-mail: dalton.meitei@gmail.com, http://cs.nits.ac.in/dalton/.

Pradip. K. Das is a professor in the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, India, e-mail: pkdas@iitg.ernet.in, http://www.iitg.ernet.in/pkdas/.

of the proposed systems.

The paper is organized in the following manner: Section II address related work. Local Binary Pattern Histogram Fourier feature is described in Section III. Proposed system-I is detailed in Section IV. Experimental results and discussion of proposed system-I is given in Section V. Proposed system-II is detailed in Section VI. Experimental results and discussion of proposed system-II is given in Section VII. Finally, conclusions and future scope are detailed in Section VIII.

II. Related work

In recent years, numerous papers were published in copy-move forgery detection (CMFD) domain. In this section, some of the related CMFD techniques with their pros and cons are covered. Copy-move forgery detection procedure is different from digital content authentication system viz as watermarking and digital signatures. In digital watermarking a signal or image as a watermark is embedded into digital images for their copyright protection. Watermark embedding and extraction procedures are given in [10], [11]. Rotation invariant uniform LBP features based method is given in [12]. In this method, extracted feature vectors are compared for forgery decision. It is observed that this method is robust against compression, noise, blurring and rotation. However, it failed to detect forgery in case of random region rotation. In [13] Multiresolution Local Binary Patterns (MLBP) features are used with two, three and four types of LBP operators. Lexicographical sorting and k-d tree are used in block matching to speed-up the algorithm. However, the computation time of this method is more in case of high-resolution images. Later, paper [14] combined the Hessian points and Center Symmetric LBP (CSLBP). In this method, the combination of the results of Hessian points and CSLBP make the features invariant to translation, scaling and illumination. However, this method is not robust against blur and rotation attacks. Combination of Steerable Pyramid Transform (SPT) and LBP is proposed in [15]. In this paper, SPT is applied into the chrominance component of the image and LBP features are extracted from each SPT sub-band. It is observed that two features selection methods, namely, feature discriminant ratio and LOGO are used to reduce the dataset dimension. However, localization of forgery is not performed in this method. Extension of [15] is given in [16]. In this method, features are extracted from each SPT sub-band by applying LBP. The LBP histograms of all the sub-bands are added together to form a feature vector. Feature vectors are then supplied to an SVM classifier. SVM uses an RBF kernel to classify an image as original or forged. This method out-performed the one reported in [15]. However, forgery localization in the image is still not done. LBP is combined with DCT in [17] to detect copy-move and splicing forgeries. In this work, the chroma component of the image is divided into overlapping blocks. LBP operator is applied to each block. LBP codes of each block was transformed into frequency domain using DCT. Standard deviations of DCT coefficients are calculated and arranged as a feature vector. These vectors are finally given to SVM for the decision of forgery. This method is robust and outperformed in small sized images. However, localization of forgery is not done. A generalized 2NN procedure for SIFT descriptors matching is proposed by [18] for CMFD. This method is very accurate with a TPR of 100%. However, improvement is needed in the detection phase for copied image patch with highly uniform texture where salient keypoints are not recovered by SIFT-based techniques. [19] gives a method based on PCA-SIFT along with k-nearest neighbors. In this paper, the dimension of extracted SIFT features are reduced by PCA and forgery is decided by selection of k-nearest neighbors. This method possessed better detection accuracy and reduced time complexity compared to SIFT. However, this method has scope for improvements in proper identification of tampered regions. To detect proper forgery in non-flat as well as flat regions, SIFT and Zernike moments are combined in [20]. SIFT is invariant to rotation and scaling but not suitable for forgery detection in flat regions. Zernike moments can detect forgery in flat regions, but it is sensitive to scaling. Therefore, in this paper, the author used both SIFT feature and Zernike moments for detecting forgery in the complete image. Paper [21], proposed a segmentation based technique using rotation invariant DAISY descriptors. Paper [22] proposed a fast dense-field technique by using PatchMatch algorithm to compute efficiently a high-quality approximate nearest neighbor field for the whole image. It is robust against geometric transformations. Image authentication based on HMM and SVM classifiers is presented in [23].

A recent detailed review of Copy-move forgery detection techniques is given in [24]. This paper critically discussed the different CMFD techniques along with their pros and cons and suggested future directions. In addition to that this paper reported difference datasets, performance measure and comparative results of some of the key CMFD techniques. A method for copy-move forgery detection based on DWT-FWHT is presented in [25]. For reducing the size of the image, Discrete Wavelet Transform (DWT) is applied to the input image and after that, the image is divided into fixed size overlapping blocks. Then, Fast Walsh-Hadamard Transform (FWHT) is applied in each block for extraction of features. For efficient matching, multi-hop jump (MHJ) algorithm is used to jump over some of the unnecessary testing blocks. Block matching in the spatial domain is given by [26]. Instead of exhaustive search, a two-step search algorithm is given for matching the 8×8 size block with other blocks. It can be observed that the computational complexity of this method is less than other conventional methods without using frequency domain features. However, this method is not suitable in case of different geometric transformation attacks. Statistical features based approach is given by [27]. In this approach, features are extracted from each block by applying the histogram of orientated gradients. Each block feature is sorted lexicographically and matching is performed for each pair of sorted blocks.

III. Local Binary Pattern Histogram Fourier features

The Local Binary Pattern (LBP) is a dominant feature for texture description [28]. LBP calculation of an image block is based on comparing each pixel with its neighborhood. A pixel is selected as the center and matched with its neighbors. If the intensity of the center pixel is greater than or equal to its neighboring pixel, then assign it with 1; otherwise, assign it with 0. After comparison with all the neighboring pixels,

a binary number for each pixel is generated. For instance, 8 surrounding pixels, this process will end up with 2^8 possible combinations, which are called Local Binary Patterns.

The LBP operator can also be extended to use neighborhoods of different sizes shown in Figure 1.

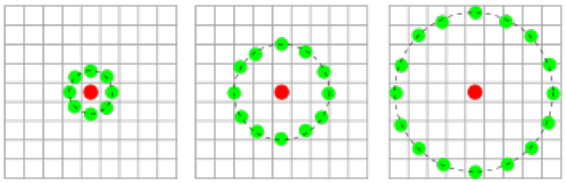

Fig. 1. Three neighborhood scenarios to calculate a local binary pattern

For this, a circular neighborhood, represented by (P_s, R_n), is defined. Here P_s represents the number of sampling points and R_n is the neighborhood radius.

