Expert Systems With Applications 182 (2021) 115282

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A fast technique to detect copy-move image forgery with reflection and non-affine transformation attacks

Anuja Dixit *, Soumen Bag

Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad 826004, India

ARTICLE INFO

Keywords:
Center Surround Extrema detector
Copy-move image forgery
Digital image forensics
Fuzzy C-Means clustering
Local Image Permutation Interval Descriptor
Passive authentication

ABSTRACT

Copy-move forgery is one of the most frequently utilized image tampering technique which uses the segment of the same image to produce manipulated image by duplicating or concealing image regions. To remove suspicious traces of forgery, various attacks are applied over the tampered image which make forgery detection process too complicated. We propose a forgery detection technique in which Center Surround Extrema (CenSurE) detector is applied for keypoint detection from images. To compute keypoint descriptors, Local Image Permutation Interval Descriptor (LIPID) is used. Keypoint matching is performed using k-Nearest Neighbor (k-NN) technique with utilization of k-d tree and Best-Bin-First (BBF) search method. Grouping over keypoints is performed using Fuzzy C-Means (FCM) clustering. We apply Random Sample Consensus (RANSAC) algorithm to remove outliers obtained during forgery detection process. Experimental results show that proposed technique can effectively detect forged images containing reflection and non-affine transformation with geometrical attacks. In addition, proposed approach also shows robustness against erosion, dilation, RGB color addition, zoom motion blur, JPEG compression, spread noise addition, and multiple copy-move attacks. Proposed scheme consumes least time in forgery detection as compared to state-of-the-art methods.

1. Introduction

Images have their applications in various prospects such as news channels, magazines, newspapers, medical probes, courts of law, forensic investigation, etc. With growing development in multimedia technology and multimedia processing tools, various image processing softwares are available. Some of the most commonly used image editing softwares are Photoshop, GNU Image Manipulation Program (GIMP), Fireworks, and Inkscape. Due to promiscuous availability of such powerful image editing tools, digital images can easily be tampered without leaving any trace of forgery which is a serious threat to society as manipulated images can distort the truth. To tackle authentication problem in digital images, various techniques have been developed. In digital image forensics (Uliyan, Jalab, Wahab, Shivakumara, & Sadeghi, 2016), image forgery detection methods are broadly classified into two categories as active and passive methods. The active method adopts use of digital watermark and digital signature to benchmark the legitimacy of image. Watermark or signature is embedded into digital images to protect their integrity and further images are verified by extracting embedded information. However, active methods impose the requirement of pre-embedded information within images which is done by utilizing special scientific methods. Such necessity of pre-requisite operations has resulted in limited availabilty of images with digital signature or watermark. In opposite, passive approaches (Jeronymo, Borges, & dos Santos Coelho, 2017) do not impose the requirement of any prior information, therefore these approaches are also addressed as blind authentication methods. Due to practicality and broad pertinency of passive approaches, they are popular area of research. Passive approaches can detect image splicing (Yildirim & Ulutaş, 2019), image resampling, and copy-move image forgery present in digital images.

Copy-move forgery (Guo, Liu, & Wu, 2013) is one of the most popular image forgery technique. In this forgery technique, a segment from image is copied and pasted at different location on the same image as shown in Fig. 1, hence copy-move forgery is also known as cloning or region duplication forgery. This forgery technique is used to over-emphasize the image information or to conceal the image segments by pasting region copied from the same image over them. To remove traces of forgery, various geometrical (rotation, scaling, etc.) and post-processing (blurring, JPEG compression, noise addition, etc.) operations are applied over the image which make forgery detection process

* Corresponding author.
E-mail addresses: anu2010cse1@gmail.com (A. Dixit), soumen@iitism.ac.in (S. Bag).

https://doi.org/10.1016/j.eswa.2021.115282
Received 7 June 2020; Received in revised form 30 March 2021; Accepted 22 May 2021
Available online 1 June 2021
0957-4174/© 2021 Elsevier Ltd. All rights reserved.

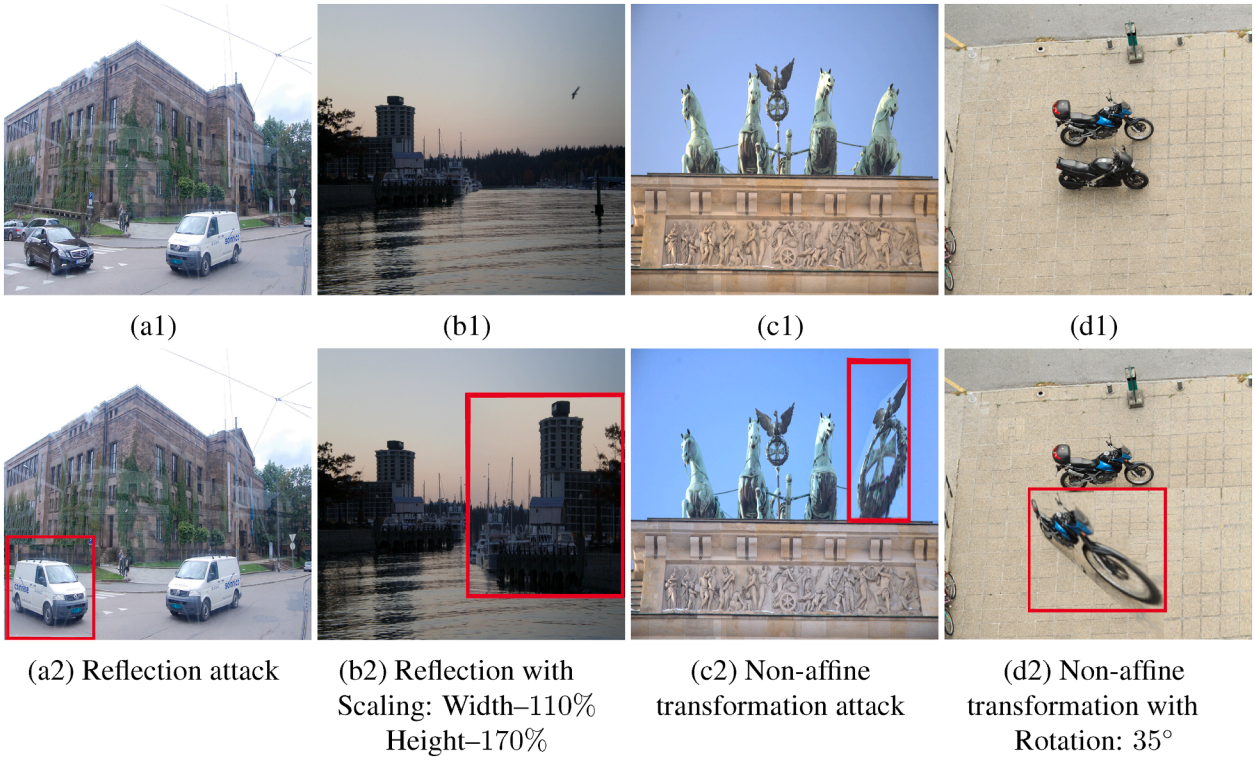

Fig. 1. {(a1), (b1)} Authentic images collected from MICC-F2000 (Amerini et al., 2011a) dataset. {(c1), (d1)} Authentic images taken from IMD (Christlein et al., 2012b) and CoMoFoD (Tralic et al., 2013a) dataset, respectively. {(a2), (b2)} Copy-move forged images manipulated using reflection and reflection with scaling attack, respectively. {(c2), (d2)} Copy-move forged images tampered using non-affine transformation and non-affine transformation with rotation attack, respectively. Forged regions are outlined with red color rectangle.

highly complex due to alterations in image features. In recent years, versatile copy-move forgery detection techniques have been developed. Primarily, copy-move forgery detection methods are classified into two categories, namely block-based and keypoint-based approaches which are discussed in the following subsections.

1.1. Block-based techniques

In block-based methods, an image is splitted into overlapping blocks of fixed dimension. Feature vectors are computed corresponding to each block and these vectors are matched to identify tampered images. To detect forged images using block-based method, Fridrich, Soukal, and Lukáš (2003) used Discrete Cosine Transform (DCT) to extract block features followed by lexicographical sorting for feature matching. Mahdian and Saic (2007) proposed a technique based on blur moment invariants to extract block features. Bashar, Noda, Ohnishi, and Mori (2010) used Discrete Wavelet Transform (DWT) with Kernel Principal Component Analysis (KPCA) to detect forged regions. Christlein, Riess, and Angelopoulou (2010) proposed a novel method using Same Affine Transformation Selection (SATS) technique. Their method attained rotation invariancy. Bravo-Solorio and Nandi (2011) used color-dependent features to reduce the number of comparisions involved in searching step. They utilized 1-D descriptor to represent feature vector for each block of the image. Ryu, Kirchner, Lee, and Lee (2013) utilized Zernike moments for block feature extraction. They applied Locality Sensitive Hashing (LSH) to perform efficient block matching. Li (2013) proposed a method based on Polar Cosine Transform (PCT) to detect image forgery as shown in Table 1. Cozzolino, Poggi, and Verdoliva (2015) proposed a method based on Circular Harmonic Transform (CHT) to extract image features. They used PatchMatch algorithm to search for approximate nearest neighbor among image patches in matching stage. Mahmood, Irtaza, Mehmood, and Mahmood (2017) applied Stationary Wavelet Transform (SWT) over input image. They utilized circular blocks to divide LL subband obtained using SWT. Further, they computed feature vectors corresponding to circular blocks using Local Binary Pattern Variance (LBPV). Dixit and Bag (2019) used Histogram of Oriented Gradients (HOG) with Singular values Decomposition (SVD) over image blocks to achieve feature descriptors. To detect forged images sustaining multiple copy-move with contrast adjustment, blurring, and color reduction attacks, Dixit and Bag (2020b) applied Gray-level Co-occurrence Matrix (GLCM), SVD, and LBP techniques for feature extraction from image blocks. Dixit and Bag (2020c) improved their work using edge-based techniques as preprocessing operation. They applied Weber Local Descriptor (WLD) with HOG for feature extraction from images blocks. Their method obtained robustness against plain as well as multiple copy-move attacks for images belonging to varied datasets as shown in Table 1.

1.2. Keypoint-based techniques

The block-based methods can adequately detect plain copy-move forgery in digital images but they have some limitations. Block-based methods cannot detect forged images sustaining significant extent of geometrical attacks as well as require high computational cost as compared to keypoint-based methods due to feature vector computation and matching of large number of image blocks. In keypoint-based methods, keypoints are extracted and matched to detect forgery present in images. Pan and Lyu (2010) used Scale Invariant Feature Transform (SIFT) for keypoint extraction. Further, they performed matching of keypoints using approximate nearest neighbor search algorithm. They utilized affine transformation to identify forged regions in image. Amerini, Ballan, Caldelli, Del Bimbo, and Serra (2011b) proposed a method which included feature extraction using SIFT algorithm. They clustered keypoints using Agglomerative Hierarchical Clustering (AHC). Later, they improved their method by using J-Linkage clustering. Chen, Lu, Ni, Sun, and Huang (2013) used Harris corner points for

Table 1
Block-based techniques for copy-move image forgery detection.

Author	Preprocessing	Feature extraction technique	Maching technique	Feature vector length	Dataset	Attacks addressed
Fridrich et al. (2003)	RGB to Gray scale, Division into overlapping blocks	DCT	Lexicographical sorting, Shift vector computation	64	Self constructed	Plain copy-move
Mahdian and Saic (2007)	Division into overlapping blocks	BLUR moment	k-d tree, Euclidean distance	24	Self constructed	AWGN, JPEG compression, Blurring, Rotation
Bashar et al. (2010)	RGB to Gray scale, Division into overlapping blocks	DWT and KPCA	Lexicographical sorting, Offset computation	10	Self constucted	Noise addition, JPEG compression
Christlein et al. (2010)	Division into circular blocks	Zernike moment, average color information, and mean intesity of circular blocks	Same affine transformation selection	–	IMD	Roation, Scaling
Bravo-Solorio and Nandi (2011)	Division into non-overlapping blocks	Color dependent features	Lexicographical sorting, Euclidean distance	4	Self constructed	Reflection, Rotation, Scaling
Ryu et al. (2013)	RGB to Gray scale, Division into circular blocks	Zernike moment	Locality sensitive hashing	12	Self constucted	AWGN, JPEG compression, Blurring, Rotation
Li (2013)	Division into overlapping blocks	PCT	Locality sensitive hashing, Approximate nearest neighbor search	–	ImageNet, Kodak lossless true color image	Rotation, JPEG compression, Gaussian noise, Blurring, Plain copy-move
Cozzolino et al. (2015)	–	Zernike, PCT, FMT	Patchmatch algorithm	Zernike–12, PCT–10, FMT–25	GRIP, IMD	Rotation, JPEG compression, Noise addition, Scaling
Mahmood et al. (2017)	RGB to Gray scale, Division into circular blocks	SWT and LBP variance	Lexicographical sorting, Euclidean distance	10	CoMoFoD, Kodak lossless true color image	Translation, Flipping, Blurring, Rotation, Scaling, Color reduction, Brightness change, Multiple copy-move
Dixit and Bag (2019)	RGB to Gray scale conversion, Division into overlapping blocks	HOG, SVD	Lexicographical sorting, Euclidean distance, Shift vector	12	CoMoFoD	Multiple copy-move with Contrast adjustment, Blurring, Color reduction
Dixit and Bag (2020b)	RGB to Gray scale conversion, Division into overlapping blocks	GLCM, SVD, LBP	Lexicographical sorting, Euclidean distance	67	CoMoFoD	Contrast adjustment, Blurring, Color reduction, Brightness change
Dixit and Bag (2020c)	RGB to Gray scale conversion, Edge detection, Division into overlapping blocks	WLD, HOG	Radix sorting, Lexicographical sorting, Correlation computation	100	CoMoFoD, CMFD_grip, CMH, Ardizzone, IMD_multipaste	Plain copy-move, Multiple copy-move, Contrast adjustment, Color reduction, Blurring, Brightness change

Table 2
Keypoint-based techniques for copy-move image forgery detection.

Author	Preprocessing	Feature extraction technique	Maching technique	Clustering technique	Feature vector length	Dataset	Attacks addressed
Pan and Lyu (2010)	RGB to Gray scale	SIFT	Best-Bin-First	K-means clustering	128	Kodak	Illumination adjustment, Rotation, Scaling, Free form copy-move
Amerini et al. (2011b)	–	SIFT	2NN	AHC	128	MICC-F220, MICC-F2000	JPEG compression, Noise addition, Rotation, Scaling
Chen et al. (2013)	RGB to Gray scale	Harris corner point with step sector statistics	Best-Bin-First	–	72	Self constructed forged images using Kodak dataset	Rotation, Scaling, Flipping, JPEG compression
Li et al. (2014)	Image segmentation using SLIC	SIFT	k-d tree with k-nearest neighbor	Expectation maximization-based algorithm	128	MICC-F600 with self constructed dataset	Rotation, Scaling, JPEG compression, Noise addition
Pun et al. (2015)	Adaptive oversegmenation using SLIC	SIFT	Adaptive block feature mapping	–	–	IMD dataset	Rotation, Scaling, JPEG compression
Silva et al. (2015)	RGB to HSV color channel	SURF	Nearest neighbor distance ratio	Use of spatial proximity and correspondence angle	64	CMH, CMEN	Roation, Scaling, JPEG compression
Ardizzone et al. (2015b)	Image division into triangles	SIFT, SURF, HARRIS Corners	Triangle matching by color, angle	–	–	Ardizzone dataset	Rotation, Scaling
Li et al. (2017)	–	MSCR with Zernike moment	N-best feature matching	Heirarchical clustering algorithm	12	MICC-F600	Multiple copy-move
Yang et al. (2017)	–	KAZE with SIFT	g2NN	–	SIFT-128, KAZE-64	IMD	Plain copy-move, Rotation, Scaling, Noise addition, JPEG compression
Warif et al. (2017)	–	SIFT, Mirror SIFT	g2NN	AHC	128	NB-CASIA	Reflection, Rotation, Scaling, Reflection with rotation, Reflection with scaling
Jin and Wan (2017)	–	SIFT with non-maximum value suppression	g2NN	J-linkage clustering	–	GRIP, FAU, MICC-F600	Rotation, Scaling
Wang et al. (2018b)	–	SURF, QPCET	Reversed generalised 2NN	–	–	FAU, GRIP	Roation, Scaling, JPEG compression, Noise addition
Dixit and Bag (2020a)	RGB to Gray scale	FAST, AKAZE, SIFT, DAISY, PCA	Nearest neighbor search	Hierarchical DBSCAN	64	Ardizzone, CASIA V2.0, USC_SIPI	Composite geometrical and post-processing attacks

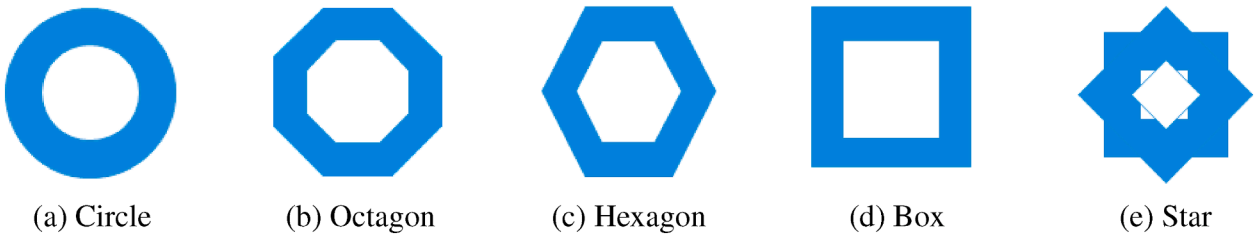

Fig. 2. Center-Surround bi-level filters with succeeding progression.

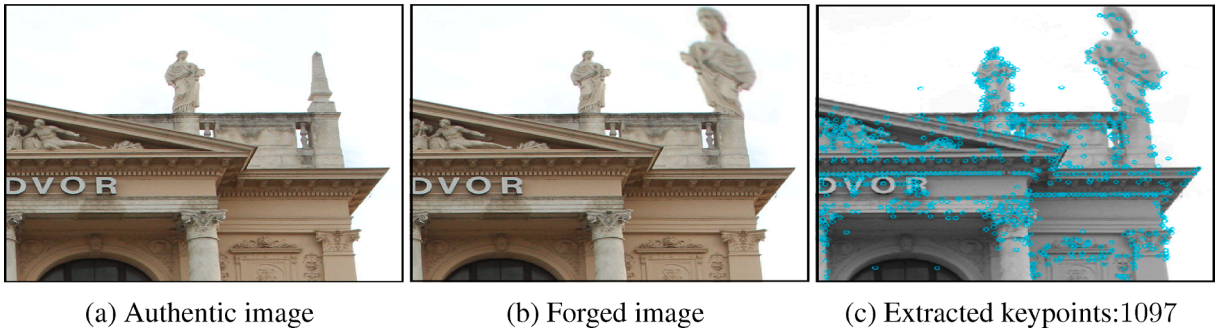

Fig. 3. (a) Original image taken from CoMoFoD (Tralic et al., 2013a) dataset. (b) Copy-move forged image sustaining duplicated region which is manipulated using reflection and non-affine transformation attack. (c) Keypoint extraction using CenSurE detector.

keypoint detection. To compute keypoint descriptors, they utilized step sector statistics. Li, Li, Yang, and Sun (2014) proposed a segmentation-based approach. They segmented the image into meaningful irregular superpixels using Simple Linear Iterative Clustering (SLIC). Further, they employed Expectation–Maximization-based algorithm for estimation of transformation matrix. Pun, Yuan, and Bi (2015) proposed a method based on adaptive oversegmentation. They utilized SIFT algorithm for keypoint extraction. Silva, Carvalho, Ferreira, and Rocha (2015) used Speeded Up Robust Features (SURF) to extract feature keypoints. Further, they employed muti-scale analysis with voting process to detect forgery. Ardizzone, Bruno, and Mazzola (2015b) proposed a method based on comparision of triangles for forgery detection. Li, Yang, Lu, and Sun (2017) applied Maximally Stable Color Region (MSCR) technique for feature detection as shown in Table 2. They computed feature descriptor using Zernike moments. They empoyed Generalised 2 Nearest Neighbor (g2NN) with affine transformation estimation for forgery detection. Yang, Li, Lu, and Weng (2017) used KAZE with SIFT for keypoint extraction as well as for keypoint descriptor computation. Warif, Wahab, Idris, Salleh, and Othman (2017) proposed a SIFT-Symmetry based novel method for detection of forged images bearing reflection attack. Jin and Wan (2017) proposed a novel approach using non-maximum value suppression algorithm for selection of keypoints. They employed J-linkage clustering to identify forged region in image. Wang, Jiao, Wang, Yang, and Niu (2018b) utilized SURF for keypoint detection. They computed keypoint descriptors using Quaternion Polar Complex Exponential Transform (QPCET). They performed matching using triangular meshes. Dixit and Bag (2020a) applied Accelerated KAZE (AKAZE) with Features from Accelerated Segment Test (FAST) for extraction of keypoints. They employed correlation-based nearest neighbour search technique with Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) for detection of copy-move forged images sustaining composite geometrical and post-processing attacks.

