Edge-Texture Feature based Image Forgery Detection with Cross Dataset Evaluation

Khurshid Asghar[a], Xianfang Sun[b], Paul L. Rosin[c], Mubbashar Saddique[d], Muhammad Hussain[e], Zulfiqar Habib[f]
[a] Department of Computer Science, University of Okara, Pakistan
[d, f] Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Pakistan
[e] Department of Computer Science, King Saud University, Riyadh, Saudi Arabia
[a, b, c]School of Computer Science & Informatics, Cardiff University, UK
[a]khasghar@uo.edu.pk, [b]SunX2@cardiff.ac.uk, [c]rosinpl@cardiff.ac.uk
[d]mubbashar.saddique@ciitlahore.edu.pk, [e]mhussain@ksu.edu.sa, [f]drzhabib@cuilahore.edu.pk

ABSTRACT: ***A digital image is a rich medium of information. The development of user-friendly image editing tools has given rise to the need for image forensics. The existing methods for the investigation of the authenticity of an image perform well on a limited set of images or certain datasets but do not generalize well across different datasets. The challenge of image forensics is to detect the traces of tampering which distorts the texture patterns. A method for image forensics is proposed, which employs Discriminative robust local binary patterns (DRLBP) for encoding tampering traces and a support vector machine (SVM) for decision making. In addition, to validate the generalization of the proposed method, a new dataset is developed that consists of historic images, which have been tampered with by professionals. Extensive experiments were conducted using the developed dataset as well as the public domain benchmark datasets; the results demonstrate the robustness and effectiveness of the proposed method for tamper detection and validate its cross-dataset generalization. Based on the experimental results, directions are suggested that can improve dataset collection as well as algorithm evaluation protocols. More broadly, discussion in the community is stimulated regarding the very important, but largely neglected, issue of the capability of image forgery detection algorithms to generalize to new test data.***

KEYWORDS*: Image forensics; Image forgery detection; Copy-move; Splicing; Cross-dataset evaluation.*

1 Introduction

Digital images are rich source of information in areas such as forensic science, medical imaging, surveillance, journalism, e-services, and social networking. On social media applications such as WhatsApp and Facebook 1.8 billion images are uploaded daily [6]. It has become much

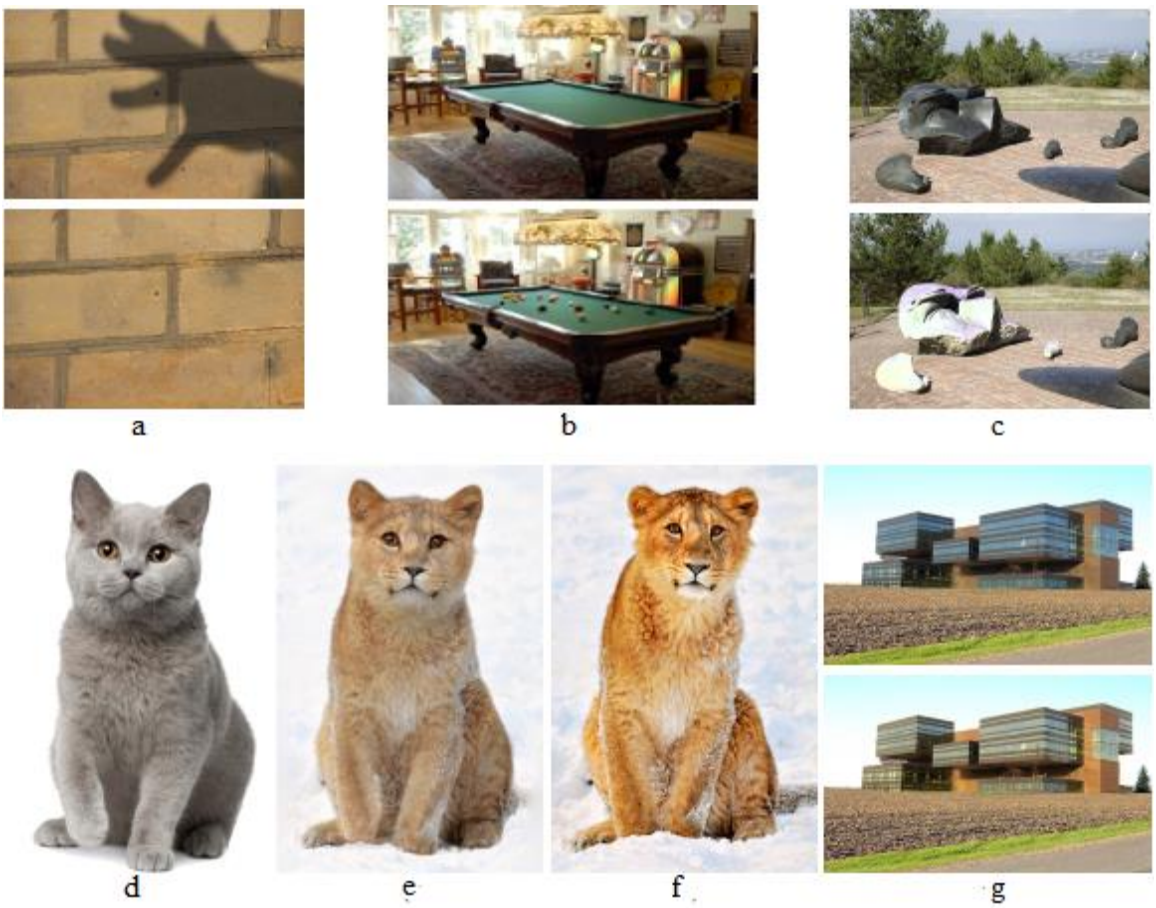


Fig. 1. a) The hand shadow from the top image has been removed on the bottom image [2], b) The balls in the bottom picture are not real, which are inserted using circle objects along with light interactions [3], c) The bottom image was filtered to perform color editing on some of the stone statues [5], d-f) The cat in (e) is a composite of the cat in (d) and the leopard in (f) [8], g) The building was spliced on the field in the top image, and in the bottom, it had its lighting adjusted to match the composition [10].

easier to manipulate the content of images due to the availability of powerful image editing tools such as Adobe Photoshop [9], Corel-DRAW [9] etc. It is difficult for humans to visually detect such image modifications [9, 11].

Figure 1 shows different ways of image tampering such as shadow removal, inserting fake objects, color filtering, image composition and illumination adjustment. An image may be tampered using the following operations: i) transferring an object or region from one image to another, or even to the same image; this is the most common type of forgery, and encompasses both splicing and copy-move operations, see Figure 2, ii) inserting fake objects into an image or manipulating an existing object to change its properties, iii) altering image parts related to lights and lighting. iv) removing an object or region from the image and hiding it.

Due to increase in the number of tampered and retouched images, digital contents are nowadays not considered a reliable source of information. It is very difficult to have reliable and

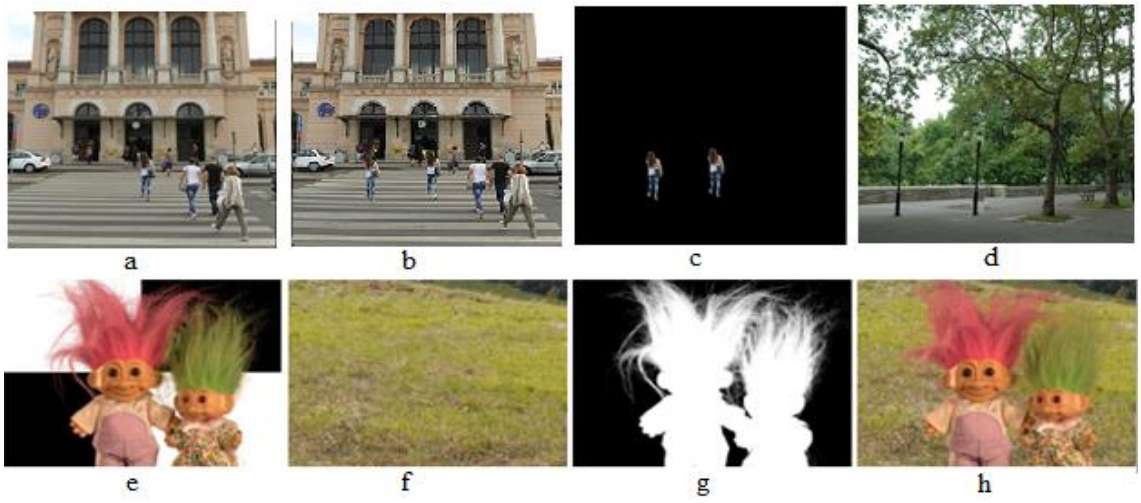


Fig. 2. Examples of copy-move (a-d) and splicing (e-h) forgeries: a) Original image [1], b) Tampering is performed by copy and pasting a girl object in the same image to another location [1], c) Transfer mask of (b), d) Object street lamp is copied to another location in the same image [4], e) Source image [7], f) Target image [7], g) Transference mask of source image, h) Spliced image.

efficient image forensic methods, due to the advancement and sophistication in image manipulation operations. Image authentication without using any prior information is called passive or blind approach [9, 12-14] and has received reasonable attention in the literature due to its ability to find forgeries in images by exploiting the traces/artifacts via modeling the artifacts of forgery (discontinuities and inconsistencies in the form of edges, lines and corners) left by the tampering process. These traces act as features for image tampering detection [9, 15, 16].

To construct a simplified and computationally efficient image forgery detection model we employ Discriminative Robust Local Binary Patterns (DRLBP) [17], which encodes the structural changes that occur in images due to forgery. During model construction, a real forgery example dataset is required to validate the model. For this purpose, a new dataset consisting of historic images that have been tampered by professionals is developed, referred to in this paper as the Forged Real Images Throughout History (FRITH). It is used to validate the developed image authentication process together with other existing image forgery evaluation benchmark datasets.

This work has the following four major contributions.

- First, a new dataset FRITH (see Section 4 for details) is developed to evaluate the image forgery detection method on realistic scenarios.