LBP code for the center pixel (x, y) of image $f(x, y)$ is calculated as:

$$LBP_{P_s,R_n}(x,y) = \sum_{p_s=0}^{P_s-1} T_f(f(x,y) - f(x_p, y_p))^p \quad (1)$$

Where $T_f(z)$ is the thresholding function.

$$T_f(z) = \begin{cases} 1 & if\ z \geqslant 0 \\ 0 & if\ z < 0 \end{cases} \quad (2)$$

Uniform local binary pattern (ULBP) is an extension of LBP. When the bit pattern is circular, ULBP consists of at most two bitwise transitions from 0 to 1 or vice versa. In LBP histogram computation, uniform patterns are used. As a consequence of this, the resultant histogram consist of a distinct bin for every uniform pattern and all other non-uniform patterns are allocated to a single bin. The rotation invariant LBP operator is acquired by rotating each bit pattern circularly to the minimum value. Discrete Fourier transform (DFT) is used to construct Local Binary Pattern Histogram Fourier features (LBP-HF), which is a rotation invariant features descriptor. Calculation of LBP-HF descriptors are based on uniform local binary pattern histograms [29]. Consider uniform LBP pattern $U_{P_s}(n, r)$ and $F(n, .)$ to be the DFT of the n^{th} row histogram $h_I U_{P_s}(n, r)$ as given in following Equation:

$$F(n,u) = \sum_{r=0}^{P_s-1} h_I U_{P_s}(n,r) \exp^{\frac{-i2\pi ur}{P_s}} \quad (3)$$

LBP-HF features are constructed by calculating the Fourier magnitude spectrum as given in Equation:

$$|F(n,u)| = \sqrt{F(n,u)\overline{F(n,u)}} \quad (4)$$

IV. Proposed System-I

The proposed system, utilizes the Local Binary Pattern Histogram Fourier features (LBP-HF), for forgery detection in tampered images. The pseudo-code for the proposed system- I is given in Algorithm 1 and processing steps for the proposed system is given as follows:

A. Preprocessing

In this step, if the input forged image is an RGB image, then convert it into grey scale image I using following Equation:

$$I = 0.2989R + 0.587G + 0.114B \quad (5)$$

where R, G and B are the red, green and blue component of the input image respectively.

B. Overlapping Blocks Division

In this system, initially, the tampered image is divided into overlapping blocks. The block size used in this algorithm is 8×8. Overlapping blocks are created by sliding the block over the original image one pixel at a time across the rows and columns. The number of overlapping blocks obtained for an image of $M \times N$ dimensions is $(M-B+1) \times (N-B+1)$ where the block size is $B \times B$. All the overlapping blocks are stored in a matrix for further processing.

C. Lexicographical Sorting

In order to reduce the matching time, lexicographical sorting is performed on obtained overlapping blocks so that similar block vectors will be adjacent to each other. For this, each overlapping block is sorted as an array and the set of all arrays are stored in a matrix Mat, which contain $(M - B + 1) \times (N - B + 1)$ rows.

D. Extraction of LBP-HF Features

In this step, rotation invariant LBP-HF features are extracted from each sorted block by calculating uniform LBP features and than applying Discrete Fourier Transform. LBP code for the center pixel (x, y) of block $B_i(x_i, y_i)$ is calculated using Equation 1 and it is given as:

$$LBP^{block}_{P_s,R_n}(x,y) = \sum_{p_s=0}^{P_s-1} T_f(B_i(x_i,y_i) - B_i(x_{pi}, y_{pi}))^p \quad (6)$$

where i varies from 1 to total number to blocks and $T_f(z)$ is the thresholding function as given in Equation 2. Using Equation 3 and 4 the LBP-HF features for each sorted block is calculated as follows:

$$|F^{block}(n,u)| = \sqrt{F^{block}(n,u)\overline{F^{block}(n,u)}} \quad (7)$$

These LBP-HF block features are stored in a matrix for further processing.

E. Feature Matching

In this step, Euclidean distances between feature vectors of corresponding blocks are calculated. For reducing false detection of forged region, distance threshold χ is determined experimentally. In this proposed system χ = 0.1 is selected by performing experiments on approximately 50 images of *CoMoFoD* dataset. The matching of the blocks start from the first row of the matrix Mat. The feature vector located in the i^{th} row is S_i, the distances of S_i with the remaining $\eta - 1$ features vectors is computed and stored in an array. Thereafter, the smallest distance SD from the array is calculated for forgery decision. This is given as:

$$SD = min[D(i, i+1), D(i, i+2), ..D(i, i+\eta)] \quad (8)$$

If SD is less than or equal to a distance threshold χ, then the corresponding blocks are considered as correctly matched and locations of these blocks are stored. Otherwise, these blocks are discarded. This matching procedure is continued until the last row of Mat is processed. Finally, all the matched block pairs are stored in a set Δ.

F. Forgery localization

The set Δ consists of all matched block pairs. Forged regions are highlighted in the tampered image by using the location of the copied and forged regions.

Algorithm 1 Proposed System-I

Input : $Forged\ image,\ Img.$
Output : $Detected\ forged\ regions\ in\ image$

procedure COPY_MOVE(Img)
  $[M,\ N\ scale] \leftarrow size(Img)$
  if $scale > 1$ **then**
    $I \leftarrow 0.2989R + 0.587G + 0.114B$ ▷ R, G and B are color components
  else
    $I \leftarrow Img$
  end if
  $Block\_size \leftarrow B \times B$
  $B \leftarrow Block\_partition(I, Block\_size)$ ▷ Overlapping block partition
  $\zeta \leftarrow (M - B + 1) \times (N - B + 1)$ ▷ ζ is the total number of blocks
  $B\_sort \leftarrow Sort(B)$ ▷ Lexicographical sorting
  $forged \leftarrow 0$
  for $x = 1\ to\ \zeta - 1$ **do**
    $S_1 \leftarrow LBP - HF\_feature(B\_sort_x)$
    $S_2 \leftarrow LBP - HF\_feature(B\_sort_{x+1})$
    $Match\_SD \leftarrow match(S_1, S_2)$ ▷ Match LBP-HF_feature
    if $Match\_SD \leqslant \chi$ **then**
      $Match \leftarrow Match\_SD$
      if $Match \geq 1$ **then**
        $Display\ the\ forged\ region$
        $forged \leftarrow 1$
      else
        $Discard\ the\ blocks$
      end if
    end if
  end for
end procedure