1.3. Our contribution

We propose a novel keypoint-based copy-move image forgery detection technique. In our method, Center Surround Extrema (CenSurE) detector is employed for keypoint extraction. To compute keypoint descriptor, Local Image Permutation Interval Descriptor (LIPID) is utilized. k-Nearest Neighbor (k-NN) matching is performed using k-d tree with Best-Bin-First (BBF) search technique to identify keypoints with similar feature descriptors. To identify forged region within image, keypoint clustering is performed using Fuzzy C-Means (FCM) clustering algorithm. Random Sample Consensus (RANSAC) algorithm is used to remove outliers obtained during forgery detection process. The main contributions of the proposed algorithm can be summed up in terms of the following advantages:

- There are only few previous researches colligated with detection of forged images bearing reflection attack. Our method can effectively detect forged images sustaining reflection attacks as well as combination of reflection with rotation, scaling, and shearing attacks.
- We have addressed some of the novel attacks such as erosion, dilation, zoom motion blur, RGB color addition, spread noise addition, and non-affine transformation over tampered images. Our method is capable to efficaciously detect forged images suffering from composite attacks including non-affine transformations with rotation, scaling, shearing, and reflection over tampered region.

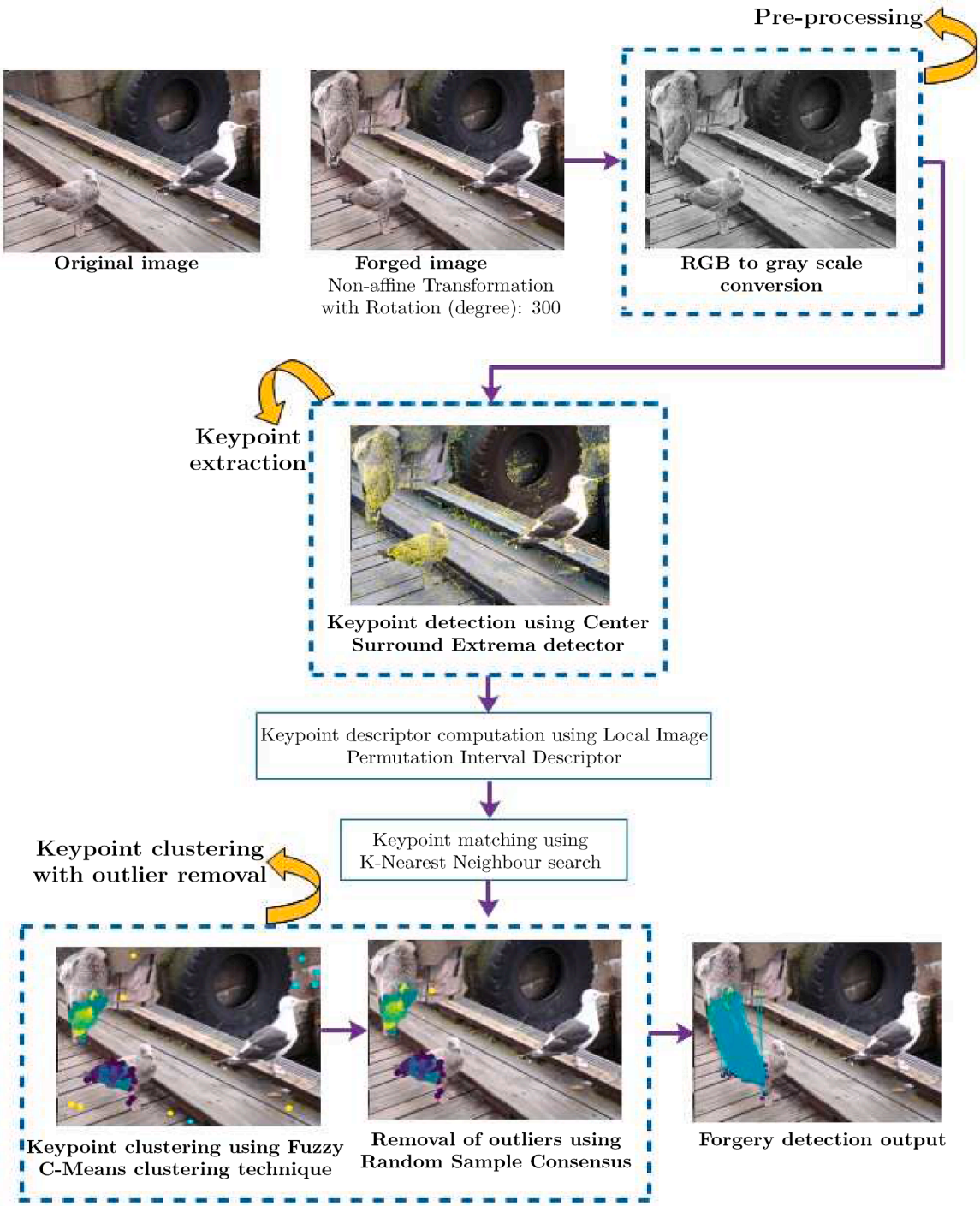

Fig. 4. Proposed methodology for the copy-move image forgery detection.

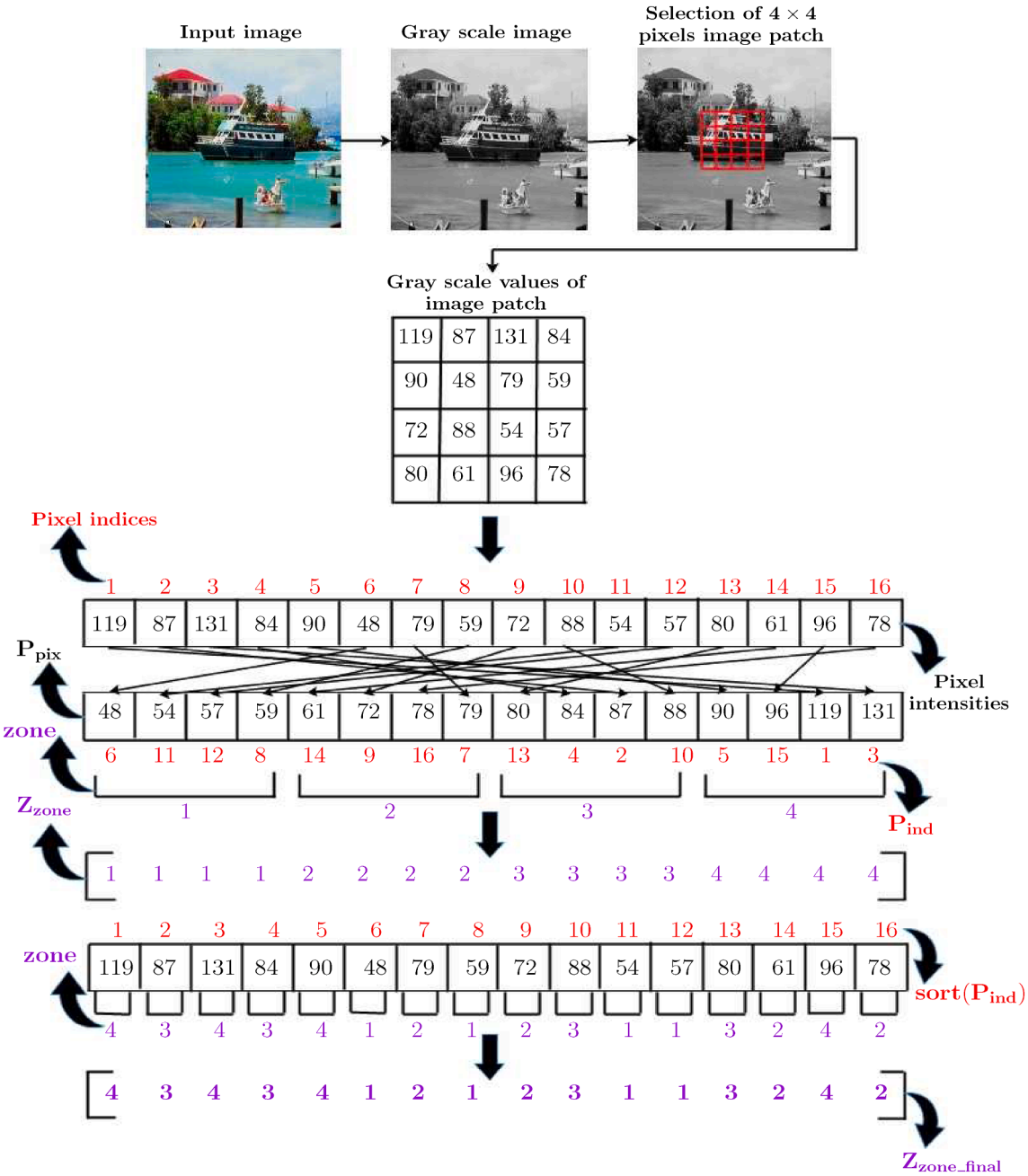

Fig. 5. Demonstration of fundamental steps for the computation of Local Image Permutation Interval Descriptor.

- Our scheme is a fast technique for copy-move image forgery detection as compared to state-of-the-art methods.
- Our algorithm can effectively detect forged images formulated using authentic images collected from different datasets, namely Ardizzone (Ardizzone et al., 2015b; Ardizzone, Bruno, & Mazzola, 2015a), CASIA TIDE v2.0 (Dong, Wang, & Tan, 2013), CoMoFoD (Tralic, Zupancic, Grgic, & Grgic, 2013a; Tralic, Zupancic, Grgic, & Grgic, 2013b), IMD (Christlein, Riess, Jordan, Riess, & Angelopoulou, 2012b; Christlein, Riess, Jordan, Riess, & Angelopoulou, 2012a), MICC- F2000 (Amerini, Ballan, Caldelli, Del Bimbo, & Serra, 2011a; Amerini et al., 2011b), and MICC-F8multi (Amerini et al., 2011a; Amerini et al., 2011b).

The rest of the paper is organized as follows. Proposed methodology is discussed in Section 2. The experimental results and comparative analysis with existing copy-move forgery detection techniques is illustrated in Section 3. Finally, the proposed work is concluded in Section 4.

2. Proposed methodology

We have classified our forgery detection method in six primary steps, namely Preprocessing, Extraction of keypoints from the image, Keypoint descriptor computation, Matching of keypoints, Clustering of keypoints, and Affine transformation estimation which are discussed in the following subsections. As preprocessing step, if the input image is a RGB color image, it is converted into gray scale image as in (1).

$$I = 0.299 \times R + 0.587 \times G + 0.114 \times B \tag{1}$$

2.1. Keypoint extraction

Keypoints are also referred as feature points or interest points. Interest points consist several types of corner, maxima shapes, and edges. Generally, the interest points are the keypoints which can be easily detected as well as they postulate less computational overhead. The detected interest points are positioned at fine location to calculate corresponding feature descriptor. Hence, an interest point can be known as the keypoint around which feature descriptor can be computed. Most commonly used interest point detectors utilize minima and maxima points such as corners and gradient peaks. In addition to that, edges, contours, and ridges are also utilized as keypoints. Most of the times, corners are given more preference than isolated maxima points or edges, since corner is a structure that can be utilized to compute angular orientation corresponding to the interest point. These interest points can be detected over color as well as gray scale components. Several interset point detector techniques apply Gaussian filter across the image and then apply a gradient operator. Gaussian filter is applied to reduce noise present in image, otherwise noise will get amplified by using gradient operators. Different techniques of interest point extraction detect keypoints with varying degree of invariance towards several operations applied over image such as scaling, rotation, perspective change, illumination change, occlusion, etc.

We have used CenSurE detector (Kosaka & Ohashi, 2015), for keypoint extraction from the gray scale image. CenSurE detector extract interest points at all scales as well as pick out extrema across scale and location. This detector utilizes bi-level kernels as center-surround filters to achieve fast computation.

2.1.1. Finding extrema

CenSurE detector computes a simplified center surround filter for all the scales and locations to identify the extrema around local neighborhood. As next step for extrema detection, obtained extremas are filtered by calculating the Harris measure (Agrawal, Konolige, & Blas, 2008) followed by elimination of those which are having weak corner response.

2.1.2. Bi-level filters

Various detectors approximated the Laplacian using the difference of Gaussians. To simplify the approximation step, CenSurE detector utilizes center-surround filters which are bi-level, i.e., they multiply the image values only by either 1 or -1. For different degrees of symmetry, progression of bi-level filters are shown in Fig. 2. We have utilized STAR detector.

2.1.3. CenSurE using STAR

The STAR keypoint detector (Lopez-Puigdollers, Traver, & Pla, 2019) that we have used is a variation of CenSurE detector which uses a star shaped kernel. Star shaped kernel is an overlayer of a 0° and a 45° oriented CenSurE based kernel which achieves better estimation of Laplacian of Gaussian (LoG). The center-surround kernels are simply summations of addition and substraction. To obtain highly efficient results, they can also be computed using integral images. The output obtained using STAR detector consists salient locations and scales of patches present in the input image.

2.1.4. Non-maximal suppression

CenSurE computes seven filter responses for each pixel of the image. Non-maximal suppression is performed over the scale space. The obtained response is suppressed if response is greater (maxima case). or less (minima case) than its neighbors around a local neighborhood for the location and scales. Pixels that are recognized as either minima or maxima in local neighborhood represents the interest point locations. For non-maximal suppression, CenSurE utilizes $3 \times 3 \times 3$ neighborhood. The magnitude of the obtained filter response provides an indication for the potency of the interest point. The interest point with greater strength is more likely to be repeatable. As weak responses are not stable so they are filtered out using thresholding. CenSurE detector supports computation of all the responses over input image without postulating the requirement of subsampling, hence all interest point locations are finely localized, while utilization of subpixel interpolation is not required.

2.1.5. Line suppression

Interest points which are obtained along edge or line are poorly localized, hence they are not very stable. Such poorly defined interest points possess large principal curvatures in direction of line, but these peaks have smaller principal curvature in the perpendicular direction. Hence, these peaks can be filtered out by utilizing the ratio of principal curvatures. CenSurE detector utilizes the second moment matrix corresponding to response function for specific scale to filter out poor responses as in (2).

$$H = \begin{bmatrix} \sum L_x^2 & \sum L_x L_y \\ \sum L_x L_y & \sum L_y^2 \end{bmatrix} \tag{2}$$

where L_x and L_y represent derivatives corresponding to response function L along x and y directions. The summation is performed over a window which linearly depends on the scale corresponding to specific interest point, i.e., for higher scale, window size gets larger. In CenSurE detector, scale-adapted Harris measure is utilized (as in (2)) which is different from the Hessian matrix utilized to filter out line responses obtained during SIFT keypoints computation. After computation of Harris measures, ratio of principal curvatures can be computed by utilizing trace and determinant of Harris measures. The CenSurE detector provides symmetric stability (Henderson & Izquierdo, 2016). The keypoints obtained using CenSurE detector over a forged image are shown in Fig. 3. The profound model followed by proposed technique for copy-move image forgery detection is shown in Fig. 4.

2.2. Keypoint descriptor computation

After obtaining keypoints using CenSurE detector, we have

computed feature descriptors corresponding to the keypoints using LIPID (Tian, Sethi, Ming, Zhang, & Ma, 2013) descriptor. This descriptor is computed using order permutation of pixels and their indices. Permutation over pixel intensities can be calculated as in (3).

$$P_{pix} = sort(\, g_1\ g_2\ \ldots\ g_N\,) = \left(g_{min}\ g_p\ \ldots\ g_q\ g_{max}\right) \tag{3}$$

where $g_1, g_2, \ldots, g_N$ represent pixel intensities for N pixels and *sort* shows the sorting operation applied over original vector containing N pixels. P_{pix} shows the ascending order permutation corresponding to pixel intensities where $\{min, p, \ldots, q, max\} \in [1, N]$. As image pixel consists two attributes, namely intensity and position index so we have also obtained another permutation P_{ind} corresponding to pixel indices as in (4).

$$P_{ind} = \left(i_{g_{min}}\ i_{g_p}\ \ldots\ i_{g_q}\ i_{g_{max}}\right) \tag{4}$$

where $i_{g_{min}}, i_{g_p}, \ldots, i_{g_q}, i_{g_{max}}$ represents indices of sorted pixel intensities $g_{min}, g_p, \ldots, g_q, g_{max}$. A vector Z_{zone} is defined using the zone division operation as in (5).

$$zone\left(j\right) = \left\lfloor \frac{j-1}{intrv} + 1 \right\rfloor \tag{5}$$

where $\lfloor . \rfloor$ shows floor operation, *intrv* denotes the number of pixels present in each zone, and $j = 1, 2, \ldots, N$. Vector Z_{zone} can be represented as in (6).

$$Z_{zone} = (1\ \ldots\ 1\ 2\ \ldots\ 2\ zone\ \ldots\ zone) \tag{6}$$

Z_{zone} can also be considered as a permutation containing *intrv* for each zone number. Here, it is to be noted that the total number of pixels (N) and the total number of zones (*nzone*) should be selected pairwise such that *nzone* is exact divisor of N to obtain the integer value of *intrv*. To achieve the final permutation Z_{zone_final}, different *zone* are utilized to assign corresponding zone numbers to initial intensity vector obtained after performing *sort* operation over P_{ind} as in (7).

$$Z_{zone_final} = zone \rightarrow sort\left(P_{ind}\right) \tag{7}$$

P_{ind} obtained in (4) is index permutation corresponding to ascending intensities and used by Locally Uniform Comparision Image Descriptor (LUCID) technique (Ziegler, Christiansen, Kriegman, & Belongie, 2012) as descriptor of keypoints. We use results obtained in (7) as LIPID descriptor, which posssess more robustness than LUCID (Tian, Yang, Zheng, Yao, & Gao, 2017). Z_{zone_final} consists permutation on zone numbers corresponding to all pixels of an image patch which represents LIPID descriptor. Fig. 5 shows the illustration for the computation of LIPID descriptor. The vector containing initial pixel values is sorted according to pixel intensities to obtain P_{pix}, while modifications occurred in their corresponding indices are stored as P_{ind}. The ordered pixel values P_{pix} are evenly divided into several *zone* and different zone numbers are assigned over them which is shown as vector Z_{zone}. After applying *sort* operation over P_{ind}, initial pixel intensity vector is obtained. Further, *zone* are utilized to assign corresponding zone numbers to initial pixel intensity vector to achieve Z_{zone_final}. The permutation of zone numbers represented as Z_{zone_final} contains the approximate locations of pixels for a patch of input image. The finally obtained permutation Z_{zone_final} is also capable to depict an image patch and possess tolerance towards several deformations applied over image.

