Copy-Move Forgery Detection Using Fast Retina Keypoint (FREAK) Descriptor

Ardeshir Ghasemi Yegane[1], Kourosh Kaini[1*], and Razieh Rastgoo[1]

***Abstract*— Image forgery, the manipulation of an image to hide some meaningful or helpful information, is widely used to manage the large amount of information being exchanged in the form of images. There are different forms of image forgery, and copy--move forgery is the most common form of it. The copy-move forgery is easy to perform yet challenging to detect due to the similarity between the original part of the image and the copied part. In this paper, we employ a keypoint descriptor inspired by the human visual system, namely the FREAK (Fast Retina Keypoint) descriptor, for robust copy-move forgery detection. This method uses the advantages of FREAK descriptor such as fast computing and low memory load compared to SIFT, SURF, and BRISK. Finally, geometric transformation parameters are extracted and discussed. Results confirm promising results in the case of image post-processing operations such as adding noise, illumination change, and geometric transformations such as rotation and scaling.**

***Index Terms*—** Copy-move forgery detection, Fast Retina Keypoint (FREAK), Keypoint descriptor.

I. INTRODUCTION

Nowadays, a large amount of information is being exchanged in the form of images/videos. Different methods in Computer Vision are used to process this information [1-9]. However, the manipulation of digital images has been straightforward because of the existence of powerful computers, advanced editing software packages, and high-resolution imaging devices [10]. Developing software, tools, and digital image processing techniques facilitated the use of image forgery techniques that are not simply detectable. To deal with this problem, digital forensics has been introduced, which are techniques to detect any type of forgery in digital images. Applications of forensics are in the fields of a court of law, criminal investigations, insurance, scientific claims, medical imaging, and so on [1,11-12]. Fig. 1 shows a typical image and its forged version.

Tampering methods are divided into three main classes: copy-move, splicing, and retouching. In copy-move tampering, one or more patches of an image are copied and moved to another location in the same image to duplicate some object or hide some other scenes of that image. In splicing forgeries, a part of one image is copied and moved to another image. In the case of retouching, some techniques are used to make some changes to images. Copy- move is one type of forgeries for which many detection methods have been proposed to solve. All forgery detection methods may be divided into two main categories: block-based and keypoint-based methods. In the block-based methods, the query image is divided into some overlapping blocks that differ from each other only by one row or one column. For each block, some features are extracted and then sorted. Neighbor feature vectors in the sorted matrix rely on features extracted from similar blocks. In keypoint-based methods, some important points of the image are detected, and features of these keypoints are extracted. After matching extracted features, duplicated patches can be found. Some forgery detection methods rely on keypoints, such as SIFT (Scale Invariant Feature Transform), SURF (Speeded Up Robust Features), BRISK (Binary Robust Invariant Scalable Keypoints), and the proposed keypoint descriptor, Fast Retina Keypoint (FREAK), introduced in [13].

The rest of this paper is organized as follows: In section II, the related works on the copy-move forgery detection field are introduced. In section III, the proposed method is introduced

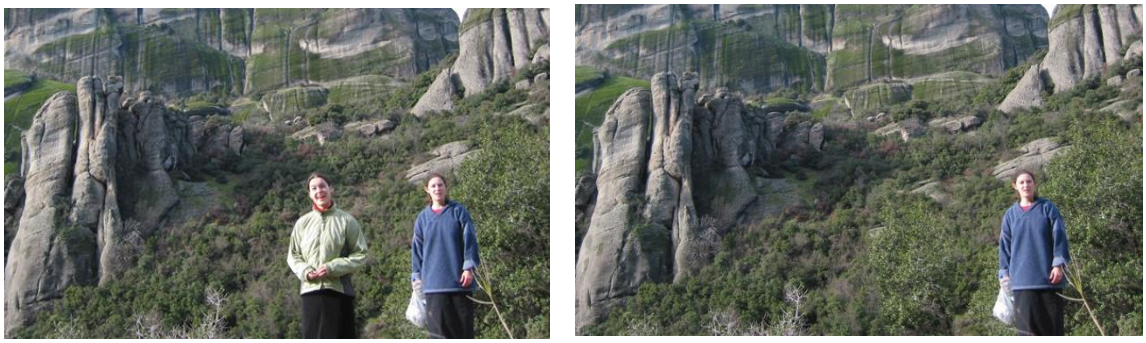

Fig. 1. (Left) A typical image, (Right) The tampered version of it.