V. Experimental Results & Discussion of Proposed System-I

The proposed system improved the detection accuracy and also reduced the computation time. For experimental purpose, we used an HP machine with Intel Core i5-3230M (2.60 GHz) processor and 4 GB memory. The dataset used for system testing purpose is CoMoFoD developed by [19]. This is the image dataset used for forgery detection application. The dataset has 260 sets of images, in which 200 sets are in small image category with size 512×512 pixels and 60 sets are in the large image category with size 3000×2000 pixels. Each set consists of the original image, forged image, color mask and binary mask. In this dataset, images are labeled in the following manner; $N1\_M1\_M2$, where $N1$ is a three digit number, which is basically the image number used in the CoMoFoD dataset, $M1$ is used for marking the images, for original images $M1$ is O and forged images $M1$ is F. $M2$ represent the post-processing methods applied on the image such as: IB for image blurring, BC for brightness change, CR for color reduction and CA for contrast adjustments. Performance of proposed forgery detection algorithm is evaluated by detection error at the image level. At the image level detection of the false positive rate F_P, some original images have been falsely detected as forged. True positive rate T_P, is the number of correctly detected forged images. Mathematically, these can be expressed as:

$$F_P = \frac{No.\ of\ original\ images\ detected\ as\ forged}{Total\ no.\ original\ images} \quad (9)$$

$$T_P = \frac{No.\ of\ forged\ images\ detected\ as\ forged}{Total\ no.\ of\ forged\ images} \quad (10)$$

In this proposed system, experiments are performed on different post processed attacked images of the CoMoFoD dataset. Attack variation with their strength and the quantitative results in terms of TPR and FPR of the proposed system are given in the following tables: Table I present the experimental results of CoMoFoD dataset images, which have gone through the blurring post-processing operation. These results show that if the strength of blurring is increased then TPR decreased and FPR increased. Table II presents the experimental results of CoMoFoD dataset images, which have gone through the brightness changes. TPR and FPR for different contrast adjustments post-processed images is presented in Table III. Table IV present the experimental results of CoMoFoD dataset images, which have gone through different color reduction post-processing operation. Experimental results of the proposed system in terms of average computational time, TPR and FPR for different block size are presented in Table V. It can be observed from the results that TPR and FPR both are increasing in case of large block sizes since the probability of covering the complete forged region in large block size is high. On the other end, as the block size is large, the number of false matches also increases. Therefore, in the proposed system, we consider the trade-off between TPR and FPR for selecting the block size and block size 8×8 is selected, which gives optimum results across all images and all post-processing attacks. Qualitative detection results of the proposed system-I for six different images of the CoMoFoD dataset is given in Figure 2. In this figure the first row shows the input tampered images (without post processing operations) and corresponding forgery detection results are depicted in the second row. The copied and forged regions are highlighted in images through coloring of matched regions using green color. Qualitative results in case of different post-processing attacks are presented in Figures 3, 4 and 5. Detection results in the presence of color reduction and contrast adjustment post-processing operations are shown in Figure 3. Similarly, Figure 4 show the detection results in the presence of brightness levels and color reduction post-processing operations. These results show that the proposed system is better in terms of TPR

and FPR for all post-processing operation viz as blurring, brightness change, contrast and color variation.

TABLE I
TPR AND FPR FOR DIFFERENT BLUR PARAMETERS OF SYSTEM-I.

Image label	σ	TPR(%)	FPR(%)
_F_IB1	0.0005	98.8	6.8
_F_IB2	0.005	97.6	10.8
_F_IB3	0.009	96.3	14.1

TABLE II
TPR AND FPR FOR DIFFERENT BRIGHTNESS LEVELS OF SYSTEM-I

Image label	lower & upper bound	TPR(%)	FPR(%)
_F_BC1	(0.01, 0.8)	99.2	5.4
_F_BC2	(0.01, 0.9)	98.7	6.3
_F_BC3	(0.01, 0.95)	97.2	8.2

TABLE III
TPR AND FPR FOR DIFFERENT CONTRAST ADJUSTMENTS OF SYSTEM-I.

Image label	lower & upper bound	TPR(%)	FPR(%)
_F_CA1	(0.01, 0.8)	99.3	5.8
_F_CA2	(0.01, 0.9)	98.6	6.6
_F_CA3	(0.01, 0.95)	97	7.9

TABLE IV
TPR AND FPR FOR DIFFERENT COLOR REDUCTIONS OF SYSTEM-I.

Image label	Intensity levels	TPR(%)	FPR(%)
_F_CR1	32	99.2	5.6
_F_CR2	64	98.6	6.3
_F_CR3	128	97.4	7.5

TABLE V
AVERAGE PROCESSING TIME, TPR AND FPR FOR DIFFERENT BLOCK SIZES OF SYSTEM-I

Block Size	Time (in Sec)	TPR (%)	FPR (%)
4×4	25.45	94.8	6.4
8×8	22.85	98.4	7.4
12×12	21.36	98.9	11.6
16×16	20.76	99.4	16.2

TABLE VI
COMPARATIVE RESULTS OF SYSTEM-II IN TERMS OF TPR, FPR AND PROCESSING TIME FOR BLUR PARAMETERS $\mu = 0, \sigma = 0.0005$

Method	Times (in Sec)	TPR(%)	FPR(%)
Kulkarni*et al.*[30]	61	82.5	11.8
Wo *et al.*[2]	30.2	93.8	7.2
Huang *et al.*[31]	135	93.2	7.4
Proposed System-I	22.85	98.6	7.1

Tables VI and VII presents the comparison of performance of the proposed system with other existing systems for the blurring post- processing operation. It is observed from comparative results that the proposed system outperformed the block based existing techniques in case of blurring attacks.

Table VIII presents the comparison of performance of the proposed system with other existing systems. It is observed from comparative results that the proposed system outperformed the block based existing techniques in most aspects.