2.3. Keypoint matching

By utilizing CenSurE keypoint detector, we have obtained group of keypoints for the input image which can be represented as $P = \{kp_1, kp_2, kp_3, \ldots, kp_N\}$, where N represents total number of keypoints. Further, we have obtained feature descriptors using LIPID technique as discussed in Section 2.2 corresponding to the keypoints where group of feature descriptors can be represented as $F = \{F_1, F_2, F_3, \ldots, F_N\}$. To identify the similar feature descriptors, every feature vector $\{F_i, 1 \leqslant i \leqslant N\}$ should be matched with all the feature descriptors $\{F_j,\ 1 \leqslant j \leqslant N,\ j \neq i\}$. If we consider two exactly similar feature descriptors as matched then most of the similar feature descriptors corresponding to the duplicated region of image which sustain additional post-processing operations will remain undetected (as these operations mildly alters the feature descriptors). Hence, to effectively identify matched feature descriptors, Euclidean distance between feature descriptors is computed which can be represented as $|F_i - F_j|$. Feature descriptors F_i and F_j are considered as a match if the ratio of Euclidean distance between feature descriptors F_i, F_j (closest neighbor of F_i) and F_i, F_k (second-closest neighbor of F_i) is less than a user-defined threshold Th as in (8).

$$\frac{|F_i - F_j|}{|F_i - F_k|} \leqslant Th \tag{8}$$

Closest neighbor of a feature descriptor F_i is defined as the feature descriptor having minimum Euclidean distance from F_i. The number of false matches obtained during forgery detection process depends on the threshold (Th). Higher value of Th induces occurrence of more false matches during forgery detection. On contrary, when smaller value of Th is used, some of the approximately similar feature descriptors remain undetected. To identify nearest neighbor corresponding to feature descriptors in high-dimensional spaces is difficult task. However, the simplest way to find nearest neighbor is utilization of exhaustive search which is computationally inefficient and requires $O(N^2)$ computations where N is the total number of descriptors. Hence, we have used k-d tree as a data structure which can store N number of feature descriptors with k-dimensional spaces in the leaves. The construction of k-d tree requires $O(Nlog_2N)$ computational operations. Basically, the k-d tree is generalised representation for simple binary tree which can be used to store, search, and sort the data. To search the nearest neighbor in k-d tree, we have utilized BBF search algorithm.

BBF search algorithm (Beis & Lowe, 1997) is utilized to detect

Algorithm 1: Algorithm for Fuzzy C-Means clustering.

1 Initialize matrix $U = [u_{ij}]$ with $U^{(0)}$
2 For k-steps, calculate the vector for centers $C^{(k)} = [c_j]$ with $U^{(k)}$ where $c_j = \frac{\sum_{i=1}^{N} u_{ij}^m . x_i}{\sum_{i=1}^{N} u_{ij}^m}$
3 Perform modification over $U^{(k)}$ and $U^{(k+1)}$ using $u_{ij} = \frac{1}{\sum_{k=1}^{v} \left(\frac{||x_i - c_j||}{||x_i - c_k||}\right)^{\frac{2}{m-1}}}$
4 **if** $||U^{(k+1)} - U^{(k)}|| < \epsilon$ **then**
5 | STOP the iteration
6 **else**
7 | return to Step 2
8 **end**

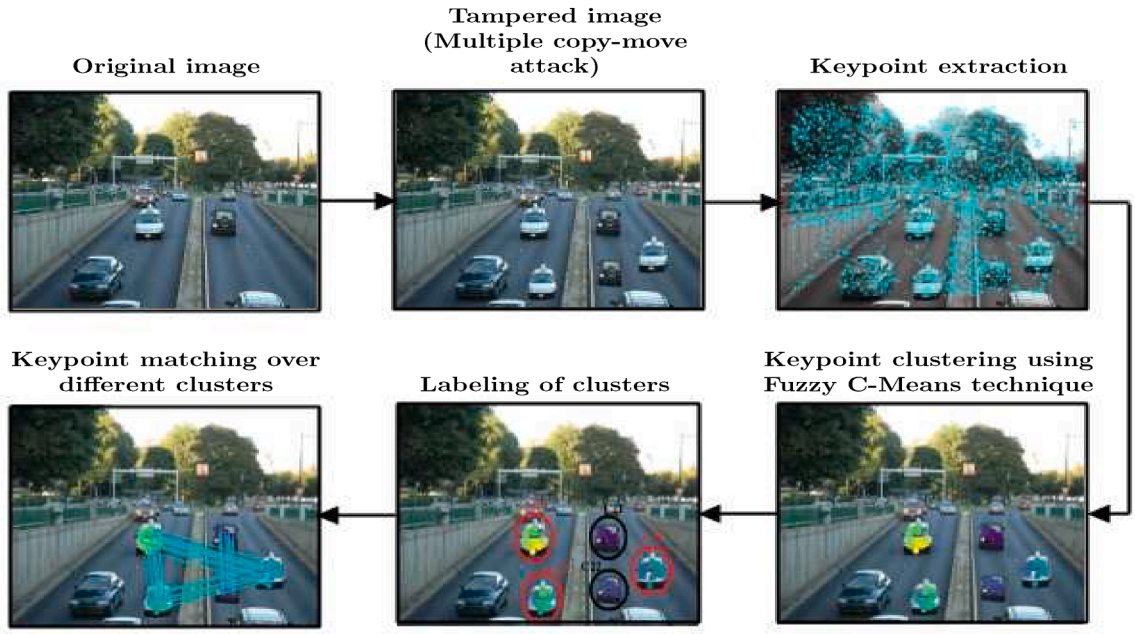

Fig. 6. Demonstration for Fuzzy C-Means clustering over keypoints.

Table 3
Datasets used for collection of images.

Dataset	Image dimension (Pixels)	Number of Images	Image category	Image format
Ardizzone (Ardizzone et al., 2015b; Ardizzone et al., 2015a)	1000 × 700 and 700 × 1000	D_0(Tampered):50, D_1(Tampered):680, D_2(Tampered):380, D_3(Authentic):50	Plant, Animal, Human, Nature, Texture, Architecture, Scene.	BMP
CASIA TIDE v2.0 (Dong et al., 2013)	320 × 240 to 800 × 600	Authentic:7200, Tampered:5123	Animal, Architecture, Nature, Article, Plant, Texture, Indoor scenes.	BMP, TIFF, JPEG
CoMoFoD (Tralic et al., 2013a; Tralic et al., 2013b)	512 × 512 and 3000 × 2000	Base images:40 (512 × 512), Base images:10 (3000 × 2000) Tampered: 260	Building, City views, Nature, Grass, Wall,Roof, Sky.	JPEG, PNG
MICC-F8multi (Amerini et al., 2011a; Amerini et al., 2011b)	2048 × 1536	Tampered: 8	Vehicles, Buildings, Birds, Landscapes, Flowers, Animals, Humans.	JPEG
MICC-F2000 (Amerini et al., 2011a; Amerini et al., 2011b)	2048 × 1536	Authentic:1300 Tampered:700	Bulidings, Landscapes, Vehicles, Humans, Flowers, Animals, Birds.	JPEG
IMD (Christlein et al., 2012b; Christlein et al., 2012a)	800 × 533 to 3888 × 2592	Base images: 48	Bridges, Buildings, Animals, Humans, Natural places, Tourist places, Statues.	PNG

approximate nearest neighbor. The data structure used for BBF search consists six different parts: pointer for the left tree, pointer for the right tree, discriminator i to divide all the feature descriptors which marks the dimension of feature descriptor for which it exhibits the highest variance, partition value v which stores the median value of all the feature descriptors for dimension i, number of feature descriptors in the k-d tree, and pointer for the list of feature descriptors. BBF algorithm also utilizes a priority queue. The information present in the priority queue consists two parts: current position of the tree and distance of point q from some node of the k-d tree where q symbolizes the query point. Since, each and every query point q does not have a matching point so, in BBF algorithm, we set Q_{max} as limit for query number. If a query point can not search the nearest neighbor even after the maximum query number Q_{max} exceeds, we drop out the query point q without searching for the other corresponding matching points.

2.4. Keypoint clustering

To cluster matched keypoints, we have applied FCM technique (Pal, Pal, Keller, & Bezdek, 2005). FCM is an efficient technique to perform clustering over the data points which can belong to two or more clusters. In order to reduce the time consumed in clustering CenSurE keypoints, the objective function as in (9) should be reduced.

$$F_m = \sum_{i=1}^{N} \sum_{j=1}^{v} u_{ij}^m \left\| x_i - c_j \right\|^2, 1 \leqslant m < \infty \tag{9}$$

Where m represents a real number greater than 1. N is the number of data points and v is the maximum number of clusters. u_{ij} shows the degree of membership of data point x_i in cluster j. x_i denotes ith component of d-dimensional data points. c_j represents d-dimensional

Table 4
Variation in range of different attacks applied over forged image.

Attack	Extent of attacks
Rotation	100°, 150°, 200°, 250°, 300°
Shearing	Widthwise→ 50, 75, 100, 125, 150
	Heightwise→ 50, 75, 100, 125, 150
Downscaling	Widthwise→ 20%, 50%,
	Heightwise→ 20%, 50%
Upscaling	Widthwise→ 150%, 200%, 250%
	Heightwise→ 150%, 200%, 250%
JPEG Compression (Quality level)	2, 5, 10, 20, 30
Zoom motion blur (Blurring factor)	0.03, 0.06, 0.09, 0.12, 0.15
Spread noise addition (Randomness)	10, 15, 20, 25, 30

center of the cluster and $\| * \|$ shows the norm used to compute similarity between candidate feature vector and the center. An improved application for the above objective function is Fuzzy partitioning (Yang & Nataliani, 2017) which utilizes the updation of membership degree u_{ij} and the cluster centers c_j as in (10) and (11), respectively.

$$u_{ij} = \frac{1}{\sum_{k=1}^{v} \left(\frac{||x_i - c_j||}{||x_i - c_k||} \right)^{\frac{2}{m-1}}} \tag{10}$$

$$c_j = \frac{\sum_{i=1}^{N} \mu_{ij}^m . x_i}{\sum_{i=1}^{N} u_{ij}^m} \tag{11}$$

The repetition of u_{ij} stops when $max_{ij}\{\left|u_{ij}^{k+1} - u_{ij}^{k}\right|\}\langle\in$ where $\in$ represents termination value whose value lies between 0 and 1 whereas k shows the number of iterations. This step provides a local minimum or saddle point for F_m. FCM algorithm assigns data points to different clusters by utilization of fuzzy memberships (Gong, Liang, Shi, Ma, & Ma, 2012).

By utilizing FCM algorithm, CenSurE keypoints are clustered by comparing their descriptors obtained using LIPID. Center keypoint and its neighbors are matched with different center keypoints and its neighbor clusters which reduce the requirement of analysis of all keypoints of the image, hence accelerate the forgery detection process. Keypoints which belong to same cluster are as similar as possible. On the

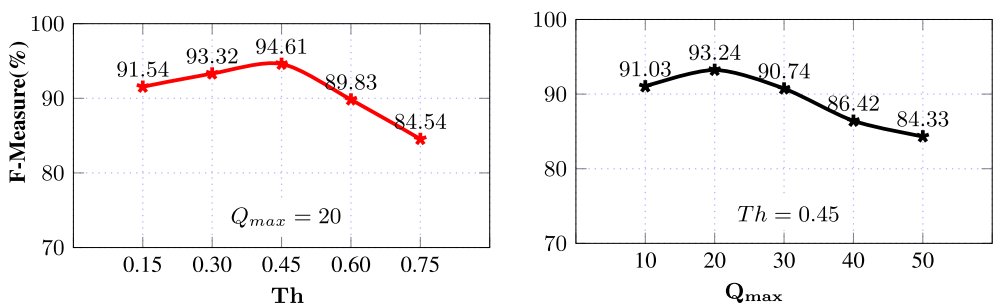

Fig. 7. Effect on F-Measure(%) occurred for the different values of threshold Th and Q_{max} during identification of matching descriptors.

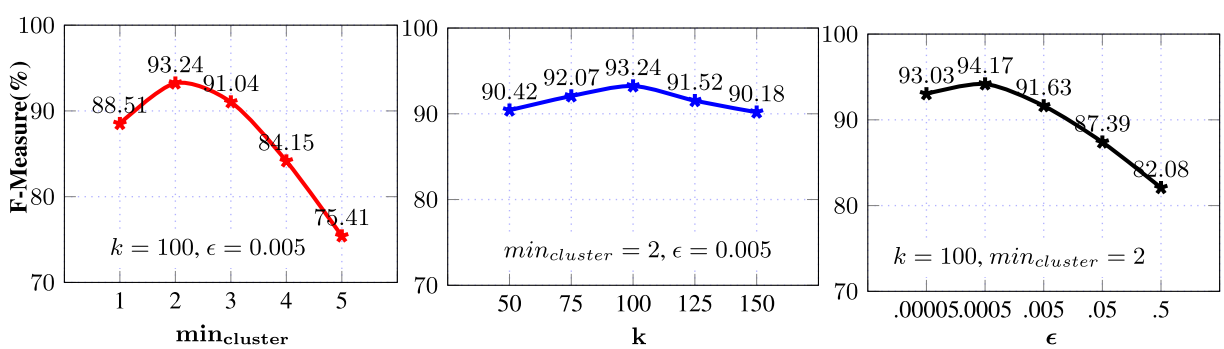

Fig. 8. Effect on F-Measure(%) based on the different values of $min_{cluster}$, k, and $\in$ parameters while performing FCM clustering.

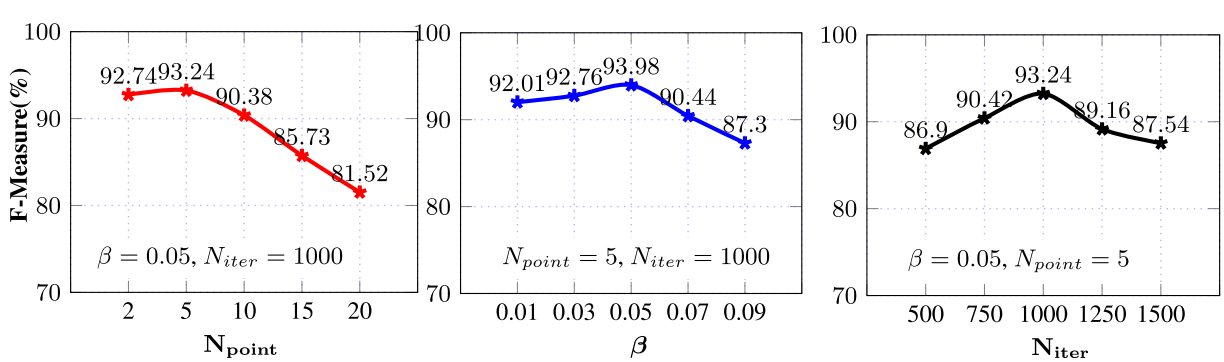

Fig. 9. Effect on F-Measure(%) based on the different values of N_{point}, β, and N_{iter} while performing affine transformation estimation with outlier removal.

contrary, all the keypoints belonging to diverse clusters are different. Three main parameters, namely number of clusters, maximum number of iterations, and minimum amount of improvement are associated with FCM clustering to find appropriate center for the cluster keypoints and close neighbors associated with center feature vector. Number of clusters defines the total number of required clusters for which clustering operation is applied. Maximum number of iterations represents the upper limit for total number of training epochs. The minimum improvement occurred over objective function by using two consecutive iterations is represented by third parameter, namely Minimum amount of improvement. The clusters of keypoints obtained using Fuzzy C-Means clustering are illustrated in Fig. 6.

2.5. Affine transformation estimation

In proposed approach, based on the prelude keypoint matching, the point co-ordinates of the matched pairs of keypoints $\{(X,\widetilde{X})_1,(X,\widetilde{X})_2,\ldots,(X,\widetilde{X})_N\}$ are utilized for estimation of the affine transformation between the copied regions. Basically, an affine transformation model is constructed to discover some potent geometrical transformations (such as rotation, scaling) and translation. For two point coordinates $X=(x,y)$ and $\widetilde{X}=(\widetilde{x},\widetilde{y})$ which are matched, affine transformation can be described as in (12).

$$\begin{pmatrix}\widetilde{x}\widetilde{y}1\end{pmatrix}=\begin{pmatrix}a & b & t_x\\ c & d & t_y\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}x\\ y\\ 1\end{pmatrix}=T\begin{pmatrix}x\\ y\\ 1\end{pmatrix} \tag{12}$$

where a, b, c, and d parameters are consociated with scaling and rotation while t_x and t_y parameters are related with translation. The matrix T is computed by resorting for at least three non-collinear data points which are matched. For estimation of affine transformations, we have utilized same method as in (Pan & Lyu, 2010; Amerini et al., 2011b). For a given set of matched points $(X_1,X_2,\ldots,X_N)$ and $(\widetilde{X}_1,\widetilde{X}_2,\ldots,\widetilde{X}_N)$, we search for matrix T which minimzes the total error function as in (13).

Table 5
Parameter setting for the proposed approach.

Parameters	Description
$Th = 0.45$	Threshold for distance ratio.
$min_{cluster} = 2$	Minimum number of clusters used in Fuzzy C-Means clustering.
$k = 100$	Number of iterations used during Fuzzy C-Means clustering.
$\epsilon = 0.005$	Termination value used in Fuzzy C-Means clustering.
$m = 2$	Fuzziness parameter in Fuzzy C-Means clustering.
Distance metric='*Euclidean*'	Distance metric used in Fuzzy C-Means clustering.
$N_{point} = 5$	Maximum number of inliers for an affine transformation estimation.
$\beta = 0.05$	Distance threshold for matched keypoints used during affine transformation estimation.
$N_{iter} = 1000$	Number of iterations used to obtain maximum inliers during affine transformation estimation.
$Q_{max} = 20$	Limit for maximum query number during nearest neighbor search.

Table 6
Quantitative results of copy-move forgery detection obtained utilizing proposed technique when images are tampered using different attacks.

Attack	F-Measure(%)	FPR(%)
Reflection	93.236	2.343
Combination of reflection	94.302	4.021
Erosion	95.138	3.177
Dilation	96.207	2.613
RGB color addition	94.631	2.733
Non-affine transform	96.183	2.174
Non-affine transform with reflection	88.362	3.046
Multiple copy-move	95.215	4.023

Algorithm 2: Algorithm for the proposed methodology.