1.Electrical and Computer Engineering Faculty, Semnan University, Semnan, Iran

Corresponding author: Kourosh.kiani@semnan.ac.ir

step by step. Section IV demonstrates the experimental results of the proposed method. Finally, the conclusions are drawn in section V.

II. RELATED WORKS

Many methods have been presented in the literature to overcome copy-move attacks. Some good studies can be found in [14-18]. The main idea of all methods proposed to solve the copy-move problem is that both copied and pasted patches have similarities in some properties that can be used as features. Using these features and searching for similar features may lead to discovering similar patches. A good forgery detection method should be able to detect duplicated patches. In addition, a good method must be stable against geometric transformations such as translation, rotation, and scaling as well as post-processing manipulations such as noise addition, JPEG compression, blurring, and so on. Almost the first attempt to solve the copy-move forgeries was proposed in [19]. It is a block-based approach that uses DCT as a feature extraction method and lexicographic as a sorting algorithm. Using a lexicographic sorting algorithm instead of the nearest neighbor algorithm was done to improve match finding and decrease running time. Several statistical methods for detecting copy-move attacks were proposed in [20, 21]. In [22], a method for reducing dimensions by applying Principle Component Analysis (PCA) to small, fixed-size blocks was proposed. This model is robust against additive noise and JPEG compression. In [23], four components—R, G, B, and Y—were used in blocks to obtain the energy distribution of luminance along with four different directions. Another method was proposed in [24] by using the singular value decomposition (SVD) for feature vector dimensionality reduction along with a discrete wavelet transform (DWT) for duplication detection and lexicographic sorting. This model is suitable for image compression and edge processing. A method for detecting duplicated patches was proposed in [25] that uses blur moment invariants, PCA, and a k-d tree. While this model uses 24 blur invariants up to the seventh order, this method suffers from high computational time. The method described in [26] uses Zernike moments that are rotation-invariant. Another method introduced in [27] uses a log-polar map for extracting descriptors into a 1-D vector and is invariant to reflection, rotation, and scaling.

Other methods extract descriptors using a relatively recent feature known as the Local Binary Pattern (LBP). LBP has some texture-related applications in image processing, such as texture categorization and copy-move forgery detection.

Local visual features such as SIFT [28], SURF [29], BRISK [30], BRIEF [31], and so on, are widely used in different image processing areas such as object recognition, object matching, and image retrieval because of their robustness to some geometric transformations such as translation, rotation, and scaling. In [32], a novel method using SIFT was introduced that is stable against changes in illumination, rotation, and scaling to detect duplicated patches. A method based on SIFT was proposed in [33] that estimates the geometric transformation parameters. This model shows a good true positive rate (TPR) in detecting duplicated regions. In this paper, we use a fast and robust descriptor, namely the FREAK descriptor. FREAK uses a comparison between pixel intensities and makes a binary string. Using this binary descriptor, copy-move forgery detection is done, and duplicated patches are discovered.

III. PROPOSED METHOD

In this paper, we propose a solution in the field of copy-move forgery detection based on extracting key point features using the FREAK descriptor. FREAK is a strong keypoint descriptor inspired by the human visual system, namely the retina. Fig. 2 depicts the proposed method's procedure. This approach works with grayscale images, so if the query image is not in grayscale format, it should be converted to grayscale. The proposed approach is separated into four major steps: (1) recognizing keypoints, (2) extracting keypoint descriptors, (3) matching extracted features to obtain forged areas, and (4) determining transform parameters using the RANSAC method.

A. Detecting keypoints

FREAK extracts keypoint descriptors from some pre-specified keypoints but it does not detect keypoints. Thus, we need some methods to detect robust keypoints to pass to the FREAK algorithm. Someone may select some random points but the result might not be satisfying. In the case of FREAK, we can use some keypoints detection methods such as corner points as keypoints. Other options may be using SURF or BRISK keypoints. There are some kinds of corner point detector algorithms in the literature [34, 35]. One of the most useful methods is the Harris algorithm that was introduced in [34].

The combination of the Harris corner detector as a keypoint detector and FREAK as a keypoint descriptor leads to good results. The other useful corner detector is FAST, which was introduced in [36]. In this paper, we use a combination of Harris corners and SURF points as keypoints in the proposed model. Some other keypoints, such as FAST, SURF, and BRISK keypoints, are used to make some comparisons.