TABLE VII
COMPARATIVE RESULTS OF SYSTEM-II IN TERMS OF TPR, FPR AND PROCESSING TIME FOR BLUR PARAMETERS $\mu = 0, \sigma = 0.005$

Method	Times (in Sec)	TPR(%)	FPR(%)
Kulkarni*et al.*[30]	61	80.5	14.8
Wo *et al.*[2]	30.2	90.8	9.12
Huang *et al.*[31]	135	91.2	9.4
Proposed System-I	22.85	98.1	7.8

TABLE VIII
COMPARISON OF SYSTEM-I WITH EXISTING METHODS IN TERMS OF AVERAGE TPR, FPR AND PROCESSING TIME

Method	Times (in Sec)	TPR(%)	FPR(%)
Kulkarni*et al.*[30]	61	82.5	11.8
Wo *et al.*[2]	30.2	93.8	7.2
Huang *et al.*[31]	135	93.2	7.4
Proposed System-I	22.85	98.4	7.4

VI. PROPOSED SYSTEM-II

In this section, the proposed system-II is described. This system is based on the extraction and matching using fast Walsh Hadamard Transform (FWHT). In this system, first the image is divided into overlapping blocks and FWHT features are extracted from each block. Thereafter, matching is performed among the extracted block FWHT features. The pseudo-code for the proposed system-II is given in Algorithm 2 and the processing steps of the proposed system-II is given as follows:

A. Preprocessing

In the preprocessing step, conversion from RGB image to grey scale image is performed in the same manner as the previous system-I.

B. Overlapping blocks division

Like the proposed system-I, in this system, the input tampered image is divided into 8×8 sized overlapping blocks. All these overlapping blocks are stored row wise in a matrix for further processing.

C. Fast Walsh Hadamard Transform features extraction

The Walsh Hadamard Transform is a non-sinusoidal, orthogonal transformation that divides the image blocks into Walsh functions. Walsh function is a set of orthogonal, rectangular waveforms. Walsh Hadamard Transform has a faster version of its, which is named as Fast Walsh Hadamard Transform (FWHT). The FWHT calculation process uses only real arithmetic operation. Therefore, it is efficient in terms of storage requirements and execution times. The FWHT for L length, overlapping block B is defined as:

$$FWHT_n = \frac{1}{L} \sum_{i=0}^{L-1} B_i Walsh(n, i) \qquad (11)$$

where $i = 0, 1....L-1$ and $Walsh(n, i)$ are the first n Walsh functions. These FWHT features are extracted from each block and stored in a matrix for further processing. Detailed descriptions about the calculation of FWHT is available in paper [9].

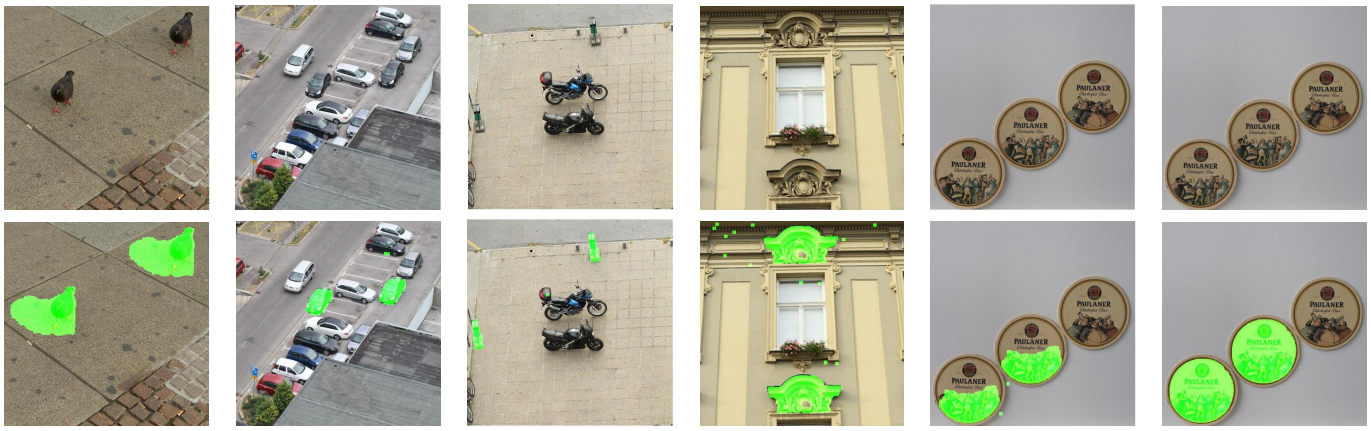

Fig. 2. Results of proposed System-I: CoMoFoD tampered images are pictured in the first row; the corresponding detection results are shown in the second row.

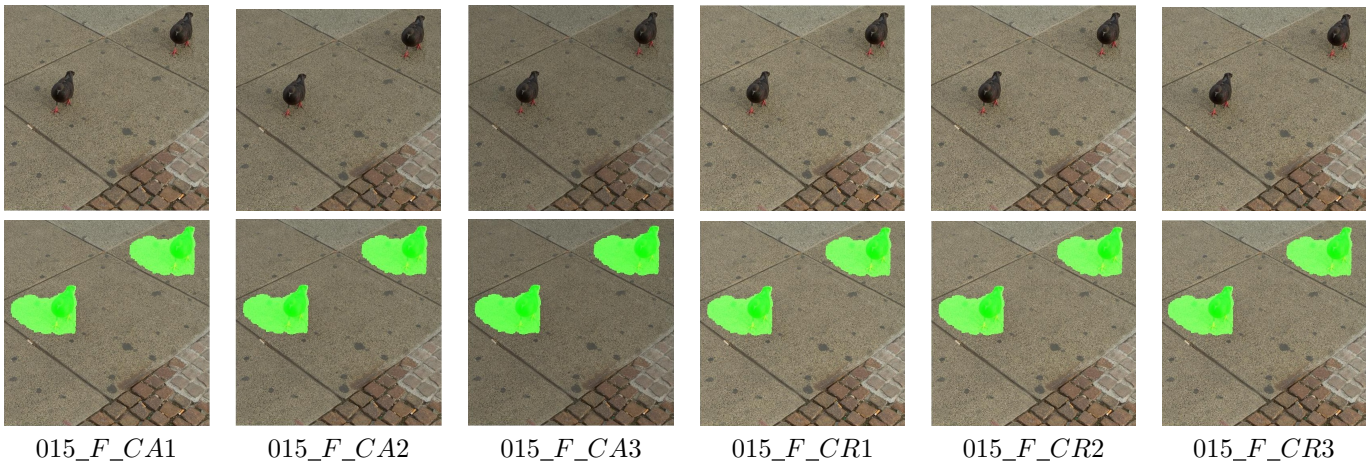

Fig. 3. Results of proposed System-I : Different attacked tampered images are shown in the first row; the corresponding detection results are shown in the second row.