Variable Declaration:

$kp_{CenSurE}$: Keypoints extracted from input image using CenSurE detector
D_{LIPID}: Descriptor obtained for extracted keypoints using LIPID
P_k: Positions corresponding to matched keypoints, $k=1,2$
C_i: Cluster obtained using Fuzzy C-Means clustering
T: Estimated affine transformation matrix
$P_{inliers}$: Keypoints identified as inliers using RANSAC technique

Input : Image Im with $M \times N$ size
Output: Forgery detection result (I_{output})

```
[M, N, ch] ← size of input image (Im)
/* Convert color image into gray-scale */
if ch > 1 then
    I ← convert Im into gray-scale image
else
    I ← Im
end
for P = 1 to M do
    for Q = 1 to N do
        /* extract keypoints from the image using CenSurE detector */
        kp_CenSurE ← CenSurE(I)
        L ← Length(kp_CenSurE)
        /* Compute keypoint descriptor using LIPID */
        D_LIPID ← LIPID(kp_CenSurE)
    end
end
/* Perform matching over descriptors */
Construct k-dimensional tree to store D_LIPID
Apply Best-Bin-First search algorithm over descriptors
for X = 1 to L do
    for Y = 1 to Length(D_LIPID) do
        [P1, P2] ← kNN(D_LIPID, kp_CenSurE)
    end
end
/* Cluster matched keypoints using Fuzzy C-Means technique */
for each matched keypints do
    Cluster C_i ← Fuzzy C-Means(P1, P2)
end
/* Apply RANSAC to eliminate the false matches */
[T, P_inliers] ← RANSAC(P1, P2)
/* Map P_inliers over image where inliers belong to different clusters */
if i > 1 after Fuzzy C-Means clustering then
    I_output ← Map(P_inliers)
else
    No forgery detection
end
```

$$argmin_T \sum_{i=1}^{N}\left|\left|\widetilde{X}_i - TX_i\right|\right|^2 \tag{13}$$

To obtain more precise transform parameters and to remove the unreliable keypoints simultaneously, we have employed a widely used technique recognized as the RANSAC algorithm. At first, the RANSAC algorithm was introduced by Fischler and Bolles (Fischler & Bolles, 1981). This technique can estimate general parameters iteratively for a certain model using an input data set which contains high percentage of outliers. To utilize RANSAC algorithm, at first, three pairs of the matched points from the candidate matches are selected which are non-collinear and utilized to estimate the affine transformation matrix T. Rest of the data points are transformed based on the matrix T. Further, distance between the corresponding matched data points is computed. If the distance is less than the user-defined threshold β then the matched data points are considered as inliers otherwise the points are viewed as outliers. After N_{iter} number of iterations, the estimated transformation

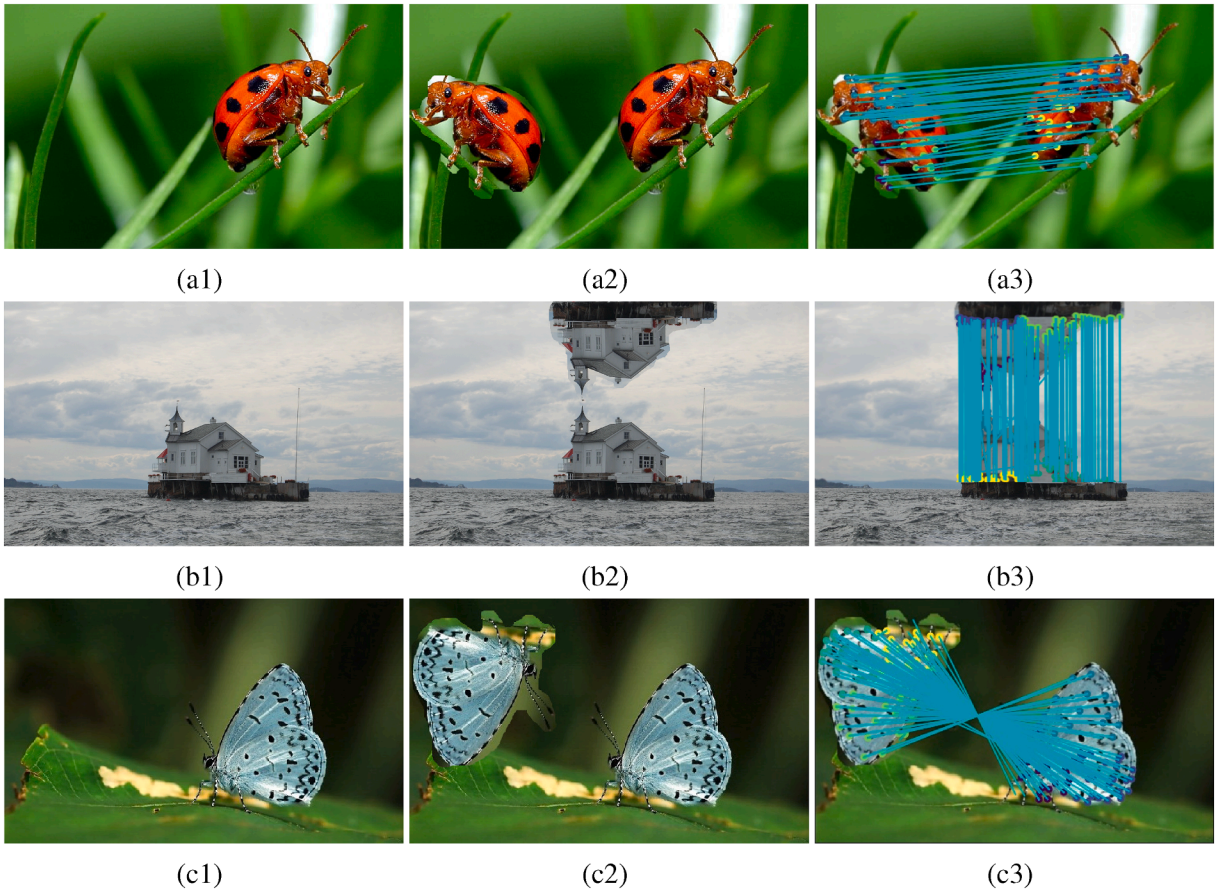

Fig. 10. {(a1), (b1), (c1)} Authentic images collected from CASIA TIDE v2.0 (Dong et al., 2013) and MICC-F2000 (Amerini et al., 2011a) dataset. {(a2), (b2), (c2)} Tampered images with forged regions sustaining horizontal reflection, vertical reflection, and combination of reflection attacks, respectively. {(a3), (b3), (c3)} Forgery detection results obtained using proposed approach.

which produces largest number of inliers is obtained. As the transformation is selected for each iterative step, if any inlier satisfies the transformation then it is removed from the succeeding round of searching. The algorithm does not stops until sufficient number of inliers N_{point} in a transformation is obtained.

3. Experimental results and discussion

We have performed all the experiments on Python version 3.7 which is installed upon a personal computer sustaining 64-bit windows platform having 8GB RAM with Intel 2.90 GHz core *i*7 processor. This section provides information related to accumulation of images from different datasets for experimentation. It includes discussion on quantitative and qualitative results obtained using proposed technique. Further, this section also contains comparision of performance of proposed technique and state-of-the-art methods.

3.1. Dataset

We have collected images from six datasets, namely Ardizzone, CASIA TIDE, CoMoFoD, MICC-F8multi, MICC-F2000, and IMD as shown in Table 3. Ardizzone dataset (Ardizzone et al., 2015b; Ardizzone et al., 2015a) contains images of size 1000 × 700 and 700 × 1000 pixels. Images are divided into four subdatasets, namely D_0, D_1, D_2, and D_3. D_0 dataset contains 50 tampered images with simple translation attack. D_1 dataset contains 680 images with rotation attack (i.e., copy-pasting image objects after rotation), while D_2 subdataset comprise 380 images bearing scaling attack (i.e., copy-pasting image objects after scaling). Subdataset D_3 consists of 50 non-tampered original images representing simple scenes with simple background. All images of Ardizzone dataset are stored in BMP file format. CASIA TIDE dataset (Dong et al., 2013) composed of two parts CASIA V1.0 and CASIA V2.0. Version 1.0 is smaller set containing 1725 color images of size 384 × 256 pixels stored in JPEG format. It contains 800 authentic and 921 tampered images with splicing attack. Version 2.0 is extension of Version 1.0. It contains 12323 color images classified into two subsets, i.e., 7200 authentic and 5123 tampered images. Version 2.0 sustains more complex images as compared to Version 1.0. Version 2.0 contains images of different sizes ranging from 320 × 240 to 800 × 600 pixels stored in BMP, TIFF, and JPEG file format. CASIA dataset contains images related to animal, indoor scenes, architecture, plant, article, nature, and texture category. The CoMoFoD dataset (Tralic et al., 2013a; Tralic et al., 2013b) contains images belonging to buildings, city views, nature, grass, wall, roof, sky, etc. This dataset contains images of two different sizes with 512 × 512 pixels (small size) and 3000 × 2000 pixels (large size). In small size category, dataset contains 40 authentic images which are utilized to produce 40 images for each of the translation, rotation, scaling, distortion, and combinational attacks. In large size image

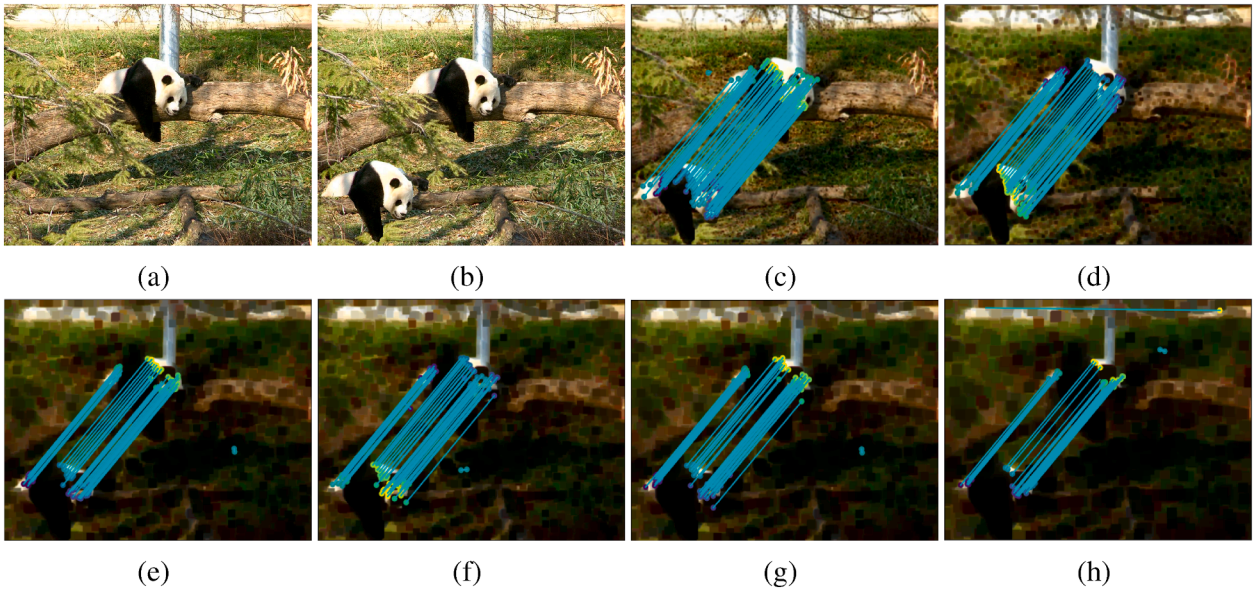

Fig. 11. (a) Authentic image fetched from Ardizzone (Ardizzone et al., 2015a) dataset. (b) Tampered image obtained using copy-move forgery. {(c)–(h)} Forgery detection outcomes achieved using proposed technique when extent of erosion attack is gradually increased over tampered image.

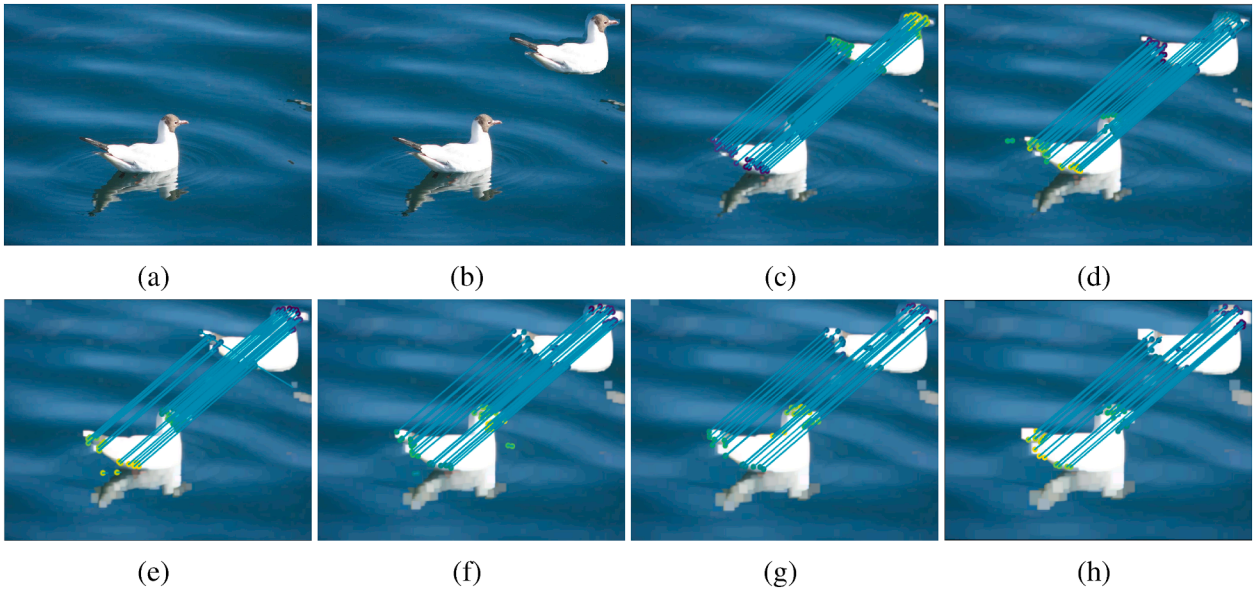

Fig. 12. (a) Authentic image taken from Ardizzone (Ardizzone et al., 2015a) dataset. (b) Manipulated image obtained using copy-move forgery. {(c)–(h)} Forgery detection outputs obtained by utilizing proposed approach when extent of dilation attack is gradually increased over manipulated image.

category, dataset contains 10 images. These images are used as base images to construct forged images with translation, translation, rotation, scaling, combination, and distortion attack. JPEG and PNG file format are used to store images. MICC-F8multi dataset (Amerini et al., 2011a; Amerini et al., 2011b) contains 8 different realistic as well as challenging images sustaining multiple copy-move attacks. This dataset contains images of size 2048×1536 pixels and stored in JPEG file format. All images are related to vehicles, bulidings, birds, landscapes, flowers, animals, humans, etc. MICC-F2000 (Amerini et al., 2011a; Amerini et al., 2011b) dataset contains 2000 images of size 2048×1536 pixels where 1300 images are authentic and 700 images are tampered. Dataset images are related to bulidings, landscapes, vehicles, humans, flowers, animals, birds, etc. All images are stored in JPEG file format. IMD dataset (Christlein et al., 2012b; Christlein et al., 2012a) contains 48 authentic images related to bridges, natural scenes, animals, buildings, statues, human, tourist places, etc. which are utilized to construct forged images containing various attacks such as rotation, scaling, JPEG compression, multiple copy-move, and combinational attack (sustaining rotation with scaling). Images are stored in PNG file format and their sizes vary from 800×533 to 3888×2592 pixels.

We have collected 40 authentic images from each of the Ardizzone, CASIA TIDE v2.0, CoMoFoD, MICC-F2000, and IMD datasets whereas 8 images (containing multiple copy-move attacks) are taken from MICC-F8multi dataset. To perform forgery detection experiments, we have constructed 200 tampered images corresponding to each of the attack, namely reflection (with rotation, scaling, and shearing), combination of reflection, non-affine transformation (with rotation, scaling, shearing, and reflection), erosion, dilation, RGB color addition (change in color balance), multiple copy-move, JPEG compression, zoom motion blur, and spread noise addition.

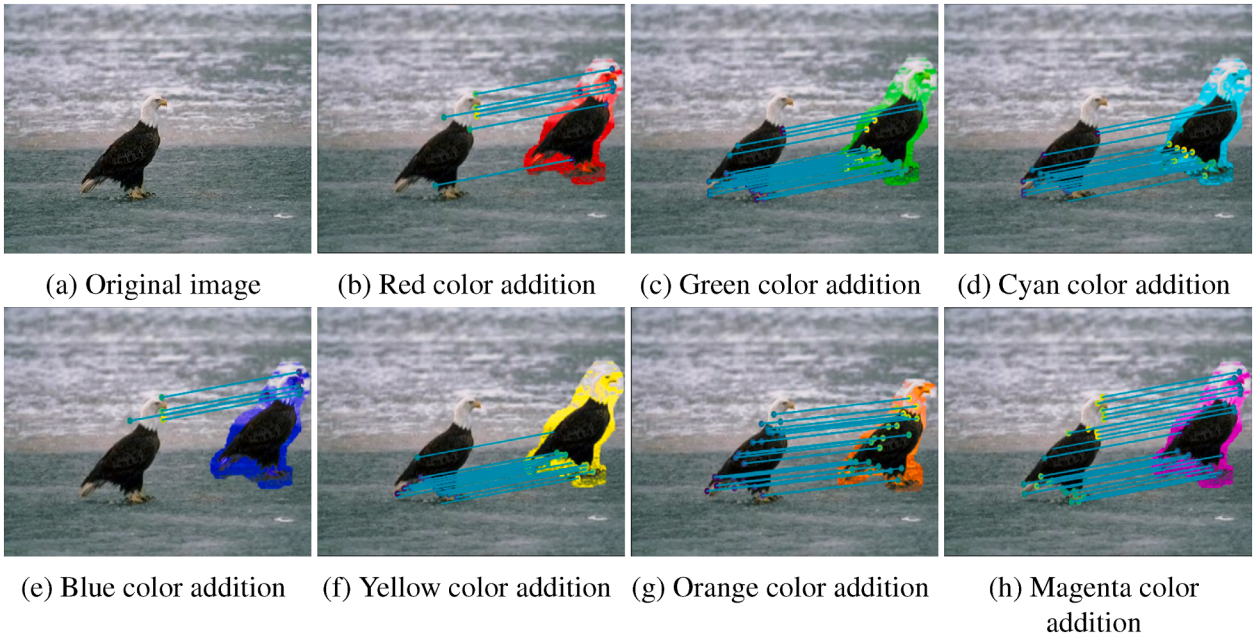

Fig. 13. (a) Original image collected from CASIA TIDE v2.0 (Dong et al., 2013) dataset. {(b)–(h)} Forgery detection outcomes obtained by utilizing proposed algorithm over forged images having tampered region with different color addition attack.

3.2. Evaluation metrics

To analyze performance of the proposed scheme, we have used two parameters, namely F-Measure and False Positive Rate (FPR) which are computed as in (14) and (15), respectively.

$$F-Measure\left(\%\right) = \frac{2 \times TP}{2 \times TP + FP + FN} \times 100 \tag{14}$$

$$FPR\left(\%\right) = \frac{FP}{FP + TN} \times 100 \tag{15}$$

where TP represents the number of tampered images which are correctly identified as tampered. FP shows the number of authentic images which are fallaciously identified as forged. TN refers to the number of images which are not tampered and correctly detected as untampered. FN denotes the number of images which are manipulated but forgery detection process wrongly identified them as authentic images.

3.3. Performance evaluation

We have analyzed performance of the proposed scheme for reflection (with rotation, scaling, and shearing), combination of reflection, non-affine transformation (with rotation, scaling, shearing, and reflection), erosion, dilation, RGB color addition (with change in color balance), multiple copy-move, JPEG compression, zoom motion blur, and spread noise addition attack by employing F-Measure and FPR parameters which is discussed in the following subsections. The various extent of operations applied over images corresponding to different attacks are shown in Table 4.

Effect of user-defined parameters on performance:

To obtain the appropriate values for different user-defined parameters, we have performed several experiments. We have constructed 200 forged images containing reflection attack and mixed them with 200 original images taken from different datasets as mentioned in Section 3.1 to achieve the most suitable values for user-defined parameters based on the obtained F-Measure(%). The proposed technique has achieved highest F-Measure(%) for $Th = 0.45$ and $Q_{max} = 20$ during identification of matching keypoint descriptors as shown in Fig. 7. While performing FCM clustering process, our method has obtained highest F-Measure(%) for $min_{cluster} = 2, k = 100$, and $\epsilon = 0.005$ as shown in Fig. 8. For affine transformation estimation with outlier removal process, our technique has achieved highest F-Measure(%) using $N_{point} = 5$, $\beta = 0.05$, and $N_{iter} = 1000$ as shown in Fig. 9. The quantitative values of different parameters utilized by the proposed technique during experimentation are provided in Table 5.

3.3.1. Reflection attack

We have performed experiment over 400 images where 200 images are authentic and collected from several datasets as specified in Table 3, while 200 images are forged sustaining tampered region with reflection attack. Proposed algorithm has achieved substantially high F-Measure and low FPR as 93.236% and 2.343%, respectively as shown in Table 6. The qualitative results obtained for forgery detection are shown in Fig. 10 {(a1)–(a3)} and Fig. 10 {(b1)–(b3)} where forged image bears horizontal and vertical reflection attack, respectively.

3.3.2. Combination of reflection

To analyze the performance of our technique against combined (dual) reflection attack, we have utilized 200 original images from various datasets and constructed 200 tampered images sustaining forged region with combined reflection attack. Our method shows robustness towards combined reflection attack and achived high F-Measure and low FPR as 94.302% and 4.021%, respectively as shown in Table 6. The quanlitative result obtained for forgery detection is shown in Fig. 10 {(c1)–(c3)}.