- Second, a robust image forgery detection method based on DRLBP and SVM is proposed to identify whether a given image is authentic or forged. Extraction of salient features are important for any image forgery detection system. Since the texture and contrast of forged images are different to those of authentic images due to structural changes after forgery, the DRLBP code is computed by assigning a weight factor (w) carrying gradient information to capture edge and texture information together. The contrast is high near the boundary of forged areas in the forged images; therefore, the voted bin value is expected to be high as compared to authentic images which provides additional information of tampering cues (edges).
- Third, the proposed approach is evaluated on cross datasets (i.e. training and testing on different datasets) to generalize to new data in real applications.
- Fourth, a thorough evaluation and comparison on a variety of benchmark datasets is performed.

The rest of the paper is organized as follows. Related works on image forgery detection are reviewed in Section 2. The detail of the proposed technique is described in Section 3. Section 4 is about datasets and evaluation criteria. System parameters are described in Section 5. Experimental results are presented, discussed, and compared with existing works in Section 6. The paper is concluded in Section 7.

2 Related Work

Over the last two decades, numerous works had been performed to detect different types of forgeries in images. Image forgery detection approaches are divided into active and passive (or blind) categories. Active approaches use detection of embedded watermarks or signatures to ensure the authenticity of images [18-23]. Such approaches are limited, because it is difficult to maintain prior information of such pre-embedded watermarks, signatures and secret keys [20]. Therefore, to detect image forgery without having any prior knowledge is an active research field.

Inspired by the research in [24] for perceiving tampered human speech, Ng et al. [25, 26] proposed to detect image forgery by means of phase and magnitude features of images. The Columbia Image Splicing Detection Evaluation (CISD) dataset was used for evaluation [27]. The detection accuracy was only 72%, due to differences in the frequency characteristics between audio signals and digital images. High order wavelet features were passed to an SVM classifier for image forgery detection in [26] and they achieved 80.15% accuracy. Wang et al. [28] detected

image forgery using the Gray Level Co-occurrence Matrix (GLCM) of the YCbCr image. The CASIA v1.0 dataset was used for evaluation. The achieved accuracy was 90.5% on the Cr channel. Subsequently, Wang et al. [29] extracted transition probability features from the Cb channel, achieving an accuracy of 95.6% on a subset of the CASIA v2.0 dataset.

A technique based on the modified Run-Length Run-Number (RLRN) was proposed by Zhao et al. [30]. He used chrominance components for feature extraction and achieved 94% detection rate. Muhammad et al. [31] decomposed Cb and Cr components using the Steerable Pyramid Transform (SPT) into sub-bands and extracted features using Local Binary Patterns (LBP) from these sub-bands. Significant features were selected and then passed to an SVM for classification. The Columbia Color DVMM dataset, CASIA v1.0 and CASIA v2.0 datasets were used for experiments. The best accuracies reported were 94.8% on CASIA v1.0 dataset, 97.33% on CASIA v2.0 dataset, and 96.39% on Columbia Color DVMM dataset. Cozzolino et al. [32] used dense features and achieved 95.92% and 94.61% detection accuracy on FAU and GRIP datasets, respectively. The datasets contain 48 and 80 authentic and copy-move forged images, respectively.

Rota et al. [33] proposed a blind deep learning approach based on Convolutional Neural Networks (CNN) for tampered image classification. They used the CAISA v2.0 dataset for experiments and achieved 97.44% detection rate.

Hussain et al. [34] evaluated image forgery detection using Weber Local Descriptor (WLD) and LBP. The tampering traces were computed from chrominance components using WLD and encoded as features using binary patterns. SVM was employed for classification. The method was evaluated on DVMM, CASIA v1.0 and CASIA v2.0 datasets. The impact of WLD and LBP to model tampering traces was thoroughly explored. The performance of the method was reasonable.

Cattaneo et al. [35] performed experimental analysis of image forgery detection and used the approach of Lin et al. [36] for JPEG tampered image detection. For tampering detection the authors in [35] estimated the image luminance quality factor and relative frequency of tampered blocks both in authentic and forged images in the CASIA v2.0 dataset and found that the images of the CASIA v2.0 dataset contain some statistical artifacts which can help the detection process. To confirm this, they first used the CASIA v2.0 dataset to evaluate the performance of Lin et al.'s algorithm. According to their experiments, the considered algorithm performs very well on the CASIA v2.0 dataset. Some variants of the original algorithm were then specifically tuned according to the characteristics of the CASIA v2.0 dataset. These variants performed better than

their original counterpart. Then a new unbiased dataset UNISA [35] was assembled and a new set of experiments was carried out on these images. The results showed that the performance of the algorithm and its variants substantially decreased, proving that the algorithm tuned on CASIA v2.0 is not robust.

Pham et al., used Markov features in DCT domain to identify whether a given image is authentic or forged. SVM was used for classification [37]. Experiments were performed using CASIA v1.0 and CASIA v2.0 datasets and achieved 96.90 % detection accuracy. The method is evaluated on limited datasets and focused only splicing forgery.

Wang et al., in [38], proposed mass filter banks using fast Fourier transformation. The features were then fed to ResNet [39], to classify whether an image is tampered or authentic. Yan et al., in [40], proposed a method based on deep learning using CNN architecture. The model is trained on recolored and natural images. The method achieved 83.98% detection accuracy and evaluated on variety of recolored and natural images. However, evaluation is not performed on forgeries like copy-move and splicing.

Review of existing image forgery detection techniques shows that encoding structural changes occurred in images due to forgery is still a challenge. The success of an image forgery detection method relies upon how it copes with the structural changes in forged images. In our experiments we explored LBP, WLD, and DRLBP texture descriptors and found that DRLBP models these structural changes well. A variety of benchmark datasets are used for evaluation in our experimental analysis. To ensure the robustness (i.e., the ability to authenticate images in general) of the proposed algorithm, a cross-dataset protocol is adopted, i.e., training and testing are performed on different data sets that have been collected independently.

3 Proposed Image Forgery Detection System

The architectural diagram of the proposed approach is shown in Figure 3. The system is composed of four major components i.e., i) preprocessing, ii) feature extraction, iii) classification model building, and, iv) testing using the trained model with cross data validation. The model is trained using an SVM classifier on a set of images (see model training component of Figure 3), then the trained model is used to test/recognize (see testing component of Figure 3) unseen authentic and forged images.

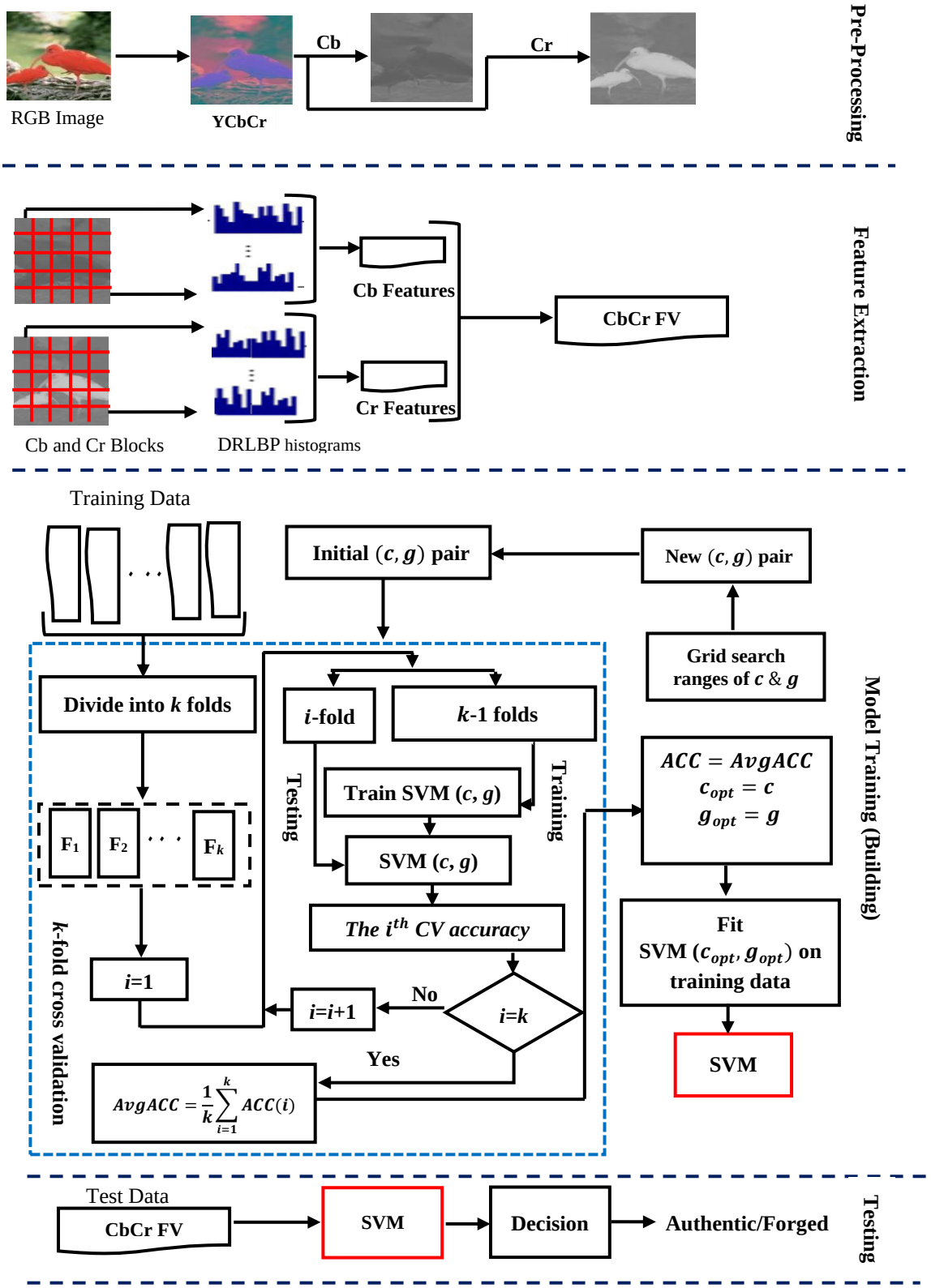


Fig. 3. The proposed architecture of image forgery detection system.

3.1 Pre-Processing

Tampering traces are embedded in the form of edge irregularities [41]. Before feature extraction it is important to select an appropriate color space. A tampered image is shown in respective components of RGB, HSV and YCbCr color spaces in Figure 4. It is observed that all components describe the image content in detail except chroma components (CbCr), which emphasize the weak signal content (little image detail) of the image. In general, the content of an image is too strong to hide the tampering traces. Edge irregularities caused by tampering can be noticed in chroma components [41], therefore, this study uses the YCbCr color space.