B. Exracting keypoint descriptors

FREAK is a method to extract keypoint descriptors in the same way as the retina does in the human visual system. FREAK uses a pattern similar to that of retinal ganglion cells in the retina. This pattern is circular, and the density of the central points is higher than the peripheral ones. This grid pattern is illustrated in Fig. 3. Only 512 pairs of this grid are used and the rest are ignored. FREAK is a binary descriptor of size 512 bits, which means it makes a comparison between the intensities of each point of every pair in the pattern. Each bit is calculated by thresholding the difference between one pair's intensities that have been smoothed using a Gaussian kernel.

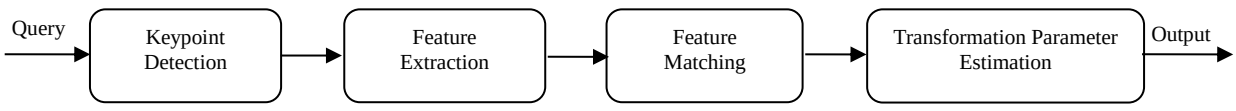

Fig. 2. A schematization of the proposed method.

Equation (1) demonstrates this concept:

$$F = \sum_{0 \le a < N} 2^a T(P_a) \quad (1)$$

Where P_a is a pair of points, N is the size of descriptor and T is thresholding function defined as:

$$T(P_a) = \begin{cases} 1 & if \ \left(I(P_a^{r_1}) - I(P_a^{r_2})\right) > 0 \\ 0 & otherwise \end{cases} \quad (2)$$

where $I(P_a^{r_1})$ and $I(P_a^{r_2})$ are smoothed intensities of two points in the pair P_a.

For each keypoint detected in the previous step, we extract a FREAK descriptor as described above. Now we have some binary descriptors, each of which is related to one keypoint in the image.

C. Matching features

In the copy-move forgery attacks, the copied patch has essentially similar characteristics to the original patch. Therefore, two similar patches in the same image have similar features extracted by the same method. So, matching features in all FREAK descriptors extracted from an image can reveal similar patches in the image.

With a query image, a set of n keypoints P={p_1,⋯,p_n } are extracted. For each keypoint, the corresponding descriptor is also extracted. All of these descriptors form a set of n descriptors D={d_1,⋯,d_n }. Given an individual feature d_i, the goal is to find the nearest neighbor feature in the feature space of n-1 remaining features. As we use the FREAK binary descriptor, the used distance function should be Hamming distance. The distance must be lower than a global threshold; T1 (matching threshold), to accept two features as similar. This method does not perform well because of some ambiguous matches that may be made. So, we use the procedure introduced in [27]. Assume that d_1 is the nearest neighbor and d_2 is the second nearest neighbor. Then, two features are tagged as similar, if the ratio between d_1 and d_2 is lower than threshold T2 (matching ratio). It means that the candidate features are matched only if the following constraint is fulfilled:

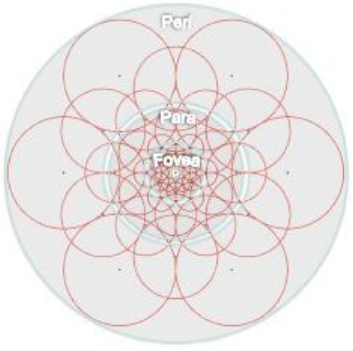

Fig. 3. Illustration of FREAK sampling pattern introduced in [1]

$$d_1/d_2 < T_2 \ where \ T_2 \in (0,1) \quad (3)$$

Using this procedure, ambiguous matches will be rejected. The two thresholds mentioned above play key roles in the number of discovered matches. In both cases, increasing the threshold leads to finding more matches. Both thresholds are in the interval of (0,1). In the case of both T1 and T2, increasing the threshold leads to an increment in discovered matched pairs but may also increase the false matches. Therefore, a tradeoff is needed.

Matching features using this method is called the "nearest neighbor ratio" method. This method is an instance of a lazy algorithm. Thus, it is a time-consuming method because of its many distance computing operations. If there are n feature vectors, the algorithm has an order of $O(n^2)$. Increasing the number of features leads to increase in the needed time for computing all distances in the order of two. So using fewer features makes the computing timeless. After matching all features, there may be some unwanted matched pairs. These pairs are outlier pairs and should be rejected. For this reason, an instance of the RANSAC algorithm is needed to discover the transformation parameters and reject the outliers. This step is described in the next section.