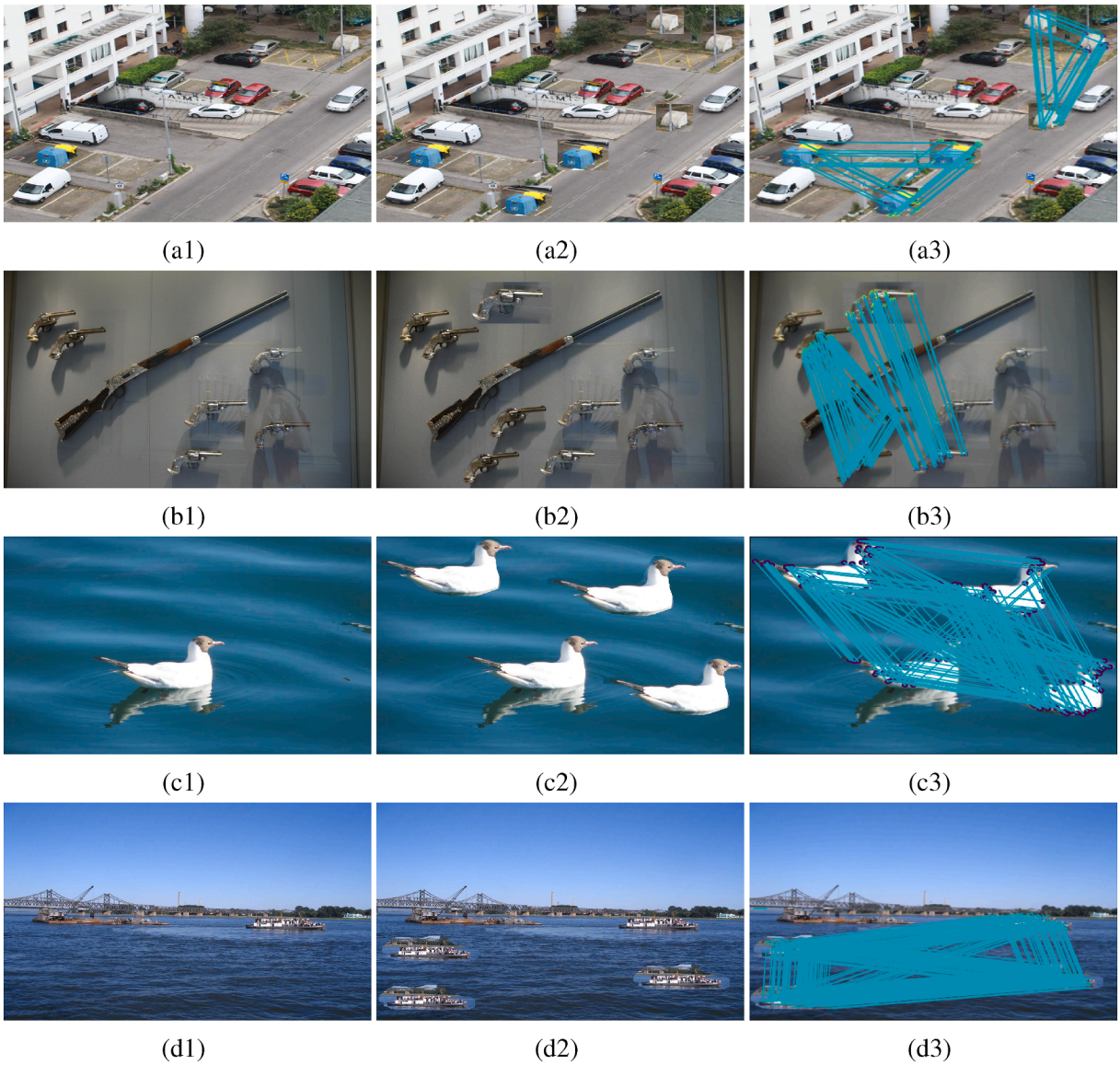

Fig. 14. {(a1), (b1), (c1), (d1)} Original images taken from CoMoFoD (Tralic et al., 2013a; Tralic et al., 2013b), MICC-F8multi (Amerini et al., 2011a) and Ardizzone (Ardizzone et al., 2015a) datasets. {(a2), (b2)} Forged images sustaining multiple copy-move pairs. {(c2), (d2)} Forged images where a source object is pasted in multiple regions of image. {(a3), (b3), (c3), (d3)} Forgery detection results obtained using proposed technique.

Table 7
Quantitative results of copy-move forgery detection obtained using proposed approach when images are manipulated using reflection with rotation, scaling, and shearing attacks.

Attack	Attack scale	F-Measure (%)	FPR (%)
Reflection with rotation	100°	94.312	3.427
	150°	90.003	4.011
	200°	87.374	4.818
	250°	86.143	6.932
	300°	84.021	7.813
Reflection with scaling	20%	84.006	8.813
	50%	90.816	5.331
	150%	93.114	3.052
	200%	85.703	6.183
	250%	86.121	7.045
Reflection with shearing	50	92.048	4.293
	75	91.136	4.865
	100	89.214	5.379
	125	86.375	6.201
	150	84.119	8.364

3.3.3. Erosion attack

Erosion operation broadens and enhances the dark areas of the image. For every pixel in 3×3 matrix, erosion assigns the lowest pixel value among the eight neighboring pixels to the current pixel. So, when erosion operation is applied over image, a dark pixel is appended around dark areas. In erosion operation, an isolated pixel present on a brighter background is converted into a big pixel consisting 9 pixels, hence produces some noise in resultant image. We have formulated 200 tampered images containing erosion attack. To analyze the performance of our scheme for forgery detection, we have utilized 400 images consisting of 200 original and 200 forged images. Our technique has achieved substantially high F-Measure and low FPR as 95.138% and 3.177%, respectively as shown in Table 6 which expresses robustness of the proposed algorithm against erosion attack applied over tampered image. The qualitative results for forgery detection obtained by utilizing proposed scheme are shown in Fig. 11.

3.3.4. Dilation attack

Dilation operation widens and enhances bright regions of the image. For each pixel in 3×3 matrix, dilation operation assigns the highest pixel value among 8 neighboring pixels to the current pixel. So, as resultant, bright pixel is appended around bright regions. An isolated pixel present on a brighter background is removed, while one bright pixel is appended around large size bright areas. This operation removes noisy pixels of the image. To evaluate performance of the proposed technique, we have created 200 tampered images sustaining dilation attack. To compute the quantitative results, we have used 400 images where 200 images are original and rest of the 200 images are tampered. Our scheme has achieved considerably high F-Measure and low FPR as 96.207% and 2.613%, respectively as shown in Table 6. The quantitative results prove the robustness of proposed technique against dilation attack over tampered image. The pictorial results obtained for forgery detection are shown in Fig. 12.

3.3.5. RGB color addition attack

We have created 200 tampered images which contain forged regions sustaining RGB color addition. To analyze performance of the proposed approach, we have utilized 400 images where 200 images are authentic and rest of the 200 images are tampered. Proposed scheme has achieved substantially high F-Measure and low FPR as 94.631% and 2.733%, respectively as shown in Table 6 which indicates the robustness of our approach against RGB color addition attack. The qualitative results for forgery detection obtained using proposed technique are shown in Fig. 13.

3.3.6. Multiple copy-move attack

To observe performance of the proposed technique for the detection of manipulated images sustaining multiple copy-move attack, we have formulated 200 forged images containing multiple duplicated regions. Multiple copy-move forged images contain two different cases of forgeries i.e., forged images with multiple copy-move pairs as well as forged images where single object is copied and pasted over the multiple regions of the image. Our technique is potentially able is detect both cases of multiple copy-move forgeries. To compute the quantitative parameters, we have collected 400 images consisting of 200 original and 200

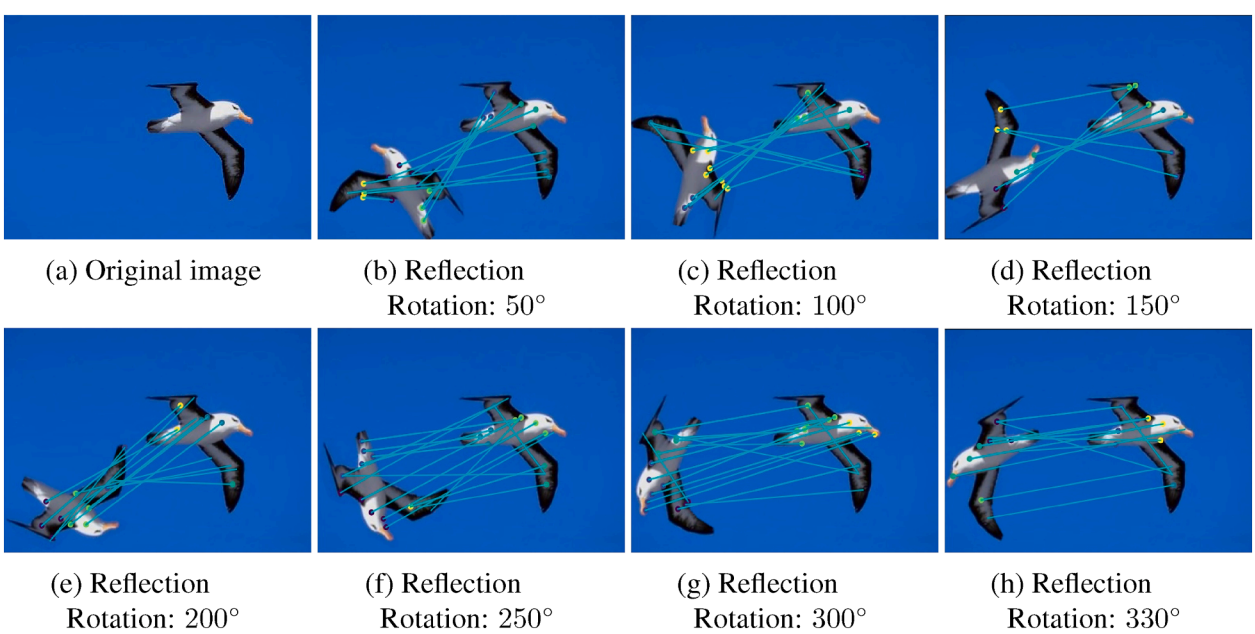

Fig. 15. (a) Authentic image taken from CASIA TIDE v2.0 (Dong et al., 2013) dataset. {(b)–(h)} Forgery detection results obtained by utilizing proposed technique where reflection with rotation attack is applied over cloned region.

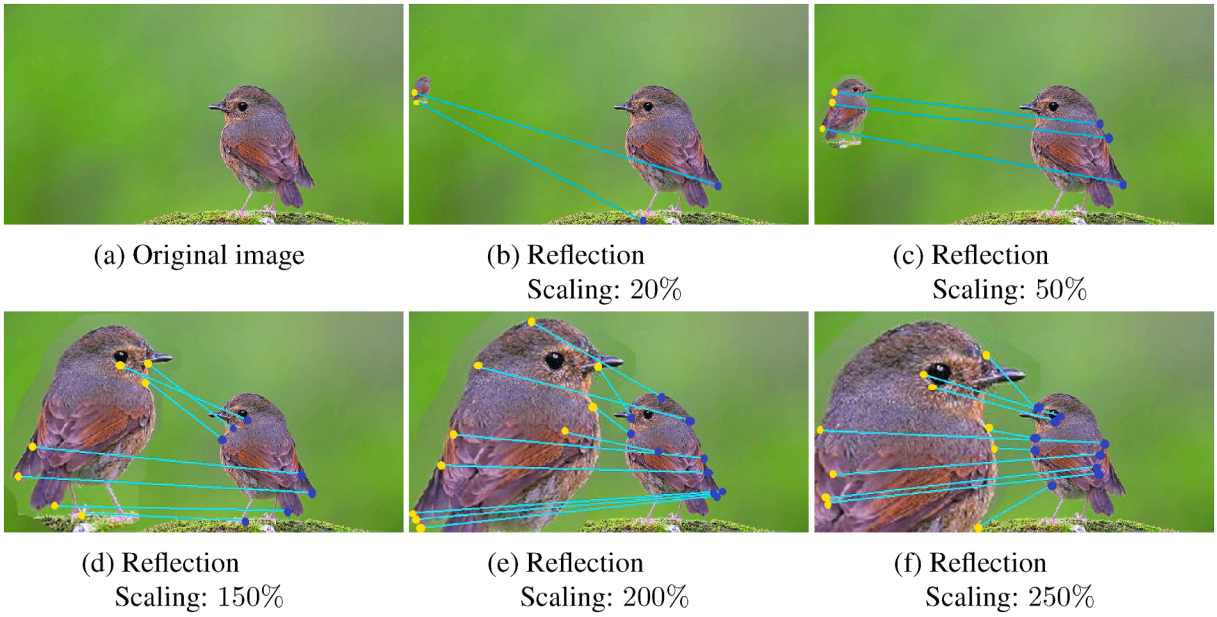

Fig. 16. (a) Original image picked up from CASIA TIDE v2.0 (Dong et al., 2013) dataset. {(b)–(f)} Forgery detection outcomes obtained using proposed approach when reflection with scaling attack is applied over duplicated region.

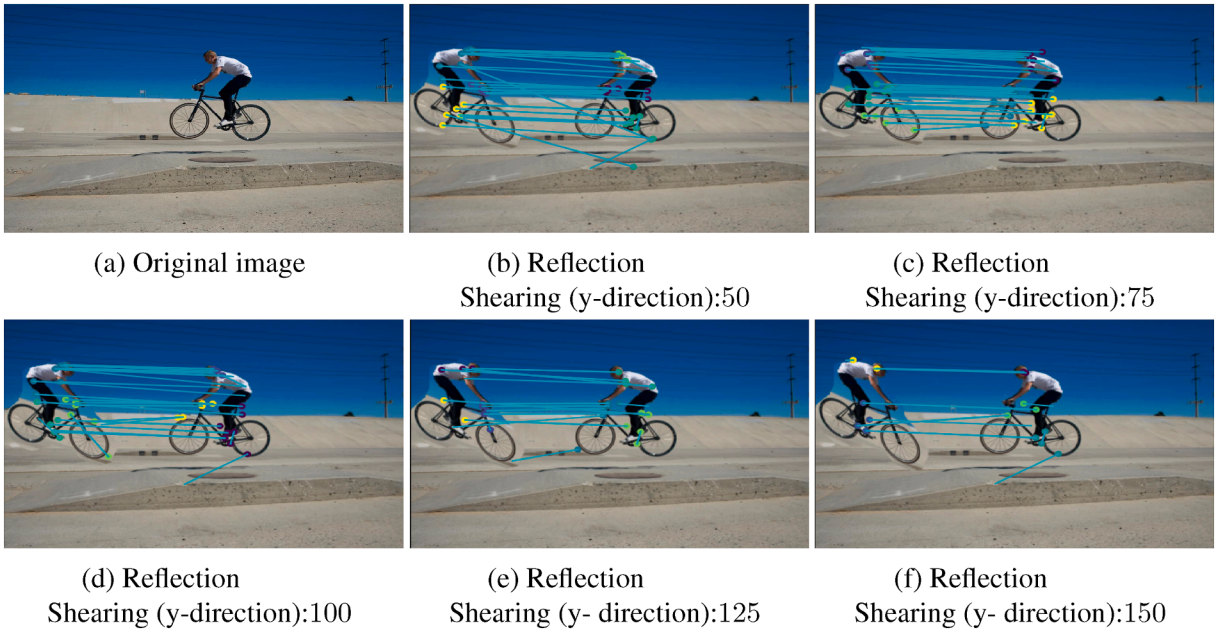

Fig. 17. (a) Authentic image collected from Ardizzone (Ardizzone et al., 2015a) dataset. {(b)–(f)} Forgery detection outputs obtained using proposed approach when reflection with shearing attack is enforced over duplicated region.

tampered images. Our method has achieved considerably high F-Measure and low FPR as 95.215% and 4.023%, respectively as shown in Table 6 which represents robustness of the proposed technique against multiple copy-move attack. We have shown pictorial results obtained using proposed technique for different cases of multiple copy-move forgery detection in Fig. 14.

3.3.7. Reflection with rotation attack

We have constructed forged images containing tampered region with reflection and rotations of 100°, 150°, 200°, 250°, and 300° as shown in Table 4. For each category of reflection with rotation attack, 40 forged images are created. Hence, we have observed performance of the proposed approach using 200 original and 200 tampered images. Our method has achieved substantially high F-Measure as 94.312%, 90.003%, 87.374%, 86.143%, and 84.021% whereas corresponding FPR as 3.472%, 4.011%, 4.818%, 6.932%, and 7.813% for reflection with rotation attack of 100°, 150°, 200°, 250°, and 300°, respectively as shown in Table 7. The qualitative outcomes obtained using our method are shown in Fig. 15. The quantitative results obtained using proposed approach shows the slight decline in F-Measure and rise in FPR for increasing degree of rotations over reflected tampered region.

3.3.8. Reflection with scaling attack

To assess performance of the proposed scheme for the detection of

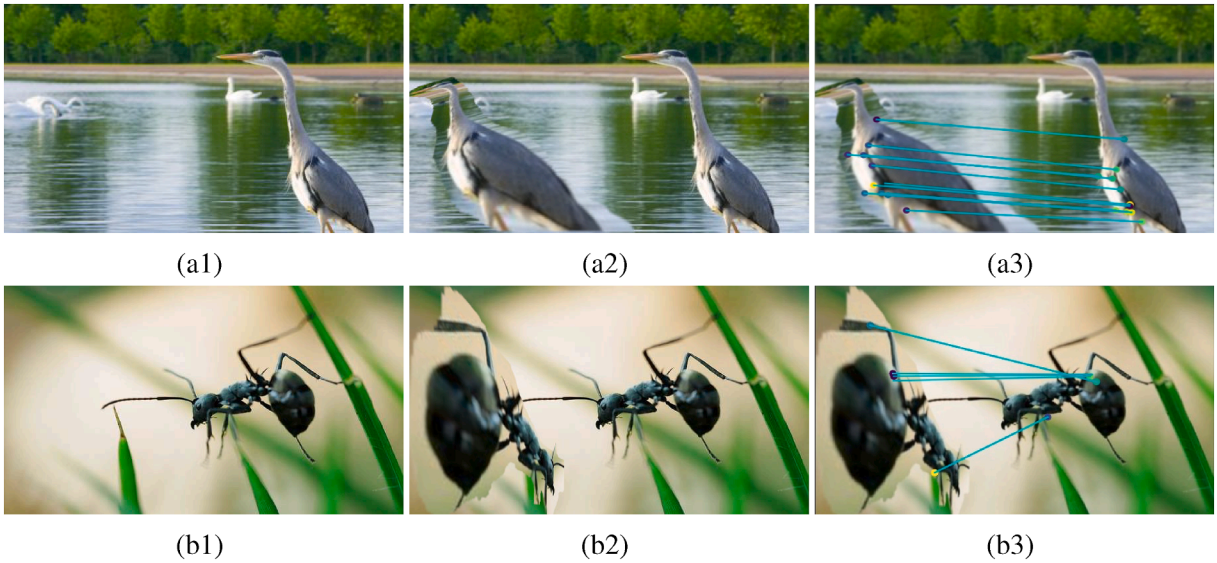

Fig. 18. {(a1), (b1)} Authentic images collected from CASIA TIDE v2.0 (Dong et al., 2013) dataset. {(a2), (b2)} Tampered images with forged regions sustaining non-affine transformation and reflection with non-affine transformation attack, respectively. {(a3), (b3)} Forgery detection results obtained using proposed approach.