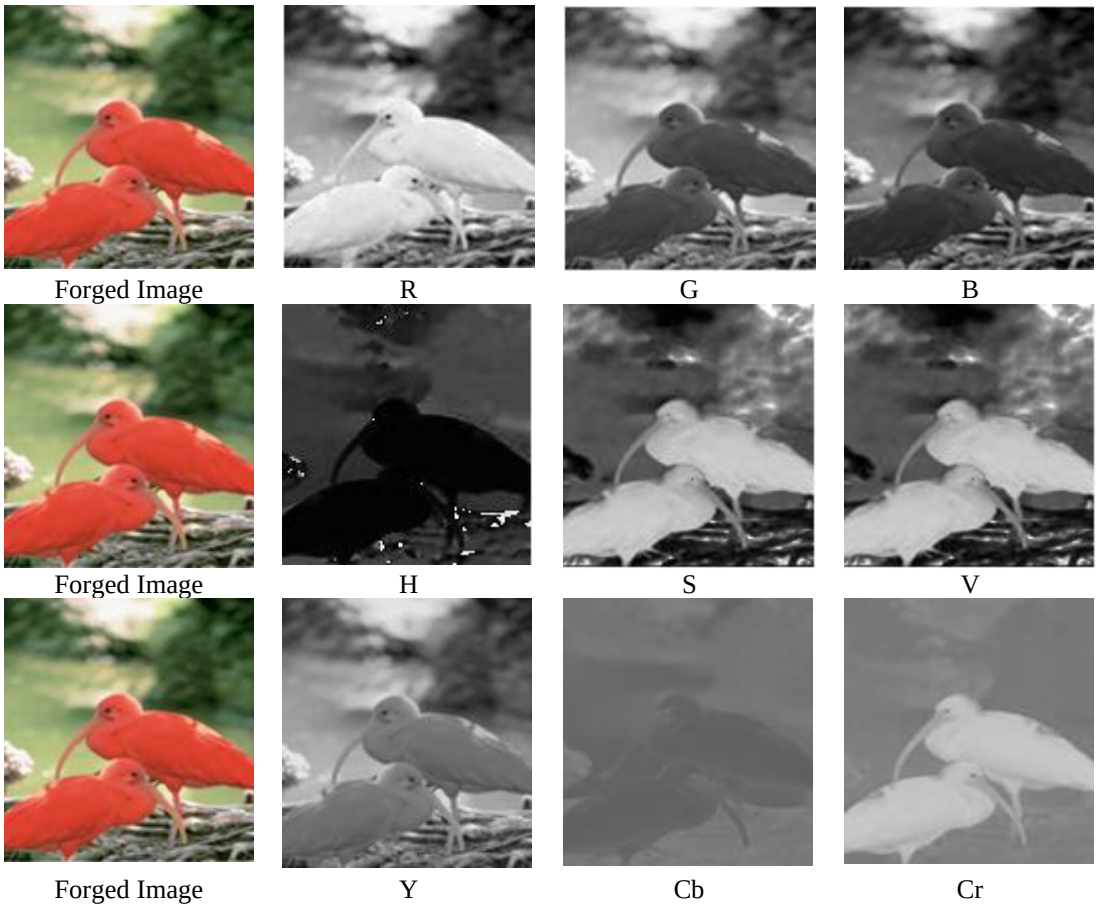


Fig. 4. Visualization of R, G, B, Y, Cb, Cr, H, S, V channels of a forged image, using RGB, YCbCr and HSV color spaces from left to right Row1 and Row2 and Row 3 respectively.

After careful visual inspection of the bird's contour in the Y, Cb and Cr components (see Figure 4), it is found that plenty of image detail cover up the forgery introduced by edges in the Y component, while in Cb (or Cr) component the bird's contour presenting the forged region is sharper than other parts of the image because Cb (or Cr) has little image content as compared to Y

(see Figure 4). Therefore, CbCr components are considered instead of Y component for features extraction.

3.2 Feature Extraction

During forgery, edges irregularities are embedded, which disturb the texture of images. Since significant difference is present in the texture of authentic and forged images in the form of small variations, the key question is how to model these small variations. LBP encodes the micro structure patterns [31], but does not capture well the orientation and edge information due to ignorance of small pixel fluctuation and sensitivity to noise. To classify whether an image is authentic or forged, microstructure patterns are required to be encoded with the strength of orientation and edge information. The new DRLBP texture descriptor better represents the microstructure patterns by assigning a weight factor (w) carrying the gradient and texture information together. In this way, DRLBP captures edge irregularities and local changes by encoding the edge and texture information together. Due to this reason, DRLBP is used in this study.

To explain how these changes are modeled an example of image forgery is explained (see Figure 5). In Figure 5a the white box region is copied and pasted into the black box region. The zoomed in view of the black box region after forgery in Figure 5c shows that the texture of the black box region after forgery is disturbed and artifacts of tampering (edges, lines, and corners etc.) are introduced. To hide these artifacts the forged image is post-processed using blurring (see Figure 5d). The zoomed-in versions of the black box region before forgery and after forgery are almost similar (see Figure 5e and Figure 5f) because tampering artifacts are invisible to human eyes after post-processing but are still present in the forged image. DRLBP texture descriptor is employed on the chrominance components of a given image to encode these structural changes due to its ability of combining edge and texture information in a single representation. The DRLBP histograms of the authentic and tampered regions are plotted in Figure 5g and Figure 5h respectively and show that the features are discriminative.

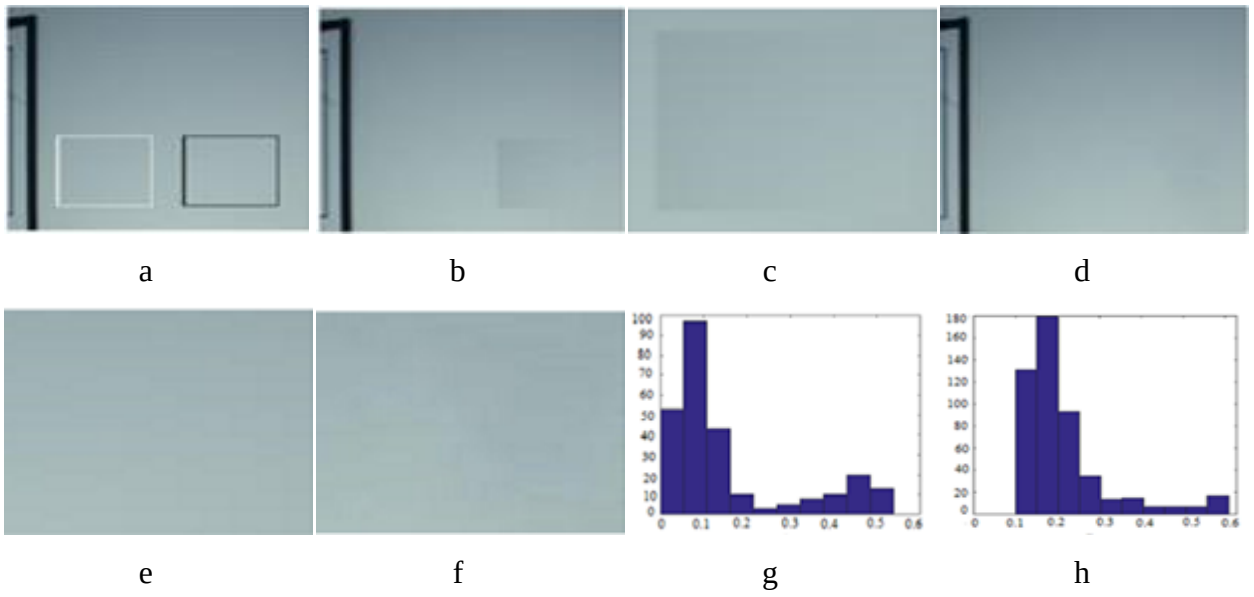


Fig. 5. a) Region to be copied (white box), Region to be forged (black box), b) Forged image, c) Zoom-in view of forged region before post-processing, d) Post-processed forged image, e) Zoom in view of region (black box) before forgery, f) Zoom in view of forged region (black box) after forgery and post-processing, g) Histogram of DRLBP descriptor of region (black box) before forgery (features value along *x*-axis and frequency along *y*-axis), h) Histogram of DRLBP descriptor of region (black box) after forgery.

3.2.1 Computation of DRLBP Descriptor

An overview of computing the DRLBP descriptor is given below, for details see [17] and [42]. DRLBP descriptor first encodes local changes in the form of LBP codes and then estimates their distribution considering the local gradient magnitude at the corresponding locations, i.e. the DRLBP descriptor encodes the local change considering the amount of change. First LBP codes with radius 1 and neighborhood 8 are calculated from the image and then the weighted histogram W_{LBP} of LBP codes is computed using the following equation:

$$W_{LBP}(i) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} w\, \delta\big(LBP_{x,y},\ i\big),\ i = 0, 1, \dots, n-1. \quad (2)$$

$$\delta(j,i) = \begin{cases} 1, & j = i \\ 0, & 0\ otherwise \end{cases}$$

where n (=2^8) is the number of LBP codes, i.e. the number of bins in the histogram, w is gradient magnitude of the pixel at location (x, y), which weights the contribution of the corresponding LBP code according to the amount of local change at the pixel and $M \times N$ is the resolution of the chrominance component. To remove the effect of the reversal in the foreground and the background, a robust weighted histogram W_{RLBP} is calculated using W_{LBP} as follows:

$$W_{RLBP}(i) = W_{LBP}(i) + W_{LBP}(2^8 - 1 - i), \quad 0 \le i < 2^7. \quad (3)$$

Further, to enhance the discriminative effect of patterns, a discriminative weighted histogram W_{DLBP} is calculated as follows:

$$W_{DLBP}(i) = |W_{LBP}(i) - W_{LBP}(2^8 - 1 - i)|,\ 0 \leq i < 2^7 . \quad (4)$$

The DRLBP is constructed by concatenating the robust LBP and the discriminative LBP as follows:

$$DRLBP = \{W_{RLBP}, W_{DLBP}\}. \quad (5)$$

After calculating DRLBP histogram from each channel $Ch \in \{Cb\ ,\ Cr\ \}$ of the given image, the DRLBP descriptor (fv) is calculated by concatenating the DRLBP histograms corresponding to channels $Ch \in \{Cb\ ,\ Cr\ \}$ as follows:

$$fv = [fv^{Cb}, fv^{Cr}]. \quad (6)$$

The descriptor fv computes the overall distribution of changes occurred due to forgery without taking into consideration their spatial locations. The incorporation of the information regarding spatial locations of patterns into fv further enhances its discriminative potential because forgery cues are of small scale and spatially localized. If features are extracted from an image, the spatial location of forgery cues may be lost. For this reason, we divide each channel of image into K blocks (sub-images), $B_1, B_2, \dots, B_K$ each of resolution $l \times m$ such that $K(l \times m) = M \times N$. The descriptor $fv\ B_i$ is computed from each block B_i and all descriptors are concatenated to form the descriptor fv^{Ch} of each channel $Ch \in \{Cb\ ,\ Cr\ \}$ i.e., $fv^{Ch} = [fv_1^{Ch}, fv_2^{Ch}, \dots, fv_K^{Ch}]$. In this way the dimension of fv for Cb or Cr is $(RLBP\ bins + DLBP\ bins) \times K$. Finally, the DRLBP descriptor representing the input image is obtained using (6). The whole process of the computation of fv is detailed in Algorithm 1.