D. Determining geometric transform parameters

RANSAC (random sample consensus) is an iterative algorithm for estimating model parameters when the data is contaminated by unwanted data. RANSAC was first introduced by Fischler and Bolles in [30] in 1981 as a method to estimate the parameters of a model having a set of data contaminated by some unwanted data named as outliers. The RANSAC algorithm has some modifications. Despite these modifications, it is essentially made of two main steps that are repeated iteratively. These steps are known as “hypothesize and test framework.”

In the hypothesize step, a minimum sample set is randomly selected from the input data, and the model parameters are computed using these samples.

In the test framework step, the algorithm verifies which elements of the entire dataset fit in the model obtained in the previous step and inserts these elements into the model. The next iteration will be done with a new model.

These two steps run iteratively. RANSAC terminates when it cannot find a better model than the one found. In the copy-move forgery field, a patch of an image is copied and moved to another place in the same image. This transformation can be modeled as a geometric transformation. A geometric transformation is made of three mains simple transformations: translation, rotation, and scaling. Each of these main transformations has its own transform matrix and parameters. For a given point (x,y) in a 2D plane, the translation, rotation, and scaling matrices are as below, respectively:

$$T = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (4)$$

$$R = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \quad (5)$$

$$S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \quad (6)$$

The combination of these base transformation matrices can be written as a single matrix as follows:

$$H = \begin{bmatrix} A & T \\ 0^T & 1 \end{bmatrix} \quad (7)$$

The vector T is a translation vector as (4). Matrix A is the combination of rotation and scaling matrices in (5) and (6). For

any point (x,y), the transformation point is computed as:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H. \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (8)$$

In the case of copy-move attacks, the copied patch is usually transformed using a transformation according to (8). Using the RANSAC algorithm, only inliers are selected to form a model, and outlier points will be rejected. Also, transformation parameters can be obtained. Thus, only keypoints that have similar features and the same transformation parameters will remain.

In this paper, we use the RANSAC algorithm two times. For the first time, after matching features, RANSAC is applied to matched features to find some inlier points and their corresponding transformation parameters. There may be some matched pairs that can fit in that transformation, but their transformation directions are opposite and therefore are rejected in the RANSAC algorithm. We use the KNN(K-Nearest Neighbors) algorithm with k = 1 overall matched pairs to add these pairs to the discovered transformation. So, all matched points are classified into two classes based on their Euclidean distance from two center points of previously found inliers. Consequently, another instance of the RANSAC algorithm is applied to these two classes, and final inlier points and transformation parameters are achieved. This process makes the results better. An image will be tagged as "forged" if the number of final inlier points is at least four.

Fig. 4 demonstrates the step-by-step output of the proposed method over the query image mentioned in fig. 1 with T1=0.6 and T2=0.8. First, all keypoints and their corresponding descriptors are extracted (Fig 4-a). After matching extracted descriptors, only those keypoints that have a match in the rest of keypoints are remain (Fig 4-b). Matched keypoints can be shown as some connected pairs (Fig 4-c). In this step, there may be some matched outlier keypoints. After applying the RANSAC algorithm, the keypoint pairs that fit in a geometric transformation are extracted (Fig 4-d). Geometric transformation parameters are obtained in this step.

Fig. 4. Step-by-step output of the proposed method for a query image including four steps: First row – left: Detected keypoints, First row – right: Matched keypoints, Second row – left: Matched keypoint pairs, Second row – right: Matched keypoint pairs after applying RANSAC algorithm.

IV. EXPERIMENTAL RESULTS

In this section, we evaluate the proposed method over some test images. The used test dataset is MICC-F220, which was introduced in [27]. To make a comparison between different keypoint detector algorithms, we use some factors, such as the average number of detected keypoints and the average number of matched keypoints. This comparison can be found in Table I. To evaluate the proposed method against other methods, we define the True Positive Rate (TPR) and False Positive Rate (FPR) metrics. These metrics are defined as follows:

$$TPR = \frac{TP}{N_f} \quad (9)$$

$$FPR = \frac{FP}{N_o} \quad (10)$$

Where TP (True Positive) is the number of images tagged as forged being forged, N_f is the number of all forged images in the dataset, FP (False Positive) is the number of images tagged as forged being original and N_o is the number of all original images in the dataset.