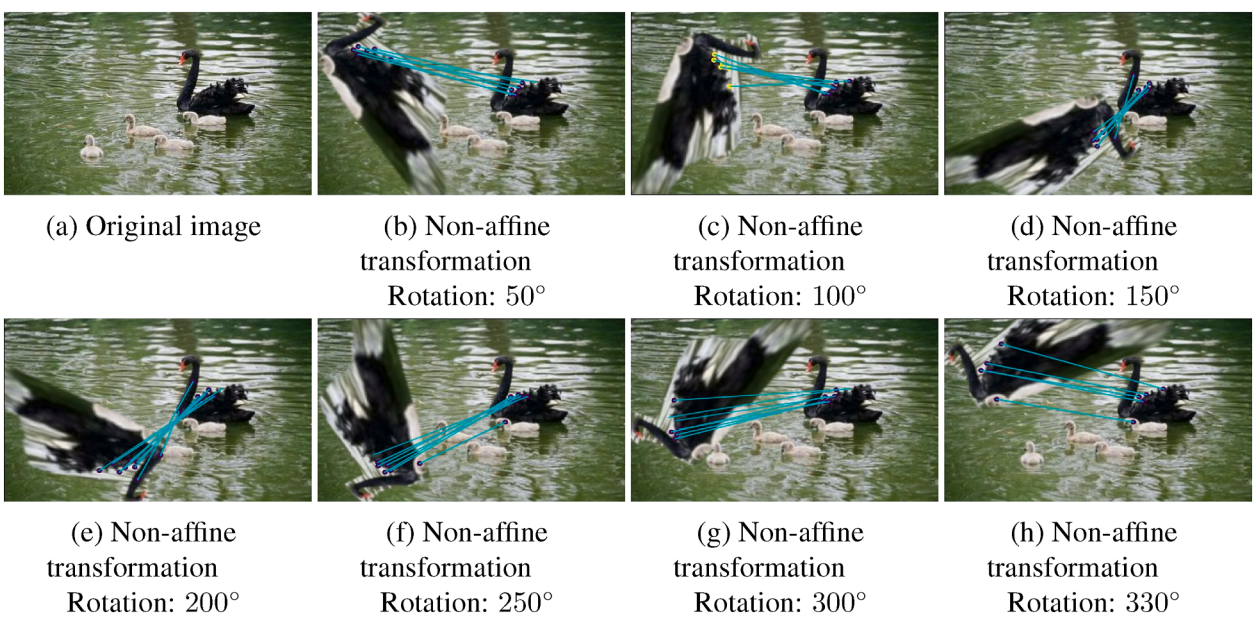

Fig. 19. (a) Original image taken from CASIA TIDE v2.0 (Dong et al., 2013) dataset. {(b)–(h)} Forgery detection results obtained by utilizing proposed technique over tampered images sustaining non-affine transformation with rotation attack over manipulated region.

tampered images sustaining reflection with scaling attack, we have formulated forged images with downscaling and upscaling attacks. For downscaling, reflected section is downscaled to 50% and 20%, while for upscaling reflected region is upscaled to 150%, 200%, and 250% as shown in Table 4. We have constructed 40 tampered images for each of the scaling extent, hence for experimentation, we have obtained 200 tampered images and mixed them with 200 authentic images. Proposed approach has achieved F-Measure as 84.006%, 90.816%, 93.114%, 85.703%, and 86.121% while corresponding FPR as 8.813%, 5.331%, 3.052%, 6.183%, and 7.045% where 20%, 50%, 150%, 200%, and 250% scaling is applied over reflected region, respectively as shown in Table 7. The quantitative results show that F-Measure slightly declines and FPR rises for downscaling from 50% to 20%. Similarly, for upscaling from 150% to 250%, F-Measure decreases while FPR increases which shows slight decline in performance for high extent of upscaling and downscaling attacks. The qualitative results obtained using our technique are shown in Fig. 16.

3.3.9. *Reflection with shearing attack*

We have constructed 200 forged images bearing reflection with shearing attack where 40 tampered images are created for each of the 50, 75, 100, 125, and 150 shearing extent over reflected region. These tampered images are mixed with 200 authentic images to evaluate performance of the proposed scheme. Our method has achieved high F-Measure as 92.048%, 91.136%, 89.214%, 86.375%, and 84.119% while corresponding FPR as 4.293%, 4.865%, 5.379%, 6.201%, and 8.364% for shearing extent 50, 75, 100, 125, and 125, respectively as shown in Table 7. The quantitative results show slight degradation in performance of the proposed approach for increasing extent of shearing operation. The qualitative results obtained for forgery detection are shown in

Table 8
Quantitative results of copy-move forgery detection obtained using proposed technique when images are forged using non-affine transformation with rotation, scaling, and shearing attacks.

Attack	Attack scale	F-Measure (%)	FPR (%)
Non-affine Transformation with rotation	100°	93.168	3.026
	150°	92.851	4.393
	200°	90.237	5.007
	250°	89.142	5.802
	300°	86.520	7.118
Non-affine Transformation with Scaling	20%	83.427	8.924
	50%	86.053	6.751
	150%	90.185	2.084
	200%	88.106	5.231
	250%	85.724	7.019
Non-affine Transformation with Shearing	50	91.823	3.084
	75	89.441	4.106
	100	88.925	6.275
	125	87.637	7.019
	150	85.113	8.729

Fig. 17.

3.3.10. Non-affine transformation attack

The non-affine transformation changes the perspective of the selected region of image. Such transformations do not preserve the length, angle, or parallelism of image objects. To evaluate performance of the proposed algorithm, we have constructed 200 forged images sustaining tampered region with non-affine transformation attack. These images are mixed with 200 authentic images. Our method has achieved high F-Measure and low FPR as 96.183% and 2.174%, respectively as shown in Table 6 which shows the robustness of the proposed method against non-affine transformation attacks. The qualitative result obtained using proposed technique for forgery detection is represented in Fig. 18 {(a1)–(a3)}.

3.3.11. Non-affine transformation with reflection attack

We have formulated 200 tampered images consisting of forged region with non-affine transformation over reflected region. Further, we have mixed 200 authentic images with 200 tampered images. Our method has achieved F-Measure and FPR as 88.362% and 3.046%, respectively as shown in Table 6 which represents the robustness of proposed scheme against non-affine tansformation with reflection attack. The qualitative result for forgery detection obtained using proposed technique is shown in Fig. 18 {(b1)–(b3)}.

3.3.12. Non-affine transformation with rotation attack

To analyze the performance of our scheme for forgery detection in images sustaining tampered region with non-affine transformation and rotation attack, we have formulated 40 images containing each of the 100°, 150°, 200°, 250°, and 300° rotations over non-affine transformed region. We have performed experiment over 400 images consisting of 200 authentic and 200 forged images. Our method shows robustness towards non-affine transformation with rotation attack and achieved high F-Measure as 93.168%, 92.851%, 90.237%, 89.142%, and 86.520% while corresponding FPR as 3.026%, 4.393%, 5.007%, 5.802%, and 7.118% for 100°, 150°, 200°, 250°, and 300° rotations over non-affine transformed region, respectively as shown in Table 8. The quantitative outcomes show that performance of the proposed approach slightly declines with increase in rotation extent over non-affine transformed region. The qualitative results obtained for forgery detection using our method are portrayed in Fig. 19.

3.3.13. Non-affine transformation with scaling attack

We have constructed 40 tampered images for each of the 20%, 50%, 150%, 200%, and 250% scaling over non-affine transformed manipulated region, i.e., both downscaling (20%, 50%) and upscaling (150%, 200%, 250%) operations are applied over forged region. To evaluate performance of the proposed technique for forgery detection, we have utilized 400 images consisting of 200 original and 200 manipulated images. Our method shows robustness toward non-affine transformation with scaling attack. Substantially high F-Measure as 84.006%, 90.816%, 93.114%, 85.703%, and 86.121% while corresponding FPR as 8.813%,

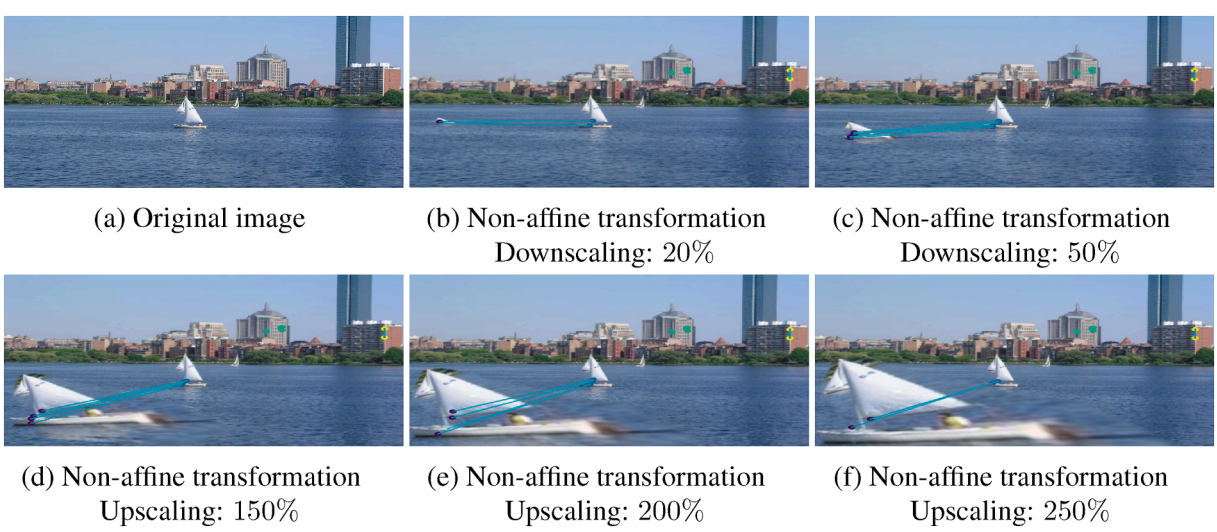

Fig. 20. (a) Original image collected from Ardizzone (Ardizzone et al., 2015a) dataset. {(b)–(f)} Forgery detection outcomes obtained using our approach when non-affine transformation with scaling attack is enforced over duplicated region.

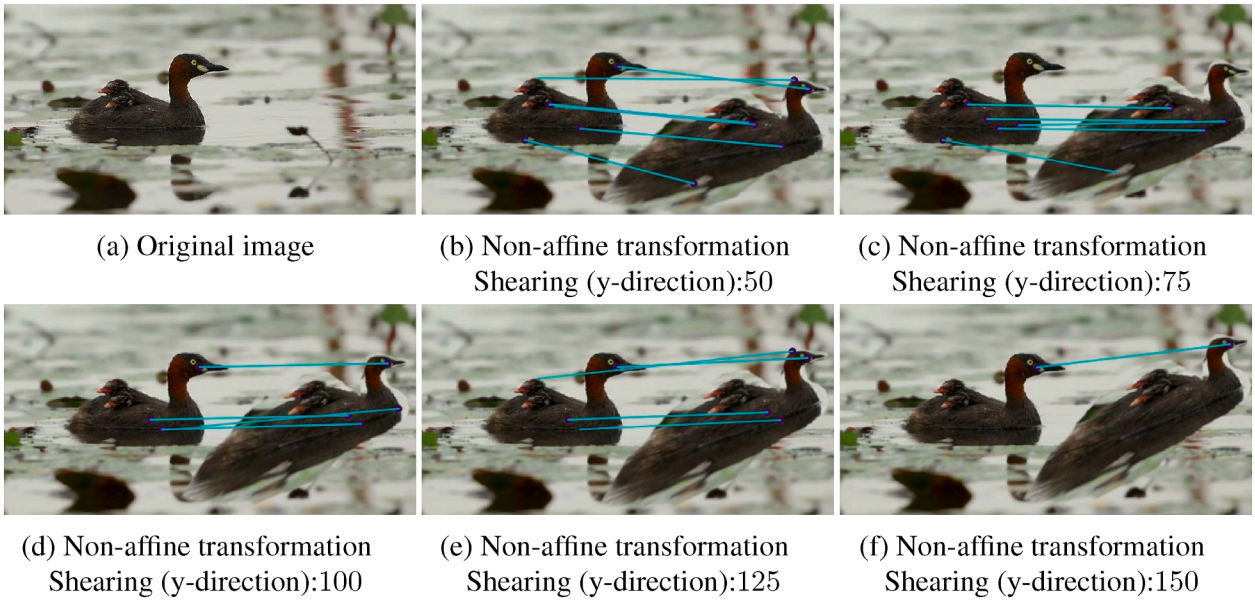

Fig. 21. (a) Original image collected from CASIA TIDE v2.0 (Dong et al., 2013) dataset. {(b)–(f)} Forgery detection outcomes obtained using our approach when tampered images sustain non-affine transformation with shearing attack over manipulated region.

Table 9
Quantitative results obtained using proposed copy-move forgery detection technique when forged images are sustaining different post-processing attacks.

Attack	Attack scale	F-Measure (%)	FPR (%)
JPEG compression (Quality level)	2	85.011	9.629
	5	88.254	7.335
	10	90.513	5.468
	20	93.116	4.029
	30	95.782	3.118
Zoom motion blur (Blurring factor)	0.03	96.186	2.784
	0.06	94.362	3.429
	0.09	91.148	5.461
	0.12	89.731	6.027
	0.15	85.147	8.356
Spread noise addition (Randomness)	10	93.114	3.267
	15	92.315	3.913
	20	91.625	4.394
	25	90.431	6.048
	30	87.183	8.529

5.331%, 3.052%, 6.183%, and 7.045% is obtained for 20%, 50%, 150%, 200%, and 250% scaling over non-affine transformed region, respectively as shown in Table 8. The quantitative results show that performance of our approach slighly declines when downscaling increases from 50% to 20%. Similarly, performance of the proposed scheme slighly declines for increase in upscaling from 150% to 250%. The pictorial results of forgery detection obtained for different extent of scaling over non-affine transformed region are shown in Fig. 20.

3.3.14. Non-affine transformation with shearing attack

To analyze performance of the proposed scheme towards detection of forged images containing non-affine tansformation with shearing attack over tampered region, we have constructed 40 tampered images for each of the 50, 75, 100, 125, and 150 shearing magnitude in horizontal and vertical directions. We have collected 400 images for experimentation consisting of 200 original and 200 tampered images. Our method has achieved considerably high F-Measure as 91.823%, 89.441%, 88.925%, 87.637%, and 85.113% while corresponding FPR as 3.084%, 4.106%, 6.275%, 7.019%, and 8.729% for shearing extent of 50, 75, 100, 125, and 150 over non-affine transformed region, respectively as shown in Table 8. The quantitative results reflect that performance of our approach slighly declines when extent of shearing increases from 50 to 150. The pictorial results for forgery detection obtained using proposed scheme are shown in Fig. 21.

3.3.15. JPEG compression attack

To evaluate performance of the proposed technique, we have considered forged images with five different categories for JPEG compression of 2, 5, 10, 20, and 30 quality levels as shown in Table 4. For each quality level of JPEG compression, we have constructed 40 tampered images. We have collected 400 images for experimentation consisting of 200 forged images sustaining JPEG compression attack and 200 authentic images. Our method has achieved sunbstantially high F-Measure as 85.011%, 88.254%, 90.513%, 93.116%, and 95.782% while corresponding FPR as 9.629%, 7.335%, 5.468%, 4.029%, and 3.118% for JPEG compression quality levels 2, 5, 10, 20, and 30, respectively as shown in Table 9. The quantitative results indicate that performance of the proposed technique slighly degrades with increasing JPEG compression from 30 to 2 quality level where low quality level represents high JPEG compression. The qualitative outcomes for forgery detection obtained using our technique are presented in Fig. 22.

3.3.16. Zoom motion blur attack

Zoom motion blur filter is also known as Radial filter. It produces blur in every direction around a center point. The default center point is the center of the selected image region. We have also used center of the selected image region as the center point. To construct forged images sustaining tampered region with zoom motion blur attack, five blurring factors as 0.03, 0.06, 0.09, 0.12, and 0.15 are considered as shown in Table 4. We have constructed 40 tampered images corresponding to each of the blurring factor. To analyze performance of the proposed technique against zoom motion blur attack, we have collected 400 images where 200 images are tampered and rest of the 200 images are authentic. Proposed technique has achieved high F-Measure as 96.186%, 94.362%, 91.148%, 89.731%, and 85.147% while corresponding FPR as 2.784%, 3.429%, 5.461%, 6.027%, and 8.356% for blurring factors of 0.03, 0.06, 0.09, 0.12, and 0.15, respectively as

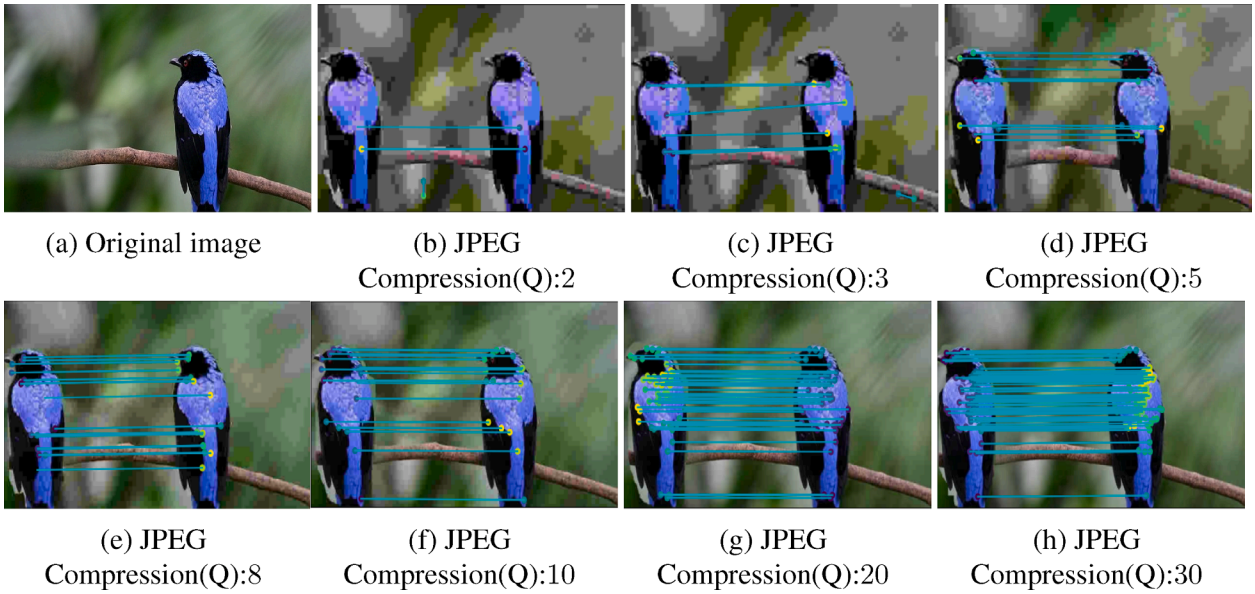

Fig. 22. (a) Original image collected from CASIA TIDE v2.0 (Dong et al., 2013) dataset. {(b)–(h)} Forgery detection results obtained using proposed approach over tampered images sustaining JPEG compression attack.