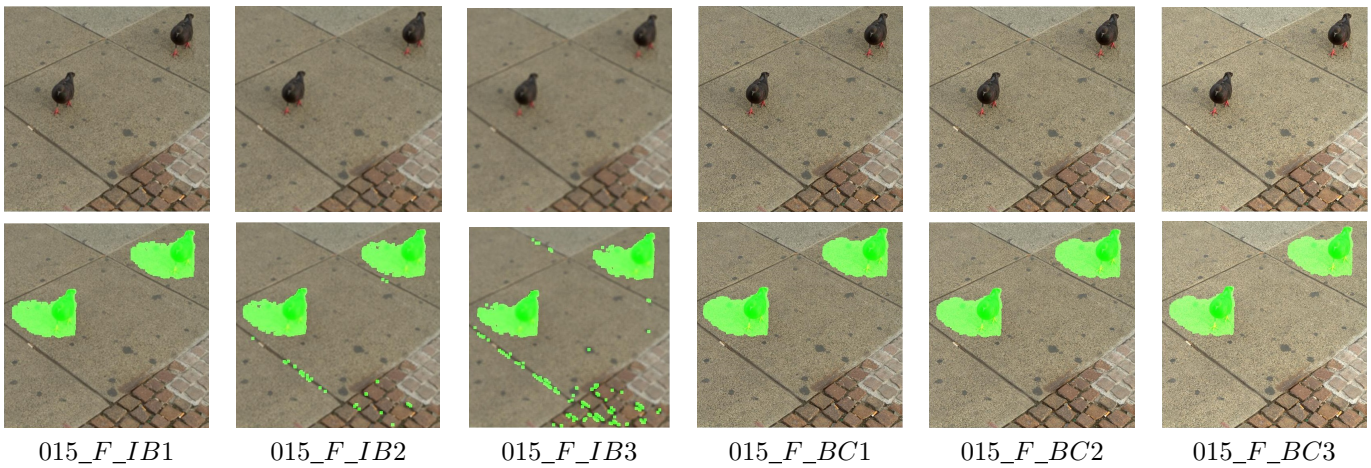

Fig. 4. Results of proposed System-I : Different attacked tampered images are in the first row; the corresponding detection results are shown in the second row.

Fig. 5. CoMoFoD tampered different attack images are pictured in the first row; the corresponding detection results are shown in the second row.

Algorithm 2 Proposed System-II

Input : *Tampered image Img*
Output : *Detected Tampered regions in image*

procedure COPY_MOVE_DETECTION(Img)
  $[M,\ N\ scale] \leftarrow size(Img)$
  if $scale > 1$ **then**
    $I \leftarrow RGB\ to\ Grey\ (Img)$ ▷ Grey scale conversion
  else
    $I \leftarrow Img$
  end if
  $Block\_size \leftarrow B \times B$
  $B \leftarrow Block\_division(I, Block\_size)$ ▷ Overlapping block division
  $\eta \leftarrow (M - B + 1) \times (N - B + 1)$ ▷ η is the total number of blocks
  $forged \leftarrow 0$
  for $X = 1\ to\ \eta$ **do**
    $Mat \leftarrow FWHT\_feature(B(X))$ ▷ Extraction of FWHT features from all blocks
    $M\_sort \leftarrow Sort(Mat)$ ▷ Lexicographical sorting
  end for
  $Count \leftarrow 0$
  for $Y = 1\ to\ length(M\_sort) - 1$ **do**
    $R \leftarrow match[M\_sort(Y,:), M\_sort(Y+1,:)]$
    $index \leftarrow R$ ▷ Store the index of consecutive matched rows
    $S_v \leftarrow Shift\_vector(1 : index, :)$ ▷ Shift vectors construction for all the matched block pairs
    $Count \leftarrow Count++$
  end for
  if $Count(S_v) \geq T_h$ **then**
    $loc \leftarrow location(S_v)$ ▷ Store the locations of corresponding shift vectors
    Display the forged region in image
    $forged \leftarrow 1$
  else
    Discard the blocks
  end if
end procedure

D. Matching procedure and Forgery localization

The FWHT transform coefficients of each block is stored row wise in a matrix Mat. There are $(M - B + 1) \times (N - B + 1)$ number of blocks of size $B \times B$, the matrix Mat consists of $(M - B + 1) \times (N - B + 1)$ rows and $B \times B$ columns. Matrix Mat is sorted lexicographically row wise. If two successive rows of the matrix Mat are matched, then the index values of the matched blocks is stored in an array and the shift count value is incremented. This process can be stated as: Initialize, the shift vector count to zero and let (i_1, j_1) and (i_2, j_2) be the index values of two matching blocks. Then shift vector S_v between these matched blocks is described as $S_v = (i_1 - i_2, j_1 - j_2)$, since the shift vectors $-S_v$ and S_v correspond to the same shift. Hence $-S_v$ is normalized through multiplying by -1, so that the shift vector becomes $S_v \geqslant 0$. After each matching iteration, shift vector count is incremented by one. When all the rows of the block matrix Mat have been processed, the final value of count indicates the total number of occurrences, where different normalized shift vectors occurred. This matching procedure finds all normalized shift vectors $s_1, s_2, ..., s_k$, whose matching value is more than a given threshold T_h. i.e, $Count(S_v(k)) > T_h$. The threshold T_h depends on the size of the smallest region that can be detected as forged. The selection of T_h affect the detection results. Larger value of T_h may result to miss some matched blocks and on the other hand smaller value of T_h may consider some false matches. The blocks that are contributing to the same shift vector are the matched blocks and detected as copied and moved blocks. Finally, these matched blocks are localized in the image through coloring using green color.