Table II shows the results of various amounts for the Matching Threshold and Matching Ratio parameters when the detection method is a combination of Harris corners and SURF points. In this table, we compare TPR, FPR, and the average time needed for any single image as metrics. It is obvious that the best result is obtained when the matching threshold and

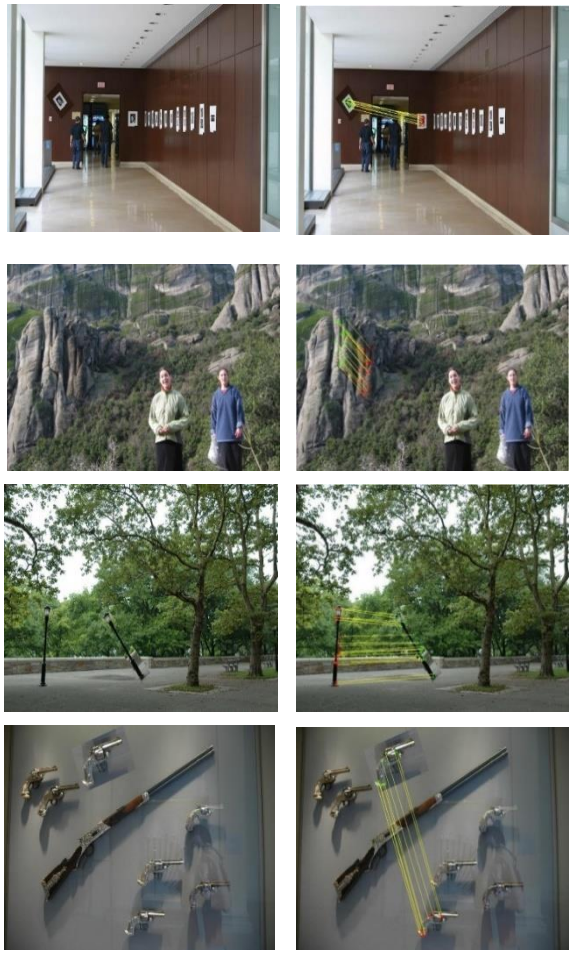

Fig.5. Some images with various post processing operations and their corresponding outputs.

Matching Ratio are 0.6 and 0.8, respectively. Table III shows the values used for parameters in the proposed model.

In Table IV, we compare four kinds of keypoint detection methods when the Matching Threshold and Matching Ratio are 0.6 and 0.8, respectively. We can see that the combination of Harris corners and SURF points has the best results.

Fig. 5 shows four typical images that have been tampered with by various post-processing operations such as rotation, scaling, illumination change, and a combination of them and their corresponding outputs. We can see that the duplicated area is easily found in all types of tampering.

Multiple tampering attacks can also be easily handled by the proposed method. Using an iterative operation, after finding one duplicated region, one set of found points should be removed from all found matches, and the same process must be done on the remaining points until no more patches can be found.

V. CONCLUSIONS

In this paper, we discussed copy-move forgery detection methods. A new method to discover image forgery detection based on the FREAK descriptor was proposed. This method combines Harris corners and SURF points and uses them as keypoints. Then FREAK descriptor is extracted for that keypoints and the matching phase is done. Using the proposed algorithm, duplicated patches are appeared, and transformation parameters are obtained. According to experimental results, the highest performance is obtained in the case of using Harris and SURF points as keypoints and matching threshold and matching ratio with values of 0.6 and 0.8, respectively. The main advantage of the proposed method is low running time and computational load. Therefore, this method can be used in cases where hardware specifications are low and the low run time is considered such as in smartphones. In addition, this method is invariant to some post-processing operations such as scaling, rotation, adding noise, illumination change, etc. In future works, we would like to employ deep learning-based models in the field.