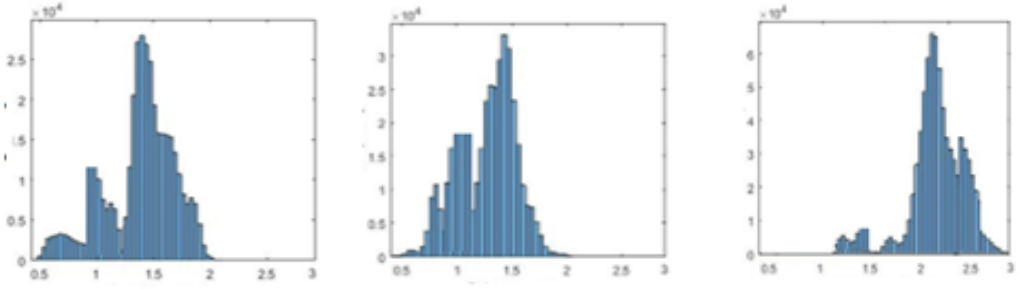
Algorithm 1: The computation of DRLBP descriptor of a given image.
Input: *RGB image I*, the number *K* of blocks
Output: *DRLBP* based feature vector fv
Procedure:

1. For a given image *I* extract chrominance components
   a. $Ch \in \{Cb\ , Cr\ \}$
2. For each $Ch \in \{Cb, Cr\ \}$
   a. Divide Ch into K blocks: $B_1, B_2, \dots, B_K$
   b. For each block $B_k, k = 1, 2, \dots, K$
      - Compute *DRLBP* histogram fv_k^{Ch}
   c. $fv^{Ch} = [fv_1^{Ch}, fv_2^{Ch}, \dots, fv_K^{Ch}]$
3. $fv = [fv^{Cb},\ fv^{Cr}]$

3.2.2 Statistical Analysis of the DRLBP Descriptor

To show that the DRLBP descriptor has the potential to discriminate authentic and tampered images, we give a statistical analysis of the descriptor in two different ways.

First, we computed the pairwise distances for the three cases using the city block between (i) authentic images, (ii) forged images, and (iii) authentic and forged images of CASIA v2.0. The cases (i) and (ii) represent intra-class distances whereas case (iii) represents inter-class distances; Figure 6 shows the histograms of the three cases. Most of the pairwise distances for the intra-class cases (Figures 6a and 6b) are between 0.0 and 2.00 while those for the inter-class case (Figure 6c) are between 1.5 and 2.5. There is an overlap of approximately 6% between pairwise distances belonging to intra-class and inter-class cases. This indicates that the DRLBP descriptor has the potential for discriminating the authentic and forged images. The effect of the overlap is reduced when a kernel SVM is used for classification because the kernel computes the distances in a higher dimensional space where the patterns become separable.



a b c

Fig. 6. a) Histograms of pairwise distances of DRLBP features of CASIA v2.0 dataset; (a) Pairwise distances within authentic class, (b) Pairwise distance within forged class, and (c) Pairwise distance between authentic and forged classes.

Secondly, the effect of DRLBP descriptor is analyzed using scatter matrix based measure because of its simplicity [43, 44]. For this purpose, two scatter matrices: (i) within class scatter matrix (WS) and (ii) between class scatter matrix (BS). WS and BS are defined as follows:

$$WS = \sum_{i=1}^{c} \sum_{j=1}^{N_i} (x_{ij} - \bar{x_i})(x_{ij} - \bar{x_i})^T, \quad (7)$$

$$BS = \sum_{i=1}^{c} N_i (\bar{x_i} - \bar{x})(\bar{x_i} - \bar{x})^T, \quad (8)$$

where x_{ij} is the feature space, N_i is the number of samples in i^{th} class, $\bar{x_i}$ is the mean vector for the i^{th} class, $\bar{x}$ is the mean vector for all classes and c is the number of classes. The traces of WS and BS represent intra-class and inter-class scatters, respectively. The features are

discriminative if the intra-class scatter is small and the inter-class scatter is high. Table 1 shows the traces of WS and BS of five datasets. In each case, the trace of BS is high and WS is small indicating that the DRLBP descriptor is discriminative (see Table 1).

Table 1. Trace of WS and BS on DRLBP features of benchmark datasets.

Dataset	Trace of WS	Trace of BS
CASIA v1.0 [45]	1.24	2.87
CASIA v2.0 [45]	1.37	2.57
CoMFoD [1]	1.48	2.21
MICC-F2000 [4]	1.28	2.01
FRITH	1.47	2.35

3.3 Classification Model Training (Building)

To identify an image as authentic or forged is a two-class problem. The process of training a classification SVM model is described in Algorithm 2.

Algorithm 2: Image Forgery Detection Model Training (Building) Procedure.
***Input*:** X_t is the set of forged/tampered images, X_a, is the set of authentic images, c and g (gamma) are the parameters of SVM with RBF kernel, $cmin$, $gmin$, $cmax$ and $gmax$ are minimum and maximum values of c (to handle misclassification) and g (to handle non-linear classification) respectively.
***Output*:** Trained classification model SVM
Procedure:

1. **for** each image I_i in X_t
   Create features vector fv_i for each tampered image using Algorithm 1
   $tfv_i = fv_i$
   Create labeled featured vector tfv_iL by assigning label 1 to tfv_i
   end for
2. **for** each image I_i in X_a
   Create features vector fv_i for each authentic image using Algorithm 1
   $afv_i = fv_i$
   Create labeled featured vector afv_iL by assigning label -1 to afv_i
   end for
3. $D \leftarrow tfv_i L \cup afv_iL$
4. $AC = 0$,
5. **for** $c = cmin\ to\ cmax$ **do**
6. **for** $g = gmin\ to\ gmax$ **do**
   Divide D into equally k folds ($k = 10$)
   for $i = 1$ to k **do**
   Train SVM(c, g) on D/F_i to get model MSVM (c, g) % all training data D except *i*th fold F_i
   Test MSVM (c, g) on fold F_i
   Record the $ACC(i)$ on fold F_i
   end for
   $AvgAC = \frac{1}{k}\sum_{i=1}^{k} ACC(i)$ % compute average accuracy on k folds
   if $AvgACC > ACC$
   $ACC = AvgACC, c_{opt} = c, g_{opt} = g$
   end if
   end for
   end for
7. Fit the SVM (c_{opt}, g_{opt}) model on training data

SVM [46], deals with two-class problems by its construction and provides better generalization among kernel based classifiers [47-49]. The SVM has a variety of kernel functions such as radial basis function (RBF), polynomial, and sigmoid kernels, etc.

Experiments are performed using these three kernels to find an optimal kernel. Experiments for identifying the optimal parameters representing the classification is performed using individual dataset or combination of datasets. A cross validation (CV) protocol is used to divide each dataset or combination of datasets into *k* (10)-folds. The SVM parameters are tuned on the training examples (9 out of the 10-folds) and that parameterization is used on the remaining (unused) fold. Each time the testing fold changes, the parameters are recalculated using *k*-1 folds on *k* iterations. Finally, the average value of *k* iterations parameters is considered final value of the trained model. All experiments are performed using the standard Lib-SVM [50], because SVM finds an optimal hyperplane with maximum margin between the two classes [46]. SVM uses the posterior probability of classification score which is the signed distance of a sample point from the decision boundary. The positive score classifies the sample point as positive, otherwise it is classified as negative [51].

3.4 Pre-trained Model Testing using Cross Dataset

Further experiments are performed to ensure the generalizability of the proposed image classification approach using the cross-dataset evaluation. In this process, the features of the test image are extracted and passed to the trained model to classify whether the image is authentic or forged.

4 Datasets and Evaluation Criteria

To build a reliable and robust image forgery detection model, training and testing on benchmark datasets is very important. We need authentic as well as forged images in datasets. Forged images should contain as many possible varieties of geometric and postprocessing operations as possible. Further, for testing a trained image forgery model on unseen images, a collection of real forged images is very important to ensure the reliability of the trained model for real practical applications. In consideration of the above facts, a description of carefully selected datasets for use in our research is given in the next subsection. Furthermore, to measure the performance of any classification model, a selection of appropriate evaluation measures is necessary. This is described in the subsection 4.2.

4.1 Datasets Description

Image forgery evaluation datasets are created using different cameras and image editing software packages. Publically available benchmark datasets: Columbia color DVMM (DVMM) [52], CASIA v1.0 [45], CASIA v2.0 [45], CoMFoD [1], UNISA [35], MICC-F220 [4] and MICC-F2000 [4] are used to evaluate and validate the proposed approach. A comprehensive experimental analysis is also performed by combining different datasets with the aim that performance may improve by increasing both variety and sample size of data. Set-A and Set-B are combination of different datasets to analyze the impact of different formats, resolutions, geometric and post-processing operations on image forgery detection. The datasets are grouped into Set-A or Set-B based on benchmarks, forgery types, postprocessing operations and number of authentic/forged images. Details of each dataset characteristics such as number of authentic and forged images, forgery types, file types, resolution, geometric and post-processing operations applied on images to make the dataset challenging are given in Table 2. An example of authentic and forged images from CSAIA v2.0, CoMFoD, UNISA and MICC-F2000 are shown in Figure 7.