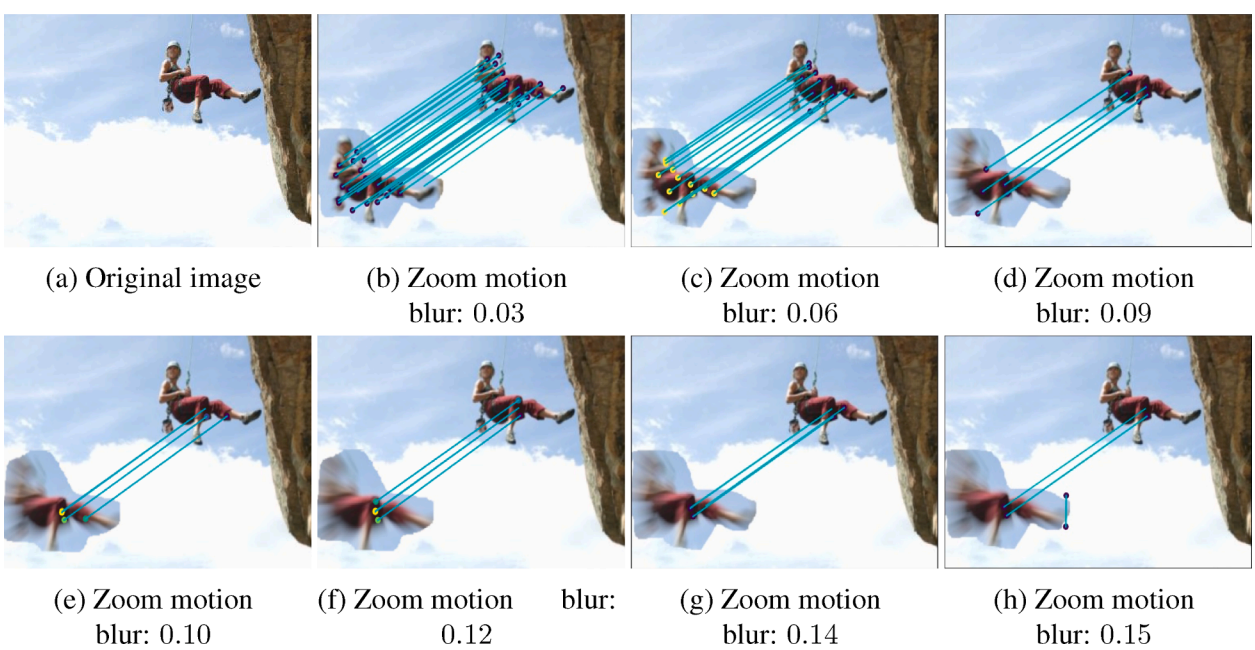

Fig. 23. (a) Original image collected from CASIA TIDE v2.0 (Dong et al., 2013) dataset. {(b)–(h)} Forgery detection results obtained using proposed approach over tampered images sustaining manipulated region with zoom motion blur attack.

shown in Table 9. The quantitative results imply that performance of the proposed approach slightly degrades when extent of zoom motion blur is increased over manipulated region of image. The pictorial results obtained for forgery detection corresponding to zoom motion blur attack are shown in Fig. 23.

3.3.17. Spread noise addition attack

The spread noise filter swaps every pixel of the selected region with another randomly selected pixel by a user defined amount. In this noise filter, user can set the distance for the pixels along horizontal and vertical axis according to which pixels can move. Number of seed points control the randomness of spreading. We have fixed number of seed points at 5 while distance along x and y direction for pixels movement varies. We have constructed 200 tampered images corresponding to spread noise addition attack where 40 images belong to each of the randomness factor 10, 15, 20, 25, and 30, respectively. To observe performance of the proposed approach against spread noise addition attack, we have gathered 400 images which include 200 original and 200 forged images. Our method has achieved high F-Measure as 93.114%, 92.315%, 91.625%, 90.431%, and 87.183% while corresponding FPR as 3.267%, 3.913%, 4.394%, 6.048%, and 8.529% for randomness factor 10, 15, 20, 25, and 30, respectively as shown in Table 9. The quantitative results show that with increase in randomness factor, performance of our method slightly degrades. The qualitative

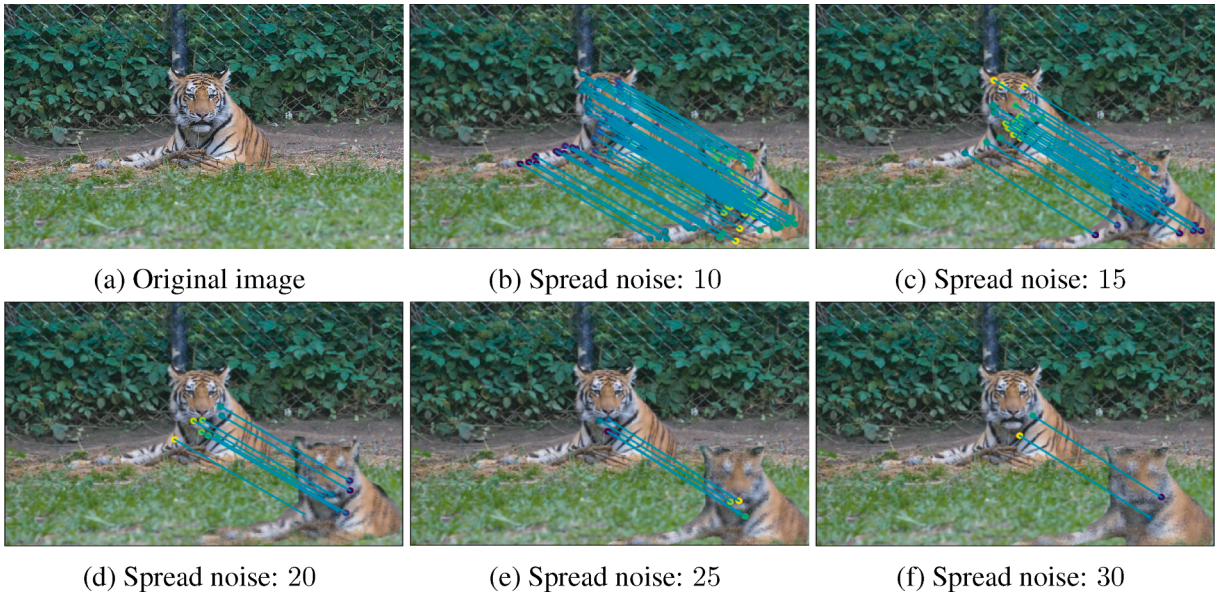

Fig. 24. (a) Authentic image picked up from Ardizzone (Ardizzone et al., 2015a) dataset. {(b)–(f)} Forgery detection results obtained using proposed technique over tampered images sustaining manipulated region with spread noise addition attack.

forgery detection results are shown in Fig. 24 where tampered images contain manipulated region suffering from spread noise addition attack.

3.4. Comparative analysis

To analyze the comparative performance of the proposed technique with respect to state- of-the-art methods, we have compared the performance of the proposed approach with several existing methods based on SIFT (Amerini et al., 2011b), ANMS + DAISY (Guo et al., 2013), Harris + BRISK (Isaac & Wilscy, 2015), ORB (Zhu, Shen, & Chen, 2016), KAZE + SIFT (Yang et al., 2017), SIFT-Symmetry (Warif et al., 2017), AKAZE + SURF (Wang, Zhang, & Zhou, 2018a), and SURF (Soni, Das, & Thounaojam, 2019) techniques for detection of copy-move forged images sustaining various attacks. For comparative analysis, we have utilized F-Measure and FPR parameters.

The quantitative comparative results obtained corresponding to detection of forged images bearing reflection, combination of reflection, non-affine transformation, and non-affine transformation with reflection attacks are shown in Table 10 {(a)–(d)}. For detection of tampered images suffering from erosion, dilation, RGB color addition, and multiple copy-move attack, the quantitative comparative results are shown in Table 11 {(a)–(d)}. The comparative quantitative results based on F-Measure and FPR parameters are shown in Fig. 25 for detection of forged images sustaining reflection with rotation, scaling, and shearing attacks. For detection of forged images bearing non-affine transformation with rotation, scaling, and shearing attacks, the comparative quantitative results are presented in Fig. 26. The quantitative comparative results incurred corresponding to the detection of manipulated images suffering from JPEG compression, zoom motion blur, and spread noise addition attacks are shown in Fig. 27. The comparative results based on F-Measure and FPR shows that our method has outperformed state-of-the-art methods for detection of forged images sustaining various complicated attacks. CenSurE detector and LIPID descriptor provided splendid combination of keypoints and their feature descriptors which rendered robustness against several post-processing and geometrical attacks. The LIPID descriptor also facilitated significant robustness against reflection attack, while most of the keypoint descriptors such as SIFT, SURF, ORB, BRISK, DAISY, KAZE, etc. holds poor potential towards identification of forged images containing reflection attack.

We have compared proposed approach with state-of-the-art methods based on the number of keypoints and length of the keypoint descriptor as shown in Table 12. The proposed approach has consumed 63.725 seconds in forgery detection for 100 images with dimension 512×512 pixels as shown in Table 13. Primarily, the four factors contribute in fast detection of copy-move forged images, namely keypoint extraction, descriptor computation, matching of keypoint descriptors, and cluster formation for matched descriptors. In proposed approach, the CenSurE detector has provided fast extraction of relevant keypoints from tampered images. LIPID has yielded comparatively shortest length of feature descriptor as 25. The small length of keypoint descriptor has contributed in reduced computational overhead involved in proposed copy-move forgery detection technique. k-NN search technique with BBF algorithm has contributed in quick matching of similar descriptors. In addition to it, Fuzzy C-Means clustering has also accelerated the process of cluster formation for similar keypoints to identify forged regions. Our technique requires least time for forgery detection as compared to state-of-the-art techniques as shown in Table 14. Overall, the comparative results show that our technique is a novel, robust, and efficient approach for copy-move image forgery detection.

4. Conclusion

The copy-move forged images are challenging to detect as the forged region may undergo several transformations to make the forgery detection unvanquishable for human eyes. Hence, we have proposed a robust approach for copy-move image forgery detection using CenSurE keypoint extractor and LIPID keypoint descriptor. To perform matching between keypoints, k-NN approach is utilized. We have used Fuzzy C-

Table 10
Comparative quantitative results of copy-move forgery detection obtained using proposed and different existing techniques when images are manipulated using reflection, combination of reflection, non-affine transformation, and non-affine transformation with reflection attack.

(a) Reflection attack

Method	F-Measure (%)	FPR (%)
SIFT-Symmetry (Warif et al., 2017)	82.149	6.116
AKAZE + SURF (Wang et al., 2018a)	39.343	10.771
KAZE + SIFT (Yang et al., 2017)	35.501	12.136
ANMS + DAISY (Guo et al., 2013)	31.766	7.896
ORB (Zhu et al., 2016)	28.367	15.610
Harris + BRISK (Isaac & Wilscy, 2015)	24.711	17.348
SIFT (Amerini et al., 2011b)	17.182	11.169
SURF (Soni et al., 2019)	13.893	13.231
Proposed	**93.236**	**2.343**

(b) Combination of reflection attack

Method	**F-Measure (%)**	**FPR (%)**
SIFT-Symmetry (Warif et al., 2017)	80.861	9.374
KAZE + SIFT (Yang et al., 2017)	38.114	13.602
AKAZE + SURF (Wang et al., 2018a)	32.482	17.194
Harris + BRISK (Isaac & Wilscy, 2015)	27.534	19.106
ANMS + DAISY (Guo et al., 2013)	25.331	11.684
ORB (Zhu et al., 2016)	21.109	16.739
SIFT (Amerini et al., 2011b)	17.532	6.013
SURF (Soni et al., 2019)	12.734	8.690
Proposed	**94.302**	**4.021**

(c) Non-affine transformation attack

Method	**F-Measure (%)**	**FPR (%)**
KAZE + SIFT (Yang et al., 2017)	85.044	8.293
AKAZE + SURF (Wang et al., 2018a)	81.747	10.414
ANMS + DAISY (Guo et al., 2013)	76.208	12.603
SIFT-Symmetry (Warif et al., 2017)	70.209	15.863
SIFT (Amerini et al., 2011b)	67.114	10.432
SURF (Soni et al., 2019)	62.372	13.604
ORB (Zhu et al., 2016)	54.701	7.183
Harris + BRISK (Isaac & Wilscy, 2015)	48.318	6.041
Proposed	**96.183**	**2.174**

(d) Non-affine transformation with reflection attack

Method	**F-Measure (%)**	**FPR (%)**
SIFT-Symmetry (Warif et al., 2017)	79.609	7.196
AKAZE + SURF (Wang et al., 2018a)	28.197	13.210
KAZE + SIFT (Yang et al., 2017)	25.013	15.386
ANMS + DAISY (Guo et al., 2013)	21.304	12.814
Harris + BRISK (Isaac & Wilscy, 2015)	19.382	10.302
ORB (Zhu et al., 2016)	14.109	8.163
SURF (Soni et al., 2019)	9.783	6.488
SIFT (Amerini et al., 2011b)	7.183	5.209
Proposed	**88.362**	**3.046**

Table 11
Comparative quantitative results of copy-move forgery detection obtained using proposed and existing techniques when tampered images sustain erosion, dilation, RGB color addition, and multiple copy-move attacks.

(a) Erosion attack

Method	F-Measure (%)	FPR (%)
KAZE + SIFT (Yang et al., 2017)	88.116	7.804
SIFT-Symmetry (Warif et al., 2017)	82.463	9.810
AKAZE + SURF (Wang et al., 2018a)	80.208	11.361
ANMS + DAISY (Guo et al., 2013)	77.131	14.013
SURF (Soni et al., 2019)	72.609	17.183
SIFT (Amerini et al., 2011b)	68.280	13.226
ORB (Zhu et al., 2016)	65.438	12.209
Harris + BRISK (Isaac & Wilscy, 2015)	61.109	10.568
Proposed	**95.138**	**3.177**

(b) Dilation attack

Method	**F-Measure (%)**	**FPR (%)**
SIFT-Symmetry (Warif et al., 2017)	84.112	7.334
KAZE + SIFT (Yang et al., 2017)	80.282	9.720
AKAZE + SURF (Wang et al., 2018a)	78.160	12.113
SURF (Soni et al., 2019)	74.629	14.300
ANMS + DAISY (Guo et al., 2013)	71.089	15.329
ORB (Zhu et al., 2016)	68.104	18.643
SIFT (Amerini et al., 2011b)	63.004	13.604
Harris + BRISK (Isaac & Wilscy, 2015)	60.723	9.028
Proposed	**96.207**	**2.613**

(c) RGB color addition attack

Method	**F-Measure (%)**	**FPR (%)**
KAZE + SIFT (Yang et al., 2017)	87.634	6.313
SIFT-Symmetry (Warif et al., 2017)	83.207	8.106
AKAZE + SURF (Wang et al., 2018a)	80.173	9.447
SIFT (Amerini et al., 2011b)	76.409	12.309
ORB (Zhu et al., 2016)	72.901	14.626
SURF (Soni et al., 2019)	70.832	11.318
ANMS + DAISY (Guo et al., 2013)	67.934	16.181
Harris + BRISK (Isaac & Wilscy, 2015)	62.782	15.793
Proposed	**94.631**	**2.733**

(d) Multiple copy-move attack

Method	**F-Measure (%)**	**FPR (%)**
KAZE + SIFT (Yang et al., 2017)	84.215	7.033
AKAZE + SURF (Wang et al., 2018a)	81.703	11.318
SIFT-Symmetry (Warif et al., 2017)	77.360	12.709
ANMS + DAISY (Guo et al., 2013)	75.349	14.625
SURF (Soni et al., 2019)	72.183	17.024
ORB (Zhu et al., 2016)	70.372	19.267
SIFT (Amerini et al., 2011b)	67.814	15.322
Harris + BRISK (Isaac & Wilscy, 2015)	64.129	20.136
Proposed	**95.216**	**4.023**

Means clustering to form cluster of matched keypoints. To remove outliers incurred during forgery detection process, we have used RANSAC algorithm. Proposed method can effectively detect forged images with various attacks based on reflection and non-affine transformation with geometrical attacks. Our approch has achieved improved results as compared to state-of-the-art methods. Proposed scheme has also proved to be robust against erosion, dilation, RGB color addition, spread noise addition, and zoom motion blur attacks which have not been analyzed by state-of-the-art techniques. Our technique is a novel and highly efficient approach for copy-move image forgery detection as it has overwhelmed state-of-the-art methods and consumed least computation time in forgery detection. As future work, we will investigate the methods to effectively detect tampered images which are manipulated using image splicing forgery technique.

CRediT authorship contribution statement

Anuja Dixit: Conceptualization, Methodology, Software, Data curation, Writing - original draft. **Soumen Bag:** Visualization, Investigation, Writing - review & editing, Supervision.

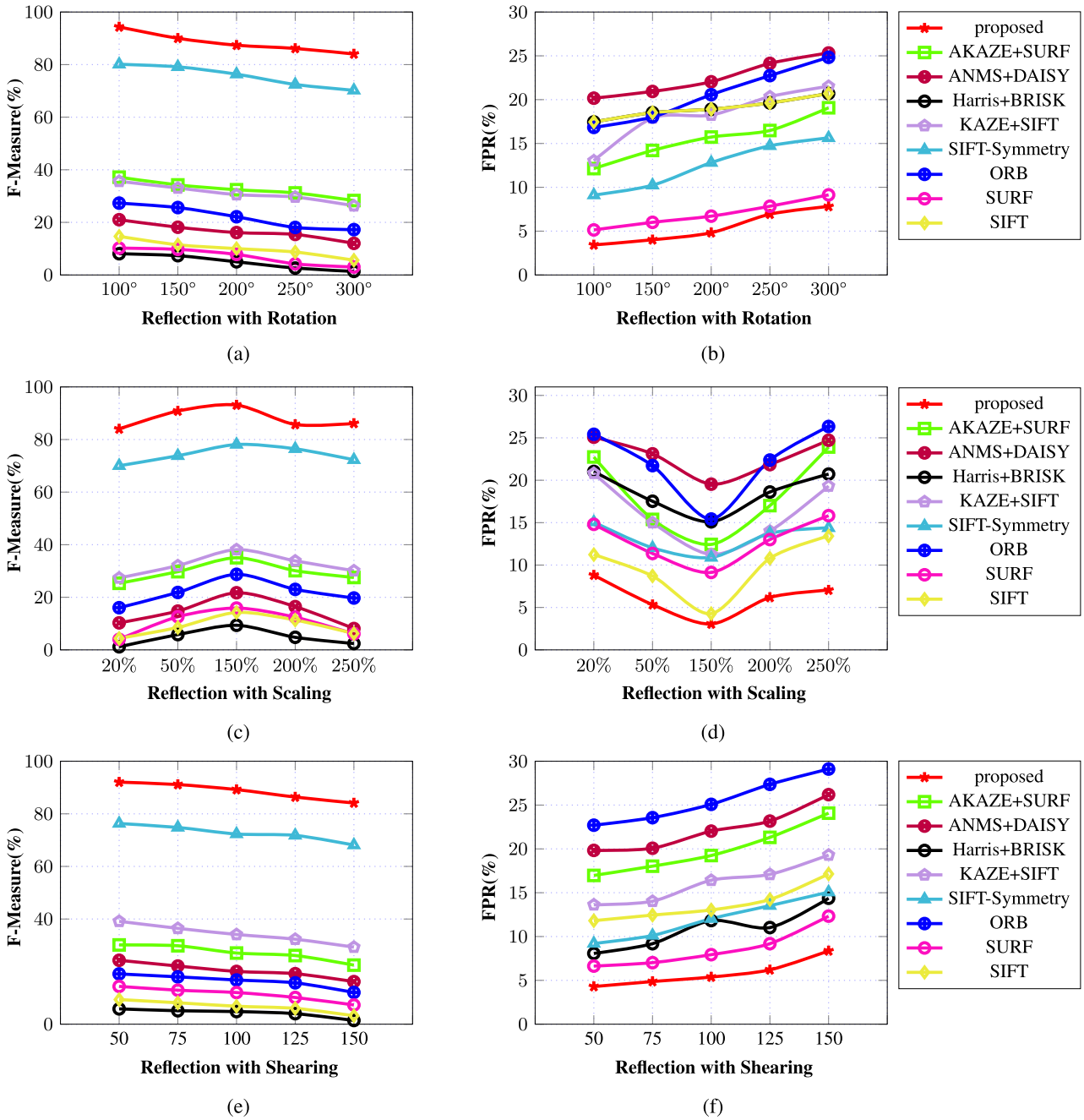

Fig. 25. (a), (c), and (e) shows comparative results for F-Measure(%) when copy-move tampered image sustain reflection with rotation, scaling, and shearing attacks, respectively whereas (b), (d), and (f) shows corresponding FPR(%) results..