TABLE I
Comparison Between Different Detection Methods on Detected Keypoints and Matched Points

Detection Method	Average Number of detected keypoints	Average Number of matched points
Harris	1218.47	6.95
FAST	1317.26	6.91
BRISK	598.40	2.76
Harris + SURF	**2098.15**	**10.07**

TABLE II
Comparison Between Different Matching Thresholds and Matching Ratio Amounts

(Matching Threshold, Matching Ratio)	True Positive Rate (%)	False Positive Rate (%)	Average Time(s)
(0.3, 0.6)	60	2.72	0.58
(0.3, 0.7)	78.18	7.27	0.79
(0.4, 0.7)	77.27	7.27	0.78
(0.5, 0.7)	77.27	9.09	0.79
(0.3, 0.8)	89.09	13.64	1.45
(0.4, 0.8)	89.09	14.54	1.49
(0.5, 0.8)	84.54	10.90	1.47
(0.6, 0.8)	**91.82**	**8.18**	**1.45**
(0.7, 0.8)	88.18	13.63	1.45
(0.8, 0.8)	89.09	11.81	1.36
(0.3, 0.9)	79.09	24.55	2.03
(0.4, 0.9)	82.72	27.27	1.99
(0.5, 0.9)	82.72	30.90	2.03
(0.6, 0.9)	76.36	31.81	2.03
(0.7, 0.9)	80.90	31.81	2.05

TABLE III
The Values Used for Model Parameters

Parameter	Value	Parameter	Value
Matching Threshold	0.3, 0.4, 0.5, 0.6	Matching Ratio	0.6, 0.7, 0.8, 0.9
T1	0.6	T2	0.8

TABLE IV
Comparison Between Different Detection Methods on TPR, FPR, and Average Time

Matching Threshold = 0.6, Matching Ratio = 0.8			
Detection Method	True Positive Rate	False Positive Rate	Average Time
Harris	75.45	4.5	0.93
FAST	64.54	8.18	0.93
BRISK	71.81	1.81	0.80
Harris + SURF	**91.82**	**8.18**	**1.45**

VI. REFERENCES

[1] Diwan, A., Sharma, R., Roy, A.K., Mitra, S.K. (2021). Keypoint-based comprehensive copy-move forgery detection. IET Image Process 15, pp. 1298–1309.

[2] Gan, Y., Zhong, J., Vong, Ch. (2022). A Novel Copy-Move Forgery Detection Algorithm via Feature Label Matching and Hierarchical Segmentation Filtering. Information Processing & Management 59, No. 102783.

[3] Li, F.F. (2022). Machine Learning in Computer Vision, https://www.cs.princeton.edu.

[4] Ouyang, J., Liu, Y., Liao, M. (2017). Copy-move forgery detection based on deep learning. 2017 10th International Congress on Image and Signal Processing, Bio-Medical Engineering and Informatics (CISP-BMEI), Shanghai, China.

[5] Rodriguez-Ortega, Y., Ballesteros, D.M., Renza, D. (2021). Copy-Move Forgery Detection (CMFD) Using Deep Learning for Image and Video Forensics, J Imaging. Vol. 7, no. 3.

[6] Liu, Y., Xia, Ch., Zhu, X., Xu, Sh. (2020). Two-Stage Copy-Move Forgery Detection with Self Deep Matching and Proposal Superglue, arXiv: 2012.08697.

[7] Zainal Abidin, A.B., Abdul Majid, H.B., Samah, A.B.A., Hashim, H.B. (2019). Copy-Move Image Forgery Detection Using Deep Learning Methods: A Review, 2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS), Johor Bahru, Malaysia.

[8] Kiani, K., Rezaeerad, S., Rastgoo, R. (2021). HMM-based Face Recognition Using SVD and Half of the Face Image, Journal of Modeling & Simulation in Electrical & Electronics Engineering (MSEEE), vol. 1, no. 2.

[9] Majidi, N., Kiani, K., Rastgoo, R. (2020). A deep model for super-resolution enhancement from a single image, Journal of AI and Data Mining, vol. 8, no. 4, pp. 451-460.

[10] Youssef, B., Atta, E. (2016). Image Forgery Detection using FREAK Binary Descriptor and Level Set Segmentation. International Journal of Scientific & Engineering Research 7.

[11] Sridevi, M., Aishwarya, S., Bokadia, D. (2019). Parallel Image Forgery Detection Using FREAK Descriptor. Information and Communication Technology for Intelligent Systems 107, pp. 619-630.

[12] Khudhair, Z.N., Mohamed, F., Kadhim, K.A. (2021). A Review on Copy-Move Image Forgery Detection Techniques. Journal of Physics: Conference Series.

[13] Alahi, A., Ortiz, R. & Vandergheynst, P. (2012). FREAK: Fast Retina Keypoint. Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 510–517.

[14] Warif, N et al. (2016). Copy-move forgery detection: Survey, challenges, and future directions. Journal of Network and Computer Applications, vol. 75, pp. 259–278.