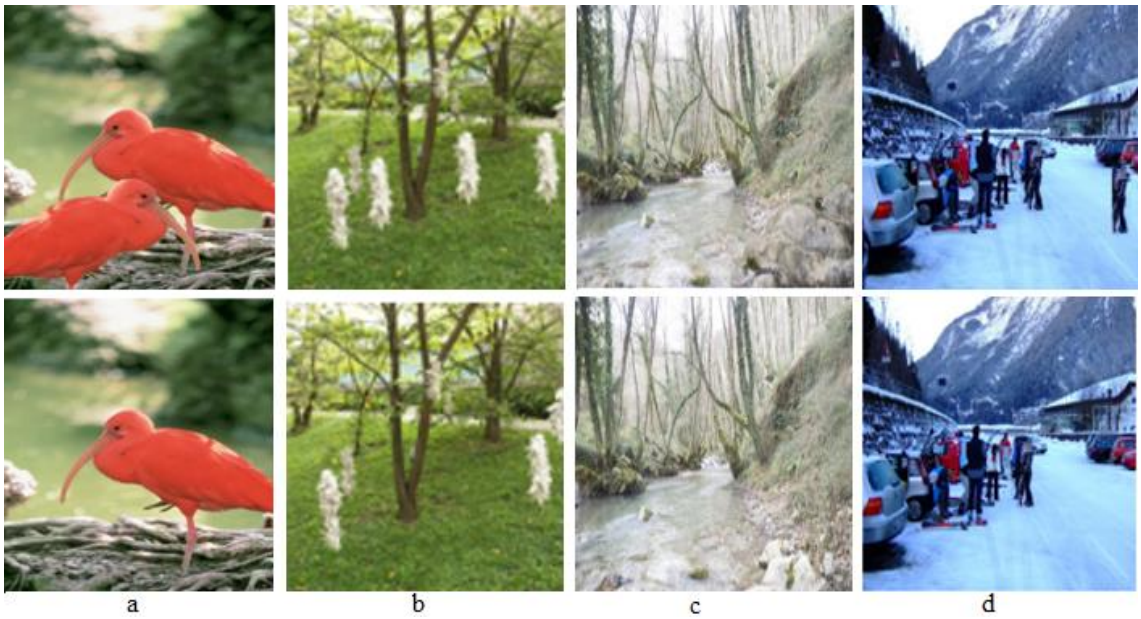


Fig. 7. Authentic (bottom), forged (top). a) CASIA v2.0: A bird object is copied, rotated, and then pasted to another location of the same image. b) CoMFoD: The white object is multiply cloned in different locations of same image. c) UNISA: In the forged image, the rock is spliced at the bottom right corner of the authentic image. d) MICC-F2000: in the forged image, an object is pasted at the right-hand side of the authentic image.

4.1.1 Forged Real Images Throughout History (FRITH) a new Dataset for Evaluation of Image Forgery Detection

In the forensic literature, many publicly available benchmarks datasets have been used for the detection of specific types of image forgeries. These benchmark datasets have been developed for supporting copy-move and splicing forgeries, having specific file formats, resolutions, geometric and post-processing operations. For example, the DVMM dataset has uncompressed authentic and forged images of sizes 757 × 568 and 1152 × 768 pixels. The CASIA v1.0 dataset has authentic and spliced images of size 384 × 256 pixels. The CoMFoD dataset has 200 sets of images of size 512 × 512, each set containing authentic and forged examples. Both authentic and forged images were post-processed to enlarge the size of dataset (10,400 post-processed authentic and forged images). MICC-F220 consists of 220 images, while MICC-F2000 contains 2000 images, all 2048 × 1536 pixels. The existing benchmark datasets have been created artificially by academic researchers with a specific goal in mind, primarily for the purposes of testing their algorithms only. However, in most cases the forgeries are fairly crude, and made by experts with the intention of forgery detection. We validated that the purpose of applying post-processing operations is to create semantically meaningful forged images. To the best of our knowledge, the applicability of existing benchmark datasets to realistic scenarios is always limited. Therefore, a benchmark dataset of semantically meaningful forged images used intentionally for false propaganda or malpractices should be made available to the researchers for the reliable testing of image forgery detection algorithms. Therefore, a new benchmark dataset has been created and labelled as Forged Real Images Throughout History (FRITH), consisting of real forgeries including many famous examples [53].

The collection of forgeries in [53] provided the starting point of creating FRITH. However, mostly [53] just contains a single image for each type of forgery, and generally does not provide the source of authentic images. For proper evaluation of image forgery detection, we require a dataset consisting of both authentic and forged image sets. Therefore, we used the forged images from [53] as queries in an internet search and selected the best quality versions of the matches to provide both the forged and their authentic samples. In total, 224 historic forged images were collected. Among these authentic (untampered) versions of 155 forged images are also obtained.

The dataset has many challenging characteristics such as: i) many images have been scanned as the originals were not digital, ii) the forged images contain a variety of image forgeries such as

copy-move and splicing forgeries by transferring objects or regions, forgery by inserting fake objects, manipulation of existing objects, forged images being post-processed using lightening effects, and image enhancement/ tuning operations. The FRITH dataset has enough variety of real copy-move and splicing forgeries, in addition to other types of forgeries such as fake objects insertion, false captioning, and image enhancement operations. In future versions, we plan to add more real forged images with their authentic ones. The dataset will be made available for the public usage and can be downloaded from http://users.cs.cf.ac.uk/Paul.Rosin/#data[1] . Example images of FRITH are shown in Figure 8 and its detail is listed in Table 2.

Fig. 8. Examples of authentic (bottom) and forged (top) images from the FRITH dataset: a) A doctored image showing Jeffrey Wong receiving an award, b) Tampered image of Obama's meeting with Iranian President Hassan Rouhani, c) The Boston Marathon bombing tampered photo showing less disturbing content, d) A digitally altered puddle of water made to appear as blood flowing from the temple of Hatshepsut in Luxor Egypt.

4.2 Evaluation Criteria

Accuracy (*ACC*), true positive rate (*TPR*), true negative rate (*TNR*), *F-Measure* and area under ROC curve (*AUC*) are widely used to evaluate image forgery detection techniques [54, 55]. We evaluate our proposed approach using *ACC*, *TPR*, *TNR*, *F-Measure*, and *AUC* with cross dataset evaluation. The evaluation measures are defined as the follows.

Accuracy (ACC)

Accuracy is the proportion of correctly predicted authentic and forged images and is defined as:

[1] A. Khurshid, S. Xianfang, R. Paul, S. Mubbashar, H. Muhammad, and H. Zulfiqar. "Forged Real Images Throughout History (FRITH) a New Dataset,"

$$ACC = \frac{(TP + TN)}{TP + TN + FN + FP} \times 100\%. \quad (9)$$

where True Positive *(TP)* is the number of tampered images, which are classified as tampered; False Negative (*FN*) is the number of tampered images, which are classified as authentic; True Negative (*TN*) is the number of authentic images, which are classified as authentic; and False Positive (*FP)* is the number of authentic images, which are classified as tampered ones.

True Positive Rate (TPR)

TPR also known as sensitivity (*SN*) is the probability of recognizing a tampered image as tampered and is computed as follows:

$$TPR = SN = \frac{TP}{TP + FN} \times 100\%. \quad (10)$$

True Negative Rate (TNR)

TNR also known as specificity (*SP*) is the probability of recognizing an authentic image as authentic and is computed as follows:

$$TNR = SP = \frac{TN}{TN + FP} \times 100\%. \quad (11)$$

F−Measure

F-measure is the harmonic mean of precision and sensitivity, and is computed as follows:

$$F - Measure = \frac{2TP}{2TP + FP + FN}. \quad (12)$$

Area under the Curve (AUC) of Receiver Operating Characteristic (ROC)

The ROC curve is used to present the performance of the binary classifier. It plots *TPR* versus *FPR* for exclusive thresholds of the classifier significances [56].

Cross Dataset Evaluation

Cross dataset evaluation (training on one dataset and testing on another dataset) is the ultimate evaluation to expose the weaknesses and ensure the robustness of any image forgery detection method. In our experimental analysis, the performance of image forgery detection is evaluated using cross dataset protocol.

Table 2. Datasets description used for evaluation of image forgery detection algorithms and cross-dataset validation.

Sr#	Dataset Name	Authentic	Forged	Forgery Types	Image Resolution	Image Formats	Geometric Operations	Post-Processing Operations
1	DVMM	183	180	Splicing	757 × 568 to 1152 × 768	TIFF and BMP	No	Uncompressed
2	CASIA v1.0	800	925	Splicing and Copy-move	384 × 256	JPEG	resize, rotation deform, and distortion	blurring, JPEG compression,
3	CASIA v2.0	7491	5123	Splicing and Copy-move	240 × 160 900 × 600	TIFF, JPEG, and BMP	resize, rotation and distortion	blurring, JPEG compression,
4	CoMFoD	5000	5000	Copy-move	512 × 512	BMP		JPEG compression, noise adding, blurring, brightness change, color reduction and contrast adjustment
5	UNISA	2000	2000	Splicing and Copy-move	4928 × 3264 6016 × 4016	TIFF, JPEG	scaling, rotation, and distortion	JPEG compression, blurring
6	FRITH	155	224	Copy-move, Splicing, Retouching, False Captioning, Fake objects insertion, Image enhancement	Variety of dimensions	JPEG, TIFF, PNG, BMP	scaling, rotation, shearing, deform and distortion	uncompressed, JPEG compression, blurring, noise adding, brightness change color reduction and image enhancement
7	MICC-F220	111	110	Copy-move	737 × 492	JPEG	scaling and rotation	JPEG compression
8	MICC-F2000	1300	700	Splicing and Copy-move	2048 × 1536	JPEG	scaling and rotation	JPEG compression blurring noise adding contrast adjustment
9	Set-A	13474	13228	Set-A is combination of DVMM, CASIA v1.0, CASIA v2.0 and CoMFoD datasets				
10	Set-B	3566	3065	Set-B is combination of MICC-F220, MICC-F2000, UNISA and FRITH datasets				

5. System Parameters

To find the best parameters of the system we performed a series of experiments by considering different combinations. Cb and Cr components are found suitable due to their better performance during experiments as also referred in methods [6, 29, 31, 41]. For calculating the DRLBP features, each component is divided into overlapped blocks with 20% overlapping rate. In the case of DRLBP, we found that the uniform (u_2) LBP (maximum two bits' transitions) with P=8 and R=1 is an appropriate choice due to its better performance as referred in [17].