VII. EXPERIMENT RESULTS & DISCUSSION OF PROPOSED SYSTEM-II

The proposed system-II is experimented using the CoMoFoD dataset forged images. Quantitative results in terms of average TPR and FPR for different post-processing operations are given in Tables IX, X, XI and XII. Figure 6 shows the forgery detection results of the proposed system-II for four different images. First column of Figure 6 consist of the input forged images; second and third columns consist of results of forgery detection and binary map of detected forgery respectively. Columns 4, 5 and 6 consist of detected

forged regions of images which are post-processed by IB1, IB2 and IB3 blurring parameter respectively. Range of blur parameter along with quantitative results of proposed system-II for blur post-processing operation are given in Table IX.

The average computational time, TPR and FPR of the proposed system-II for different block sizes are presented in Table XI. It can be observed from the results that TPR and FPR both are increasing in case of large block sizes. The probability of covering complete forged region in large block size is high and at the same time as the block size is large, the number of false matches also increases. Therefore, like system-I, in the proposed system-II also, considering the trade-off between TPR and FPR for selecting the block size and the block size 8×8 is selected. This gives optimum results across all the post-processing attacks. From experimental results it is observed that the average processing time for the proposed system-II is slightly more than proposed system-I. However, system-II is more invariant to blur post-processing operation.

TABLE IX
TPR AND FPR FOR DIFFERENT BLUR PARAMETERS OF SYSTEM-II

Image label	σ	TPR(%)	FPR(%)
_F_IB1	0.0005	99.8	6.2
_F_IB2	0.005	99.2	7.8
_F_IB3	0.009	98.3	10.1

TABLE X
TPR AND FPR FOR DIFFERENT BRIGHTNESS LEVELS OF SYSTEM-II.

Image label	lower & upper bound	TPR(%)	FPR(%)
_F_BC1	(0.01, 0.8)	98.4	6.2
_F_BC2	(0.01, 0.9)	97.5	7.1
_F_BC3	(0.01, 0.95)	96.2	9.1

TABLE XI
TPR AND FPR FOR DIFFERENT CONTRAST ADJUSTMENTS OF SYSTEM-II

Image label	lower & upper bound	TPR(%)	FPR(%)
_F_CA1	(0.01, 0.8)	98.5	6.4
_F_CA2	(0.01, 0.9)	97.6	7.2
_F_CA3	(0.01, 0.95)	96.2	8.6

TABLE XII
TPR AND FPR FOR DIFFERENT COLOR REDUCTIONS OF SYSTEM-II

Image label	Intensity levels	TPR(%)	FPR(%)
_F_CR1	32	98.1	6.6
_F_CR2	64	97.6	7.3
_F_CR3	128	96.3	8.2

TABLE XIII
AVERAGE PROCESSING TIME, TPR AND FPR OF PROPOSED SYSTEM-II

Block Size	Time (in Sec)	TPR (%)	FPR (%)
4×4	35.45	95.8	6.8
8×8	32.65	98.3	7.3
12×12	29.36	98.9	10.6
16×16	28.76	99.6	14.2

TABLE XIV
COMPARATIVE RESULTS OF SYSTEM-II IN TERMS OF TPR, FPR AND PROCESSING TIME FOR BLUR PARAMETERS $\mu = 0, \sigma = 0.0005$

Method	Times (in Sec)	TPR(%)	FPR(%)
Khan *et al.*[30]	61	82.5	11.8
Yang *et al.*[25]	30.2	93.8	7.2
Huang *et al.*[31]	135	93.2	7.4
Proposed method	32.65	99.8	6.2

TABLE XV
COMPARATIVE RESULTS OF SYSTEM-II IN TERMS OF TPR, FPR AND PROCESSING TIME FOR BLUR PARAMETERS $\mu = 0, \sigma = 0.005$

Method	Times (in Sec)	TPR(%)	FPR(%)
Khan *et al.*[30]	61	80.5	14.8
Yang *et al.*[25]	30.2	90.8	9.12
Huang *et al.*[31]	135	91.2	9.4
Proposed method	32.65	99.2	7.8

TABLE XVI
COMPARATIVE RESULTS OF SYSTEM-II IN TERMS OF TPR, FPR AND PROCESSING TIME FOR BLUR PARAMETERS $\mu = 0, \sigma = 0.009$

Method	Times (in Sec)	TPR(%)	FPR(%)
Khan *et al.*[30]	61	78.5	18.8
Yang *et al.*[25]	30.2	85.8	16.12
Huang *et al.*[31]	135	83.2	15.4
Proposed method	32.65	98.3	10.1

Qualitative results of the proposed system-II for different post-processing operations are presented in Figure 7. The first row of the Figure 7 show the detection results in the presence of contrast adjustment and brightness variation post-processing attacks. The second row of this figure shows the detection results for the proposed system-II in the presence of different color reduction post-processing attacks. Tables XIV, XV and XVI present the performance comparison of the proposed system-II with other existing systems in presence of different blur parameters. It is observed from comparative results that this system outperformed the existing techniques in all aspects except computation time which is slightly more than the one reported in [25].

VIII. CONCLUSIONS AND FUTURES SCOPE

In this paper, we have proposed two different systems for block-based copy-move forgery detection. The proposed system-I is based on the Local Binary Pattern Histogram Fourier Features and the proposed system-II is based on the Fast Walsh Hadamard transform. Due to rotation invariant characteristic of LBP-HF, the proposed system-I is efficient in forgery detection in comparison to existing block-based methods. Table V gives the quantitative performance of proposed system-I for different block sizes. Figure 3 and Figure 4 show the performance of the system-I on different post-processing attacks. Figure 6 show the results of system-II on different blurred images. Experimental results show that both proposed systems are able to detect small copied regions with the minimum false match. System-I is able to detect forgery accurately across all the post-processing operations mentioned in this paper. Experimental results of system-II shows that it is more invariant to blurring post-processing operation than system-I. Tables VIII and XII present the comparison of the proposed system-I and system-II with