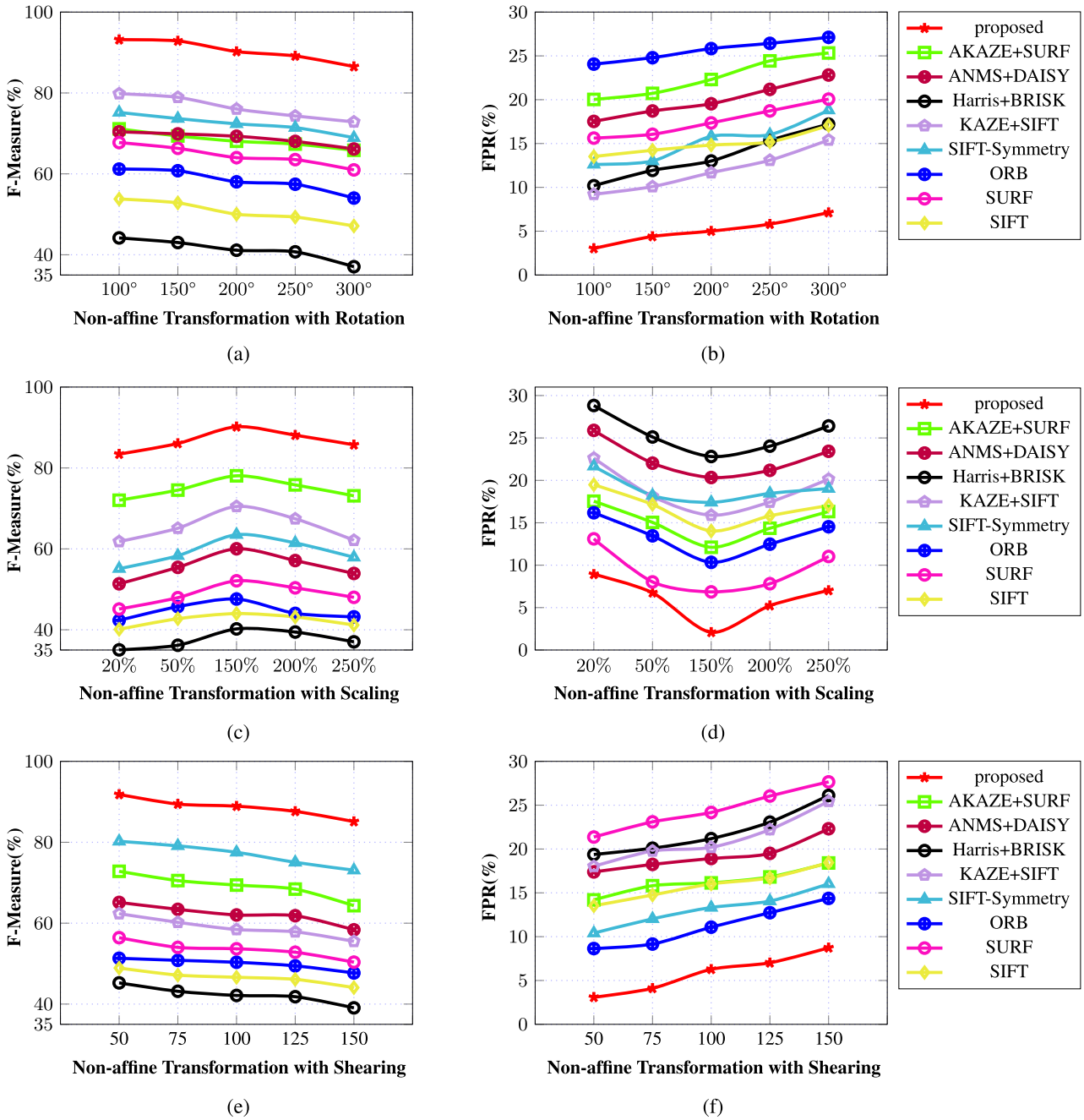

Fig. 26. (a), (c), and (e) shows comparative results for F-Measure(%) when copy-move tampered image suffer from non-affine transformation with rotation, scaling, and shearing attack, respectively whereas (b), (d), and (f) shows corresponding FPR(%) results.

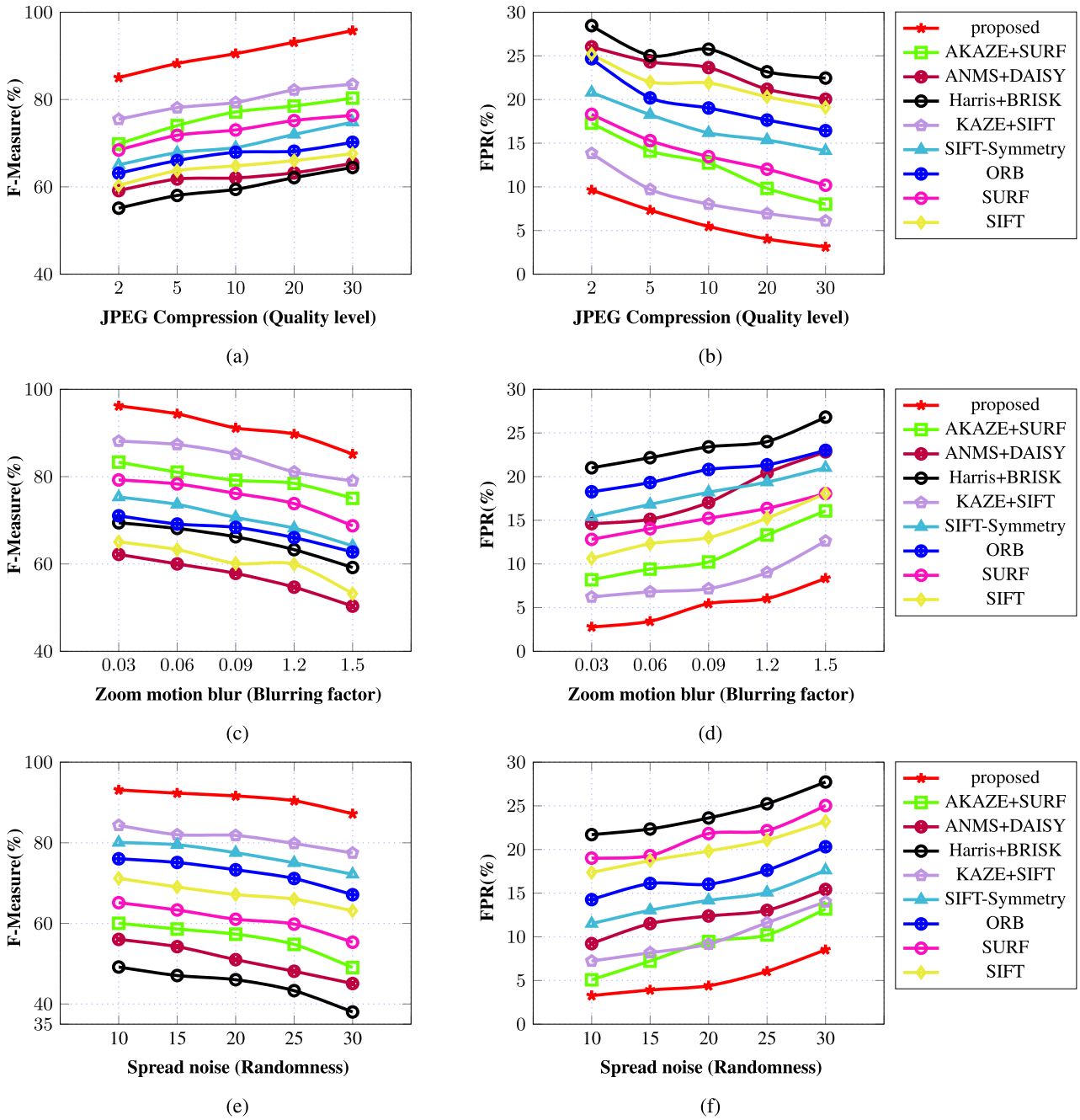

Fig. 27. (a), (c), and (e) represents comparative results for F-Measure(%) when copy-move forged images bear JPEG compression, zoom motion blur, and spread noise attack, respectively whereas (b), (d), and (f) shows corresponding FPR(%) results.

Table 12
Comparision based on number of keypoints and descriptor length obtained using different existing techniques over 512 × 512 pixels image.

Method	Keypoint extractor	Keypoint descriptor	No. of keypoints	Length of descriptor
KAZE + SIFT (Yang et al., 2017)	KAZE, SIFT	KAZE, SIFT	KAZE:276, SIFT:1535	KAZE:61, SIFT:128
SIFT (Amerini et al., 2011b)	SIFT	SIFT	SIFT:1535	SIFT:128
SIFT-Symmetry (Warif et al., 2017)	SIFT	Mirror SIFT	SIFT: 1535	Mirror SIFT:128
AKAZE + SURF (Wang et al., 2018a)	AKAZE, SURF	AKAZE, SURF	AKAZE:276, SURF:1077	AKAZE:61, SURF:64
ANMS + DAISY (Guo et al., 2013)	ANMS	DAISY	ANMS:1139	DAISY:104
SURF (Soni et al., 2019)	SURF	SURF	SURF:1077	SURF:64
Harris + BRISK (Isaac & Wilscy, 2015)	Harris	BRISK	Harris:555	BRISK:64
ORB (Zhu et al., 2016)	ORB	ORB	ORB:1313	ORB:32
Proposed	**CenSurE**	**LIPID**	**CenSurE:1207**	**LIPID:25**

Table 13
Time consumed in each step of proposed copy-move forgery detection technique for 100 images with dimension 512 × 512 pixels.

Steps	Time (seconds)
Preprocessing	0.717
Keypoint extraction	19.438
Descriptor computation	11.076
Keypoint matching	15.813
Keypoint clustering	9.521
Affine transformation estimation	7.160

Table 14
Comparision of time required by different techniques in copy-move forgery detection of 100 images with dimension 512 × 512 pixels.

Method	Time (seconds)
KAZE + SIFT (Yang et al., 2017)	278.763
AKAZE + SURF (Wang et al., 2018a)	235.644
ANMS + DAISY (Guo et al., 2013)	181.383
SIFT-Symmetry (Warif et al., 2017)	170.035
SIFT (Amerini et al., 2011b)	155.128
Harris + BRISK (Isaac & Wilscy, 2015)	145.469
SURF (Soni et al., 2019)	117.662
ORB (Zhu et al., 2016)	91.558
Proposed	**63.725**

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Agrawal, M., Konolige, K., & Blas, M. R. (2008). Censure: Center surround extremas for realtime feature detection and matching. In *European Conference on Computer Vision* (pp. 102–115). Springer.

Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., & Serra, G. (2011a). *Micc dataset*. http://lci.micc.unifi.it/labd/2015/01/copy-move-forgery-detection-and-localization/.

Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., & Serra, G. (2011b). A sift-based forensic method for copy–move attack detection and transformation recovery. *IEEE Transactions on Information Forensics and Security, 6*(3), 1099–1110.

Ardizzone, E., Bruno, A., & Mazzola, G. (2015a). *Ardizzone dataset*. http://www.diid.unipa.it/cvip/?pageid=48.

Ardizzone, E., Bruno, A., & Mazzola, G. (2015b). Copy–move forgery detection by matching triangles of keypoints. *IEEE Transactions on Information Forensics and Security, 10*(10), 2084–2094.

Bashar, M., Noda, K., Ohnishi, N., & Mori, K. (2010). Exploring duplicated regions in natural images. *IEEE Transactions on Image Processing, 99*, 1–40.

Beis, J. S., & Lowe, D. G. (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proceedings of IEEE Computer Vision and Pattern Recognition* (pp. 1000–1006). IEEE.

Bravo-Solorio, S., & Nandi, A. K. (2011). Exposing duplicated regions affected by reflection, rotation and scaling. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1880–1883). IEEE.

Chen, L., Lu, W., Ni, J., Sun, W., & Huang, J. (2013). Region duplication detection based on harris corner points and step sector statistics. *Journal of Visual Communication and Image Representation, 24*(3), 244–254.

Christlein, V., Riess, C., & Angelopoulou, E. (2010). On rotation invariance in copy-move forgery detection. In *IEEE International Workshop on Information Forensics and Security* (pp. 1–6). IEEE.

Christlein, V., Riess, C., Jordan, J., Riess, C., & Angelopoulou, E. (2012a). An evaluation of popular copy-move forgery detection approaches. *IEEE Transactions on Information Forensics and Security, 7*(6), 1841–1854.

Christlein, V., Riess, C., Jordan, J., Riess, C., & Angelopoulou, E. (2012b). *Imd dataset*. https://www5.cs.fau.de/research/data/image-manipulation/.

Cozzolino, D., Poggi, G., & Verdoliva, L. (2015). Efficient dense-field copy–move forgery detection. *IEEE Transactions on Information Forensics and Security, 10*(11), 2284–2297.

Dixit, A., & Bag, S. (2019). Utilization of hog-svd based features with connected component labeling for multiple copy-move image forgery detection. In *2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA)* (pp. 1–9). IEEE.

Dixit, A., & Bag, S. (2020a). Composite attacks-based copy-move image forgery detection using akaze and fast with automatic contrast thresholding. *IET Image Processing, 14*(17), 4528–4542.

Dixit, A., & Bag, S. (2020b). Copy–move image forgery detection using gray-tones with texture description. In *Proceedings of 3rd International Conference on Computer Vision and Image Processing* (pp. 75–86). Springer.

Dixit, A., & Bag, S. (2020c). Utilization of edge operators for localization of copy-move image forgery using wld-hog features with connected component labeling. *Multimedia Tools and Applications, 79*(35), 26061–26097.

Dong, J., Wang, W., & Tan, T. (2013). Casia image tampering detection evaluation database. In *IEEE China Summit and International Conference on Signal and Information Processing* (pp. 422–426). IEEE.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM, 24*(6), 381–395.

Fridrich, A. J., Soukal, B. D., & Lukáš, A. J. (2003). Detection of copy-move forgery in digital images. In *Proceedings of Digital Forensic Research Workshop. Citeseer*.

Gong, M., Liang, Y., Shi, J., Ma, W., & Ma, J. (2012). Fuzzy c-means clustering with local information and kernel metric for image segmentation. *IEEE Transactions on Image Processing, 22*(2), 573–584.

Guo, J.-M., Liu, Y.-F., & Wu, Z.-J. (2013). Duplication forgery detection using improved daisy descriptor. *Expert Systems with Applications, 40*(2), 707–714.

Henderson, C., & Izquierdo, E. (2016). Symmetric stability of low level feature detectors. *Pattern Recognition Letters, 78*, 36–40.

Isaac, M. M., & Wilscy, M. (2015). Copy-move forgery detection based on harris corner points and brisk. In *Proceedings of the Third International Symposium on Women in Computing and Informatics* (pp. 394–399).

Jeronymo, D. C., Borges, Y. C. C., & dos Santos Coelho, L. (2017). Image forgery detection by semi-automatic wavelet soft-thresholding with error level analysis. *Expert Systems with Applications, 85*, 348–356.

Jin, G., & Wan, X. (2017). An improved method for sift-based copy–move forgery detection using non-maximum value suppression and optimized j-linkage. *Signal Processing: Image Communication, 57*, 113–125.

Kosaka, N., & Ohashi, G. (2015). Vision-based nighttime vehicle detection using censure and svm. *IEEE Transactions on Intelligent Transportation Systems, 16*(5), 2599–2608.

Li, J., Li, X., Yang, B., & Sun, X. (2014). Segmentation-based image copy-move forgery detection scheme. *IEEE Transactions on Information Forensics and Security, 10*(3), 507–518.

Li, J., Yang, F., Lu, W., & Sun, W. (2017). Keypoint-based copy-move detection scheme by adopting mscrs and improved feature matching. *Multimedia Tools and Applications, 76*(20), 20483–20497.

Li, Y. (2013). Image copy-move forgery detection based on polar cosine transform and approximate nearest neighbor searching. *Forensic Science International, 224*(1–3), 59–67.

Lopez-Puigdollers, D., Traver, V. J., & Pla, F. (2019). Recognizing white blood cells with local image descriptors. *Expert Systems with Applications, 115*, 695–708.

Mahdian, B., & Saic, S. (2007). Detection of copy–move forgery using a method based on blur moment invariants. *Forensic Science International, 171*(2–3), 180–189.

Mahmood, T., Irtaza, A., Mehmood, Z., & Mahmood, M. T. (2017). Copy–move forgery detection through stationary wavelets and local binary pattern variance for forensic analysis in digital images. *Forensic Science International, 279*, 8–21.

Pal, N. R., Pal, K., Keller, J. M., & Bezdek, J. C. (2005). A possibilistic fuzzy c-means clustering algorithm. *IEEE Transactions on Fuzzy Systems, 13*(4), 517–530.

Pan, X., & Lyu, S. (2010). Region duplication detection using image feature matching. *IEEE Transactions on Information Forensics and Security, 5*(4), 857–867.

Pun, C.-M., Yuan, X.-C., & Bi, X.-L. (2015). Image forgery detection using adaptive oversegmentation and feature point matching. *IEEE Transactions on Information Forensics and Security, 10*(8), 1705–1716.

Ryu, S.-J., Kirchner, M., Lee, M.-J., & Lee, H.-K. (2013). Rotation invariant localization of duplicated image regions based on zernike moments. *IEEE Transactions on Information Forensics and Security, 8*(8), 1355–1370.

Silva, E., Carvalho, T., Ferreira, A., & Rocha, A. (2015). Going deeper into copy-move forgery detection: Exploring image telltales via multi-scale analysis and voting processes. *Journal of Visual Communication and Image Representation, 29*, 16–32.

Soni, B., Das, P. K., & Thounaojam, D. M. (2019). Geometric transformation invariant block based copy-move forgery detection using fast and efficient hybrid local features. *Journal of Information Security and Applications, 45*, 44–51.

Tian, T., Sethi, I., Ming, D., Zhang, Y., & Ma, J. (2013). Lipid: Local image permutation interval descriptor. In *International Conference on Machine Learning and Applications* (Vol. 2, pp. 513–518). IEEE.

Tian, T., Yang, F., Zheng, K., Yao, H., & Gao, Q. (2017). A fast local image descriptor based on patch quantization. In *International Conference on Human Centered Computing* (pp. 64–75). Springer.

Tralic, D., Zupancic, I., Grgic, S., & Grgic, M. (2013a). *Comofod dataset*. https://www.vcl.fer.hr/comofod/index.html.

Tralic, D., Zupancic, I., Grgic, S., & Grgic, M. (2013b). Comofod–new database for copy-move forgery detection. In *Proceedings ELMAR-2013* (pp. 49–54). IEEE.

Uliyan, D. M., Jalab, H. A., Wahab, A. W. A., Shivakumara, P., & Sadeghi, S. (2016). A novel forged blurred region detection system for image forensic applications. *Expert Systems with Applications, 64*, 1–10.

Wang, C., Zhang, Z., & Zhou, X. (2018a). An image copy-move forgery detection scheme based on a-kaze and surf features. *Symmetry, 10*(12), 706.

Wang, X.-Y., Jiao, L.-X., Wang, X.-B., Yang, H.-Y., & Niu, P.-P. (2018b). A new keypoint-based copy-move forgery detection for color image. *Applied Intelligence, 48*(10), 3630–3652.

Warif, N. B. A., Wahab, A. W. A., Idris, M. Y. I., Salleh, R., & Othman, F. (2017). Sift-symmetry: a robust detection method for copy-move forgery with reflection attack. *Journal of Visual Communication and Image Representation, 46*, 219–232.
Yang, F., Li, J., Lu, W., & Weng, J. (2017). Copy-move forgery detection based on hybrid features. *Engineering Applications of Artificial Intelligence, 59*, 73–83.
Yang, M.-S., & Nataliani, Y. (2017). Robust-learning fuzzy c-means clustering algorithm with unknown number of clusters. *Pattern Recognition, 71*, 45–59.
Yildirim, E. O., & Ulutaş, G. (2019). Augmented features to detect image splicing on swt domain. *Expert Systems with Applications, 131*, 81–93.
Zhu, Y., Shen, X., & Chen, H. (2016). Copy-move forgery detection based on scaled orb. *Multimedia Tools and Applications, 75*(6), 3221–3233.
Ziegler, A., Christiansen, E., Kriegman, D., & Belongie, S. J. (2012). Locally uniform comparison image descriptor. *Advances in Neural Information Processing Systems*, 1–9.