The optimization of the SVM parameters was done using the training datasets, and we found that the RBF kernel had the best performance. The RBF kernel involves two parameters: c and g. The setting of these parameters plays a significant role in classification. The parameter c is used to balance the model complexity by fitting minimum error rate. The kernel function parameter g is used to determine the nonlinear mapping from the input space to the high-dimensional feature space [57]. Kernel parameters c and g, were tuned using grid-search method, and found $c = 2^5$ and $g = 2^{-5}$ best. Different k-fold cross validations (CV) such as 5-fold, 7-fold, and 10-fold were considered to best fit training data on classification model, and we found that 10-fold CV was most appropriate due to its lower sensitivity while dividing data for training and testing/validation for model fitting.

6. Experimental Results, Comparison, and Discussion

The classification accuracy of the proposed method on different datasets is presented in Figure 9.

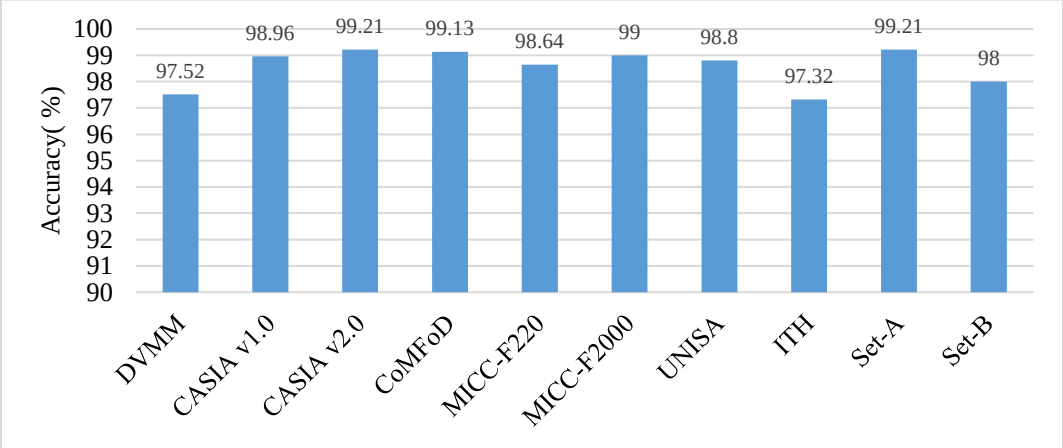


Fig. 9. Classification accuracy of proposed method on different datasets in terms of training on same dataset and testing on same dataset.

The results reported here were obtained using the optimal parameters values of the system.

6.1 Comparison of Proposed Method Terms of Training and Testing on the same Dataset.

In this section, the results of the proposed method and state-of-the-art methods are compared, in terms of training and testing on same dataset (see Table 3)

Table 3. Comparison of Proposed method with recent state-of-the-art methods in terms of training and testing on same dataset.

Train and Test Dataset	Approaches	*ACC* (%)	*TPR* (%)	*TNR* (%)	*F-Measure*	*AUC*
DVMM	**Proposed**	**97.52**	**96.67**	**98.36**	**0.97**	**0.97**
	Alahmadi et al. [58]	96.66	96.33	79.09	--	0.96
	Hussain et al. [34]	94.19	--	--	--	--
	Muhammad et al. [31]	96.39	--	--	--	--
	Rao et al. [59]	96.38	--	--	--	--
	Pham et al. [37]	96.90	--	--	--	--
	Wang et al. [38]	82.31	--	--	--	--
CASIA v1.0	**Proposed**	**98.96**	**99.03**	**98.88**	**0.99**	**0.98**
	Alahmadi et al. [58]	97.00	98.24	97.07	--	0.97
	Shen et al. [60]	97.00	--	--	--	--
	Alfy and Qureshi [6]	98.65	98.80	98.39	--	0.99
	Goh and Thing [61]	90.18	--	--	--	--
	Hussain et al. [34]	96.53	--	--	--	--
	Muhammad et al. [31]	94.89	93.91	--	--	0.93
	Rao et al. [59]	98.04	--	--	--	--
	Pham et al. [37]	96.90	--	--	--	--
CASIA v2.0	**Proposed**	**99.21**	**99.02**	**99.33**	**0.99**	**0.99**
	Cattaneo et al. [35]	90.00	--	--	--	--
	Alahmadi et al. [58]	97.50	98.45	96.84	--	0.97
	Rota et al. [33]	97.44	96.16	97.44	--	0.99
	Shen et al. [60]	98.00	--	--	--	--
	Alfy and Qureshi [6]	99.00	99.55	99.65	--	0.99
	Hussain et al. [34]	94.17	--	--	--	--
	Muhammad et al. [31]	97.33	98.50	--	--	0.97
	Rao et al. [59]	97.83	--	--	--	--
	Pham et al. [37]	96.90	--	--	--	--
MICC-F220	**Proposed**	**99.64**	**99.9**	**99.20**	**0.99**	**0.99**
	Amerini et al. [4]	--	98.21	91.84	--	--
	Wang et al. [38]	98.92	--	--	--	--
MICC-F2000	**Proposed**	**99.64**	**98.57**	**99.23**	**0.99**	**0.99**
	Amerini et al. [4]	--	93.43	89.04	--	--
	Wang et al. [38]	99.14	--	--	--	--

Table 3. . .

Train and Test Dataset	Approaches	*ACC* (%)	*TPR* (%)	*TNR* (%)	*F-Measure*	*AUC*
Set-A	**Proposed**	**99.10**	**99.20**	**99.18**	**0.99**	**0.99**
	Cattaneo et al. [35]	92.88	93.74	92.03	0.93	0.92
	Hussain et al. [34]	97.37	98.28	96.48	0.97	0.97
	Alahmadi et al. [58]	97.50	98.45	96.84	0.97	0.97
	Wang et al. [38]	98.00	98.00	98.00	0.98	0.98
Set-B	**Proposed**	**98.02**	**97.88**	**98.15**	**0.98**	**0.98**
	Cattaneo et al. [35]	89.88	88.09	91.42	0.89	0.90
	Hussain et al. [34]	96.52	96.25	96.75	0.96	0.96
	Alahmadi et al. [58]	97.50	98.45	96.84	0.97	0.97
	Wang et al. [38]	97.41	96.00	96.57	0.97	0.97

The comparison shows that the proposed method has better performance on different datasets in terms of training and testing on the same dataset, and the proposed method is robust against different geometric and post-processing operations applied on forged images of these datasets. The reason for this robustness is the ability of the DRLBP texture descriptor to model the structural changes in images that occurred due to forgery. The results of the proposed method are also comparable with the method of Yan et al., [40] which is trained using CNN architecture. The proposed method best detection accuracy on the combination of different datasets is 99.10 %, while the method of Yan et al., best detection accuracy is 86.89 %.

6.2 Comparison of Proposed Method in Terms of Training and Testing on Different Dataset.

Usually the same dataset is divided into two parts for training and testing or an n-fold strategy is applied but on the same dataset. For successful practical applications, it is necessary to develop the model through the process of training/validating on one dataset and finally testing the model on another dataset acquired from different sources, which is called cross dataset validation. For this purpose, four state-of-the-art methods [34, 35, 38, 58] are implemented together with the proposed approach.

Table 4. Comparison of the proposed method and other recent state-of-the-art methods on cross dataset evaluation Testing Dataset results are reported.

Testing Dataset	Training Dataset	Approaches	*ACC* (%)	*TPR* (%)	*TNR* (%)	*F-Measure*	*AUC*
MICC-F220	Set-A	**Proposed**	**84.16**	**86.36**	**81.98**	**0.84**	**0.84**
		Cattaneo et al., [35]	67.12	61.26	72.97	0.65	0.67
		Hussain et al., [34]	74.21	70.91	77.48	0.73	0.74
		Alahmadi et al., [58]	65.33	60.34	71.97	0.64	0.65
		Wang et al., [38]	82.61	83.63	80.89	0.81	0.81

Table 4. . .

Testing Dataset	Training Dataset	Approaches	*ACC* (%)	*TPR* (%)	*TNR* (%)	*F-Measure*	*AUC*
MICC-F2000	Set-A	**Proposed**	**86.50**	**83.33**	**88.46**	**0.81**	**0.85**
		Cattaneo et al., [35]	69.75	56.43	76.92	0.67	0.68
		Hussain et al., [34]	76.11	70.71	78.95	0.73	0.74
		Alahmadi et al., [58]	75.00	74.17	77.59	0.75	0.75
		Wang et al., [38]	83.05	84.13	84.16	0.82	0.82
UNISA	Set-A	**Proposed**	**77.46**	**85.00**	**70.00**	**0.79**	**0.77**
		Cattaneo et al., [35]	56.25	60.00	52.50	0.58	0.56
		Hussain et al., [34]	67.50	70.00	65.00	0.68	0.67
		Alahmadi et al., [58]	68.05	72.00	67.90	0.69	0.69
		Wang et al., [38]	75.34	80.20	68.29	0.75	0.75
FRITH	Set-A	**Proposed**	**74.39**	**72.55**	**77.42**	**0.78**	**0.77**
		Cattaneo et al., [35]	48.78	47.60	51.61	0.53	0.49
		Hussain et al., [34]	63.41	62.75	64.25	0.66	0.63
		Alahmadi et al., [58]	69.94	68.57	69.52	0.71	0.71
		Wang et al., [38]	72.93	71.15	75.24	0.75	0.75

A series of experiments were performed to analyze the performance of the proposed method on cross dataset testing. We trained the model on Set-A dataset and then tested it on the MICC-F220, MICC-F2000, UNISA and FRITH datasets (see Table 4).

Table 5. Comparison of the proposed method and other state-of-the-art methods on cross dataset evaluation, Testing Dataset results are reported.