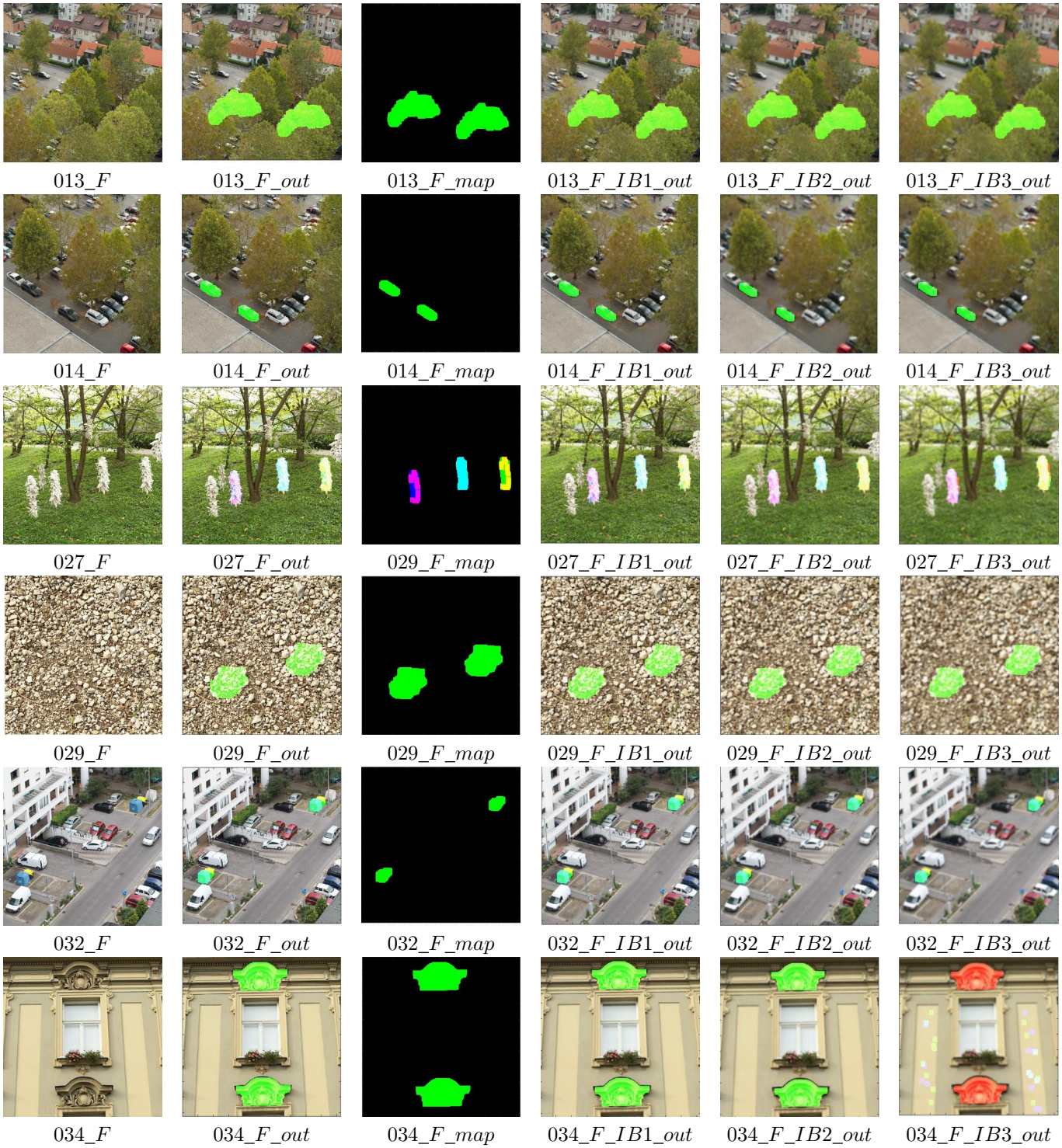

Fig. 6. Results of proposed System-II: Tampered input images are shown in the first column, second column presents the corresponding detection results, third column depicts the output binary map and detection results using different blur parameters are given from columns four to six.

existing techniques respectively. It showed that the proposed systems outperformed the existing block-based approaches. However, some improvements are needed to detect multiple forgeries present in the image and detection of accurate forgery in highly similar regions including post-processing operations. This work can also be extended for detection of forgery in the presence of geometric transformation attacks.

REFERENCES

[1] X.-y. Wang, Y.-n. Liu, H. Xu, P. Wang, and H.-y. Yang, "Robust copy-move forgery detection using quaternion exponent moments," *Pattern Analysis and Applications*, pp. 1–17, 2016.

[2] Y. Wo, K. Yang, G. Han, H. Chen, and W. Wu, "Copy-move forgery detection based on multi-radius PCET," *IET Image Processing*, vol. 11, no. 2, pp. 99–108, 2017.

[3] J. Zhao and J. Guo, "Passive forensics for copy-move image forgery using a method based on {DCT} and {SVD}," *Forensic Science International*, vol. 233, no. 13, pp. 158–166, 2013.

[4] G. Muhammad, M. Hussain, K. Khawaji, and G. Bebis, "Blind copy move image forgery detection using dyadic undecimated wavelet transform," in *International Conference on Digital Signal Processing (DSP)*, July 2011, pp. 1–6.

[5] E. Ardizzone, A. Bruno, and G. Mazzola, "Detecting multiple copies in tampered images," in *IEEE International Conference on Image Processing*, Sept 2010, pp. 2117–2120.