[15] Birajdar, G.K. & Mankar, V.H. (2013). Digital image forgery detection using passive techniques: A survey. Digital Investigation, Vol. 10, no. 3, pp. 226-245.

[16] Sridevi M., Mala C. & Sanyam S. (2012). Comparative Study of Image Forgery and Copy-Move Techniques. In: Wyld D., Zizka J., Nagamalai D. (eds). Advances in Computer Science, Engineering & Applications. Advances in Intelligent and Soft Computing, vol. 166. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-30157-5_71.

[17] Meena, K.B. & Tyagi, V. (2019) Image Forgery Detection: Survey and Future Directions. In: Shukla R., Agrawal J., Sharma S., Singh Tomer G. (eds) Data, Engineering, and Applications. Springer, Singapore. https://doi.org/10.1007/978-981-13-6351-1_14.

[18] Al_Azrak, F.M., Sedik, A., Dessowky, M.I. et al. (2020). An efficient method for image forgery detection based on trigonometric transforms and deep learning. Multimed Tools Appl, vol. 79, pp. 18221–18243. https://doi.org/10.1007/s11042-019-08162-3.

[19] Fridrich, J., Soukal, D. & Lukas, J. (2003). Detection of copy-move forgery in the digital images. Proceeding of DFRWS, Cleveland, OH.

[20] Farid, H. (2003). A picture tells a thousand lies. New Scientist, 6 Sep 2003.

[21] Popescu, A. & Farid, H. (2005). Exposing Digital Forgeries in Color Filter Array Interpolated Images. IEEE Trans. on signal processing, vol. 53, no. 10, pp. 3948-3959.

[22] Popescu, A. & Farid, H. (2004). Exposing digital forgeries by detecting duplicated image regions. Dartmouth College, Computer Science, Tech. Rep. TR2004-515.

[23] Luo, W., Huang, J. & Qiu, G. (2006). Robust detection of region-duplication forgery in digital image. Proc. of ICPR, Washington, USA.

[24] Li, G., Wu, Q., Tu, D. & Sun, S.J. (2007). A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD. Proc. of IEEE ICME, Beijing, China.

[25] Mahdian, B & Saic, S. (2007). Detection of copy-move forgery using a method based on blur moment invariants. Forensic Science International, vol. 171, no. 2-3, pp. 180–189.

[26] Ryu, S.J., Lee, M.-J., Lee, H.-K. (2010). Detection of copy-rotate-move forgery using Zernike moments. Proc. of International Workshop on Information Hiding, Calgary, Canada.

[27] Bravo, S. & Nandi, A.K. (2011). Automated detection and localization of duplicated regions affected by reflection, rotation, and scaling in image forensics. Signal Processing, vol. 91, pp. 1759–70.

[28] D. Lowe. (1999). Object recognition from local scale-invariant features., 1999. The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157.

[29] Bay, H., Tuytelaars, T. & Van Gool. T. (2006). SURF: Speeded up robust features. ECCV, Graz, Austria, pp. 404–417.

[30] Leutenegger, S., Chli, M. & Siegwart, R. (2011). Brisk: Binary robust invariant scalable keypoints. 2011 International Conference on Computer Vision,

Barcelona, Spain.

[31] Calonder, M., Lepetit, V., Strecha, C. & Fua, P. (2010). Brief: Binary robust independent elementary features. ECCV, Heraklion, Crete, Greece, pp. 778–792.

[32] Huang, H., Guo, W. & Zhang, Y. (2008). Detection of copy-move forgery in digital images using SIFT algorithm. Proc. of IEEE Pacific-Asia Workshop on Computational Intell. and Industrial Application, Wuhan, China.

[33] Irene, L., Caldelli, R., Del, A. & Serra, G. (2011). A SIFT-based forensic method for copy-move attack detection and transformation recovery. IEEE Trans Inf Forensics Security, vol. 6, pp. 1099–110.

[34] Harris, C. & Stephens, M. (1988). A combined corner and edge detector. In Alvey vision conference, Manchester, UK, vol. 15, pp. 147-151.

[35] Rosten, E. & Drummond, T. (2006). Machine learning for high-speed corner detection. ECCV, Graz, Austria, vol. 1, pp. 430–443.

[36] Fischler, M. & Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, vol. 24, no. 6, pp. 381–395.