Testing Dataset	Training Dataset	Approaches	*ACC* (%)	*TPR* (%)	*TNR* (%)	*F-Measure*	*AUC*
Set-A	Set-B	**Proposed**	**81.27**	**81.10**	**81.45**	**0.81**	**0.81**
		Cattaneo et al., [35]	68.54	68.25	68.83	0.68	0.68
		Hussain et al., [34]	72.29	72.03	72.54	0.72	0.72
		Alahmadi et al., [58]	73.92	73.30	73.45	0.73	0.73
		Wang et al., [38]	79.72	78.01	78.39	0.78	0.78
Set-B	Set-A	**Proposed**	**77.89**	**77.16**	**78.52**	**0.76**	**0.78**
		Cattaneo et al., [35]	66.10	60.85	70.89	0.63	0.65
		Hussain et al., [34]	69.84	67.37	71.96	0.67	0.69
		Alahmadi et al., [58]	70.29	71.13	72.54	0.71	0.71
		Wang et al., [38]	76.98	75.61	77.25	0.75	0.75

To determine the robustness of the image forgery classification model, experiments were performed by training the model on the Set-A dataset and then performing testing on the Set-B dataset and vice versa (see Table 5). The cross-dataset performance of the proposed system is better than the state-of-the-art methods, which indicates that the proposed method has better robustness.

Our work adds to previous reports using cross dataset testing, which is an important area of research and an important component in real practice where different images need to be classified. Our experiments with cross dataset testing showed that our proposed method achieved better performance than those of [34, 35, 38, 58] (see Figure 10).

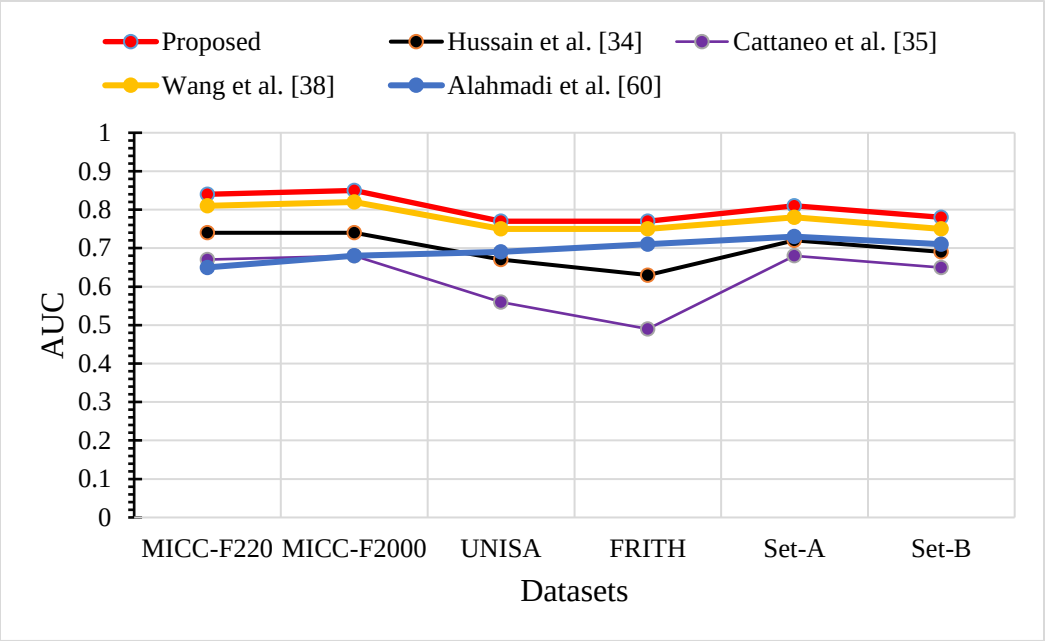


Fig. 10. AUC comparison of the Proposed Method, Hussain et al., and Cattaneo et al., Testing dataset/Training dataset (MICC-F220/Set-A, MICC-F2000/Set-A, UNISA/Set-A, FRITH/Set-A, Set-A/Set-B, and Set-B/Set-A).

6.3 Discussion

The objective of this paper was to perform a comprehensive analysis of image forgery detection algorithms and the role of datasets used to evaluate these algorithms. We introduced an edge-texture feature based approach for classifying authentic and tampered images. The novelty in our experimental analysis is that: i) we explored state-of-the-art texture descriptors and found DRLBP to be a robust texture descriptor, which models the structural changes occurred in images due to forgery using edge-texture features that incorporate information such as texture, boundary

discontinuities and inconsistencies. ii) we validated our approach and four state-of-the-art methods [34, 35, 38, 58] by performing a series of experiments on publicly available datasets. iii) we also prepared a new dataset FRITH to evaluate an image forgery detection technique on forged images used intentionally for false propaganda or malpractices rather than datasets designed specifically by academic researchers. From the experimental analysis, it is observed that success of any forgery detection system depends on: i) modeling the artifacts of forgery in a precise way; ii) training a model on samples with as many as possible different types of forgeries, geometric transformations and postprocessing operations rather than increasing the size of samples in general.

Detecting forgery that has been carried out by inserting a new object or manipulating an existing object is also a challenging task. Scene lighting and geometry parameters may help to detect such tampering. Experiments revealed that exploiting the texture of such suspected images may give a reasonable cue to detect such tampering. We recommend that there must be a large dataset containing object insertion and manipulation forgeries, such as FRITH (it has some examples of such manipulations), to ensure the robustness of an image forgery detection system in real scenarios.

Erasing manipulations disturb the structural changes occurred in images due to forgery and can be traced by exploiting the JPEG compression artifacts, if the original images were compressed after such tampering. From a forensics point of view, forgery by means of changing the lighting conditions of an image is dangerous due to their potential of concealing forgeries. For example, a splicing forgery may be concealed by changing the lightening parameters, Again JPEG compression artifacts may help to find such traces.

Image enhancement operations such as blurring (filtering), noise, contrast adjustment, are applied on forged images with the intention to remove low-level traces of forgery. We observed from experimental analysis by counting small pixel fluctuations and having texture information together with edges that it is possible to detect such traces because image enhancement operations only soften the edges, and not erase them completely.

6.3.1 Failure Analysis

The proposed method achieved good performance, however, is less effective for some cases. The proposed method and methods in [34, 35, 38, 58], failed to predict some real forgeries shown in Table 6. After analyzing the failure cases, it is found that texture and edges of the forged images contain a mix of colors from the foreground and background of the source image which still a

challenge. We will address such problems in future by exploring manipulation-relevant features using deep learning approaches.

Table 6. Example images of failure cases of forged images from FRITH dataset.

	In this image text in the passport is manipulated and the image of the passport is used to gain some illegal benefit. The person in the image altered his particulars to hide the passport contents.
	The Polish subsidiary of Microsoft ran a version of a company marketing campaign in which the photo was altered by replacing the face of middle person.
	The image was August 2007 cove of the scientific publication Nature showing three aircrafts measuring atmospheric pressure. The top and bottom aircrafts are cloned.
	In this image the Swiss tabloid Blick digitally altered a puddle of water to appear as blood flowing from the temple to show terrorist attack at the temple of Hatshepsut in Luxor Egypt.
	In the image lady Franken dressed up like a baby bunny, wearing adult diapers and clutching a fluffy white teddy bear is fake.

7. Conclusion and Future Work

In this paper, a novel image forgery detection method based on DRLBP and SVM has been proposed. The chrominance components of an input image are divided into overlapping blocks, and then the DRLBP code of each block is calculated. Later, histograms of all the blocks of both Cb and Cr components are used as features. Classification is performed using an SVM. The method was extensively evaluated on individual and combined benchmark datasets in terms of training and testing on splits of the same dataset, and on different datasets (i.e., cross dataset validation).

The proposed method was evaluated using eight benchmark datasets: DVMM, CASIA v1.0, CASIA v2.0, CoMFoD, MICC-F220, MICC-F2000, UNISA, FRITH, Set-A and Set-B. The classification accuracy of our method is consistent across the eight datasets, and it has better performance than state-of-the-art methods due to the effective modeling structural changes occurring in tampered images by DRLBP texture descriptor. The results on combinations (Set-A and Set-B) of datasets indicate that the proposed method is robust and consistent under different post-processing operations, file types, and image resolutions (small, medium, and high). The cross-dataset evaluation (training on one dataset and testing on another dataset) shows that the performance of the proposed method is significantly better than state-of-the-art methods. DRLBP is an elegant texture descriptor to represent important features of image tampering and helps in classifying whether an image is tampered or authentic. Furthermore, the approach is robust against different geometric transformations and post-processing operations. Cross-dataset evaluation is the ultimate test to expose the weaknesses and robustness of any image forgery detection method.

The results of this study are better than other state-of-the-art image forgery detection methods in terms of cross-dataset validation, there is still room for improving the approach to ensure the robustness of an image forgery detection method on unseen images. It is believed that the research community should adopt the cross-dataset validation procedure from now on. From the experimental analysis, it is considered that statistical artifacts of possible types of image forgeries must be presented by benchmark datasets to enable the development off a robust model. As future work it is planned to localize the tampered regions and tune the parameters using meta-heuristics methods to improve the cross-dataset validation performance. dynamic learning of the classification method when tested on unseen images is another plan for future research.

Acknowledgement

The authors are grateful to Higher Education Commission (HEC) Pakistan for research grant under International Research Support Initiative Program (IRSIP) vide grant no. 1-8/HEC/HRD/2017/6950 at School of Computer Science and Informatics, Cardiff University, UK.

References

1 D. Tralic, I. Zupancic, S. Grgic, and M. Grgic, "CoMoFoD—New database for copy-move forgery detection", In Proceedings of *55th ELMAR International Symposium,* Zadar, Croatia, 2013, pp. 49-54.

2 M. Gryka, M. Terry, and G. J. Brostow, "Learning to remove soft shadows," *ACM Transactions on Graphics (TOG),* vol. 34, no. 5, pp. 153-167, 2015.

3 K. Karsch, K. Sunkavalli, S. Hadap, N. Carr, H. Jin, R. Fonte, M. Sittig, and D. Forsyth, "Automatic scene inference for 3D object compositing," *ACM Transactions on Graphics (TOG),* vol. 33, no. 3, pp. 32, 2014.

4 I. Amerini, L. Ballan, R. Caldelli, A. Del 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, 2011.

5 E. S. Gastal, and M. M. Oliveira, "High‐Order Recursive Filtering of Non‐Uniformly Sampled Signals for Image and Video Processing," *EUROGRAPHICS,* vol. 34, no. 2, pp. 81-93, 2015.