[6] D. G. Lowe, "Distinctive image features from scale-invariant key-

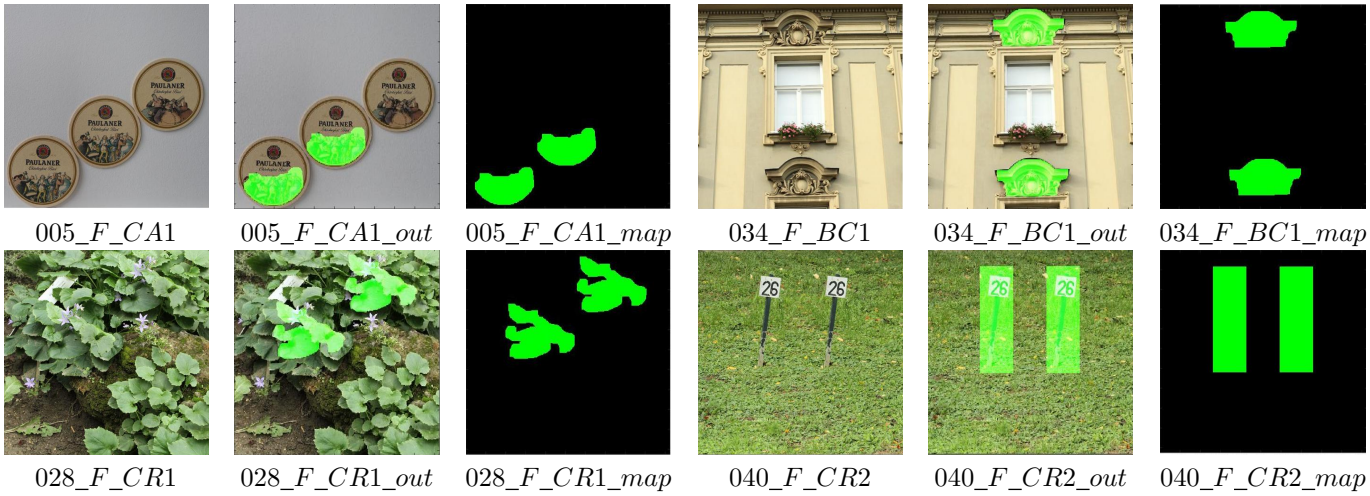

Fig. 7. Results of proposed System-II: Tampered images (in presence of contrast adjustment and brightness variation) and corresponding detection results are shown in first row. Tampered images (in presence of color reduction) and corresponding detection results are shown in second row.

points," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[8] T. Ahonen, J. Matas, C. He, and M. Pietikäinen, *Rotation Invariant Image Description with Local Binary Pattern Histogram Fourier Features*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 61–70.

[9] T. Beer, "Walsh transforms," *American Journal of Physics*, vol. 49, no. 5, pp. 466–472, 1981.

[10] R. K. Jha, B. Soni, and K. Aizawa, "Logo extraction from audio signals by utilization of internal noise," *IETE Journal of Research*, vol. 59, no. 3, pp. 270–279, 2013.

[11] R. K. Jha, B. Soni, R. Chouhan, and K. Aizawa, *Improved Watermark Extraction from Audio Signals by Scaling of Internal Noise in DCT Domain*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 235–243.

[12] L. Li, S. Li, H. Zhu, S. Chu, J. F. Roddick, and J. Pan, "An efficient scheme for detecting copy-move forged images by local binary patterns," *IEEE Transactions on Image Processing*, vol. 4, no. 1, pp. 46–56, 2016.

[13] R. Davarzani, K. Yaghmaie, S. Mozaffari, and M. Tapak, "Copy-move forgery detection using multiresolution local binary patterns," *Forensic Science International*, vol. 231, no. 13, pp. 61–72, 2013.

[14] D. M. Uliyan, H. A. Jalab, and A. W. A. Wahab, "Copy move image forgery detection using hessian and center symmetric local binary pattern," in *IEEE Conference on Open Systems (ICOS)*, Aug 2015, pp. 7–11.

[15] G. Muhammad, M. H. Al-Hammadi, M. Hussain, A. M. Mirza, and G. Bebis, "Copy move image forgery detection method using steerable pyramid transform and texture descriptor," in *Eurocon 2013*, July 2013, pp. 1586–1592.

[16] G. Muhammad, M. H. Al-Hammadi, M. Hussain, and G. Bebis, "Image forgery detection using steerable pyramid transform and local binary pattern," *Machine Vision and Applications*, vol. 25, no. 4, pp. 985–995, 2014.

[17] A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, G. Bebis, and H. Mathkour, "Passive detection of image forgery using DCT and local binary pattern," *Signal, Image and Video Processing*, vol. 11, no. 1, pp. 81–88, 2017.

[18] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, and G. Serra, "A SIFT-based forensic method for copy-move attack detection and transformation recovery," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1099–1110, Sept 2011.

[19] K. Li, H. Li, B. Yang, Q. Meng, and S. Luo, *Detection of Image Forgery Based on Improved PCA-SIFT*. Springer International Publishing, 2014, pp. 679–686.

[20] Z. Mohamadian and A. A. Pouyan, "Detection of duplication forgery in digital images in uniform and non-uniform regions," in *International Conference on Computer Modelling and Simulation*, April 2013, pp. 455–460.

[21] R. Sekhar and R. S. Shaji, *A Study on Segmentation-Based Copy-Move Forgery Detection Using DAISY Descriptor*. New Delhi: Springer India, 2016, pp. 223–233.

[22] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient dense-field copy-move forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2284–2297, Nov 2015.

[23] M. F. Hashmi, A. R. Hambarde, and A. G. Keskar, "Robust image authentication based on HMM and SVM classifiers," *Engineering Letters*, vol. 22, no. 4, pp. 183–193, 2014.

[24] B. Soni, P. K. Das, and D. M. Thounaojam, "CMFD: A detailed review of block based and key feature based techniques in image copy-move forgery detection," *IET Image Processing*, November 2017.

[25] Y. Bin, S. Xingming, C. Xianyi, Z. Jianjun, and L. Xu, "An efficient forensic method for copymove forgery detection based on DWT-FWHT," *Radioengineering*, vol. 22, no. 4, pp. 1098–1105, Nov. 2015.

[26] Y. Shin, "Fast detection of copy-move forgery image using two step search algorithm," *International Journal of Security and Its Applications*, vol. 10, no. 5, pp. 203–214, 2016.

[27] J.-C. Lee, C.-P. Chang, and W.-K. Chen, "Detection of copy-move image forgery using histogram of orientated gradients," *Information Sciences*, vol. 321, no. C, pp. 250–262, Nov. 2015.

[28] J. Zhang and T. Tan, "Brief review of invariant texture analysis methods," *Pattern Recognition*, vol. 35, no. 3, pp. 735–747, 2002.

[29] G. Zhao, T. Ahonen, J. Matas, and M. Pietikainen, "Rotation-invariant image and video description with local binary pattern features," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1465–1477, April 2012.

[30] A. Kulkarni and S. Khan, "An efficient method for detection of copy-move forgery using discrete wavelet transform," *International Journal on Computer Science and Engineering*, vol. 2, no. 5, pp. 1801–1806, 2010.

[31] Y. Huang, W. Lu, W. Sun, and D. Long, "Improved DCT-based detection of copy-move forgery in images," *Forensic Science International*, vol. 206, no. 1, pp. 178–184, 2011.