6 E.-S. M. El-Alfy, and M. A. Qureshi, "Robust content authentication of gray and color images using lbp-dct markov-based features," *Multimedia Tools and Applications,* vol. 76, no. 12, pp. 1-22, 2016.

7 V. Schetinger, M. Iuliani, A. Piva, and M. M. Oliveira, "Digital image forensics vs. image composition: An indirect arms race," *arXiv preprint arXiv:1601.03239,* 2016.

8 J. Liao, R. S. Lima, D. Nehab, H. Hoppe, P. V. Sander, and J. Yu, "Automating image morphing using structural similarity on a halfway domain," *ACM Transactions on Graphics (TOG),* vol. 33, no. 5, pp. 168, 2014.

9 A. Khurshid, H. Zulfiqar, and H. Muhammad, "Copy-move and splicing image forgery detection and localization techniques: a review," *Australian Journal of Forensic Sciences,* vol. 49, no. 3, pp. 281-307, 2017.

10 S. Xue, A. Agarwala, J. Dorsey, and H. Rushmeier, "Understanding and improving the realism of image composites," *ACM Transactions on Graphics (TOG),* vol. 31, no. 4, pp. 84(1)-84(10), 2012.

11 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,* pp. 262-282, 2017.

12 G. K. Birajdar, and V. H. Mankar, "Digital image forgery detection using passive techniques: A survey," *Digital Investigation,* vol. 10, no. 3, pp. 226-245, 2013.

13 J. Kamenicky, M. Bartos, J. Flusser, B. Mahdian, J. Kotera, A. Novozamsky, S. Saic, F. Sroubek, M. Sorel, and A. Zita, "PIZZARO: Forensic analysis and restoration of image and video data," *Forensic Science International,* vol. 264, pp. 153-166, 2016.

14 R. Pandey, S. Singh, and K. Shukla, "Passive forensics in image and video using noise features: A review," *Digital Investigation,* vol. 19, no. 1, pp. 1-28, 2016.

15 J. A. Redi, W. Taktak, and J.-L. Dugelay, "Digital image forensics: a booklet for beginners," *Multimedia Tools and Applications,* vol. 51, no. 1, pp. 133-162, 2011.

16 T. Qazi, K. Hayat, S. U. Khan, S. A. Madani, I. A. Khan, J. Kolodziej, H. Li, W. Lin, K. C. Yow, and C. Z. Xu, "Survey on blind image forgery detection," *IET Image Processing,* vol. 7, no. 7, pp. 660-670, 2013.

17 A. Satpathy, X. Jiang, and H. L. Eng, "LBP-based edge-texture features for object recognition," *IEEE Transactions on Image Processing,* vol. 23, no. 5, pp. 1953-1964, 2014.

18 R. Chamlawi, A. Khan, and I. Usman, "Authentication and recovery of images using multiple watermarks," *Computers & Electrical Engineering,* vol. 36, no. 3, pp. 578-584, 2010.

19 T.-Y. Lee, and S. D. Lin, "Dual watermark for image tamper detection and recovery," *Pattern Recognition,* vol. 41, no. 11, pp. 3497-3506, 2008.

20 I. Prathap, V. Natarajan, and R. Anitha, "Hybrid robust watermarking for color images," *Computers & Electrical Engineering,* vol. 40, no. 3, pp. 920-930, 2014.

21 O. M. Al-Qershi, and B. E. Khoo, "Passive detection of copy-move forgery in digital images: State-of-the-art," *Forensic Science International,* vol. 231, no. 1, pp. 284-295, 2013.

22 P. Korus, "Digital image integrity–a survey of protection and verification techniques," *Digital Signal Processing,* vol. 71, no. 5, pp. 1-26, 2017.

23 M. Hussain, A. W. A. Wahab, Y. I. B. Idris, A. T. Ho, and K.-H. Jung, "Image steganography in spatial domain: A survey," *Signal Processing: Image Communication,* vol. 65, pp. 46-66, 2018.

24 H. Farid, *Detecting digital forgeries using bispectral analysis,* Technical Report AIM-1657, AI Lab. Massachusetts Institute of Technology, Cambridge, USA, 1999.

25 T. Ng, and S. Chang, "A model for image splicing", In Proceedings of *International conference on image processing* Singapore, 2004, pp. 1169-1172.

26 T. T. Ng, S. F. Chang, and Q. Sun, "Blind detection of photomontage using higher order statistics", In Proceedings of *international symposium on circuits and systems*, Vancouver, Canada, 2004, pp. 688-691.

27 T. T. Ng, S. F. Chang, and Q. Sun, "A data set of authentic and spliced image blocks," *Columbia University, ADVENT Tech. Rep.*, pp. 203-204, 2004.

28 W. Wang, J. Dong, and T. Tan, "Effective image splicing detection based on image chroma", In Proceedings of *16th IEEE international conference on image processing*, Cairo, Egypt, 2009, pp. 1257-1260.

29 W. Wang, J. Dong, and T. Tan, "Image tampering detection based on stationary distribution of Markov Chain", In Proceedings of *17th IEEE international conference on image processing* Hong Kong, 2010, pp. 2101-2104.

30 X. Zhao, J. Li, S. Li, and S. Wang, "Detecting digital image splicing in chroma spaces", In Proceedings of *International workshop on digital watermarking*, Berlin, Germany, 2010, pp. 12-22.

31 G. Muhammad, M. 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.

32 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, 2015.

33 P. Rota, E. Sangineto, V. Conotter, and C. Pramerdorfer, "Bad teacher or unruly student: Can deep learning say something in image forensics analysis?", In Proceedings of *23rd international conference on pattern recognition*, Cancún, Mexico, 2016, pp. 2503-2508.

34 M. Hussain, S. Qasem, G. Bebis, G. Muhammad, H. Aboalsamh, and H. Mathkour, "Evaluation of image forgery detection using multi-scale weber local descriptors," *International Journal on Artificial Intelligence Tools*, vol. 24, no. 4, pp. 1-28, 2015.

35 G. Cattaneo, G. Roscigno, and U. F. Petrillo, "Improving the experimental analysis of tampered image detection algorithms for biometric systems," *Pattern Recognition Letters*, vol. 113, no. 1, pp. 93-101, 2017.

36 Z. Lin, J. He, X. Tang, and C.-K. Tang, “Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis,” *Pattern Recognition,* vol. 42, no. 11, pp. 2492-2501, 2009.

37 N. T. Pham, J.-W. Lee, G.-R. Kwon, and C.-S. Park, “Efficient image splicing detection algorithm based on markov features,” *Multimedia Tools and Applications*, pp. 1-15, 2018.

38 L. Wang, and S.-i. Kamata, "Forgery image detection via mask filter banks based CNN", In Proceedings of *10th International Conference on Graphics and Image Processing*, Chengdu, China, 2019, pp. 1-6.

39 K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", In Proceedings of *IEEE conference on computer vision and pattern recognition*, Las Vegas, Nevada, USA, 2016, pp. 770-778.

40 Y. Yan, W. Ren, and X. Cao, “Recolored image detection via a deep discriminative model,” *IEEE Transactions on Information Forensics and Security,* vol. 14, no. 1, pp. 5-17, 2019.

41 X. Zhao, S. Li, S. Wang, J. Li, and K. Yang, “Optimal chroma-like channel design for passive color image splicing detection,” *EURASIP Journal on Advances in Signal Processing,* vol. 2012, no. 1, pp. 1-11, 2012.

42 N. Dalal, and B. Triggs, "Histograms of oriented gradients for human detection", In Proceedings of *IEEE conference on computer vision and pattern recognition*, San Diego, CA, USA, 2005, pp. 886-893.

43 K. Fukunaga, *Introduction to statistical pattern recognition*: Elsevier, 2013.

44 A. R. Webb, *Statistical pattern recognition*: John Wiley & Sons, 2003.

45 J. Dong, W. Wang, and T. Tan, "CASIA image tampering detection evaluation database", In Proceedings of *IEEE China summit & international conference on signal and information processing* Xi'an, China, 2013, pp. 422-426.

46 C. Cortes, and V. Vapnik, “Support-vector networks,” *Machine learning,* vol. 20, no. 3, pp. 273-297, 1995.

47 V. Vapnik, *The nature of statistical learning theory*: Springer science & business media, 2013.

48 N. Cristianini, and J. Shawe Taylor, *An introduction to support vector machines and other kernel-based learning methods*, USA: Cambridge University Press, 2000.

49 C. W. Hsu, and C. J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415-425, 2002.

50 C. C. Chang, and C. J. Lin, “LIBSVM: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 27:1-27:10, 2011.

51 J. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61-74, 1999.

52 T. T. Ng, J. Hsu, and S. Chang, "Columbia image splicing detection evaluation dataset," 2009.

53 "Photo tampering through out history," 2017; http://pth.izitru.com/.

54 C. Richao, Y. Gaobo, and Z. Ningbo, “Detection of object-based manipulation by the statistical features of object contour,” *Forensic science international*, vol. 236, pp. 164-169, 2014.

55 L. Su, T. Huang, and J. Yang, “A video forgery detection algorithm based on compressive sensing,” *Multimedia Tools and Applications*, vol. 74, no. 17, pp. 1-16, 2014.

56 M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation", In Proceedings of *Australasian joint conference on artificial intelligence*, Berlin, Germany 2006, pp. 1015-1021.

57 M. Hussain, S. K. Wajid, A. Elzaart, and M. Berbar, "A comparison of SVM kernel functions for breast cancer detection", In Proceedings of *International Conference on Computer Graphics, Imaging and Visualization (CGIV)*, Singapore, Singapore, 2011, pp. 145-150.

58 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.

59 Y. Rao, and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images", In Proceedings of *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2016, pp. 1-6.

60 X. Shen, Z. Shi, and H. Chen, “Splicing image forgery detection using textural features based on the grey level co-occurrence matrices,” *IET Image Processing*, vol. 11, no. 1, pp. 44-53, 2016.

61 J. Goh, and V. L. Thing, “A hybrid evolutionary algorithm for feature and ensemble selection in image tampering detection,” *International Journal of Electronic Security and Digital Forensics,* vol. 7, no. 1, pp. 76-104, 2015.